Confidentiality level: external use

UM32G421 User Manual

V1.3.1

Unicmicro (Guangzhou) Co., Ltd.

http://www.unicmicro.com/

Copyright & Disclaimer

Contents

List of Tables

List of Figures

1 Documentation Conventions

1.1 List of Abbreviations for Registers

The following abbreviations are used in register descriptions:

Read/write (R/W)	Software can read and write to this bit.
Read-only (R)	Software can only read this bit.
Write-only (W)	Software can only write to this bit, and reading this bit returns the reset value.
Read/clear write 1 (R/W1C)	Software can read as well as clear this bit by writing 1. Writing 0 has no effect on the bit value.
Read/clear write 0 (R/W0C)	Software can read as well as clear this bit by writing 0. Writing 1 has no effect on the bit value.
Reserved (RSV)	This register bit is reserved and has no function.

1.2 Glossary

This section outlines the definitions of acronyms and abbreviations used in this document:

- In this document, the Cortex-M4 core with FPU is referred to as Cortex-M4F.
- The CPU core integrates two debug ports:
  - JTAG debug port (JTAG-DP) provides a 5-pin standard interface based on the joint test action group (JTAG) protocol.
  - SWD debug port (SWD-DP) provides a 2-pin (clock and data) interface based on the serial wire debug (SWD) protocol.
- Word: data/instruction of 32-bit length
- Half-word: data/instruction of 16-bit length
- Byte: data of 8-bit length
- Double-word: data of 64-bit length

- IAP (in-application programming): IAP is the ability to re-program the Flash memory of a microcontroller while the user program is running.
- ICP (in-circuit programming): ICP is the ability to program the Flash memory of a microcontroller using the JTAG protocol, the SWD protocol or the bootloader while the device is mounted on the user application board.
- I-Code: this bus connects the instruction bus of the CPU core to the Flash memory instruction interface. Prefetching is performed on this bus.
- D-Code: this bus connects the D-Code bus (literal load and debug access) of CPU to the Flash memory data interface.
- Option bytes: production configuration bits stored in the Flash memory
- OBL: option byte loader
- AHB: advanced high-performance bus
- CPU: Cortex-M4F

2 Product Introduction

2.1 Overview

The UM32x42x series are general microprocessor chips based on the ARM® Cortex® -M4 core with high performance and low power. The M4 core implements a full set of DSP (digital signal processing) instructions and features a floating-point unit (FPU) and a memory protection unit (MPU). The system clock frequency is up to 168 MHz (204 MHz @ boost mode). The maximum capacity of the internal Flash is 256 KB, and that of SRAM is 64 KB. They are available at a wide supply voltage of 1.8 V to 3.6 V and in the industrial temperature range from -40℃ to 105℃.

The UM32x42x series chips feature rich peripherals, including 1 x USB FS Device interface, 2 × 12-bit high-speed ADCs, 1 × 12-bit DAC, internal temperature sensor, 3 × comparators, 3 × OPAs, 2 × USARTs, 4 × UARTs, 3 × SPIs, 3 × I2C interfaces, 1 × I2S interfaces, 2 × IWDTs, 1 × CAN-FD bus interface, 12 × counters/timers (advanced control timer and general timer), 2 x LPUARTs, 2 × LPTIMs, 1 x 32-bit RTC and counter, GPIO with up to 54 channels. They also integrate hardware Cordic accelerator (supporting sin, cos, arctan, square root, multiplication, division, etc.), encryption/decryption engine (including AES etc.), and one RNG that can generate random key.

Applications:

- Motor
- Smart door locks
- IoT, etc.

2.2 Main Features

- **ARM Cortex™-M4F core with high performance**
  - Main frequency up to 168 MHz (204 MHz @ boost mode)
  - Memory protection unit (MPU)
  - 16 programmable interrupt priority levels
  - IEEE 754 compliant FPU
  - DSP instruction extension with 32-bit single-cycle hardware divider
  - 256B Flash and 64KB SRAM (8KB of which can be used as backup SRAM)
  - One DMA controller with 8 channels in total
  - Online debugging using SWD/JTAG protocol
- **Bootloader**
  - Supporting ISP and IAP for Flash memory
- **Flexible clock source**
  - External crystal input: 1 MHz–48 MHz
  - Internal 96 MHz RCH oscillator
  - Internal 32 kHz RCL oscillator
  - Internal PLL (supporting integer-N, fractional-N and spread spectrum modes)
  - External 32.768 kHz XTL and 32 kHz RTC
- **Analog peripherals**
  - 2 x 12-bit successive ADCs of up to 5.25 MSPS, with 16 external channels and 4 internal channels (connected with OPAx for measuring VDDH / VBAT; with internal 1.2 V voltage reference) Supporting single-ended and differential mode inputs
  - 1 x 12-bit 1 MSPS DAC with buffer
  - 3 x rapid analog comparators (ACMP)
  - 3 x operational amplifiers (OPA) supporting single-ended PGA or comparator mode

- Internal 1.5 / 2 / 2.5 / 3 V voltage reference (VREF)
- Internal temperature sensor (TS)

- **Hardware acceleration co-processor**
  - CORDIC co-processor supports functions as follows: $m \cdot \sin\theta$, $m \cdot \cos\theta$, atan2(y, x), $\sqrt{x^2+y^2}$, $y \cdot x$, y/x, $\sinh w$, $\cosh w$, $tanh^{-1}(y/x)$, ln(x), $\sqrt{x}$,etc.

- **Rich communication interfaces**
  - Up to 4 x UARTs, of which UART1 is enhanced UART, supporting IrDA and 9-bit data communication
  - 2 x USARTs supporting UART / SPI / IrDA / LIN
  - 2 x LPUARTs
  - 3 x I2Cs (up to 1 Mbps), supporting SMBus
  - 1 x I2S, supporting I2S / PCM
  - Up to 3 x universal SPIs, of which SPI2 is enhanced SPI, supporting 4–32-bit data communication
  - 1 x CAN industrial bus, supporting CANFD with communication rate up to 5 Mbps, compatible with CAN2.0A/B
  - Up to 37 x PWM outputs
  - 1 x full-speed USB20 device interface (with PHY)

- **Encryption engine**
  - Supporting AES 128/256
  - A random number generator (RNG) can generate random key.

- **Timer**
  - 2 x 16-bit ATimers (eQCT™: TIM0 and TIM7), including a 16-bit auto-reload counter and a programmable prescaler; supporting phase-shifting of two-phase hardware, input capture, output compare and PWM output (complementary PWM with dead-

time insertion); supporting system clock multiplication accuracy. Each ATimer supports 4 PWM outputs; 8 PWM outputs in total.

- 6 x 16-bit general-purpose timers (TIM1–4, TIM8–9), supporting system clock multiplication accuracy. Each supports 4 PWM outputs; 24 PWM outputs in total.
- 3 x 16-bit general timers (TIM14 / 15 / 16), each with one input capture and two complementary PWM outputs with dead-time insertion; 3 PWM outputs in total.
- 1 x 16-bit basic timers (TIM5)
- 2 x 16-bit low-power timers (LPTIM0–1), with 2 PWM outputs in total.
- 1 x 24-bit SysTick timer

- **Real-time clock (RTC)**
  - Support calendar (with seconds, minutes, hours, weekday, date, month and year in BCD format), and calibration
- **GPIO**
  - Up to 54 GPIOs
  - Up to 28 pins support 5 V withstand voltage.
- **Power Management**
  - Low-power modes: Sleep, Stop, Standby and DeepStandby
  - Born-out reset (BOR) and low-voltage detection (LVD), with under-voltage interrupt and forced reset generated from two sets of detection points respectively
  - Power-on reset (POR) and power-down reset (PDR)
  - Integrated power management unit (PMU)
- **Electrical characteristics**
  - Operating voltage: 1.8–3.6 V
  - Operating temperature: -40–+105℃
- **Security**

- Anti-copy board to prevent programs in eFlash from being pirated
- CRC16-CCITT / CRC32 hardware accelerator
- Write protection, multiple read protection
- External clock stop monitoring
- 128-bit UUID

3 Memory and Bus Architecture

3.1 System Architecture

The main system consists of 32-bit multilayer AHB bus matrix that interconnects:

- **Four masters:**
  - Cortex™-M4F with core I-bus, D-bus and S-bus
  - DMA0
- **Six slaves:**
  - Internal Flash memory on I/D-code bus
  - Main internal SRAM0 (32 KB)
  - Auxiliary internal SRAM1 (32 KB)
  - AHB0 peripheral
  - AHB2APB0 peripheral
  - AHB2APB1 peripheral

3.2 Bus Architecture Diagram

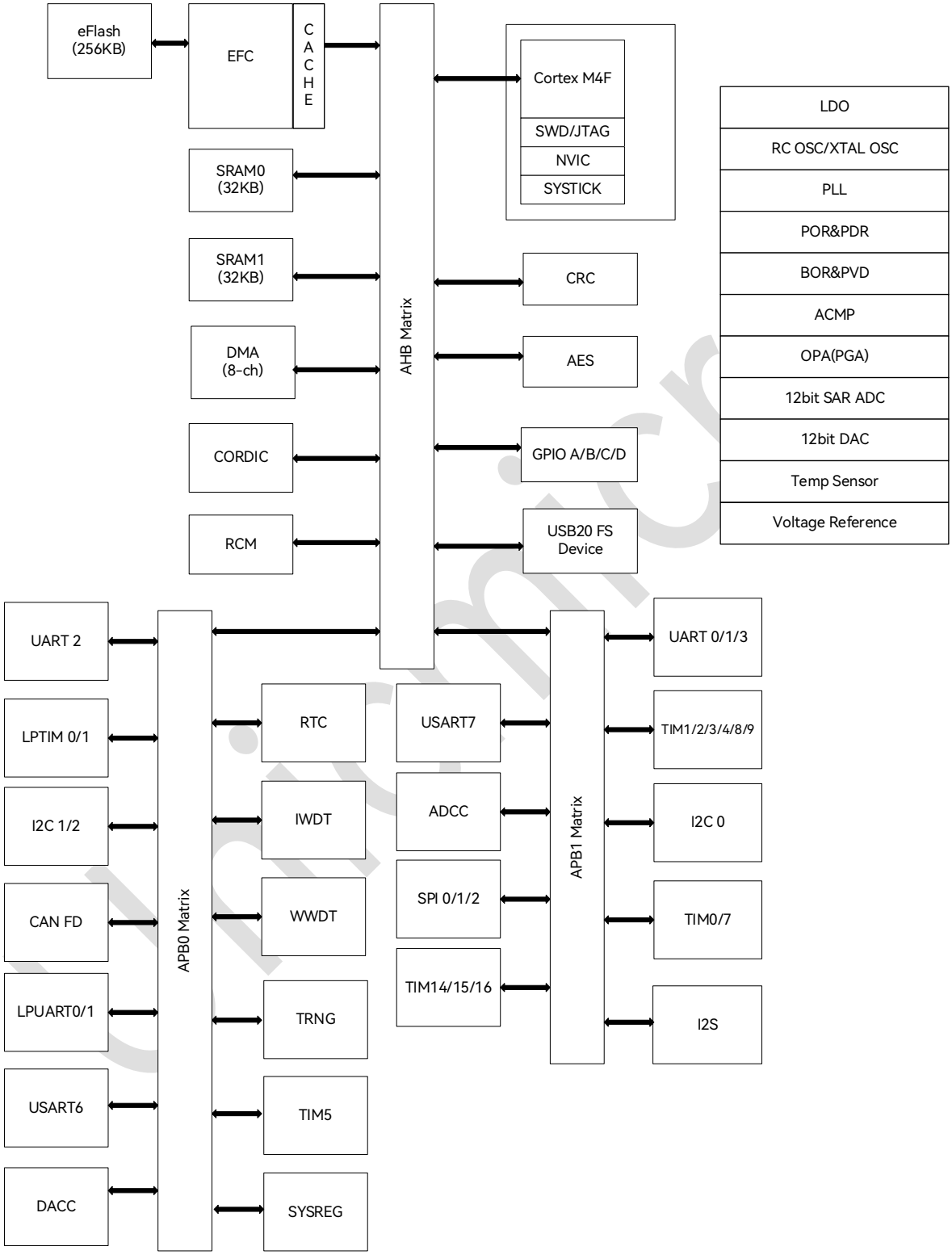


Figure 3-1: Bus Architecture Diagram

3.3 Memory Mapping

Program memory, data memory, registers and I/O ports are organized within the same linear 4-GB address space.

The bytes are coded in memory in little-endian format. The lowest numbered byte in a word is considered the word's least significant byte and the highest numbered byte the most significant.

For the detailed mapping of peripheral registers, please refer to the related chapters.

All the memory map areas that are not allocated to on-chip memories and peripherals are considered "reserved".

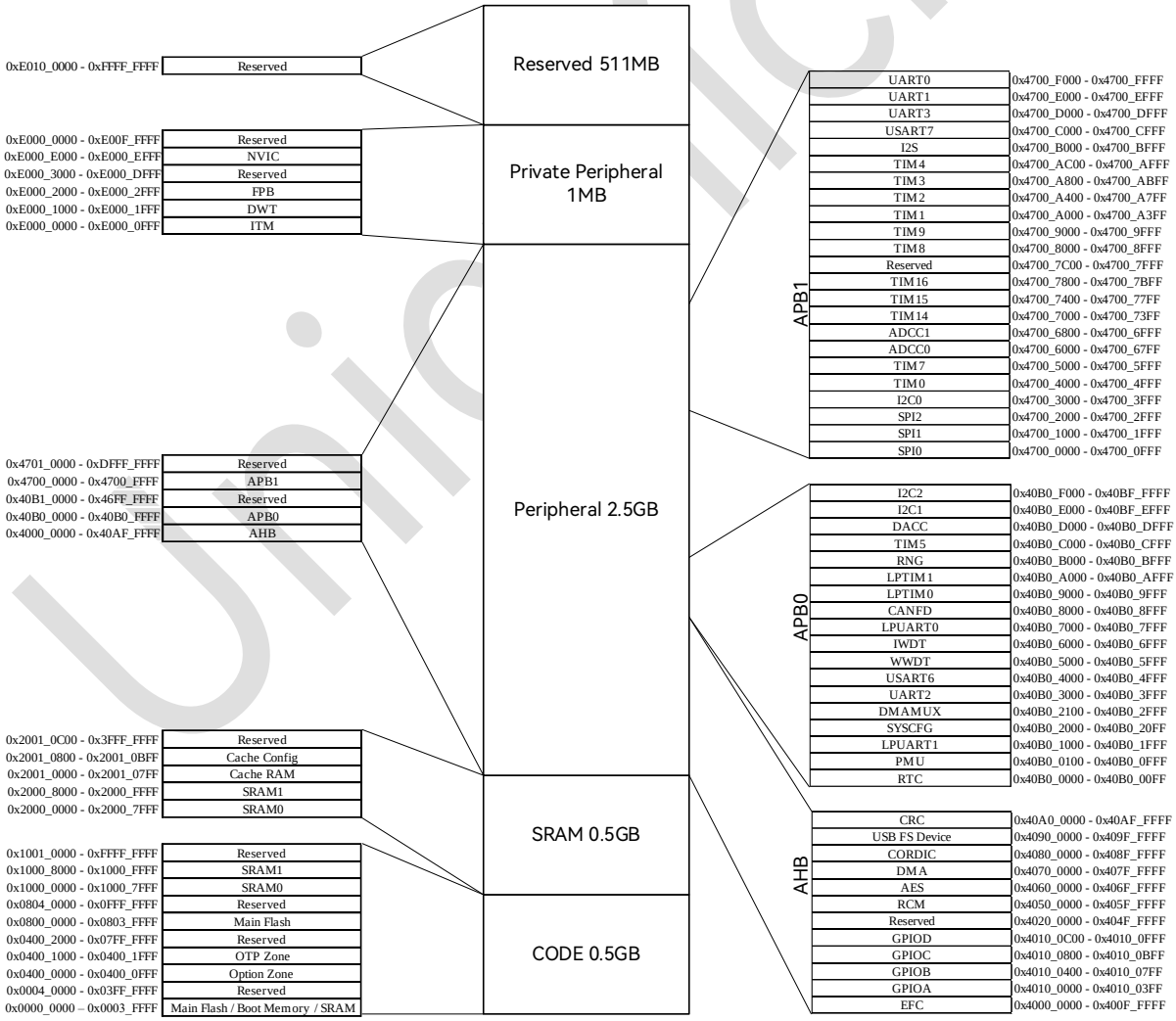


Figure 3-2: Memory Address Mapping

Table 3-1: Memory Remapping Address

Block / Module	Boundary Address	Size
Main FLASH 1	0x0000_0000–0x0003_FFFF 0x0800_0000–0x0803_FFFF	256 KB
SRAM0	0x2000_0000–0x2000_7FFF 0x1000_0000–0x1000_7FFF	32 KB
SRAM1	0x2000_8000–0x2000_FFFF 0x1000_8000–0x1000_FFFF	32 KB
SRAM1 (BACKUP)	0x2000_E000–0x2000_FFFF	8 KB
Cache Config	0x2001_0800–0x2001_0BFF	1 KB

Table 3-2: Memory Mapping Address

Block / Module		Boundary Address
AHB	EFC	0x4000_0000–0x400F_FFFF
	GPIOA	0x4010_0000–0x4010_03FF
	GPIOB	0x4010_0400–0x4010_07FF
	GPIOC	0x4010_0800–0x4010_0BFF
	GPIOD	0x4010_0C00–0x4010_0FFF
	RCM	0x4050_0000–0x405F_FFFF
	AES	0x4060_0000–0x406F_FFFF
	DMA	0x4070_0000–0x407F_FFFF
	CORDIC	0x4080_0000–0x408F_FFFF
	USB FS Device	0x4090_0000–0x409F_FFFF
	CRC	0x40A0_0000–0x40AF_FFFF
APB0	RTC	0x40B0_0000–0x40B0_00FF
	PMU	0x40B0_0100–0x40B0_0FFF
	LPUART1	0x40B0_1000–0x40B0_1FFF
	SYSCFG	0x40B0_2000–0x40B0_20FF
	DMAMUX	0x40B0_2100–0x40B0_2FFF
	UART2	0x40B0_3000–0x40B0_3FFF
	USART6	0x40B0_4000–0x40B0_4FFF
	WWDT	0x40B0_5000–0x40B0_5FFF
	IWDT	0x40B0_6000–0x40B0_6FFF
	LPUART0	0x40B0_7000–0x40B0_7FFF
	CANFD	0x40B0_8000–0x40B0_8FFF
	LPTIM0	0x40B0_9000–0x40B0_9FFF

Block / Module		Boundary Address
	LPTIM1	0x40B0_A000-0x40B0_AFFF
	RNG	0x40B0_B000-0x40B0_BFFF
	TIM5	0x40B0_C000-0x40B0_CFFF
	DACC	0x40B0_D000-0x40B0_DFFF
	I2C1	0x40B0_E000-0x40B0_EFFF
	I2C2	0x40B0_F000-0x40B0_FFFF
APB1	SPI0	0x4700_0000-0x4700_0FFF
	SPI1	0x4700_1000-0x4700_1FFF
	SPI2	0x4700_2000-0x4700_2FFF
	I2C0	0x4700_3000-0x4700_3FFF
	TIM0	0x4700_4000-0x4700_4FFF
	TIM7	0x4700_5000-0x4700_5FFF
	ADCC0	0x4700_6000-0x4700_67FF
	ADCC1	0x4700_6800-0x4700_6FFF
	TIM14	0x4700_7000-0x4700_73FF
	TIM15	0x4700_7400-0x4700_77FF
	TIM16	0x4700_7800-0x4700_7BFF
	TIM8	0x4700_8000-0x4700_8FFF
	TIM9	0x4700_9000-0x4700_9FFF
	TIM1	0x4700_A000-0x4700_A3FF
	TIM2	0x4700_A400-0x4700_A7FF
	TIM3	0x4700_A800-0x4700_ABFF
	TIM4	0x4700_AC00-0x4700_AFFF
	I2S	0x4700_B000-0x4700_BFFF
	USART7	0x4700_C000-0x4700_CFFF
	UART3	0x4700_D000-0x4700_DFFF
	UART1	0x4700_E000-0x4700_EFFF
	UART0	0x4700_F000-0x4700_FFFF

4 Memory System

4.1 FLASH

4.1.1 Overview

The Flash memory module features the capacity of up to 256 KB, and the Flash controller (EFC) performs read, write, erase and other operations on EFlash with the cooperation of CPU.

4.1.2 Main Features

- 8/16/32-bit Flash memory read operation, 32-bit program operation
- On-chip Flash with a capacity of 256 KB has 64 pages in total, each of 4 KB
- Support page erase and chip erase
- Flash controller supports continuous programming, saving time when programming data into a 512-byte aligned memory.
- Configurable read waiting time
- Erase/write protection
- Automatic bus locking
- Automatic read-back verification after programming
- OTP (one-time programmable) area
- Support instruction cache

4.1.3 Functional Description

Flash controller can perform read, write, erase and other operations.

It requires a frequency greater than 1 MHz for erasing. The user shall configure the EFC_TIME register according to the actual operating frequency.

Flash can be programmed with 32-bit wide data, which shall be unlocked before programming.

Flash controller supports continuous programming, saving time when programming data into a 512-byte aligned memory. Continuous programming shall be performed on memory other than this Flash.

The controller features erase verification function that can automatically check whether all data bits in the Flash are changed to 1 in page erase and mass erase modes.

4.1.3.1 Flash Bank 1 Address Mapping

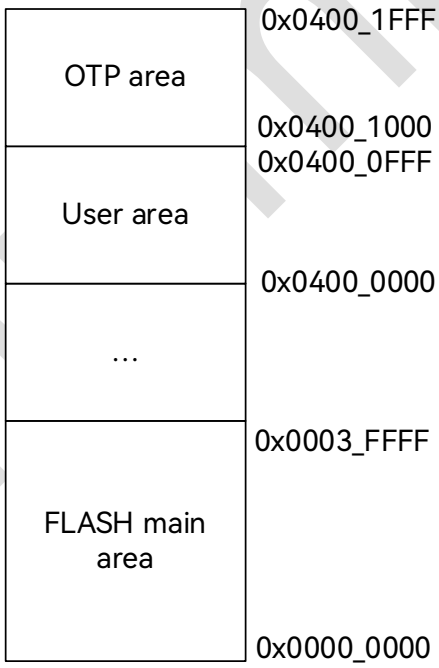


Figure 4-1: EFC Memory Address Mapping

4.1.3.2 User Option Byte

The user option page contains addresses from 0x0400_0000 to 0x0400_0FFF, some of which have special features, as shown in the following table:

Table 4-1: EFC Special Feature Address Description

Name	Address	Bit Width	Content
IWDT_HW_START	0x04000F90	8 bits	Automatically start IWDT after the chip is booted: 0xAB: automatically start IWDT Other: do not start IWDT automatically
WRITE_PROTECT	0x04000FA0	32 bits	Disable erasing of pages in the specified numbering interval (0–63) in the main Flash memory: Bit[31:24]: A5: write protection activated; others: write protection not activated. Bit[23:14]: reserved Bit[13:8]: ending number of specified protection page Bit[7:6]: reserved Bit[5:0]: starting number of specified protection page
DEBUG_DISABLE	0x04000FB0	8 bits	Disable JTAG or SWG debug port operation: 0xAB: debug port disabled, JTAG will not be recognized Others: debug port operation not disabled
READ_PROTECT	0x04000FC0	8 bits	Writing different specific values sets different levels of protection. The following levels of protection are supported: • No protection • Low-level protection • High-level protection

4.1.3.3 One-time Programmable (OTP) Area

Table 4-2: OTP Area of EFC

No.	OTP Storage Area Address	Size	OTP Lock Control Address	Remarks
0	0x04001000–0x040010FF	256 bytes	0x04001F80	-
1	0x04001100–0x040011FF	256 bytes	0x04001F84	-
2	0x04001200–0x040012FF	256 bytes	0x04001F88	-
3	0x04001300–0x040013FF	256 bytes	0x04001F8C	-
4	0x04001400–0x040014FF	256 bytes	0x04001F90	-

No.	OTP Storage Area Address	Size	OTP Lock Control Address	Remarks
5	0x04001500–0x040015FF	256 bytes	0x04001F94	-
6	0x04001600–0x040016FF	256 bytes	0x04001F98	-
7	0x04001700–0x040017FF	256 bytes	0x04001F9C	-
8	0x04001800–0x040018FF	256 bytes	0x04001FA0	-
9	0x04001900–0x040019FF	256 bytes	0x04001FA4	-
10	0x04001A00–0x04001AFF	256 bytes	0x04001FA8	-
11	0x04001B00–0x04001BFF	256 bytes	0x04001FAC	-
12	0x04001C00–0x04001CFF	256 bytes	0x04001FB0	-
13	0x04001D00–0x04001DFF	256 bytes	-	Reserved area, unavailable
14	0x04001E00–0x04001EFF	256 bytes	-	Reserved area, unavailable
15	0x04001F00–0x04001F7F	128 bytes	-	Reserved area, unavailable
16	0x04001F80–0x04001FFF	128 bytes	-	OTP lock control address

4.1.4 Register Description

Register base address: 0x4000_0000

The registers are listed below:

Table 4-3: List of EFC Registers

Offset Address	Name	Description
0x00	EFC_CTRL	Control register
0x04	EFC_TIME	Timing register
0x08	EFC_SEC	Security register
0x0C	EFC_STATUS	Status register
0x10	EFC_INTSTATUS	Interrupt status register
0x14	EFC_INTEN	Interrupt enable register
0x18	EFC_LPCR	Low-power control register
0x1C	EFC_ADDRREC	Last operation address register

Registers are detailed in the following sections.

4.1.4.1 Control Register (EFC_CTRL)

Offset address: 0x00

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:6	RSV	-	-	Reserved
5	ERASEVE	R/W	0x0	Erase verification enable: 1: enabled 0: disabled
4	PROGVE	R/W	0x0	Programming verification enable (valid only in single programming): 1: enabled 0: disabled
3	MERASES	R/W	0x0	Mass erase mode (erase all pages in main memory) selection bit: 1: selected 0: not selected
2	PERASES	R/W	0x0	Page erase mode selection bit: 1: selected 0: not selected
1	CONPROGS	R/W	0x0	Continuous programming mode selection bit: 1: selected 0: not selected
0	PROGS	R/W	0x0	Single programming mode selection bit: 1: selected 0: not selected

Note: Only one of bits 0–3 can be selected.

4.1.4.2 Timing Register (EFC_TIME)

Offset address: 0x04

Reset value: 0x0000 6018

Bit	Name	Attribute	Reset Value	Description
31:24	REGWE	W	0x0	This register is writable when this field is written with A5.

Bit	Name	Attribute	Reset Value	Description
				Please write 0xA50XXXXX to this register once.
23:18	RSV	-	-	Reserved
17:16	RECYC	R/W	0x0	Waiting time between two reads
15:12	RWAITCYC	R/W	6	Setting bit of read wait cycle, which can be large, refer to the table below
11:9	RSV	-	-	Reserved
8:0	FREQ	R/W	24	Erase time scale, please fill in (EFC operating frequency - 1), in MHz The minimum value is 0, which corresponds to the minimum frequency of 1 MHz required for erasing. (Withstand voltage: -10%~+30%)

Table 4-4: Relationship between RWAITCYC (Read Wait Cycle) and Frequency

Frequency/MHz	Read Wait Cycle
< 50	0
50~100	1
100~150	2
150~200	3
200~204	4
…	…

4.1.4.3 Security Register (EFC_SEC)

Offset address: 0x08

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:0	UNLOCK	R/W	0x0	This register shall be unlocked when programming/erasing Flash. Write 0x55AAAA55 to unlock this register, so that 1 can be read from this register. It will be relocked when writing other values or when programming/erasing starts.

4.1.4.4 Status Register (EFC_STATUS)

Offset address: 0x0C

Reset value: 0x0000 0001

Bit	Name	Attribute	Reset Value	Description
31:9	RSV	-	-	Reserved
8	VDDHS	R	0x0	Status indication bit of supply voltage VDDH (LVD): 0: normal status 1: low voltage warning Note: LVD status register
7:6	RSV	-	-	Reserved
5:4	LPS	R	0x0	Low-power mode status indication bit: 11: being in power-down mode 10: being in sleep mode 01: operating at low voltage 00: operating at high voltage
3	RSV	-	-	Reserved
2	CONPROGRDY	R/W	0x0	Flash continuous programming status indication bit (writing 0 to this bit can exit the continuous programming mode): 1: waiting for the next continuous programming 0: not in the state of waiting for the next continuous programming
1	RSV	-	-	Reserved
0	RDYS	R	0x1	Flash status indication bit: 1: Flash being idle 0: Flash being busy

4.1.4.5 Interrupt Status Register (EFC_INTSTATUS)

Offset address: 0x10

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:4	RSV	-	-	Reserved

Bit	Name	Attribute	Reset Value	Description
3	EVFS	R/W1C	0x0	Erase verification failure status bit: Writing 1 clears this bit. If interrupt is allowed, an interrupt will be generated. 1: erase verification failed, not all data in Flash is 1 0: no erase verification failure occurred
2	PVFS	R/W1C	0x0	Write verification failure status bit: Writing 1 clears this bit. If interrupt is allowed, an interrupt will be generated. 1: write verification failed, the data in Flash does not match the written data. 0: no write verification failure occurred
1	VDDHLS	R/W1C	0x0	Low VDDH (LVD) interrupt status bit: Writing 1 clears this bit. If interrupt is allowed, an interrupt will be generated. 1: low VDDH warning occurred 0: normal status
0	OPDS	R/W1C	0x0	Programming/erasing done interrupt status bit: Writing 1 clears this bit. If interrupt is allowed, an interrupt will be generated. 1: programming/erasing done 0: programming/erasing undone

4.1.4.6 Interrupt Enable Register (EFC_INTEN)

Offset address: 0x14

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:4	RSV	-	-	Reserved
3	EVFE	R/W	0x0	Erasing verification failure interrupt enable: 1: enabled 0: disabled
2	PVFE	R/W	0x0	Writing verification failure interrupt enable: 1: enabled 0: disabled

Bit	Name	Attribute	Reset Value	Description
1	VDDLE	R/W	0x0	Low VDDH interrupt enable: 1: enabled 0: disabled
0	OPDE	R/W	0x0	Programming/erasing done interrupt enable: 1: enabled 0: disabled

4.1.4.7 Low-power Control Register (EFC_LPCR)

Offset address: 0x18

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:24	REGWE	W	0x0	This register is writable when this field is written with A5. Please write 0xA500000X to this register once.
23:1	RSV	-	-	Reserved
0	MODES	R/W	0x0	Selection bit for Flash behavior when the system enters low-power mode: 1: entering power-down mode 0: entering sleep mode Note: In stop mode, both modes can be selected. In standby mode, it is recommended to be set as 1.

4.1.4.8 Last Operation Address Register (EFC_ADDRREC)

Offset address: 0x1C

Reset value: 0x0C07 FFFF

Bit	Name	Attribute	Reset Value	Description
31:27	RSV	-	-	Reserved
26:0	ADDR	R	0xC07FFFF	Record the address where the last erasing or programming was completed.

4.1.5 Operation Procedure

4.1.5.1 Timing Setting

When using Flash, no matter what operation is performed, it is required to set the EFC_TIME register first to make the control value in the correct range.

1. Confirm according to the system frequency that the value (system frequency - 1) to be entered into the FREQ field is correct before using the programming, erasing, and low-power mode functions (using low-power mode does not require the system frequency to be greater than 1 MHz, but just fill in 0 if it is lower than 1 MHz).
2. Confirm according to the system frequency that the value to be entered into the RWAITCYC (Read_Wait_Cycle) is not less than the required value. To increase the system frequency, the EFC_TIME register shall be modified before increasing the system frequency; while to decrease the system frequency, the system frequency shall be decreased before modifying the EFC_TIME register.
3. Writing the EFC_TIME register once requires a combination of writing eigenvalues to the REGWE (Reg_Wr_Enable) field.

4.1.5.2 Single Programming

1. Operate __set_PRIMASK() to disable the general interrupt.
2. Configure the EFC_CTRL register to select single programming mode with the option to enable programming verification.
3. Unlock the EFC_SEC register and write 0x55AAAA55 at a time.
4. Write programming data to the address to be programmed.
5. Wait for EFC_STATUS[0] to be idle.
6. Reset the EFC_CTRL register to its default value.
7. Operate __set_PRIMASK() to enable the general interrupt.

8. Operate the EFC_SEC register to lock the Flash and write 0x55AA0000 at a time.

4.1.5.3 Continuous Programming

Programming can be carried out multiple times within the range of 512 bytes, and the control program can be really accelerated only when it is running in memory other than Flash.

1. Configure the EFC_CTRL register to select continuous programming mode.
2. Unlock the EFC_SEC register and write 0x55AAAA55 at a time.
3. Write programming data to the address to be programmed.
4. Repeat steps 2 and 3 as required.
5. Upon completion of the last programming, write 0 to the EFC_STATUS register to end continuous programming and prevent waiting in continuous programming mode.
6. Reset the EFC_CTRL register to its default value.
7. Operate the EFC_SEC register to lock the Flash and write 0x55AA0000 at a time.

4.1.5.4 Page Erase

1. Operate __set_PRIMASK() to disable the general interrupt.
2. Configure the EFC_CTRL register to select page erase mode.
3. Unlock the EFC_SEC register and write 0x55AAAA55 at a time.
4. Write arbitrary data to any address in the page to be erased.
5. Wait for EFC_STATUS[0] to be idle.
6. Reset the EFC_CTRL register to its default value.
7. Operate the EFC_SEC register to lock the Flash and write 0x55AA0000 at a time.
8. Operate __set_PRIMASK() to enable the general interrupt.

4.1.5.5 Mass Erase

Mass erase will erase all pages in the Flash main memory.

1. Operate __set_PRIMASK() to disable the general interrupt.
2. Configure the EFC_CTRL register to select mass erase mode.
3. Unlock the EFC_SEC register and write 0x55AAAA55 at a time.
4. Write arbitrary data to any address in the main memory.
5. Wait for EFC_STATUS[0] to be idle.
6. Reset the EFC_CTRL register to its default value.
7. Operate the EFC_SEC register to lock the Flash and write 0x55AA0000 at a time.
8. Operate __set_PRIMASK() to enable the general interrupt.

4.2 SRAM

4.2.1 Overview

SRAM is mainly used for code running, storing variables and data or stacks during program execution, with a maximum capacity of 64 KB (including SRAM0 and SRAM1 with a capacity of 32 KB respectively, supporting 0-wait access) and a maximum address range of 0x2000_0000−0x2000_FFFF.

4.2.2 Main Features

- Can be accessed as bytes, half-words or full-words
- DMA available

4.3 CACHE

The cache controller can cache the data fetched from Flash and accelerate the next read for higher system performance. This cache is instruction cache (I-Cache) with a size of 2 KB. It is optionally to enable cache line prefetch, and the cache memory will be automatically initialized when the cache is activated.

4.3.1 Register Description

The registers controlled by I-Cache are as follows:

I-Cache register base address: 0x2001_0800

Table 4-5: List of Cache Register

Offset Address	Name	Description
0x00	CACHE_CTRL	Control register

Registers are detailed in the following sections.

4.3.1.1 Control Register (CACHE_CTRL)

Offset address: 0x00

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:2	RSV	-	-	Reserved
1	CACHE_CLR	W	0x0	Writing 1 clears all cached contents in the Cache, and each time the Cache is enabled, it is required to write 1 to this bit. 1: clear all cached data in Cache 0: do not clear cached data in Cache
0	CACHE_EN	R/W	0x0	Cache enable: 0: disabled 1: enabled

4.3.2 Operation Procedure

4.3.2.1 Enabling Cache

To enable Cache, write 1 to the CACHE_EN and CACHE_CLR bits.

When CACHE_EN is not 1, i.e., Cache is not enabled, Cache Data Space can be used as system SRAM. When CACHE_EN is 1, a data error will occur when reading or writing the Cache Data Space.

5 Power Management Unit (PMU)

5.1 Chip Power Supply

The chip is supplied with single power supply and VBAT backup power supply. It requires an external operating supply voltage between 1.8 V and 3.6 V, and a digital circuit operating voltage generated by the built-in LDO.

The system power supply scheme is shown below.

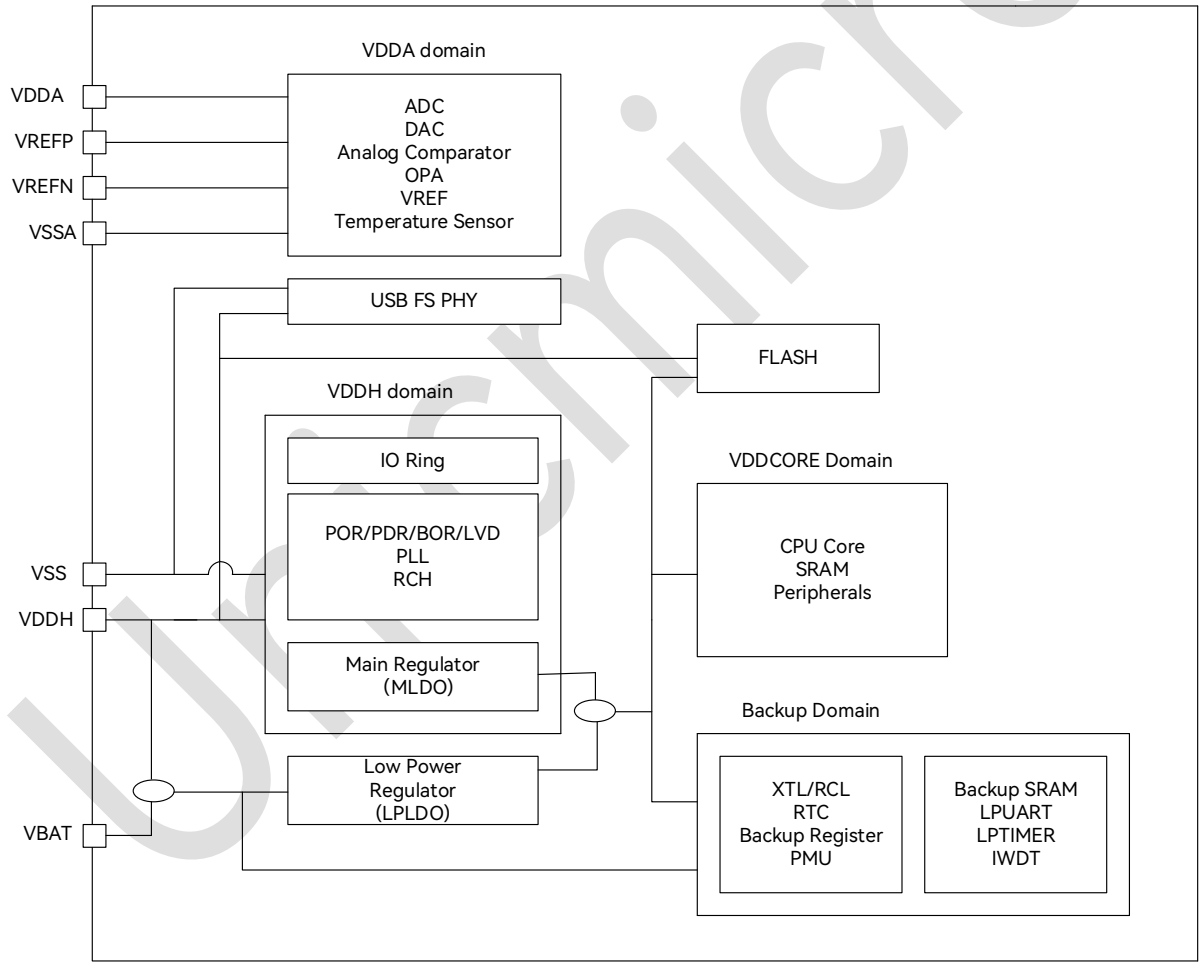


Figure 5-1: System Power Supply Scheme

5.2 Internal PMU Module Function

5.2.1 Introduction to Internal PMU Module

The chip provides different power consumption modes based on application scenarios to reduce the overall power consumption.

- Run mode
- Sleep mode
- Stop mode
- Standby mode
- DeepStandby mode

The low-power modes and wake-up information are shown in the following table:

Table 5-1: Low-power Modes Summary

Power Mode	Description	Entry Condition	Wakeup Source
Run mode	All supplies are powered on, and the high-speed clock remains active.	-	-
Sleep mode	All supplies are powered on, with M4 core high-speed clock off and other peripherals on.	1. PMU_STANDBY_EN = 0, PMU_MODE[1:0] = 00, PMU_BKSRAMOFF = 0 2. EFC_Sys_Mode = 0 3. WFI/WFE	1. Wakeup on interrupt of any GPIO 2. Wakeup on interrupt of any peripheral 3. Wakeup on reset (RESETN, IWDT reset)
Stop mode	All supplies are powered on, with high-speed clock off and 32 K low-speed clock on.	1. PMU_STANDBY_EN = 0, PMU_MODE[1:0] = 01, PMU_BKSRAMOFF = 0 2. EFC_Sys_Mode = 0 3. Set SLEEPDEEP = 1 for M4 core. 4. WFI/WFE	1. Wakeup on interrupt of any GPIO 2. Wakeup on interrupt of PC13 (tamper) 3. RTC wakeup 4. Wakeup on IWDT reset or interrupt 5. External LPUART wakeup 6. LPTIM0–1 wakeup

Power Mode	Description	Entry Condition	Wakeup Source
Standby0 mode	The CORE domain is powered off, the BBU domain is kept on, and the 32 K low-speed clock remains active. BKSRAM / IWDT / LPTIM / LPUART are powered on.	1. PMU_STANDBY_EN = 1, PMU_MODE[1:0] = 10, PMU_BKSRAMOFF = 0 2. EFC_Sys_Mode = 1 3. Set SLEEPDEEP = 1 for M4 core. 4. WFI/WFE	1. xternal pins PA0, PA2, PC0, PC2 and PC3 2. Wakeup on interrupt of PC13 (tamper) 3. RTC wakeup 4. LPTIM0−1 wakeup 5. LPUART (only PC2 pin wakeup) 6. Wakeup on IWDT reset or interrupt 7. Wakeup on reset (RESETN)
Standby1 mode	The CORE domain is powered off, the BBU domain is kept on, and the 32 K low-speed clock remains active. BKSRAM / IWDT / LPTIM / LPUART are powered off.	1. PMU_STANDBY_EN = 1, PMU_MODE[1:0] = 10, PMU_BKSRAMOFF = 1 2. EFC_Sys_Mode = 1 3. Set SLEEPDEEP = 1 for M4 core. 4. WFI/WFE	1. Wakeup on external pins of PA0, PA2, PC0, PC2, PC3 and PC13 2. Wakeup on interrupt of PC13 (tamper) 3. Wakeup on interrupt of RTC 4. Wakeup on reset (RESETN)
DeepStandby0 mode	The CORE domain is powered off, the BBU domain is kept on, and the 32 K low-speed clock is inactive. BKSRAM / IWDT / LPTIM / LPUART are powered on.	1. PMU_STANDBY_EN = 1, PMU_MODE[1:0] = 11, PMU_BKSRAMOFF = 0 2. EFC_Sys_Mode = 1 3. Set SLEEPDEEP = 1 for M4 core. 4. WFI/WFE	1. Wakeup on external pins of PA0, PA2, PC0, PC2, PC3 and PC13 2. Wakeup on reset (RESETN)
DeepStandby1 mode	The CORE domain is powered off, the BBU domain is kept on, and the 32 K low-speed clock is inactive. BKSRAM / IWDT / LPTIM / LPUART are powered off.	1. PMU_STANDBY_EN = 1, PMU_MODE[1:0] = 11, PMU_BKSRAMOFF = 1 2. EFC_Sys_Mode = 1 3. Set SLEEPDEEP = 1 for M4 core. 4. WFI/WFE	1. Wakeup on external pins of PA0, PA2, PC0, PC2, PC3 and PC13 2. Wakeup on reset (RESETN)
Power-off mode	All supplies are powered off.	Power off external VDDH & VBAT.	Power on

Notes:

- BBU domain includes RTC, backup register and PMU logic.
- In standby mode, I/O status hold is optional. During wakeup, the I/O pins return to the power-on reset state (most of the I/O pins return to the high-impedance state).

5.2.2 Run Mode

In run mode, according to different application scenarios, it is also possible to reduce power consumption by reducing the operating frequency or turning off some unused peripherals.

5.2.3 Sleep Mode

In Sleep mode, CPU stops working and the interrupt handling function is retained. The clock and reset of other peripheral modules can be set by software. This low-power mode is entered by the CPU by executing the specific instruction WFI, and the wake-up is triggered by interrupt.

5.2.4 Stop Mode

In Stop mode, the system RCH/XTH/PLL stop working (RCL/XTL remain active) and the interrupt handling function is retained. The clock and reset of other peripheral modules can be set by software. This low-power mode is entered by the CPU by executing the specific instruction WFI, and the wake-up is triggered by interrupt.

5.2.5 Standby Mode

In Standby mode, RTC keeps running, LPUART/LPTIM0−1 can be configured as active or inactive, and BKSRAM can be configured to be powered off or not.

5.2.6 DeepStandby Mode

In DeepStandby mode, the internal low-speed clock stops working, the backup register is retained, and BKSRAM can be powered off or retained.

5.3 Register Description

Note: If XTL or RCL is selected as the system clock, the register of this PMU is not accessible.

Register base address: 0x40B0_0100

The PMU registers are listed below:

Table 5-2: List of PMU Registers

Offset Address	Name	Description
0x00	PMU_MR	PMU mode register
0x04	PMU_PDWKCR	Power-down wakeup control register
0x08	PMU_PUSTCR	Power-up settling time configuration register
0x0C	PMU_VDCR	PDR/BOR/LVD configuration register
0x14	PMU_SASR	System auxiliary status register
0x1C	PMU_XTLCR	XTL configuration register
0x20	PMU_RCLCR	RCL configuration register
0x24	PMU_FCCR	Function clock control register
0x28	PMU_FRCR	Function reset control register
0x50	PMU_CPR	Configuration protection register

5.3.1 PMU Mode Register (PMU_MR)

Offset address: 0x00

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:21	RSV	-	-	Reserved
20	IO_KEEP	R/W	0	IO state (in standby mode): 0: analog high-impedance state 1: IO remains in the state before power-down (but will change after wake-up)
19	LPTIM0_TRIGEN	R/W	0	LPTIM0 TRIG input selection: 0: configured by the system in normal run mode; input from PC3 in Standby mode 1: fixed input from PC3
18	LPTIM0_INEN	R/W	0	LPTIM0 IN input selection: 0: configured by the system in normal run mode; input from PC0 in Standby mode

Bit	Name	Attribute	Reset Value	Description
				1: fixed input from PC0
17	LPUART1_INEN	R/W	0	LPUART1 IN input selection: 0: configured by the system in normal run mode; input from PA2 in Standby mode 1: fixed input from PA2
16	LPUART0_INEN	R/W	0	LPUART0 IN input selection: 0: configured by the system in normal run mode; input from PC2 in Standby mode 1: fixed input from PC2
15:9	RSV	-	-	Reserved
8	VOLTAGE_SEL	R/W	0	MLDO output voltage selection: 0: 1.1-V output (adopting trimming value at 1.1 V) 1: 0.9-V output (adopting trimming value at 0.9 V)
7:5	RSV	-	-	Reserved
4	STDBY_EN	R/W	0	Standby mode enable: 0: Standby mode disabled 1: Standby mode enabled
3	STOP_CLK_SEL	R/W	0	Stop mode clock selection: 0: when entering Stop mode, switch to RCH first, and still use RCH on wakeup 1: keep the working clock before entering Stop mode unchanged (e.g. PLL clock)
2	BKSRAMOFF	R/W	0	BKSRAM and other logics power-down control: 0: BKSRAM, IWDT, LPUART0–1 and LPTIM0–1 not powered down 0: BKSRAM, IWDT, LPUART0–1 and LPTIM0–1 powered down
1:0	PMU_MODE	R/W	0	PMU mode register: 2'b00: Run mode 2'b01: Stop mode 2'b10: Standby mode/Power-down

Bit	Name	Attribute	Reset Value	Description
				mode 2'b11: DeepStandby mode/Deep power-down mode

5.3.2 Power-down Wakeup Control Register (PMU_PDWKCR)

Offset address: 0x04

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31	IO_WK_CLR	W	-	Clear the wakeup status bits of PA0/PA2/PC0/PC2/PC3/PC13. Note: Before clearing the wakeup status bit, it is required to make the wakeup enable bit invalid first, and then write 1 to clear all wakeup status flag bits.
30	RESETN_WK_FLAG	R	0	External RESETN wakeup status bit (valid only in Standby & DeepStandby modes): 0: not woken up by the RESETN pin 1: woken up by the RESETN pin (in the case of single-pin wakeup), or possibly woken up by the RESETN pin (in the case of multiple-pin wakeup) Note: This bit is cleared by writing 1 to bit [31].
29	PC13_WK_FLAG	R	0	PC13 wakeup status bit (valid only in Standby & DeepStandby modes): 0: not woken up by PC13 1: woken up by PC13 (in the case of single-pin wakeup), or possibly woken up by PC13 (in the case of multiple-pin wakeup) Note: This bit is cleared by writing 1 to bit [31].
28	PC3_WK_FLAG	R	0	PC3 wakeup status bit (valid only in Standby & DeepStandby modes): 0: not woken up by PC3 1: woken up by PC3 (in the case of single-pin wakeup), or possibly woken up by PC3 (in the

Bit	Name	Attribute	Reset Value	Description
				case of multiple-pin wakeup) Note: This bit is cleared by writing 1 to bit [31].
27	PC2_WK_FLAG	R	0	PC2 wakeup status bit (valid only in Standby & DeepStandby modes): 0: not woken up by PC2 1: woken up by PC2 (in the case of single-pin wakeup), or possibly woken up by PC2 (in the case of multiple-pin wakeup) Note: This bit is cleared by writing 1 to bit [31].
26	PC0_WK_FLAG	R	0	PC0 wakeup status bit (valid only in Standby & DeepStandby modes): 0: not woken up by PC0 1: woken up by PC0 (in the case of single-pin wakeup), or possibly woken up by PC0 (in the case of multiple-pin wakeup) Note: This bit is cleared by writing 1 to bit [31].
25	PA2_WK_FLAG	R	0	PA2 wakeup status bit (valid only in Standby & DeepStandby modes): 0: not woken up by PA2 1: woken up by PA2 (in the case of single-pin wakeup), or possibly woken up by PA2 (in the case of multiple-pin wakeup) Note: This bit is cleared by writing 1 to bit [31].
24	PA0_WK_FLAG	R	0	PA0 wakeup status bit (valid only in Standby & DeepStandby modes): 0: not woken up by PA0 1: woken up by PA0 (in the case of single-pin wakeup), or possibly woken up by PA0 (in the case of multiple-pin wakeup) Note: This bit is cleared by writing 1 to bit [31].
23:22	RSV	-	-	Reserved
21	PC13_WKEG	R/W	0	PC13 wakeup edge selection: 0: rising-edge trigger 1: falling-edge trigger
20	PC3_WKEG	R/W	0	PC3 wakeup edge selection: 0: rising-edge trigger

Bit	Name	Attribute	Reset Value	Description
				1: falling-edge trigger
19	PC2_WKEG	R/W	0	PC2 wakeup edge selection: 0: rising-edge trigger 1: falling-edge trigger
18	PC0_WKEG	R/W	0	PC0 wakeup edge selection: 0: rising-edge trigger 1: falling-edge trigger
17	PA2_WKEG	R/W	0	PA2 wakeup edge selection: 0: rising-edge trigger 1: falling-edge trigger
16	PA0_WKEG	R/W	0	PA0 wakeup edge selection: 0: rising-edge trigger 1: falling-edge trigger
15:14	RSV	-	-	Reserved
13	LPUART1_WEK	R/W	0	LPUART1 interrupt wakeup enable: 0: wakeup event invalid 1: wakeup event valid
12	PC13_WKE	R/W	0	PC13 interrupt wakeup enable or TAMPER pin input enable: 0: wakeup event invalid or TAMPER pin input disabled 1: wakeup event valid or TAMPER pin input enabled
11	PC3_WKE	R/W	0	PC3 interrupt wakeup enable: 0: wakeup event invalid 1: wakeup event valid
10	PC2_WKE	R/W	0	PC2 interrupt wakeup enable: 0: wakeup event invalid 1: wakeup event valid
9	PC0_WKE	R/W	0	PC0 interrupt wakeup enable or TAMPER pin input enable: 0: wakeup event invalid or TAMPER pin input disabled 1: wakeup event valid or TAMPER pin input enabled
8	PA2_WKE	R/W	0	PA2 interrupt wakeup enable or TAMPER pin

Bit	Name	Attribute	Reset Value	Description
				input enable: 0: wakeup event invalid or TAMPER pin input disabled 1: wakeup event valid or TAMPER pin input enabled
7	LPTIM1_WKE	R/W	0	LPTIM1 interrupt wakeup enable: 0: wakeup event invalid 1: wakeup event valid
6	LPTIM0_WKE	R/W	0	LPTIM0 interrupt wakeup enable: 0: wakeup event invalid 1: wakeup event valid
5	LPUART0_WKE	R/W	0	LPUART0 interrupt wakeup enable: 0: wakeup event invalid 1: wakeup event valid
4	IWDT_WKE	R/W	0	IWDT interrupt wakeup enable: 0: wakeup event invalid 1: wakeup event valid
3	RTC_TAMP_WKE	R/W	0	RTC TAMPER interrupt wakeup enable: 0: wakeup event invalid 1: wakeup event valid Note: In DeepStandby mode, it is woken up upon PC13 interrupt event.
2	RTC_ALARM_WKE	R/W	0	RTC ALARM interrupt wakeup enable: 0: wakeup event invalid 1: wakeup event valid
1	PA0_WKE	R/W	0	PA0 reset wakeup enable or TAMPER pin input enable: 0: wakeup event invalid or TAMPER pin input disabled 1: wakeup event valid or TAMPER input pin enabled
0	RSTN_WKE	R/W	0	External pin reset wakeup enable (falling-edge trigger): 0: wakeup event invalid 1: wakeup event valid

5.3.3 Power-up Settling Time Configuration Register (PMU_PUSTCR)

Offset address: 0x08

Reset value: 0x0000 007F

Bit	Name	Attribute	Reset Value	Description
31:7	RSV	-	-	Reserved
6:0	CNT_VALUE	R/W	7'h7F	MLDO settling time setting: The default value is 128 clock cycles of 32 K (around 4 ms), and it can be set to 3 or less in the actual application.

5.3.4 PDR/BOR/LVD Configuration Register (PMU_VDCR)

Offset address: 0x0C

Reset value: 0x0000 A701

Bit	Name	Attribute	Reset Value	Description
31:24	RSV	-	-	Reserved
23	EN_PSWT	R/W	0	PSWT test selection signal: 0: no PSW test signal output 1: output PSW test signal Note: The output voltage of PSW is observed at the PA0 pin, which shall be configured for analog pin function.
22:18	RSV	-	-	Reserved
17	LVD_INT_EN	R/W	0	LVD interrupt enable register: 0: disabled 1: enabled
16	LVD_RST_EN	R/W	0	LVD reset enable register: 0: disabled 1: enabled
15:12	LVDS	R/W	4'hA	LVD settings Refer to Table 5-3
11:8	BORS	R/W	4'h7	BOR settings Refer to Table 5-4

Bit	Name	Attribute	Reset Value	Description
7:6	PDRS	R/W	2'b00	PDR settings Refer to Table 5-5 Note: It is not recommended to modify this register.
5	RSV	-	-	Reserved
4	LVD_FILTER_EN	R/W	0	LVD filter enable bit: 0: disabled 1: enabled (2 clock cycles of 32 K)
3	RSV	R/W	0	Reserved
2	LVD_EN	R/W	0	Low-voltage detection enable: 0: disabled 1: enabled
1	BOR_EN	R/W	0	Brown-out reset control: 0: BOR detection disabled, no reset generated 1: BOR detection enabled, reset generated Note: This bit must be set to 1 during use to enable the BOR detection reset.
0	PDR_EN	R/W	1	Power-down reset control: 0: PDR detection disabled, no reset generated 1: PDR detection enabled, reset generated Note: This bit shall be held at 1 in run mode, and setting it to 0 is only permissible in test mode.

Table 5-3 : LVD (VDT) Settings

Parameter	Symbol	Condition	Min.	Typ.	Max.	Unit
Operating current	-	-	-	40	-	nA
LVD trigger point	V_{LVD}	LVDS[3:0] = 0000	-	1.59	-	V
		LVDS[3:0] = 0001	-	1.68	-	V
		LVDS[3:0] = 0010	-	1.78	-	V
		LVDS[3:0] = 0011	-	1.88	-	V
		LVDS[3:0] = 0100	-	1.98	-	V
		LVDS[3:0] = 0101	-	2.08	-	V
		LVDS[3:0] = 0110	-	2.18	-	V
		LVDS[3:0] = 0111	-	2.28	-	V

Parameter	Symbol	Condition	Min.	Typ.	Max.	Unit
		LVDS[3:0] = 1000	-	2.38	-	V
		LVDS[3:0] = 1001	-	2.48	-	V
		LVDS[3:0] = 1010	-	2.58	-	V
		LVDS[3:0] = 1011	-	2.67	-	V
		LVDS[3:0] = 1100	-	2.77	-	V
		LVDS[3:0] = 1101	-	2.87	-	V
		LVDS[3:0] = 1110	-	2.97	-	V
		LVDS[3:0] = 1111	-	3.07	-	V

Table 5-4: BOR Settings

Parameter	Symbol	Condition	Min.	Typ.	Max.	Unit
Operating current	-	-	-	40	-	nA
BOR trigger point	V_{BOR}	BORS[3:0] = 0000	-	1.56	-	V
		BORS[3:0] = 0001	-	1.66	-	V
		BORS[3:0] = 0010	-	1.75	-	V
		BORS[3:0] = 0011	-	1.85	-	V
		BORS[3:0] = 0100	-	1.95	-	V
		BORS[3:0] = 0101	-	2.05	-	V
		BORS[3:0] = 0110	-	2.14	-	V
		BORS[3:0] = 0111	-	2.24	-	V
		BORS[3:0] = 1000	-	2.34	-	V
		BORS[3:0] = 1001	-	2.43	-	V
		BORS[3:0] = 1010	-	2.53	-	V
		BORS[3:0] = 1011	-	2.63	-	V
		BORS[3:0] = 1100	-	2.73	-	V
		BORS[3:0] = 1101	-	2.83	-	V
		BORS[3:0] = 1110	-	2.92	-	V
		BORS[3:0] = 1111	-	3.02	-	V

Table 5-5: PDR Settings

Parameter	Symbol	Condition	Min.	Typ.	Max.	Unit
Operating current	-	-	-	40	-	nA
PDR rising trigger point	V_{PDR_R}	PDRS[1:0] = 00	-	1.57	-	V
		PDRS[1:0] = 01	-	1.77	-	V
		PDRS[1:0] = 10	-	1.87	-	V
		PDRS[1:0] = 11	-	1.97	-	V

Parameter	Symbol	Condition	Min.	Typ.	Max.	Unit
PDR falling trigger point	V_{PDR_F}	PDRS[1:0] = 00	-	1.49	-	V
		PDRS[1:0] = 01	-	1.67	-	V
		PDRS[1:0] = 10	-	1.77	-	V
		PDRS[1:0] = 11	-	1.87	-	V
PDR trigger point hysteresis voltage	V_{PDR_HY}	PDRS[1:0] = 00	-	0.08	-	V
		PDRS[1:0] = 01	-	0.1	-	V
		PDRS[1:0] = 10	-	0.1	-	V
		PDRS[1:0] = 11	-	0.1	-	V

5.3.5 System Auxiliary Status Register (PMU_SASR)

Offset address: 0x14

Reset value: 0x0000 0234

Bit	Name	Attribute	Reset Value	Description
31:26	RSV	-	-	Reserved
25:24	TAMPERPIN_SEL	R/W	0	TAPMER pin input selection register: 00: select PC13 as TAMPER pin 01: select PA0 as TAMPER pin 10: select PA2 as TAMPER pin 11: select PC0 as TAMPER pin
23:22	RSV	-	0	Reserved
21	PC5_ADCEN	R/W	0	PC5 pin function selection bit: 0: support digital function (support 5-V withstand voltage input) 1: support ADC analog function (not support 5-V withstand voltage input)
20	PC4_ADCEN	R/W	0	PC4 pin function selection bit: 0: support digital function (support 5-V withstand voltage input) 1: support ADC analog function (not support 5-V withstand voltage input)
19	RSV	-	-	Reserved
18:17	VBAT_DIV_SEL	R/W	0	VBAT voltage divider selection: 00: 1/4 VBAT 01: 1/3 VBAT

Bit	Name	Attribute	Reset Value	Description
				10: 1/2 VBAT 11: 2/3 VBAT
16	VBAT_DIV_EN	R/W	0	VBAT voltage divider function enable: 0: disabled 1: enabled
15:14	RSV	-	-	Reserved
13	LVD_FLAG	R/W	0	LVD reset status flag bit: 0: no reset occurred 1: reset occurred Note: This bit is cleared by writing 1 to the RST_FLAG_CLR bit, and is active only when LVD reset occurs on PMU.
12	BOR_FLAG	R/W	0	BOR reset status flag bit: 0: no reset occurred 1: reset occurred Note: This bit is cleared by writing 1 to the RST_FLAG_CLR bit, and is active only when BOR reset occurs on PMU.
11	RSV	-	-	Reserved
10	XTL_RSTN_FLAG	R	0	XTL exception status flag bit: 0: no exception occurred 1: exception occurred Notes: 1) This bit is cleared by writing 1 to the RST_FLAG_CLR bit. 2) The XTL detection circuit function is recommended to be enabled only when XTL is used as the BBU module clock. After reset, the clock automatically switches to RCL. In addition, this status will be sent to the CPU core as an interrupt signal, sharing the same interrupt number with the RTC TAMPER interrupt.
9	CORE_PDN_FLAG	R/W	1	Core power-down reset status flag bit: 0: no reset occurred 1: reset occurred

Bit	Name	Attribute	Reset Value	Description
				Note: This bit is cleared by writing 1 to the RST_FLAG_CLR bit.
8	RSV	-	-	Reserved
7	RST_FLAG_CLR	W	0	Clear reset flag bit: 0: not clear 1: clear the reset flag
6	XTL_DT_EN	R/W	0	XTL detection circuit enable: 0: disabled 1: enabled
5	LVD_PMU_EN	R/W	1	Whether the LVD reset signal can reset PMU state machine: 0: PMU state machine cannot be reset. 1: PMU state machine can be reset.
4	BOR_PMU_EN	R/W	1	Whether the BOR reset signal can reset PMU state machine: 0: PMU state machine cannot be reset. 1: PMU state machine can be reset.
3	EFC_LOW_VDD_EN	R/W	0	Low voltage warning enable: 0: low voltage warning disabled 1: low voltage warning enabled
2:1	RSV	R/W	2'b10	Reserved
0	BAT_SEL	R/W	0	External battery backup status register: 0: no external battery backup power (only one main power supply for all system chips) 1: external battery backup power exists (two external power supplies for system chip) If the external V_{DDH} is abnormally powered off, the PDR_EN bit is recommended to be configured as 1. If the chip actively enters the low-power mode and then V_{DDH} is powered off, the PDR_EN bit is recommended to be configured as 0. If BAT_SEL = 1, PDR will not reset the BBU logic.

5.3.6 XTL Configuration Register (PMU_XTLCR)

Offset address: 0x1C

Reset value: 0x0000 9240

Bit	Name	Attribute	Reset Value	Description
31:17	RSV	-	-	Reserved
15:13	ISET	R/W	3'b100	Oscillator current
12:10	RSET	R/W	3'b100	Bias resistance
9:7	GSET	R/W	3'b100	Gain of oscillator
6:4	FBRSET	R/W	3'b100	Feedback resistance
3:2	RSV	-	-	Reserved
1	XTL_EN	R/W	0	XTL enable: 0: disabled 1: enabled
0	LSCLK_SEL	R/W	0	Low-speed clock selection: 0: RCL selected 1: XTL selected

5.3.7 RCL Configuration Register (PMU_RCLCR)

Offset address: 0x20

Reset value: 0xXXXX XXXX

Bit	Name	Attribute	Reset Value	Description
31:13	RSV	-	-	Reserved
12	RCL_TRIM_SEL	R/W	1	RCL trimming value selection bit: 0: use the value saved in this register 1: use the value read after EFC pre-fetch
11:1	RSV	-	-	Reserved
0	RCL_PD	R/W	0	RCL enable: 0: RCL enabled 1: RCL disabled (not recommended) Note: RCL is also controlled by other logics. RCL is automatically disabled in DeepStandby mode while automatically enabled when there is an external wakeup.

5.3.8 Function Clock Control Register (PMU_FCCR)

Offset address: 0x24

Reset value: 0x0000 005F

Bit	Name	Attribute	Reset Value	Description
31:7	RSV	-	-	Reserved
6	RTCEN	R/W	1	RTC module clock enable: 0: disabled 1: enabled
5	RSV	-	-	Reserved
4	LPUART1EN	R/W	1	LPUART1 module clock enable: 0: disabled 1: enabled
3	LPTIM1EN	R/W	1	LPTIM1 module clock enable: 0: disabled 1: enabled
2	LPTIM0EN	R/W	1	LPTIM0 module clock enable: 0: disabled 1: enabled
1	IWDTEN	R/W	1	IWDT module clock enable: 0: disabled 1: enabled
0	LPUART0EN	R/W	1	LPUART0 module clock enable: 0: disabled 1: enabled

5.3.9 Function Reset Control Register (PMU_FRCR)

Offset address: 0x28

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:7	RSV	-	-	Reserved
6	RTCRST	R/W	0	RTC control register reset enable: 0: disabled 1: enabled
5	RSV	-	-	Reserved

Bit	Name	Attribute	Reset Value	Description
4	LPUART1RST	R/W	0	LPUART1 control register reset enable: 0: disabled 1: enabled
3	LPTIM1RST	R/W	0	LPTIM1 control register reset enable: 0: disabled 1: enabled
2	LPTIM0RST	R/W	0	LPTIM0 control register reset enable: 0: disabled 1: enabled
1	IWDTRST	R/W	0	IWDT control register reset enable: 0: disabled 1: enabled
0	LPUART0RST	R/W	0	LPUART0 control register reset enable: 0: disabled 1: enabled

5.3.10 Configuration Protection Register (PMU_CPR)

Offset address: 0x50

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:16	RSV	-	-	Reserved
15:0	CONFIGEN	R/W	0	Configuration protection register: Write 16'hABCD to enable the PMU register write operation, only after which can write operation be performed on PMU-related configuration registers. Write 16'h459E to disable the write operation to PMU register.

6 Reset and Clock Module (RCM)

6.1 Clock Logic

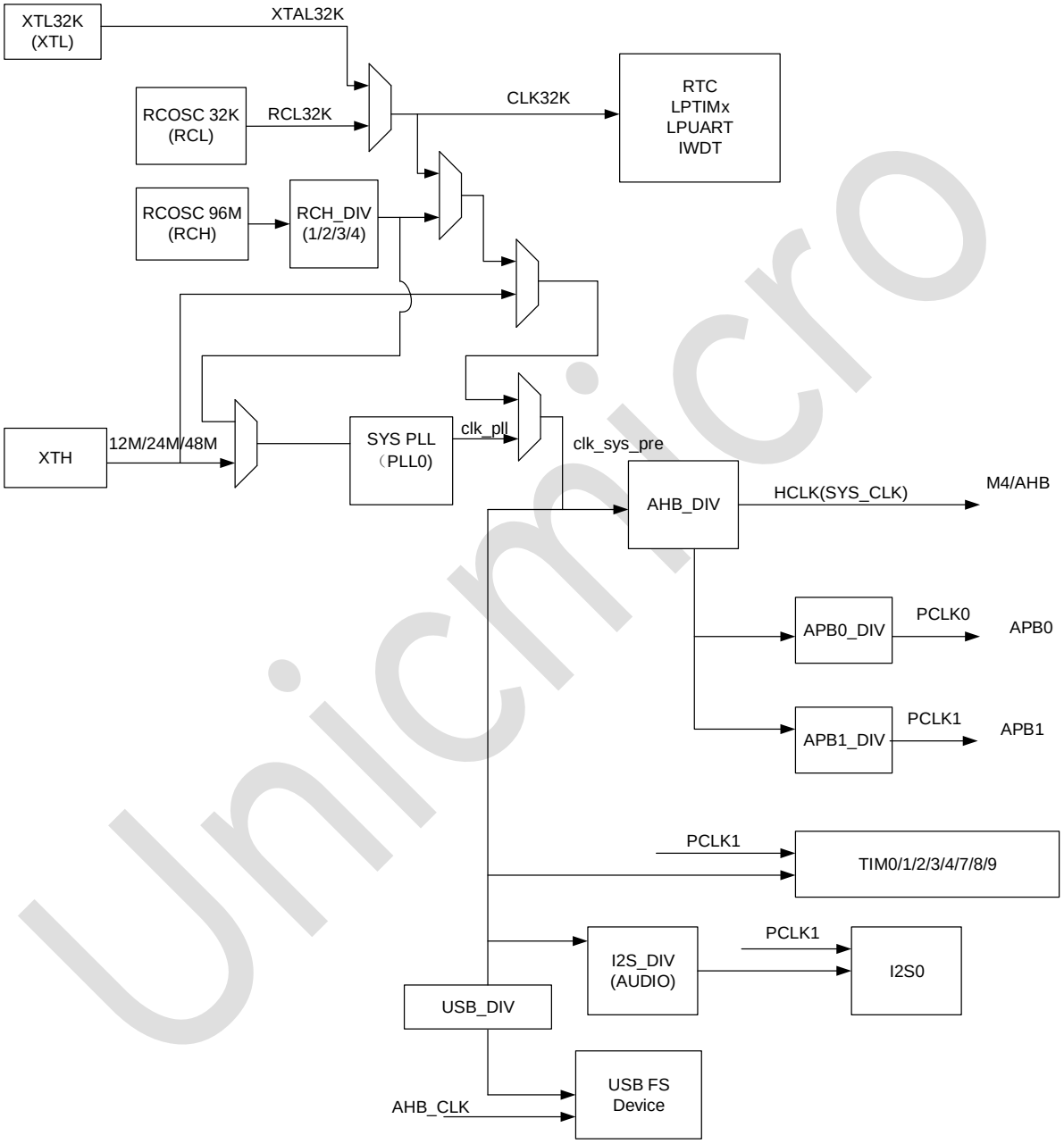


Figure 6-1: Clock Logic

Four different sources can be used to drive the system clock:

- High-precision internal RCH of 96 MHz
- Internal RCL of 32 kHz

- External crystal XTL of 32.768 kHz
- External crystal XTH

There is an internal PLL for system, audio and USB clocks.

The clock controller distributes the clocks coming from different oscillators to the core and the peripherals. It features a 96 MHz internal RC oscillator (RCH), an external high-speed crystal oscillator (XTH), a 32 kHz internal RC oscillator (RCL), a low-speed external crystal oscillator (XTL), a phase-locked loop (PLL), an XTH monitor, a clock prescaler, a clock multiplexer and a clock gating circuitry.

AHB, APB and Cortex™-M4 are derived from the system clock (SYS_CLK), and the clock source of SYS_CLK can be XTH, RCH, PLL, or RCL/XTL with a low frequency of 32 kHz. The independent watchdog is clocked from a low-frequency RCL or XTL, and the RTC is clocked from RCL or XTL.

Notes:

- If audio support is required, PLL fractional frequency division can be configured.
- For applications with strict EMC requirements (motor, etc.), PLL spread spectrum function can be enabled.

6.2 Reset Logic

There are two types of reset, defined as:

- Power reset
- System reset

6.2.1 Power Reset

Power reset, also known as cold reset, resets all systems except for those in the backup domain. System reset will reset everything except for the backup domain, including the

processor core and peripheral IPs. The backup domain reset affects only the backup domain. A reset can be triggered by external signal, internal event and reset generator.

A power reset is generated when one of the following events occurs:

- Power-on/power-down reset (POR/PDR)
- Brown-out reset (BOR)
- When exiting from Standby mode

The power reset acts on registers except the backup domain. The power reset is active low and becomes inactive when the internal LDO power reference is ready to provide core voltage (e.g. 1.1 V). The reset service routine vector is fixed at address 0xXXXX_0004 in the memory map.

6.2.2 System Reset

A system reset is generated when one of the following events occurs:

- Power-on reset (POR)
- External reset (RESETN)
- M4 window watchdog counting termination (WWDT_RSTN)
- Independent watchdog counting termination (IWDT_RSTN)
- The SYSRESETREQ bit in Cortex™-M4 interrupt application and reset control register is set to 1.

System reset will reset everything except for the backup domain, including the processor core and peripheral IPs.

There are seven reset sources, all of which are active low. Each reset signal allows to re-run the CPU. Most registers will be reset again and the program counter (PC) will be remapped at address 0x0000 0000.

Table 6-1: System Reset Source

Reset Signal Name	Active Level	Description
RESETN	Low	Global hardware reset pin: reset the whole chip (except BBU and other registers)
POR_RSTN	Low	1.1 V power-on reset: reset the whole chip
BOR_RSTN	Low	Brown-out reset: reset the whole chip
SOFT_RSTN	Low	Global soft reset control register: reset the system
IWDT_RSTN	Low	Watchdog timer reset: reset the system
WWDT_RSTN	Low	Window watchdog timer reset: reset the system
Block reset signals	Low	Reset a single module

Table 6-2: Reset Mode

Reset Mode	Generation Condition
POR & PDR	Power on with V_{DDH} (1.8–3.6 V) and internal core voltage.
RESETN pin reset	External RESETN pin inputs low level voltage.
BOR	V_{DDH} drops below the V_{BOR} threshold.
LVD reset	V_{DDH} drops below the V_{LVD} threshold.
Window watchdog reset (WWDT)	-
Independent watchdog reset (IWDT)	-
Reset upon wakeup from power-down mode	By setting the reset generated by the power-down mode, the core wakes up from the reset state upon occurrence of the power-down wakeup event.
Software reset	-
Reset upon abnormal oscillation stop of external high-speed oscillator	The abnormal oscillation stop of external high-speed oscillator is detected.

6.3 Register Description

Register base address: 0x4050_0000

Table 6-3: List of RCM Registers

Offset Address	Name	Description
0x000	RCM_CR0	Clock control register 0
0x008	RCM_PLL0CFGR0	PLL0 configuration register 0
0x00C	RCM_PLL0CFGR1	PLL0 configuration register 1
0x010	RCM_PLL0CFGR2	PLL0 configuration register 2
0x020	RCM_PLLTSR	PLL stabilization time setting register

Offset Address	Name	Description
0x030	RCM_CFGR0	Clock configuration register 0
0x034	RCM_CFGR1	Clock configuration register 1
0x038	RCM_CFGR2	Clock configuration register 2
0x040	RCM_CIFR	Clock interrupt flag register
0x044	RCM_CIER	Clock interrupt enable register
0x060	RCM_AHBCKENR	AHB peripheral clock enable register
0x06C	RCM_APB0CKENR	APB0 peripheral clock enable register
0x070	RCM_APB1CKENR	APB1 peripheral clock enable register
0x07C	RCM_AHBRSTR	AHB peripheral reset enable register
0x088	RCM_APB0RSTR	APB0 peripheral reset enable register
0x08C	RCM_APB1RSTR	APB1 peripheral reset enable register
0x098	RCM_SOFTRSTR	Software reset register
0x0E0	RCM_RFR	Reset flag register
0x0E4	RCM_EXRSTCR	External reset pin control register
0x0F0	RCM_RCMPR	RCM write protection register

6.3.1 Clock Control Register 0 (RCM_CR0)

Offset address: 0x000

Reset value: 0x0020 0421

Bit	Name	Attribute	Reset Value	Description
31:29	RSV	-	-	Reserved
28	PLLSRC	R/W	0	PLL0 input clock selection: 0: RCH selected as PLL0 input clock 1: XTH selected as PLL0 input clock Note: It can be configured only when PLL0 is disabled.
27:26	RSV	-	-	Reserved
25	PLL0STB	R	0	PLL0 stabilization flag: 0: unstable 1: stable
24	PLL0EN	R/W	0	PLL0 enable: 0: disabled 1: enabled

Bit	Name	Attribute	Reset Value	Description
23:22	XTH_SF	R/W	0	XTH oscillation frequency selection (SF0, SF1): 2'b00: 1−4 MHz 2'b01: 4.1−12 MHz 2'b10: 12.1−24 MHz 2'b11: 24.1−48 MHz
21:20	XTHSS	R/W	2’b10	XTH stabilization time selection: 2’b00: 4096 cycles 2’b01: 16384 cycles 2’b10: 32768 cycles 2’b11: 65535 cycles
19	RSV	-	-	Reserved
18	XTH_BYP	R/W	0	XTH bypass selection: 1: external crystal not required, clock input directly from pin 0: out from XTH
17	XTH_STB	R	0	XTH stabilization flag: 1: stable 0: unstable
16	XTH_EN	R/W	0	XTH enable: 0: disabled 1: enabled
15:14	XTH_SFRB	R/W	0	XTH feedback resistance selection: The resistance configuration signal is of the following correspondence: 00: 500 kΩ 01: 100 kΩ 1X: HiZ
13:11	RSV	-	-	Reserved
10	RCH_STB	R	1	Internal RCH stabilization flag: 1: RCH is stable, can be used for internal circuit. 0: RCH is not stable, can not be used for internal circuit.

Bit	Name	Attribute	Reset Value	Description
9:8	RCHSS	R/W	2'b00	Internal RCH stabilization time selection: 11: 256 cycles 10: 64 cycles 01: 16 cycles 00: 4 cycles Note: This value can be reduced in Stop mode to speed up the wakeup time.
7:6	RSV	-	-	Reserved
5:4	XTLSS	R/W	2'h2	External XTL stabilization time selection: 2'b00: 1024 cycles 2'b01: 4096 cycles 2'b10: 16384 cycles 2'b11: 32768 cycles
3	XTH_RST_INT_EN	R/W	0	Enable reset or interrupt upon XTH exception: 0: disabled 1: enabled Note: This register will not be reset by the XTH exception. The register value shall be cleared by writing 0 to clear the oscillation stop interrupt.
2	XTH_STOP_SEL	R/W	0	XTH exception status selection: 0: interrupt generated upon clock exception 1: reset generated upon clock exception Note: This register will not be reset by the XTH exception. The register value shall be cleared by writing 0.
1	XTH_MEN	R/W	0	XTH monitoring enable: 0: disabled 1: enabled
0	RCH_EN	R/W	1	Internal RCH enable: 0: disabled 1: enabled Note: When the system enters Standby or DeepStandby mode, RCH will automatically turn off.

6.3.2 PLL0 Configuration Register 0 (RCM_PLL0CFGR0)

Offset address: 0x008

Reset value: 0x0000 8290

Bit	Name	Attribute	Reset Value	Description
31:21	SSRATE	R/W	0	Spread spectrum slope setting
20:15	PLL0_DM	R/W	6'h1	PLL0_DM signal
14:5	PLL0_DN	R/W	10'h14	PLL0_DM signal: the value written shall be greater than or equal to 16.
4:2	PLL0_DP	R/W	3'h4	PLL0_DP signal
1	PLL0_BYP	R/W	0	PLL0 bypass control signal: 0: clock signal comes from PLL VCO 1: PLL VCO bypassed and clock signal comes from REF CLK
0	PLL0_CFGEN	R/W	0	PLL0 configuration enable (with parameter change): 0: without parameter change 1: with configuration change Note: After the PLL parameter configuration is changed, enable this bit and wait for the PLL to stabilize before switching the PLL clock.

Notes:

1. For integer frequency division, the formula of PLL0 output clock is as follows:

   $f_{CLKO} = f_{refclk} * PLL0\_DN / (PLL0\_DM * PLL0\_DP)$

2. For fractional frequency division, the formula of PLL0 output clock is as follows:

   $f_{CLKO} = f_{refclk} * (PLL0\_DN + PLLI0\_FRAC / 2^{24}) / (PLL0\_DM * PLL0\_DP)$

3. For the above two formulas, the frequency of (f_{refclk} * PLL0_DN/PLL0_DM) shall be greater than 300 MHz and less than 600 MHz.

6.3.3 PLL0 Configuration Register 1 (RCM_PLL0CFGR1)

Offset address: 0x00C

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:26	RSV	-	-	Reserved
25:24	SSC_MODE	R/W	0	PLL0 spread spectrum mode selection: 2'b00: lower spread spectrum 2'b01: center spread spectrum 2'b10: upper spread spectrum 2'b11: reserved
23:0	SLOPE	R/W	0	Slope factor of PLL0 spread spectrum

6.3.4 PLL0 Configuration Register 2 (RCM_PLL0CFGR2)

Offset address: 0x010

Reset value: 0x0800 0000

Bit	Name	Attribute	Reset Value	Description
31:30	RSV	-	-	Reserved
29:26	COUNT	R/W	4'h2	PLL control signal: COUNT[0]: dithering mode enable (1: enabled) COUNT[1]: PLL fractional part logic reset control (1: fractional part logic reset; 0: fractional part logic reset release); in fractional mode and spread spectrum mode, this bit is written as 0. COUNT[3:2]: reserved
25:24	MODE	R/W	0	Operation mode selection: 2'b00: integer mode 2'b01: fractional mode 2'b10: spread spectrum mode 2'b11: reserved
23:0	FRAC	R/W	0	PLL0 fractional division factor

6.3.5 PLL Stabilization Time Setting Register (RCM_PLLTSR)

Offset address: 0x020

Reset value: 0x0000 2EE0

Bit	Name	Attribute	Reset Value	Description
31:17	RSV	-	-	Reserved
16	PLL_LT_EN	R/W	0	PLL output clock: 0: the software waits for PLL0STB or PLL1STB before switching the system clock to PLL (software waiting) 1: PLL stabilizes before outputting clock to be applied to the system Note: This bit shall be set to 1 if PLL is selected as the system clock in Stop mode and it is expected to be kept after wakeup.
15:0	PLL_LT	R/W	16'h2EE0	PLL lock time setting register: 16'h2EE0 (16'd12000) corresponds to PLL lock time of 0.5 ms (counting with 24 MHz clock)

6.3.6 Clock Configuration Register 0 (RCM_CFGR0)

Offset address: 0x030

Reset value: 0x0000 80C0

Bit	Name	Attribute	Reset Value	Description
31:28	I2S_DIV	R/W	0	I2S_MCLK clock division factor: 4'b0xxx: not divided 4'b1000: divided by 2 4'b1001: divided by 4 4'b1010: divided by 8 4'b1011: divided by 16 4'b1100: divided by 32 4'b1101: divided by 64 4'b1110: divided by 128 4'b1111: divided by 256
27:21	RSV	-	-	Reserved

Bit	Name	Attribute	Reset Value	Description
20	I2S_SRC	R/W	0	I2S clock source selection: 0: PLL0 clock output as I2S clock 1: external pin I2S_MCLK input clock as I2S clock
19:16	APB1_DIV	R/W	0	APB1 clock division factor: 4'b0xxx: not divided 4'b1000: divided by 2 4'b1001: divided by 4 4'b1010: divided by 8 4'b1011: divided by 16 4'b1100: divided by 32 4'b1101: divided by 64 4'b1110: divided by 128 4'b1111: divided by 256
15:12	APB0_DIV	R/W	4'b1000	APB0 clock division factor: 4'b0xxx: not divided 4'b1000: divided by 2 4'b1001: divided by 4 4'b1010: divided by 8 4'b1011: divided by 16 4'b1100: divided by 32 4'b1101: divided by 64 4'b1110: divided by 128 4'b1111: divided by 256
11:8	AHB_DIV	R/W	0	AHB clock division factor: 4'b0xxx: not divided 4'b1000: divided by 2 4'b1001: divided by 4 4'b1010: divided by 8 4'b1011: divided by 16 4'b1100: divided by 32 4'b1101: divided by 64 4'b1110: divided by 128 4'b1111: divided by 256
7:6	RCH_DIV	R/W	2'b11	Internal RCH division factor: At power-on prefetch, RCH (96 MHz) is divided by a fixed factor of 4 to generate

Bit	Name	Attribute	Reset Value	Description
				24 MHz clocks. 2'b00: divided by 1 2'b01: divided by 2 2'b10: divided by 3 2'b11: divided by 4 Note: Normal operation is performed with 24 MHz clock.
5: 4	RSV	-	-	Reserved
3:2	SYS_SWS	R	2'b00	System clock switching status: 2'b00: RCH selected as system clock 2'b01: XTH selected as system clock 2'b10: PLL0 output clock selected as system clock 2'b11: 32 kHz RCL or XTL selected as system clock Note: There exists the case that XTH is not available and it is switched to RCH.
1:0	SYS_SW	R/W	2'b00	System clock switching: 2'b00: RCH selected as system clock 2'b01: XTH selected as system clock 2'b10: PLL0 output clock selected as system clock 2'b11: 32 kHz RCL or XTL selected as system clock

6.3.7 Clock Configuration Register 1 (RCM_CFGR1)

Offset address: 0x034

Reset value: 0xFF04 2002

Bit	Name	Attribute	Reset Value	Description
31:28	USART7_DIV	R/W	4'hF	USART7 clock division factor (based on PCLK1 clock): 4'b0000: reserved 4'b0001: divided by 2 ……

Bit	Name	Attribute	Reset Value	Description
				4’b1111: divided by 16
27:24	USART6_DIV	R/W	4’hF	USART6 clock division factor (based on PCLK0 clock): 4’b0000: reserved 4’b0001: divided by 2 …… 4’b1111: divided by 16
23:21	RSV	-	-	Reserved
20	TIM_CLK_SEL 1	R/W	0	Select clock source for TIM0/1/2/3/4/7/8/9 (TIM14/15/16 clock source is always PCLK1): 0: PCLK1 selected 1: SYSPLL selected
19	RSV	-	-	Reserved
18:16	USB_DIV (48M_DIV)	R/W	3’b100	Set division factor for USB module operating clock (if USB is used, appropriate division factor must be configured to generate 48 MHz clock): 3’b000: not divided 3’b001: divided by 2 3’b010: divided by 3 3’b011: divided by 4 3’b100: divided by 5 3’b101: divided by 6 3’b110: divided by 7 3’b111: divided by 8
15	RSV	-	-	Reserved
14:12	MCO1_DIV	R/W	3’b010	MCO1 division factor: 3’b000: not divided 3’b001: divided by 2 3’b010: divided by 4 3’b011: divided by 8 3’b100: divided by 16 3’b101: divided by 32 3’b110: divided by 64 3’b111: divided by 128
11	MCO1_DIV_E	R/W	0	MCO1 prescaler enable:

Bit	Name	Attribute	Reset Value	Description
	N			0: disabled 1: enabled
10:8	MCO1	R/W	3'b000	MCO1 clock output: 3'b000: RCH_DIV (e.g. 24 MHz) selected to be output onto MCO1 pin (PC9) 3'b001: XTH selected to be output onto MCO1 pin (PC9) 3'b001: low-speed clock (RCL or XTL) selected to be output onto MCO1 pin (PC9) 3'b011: XTL selected to be output onto MCO1 pin (PC9) 3'b001: PLL0 clock selected to be output onto MCO1 pin (PC9) 3'b101: reserved 3'b110: reserved 3'b111: AHB clock selected to be output onto MCO1 pin (PC9)
7	MCO0_DIV_EN	R/W	0	MCO0 prescaler enable: 0: disabled 1: enabled
6:4	MCO0_DIV	R/W	3'b000	MCO0 division factor: 3'b000: not divided 3'b001: divided by 2 3'b010: divided by 4 3'b011: divided by 8 3'b100: divided by 16 3'b101: divided by 32 3'b110: divided by 64 3'b111: divided by 128
3	RSV	-	-	Reserved
2:0	MCO0	R/W	3'b010	MCO0 clock output: 3'b000: RCH_DIV (e.g. 24 MHz) selected to be output onto MCO0 pin (PA8) 3'b001: XTH selected to be output onto MCO0 pin (PA8) 3'b001: low-speed clock (RCL or XTL)

Bit	Name	Attribute	Reset Value	Description
				selected to be output onto MCO0 pin (PA8) 3'b011: XTL clock selected to be output onto MCO0 pin (PA8) 3'b001: PLL0 clock selected to be output onto MCO0 pin (PA8) 3'b101: reserved 3'b110: RCL clock selected to be output onto MCO0 pin (PA8) 3'b111: AHB clock selected to be output onto MCO0 pin (PA8)

6.3.8 Clock Configuration Register 2 (RCM_CFGR2)

Offset address: 0x038

Reset value: 0x0000 00FF

Bit	Name	Attribute	Reset Value	Description
31:10	RSV	-	-	Reserved
9	LPUART_DIVEN	R/W	0	LPUART prescaler clock: 0: disabled 1: enabled
8:0	LPUART_DIV	R/W	9'hFF	LPUART0/1 prescaler division factor (based on PCLK0 clock): Prescaler clock is: PCLK0 / (LPUART_DIV + 1)

6.3.9 Clock Interrupt Flag Register (RCM_CIFR)

Offset address: 0x040

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:8	RSV	-	-	Reserved
7	CSSF	R/W	0	Clock security system interrupt flag. Writing 1 to clear the interrupt. 0: XTH is normal 1: XTH output failure

Bit	Name	Attribute	Reset Value	Description
6:5	RSV	-	-	Reserved
4	PLL0RDYF	R/W	0	PLL0 clock ready interrupt flag. Writing 1 clears this interrupt. 0: PLL0 ready interrupt disabled 1: PLL0 ready interrupt enabled
3	HSERDYF	R/W	0	XTH ready interrupt flag. Writing 1 clears this interrupt. 0: XTH ready interrupt disabled 1: XTH ready interrupt enabled
2	RSV	-	-	Reserved
1	LSERDYF	R/W	0	XTL ready interrupt flag. Writing 1 clears this interrupt. 0: XTL ready interrupt disabled 1: XTL ready interrupt enabled
0	RSV	-	-	Reserved

6.3.10 Clock Interrupt Enable Register (RCM_CIER)

Offset address: 0x044

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:8	RSV	-	-	Reserved
7	CSSIE	R/W	0	Clock security system interrupt enable: 0: disabled 1: enabled
6:5	RSV	-	-	Reserved
4	PLL0RDYIE	R/W	0	PLL0 ready interrupt enable: 0: disabled 1: enabled
3	HSERDYIE	R/W	0	XTH ready interrupt enable: 0: disabled 1: enabled
2	RSV	-	-	Reserved
1	LSERDYIE	R/W	0	XTL ready interrupt enable: 0: disabled 1: enabled

Bit	Name	Attribute	Reset Value	Description
0	RSV	-	-	Reserved

6.3.11 AHB Peripheral Clock Enable Register (RCM_AHBCKENR)

Offset address: 0x060

Reset value: 0x0000 F911

Bit	Name	Attribute	Reset Value	Description
31:16	RSV	-	-	Reserved
15	GPIODEN	R/W	1	GPIOD module clock enable: 0: disabled 1: enabled
14	GPIOCEN	R/W	1	GPIOC module clock enable: 0: disabled 1: enabled
13	GPIOBEN	R/W	1	GPIOB module clock enable: 0: disabled 1: enabled
12	GPIOAEN	R/W	1	GPIOA module clock enable: 0: disabled 1: enabled
11	EFCEN	R/W	1	EFC module clock enable (generally not modified): 0: disabled 1: enabled
10	RSV	-	-	Reserved
9	SRAM1EN	R/W	0	SRAM1 controller clock enable: 0: disabled 1: enabled
8	SRAM0EN	R/W	1	SRAM0 controller clock enable: 0: disabled 1: enabled
7:6	RSV	-	-	Reserved
5	DMA0EN	R/W	0	DMA0 controller clock enable: 0: disabled 1: enabled

Bit	Name	Attribute	Reset Value	Description
4	CRCEN	R/W	1	CRC controller clock enable: 0: disabled 1: enabled
3	AESEN	R/W	0	AES module clock enable: 0: disabled 1: enabled
2	CORDICEN	R/W	0	CORDIC module clock enable: 0: disabled 1: enabled
1	RSV	-	-	Reserved
0	USB0EN	R/W	1	USB0 (device) controller clock enable: 0: disabled 1: enabled

6.3.12 APB0 Peripheral Clock Enable Register (RCM_APB0CKENR)

Offset address: 0x06C

Reset value: 0x0000 0040

Bit	Name	Attribute	Reset Value	Description
31:16	RSV	-	-	Reserved
15	I2C2EN	R/W	0	I2C2 controller clock enable: 0: disabled 1: enabled
14	I2C1CEN	R/W	0	I2C1 controller clock enable: 0: disabled 1: enabled
13	DACCEN	R/W	0	DAC controller clock enable: 0: disabled 1: enabled
12	TIM5EN	R/W	0	TIM5 controller clock enable: 0: disabled 1: enabled
11	TRNGEN	R/W	0	TRNG controller clock enable: 0: disabled 1: enabled

Bit	Name	Attribute	Reset Value	Description
10:9	RSV	-	-	Reserved
8	CAN0EN	R/W	0	CAN0 controller clock enable: 0: disabled 1: enabled
7	RSV	-	-	Reserved
6	WWDTEN	R/W	1	WWDT controller clock enable: 0: disabled 1: enabled
5	USART6EN	R/W	0	USART6 controller clock enable: 0: disabled 1: enabled
4	UART2EN	R/W	0	UART2 controller clock enable: 0: disabled 1: enabled
3:0	RSV	-	-	Reserved

6.3.13 APB1 Peripheral Clock Enable Register (RCM_APB1CKENR)

Offset address: 0x070

Reset value: 0x0008 0000

Bit	Name	Attribute	Reset Value	Description
31:22	RSV	-	-	Reserved
21	TIM9EN	R/W	0	TIM9EN controller clock enable: 0: disabled 1: enabled
20	TIM8EN	R/W	0	TIM8 controller clock enable: 0: disabled 1: enabled
19	UART0EN	R/W	1	UART0 controller clock enable: 0: disabled 1: enabled Note: If TIM8 is to be enabled, this bit also shall be written with 1.
18	UART1EN	R/W	0	UART1 controller clock enable: 0: disabled 1: enabled

Bit	Name	Attribute	Reset Value	Description
17	UART3EN	R/W	0	UART3 controller clock enable: 0: disabled 1: enabled
16	USART7EN	R/W	0	USART7 controller clock enable: 0: disabled 1: enabled
15	I2SEN	R/W	0	I2S controller clock enable: 0: disabled 1: enabled
14	TIM4EN	R/W	0	TIM4EN controller clock enable: 0: disabled 1: enabled
13	TIM3EN	R/W	0	TIM3 controller clock enable: 0: disabled 1: enabled
12	TIM2EN	R/W	0	TIM2EN controller clock enable: 0: disabled 1: enabled
11	TIM1EN	R/W	0	TIM1 controller clock enable: 0: disabled 1: enabled
10	TIM16EN	R/W	0	TIM16EN controller clock enable: 0: disabled 1: enabled
9	TIM15EN	R/W	0	TIM15 controller clock enable: 0: disabled 1: enabled
8	TIM14EN	R/W	0	TIM14 controller clock enable: 0: disabled 1: enabled
7	ADCC1EN	R/W	0	ADCC1 module clock enable: 0: disabled 1: enabled
6	ADCC0EN	R/W	0	ADCC0 module clock enable: 0: disabled 1: enabled

Bit	Name	Attribute	Reset Value	Description
5	TIM7EN	R/W	0	TIM7 controller clock enable: 0: disabled 1: enabled
4	TIM0EN	R/W	0	TIM0 controller clock enable: 0: disabled 1: enabled
3	I2C0EN	R/W	0	I2C0 controller clock enable: 0: disabled 1: enabled
2	SPI2EN	R/W	0	SPI2 controller clock enable: 0: disabled 1: enabled
1	SPI1EN	R/W	0	SPI1 controller clock enable: 0: disabled 1: enabled
0	SPI0EN	R/W	0	SPI0 controller clock enable: 0: disabled 1: enabled

6.3.14 AHB Peripheral Reset Enable Register (RCM_AHBRSTR)

Offset address: 0x07C

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:21	RSV	-	-	Reserved
20	LOCKUP_EN	R/W	0	M4 LOCKUP reset enable: 0: disabled 1: enabled (a lockup reset will reset the system) Note: This bit is only affected by POR and PDR, other resets do not reset the value of this bit.
19:16	RSV	-	-	Reserved
15	GPIODRST	R/W	0	GPIOD module reset enable: 0: disabled 1: enabled

Bit	Name	Attribute	Reset Value	Description
14	GPIOCRST	R/W	0	GPIO controller reset enable: 0: disabled 1: enabled
13	GPIOBRST	R/W	0	GPIOB controller reset enable: 0: disabled 1: enabled
12	GPIOARST	R/W	0	GPIOA control register reset enable: 0: disabled 1: enabled
11:10	RSV	-	-	Reserved
9	SRAM1RST	R/W	0	SRAM1 controller reset enable: 0: disabled 1: enabled
8	SRAM0RST	R/W	0	SRAM0 control register reset enable: 0: disabled 1: enabled
5	DMAC0RST	R/W	0	DMAC0 control register reset enable: 0: disabled 1: enabled
4	CRCRST	R/W	0	CRC controller reset enable: 0: disabled 1: enabled
3	AESRST	R/W	0	AES controller reset enable: 0: disabled 1: enabled
2	CORDICRST	R/W	0	CORDIC control register reset enable: 0: disabled 1: enabled
1	RSV	-	-	Reserved
0	USB0RST	R/W	0	USB0 controller reset enable: 0: disabled 1: enabled

6.3.15 APB0 Peripheral Reset Enable Register (RCM_APB0RSTR)

Offset address: 0x088

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:16	RSV	-	-	Reserved
15	I2C2RST	R/W	0	I2C2 controller reset enable: 0: disabled 1: enabled
14	I2C1RST	R/W	0	I2C1 controller reset enable: 0: disabled 1: enabled
13	DACCRST	R/W	0	DAC controller reset enable: 0: disabled 1: enabled
12	TIM5RST	R/W	0	TIM5 controller reset enable: 0: disabled 1: enabled
11	TRNGRST	R/W	0	TRNG control register reset enable: 0: disabled 1: enabled
10:9	RSV	-	-	Reserved
8	CAN0RST	R/W	0	CAN0 controller reset enable: 0: disabled 1: enabled
7	RSV	-	-	Reserved
6	WWDTRST	R/W	0	WWDT control register reset enable: 0: disabled 1: enabled
5	USART6RST	R/W	0	USART6 controller reset enable: 0: disabled 1: enabled
4	UART2RST	R/W	0	UART2 controller reset enable: 0: disabled 1: enabled
3:0	RSV	-	-	Reserved

6.3.16 APB1 Peripheral Reset Enable Register (RCM_APB1RSTR)

Offset address: 0x08C

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:22	RSV	-	-	Reserved
21	TIM9RST	R/W	0	TIM9 controller reset enable: 0: disabled 1: enabled
20	TIM8RST	R/W	0	TIM8 controller reset enable: 0: disabled 1: enabled
19	UART0RST	R/W	0	UART0 controller reset enable: 0: disabled 1: enabled
18	UART1RST	R/W	0	UART1 controller reset enable: 0: disabled 1: enabled
17	UART3RST	R/W	0	UART3 controller reset enable: 0: disabled 1: enabled
16	USART7RST	R/W	0	USART7 controller reset enable: 0: disabled 1: enabled
15	I2SRST	R/W	0	I2S controller reset enable: 0: disabled 1: enabled
14	TIM4RST	R/W	0	TIM4 controller reset enable: 0: disabled 1: enabled
13	TIM3RST	R/W	0	TIM3 controller reset enable: 0: disabled 1: enabled
12	TIM2RST	R/W	0	TIM2 controller reset enable: 0: disabled 1: enabled

Bit	Name	Attribute	Reset Value	Description
11	TIM1RST	R/W	0	TIM1 controller reset enable: 0: disabled 1: enabled
10	TIM16RST	R/W	0	TIM16 controller reset enable: 0: disabled 1: enabled
9	TIM15RST	R/W	0	TIM15 controller reset enable: 0: disabled 1: enabled
8	TIM14RST	R/W	0	TIM14 controller reset enable: 0: disabled 1: enabled
7	ADCC1RST	R/W	0	ADCC1 reset enable: 0: disabled 1: enabled
6	ADCC0RST	R/W	0	ADCC0 reset enable: 0: disabled 1: enabled
5	TIM7RST	R/W	0	TIM7 controller reset enable: 0: disabled 1: enabled
4	TIM0RST	R/W	0	TIM0 controller reset enable: 0: disabled 1: enabled
3	I2C0RST	R/W	0	I2C0 reset enable: 0: disabled 1: enabled
2	SPI2RST	R/W	0	SPI2 controller reset enable: 0: disabled 1: enabled
1	SPI1RST	R/W	0	SPI1 controller reset enable: 0: disabled 1: enabled
0	SPI0RST	R/W	0	SPI0 controller reset enable: 0: disabled 1: enabled

6.3.17 Software Reset Register (RCM_SOFTRSTR)

Offset address: 0x098

Reset value: 0x0000 0001

Bit	Name	Attribute	Reset Value	Description
31:0	SOFTRST	R/W	1	Software reset register: when this bit is written with 32'hA5A5_4321, a software reset will be generated to reset the CPU and all IPs on the AHB/APB bus. Also, the eFlash address will be remapped (to 1). When reading: 0: the system is about to perform a software reset. 1: the system does not perform a software reset. When writing: 32'hA5A5_4321: a software reset is generated Other values: no software reset generated Note: Generally not used.

6.3.18 Reset Flag Register (RCM_RFR)

Offset address: 0x0E0

Reset value: 0x0000 0003

Bit	Name	Attribute	Reset Value	Description
31:17	RSV	-	-	Reserved
16	RSTM	W	0	Reset flag clear: 0: not clear 1: clear
15:11	RSV	-	-	Reserved
10	LOCKUPRSTF	R	0	LOCKUP reset status flag: 0: no reset occurred 1: reset occurred
9	LVDRSTF	R	0	LVD status flag:

Bit	Name	Attribute	Reset Value	Description
				0: no reset occurred 1: reset occurred
8	SYSSOFTRSTF	R	0	System software reset status flag: 0: no reset occurred 1: reset occurred
7	BORRSTF	R	0	BOR status flag: 0: no reset occurred 1: reset occurred
6	XTHRSTF	R	0	XTH clock reset status flag: 0: no reset occurred 1: reset occurred
5	WWDTRSTF	R	0	Window watchdog reset status flag: 0: no reset occurred 1: reset occurred
4	IWDTRSTF	R	0	Independent watchdog reset status flag: 0: no reset occurred 1: reset occurred
3	SOFTRSTF	R	0	M4 software reset status flag: 0: no reset occurred 1: reset occurred
2	PINRSTF	R	0	External pin RESETN reset status flag: 0: no reset occurred 1: reset occurred
1	PORRSTF	R	1	POR & PDR or core BOR status flag bit: 0: no reset occurred 1: reset occurred
0	RSV	R	1	Reserved

Note: This register can only be reset by POR.

6.3.19 External Reset Control Register (RCM_EXRSTCR)

Offset address: 0x0E4

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:2	RSV	-	-	Reserved
1	RESETN_DISABLE	R/W	0	External reset enable: 1: disabled 0: enabled Note: As the reset is disabled, PD13 can be used as a normal GPIO input.
0	EXT_FILTER_EN	R/W	0	External reset filter enable: 1: enabled 0: disabled

6.3.20 RCM Configuration Protection Register (RCM_RCMPR)

Offset address: 0x0F0

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:0	RCMPR	R/W	0	RCM register write protection: Write 0xA5A5_5A5A to this register to enable write operation of RCM register, and write other values to disable write operation. After configuring the RCM clock reset configuration register, it is recommended to write other values to disable the write operation and protect each configured RCM register value. When reading: 0: write operation disabled 1: write operation enabled When writing: 32'hA5A5_5A5A: write operation enabled Other values: write operation disabled

7 GPIO

7.1 Overview

GPIO contains general data input and output interfaces, which can be shared with other functional pins, depending on the chip configuration. With these data interfaces, any number of pins can be configured as interrupt signal inputs.

7.2 Main Features

- The direction of any I/O port can be configured by software.
- Each input pin can be configured with edge or level trigger for the interrupt.
- Input filter

7.3 Functional Description

7.3.1 GPIO Input and Output

Generate output signal and synchronize input signal.

7.3.2 GPIO Interrupt Generation

Capture the input signal and generate an interrupt. In the case of filtering being disabled, the level detection interrupt can be triggered without clock.

7.4 Register Description

GPIOA register base address: 0x4010_0000

GPIOB register base address: 0x4010_0400

GPIOC register base address: 0x4010_0800

GPIOD register base address: 0x4010_0C00

The registers are listed below:

Table 7-1: List of GPIO Registers

Offset Address	Name	Description
0x00	GPIOx_MODE	Function mode register
0x04	GPIO_SET	Output set register
0x08	GPIO_CLR	Output clear register
0x0C	GPIO_ODATA	Output data register
0x10	GPIO_IDATA	Input data register
0x14	GPIO_IEN	Interrupt enable register
0x18	GPIO_IS	Interrupt trigger mode register
0x1C	GPIO_IBE	Interrupt edge-trigger mode register
0x20	GPIO_IEV	Interrupt level-trigger setting register
0x24	GPIO_IC	Interrupt clear register
0x28	GPIO_RIS	Raw interrupt status register
0x2C	GPIO_MIS	Masked interrupt status register
0x30	GPIO_DBEN	Filter enable register
0x34	GPIO_DBL	Filter length register
0x38	GPIO_LOCK	Configuration lock register
0x3C	GPIO_IM	Input mode register
0x40	GPIO_PULL	Pull-up/down register
0x48	GPIO_SR	Driving speed register
0x4C	GPIO_DS	Driving capability register
0x50	GPIO_AFL	Alternate function low register
0x54	GPIO_AFH	Alternate function high register

Registers are detailed in the following sections.

7.4.1 Function Mode Register (GPIOx_MODE)

Offset address: 0x00

Reset value for GPIOA: 0xABFF FFFF

Reset value for GPIOB: 0xFFFF FE8F

Reset value for GPIOC/GPIOD: 0xFFFF FFFF

Bit	Name	Attribute	Reset Value	Description
31:0	MODE	R/W	0xFFFF FFFF (except GPIOA and GPIOB)	32-bit register with every 2 bits corresponding to one IO PAD; configure the GPIO mode: 00: general-purpose input mode 01: general-purpose output mode 10: alternate function mode 11: analog mode

7.4.2 Output Set Register (GPIO_SET)

Offset address: 0x04

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:16	RSV	-	-	Reserved
15:0	SET	W	0x0	16-bit register with each bit corresponding to one IO PAD: 0: invalid operation 1: I/O is set by writing 1 when I/O is configured as output.

7.4.3 Output Clear Register (GPIO_CLR)

Offset address: 0x08

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:16	RSV	-	-	Reserved
15:0	SET	W	0x0	16-bit register with each bit corresponding to one IO PAD: 0: invalid operation 1: I/O is set by writing 1 when I/O is configured as output.

7.4.4 Output Data Register (GPIO_ODATA)

Offset address: 0x0C

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:16	RSV	-	-	Reserved
15:0	ODATA	R/W	0x0	16-bit register with each bit corresponding to one I/O PAD: When the GPIO is output active, the write operation is directed at the external pin, and the read operation is performed to obtain the external pin output value.

7.4.5 Input Data Register (GPIO_IDATA)

Offset address: 0x10

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:16	RSV	-	-	Reserved
15:0	IDATA	R	0x0	16-bit register with each bit corresponding to one IO PAD: When the GPIO is input active, the actual value of the external pin can be read.

7.4.6 Interrupt Enable Register (GPIO_IEN)

Offset address: 0x14

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:16	RSV	-	-	Reserved
15:0	IEN	R/W	0x0	16-bit register with each bit corresponding to one IO PAD: 0: corresponding pin interrupt disabled 1: corresponding pin interrupt enabled

7.4.7 Interrupt Trigger Mode Register (GPIO_IS)

Offset address: 0x18

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:16	RSV	-	-	Reserved
15:0	IS	R/W	0x0	16-bit register with each bit corresponding to one IO PAD: 0: edge trigger 1: level trigger

7.4.8 Interrupt Edge Trigger Setting Register (GPIO_IBE)

Offset address: 0x1C

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:16	RSV	-	-	Reserved
15:0	IBE	R/W	0x0	16-bit register with each bit corresponding to one IO PAD: 0: single-edge trigger 1: double-edge trigger

7.4.9 Interrupt Level Trigger Setting Register (GPIO_IEV)

Offset address: 0x20

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:16	RSV	-	-	Reserved
15:0	IEV	R/W	0x0	16-bit register with each bit corresponding to one IO PAD: 0: falling-edge/low-level trigger 1: rising-edge/high-level trigger

7.4.10 Interrupt Clear Register (GPIO_IC)

Offset address: 0x24

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:16	RSV	-	-	Reserved
15:0	IC	W	0x0	16-bit register with each bit corresponding to one IO PAD: 0: invalid operation 1: clear interrupt of the corresponding pin

7.4.11 Raw Interrupt Status Register (GPIO_RIS)

Offset address: 0x28

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:16	RSV	-	-	Reserved
15:0	RIS	R	0x0	16-bit register with each bit corresponding to one IO PAD: 0: no interrupt pending on the corresponding pin 1: with interrupt pending on the corresponding pin

7.4.12 Masked Interrupt Status Register (GPIO_MIS)

Offset address: 0x2C

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:16	RSV	-	-	Reserved
15:0	MIS	R	0x0	16-bit register with each bit corresponding to one IO PAD: 0: no interrupt on the corresponding pin output to the system 1: with interrupt on the corresponding pin output to the system

7.4.13 Filter Enable Register (GPIO_DBEN)

Offset address: 0x30

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:16	RSV	-	-	Reserved
15:0	DBEN	R/W	0x0	16-bit register with each bit corresponding to one IO PAD: 0: input filter disabled 1: input filter enabled Note: Only one of them can be selected for input filtering. This filtering is not only for GPIO inputs, but also for other IO inputs.

7.4.14 Filter Length Register (GPIO_DBL)

Offset address: 0x34

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:0	DBL	R/W	0x0	32-bit register, number of filter cycles

7.4.15 Configuration Lock Register (GPIO_LOCK)

Offset address: 0x38

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:16	RSV	-	-	Reserved
15:0	LOCK	R/W	0x0	16-bit register with each bit corresponding to one IO PAD. Setting it to 1 to lock the configuration of corresponding IO until the next reset. The frozen registers are GPIO_MODE, GPIO_IM, GPIO_PULL, GPIO_SR, GPIO_DS, GPIO_AFH and GPIO_AFL.

7.4.16 Input Mode Register (GPIO_IM)

Offset address: 0x3C

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:16	RSV	-	-	Reserved
15:0	IM	R/W	0x0	16-bit register with each bit corresponding to one IO PAD; configure the GPIO input mode: 0: CMOS 1: Schmitt trigger

7.4.17 Pull-up/down Register (GPIO_PULL)

Offset address: 0x40

Reset value for GPIOA: 0xE000E000

Reset value for GPIOB: 0x0014 0000

Reset value for GPIOC/GPIOD: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:16	PE	R/W	0x0	16-bit register with each bit corresponding to one IO PAD Pull-up/down enable: 0: disabled 1: enabled
15:0	PS	R/W	0x0	16-bit register with each bit corresponding to one IO PAD Pull direction selection: 0: pull down 1: pull up Note: The configuration corresponding to PA14 and PB4 is different from the configuration of other pins. The corresponding bits of these two pins are defined as follows: 0: pull up 1: pull down

7.4.18 Driving Speed Register (GPIO_SR)

Offset address: 0x48

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:16	RSV	-	-	Reserved
15:0	SR	R/W	0x0	16-bit register with each bit corresponding to one IO PAD; configure the GPIO driving speed: 0: high speed 1: low speed

7.4.19 Driving Capability Register (GPIO_DS)

Offset address: 0x4C

Reset value: 0x5555 5555

Bit	Name	Attribute	Reset Value	Description
31:0	DS	R/W	0x5555_5555	32-bit register with every 2 bits corresponding to one IO PAD; configure the GPIO output driving capability: 00: 2 mA 01: 6 mA 10: 14 mA 11: 20 mA

7.4.20 Alternate Function Low Register (GPIO_AFL)

Offset address: 0x50

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:0	AFL	R/W	0x0	32-bit register with every 4 bits corresponding to one IO pad (0–7); configure the GPIO alternate function: 0000: alternate function 0 0001: alternate function 1 0010: alternate function 2 … 1111: alternate function 15 (Corresponding to AF0–AF15 in the pin alternate function table)

7.4.21 Alternate Function High Register (GPIO_AFH)

Offset address: 0x54

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:0	AFH	R/W	0x0	32-bit register with every 4 bits corresponding to one IO pad (8–15); configure the GPIO alternate function: 0000: alternate function 0 0001: alternate function 1 0010: alternate function 2 … 1111: alternate function 15 (Corresponding to AF0–AF15 in the pin alternate function table)

7.5 Operation Procedure

7.5.1 Input

1. Enable the GPIOx clock in RCM module.
2. Configure the GPIOx_MODE register to program GPIO as input.
3. Use GPIO_IDATA to obtain the input pin level.

7.5.2 Output

1. Enable the GPIOx clock in RCM module.
2. Configure the GPIOx_MODE register to program GPIO as output.
3. Use GPIO_SET/GPIO_CLR or GPIO_ODATA to set the output level.

7.5.3 Interrupt Trigger Mode

1. Enable the GPIOx clock in RCM module.
2. Configure GPIOx_MODE as input.
3. Clearing the interrupt via GPIO_IC register to avoid exceptions.
4. Configure the GPIO_IS register to select the edge or level trigger mode.
5. In edge trigger mode, configure the GPIO_IBE register to determine whether it is single-edge trigger or double-edge trigger.
6. In single-edge trigger mode, configure the GPIO_IEV register to determine whether it is rising-edge trigger or falling-edge trigger.
7. In level trigger mode, configure the GPIO_IEV register to determine whether it is high-level trigger or low-level trigger.
8. Clearing the interrupt via GPIO_IC register.
9. Configure the GPIO_IEN register to enable the interrupt of corresponding bit.

8 Nested Vectored Interrupt Controller (NVIC)

The ARM Cortex-M4 processor and the nested vectored interrupt controller (NVIC) prioritize and handle all exceptions in the handler mode. When an exception occurs, the system automatically pushes the current operating state of the processor on the stack, and then automatically pops it off the stack after executing the interrupt service subroutine (ISR).

Vector fetching is carried out in parallel with pushing the current operating state to improve the interrupt entry efficiency. The processor supports tail chaining and can realize back-to-back interrupt, greatly reducing the overhead caused by repeatedly switching operating states.

All interrupt types are shown in the following table:

Table 8-1 : Cortex-M4 Interrupt Sources

IRQ No.	Peripheral Interrupt	Peripheral Interrupt Description	Vector Address
0	WWDT	Window watchdog interrupt	0x0000_0040
1	LVD	LVD interrupt	0x0000_0044
2	TAMP_STAMP	RTC tamper and timestamp interrupts	0x0000_0048
3	RTC_WKUP_ALARM	RTC wakeup and alarm interrupt	0x0000_004C
4	EFC	EFC interrupt	0x0000_0050
5	RCM	Clock reset interrupt	0x0000_0054
6	EXTI0	EXTI[0] line interrupt	0x0000_0058
7	EXTI1	EXTI[1] line interrupt	0x0000_005C
8	EXTI2	EXTI[2] line interrupt	0x0000_0060
9	EXTI3	EXTI[3] line interrupt	0x0000_0064
10	EXTI4	EXTI[4] line interrupt	0x0000_0068
11	DMAC0 ch0	DMAC0 channel 0 global interrupt	0x0000_006C
12	DMAC0 ch1	DMAC0 channel 1 global interrupt	0x0000_0070
13	DMAC0 ch2	DMAC0 channel 2 global interrupt	0x0000_0074
14	DMAC0 ch3	DMAC0 channel 3 global interrupt	0x0000_0078
15	DMAC0 ch4	DMAC0 channel 4 global interrupt	0x0000_007C

IRQ No.	Peripheral Interrupt	Peripheral Interrupt Description	Vector Address
16	DMAC0 ch5	DMAC0 channel 5 global interrupt	0x0000_0080
17	DMAC0 ch6	DMAC0 channel 6 global interrupt	0x0000_0084
18	DMAC0_ch7	DMAC0 channel 7 global interrupt	0x0000_0088
19	ADC0	ADC0 global interrupt	0x0000_008C
20	ADC1	ADC1 global interrupt	0x0000_0090
21	CAN0	CAN0 global interrupt	0x0000_0094
22	EXTI9−5	EXTI[9:5] line interrupt	0x0000_0098
23	TIM0_BRK	TIM0 break interrupt	0x0000_009C
24	TIM0_UP	TIM0 update interrupt	0x0000_00A0
25	TIM0_TRG_COM	TIM0 trigger and commutation interrupt	0x0000_00A4
26	TIM0_CC	TIM0 capture compare interrupt	0x0000_00A8
27	TIM1	TIM1 global interrupt	0x0000_00AC
28	TIM2	TIM2 global interrupt	0x0000_00B0
29	TIM3	TIM3 global interrupt	0x0000_00B4
30	I2C0	I2C0 global interrupt	0x0000_00B8
31	I2C1	I2C1 global interrupt	0x0000_00BC
32	SPI0	SPI0 global interrupt	0x0000_00C0
33	SPI1	SPI1 global interrupt	0x0000_00C4
34	UART0	UART0 global interrupt	0x0000_00C8
35	UART1	UART1 global interrupt	0x0000_00CC
36	UART2	UART2 global interrupt	0x0000_00D0
37	EXTI15−10	EXTI[15:10] line interrupt	0x0000_00D4
38	TIM5	TIM5 global interrupt	0x0000_00D8
39	TIM7_BRK	TIM7 break interrupt	0x0000_00DC
40	TIM7_UP	TIM7 update interrupt	0x0000_00E0
41	TIM7_TRG_COM	TIM7 trigger and commutation interrupt	0x0000_00E4
42	TIM7_CC	TIM7 capture compare interrupt	0x0000_00E8
43	TIM4	TIM4 global interrupt	0x0000_00EC
44	SPI2	SPI2 global interrupt	0x0000_00F0
45	UART3	UART3 global interrupt	0x0000_00F4
46	TIM14	TIM14 global interrupt	0x0000_00F8
47	TIM15	TIM15 global interrupt	0x0000_00FC
48	OPA0	OPA0 interrupt	0x0000_0100
49	OPA1	OPA1 interrupt	0x0000_0104
50	I2C2	I2C2 global interrupt	0x0000_0108
51	USB0 Controller	USB0 controller global interrupt	0x0000_010C

IRQ No.	Peripheral Interrupt	Peripheral Interrupt Description	Vector Address
52	USART6	USART6 global interrupt	0x0000_0110
53	USART7	USART7 global interrupt	0x0000_0114
54	FPU	FPU global interrupt	0x0000_0118
55	ACMP0	ACMP0 output interrupt	0x0000_011C
56	ACMP1	ACMP1 output interrupt	0x0000_0120
57	ACMP2	ACMP2 output interrupt	0x0000_0124
58	I2S	I2S global interrupt	0x0000_0128
59	IWDT	IWDT global interrupt	0x0000_012C
60	LPUART0	LPUART0 global interrupt	0x0000_0130
61	LPTIMER0	LPTIMER0 global interrupt	0x0000_0134
62	LPTIMER1	LPTIMER1 global interrupt	0x0000_0138
63	TIM16	TIM16 global interrupt	0x0000_013C
64	DACC	DACC global interrupt	0x0000_0140
65	AES	AES global interrupt	0x0000_0144
66	LPUART1	LPUART1 global interrupt	0x0000_0148
67	OPA2	OPA2 interrupt	0x0000_014C
68	TIM8	TIM8 global interrupt	0x0000_0150
69	TIM9	TIM9 global interrupt	0x0000_0154
70	DMAMUX	DMAMUX interrupt	0x0000_0158

9 System Configuration Controller (SYSCFG)

Register base address: 0x40B0_2000

The registers are listed below:

Table 9-1: List of SYSCFG Registers

Offset Address	Name	Description
0x00	SYSCFG_MEMREMAP	Memory remap register
0x08	SYSCFG_ADCETSR	ADC external trigger selection register
0x0C	SYSCFG_TIMCFGR	TIM break control register
0x10	SYSCFG_EXTICR0	External interrupt configuration register 0
0x14	SYSCFG_EXTICR1	External interrupt configuration register 1
0x18	SYSCFG_EXTICR2	External interrupt configuration register 2
0x1C	SYSCFG_EXTICR3	External interrupt configuration register 3
0x20	SYSCFG_EXTIWR	External interrupt wakeup configuration register
0x24	SYSCFG_MISCCR	MISC control register
0x28	SYSCFG_SCRVR	Systick counter reference value register
0x30	SYSCFG_UARTTICR	UART transmission invert configuration register
0x34	SYSCFG_TIM141516CFGR	TIM14~16 break control register
0x40	SYSCFG_VCSR	Voltage switching configuration status register
0x48	SYSCFG_LPMSR	Low-power mode status register
0x50	SYSCFG_RSV0R	Reserved register 0
0x54	SYSCFG_RSV1R	Reserved register 1

9.1 Register Description

9.1.1 Memory Remap Register (SYSCFG_MEMREMAP)

Offset address: 0x00

Reset value: 0x0000 0015

Bit	Name	Attribute	Reset Value	Description
31:16	PROTECT	R	0	Specific code write-protection register: 16'h5AA5: bit[4] and bits[1:0] enabled; other values: bit[4] and bits[1:0] disabled
15:5	RSV	-	-	Reserved
4	REMAP_RSTN	R/W	1	Remap software reset: 0: reset 1: not reset
3:2	BOOT_PIN_STATUS	R	01	Boot mode selection (BOOT1(PB2), BOOT0, or depend on NVR of EFC) 00: boot from main Flash memory 01: boot from Flash NVR 10: reserved 11: boot from SRAM0/1 Note: If it is selected by NVR of EFC to boot from the main Flash memory, the bit value is 2'b00. If it is to boot from ROM (Flash NVR), these two bits reflect the status of external BOOT1 and BOOT0. According to this two-bit register, Bootloader (Flash NVR) can determine whether to enter SRAM0 or main Flash.
1:0	MEMMODE (BOOT_STATUS)	R/W	01	Memory mapping selection: 2'b00: main Flash memory mapped at 0x0000_0000 2'b01: Flash NVR mapped at 0x0000_0000 2'b10: reserved 2'b11: SRAM0/1 mapped at 0x0000_0000 Note: It is the actual address mapping situation (or boot status) of the current chip.

Notes:

1. There are three ways to determine what is selected to boot from:

   - Prefetch phase, with the highest priority, can select to boot from main Flash memory.
   - If NVR is not selected, refer to the BOOT1 and BOOT0 pins captured after reset to select the boot mode.

BOOT1 (PB2)	BOOT0	Boot
x	0	Boot from main Flash memory
0	1	Boot from system memory
1	1	Boot from SRAM

   - The chip operation process can also be carried out by modifying this register. After writing, the software reset for the corresponding mode.

2. This register cannot be reset by IWDT or software, but only by POR, PDR, RESETN or BOR signal.

9.1.2 ADC External Trigger Selection Register (SYSCFG_ADCETSR)

Offset address: 0x08

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:7	RSV	-	-	Reserved
6:4	INJ_TRIG_SEL	R/W	0	External trigger source selection in ADC injection mode: 000: EXTI15 001: EXTI9 010: EXTI5 011: LPTIM0_OUT 100: LPTIM1_OUT 101: ACMP0_OUT 110: ACMP1_OUT 111: ACMP2_OUT

Bit	Name	Attribute	Reset Value	Description
3	RSV	-	-	Reserved
2:0	RGL_TRIG_SEL	R/W	0	External trigger source selection in ADC regular mode: 000: EXTI11 001: EXTI10 010: EXTI6 011: LPTIM0_OUT 100: LPTIM1_OUT 101: ACMP0_OUT 110: ACMP1_OUT 111: ACMP2_OUT

9.1.3 TIM Break Control Register (SYSCFG_TIMCFGR)

Offset address: 0x0C

Reset value: 0x0010 0103

Bit	Name	Attribute	Reset Value	Description
31:29	RSV	-	-	Reserved
28	TIM7_OPA2_EN	R/W	0	OPA2 comparator break output enable (TIM7): 0: disabled 1: enabled
27	TIM7_OPA1_EN	R/W	0	OPA1 comparator output break enable (TIM7): 0: disabled 1: enabled
26	TIM7_OPA0_EN	R/W	0	OPA0 comparator output break enable (TIM7): 0: disabled 1: enabled
25	TIM7_ACMP2_EN	R/W	0	ACMP2 comparator break output enable (TIM7): 0: disabled 1: enabled
24	TIM7_ACMP1_EN	R/W	0	ACMP1 comparator break output enable

Bit	Name	Attribute	Reset Value	Description
				(TIM7): 0: disabled 1: enabled
23	TIM7_ACMP0_EN	R/W	0	ACMP0 comparator break output enable (TIM7): 0: disabled 1: enabled
22	TIM7_LOCKUP_EN	R/W	0	LOCKUP break input enable (TIM7): 0: disabled 1: enabled
21	TIM7_LVD_EN	R/W	0	LVD break input enable (TIM7): 0: disabled 1: enabled
20	TIM7_BKIN_EN	R/W	1	TIM7 external BKIN break input enable: 0: disabled 1: enabled
19	OPA2_INV	R	0	OPA2 output inverted to break of TIM0/TIM7/TIM14/TIM15/TIM16: 0: non-inverted 1: inverted
18	OPA1_INV	-	0	OPA1 output inverted to break of TIM0/TIM7/TIM14/TIM15/TIM16: 0: non-inverted 1: inverted
17	OPA0_INV	-	0	OPA0 output inverted to break of TIM0/TIM7/TIM14/TIM15/TIM16: 0: non-inverted 1: inverted
16	TIM0_OPA2_EN	R/W	0	OPA2 comparator break output enable (TIM0): 0: disabled 1: enabled
15	TIM0_OPA1_EN	R/W	0	OPA1 comparator break output enable (TIM0): 0: disabled 1: enabled

Bit	Name	Attribute	Reset Value	Description
14	TIM0_OPA0_EN	R/W	0	OPA0 comparator break output enable (TIM0): 0: disabled 1: enabled
13	TIM0_ACMP2_EN	R/W	0	ACMP2 comparator break output enable (TIM0): 0: disabled 1: enabled
12	TIM0_ACMP1_EN	R/W	0	ACMP1 comparator break output enable (TIM0): 0: disabled 1: enabled
11	TIM0_ACMP0_EN	R/W	0	ACMP0 comparator break output enable (TIM0): 0: disabled 1: enabled
10	TIM0_LOCKUP_EN	R/W	0	LOCKUP break input enable (TIM0): 0: disabled 1: enabled
9	TIM0_LVD_EN	R/W	0	LVD break input enable (TIM0): 0: disabled 1: enabled
8	TIM0_BKIN_EN	R/W	1	TIM0 external BKIN break input enable: 0: disabled 1: enabled
7	ACMP2_INV	R/W	0	ACMP2 output inverted to break of TIM0/TIM7/TIM14/TIM15/TIM16: 0: non-inverted 1: inverted
6	ACMP1_INV	R/W	0	ACMP1 output inverted to break of TIM0/TIM7/TIM14/TIM15/TIM16: 0: non-inverted 1: inverted
5	ACMP0_INV	R/W	0	ACMP0 output inverted to break of TIM0/TIM7/TIM14/TIM15/TIM16: 0: non-inverted

Bit	Name	Attribute	Reset Value	Description
				1: inverted
4	TIM_DEBUG_STOP	R/W	0	Timerx debug control signal: 0: invalid 1: valid
3	TIM7_BKIN_INV	R/W	0	TIM7 BKIN input inversion signal: 0: non-inverted 1: inverted
2	TIM0_BKIN_INV	R/W	0	TIM0 BKIN input inversion selection: 0: non-inverted 1: inverted
1	TIM7_BRKR	R/W	1	TIM7 break mode control register: 1: OCx and OCxn being 0 at break 0: OCx and OCxn being OCxp and OCxnp respectively at break
0	TIM0_BRKR	R/W	1	TIM0 break mode control register: 1: OCx and OCxn being 0 at break 0: OCx and OCxn being OCxp and OCxnp respectively at break

9.1.4 External Interrupt Configuration Register 0 (SYSCFG_EXTICR0)

Offset address: 0x10

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:28	RSV	-	-	Reserved
27	EXTI3_WK_FLAG	R	0	EXTI3 wakeup source flag in Stop mode: 0: not woken up by this bit 1: woken up by this bit
26	EXTI2_WK_FLAG	R	0	EXTI2 wakeup source flag in Stop mode: 0: not woken up by this bit 1: woken up by this bit
25	EXTI1_WK_FLAG	R	0	EXTI1 wakeup source flag in Stop mode: 0: not woken up by this bit

Bit	Name	Attribute	Reset Value	Description
				1: woken up by this bit
24	EXTI0_WK_FLAG	R	0	EXTI0 wakeup source flag in Stop mode: 0: not woken up by this bit 1: woken up by this bit
23	EXTI3_WK_EDGE_SEL	R/W	0	EXTI3 wakeup edge selection in Stop mode: 0: rising edge 1: falling edge
22	EXTI2_WK_EDGE_SEL	R/W	0	EXTI2 wakeup edge selection in Stop mode: 0: rising edge 1: falling edge
21	EXTI1_WK_EDGE_SEL	R/W	0	EXTI1 wakeup edge selection in Stop mode: 0: rising edge 1: falling edge
20	EXTI0_WK_EDGE_SEL	R/W	0	EXTI0 wakeup edge selection in Stop mode: 0: rising edge 1: falling edge
19	EXTI3_WKEN	R/W	0	EXTI3 wakeup enable in Stop mode: 0: disabled 1: enabled
18	EXTI2_WKEN	R/W	0	EXTI2 wakeup enable in Stop mode: 0: disabled 1: enabled
17	EXTI1_WKEN	R/W	0	EXTI1 wakeup enable in Stop mode: 0: disabled 1: enabled
16	EXTI0_WKEN	R/W	0	EXTI0 wakeup enable in Stop mode: 0: disabled 1: enabled
15:12	EXTI3	R/W	0	External interrupt configuration bit: 4’b0000: PA[3] pin 4’b0001: PB[3] pin 4’b0010: PC[3] pin

Bit	Name	Attribute	Reset Value	Description
				4’b0011: reserved 4’b0100: reserved 4’b0101: reserved 4’b0110: reserved 4’b0111: reserved 4’b1000: reserved
11:8	EXTI2	R/W	0	External interrupt configuration bit: 4’b0000: PA[2] pin 4’b0001: PB[2] pin 4’b0010: PC[2] pin 4’b0011: PD[2] pin 4’b0100: reserved 4’b0101: reserved 4’b0110: reserved 4’b0111: reserved 4’b1000: reserved
7:4	EXTI1	R/W	0	External interrupt configuration bit: 4’b0000: PA[1] pin 4’b0001: PB[1] pin 4’b0010: PC[1] pin 4’b0011: reserved 4’b0100: reserved 4’b0101: reserved 4’b0110: reserved 4’b0111: reserved 4’b1000: reserved
3:0	EXTI0	R/W	0	External interrupt configuration bit: 4’b0000: PA[0] pin 4’b0001: PB[0] pin 4’b0010: PC[0] pin 4’b0011: PD[0] pin 4’b0100: reserved 4’b0101: reserved 4’b0110: reserved 4’b0111: reserved 4’b1000: reserved

9.1.5 External Interrupt Configuration Register 1 (SYSCFG_EXTICR1)

Offset address: 0x14

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:28	RSV	-	-	Reserved
27	EXTI7_WK_FLAG	R	0	EXTI7 wakeup source flag in Stop mode: 0: not woken up by this bit 1: woken up by this bit
26	EXTI6_WK_FLAG	R	0	EXTI6 wakeup source flag in Stop mode: 0: not woken up by this bit 1: woken up by this bit
25	EXTI5_WK_FLAG	R	0	EXTI5 wakeup source flag in Stop mode: 0: not woken up by this bit 1: woken up by this bit
24	EXTI4_WK_FLAG	R	0	EXTI4 wakeup source flag in Stop mode: 0: not woken up by this bit 1: woken up by this bit
23	EXTI7_WK_EDGE_SEL	R/W	0	EXTI7 wakeup edge selection in Stop mode: 0: rising edge 1: falling edge
22	EXTI6_WK_EDGE_SEL	R/W	0	EXTI6 wakeup edge selection in Stop mode: 0: rising edge 1: falling edge
21	EXTI5_WK_EDGE_SEL	R/W	0	EXTI5 wakeup edge selection in Stop mode: 0: rising edge 1: falling edge
20	EXTI4_WK_EDGE_SEL	R/W	0	EXTI4 wakeup edge selection in Stop mode: 0: rising edge 1: falling edge

Bit	Name	Attribute	Reset Value	Description
19	EXTI7_WKEN	R/W	0	EXTI7 wakeup enable in Stop mode: 0: disabled 1: enabled
18	EXTI6_WKEN	R/W	0	EXTI6 wakeup enable in Stop mode: 0: disabled 1: enabled
17	EXTI5_WKEN	R/W	0	EXTI5 wakeup enable in Stop mode: 0: disabled 1: enabled
16	EXTI4_WKEN	R/W	0	EXTI4 wakeup enable in Stop mode: 0: disabled 1: enabled
15:12	EXTI7	R/W	0	External interrupt configuration bit: 4'b0000: PA[7] pin 4'b0001: PB[7] pin 4'b0010: PC[7] pin 4'b0011: reserved 4'b0100: reserved 4'b0101: reserved 4'b0110: reserved 4'b0111: reserved 4'b1xxx: reserved
11:8	EXTI6	R/W	0	External interrupt configuration bit: 4'b0000: PA[6] pin 4'b0001: PB[6] pin 4'b0010: PC[6] pin 4'b0011: reserved 4'b0100: reserved 4'b0101: reserved 4'b0110: reserved 4'b0111: reserved 4'b1xxx: reserved
7:4	EXTI5	R/W	0	External interrupt configuration bit: 4'b0000: PA[5] pin 4'b0001: PB[5] pin 4'b0010: PC[5] pin

Bit	Name	Attribute	Reset Value	Description
				4’b0011: reserved 4’b0100: reserved 4’b0101: reserved 4’b0110: reserved 4’b0111: reserved 4’b1xxx: reserved
3:0	EXTI4	R/W	0	External interrupt configuration bit: 4’b0000: PA[4] pin 4’b0001: PB[4] pin 4’b0010: PC[4] pin 4’b0011: reserved 4’b0100: reserved 4’b0101: reserved 4’b0110: reserved 4’b0111: reserved 4’b1xxx: reserved

9.1.6 External Interrupt Configuration Register 2 (SYSCFG_EXTICR2)

Offset address: 0x18

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:28	RSV	-	-	Reserved
27	EXTI11_WK_FLAG	R	0	EXTI11 wakeup source flag in Stop mode: 0: not woken up by this bit 1: woken up by this bit
26	EXTI10_WK_FLAG	R	0	EXTI10 wakeup source flag in Stop mode: 0: not woken up by this bit 1: woken up by this bit
25	EXTI9_WK_FLAG	R	0	EXTI9 wakeup source flag in Stop mode: 0: not woken up by this bit 1: woken up by this bit
24	EXTI8_WK_FLAG	R	0	EXTI8 wakeup source flag in Stop mode:

Bit	Name	Attribute	Reset Value	Description
				0: not woken up by this bit 1: woken up by this bit
23	EXTI11_WK_EDGE_SEL	R/W	0	EXTI11 wakeup edge selection in Stop mode: 0: rising edge 1: falling edge
22	EXTI10_WK_EDGE_SEL	R/W	0	EXTI10 wakeup edge selection in Stop mode: 0: rising edge 1: falling edge
21	EXTI9_WK_EDGE_SEL	R/W	0	EXTI9 wakeup edge selection in Stop mode: 0: rising edge 1: falling edge
20	EXTI8_WK_EDGE_SEL	R/W	0	EXTI8 wakeup edge selection in Stop mode: 0: rising edge 1: falling edge
19	EXTI11_WKEN	R/W	0	EXTI11 wakeup enable in Stop mode: 0: disabled 1: enabled
18	EXTI10_WKEN	R/W	0	EXTI10 wakeup enable in Stop mode: 0: disabled 1: enabled
17	EXTI9_WKEN	R/W	0	EXTI9 wakeup enable in Stop mode: 0: disabled 1: enabled
16	EXTI8_WKEN	R/W	0	EXTI8 wakeup enable in Stop mode: 0: disabled 1: enabled
15:12	EXTI11	R/W	0	External interrupt configuration bit: 4'b0000: PA[11] pin 4'b0001: PB[11] pin 4'b0010: PC[11] pin 4'b0011: reserved 4'b0100: reserved

Bit	Name	Attribute	Reset Value	Description
				4’b0101: reserved 4’b0110: reserved 4’b0111: reserved 4’b1000: reserved
11:8	EXTI10	R/W	0	External interrupt configuration bit: 4’b0000: PA[10] pin 4’b0001: PB[10] pin 4’b0010: PC[10] pin 4’b0011: PD[10] pin 4’b0100: reserved 4’b0101: reserved 4’b0110: reserved 4’b0111: reserved 4’b1000: reserved
7:4	EXTI9	R/W	0	External interrupt configuration bit: 4’b0000: PA[9] pin 4’b0001: PB[9] pin 4’b0010: PC[9] pin 4’b0011: PD[9] pin 4’b0100: reserved 4’b0101: reserved 4’b0110: reserved 4’b0111: reserved 4’b1000: reserved
3:0	EXTI8	R/W	0	External interrupt configuration bit: 4’b0000: PA[8] pin 4’b0001: PB[8] pin 4’b0010: PC[8] pin 4’b0011: PD[8] pin 4’b0100: reserved 4’b0101: reserved 4’b0110: reserved 4’b0111: reserved 4’b1000: reserved

9.1.7 External Interrupt Configuration Register 3 (SYSCFG_EXTICR3)

Offset address: 0x1C

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:28	RSV	-	-	Reserved
27	EXTI15_WK_FLAG	R	0	EXTI15 wakeup source flag in Stop mode: 0: not woken up by this bit 1: woken up by this bit
26	EXTI14_WK_FLAG	R	0	EXTI14 wakeup source flag in Stop mode: 0: not woken up by this bit 1: woken up by this bit
25	EXTI13_WK_FLAG	R	0	EXTI13 wakeup source flag in Stop mode: 0: not woken up by this bit 1: woken up by this bit
24	EXTI12_WK_FLAG	R	0	EXTI12 wakeup source flag in Stop mode: 0: not woken up by this bit 1: woken up by this bit
23	EXTI15_WK_EDGE_SEL	R/W	0	EXTI15 wakeup edge selection in Stop mode: 0: rising edge 1: falling edge
22	EXTI14_WK_EDGE_SEL	R/W	0	EXTI14 wakeup edge selection in Stop mode: 0: rising edge 1: falling edge
21	EXTI13_WK_EDGE_SEL	R/W	0	EXTI13 wakeup edge selection in Stop mode: 0: rising edge 1: falling edge
20	EXTI12_WK_EDGE_SEL	R/W	0	EXTI12 wakeup edge selection in

Bit	Name	Attribute	Reset Value	Description
				Stop mode: 0: rising edge 1: falling edge
19	EXTI15_WKEN	R/W	0	EXTI15 wakeup enable in Stop mode: 0: disabled 1: enabled
18	EXTI14_WKEN	R/W	0	EXTI14 wakeup enable in Stop mode: 0: disabled 1: enabled
17	EXTI13_WKEN	R/W	0	EXTI13 wakeup enable in Stop mode: 0: disabled 1: enabled
16	EXTI12_WKEN	R/W	0	EXTI12 wakeup enable in Stop mode: 0: disabled 1: enabled
15:12	EXTI15	R/W	0	External interrupt configuration bit: 4’b0000: PA[15] pin 4’b0001: PB[15] pin 4’b0010: PC[15] pin 4’b0011: PD[15] pin 4’b0100: reserved 4’b0101: reserved 4’b0110: reserved 4’b0111: reserved 4’b1000: reserved
11:8	EXTI14	R/W	0	External interrupt configuration bit: 4’b0000: PA[14] pin 4’b0001: PB[14] pin 4’b0010: PC[14] pin 4’b0011: PD[14] pin 4’b0100: reserved 4’b0101: reserved 4’b0110: reserved 4’b0111: reserved 4’b1000: reserved
7:4	EXTI13	R/W	0	External interrupt configuration bit:

Bit	Name	Attribute	Reset Value	Description
				4’b0000: PA[13] pin 4’b0001: PB[13] pin 4’b0010: PC[13] pin 4’b0011: PD[13] pin 4’b0100: reserved 4’b0101: reserved 4’b0110: reserved 4’b0111: reserved 4’b1000: reserved
3:0	EXTI12	R/W	0	External interrupt configuration bit: 4’b0000: PA[12] pin 4’b0001: PB[12] pin 4’b0010: PC[12] pin 4’b0011: reserved 4’b0100: reserved 4’b0101: reserved 4’b0110: reserved 4’b0111: reserved 4’b1000: reserved

9.1.8 External Interrupt Wakeup Configuration Register (SYSCFG_EXTIWR)

Offset address: 0x20

Reset value: 0x0002 0000

Bit	Name	Attribute	Reset Value	Description
31:18	RSV	-	-	Reserved
17	INT_SEL	R/W	1	Interrupt response mode: 0: reserved 1: one of multiple GPIOs selected as interrupt Note: This bit shall not be modified and must be 1.
16	IESEL	R/W	0	Port interrupt mode selection bit: 1: Stop mode

Bit	Name	Attribute	Reset Value	Description
				0: Run mode Notes: ● When the system is in Run mode, the system clock will not be turned off, and IESEL can be set to 0. At this time, the external signal source that triggers the port interrupt will generate an interrupt signal after synchronization with the system clock, which can filter out glitches of the external signal source. ● When the system is in Stop mode, the system clock will be turned off, and IESEL can be set to 1. At this time, the external signal source that triggers the port interrupt will generate an interrupt signal directly without filtering out glitches of the external signal source.
15	EXTI15_CLR	R/W	0	When the IESEL bit in the register is 1 and the wakeup source is EXTI15 (INT_SEL = 1), write this bit to 1 after the system wakes up to clear the GPIO wakeup interrupt. 1: clear GPIO wakeup interrupt after system wakeup 0: no operation
14	EXTI14_CLR	R/W	0	When the IESEL bit in the register is 1 and the wakeup source is EXTI14 (INT_SEL = 1), write this bit to 1 after the system wakes up to clear the GPIO wakeup interrupt. 1: clear GPIO wakeup interrupt after system wakeup 0: no operation
13	EXTI13_CLR	R/W	0	When the IESEL bit in the register is 1 and the wakeup source is EXTI13 (INT_SEL = 1), write this bit to 1 after the system wakes up to clear the GPIO wakeup interrupt. 1: clear GPIO wakeup interrupt after system

Bit	Name	Attribute	Reset Value	Description
				wakeup 0: no operation
12	EXTI12_CLR	R/W	0	When the IESEL bit in the register is 1 and the wakeup source is EXTI12 (INT_SEL = 1), write this bit to 1 after the system wakes up to clear the GPIO wakeup interrupt. 1: clear GPIO wakeup interrupt after system wakeup 0: no operation
11	EXTI11_CLR	R/W	0	When the IESEL bit in the register is 1 and the wakeup source is EXTI11 (INT_SEL = 1), write this bit to 1 after the system wakes up to clear the GPIO wakeup interrupt. 1: clear GPIO wakeup interrupt after system wakeup 0: no operation
10	EXTI10_CLR	R/W	0	When the IESEL bit in the register is 1 and the wakeup source is EXTI10 (INT_SEL = 1), write this bit to 1 after the system wakes up to clear the GPIO wakeup interrupt. 1: clear GPIO wakeup interrupt after system wakeup 0: no operation
9	EXTI9_CLR	R/W	0	When the IESEL bit in the register is 1 and the wakeup source is EXTI9 (INT_SEL = 1), write this bit to 1 after the system wakes up to clear the GPIO wakeup interrupt. 1: clear GPIO wakeup interrupt after system wakeup 0: no operation
8	EXTI8_CLR	R/W	0	When the IESEL bit in the register is 1 and the wakeup source is EXTI8 (INT_SEL = 1), write this bit to 1 after the system wakes up to clear the GPIO wakeup interrupt. 1: clear GPIO wakeup interrupt after system wakeup

Bit	Name	Attribute	Reset Value	Description
				0: no operation
7	EXTI7_CLR	R/W	0	When the IESEL bit in the register is 1 and the wakeup source is EXTI7 (INT_SEL = 1), write this bit to 1 after the system wakes up to clear the GPIO wakeup interrupt. 1: clear GPIO wakeup interrupt after system wakeup 0: no operation
6	EXTI6_CLR	R/W	0	When the IESEL bit in the register is 1 and the wakeup source is EXTI6 (INT_SEL = 1), write this bit to 1 after the system wakes up to clear the GPIO wakeup interrupt. 1: clear GPIO wakeup interrupt after system wakeup 0: no operation
5	EXTI5_CLR	R/W	0	When the IESEL bit in the register is 1 and the wakeup source is EXTI5 (INT_SEL = 1), write this bit to 1 after the system wakes up to clear the GPIO wakeup interrupt. 1: clear GPIO wakeup interrupt after system wakeup 0: no operation
4	EXTI4_CLR	R/W	0	When the IESEL bit in the register is 1 and the wakeup source is EXTI4 (INT_SEL = 1), write this bit to 1 after the system wakes up to clear the GPIO wakeup interrupt. 1: clear GPIO wakeup interrupt after system wakeup 0: no operation
3	EXTI3_CLR	R/W	0	When the IESEL bit in the register is 1 and the wakeup source is EXTI3 (INT_SEL = 1), write this bit to 1 after the system wakes up to clear the GPIO wakeup interrupt. 1: clear GPIO wakeup interrupt after system wakeup 0: no operation

Bit	Name	Attribute	Reset Value	Description
2	EXTI2_CLR	R/W	0	When the IESEL bit in the register is 1 and the wakeup source is EXTI2 (INT_SEL = 1), write this bit to 1 after the system wakes up to clear the GPIO wakeup interrupt. 1: clear GPIO wakeup interrupt after system wakeup 0: no operation
1	EXTI1_CLR	R/W	0	When the IESEL bit in the register is 1 and the wakeup source is EXTI1 (INT_SEL = 1), write this bit to 1 after the system wakes up to clear the GPIO wakeup interrupt. 1: clear GPIO wakeup interrupt after system wakeup 0: no operation
0	EXTI0_CLR	R/W	0	When the IESEL bit in the register is 1 and the wakeup source is EXTI0 (INT_SEL = 1), write this bit to 1 after the system wakes up to clear the GPIO wakeup interrupt. 1: clear GPIO wakeup interrupt after system wakeup 0: no operation

9.1.9 MISC Control Register (SYSCFG_MISCCR)

Offset address: 0x24

Reset value: 0x0000 0030

Bit	Name	Attribute	Reset Value	Description
31:10	RSV	-	-	Reserved
9	PC5_ADCINEN	R/W	0	PC5 ADC input enable: 0: disabled 1: enabled Note: The ADC input can only be activated if the corresponding register in the PMU is enabled simultaneously.

Bit	Name	Attribute	Reset Value	Description
8	PC4_ADCINEN	R/W	0	PC4 ADC input enable: 0: disabled 1: enabled Note: The ADC input can only be activated if the corresponding register in the PMU is enabled simultaneously.
7:6	RSV	-	-	Reserved
5	EFC_VOL_PD_EN	R/W	1	EFC power-down ready signal (Standby mode): 0: invalid 1: valid Note: Generally not modified.
4	EFC_CLK_PD_EN	R/W	1	EFC clock stop ready signal (Stop mode): 0: invalid 1: valid Note: Generally not modified.
3	RSV	-	-	Reserved
2	LVD_INT_EN	R/W	0	LVD interrupt enable: 0: LVD interrupt disabled 1: LVD interrupt enabled
1	RSV	-	-	Reserved
0	NMIEN	R/W	0	NMI enable: 0: NMI disabled 1: NMI enabled (LVD interrupt)

9.1.10 Systick Calibration Reference Value Register (SYSCFG_SCRVR)

Offset address: 0x28

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:24	RSV	-	-	Reserved
23:0	STCALIB	R/W	0	Systick calibration reference value

9.1.11 UART Transmission Invert Configuration Register (SYSCFG_UARTTICR)

Offset address: 0x0030

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:12	RSV	-	-	Reserved
11	USART7_TXINV	R/W	0	USART7 TX output signal inversion configuration register: 0: normal output 1: inverted output
10	USART7_RXINV	R/W	0	USART7 RX input signal inversion configuration register: 0: normal input 1: inverted input
9	USART6_TXINV	R/W	0	USART6 TX output signal inversion configuration register: 0: normal output 1: inverted output
8	USART6_RXINV	R/W	0	USART6 RX input signal inversion configuration register: 0: normal input 1: inverted input
7	UART3_TXINV	R/W	0	UART3 TX output signal inversion configuration register: 0: normal output 1: inverted output
6	UART3_RXINV	R/W	0	UART3 RX input signal inversion configuration register: 0: normal input 1: inverted input
5	UART2_TXINV	R/W	0	UART2 TX output signal inversion configuration register: 0: normal output 1: inverted output

Bit	Name	Attribute	Reset Value	Description
4	UART2_RXINV	R/W	0	UART2 RX input signal inversion configuration register: 0: normal input 1: inverted input
3	UART1_TXINV	R/W	0	UART1 TX output signal inversion configuration register: 0: normal output 1: inverted output
2	UART1_RXINV	R/W	0	UART1 RX input signal inversion configuration register: 0: normal input 1: inverted input
1	UART0_TXINV	R/W	0	UART0 TX output signal inversion configuration register: 0: normal output 1: inverted output
0	UART0_RXINV	R/W	0	UART0 RX input signal inversion configuration register: 0: normal input 1: inverted input

9.1.12 TIM14~16 Break Control Register (SYSCFG_TIM141516CFGR)

Offset address: 0x0034

Reset value: 0x0040 2010

Bit	Name	Attribute	Reset Value	Description
31	RSV	-	-	Reserved
30	TIM16_OPA2_EN	R/W	0	OPA2 comparator break output enable (TIM16): 0: disabled 1: enabled
29	TIM16_OPA1_EN	R/W	0	OPA1 comparator break output enable (TIM16):

Bit	Name	Attribute	Reset Value	Description
				0: disabled 1: enabled
28	TIM16_OPA0_EN	R/W	0	OPA0 comparator break output enable (TIM16): 0: disabled 1: enabled
27	TIM16_ACMP2_EN	R/W	0	ACMP2 comparator break output enable (TIM16): 0: disabled 1: enabled
26	TIM16_ACMP1_EN	R/W	0	ACMP1 comparator break output enable (TIM16): 0: disabled 1: enabled
25	TIM16_ACMP0_EN	R/W	0	ACMP0 comparator break output enable (TIM16): 0: disabled 1: enabled
24	TIM16_LOCKUP_EN	R/W	0	LOCKUP break input enable (TIM16): 0: disabled 1: enabled
23	TIM16_LVD_EN	R/W	0	LVD break input enable (TIM16): 0: disabled 1: enabled
22	TIM16_BKIN_EN	R/W	1	TIM16 external BKIN break input enable: 0: disabled 1: enabled
21	TIM15_OPA2_EN	R/W	0	OPA2 comparator break output enable (TIM15): 0: disabled 1: enabled
20	TIM15_OPA1_EN	R/W	0	OPA1 comparator break output enable (TIM15): 0: disabled 1: enabled
19	TIM15_OPA0_EN	R/W	0	OPA0 comparator break output enable

Bit	Name	Attribute	Reset Value	Description
				(TIM15): 0: disabled 1: enabled
18	TIM15_ACMP2_EN	R/W	0	ACMP2 comparator break output enable (TIM15): 0: disabled 1: enabled
17	TIM15_ACMP1_EN	R/W	0	ACMP1 comparator break output enable (TIM15): 0: disabled 1: enabled
16	TIM15_ACMP0_EN	R/W	0	ACMP0 comparator break output enable (TIM15): 0: disabled 1: enabled
15	TIM15_LOCKUP_EN	R/W	0	LOCKUP break input enable (TIM15): 0: disabled 1: enabled
14	TIM15_LVD_EN	R/W	0	LVD break input enable (TIM15): 0: disabled 1: enabled
13	TIM15_BKIN_EN	R/W	1	TIM15 external BKIN break input enable: 0: disabled 1: enabled
12	TIM14_OPA2_EN	R/W	0	OPA2 comparator break output enable (TIM14): 0: disabled 1: enabled
11	TIM14_OPA1_EN	R/W	0	OPA1 comparator break output enable (TIM14): 0: disabled 1: enabled
10	TIM14_OPA0_EN	R/W	0	OPA0 comparator break output enable (TIM14): 0: disabled 1: enabled

Bit	Name	Attribute	Reset Value	Description
9	TIM14_ACMP2_EN	R/W	0	ACMP2 comparator break output enable (TIM14): 0: disabled 1: enabled
8	TIM14_ACMP1_EN	R/W	0	ACMP1 comparator break output enable (TIM14): 0: disabled 1: enabled
7	TIM14_ACMP0_EN	R/W	0	ACMP0 comparator break output enable (TIM14): 0: disabled 1: enabled
6	TIM14_LOCKUP_EN	R/W	0	LOCKUP break input enable (TIM14): 0: disabled 1: enabled
5	TIM14_LVD_EN	R/W	0	LVD break input enable (TIM14): 0: disabled 1: enabled
4	TIM14_BKIN_EN	R/W	1	TIM14 external BKIN break input enable: 0: disabled 1: enabled
3	RSV	-	-	Reserved
2	TIM16_BKIN_INV	R/W	0	TIM16 BKIN input inversion signal: 0: non-inverted 1: inverted
1	TIM15_BKIN_INV	R/W	0	TIM15 BKIN input inversion selection: 0: non-inverted 1: inverted
0	TIM14_BKIN_INV	R/W	0	TIM14 BKIN input inversion signal: 0: non-inverted 1: inverted

9.1.13 Voltage Switching Configuration Status Register (SYSCFG_VCSR)

Offset address: 0x40

Reset value: 0x0001 FFFF

Bit	Name	Attribute	Reset Value	Description
31:17	RSV	-	-	Reserved
16	VOL_STB	R	1	Voltage switching stabilization status register: 0: switching process 1: voltage switching being stable
15:0	VOL_STB_CNT	R/W	16'hFFFF	Voltage switching stabilization time (software-configurable, calculated with RCH at a 24 MHz clock cycle)

Notes:

1. Under normal operation at 1.1 V, this register group does not need to be considered.
2. During voltage switching (from 1.1 V to 0.9 V, or from 0.9 V to 1.1 V), the system clock must be set to RCH (24 MHz).

9.1.14 Low-power Mode Status Register (SYSREG_LPMSR)

Offset address: 0x48

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:5	RSV	-	-	Reserved
4	STDBY_ENS	R	0	Standby mode enable status: 0: Standby mode disabled 1: Standby mode enabled
3	STOP_CLK_SELS	R	0	Stop mode clock selection enable status: 0: when entering Stop mode, switch to RCH first, and still use RCH on wakeup 1: keep the working clock before entering Stop mode unchanged (e.g. PLL

Bit	Name	Attribute	Reset Value	Description
				clock)
2	BKSRAMOFFS	R	0	BKSRAM and other logics power-down control status: 0: BKSRAM, IWDT, LPUART and LPTimers 0–1 not powered down 0: BKSRAM, IWDT, LPUART and LPTimers 0–1 powered down
1:0	PMU_MODES	R	0	PMU mode register status: 2'b00: Run mode 2'b01: Stop mode 2'b10: Standby mode/Power-down mode 2'b11: DeepStandby mode/Deep power-down mode

Note: This register is read-only and reflects the status of the corresponding registers in the PMU domain, facilitating a quicker review of the status after waking up.

9.1.15 Reserved Register 0 (SYSCFG_RSV0R)

Offset address: 0x50

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:0	RSV0	R/W	0	Reserved register 0 (used for software to store status information in special applications)

Note: The reset of this register can only be influenced by power-on or power-off reset sources.

9.1.16 Reserved Register 1 (SYSCFG_RSV1R)

Offset address: 0x54

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:0	RSV1	R/W	0	Reserved register 1 (used for software to store status information in special applications)

Note: The reset of this register can only be influenced by power-on or power-off reset sources.

10 Cyclic Redundancy Check Calculation Unit (CRC)

10.1 Overview

The CRC controller can perform CRC calculation using various polynomials based on CRC32 and CRC16 standards.

10.2 Main Features

- Uses the following polynomials:

  $x^{16} + x^{12} + x^5 + 1$

  $x^{16} + x^{15} + x^2 + 1$

  $x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x^1 + 1$

- Can input byte, half-word and full-word data
- The input data can be reversed by bit, half-word, byte, initial value and result value.

10.3 Functional Description

10.3.1 CRC Calculation

CRC calculation can be performed in multiple formats.

10.3.2 Common CRC Formats

Table 10-1: Common CRC Formats

Name	Polynomial	Initial value	Input data reversed by byte	Output data reversed	Result XOR
CRC-16 / CCITT	1021	0	Y	Y	0
CRC-16 / CCITT-FALSE	1021	FFFF	N	N	0
CRC-16 / X25	1021	FFFF	Y	Y	FFFF
CRC-16 / XMODEM	1021	0	N	N	0

Name	Polynomial	Initial value	Input data reversed by byte	Output data reversed	Result XOR
CRC-16 / IBM	8005	0	Y	Y	0
CRC-16 / MAXIM	8005	0	Y	Y	FFFF
CRC-16 / USB	8005	FFFF	Y	Y	FFFF
CRC-16 / MODBUS	8005	FFFF	Y	Y	0
CRC-32	04C11DB7	FFFFFFFF	Y	Y	FFFFFFFF
CRC-32 / MPEG-2	04C11DB7	FFFFFFFF	N	N	0

Note: XOR function for results is not provided in this module. If necessary, please take out the results and then calculate them separately.

10.4 Register Description

Register base address: 0x40A0_0000

The registers are listed below:

Table 10-2: List of CRC Registers

Offset Address	Name	Description
0x00	CRC_DATA	Data register
0x04	CRC_CFG	Configuration register
0x08	CRC_INIT	Initial value register

10.4.1 Data Register (CRC_DATA)

Offset address: 0x00

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:0	DATA	R/W	0x0	This register is used to write new data to the CRC calculator, and it holds the previous CRC calculation result when it is read.

10.4.2 Configuration Register (CRC_CFG)

Offset address: 0x04

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
30:9	RSV	-	-	Reserved
8	INIT_REV	R/W	0x0	Initial value reverse: 0: not reversed 1: reversed
7	DOUT_REV	R/W	0x0	Output data reverse: 0: not reversed 1: reversed
6	RSV	-	-	Reserved
5	DIN_REV	R/W	0x0	Input data reverse: 0: not reversed 1: reversed
4:3	WIDTH_DIN	R/W	0x0	Input data width selection: 00: 8 bits 01: 16 bits 10/11: 32 bits
2:1	POL	R/W	0x0	CRC polynomial selection: 00: CRC16-1021 $(x^{16} + x^{12} + x^5 + 1)$; 01: CRC16-8005 $(x^{16} + x^{15} + x^2 + 1)$; 10/11: CRC32-04C11DB7 $(x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x^1 + 1)$
0	RESET	W	0x0	This bit is set to clear the calculation result and set the data register to the initial value, and it is automatically cleared.

10.4.3 Initial Value Register (CRC_INIT)

Offset address: 0x08

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:0	INIT	R/W	0x0	Write the initial value.

10.5 Operation Procedure

1. Configure CRC_INIT[31: 0] to determine the initial value.
2. Configure the CRC_CFG register to select the CRC polynomial, data width and data reverse order, and load the initial value.
3. Write data to CRC_DATA[31: 0], which can be written continuously.
4. Fetch the CRC calculation result from CRC_DATA[31: 0].

11 Direct Memory Access Controller (DMA)

11.1 Overview

The DMA is used to provide high-speed data transfer between peripherals and memory as well as from memory to memory. Data can be quickly moved by DMA without any CPU actions, which keeps the CPU resources free for other operations, thus improving the system efficiency.

11.2 Main Features

- Controllable data transmission among multiple modules
- Memory-to-memory, memory-to-peripheral, peripheral-to-memory, and peripheral-to-peripheral data transfers
- Provided with 8 DMA channels
- Configurable bit width and block length of data transfer
- Channel 0 & 1 with 4 x 32-bit FIFOs, channel 2–7 with 2 x 32-bit FIFOs
- Block length up to 4095
- Invariant transmission and incremental / decremental transmission of source address
- Invariant transmission and incremental / decremental transmission of destination address

11.3 Functional Description

11.3.1 Flow Control

DMA supports transfers where the DMA itself acts as a flow controller. For source and destination transfer, the following modes are supported:

- Memory-to-memory

- Memory-to-peripheral
- Peripheral-to-memory
- Peripheral-to-peripheral

11.3.2 Handshake Signal

When both the source and destination are set as peripheral, it is necessary to specify the handshake signal number used by the peripheral. DMA will initiate a transfer only when the handshake signal is valid.

The handshake signal can be generated not only by peripheral hardware, but also by software writing DMA register.

For the mapping of different DMA requests, refer to the Chapter DMAMUX.

Notes:

- The handshake signal (peripheral) and the 8 channels of DMA can be freely configured according to the application.
- The 8 channels of DMA can be freely configured by software to adapt to different peripheral applications.

11.4 Register Description

DMA0 register base address: 0x4070_0000

The registers are listed below:

Table 11-1: List of DMA Registers

Offset Address	Name	Description
0x00	DMA_SAR0	Channel 0 source address register
0x08	DMA_DAR0	Channel 0 destination address register
0x18	DMA_CTL0	Channel 0 control register
0x1C	DMA_CTLH0	Channel 0 control register
0x40	DMA_CFG0	Channel 0 configuration register

Offset Address	Name	Description
0x44	DMA_CFGH0	Channel 0 configuration register
0x58	DMA_SAR1	Channel 1 source address register
0x60	DMA_DAR1	Channel 1 destination address register
0x70	DMA_CTL1	Channel 1 control register
0x74	DMA_CTLH1	Channel 1 control register
0x98	DMA_CFG1	Channel 1 configuration register
0x9C	DMA_CFGH1	Channel 1 configuration register
0xB0	DMA_SAR2	Channel 2 source address register
0xB8	DMA_DAR2	Channel 2 destination address register
0xC8	DMA_CTL2	Channel 2 control register
0xCC	DMA_CTLH2	Channel 2 control register
0xF0	DMA_CFG2	Channel 2 configuration register
0XF4	DMA_CFGH2	Channel 2 configuration register
0x108	DMA_SAR3	Channel 3 source address register
0x110	DMA_DAR3	Channel 3 destination address register
0x120	DMA_CTL3	Channel 3 control register
0x124	DMA_CTLH3	Channel 3 control register
0x148	DMA_CFG3	Channel 3 configuration register
0x14C	DMA_CFGH3	Channel 3 configuration register
0x160	DMA_SAR4	Channel 4 source address register
0x168	DMA_DAR4	Channel 4 destination address register
0x178	DMA_CTL4	Channel 4 control register
0x17C	DMA_CTLH4	Channel 4 control register
0x1A0	DMA_CFG4	Channel 4 configuration register
0x1A4	DMA_CFGH4	Channel 4 configuration register
0x1B8	DMA_SAR5	Channel 5 source address register
0x1C0	DMA_DAR5	Channel 5 destination address register
0x1D0	DMA_CTL5	Channel 5 control register
0x1D4	DMA_CTLH5	Channel 5 control register
0x1F8	DMA_CFG5	Channel 5 configuration register
0x1FC	DMA_CFGH5	Channel 5 configuration register
0x210	DMA_SAR6	Channel 6 source address register
0x218	DMA_DAR6	Channel 6 destination address register
0x228	DMA_CTL6	Channel 6 control register
0x22C	DMA_CTLH6	Channel 6 control register
0x250	DMA_CFG6	Channel 6 configuration register

Offset Address	Name	Description
0x254	DMA_CFGH6	Channel 6 configuration register
0x268	DMA_SAR7	Channel 7 source address register
0x270	DMA_DAR7	Channel 7 destination address register
0x280	DMA_CTL7	Channel 7 control register
0x284	DMA_CTLH7	Channel 7 control register
0x2A8	DMA_CFG7	Channel 7 configuration register
0x2AC	DMA_CFGH7	Channel 7 configuration register
0x2C0	DMA_RAWTFR	Raw transfer complete interrupt register
0x2C8	DMA_RAWBLOCK	Raw block transfer interrupt register
0x2D0	DMA_RAWSRCTRAN	Raw source transfer interrupt register
0x2D8	DMA_RAWDSTTRAN	Raw destination transfer interrupt register
0x2E0	DMA_RAWERR	Raw error interrupt register
0x2E8	DMA_STATUSTFR	Transfer complete interrupt status register
0x2F0	DMA_STATUSBLOCK	Block transfer interrupt status register
0x2F8	DMA_STATUSSRCTRAN	Source transfer interrupt status register
0x300	DMA_STATUSDSTTRAN	Destination transfer interrupt status register
0x308	DMA_STATUSERR	Error interrupt status register
0x310	DMA_MASKTFR	Transfer complete interrupt mask register
0x318	DMA_MASKBLOCK	Block transfer interrupt mask register
0x320	DMA_MASKSRCTRAN	Source transfer interrupt mask register
0x328	DMA_MASKDSTTRAN	Destination transfer interrupt mask register
0x330	DMA_MASKERR	Error interrupt mask register
0x338	DMA_CLEARTFR	Transfer complete interrupt clear register
0x340	DMA_CLEARBLOCK	Block transfer interrupt clear register
0x348	DMA_CLEARSRCTRAN	Source transfer interrupt clear register
0x350	DMA_CLEARDSTTRAN	Destination transfer interrupt clear register
0x358	DMA_CLEARERR	Error interrupt clear register
0x360	DMA_STATUSINT	Interrupt status register
0x368	DMA_REQSRCREG	Source transfer request signal register
0x370	DMA_REQDSTREG	Destination transfer request signal register
0x378	DMA_SGLREQSRCREG	Source transfer single signal register
0x380	DMA_SGLREQDSTREG	Destination transfer single signal register
0x388	DMA_LSTSRCREG	Source transfer last signal register
0x390	DMA_LSTDSTREG	Destination transfer last signal register
0x398	DMA_CFGREG	DMA module enable register
0x3A0	DMA_CHENREG	Channel enable register

11.4.1 Source Address Register (DMA_SARx)

Offset address: 0x00/0x58/0xB0/0x108/0x160/0x1B8/0x210/0x268

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:0	SAR	R/W	0x0	DMA transfer source address, automatically updated

Note: This register must be set when the channel is disabled (CH_EN = 0 for DMA_CHENREG).

11.4.2 Destination Address Register (DMA_DARx)

Offset address: 0x08/0x60/0xB8/0x110/0x168/0x1C0/0x218/0x270

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:0	DAR	R/W	0x0	DMA transfer destination address, automatically updated

Note: This register must be set when the channel is disabled (CH_EN = 0 for DMA_CHENREG).

11.4.3 Control Register (DMA_CTLx)

Offset address: 0x18/0x70/0xC8/0x120/0x178/0x1D0/0x228/0x280

Reset value: 0x0030 4801

Bit	Name	Attribute	Reset Value	Description
31:22	RSV	-	-	Reserved
21:20	TT_FC	R/W	0x3	Transfer type and flow control: 0x0: memory-to-memory 0x1: memory-to-peripheral 0x2: peripheral-to-memory 0X3: peripheral-to-peripheral
19:16	RSV	-	-	Reserved
15:14	SRC_MSIZE	R/W	0x1	Burst length of source transfer, the number of transfers read each time the source peripheral request handshake signal is available: 0x0: 1

Bit	Name	Attribute	Reset Value	Description
				0x1: 4 0x2: 8 (only for channel 0 and channel 1) 0x3：reserved
13	RSV	-	-	Reserved
12:11	DST_MSIZE	R/W	0x1	Burst length of destination transfer, the number of transfers written each time the destination peripheral request handshake signal is available: 0x0: 1 0x1: 4 0x2: 8 (only for channel 0 and channel 1) 0x3：reserved
10:9	SINC	R/W	0x0	Source address incrementation: 0x0: increment 0x1: decrement 0x2/0x3: invariant
8:7	DINC	R/W	0x0	Destination address incrementation: 0x0: increment 0x1: decrement 0x2/0x3: invariant
6	RSV	-	-	Reserved
5:4	SRC_TR_WIDTH	R/W	0x0	Source transfer data width: 0x0: 8 bits 0x1: 16 bits 0x2: 32 bits
3	RSV	-	-	Reserved
2:1	DST_TR_WIDTH	R/W	0x0	Destination transfer data width: 0x0: 8 bits 0x1: 16 bits 0x2: 32 bits
0	INT_EN	R/W	0x1	Interrupt enable: 0x0: disabled 0X1: enabled

Note: This register must be set when the channel is disabled (CH_EN = 0 for DMA_CHENREG).

11.4.4 Control Register (DMA_CTLHx)

Offset address: 0x1C/0x74/0xCC/0x124/0x17C/0x1D4/0x22C/0x284

Reset value: 0x0000 0002

Bit	Name	Attribute	Reset Value	Description
31:12	RSV	-	-	Reserved
11:0	BLOCK_TS	R/W	0x2	The block transfer source data size is in the unit of source transfer width, the total number of transfer is determined by the source transfer, and the number of destination transfer varies automatically according to the bit widths of source and destination transfers. Once the transfer begins, no matter what flow control is, the readback value is the total number of data that has been read from the source.

Note: This register must be set when the channel is disabled (CH_EN = 0 for DMA_CHENREG).

11.4.5 Configuration Register (DMA_CFGx)

Offset address: 0x40/0x98/0xF0/0x148/0x1A0/0x1F8/0x250/0x2A8

Reset value: 0x0000_0e00 + 0x20 * x (x = 0−7)

Bit	Name	Attribute	Reset Value	Description
31	RELOAD_DST	R/W	0x0	Auto-reload a destination transfer: 0x0: disabled 0x1: enabled; when a block transfer ends, reset DARx to its initial value and start a new block transfer.
30	RELOAD_SRC	R/W	0x0	Auto-reload a source transfer: 0x0: disabled 0x1: enabled; when a block transfer ends, reset SARx to its initial value and start a new block transfer.
29:20	RSV	-	-	Reserved

Bit	Name	Attribute	Reset Value	Description
19	SRC_HS_POL	R/W	0x0	Please use 0x0: handshake signal active high
18	DST_HS_POL	R/W	0x0	Please use 0x0: handshake signal active high
17:12	RSV	-	-	Reserved
11	HS_SEL_SRC	R/W	0x1	Source transfer handshake signal selection: 0x0: hardware handshake 0x1: software handshake
10	HS_SEL_DST	R/W	0x1	Destination transfer handshake signal selection: 0x0: hardware handshake 0x1: software handshake
9	FIFO_EMPTY	R	0x1	FIFO empty indication for this channel: 0x0: FIFO non-empty 0x1: FIFO empty
8	CH_SUSP	R/W	0x0	Suspend transfer on this channel: 0x0: normal transfer 0x1: transfer suspended
7:5	CH_PRIOR	R/W	Different for each channel	Specify the channel priority, with 0 as the lowest
4:0	RSV	-	-	Reserved

Note: This register must be set when the channel is disabled (CH_EN = 0 for DMA_CHENREG).

11.4.6 Configuration Register (DMA_CFGHx)

Offset address: 0x44/0x9C/0xF4/0x14C/0x1A4/0x1FC/0x254/0x2AC

Reset value: 0x0000 0004

Bit	Name	Attribute	Reset Value	Description
31:15	RSV	-	-	Reserved
14:11	DEST_PER	R/W	0x0	Destination peripheral handshake signal number: This item is ignored when the destination is specified as memory in TT_FC.
10:7	SRC_PER	R/W	0x0	Source peripheral handshake signal number:

Bit	Name	Attribute	Reset Value	Description
				This item is ignored when the source is specified as memory in TT_FC.
6:5	RSV	-	-	Reserved
4:2	PROTCTL	R/W	0x1	Drive HPROT[3:1]
1	FIFO_MODE	R/W	0x0	Specify how much available data/space is needed for initiating a burst transfer: 0x0: transfer can be initiated as long as there is one available data/space. 0x1: destination transfer can be initiated when the available data is greater than or equal to half the FIFO depth; source transfer can be initiated when the space is greater than or equal to half the FIFO depth; except at the end of a burst or block transfer.
0	FCMODE	R/W	0x0	Flow control mode / data prefetch: 0x0: prefetch enabled, read when source data transfer is available. 0x1: prefetch disabled, source data transfer not initiated until destination data transfer is completed.

Note: This register must be set when the channel is disabled (CH_EN = 0 for DMA_CHENREG).

11.4.7 Raw Transfer Complete Interrupt Register (DMA_RAWTFR)

Offset address: 0x2C0

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:8	RSV	-	-	Reserved
7:0	RAWTFR	R/W	0x0	Bit x indicates the raw transfer complete interrupt (TFR) status of channel x. This interrupt will be triggered when the channel completes all transfers. Note: Writing directly to this register is not recommended during normal use.

11.4.8 Raw Block Transfer Interrupt Register (DMA_RAWBLOCK)

Offset address: 0x2C8

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:8	RSV	-	-	Reserved
7:0	RAWBLOCK	R/W	0x0	Bit x indicates the raw block transfer interrupt status of channel x. This interrupt will be triggered when the channel completes a block transfer. Note: Writing directly to this register is not recommended during normal use.

11.4.9 Raw Source Transfer Interrupt Register (DMA_RAWSRCTRAN)

Offset address: 0x2D0

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:8	RSV	-	-	Reserved
7:0	RAWSRCTRAN	R/W	0x0	Bit x indicates the raw source transfer interrupt status of channel x. This interrupt will be triggered when the channel completes a burst/single transfer in response to a handshake signal from the source peripheral. Note: Writing directly to this register is not recommended during normal use.

11.4.10 Raw Destination Transfer Interrupt Register (DMA_RAWDSTTRAN)

Offset address: 0x2D8

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:8	RSV	-	-	Reserved
7:0	RAWDSTTRAN	R/W	0x0	Bit x indicates the raw destination transfer interrupt status of channel x. This interrupt will be triggered when the channel completes a burst/single transfer in response to a handshake signal from the destination peripheral. Note: Writing directly to this register is not recommended during normal use.

11.4.11 Raw Error Interrupt Register (DMA_RAWERR)

Offset address: 0x2E0

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:8	RSV	-	-	Reserved
7:0	RAWERR	R/W	0x0	Bit x indicates the raw block error interrupt status of channel x. This interrupt will be triggered when the channel receives an error response from HRSP during transfer, causing the transfer to be canceled and the channel to be shut down. Note: Writing directly to this register is not recommended during normal use.

11.4.12 Transfer Complete Interrupt Status Register (DMA_STATUSTFR)

Offset address: 0x2E8

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:8	RSV	-	-	Reserved
7:0	STATUSTFR	R	0x0	Bit x indicates the transfer complete interrupt (TFR) output status of channel x. This interrupt will be triggered when the channel completes all transfers. This register will not be set to 1 if the interrupt is masked.

Note: Writing to this register is prohibited.

11.4.13 Block Transfer Interrupt Status Register (DMA_STATUSBLOCK)

Offset address: 0x2F0

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:8	RSV	-	-	Reserved
7:0	STATUSBLOCK	R	0x0	Bit x indicates the block transfer interrupt output status of channel x. This interrupt is triggered when the channel completes a block transfer. This register will not be set to 1 if the interrupt is masked.

Note: Writing to this register is prohibited.

11.4.14 Source Transfer Interrupt Status Register (DMA_STATUSSRCTRAN)

Offset address: 0x2F8

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:8	RSV	-	-	Reserved
7:0	STATUSSRCTRAN	R	0x0	Bit x indicates the source transfer interrupt output status of channel x. This interrupt will be triggered when the channel completes a burst/single transfer in response to a handshake signal from the source peripheral. This register will not be set to 1 if the interrupt is masked.

Note: Writing to this register is prohibited.

11.4.15 Destination Transfer Interrupt Status Register (DMA_STATUSDSTRAN)

Offset address: 0x300

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:8	RSV	-	-	Reserved
7:0	STATUSDSTTRAN	R	0x0	Bit x indicates the destination transfer interrupt output status of channel x. This interrupt will be triggered when the channel completes a burst/single transfer in response to a handshake signal from the destination peripheral. This register will not be set to 1 if the interrupt is masked.

Note: Writing to this register is prohibited.

11.4.16 Error Interrupt Status Register (DMA_STATUSERR)

Offset address: 0x308

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:8	RSV	-	-	Reserved
7:0	STATUSERR	R	0x0	Bit x indicates the error interrupt output status of channel x. This interrupt will be triggered when the channel receives an error response from HRSP during transfer, causing the transfer to be canceled and the channel to be shut down. This register will not be set to 1 if the interrupt is masked.

Note: Writing to this register is prohibited.

11.4.17 Transfer Complete Interrupt Mask Register (DMA_MASKTFR)

Offset address: 0x310

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:16	RSV	-	-	Reserved
15:8	MASKTFR_WE	W	0x0	Bit (x + 8) is the write activation bit for channel x: 0x0: write disabled 0x1: write enabled Valid for once, read as 0
7:0	MASKTFR	R/W	0x0	Bit x indicates the transfer complete interrupt (TFR) mask status of channel x: 0x0: masked 0x1: not masked

For example:

- To set bit 0 to 1, write 0x101 to this register without affecting bit 1.
- To set bit 1 to 0, write 0x200 to this register without affecting bit 0.
- To set bit 0 and bit 1 to 1, write 0x303 to this register.

11.4.18 Block Transfer Interrupt Mask Register (DMA_MASKBLOCK)

Offset address: 0x318

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:16	RSV	-	-	Reserved
15:8	MASKBLOCK_WE	W	0x0	Bit (x + 8) is the write activation bit for channel x: 0x0: write disabled 0x1: write enabled Valid for once, read as 0
7:0	MASKBLOCK	R/W	0x0	Bit x indicates the block transfer interrupt mask status of channel x: 0x0: masked 0x1: not masked

11.4.19 Source Transfer Interrupt Mask Register (DMA_MASKSRCTRAN)

Offset address: 0x320

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:16	RSV	-	-	Reserved
15:8	MASKSRCTRAN_WE	W	0x0	Bit (x + 8) is the write activation bit for channel x: 0x0: write disabled 0x1: write enabled Valid for once, read as 0

Bit	Name	Attribute	Reset Value	Description
7:0	MASKSRCTRAN	R/W	0x0	Bit x indicates the source transfer interrupt mask status of channel x: 0x0: masked 0x1: not masked

11.4.20 Destination Transfer Interrupt Mask Register (DMA_MASKDSTTRAN)

Offset address: 0x328

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:16	RSV	-	-	Reserved
15:8	MASKDSTTRAN_WE	W	0x0	Bit (x + 8) is the write activation bit for channel x: 0x0: write disabled 0x1: write enabled Valid for once, read as 0
7:0	MASKDSTTRAN	R/W	0x0	Bit x indicates the destination transfer interrupt mask status of channel x: 0x0: masked 0x1: not masked

11.4.21 Error Interrupt Mask Register (DMA_MASKERR)

Offset address: 0x330

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:16	RSV	-	-	Reserved
15:8	MASKERR_WE	W	0x0	Bit (x + 8) is the write activation bit for channel x: 0x0: write disabled 0x1: write enabled Valid for once, read as 0

Bit	Name	Attribute	Reset Value	Description
7:0	MASKERR	R/W	0x0	Bit x indicates the error interrupt mask status of channel x: 0x0: masked 0x1: not masked

11.4.22 Transfer Complete Interrupt Clear Register (DMA_CLEARTFR)

Offset address: 0x338

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:8	RSV	-	-	Reserved
7:0	CLEARTFR	W1C	0x0	Writing 1 to bit x clears the transfer complete interrupt (TFR) for channel x.

Note: Reading this register is prohibited.

11.4.23 Block Transfer Interrupt Clear Register (DMA_CLEARBLOCK)

Offset address: 0x340

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:8	RSV	-	-	Reserved
7:0	CLEARBLOCK	W1C	0x0	Writing 1 to bit x clears the block transfer interrupt for channel x.

Note: Reading this register is prohibited.

11.4.24 Source Transfer Interrupt Clear Register (DMA_CLEARSRCTRAN)

Offset address: 0x348

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:8	RSV	-	-	Reserved
7:0	CLEARSRCTRAN	W1C	0x0	Writing 1 to bit x clears the source transfer interrupt for channel x.

Note: Reading this register is prohibited.

11.4.25 Destination Transfer Interrupt Clear Register (DMA_CLEARDSTTRAN)

Offset address: 0x350

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:8	RSV	-	-	Reserved
7:0	CLEARDSTTRAN	W1C	0x0	Writing 1 to bit x clears the destination transfer interrupt for channel x.

Note: Reading this register is prohibited.

11.4.26 Error Interrupt Clear Register (DMA_CLEARERR)

Offset address: 0x358

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:8	RSV	-	-	Reserved
7:0	CLEARERR	W1C	0x0	Writing 1 to bit x clears the error interrupt for channel x.

Note: Reading this register is prohibited.

11.4.27 Interrupt Status Register (DMA_STATUSINT)

Offset address: 0x360

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:5	RSV	-	-	Reserved

Bit	Name	Attribute	Reset Value	Description
4	ERR	R	0x0	This bit is set when any of bits [7: 0] (channels 7–0) in the STATUSERR register is "1".
3	DSTTRAN	R	0x0	This bit is set when any of bits [7: 0] (channels 7–0) in the STATUSDSTTRAN register is "1".
2	SRCTRAN	R	0x0	This bit is set when any of bits [7: 0] (channels 7–0) in the STATUSSRCTRAN register is "1".
1	BLOCK	R	0x0	This bit is set when any of bits [7: 0] (channels 7–0) in the STATUSBLOCK register is "1".
0	TFR	R	0x0	This bit is set when any of bits [7: 0] (channels 7–0) in the STATUSTFR register is "1".

Note: Writing to this register is prohibited.

11.4.28 Source Transfer Request Signal Register (DMA_REQSRCREG)

Offset address: 0x368

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:16	RSV	-	-	Reserved
15:8	SRC_REQ_WE	W	0x0	Bit (x + 8) is the write activation bit for channel x: 0x0: write disabled 0x1: write enabled Valid for once, read as 0
7:0	SRC_REQ	R/W	0x0	Bit x is the request handshake signal for source transfer of channel x.

11.4.29 Destination Transfer Request Signal Register (DMA_REQDSTREG)

Offset address: 0x370

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:16	RSV	-	-	Reserved

Bit	Name	Attribute	Reset Value	Description
15:8	DST_REQ_WE	W	0x0	Bit (x + 8) is the write activation bit for channel x: 0x0: write disabled 0x1: write enabled Valid for once, read as 0
7:0	DST_REQ	R/W	0x0	Bit x is the request handshake signal for destination transfer of channel x.

11.4.30 Source Transfer Single Signal Register (DMA_SGLREQSRCREG)

Offset address: 0x378

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:16	RSV	-	-	Reserved
15:8	SRC_SGLREQ_WE	W	0x0	Bit (x + 8) is the write activation bit for channel x: 0x0: write disabled 0x1: write enabled Valid for once, read as 0
7:0	SRC_SGLREQ	R/W	0x0	Bit x is the single handshake signal for source transfer of channel x.

11.4.31 Destination Transfer Single Signal Register (DMA_SGLREQDSTREG)

Offset address: 0x380

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:16	RSV	-	-	Reserved
15:8	DST_SGLREQ_WE	W	0x0	Bit (x + 8) is the write activation bit for channel x: 0x0: write disabled 0x1: write enabled Valid for once, read as 0

Bit	Name	Attribute	Reset Value	Description
7:0	DST_SGLREQ	R/W	0x0	Bit x is the single handshake signal for destination transfer of channel x.

11.4.32 Source Transfer Last Signal Register (DMA_LSTSRCRE)

Offset address: 0x388

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:16	RSV	-	-	Reserved
15:8	LSTSRC_WE	W	0x0	Bit (x + 8) is the write activation bit for channel x: 0x0: write disabled 0x1: write enabled Valid for once, read as 0
7:0	LSTSRC	R/W	0x0	Bit x is the last handshake signal for source transfer of channel x.

11.4.33 Destination Transfer Last Signal Register (DMA_LSTDSTREG)

Offset address: 0x390

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:16	RSV	-	-	Reserved
15:8	LSTDST_WE	W	0x0	Bit (x + 8) is the write activation bit for channel x: 0x0: write disabled 0x1: write enabled Valid for once, read as 0
7:0	LSTDST	R/W	0x0	Bit x is the last handshake signal for destination transfer of channel x.

11.4.34 DMA Module Enable Register (DMA_CFGREG)

Offset address: 0x398

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:1	RSV	-	-	Reserved
0	DMA_EN	R/W	0x0	DMA enable: 0x0: DMA module function disabled 0x1: DMA module function enabled; this bit shall be enabled before enabling the channel.

11.4.35 Channel Enable Register (DMA_CHENREG)

Offset address: 0x3A0

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:16	RSV	-	-	Reserved
15:8	CH_EN_WE	W	0x0	Bit (x + 8) is the write activation bit for channel x: 0x0: write disabled 0x1: write enabled Valid for once, read as 0
7:0	CH_EN	R/W	0x0	Bit x is the enable bit for channel x. This bit will be automatically set to 0 upon completion of transfer.

For example:

- To set bit 0 to 1, write 0x101 to this register without affecting bit 1.
- To set bit 1 to 0, write 0x200 to this register without affecting bit 0.
- To set bit 0 and bit 1 to 1, write 0x303 to this register.

11.5 Operation Procedure

11.5.1 Basic Hardware Flow Control

1. Configure the DMA_SARx register to specify the source address.
2. Configure the DMA_DARx register to specify the destination address.
3. Configure the DMA_CTLx register:
   A. to enable interrupt if required.
   B. to set the transfer data width and size.
   C. to select the flow control type; note that SRAM has no handshake signal, and SRAM is identified as memory in TT_FC.
   D. to select whether the address is incremented or decremented.
   E. to select the burst transfer length.
4. Configure the DMA_CFGx register:
   A. to select the channel priority.
   B. to set whether to automatically restart the transmission.
   C. to select the handshake signal.
   D. to set the HS_SEL field to 0.
5. Configure the DMA_MASKBLOCK register to enable interrupt for corresponding channel if required.
6. Configure the DMA_CFGREG register to enable DMA.
7. Configure the DMA_CHENREG register to enable the channel.
8. Wait for the interrupt or query DMA_CHENREG.
9. Clear the interrupt.

11.5.2 Software Flow Control

1. Configure the DMA_SARx register to specify the source address.

2. Configure the DMA_DARx register to specify the destination address.
3. Configure the DMA_CTLx register:
   A. to enable interrupt if required.
   B. to set the transfer data width and size.
   C. to select the flow control type.
   D. to select whether the address is incremented or decremented.
   E. to select the burst transfer length.
4. Configure the DMA_CFGx register:
   A. to select the channel priority.
   B. to set the HS_SEL field to 1.
5. Configure the DMA_MASKBLOCK register to enable interrupt for corresponding channel if required.
6. Configure the DMA_CFGREG register to enable DMA.
7. Configure the DMA_CHENREG register to enable the channel.
8. Software writes to the software handshake register to trigger the transmission.
9. Wait for the interrupt or query DMA_CHENREG.
10. Clear the interrupt.

12 DMA Request Multiplexer (DMAMUX)

12.1 Overview

The chip features an integrated DMA request multiplexer (DMAMUX) module, which can freely map to the 8 handshake signals of the DMA controller, and match them correspondingly to the 8 channels of the DMA controller. The controller is capable of generating random numbers.

12.2 Main Features

- Up to 8-channel programmable DMA request line multiplexer output
- Up to 4-channel DMA request generator
- Up to 8 trigger inputs to DMA request generator
- Up to 16 synchronization inputs
- Each DMA request generator channel has:
  - DMA request trigger input selector
  - DMA request counter
  - Event overrun flag for selected DMA request trigger input
- Each DMA request line multiplexer channel output has:
  - Up to 81 input DMA request lines from peripherals
  - One DMA request line output
  - Synchronization input selector
  - DMA request counter
  - Event overrun flag for selected synchronization input
  - One event output, for DMA request chaining

12.3 Functional Description

12.3.1 Handshake Signal Selection

DMAMUX provides 8 handshake signal output channels, and one of the numerous input signals can be selected in the CxCR register and output to the corresponding handshake signal output channel.

12.3.2 Trigger Generation of Handshake Signal

DMAMUX supports checking trigger signals and generating handshake signals, providing a total of 4 channels. The DMAMUX req genX handshake signal can be generated by selecting a trigger signal in the RGxCR register and setting the number of DMA handshake requests generated per trigger.

12.3.3 Synchronizing Handshake Signal with Trigger Signal

The handshake signal synchronization function can be enabled in the CxCR register, allowing the setting of the active level of trigger signal and the number of DMA handshakes allowed per trigger. When the trigger signal is not present, DMA handshake requests will be suspended until the trigger signal appears, after which the specified number of handshake signals will be allowed to be transmitted to the DMA controller.

12.3.4 Trigger Generation of DMA Transfer Event

The event generation function can be enabled in the CxCR register; each time a specified number of DMA handshakes occurs, a DMA transfer event will be generated, which can be used as a trigger signal.

12.3.5 Handshake Signal Source and Trigger Signal Source

Table 12-1: Handshake Signal Source and Trigger Signal Source

No.	DMA Request Source	Trigger Counter Source
0	DMAMUX_req_gen0	EXTI0
1	DMAMUX_req_gen1	EXTI1
2	DMAMUX_req_gen2	EXTI2
3	DMAMUX_req_gen3	EXTI3
4	SPI0 TX	EXTI4
5	SPI0 RX	EXTI5
6	SPI1 TX	EXTI6
7	SPI1 RX	EXTI7
8	SPI2 TX	EXTI8
9	SPI2 RX	EXTI9
10	UART1 TX	EXTI10
11	UART1 RX	EXTI11
12	DAC	EXTI12
13	ADC0	EXTI13
14	ADC1	EXTI14
15	TIM0_CH1	EXTI15
16	TIM0_CH2	DMAMUX_event_0
17	TIM0_CH3	DMAMUX_event_1
18	TIM0_CH4	DMAMUX_event_2
19	TIM0_TRIG/TIM0_COM	DMAMUX_event_3
20	TIM0_UP	LPTimer0_out
21	TIM7_CH1	LPTimer1_out
22	TIM7_CH2	TIM5_TRGO
23	TIM7_CH3	TIM14_out
24	TIM7_CH4	TIM15_out
25	TIM7_TRIG/TIM7_COM	TIM16_out
26	TIM7_UP	-
27	TIM1_CH1	-
28	TIM1_CH2	-
29	TIM1_CH3	-
30	TIM1_CH4	-
31	TIM1_TRIG	-
32	TIM1_UP	-

No.	DMA Request Source	Trigger Counter Source
33	TIM2_CH1	-
34	TIM2_CH2	-
35	TIM2_CH3	-
36	TIM2_CH4	-
37	TIM2_TRIG	-
38	TIM2_UP	-
39	TIM3_CH1	-
40	TIM3_CH2	-
41	TIM3_CH3	-
42	TIM3_CH4	-
43	TIM3_TRIG	-
44	TIM3_UP	-
45	TIM4_CH1	-
46	TIM4_CH2	-
47	TIM4_CH3	-
48	TIM4_CH4	-
49	TIM4_TRIG	-
50	TIM4_UP	-
51	TIM5_UP	-
52	USART6 TX	-
53	USART6 RX	-
54	USART7 TX	-
55	USART7 RX	-
56	I2C1 TX	-
57	I2C1 RX	-
58	I2C2 TX	-
59	I2C2 RX	-
60	TIM8_CH1	-
61	TIM8_CH2	-
62	TIM8_CH3	-
63	TIM8_CH4	-
64	TIM8_TRIG	-
65	TIM8_UP	-
66	TIM9_CH1	-
67	TIM9_CH2	-
68	TIM9_CH3	-

No.	DMA Request Source	Trigger Counter Source
69	TIM9_CH4	-
70	TIM9_TRIG	-
71	TIM9_UP	-
72	TIM14	-
73	TIM15	-
74	TIM16	-
75	Cordic in	-
76	Cordic out	-
77	I2S TX	-
78	I2S RX	-
79	CANFD_TX	-
80	CANFD_RX	-

12.4 Register Description

Register base address: 0x40B0_2100

The registers are listed below:

Table 12-2: List of DMAMUX Registers

Offset Address	Name	Description
0x00	DMAMUX_C0CR	Channel 0 control register
0x04	DMAMUX_C1CR	Channel 1 control register
0x08	DMAMUX_C2CR	Channel 2 control register
0x0C	DMAMUX_C3CR	Channel 3 control register
0x10	DMAMUX_C4CR	Channel 4 control register
0x14	DMAMUX_C5CR	Channel 5 control register
0x18	DMAMUX_C6CR	Channel 6 control register
0x1C	DMAMUX_C7CR	Channel 7 control register
0x80	DMAMUX_CSR	Channel interrupt status register
0x84	DMAMUX_CFR	Channel interrupt clear register
0x100	DMAMUX_RG0CR	DMA request generation channel 0 control register
0x104	DMAMUX_RG1CR	DMA request generation channel 1 control register
0x108	DMAMUX_RG2CR	DMA request generation channel 2 control register
0x10C	DMAMUX_RG3CR	DMA request generation channel 3 control register
0x140	DMAMUX_RGSR	DMA request generation interrupt status register
0x144	DMAMUX_RGCFR	DMA request generation interrupt clear register

12.4.1 Channel x Control Register (DMAMUX_CxCR)

Offset address: 0x00/0x04/0x08/0x0C/0x10/0x14/0x18/0x1C, x = 0−7

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:29	RSV	-	-	Reserved
28:24	SYNC_ID	R/W	0	Select the trigger signal for synchronization
23:19	NBREQ	R/W	0	Number of DMA requests allowed after each synchronization; or the number of requests required to generate a DMA transfer event
18:17	SPOL	R/W	0	Define the edge polarity of the selected synchronization input: 0x0: synchronization disabled 0x1: rising-edge trigger 0x2: falling-edge trigger 0x3: rising-edge and falling-edge trigger
16	SE	R/W	0	Synchronization enable: 0: disabled 1: enabled
15:10	RSV	-	-	Reserved
9	EGE	R/W	0	Transfer event generation enable: 0: disabled 1: enabled
8	SOIE	R/W	0	Synchronization overrun interrupt enable: 0: disabled 1: enabled
7	RSV	-	-	Reserved
6:0	DMAREQ_ID	R/W	0	Select the input DMA request

12.4.2 Channel Interrupt Status Register (DMAMUX_CSR)

Offset address: 0x80

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:8	RSV	-	-	Reserved

Bit	Name	Attribute	Reset Value	Description
7:0	SOF	R	0	Each bit corresponds to the synchronization overrun interrupt state of a channel, which is set when the DMA transfer caused by the previous synchronization signal is not fully completed and a new synchronization signal is present.

12.4.3 Channel Interrupt Clear Register (DMAMUX_CFR)

Offset address: 0x084

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:8	RSV	-	-	Reserved
7:0	CSOF	W	0	Writing 1 in each bit clears the corresponding synchronization overrun interrupt of a channel.

12.4.4 DMA Request Generator Channel x Configuration Register (DMAMUX_RGxCR)

Offset address: 0x100/0x104/0x108/0x10C, x = 0–3

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:24	RSV	-	-	Reserved
23:19	GNBREQ	R/W	0	Defines the number of DMA requests to be generated after a trigger event. The actual number of generated DMA requests is GNBREQ + 1.
18:17	GPOL	R/W	0	Defines the edge polarity of the selected trigger input: 0x0: no event, i.e. none trigger detection nor generation 0x1: rising-edge trigger 0x2: falling-edge trigger 0x3: rising-edge and falling-edge trigger
16	GE	R/W	0	DMA request generator channel x enable:

Bit	Name	Attribute	Reset Value	Description
				0: disabled 1: enabled
15:9	RSV	-	-	Reserved
8	OIE	R/W	0	Trigger overrun interrupt enable: 0: disabled 1: enabled
7:5	RSV	-	-	Reserved
4:0	SIG_ID	R/W	0	Trigger signal source

12.4.5 DMA Request Generator Interrupt Status Register (DMAMUX_RGSR)

Offset address: 0x140

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:4	RSV	-	-	Reserved
3:0	OF	R	0	Each bit corresponds to the trigger overrun interrupt state of a channel, which is set when the DMA transfer caused by the previous trigger signal is not fully completed and a new trigger signal is present.

12.4.6 DMA Request Generator Interrupt Clear Flag Register (DMAMUX_RGCFR)

Offset address: 0x144

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:4	RSV	-	-	Reserved
3:0	COF	W	0	Writing 1 in each bit clears the corresponding trigger overrun interrupt of a channel.

12.5 Operation Procedure

12.5.1 Selecting Handshake Signal Source for Output to DMA Controller

Just select a signal source in CxCR[5:0].

12.5.2 Trigger Generation of Handshake Signal

1. Configure RGxCR to select the trigger signal input source, polarity, and number of trigger generation requests.
2. Configure CxCR to output the req_gen signal to the DMA controller.
3. Configure RGxCR to activate the trigger.

12.5.3 Trigger Signal for Synchronizing Handshake Signal

1. Configure CxCR to select the trigger signal input source, polarity, number of synchronization times, and handshake signal source for the synchronization function.
2. Configure CxCR to activate the synchronization.

12.5.4 DMA Transfer Event

1. Configure CxCR to select the number of transfers required to generate the event.
2. Configure RGxCR to use DMA event as the trigger source.
3. Configure CxCR to activate the transfer event generation.

13 Hardware Acceleration Co-processor (CORDIC)

13.1 Overview

The CORDIC co-processor provides hardware acceleration of certain mathematical functions such as $m \cdot \sin\theta$, $m \cdot \cos\theta$, atan2(y, x), $\sqrt{x^2+y^2}$, $y \cdot x$, y/x, $\sinh w$, $\cosh w$, $tanh^{-1}(y/x)$, ln(x), $\sqrt{x}$, etc.

13.2 Main Features

- 24-bit CORDIC rotation engine
- AHB interface supporting 16-bit and 32-bit fixed point input and output formats
- DMA available
- Configurable input/output data address

13.3 Functional Description

13.3.1 Data Format and Input/Output

This module adopts 24-bit data internally for operation, and the data input and output can be selected in either 32-bit or 16-bit format for data transfer. The data format range in this section applies to all function modes.

When the 32-bit format is selected, the higher 24 bits are the input and output data, and the lower 8 bits are meaningless. If signed, the signed 32-bit format can be used for operation in the system. The decimal point of the data is located between bit [23] and bit [22]. If the data is signed, the whole data is represented in the range of [-1, (2^{23} - 1) / 2^{23}], i.e. [0x80000000, 0x7FFFFFFF]. If the data is not signed, it is represented in the range of [0, (2^{24} - 1) / 2^{23}], i.e.

[0X0, 0xFFFFFF00]. The following is an example of data conversion [applicable for both positive and negative numbers]:

Input Data x [DEC]	Calculation Formula	Converted Data [DEC]	Converted Data [HEX]
0.25	$x * 2^{31}$	536870912	0x20000000

When the 16-bit format is selected, the 16-bit data is the higher 16 bits of the internal data, and the lower 8 bits of the input data are filled with 0. If the data is signed, the signed 16-bit format can be used for operation in the system, and the whole data is represented in the range of [-1, (215 - 1) / 215], i.e. [0x8000, 0x7FFF]. If the data is not signed, it is represented in the range of [0, (216 - 1) / 215], i.e. [0x0, 0xFFFF].

Input Data x [DEC]	Calculation Formula	Converted Data [DEC]	Converted Data [HEX]
0.25	$x * 2^{15}$	8192	0x00002000

If two data inputs are required for an operation, the operation will start automatically when both data are written; if only one data is required, writing one data will automatically starts the operation.

13.3.2 Function Mode 0: $m \cdot \sin\theta$ / $m \cdot \cos\theta$

When the function mode 0: $m \cdot \sin\theta$ / $m \cdot \cos\theta$ is selected, the input and output data are as follows:

Table 13-1: Input and Output Table of Cordic Controller Function Mode 0

	Input Data 1	Input Data 2	Output Data 1	Output Data 2
Content	θ	m	$m \cdot \sin\theta$	$m \cdot \cos\theta$
Sign bit	True	True	True	True
Remarks	The unit is $\pi\ rad$, $\theta \in [-\frac{1}{4}, \frac{1}{4}]$	1 can be represented by 0x7FFFFF (XX) (32-bit mode) or 0x7FFF (16-bit mode), where no multiplication is performed.	-	-

Relevant calculation: $\tan\theta = \sin\theta \div \cos\theta$

Notes:

1. The sign bit indicates whether the data is signed, and the MSB bit of the signed data represents the sign.

   True: signed; False: not signed.

2. For the data input range, please refer to Chapter 13.3.1 Data Format and Input/Output.

3. The above notes are applicable to all function modes.

13.3.3 Function Mode 1: atan2(y,x) / $\sqrt{(x^2 + y^2)}$

When the function mode 1: atan2(y, x) / $\sqrt{x^2 + y^2}$ is selected, the input and output data are as follows:

Table 13-2: Input and Output Table of Cordic Controller Function Mode 1

	Input Data 1	Input Data 2	Output Data 1	Output Data 2
Content	y	x	atan2(y, x)	$\sqrt{x^2 + y^2}$
Sign bit	True	True	True	False
Remarks	-	-	The unit is $\pi\ rad$.	Bit [23] indicates that the decimal point is preceded by 0 or 1, not the sign, and the range is [0, $\sqrt{2}$].

Relevant calculation: $\tan^{-1} x = \text{atan2}(x \cdot 2^{-n}, 2^{-n})$

13.3.4 Function Mode 2: $y \cdot x$

When the function mode 2: $y \cdot x$ is selected, the input and output data are as follows:

Table 13-3: Input and Output Table of Cordic Controller Function Mode 2

	Input Data 1	Input Data 2	Output Data 1	Output Data 2
Content	y	x	$y \cdot x$	-
Sign bit	True	True	True	-
Remarks	-	-	-	-

Note: Input calculation with y=-1 and x=-1 at the same time is not supported.

13.3.5 Function Mode 3: y/x

When the function mode 3: y/x is selected, the input and output data are as follows:

Table 13-4: Input and Output Table of Cordic Controller Function Mode 3

	Input Data 1	Input Data 2	Output Data 1	Output Data 2
Content	y	x	y/x	-
Sign bit	True	True	True	-
Remarks	It is required that $\|y\| \leq \|x\|$.		-	-

13.3.6 Function Mode 4: $\sinh w$ / $\cosh w$

When the function mode 4: $\sinh w$ / $\cosh w$ is selected, the input and output data are as follows:

Table 13-5: Input and Output Table of Cordic Controller Function Mode 4

	Input Data 1	Input Data 2	Output Data 1	Output Data 2
Content	$2^{-1} \cdot w$	-	$2^{-1} \cdot \sinh w$	$2^{-1} \cdot \cosh w$
Sign bit	True	-	True	False
Remarks	$w \in [-1.1181, 1.1181]$	-	Range in [0, 1.366]	Range in [0, 1.693]

Relevant calculation: $e^w = \sinh w + \cosh w$

13.3.7 Function Mode 5: $\tanh^{-1}(y/x)$

When the function mode 5: $\tanh^{-1}(y/x)$ is selected, the input and output data are as follows:

Table 13-6: Input and Output Table of Cordic Controller Function Mode 5

	Input Data 1	Input Data 2	Output Data 1	Output Data 2
Content	y	x	$2^{-1} \cdot \theta$	-
Sign bit	True	False	True	-
Remarks	It is able to take the point (y, x) from the ranges of $y \in [-1,1)$ and $x \in [0,2)$, or enter $y = 2^{-n} \cdot \tanh\theta$ and $x = 2^{-n}$, requiring $(y/x) \in [-0.8069, 0.8069]$.		-	-

13.3.8 Function Mode 6: ln(x)

When the function mode 6: ln(x) is selected, the input and output data format varies depending on the range of input data, requiring a numerical range SCALE in the control register, as follows:

When $x \in [0.1069, 1)$, make SCALE = 0.

Table 13-7: Input and Output Table 1 of Cordic Controller Function Mode 6

	Input Data 1	Input Data 2	Output Data 1	Output Data 2
Content	x	-	$2^{-2} \cdot \ln x$	-
Sign bit	False	-	True	-
Remarks	$x \in [0.1069, 1)$	-	-	-

When $x \in (1, 3)$, make SCALE = 1.

Table 13-8: Input and Output Table 2 of Cordic Controller Function Mode 6

	Input Data 1	Input Data 2	Output Data 1	Output Data 2
Content	$2^{-1} \cdot x$	--	$2^{-2} \cdot \ln x$	-
Sign bit	False	--	True	-
Remarks	$x \in (1, 3)$	-	-	-

When $x \in [3, 7)$, make SCALE = 2.

Table 13-9: Input and Output Table 3 of Cordic Controller Function Mode 6

	Input Data 1	Input Data 2	Output Data 1	Output Data 2
Content	$2^{-2} \cdot x$	-	$2^{-2} \cdot \ln x$	-
Sign bit	False	-	True	-
Remarks	$x \in [3,7)$	-	-	-

When $x \in [7, 9.35]$, make SCALE = 3.

Table 13-10: Input and Output Table 4 of Cordic Controller Function Mode 6

	Input Data 1	Input Data 2	Output Data 1	Output Data 2
Content	$2^{-3} \cdot x$	-	$2^{-2} \cdot \ln x$	-
Sign bit	False	-	True	-
Remarks	$x \in [7, 9.35]$	-	-	-

Relevant calculation: $w^t = e^{t \ln w}$

13.3.9 Function Mode 7: $\mathbf{Sqr(x)}$

When the function mode 7: $\sqrt{x}$, i.e. Sqr(x), is selected, the input and output data format varies depending on the range of input data, requiring a numerical range SCALE in the control register, as follows:

When $x \in [0.1069, 1)$, make SCALE = 0.

Table 13-11: Input and Output Table 1 of Cordic Controller Function Mode 7

	Input Data 1	Input Data 2	Output Data 1	Output Data 2
Content	x	-	$\sqrt{x}$	-
Sign bit	False	-	False	-
Remarks	$x \in [0.1069, 1)$	-	-	-

When $x \in (1, 3)$, make SCALE = 1.

Table 13-12: Input and Output Table 2 of Cordic Controller Function Mode 7

	Input Data 1	Input Data 2	Output Data 1	Output Data 2
Content	$2^{-1} \cdot x$	-	$2^{-1} \cdot \sqrt{x}$	-
Sign bit	False	-	False	-
Remarks	$x \in (1, 3)$	-	-	-

When $x \in [3, 7)$, make SCALE = 2.

Table 13-13: Input and Output Table 3 of Cordic Controller Function Mode 7

	Input Data 1	Input Data 2	Output Data 1	Output Data 2
Content	$2^{-2} \cdot x$	-	$2^{-2} \cdot \sqrt{x}$	-
Sign bit	False	-	False	-
Remarks	$x \in [3,7)$	-	-	-

When $x \in [7, 9.35]$, make SCALE = 3.

Table 13-14: Input and Output Table 4 of Cordic Controller Function Mode 7

	Input Data 1	Input Data 2	Output Data 1	Output Data 2
Content	$2^{-3} \cdot x$	-	$2^{-3} \cdot \sqrt{x}$	-
Sign bit	False	-	False	-
Remarks	$x \in [7, 9.35]$	-	-	-

13.4 Register Description

Register base address: 0x4080_0000

The registers are listed below:

Table 13-15: List of Cordic Controller Registers

Offset Address	Name	Description
0x00	CORDIC_CTRL	Control register
0x04	CORDIC_DIN1	Input register 1
0x08	CORDIC_DIN2	Input register 2
0x0C	CORDIC_DOUT1	Output register 1
0x10	CORDIC_DOUT2	Output register 2

Registers are detailed in the following sections.

13.4.1 Control Register (CORDIC_CTRL)

Offset address: 0x00

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31	DATA_READY	R	0x0	0: no currently-available operation result 1: with currently-available operation result
30:22	RSV	-	-	Reserved
21:20	SCALE	R/W	0x0	Scaling factor of argument calculated via $\ln(x)$ and $\sqrt{x}$ is determined by the value of x.
19:18	RSV	-	-	Reserved
17	DMA_OUT	R/W	0x0	Output data DMA enable: 0: output data DMA disabled 1: output data DMA enabled
16	DMA_IN	R/W	0x0	Input data DMA enable: 0: input data DMA disabled 1: input data DMA enabled
15:14	RSV	-	-	Reserved
13	MERGE_OUT	R/W	0x0	Output data merge, available only when the 16-bit bus mode is used for data output: 0: Data is output in two parts.

Bit	Name	Attribute	Reset Value	Description
				1: The lower 16 bits of the data output register 1 is read as the first argument, and the higher 16 bits is read as the second argument.
12	MERGE_IN	R/W	0x0	Input data merge, available only when the 16-bit bus mode is used for data input: 0: Data is input in two parts. 1: The lower 16 bits of the data input register 1 is written with the first argument, and the higher 16 bits is written with the second argument.
11	ADDR_OUT	R/W	0x0	Output address mode: 0: Read data from two data output registers respectively. 1: Both arguments can be read sequentially from data output register 1.
10	ADDR_IN	R/W	0x0	Input address mode: 0: Two arguments shall be written into two data input registers respectively. 1: Both arguments can be written sequentially into data input register 1.
9	WIDTH_OUT	R/W	0x0	Data output transfer mode: 0: 32-bit bus transfer mode adopted for data output 1: 16-bit bus transfer mode adopted for data output
8	WIDTH_IN	R/W	0x0	Data input transfer mode: 0: 32-bit bus transfer mode adopted for data input 1: 16-bit bus transfer mode adopted for data input
7:4	ITERATION	R/W	12	(Number of iterations in operation / 2) The more number of iterations, the more accurate it is and the longer it takes. The available range is [4, 12].
3	RSV	-	-	Reserved

Bit	Name	Attribute	Reset Value	Description
2:0	MODE	R/W	0x0	Function modes: 0: mode 0 selected: $m \cdot \sin\theta$ / $m \cdot \cos\theta$ 1: mode 1 selected: atan2(y,x) / $\sqrt{x^2+y^2}$ 2: mode 2 selected: $y \cdot x$ 3: mode 3 selected: y/x 4: mode 4 selected: $\sinh w$ / $\cosh w$ 5: mode 5 selected: $\tanh^{-1}(y/x)$ 6: mode 6 selected: ln(x) 7: mode 7 selected: $\sqrt{x}$

13.4.2 Data Input Register 1 (CORDIC_DIN1)

Offset address: 0x04

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:0	DIN1	W	0x0	Write input data to this register.

13.4.3 Data Input Register 2 (CORDIC_DIN2)

Offset address: 0x08

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:0	DIN2	W	0x0	Write input data to this register.

13.4.4 Data Output Register 1 (CORDIC_DOUT1)

Offset address: 0x0C

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:0	DOUT1	R	0x0	Read the operation result from this register.

13.4.5 Data Output Register 2 (CORDIC_DOUT2)

Offset address: 0x10

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:0	DOUT2	R	0x0	Read the operation result from this register.

13.5 Operation Procedure

1. Enable the CORDIC module clock and release the reset.
2. Set the input/output address format, data format, etc.
3. Set the number of operations (the higher the number, the more accurate the result).
4. Set the operation mode and the input/output data transfer mode.
5. Input the appropriate data to CORDIC_DIN1 and CORDIC_DIN2 according to the set operation mode.
6. Wait for CORDIC_CTRL[31] to be set.
7. Read the two output data from CORDIC_DOUT1 and CORDIC_DOUT2.

13.6 Calculation Example

The relevant calculation background in this section is the default configuration of CORDIC_CTRL: [input/output calculation in other modes can refer to this mode]

Default configuration: 32-bit input/output data format

Data input and output: input to two input registers

Data transfer mode: CPU

Calculation function mode: function mode 0

Calculation method for converting input data 1 into the hardware-specified format:

$$0.25\ [DEC] = 0.25 * 2^{31}\ [DEC]$$

$$= 536,870,912\ [DEC]$$

= 0x20000000 [HEX]

Calculation method for converting input data 2 into the hardware-specified format:

0.00390625 [DEC] = 0.00390625 * 2^{31} [DEC]

= 8388608 [DEC]

= 0x00800000 [HEX]

Input the above calculation results into CORDIC_DIN1 and CORDIC_DIN2 respectively, then the hardware can start the calculation, the results of which can be obtained from CORDIC_DOUT1 and CORDIC_DOUT2.

Verify the calculation results:

Theoretical calculation result of output data 1:

m * sinθ = 0.00390625 * (sin (0.25 * π)) [DEC]

= 0.00390625 * 0.7071067811865 [DEC]

= 0.00276213586400995 12671907982 [DEC] * 2^{31}

= 5931641.601515722055569 [DEC]

= 5A8279 [HEX]

Theoretical calculation result of output data 2:

m * cosθ = 0.00390625 * (cos (0.25 * π)) [DEC]

= 0.00390625 * 0.7071067811865 [DEC]

= 0.00276213586400995 [DEC]

= 5,931,641.601515722055569 [DEC] * 2^{31}

= 5A8279 [HEX]

The result of hardware automatic calculation is 0x5a8300 [error: 135, mainly from the decimal point].

14 Advanced Encryption and Decryption Algorithm Accelerator (AES)

14.1 Overview

AES algorithm is a block algorithm. Both the encryption algorithm and the key expansion algorithm adopt nonlinear iterative structure. The decryption algorithm has the same structure as the encryption algorithm except that the round keys are used in reverse order, that is, the decryption round key is the inverse of the encryption round key.

14.2 Main Features

- Cipher key lengths of 128, 192 or 256 bits
- AES encryption and decryption
- Electronic code book (ECB) and cipher block chaining (CBC) modes
- SWAP mode supported for data input and output, i.e., big-endian / little-endian format configurable
- Support hardware MASK during encryption and decryption

14.3 Functional Description

14.3.1 Encryption and Decryption

Load the key, and then run the encryption or decryption algorithm.

14.4 Register Description

Register base address: 0x4060_0000

The registers are listed below:

Table 14-1: List of AES Registers

Offset Address	Name	Description
0x00	AES_DATAIN	Data input register
0x04	AES_KEYIN	Key input register
0x08	AES_IVIN	Initial vector input register
0x0C	AES_CONTROL	Control register
0x10	AES_STATE	Status register
0x14	AES_DATAOUT	Data output register

Registers are detailed in the following sections.

14.4.1 Data Input Register (AES_DATAIN)

Offset address: 0x00

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:0	DATAIN	W	0x00000000	32-bit register for storing plaintext or ciphertext data, shall be written four times continuously to form a 128-bit data stream.

14.4.2 Key Input Register (AES_KEYIN)

Offset address: 0x04

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:0	KEYIN	W	0x00000000	32-bit register for storing key data, shall be written four times continuously to form a 128-bit key.

14.4.3 Initial Vector Input Register (AES_IVIN)

Offset address: 0x08

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:0	IVIN	W	0x00000000	32-bit register for storing the initial vector in CBC operation mode, shall be written four times continuously to form a 128-bit initial vector. This register is valid only in CBC mode, and does not work in ECB mode even if data is written.

14.4.4 Control Register (AES_CONTROL)

Offset address: 0x0C

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:12	RSV	-	-	Reserved
11:9	ALL_ROUND	R/W	0x0	Total number of rounds with pseudo-AES algorithm enabled: 0: 2 rounds 1: 4 rounds 2: 8 rounds 3: 16 rounds 4: 32 rounds 5: 64 rounds Others: reserved
8	VAES_EN	R/W	0x0	Pseudo-AES algorithm enable: 0: disabled 1: enabled
7:6	KEY_MODE	R/W	0x0	AES key length selection: 00: AES_128 01: AES_192 10: AES_256

Bit	Name	Attribute	Reset Value	Description
5	ECB	R/W	0x0	Mode indicator bit: 0: ECB mode 1: CBC mode
4	SWAP	R/W	0x0	SWAP mode enable for data input and output: 0: SWAP mode disabled 1: SWAP mode enabled
3	INT_EN	R/W	0x0	Interrupt enable bit: 0: disabled 1: enabled
2	CRYPT	R/W	0x0	Encryption/decryption indicator bit: 0: encryption performed 1: decryption performed
1	KEY_START	W	0x0	Key expansion algorithm start bit: 0: Writing 0 has no effect. Reading this bit always returns 0. 1: Writing 1 initiates the key expansion algorithm. Reading this bit always returns 0.
0	CRYPT_START	W	0x0	Encryption/decryption start bit: 0: Writing 0 has no effect. Reading this bit always returns 0. 1: Writing 1 initiates the encryption / decryption. Reading this bit always returns 0.

14.4.5 Status Register (AES_STATE)

Offset address: 0x10

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:3	RSV	-	-	Reserved
2	CBCDONE	R	0x0	CBC of block 1 done bit: 0: CBC of block 1 undone. This bit is valid only in CBC mode, and does not work in ECB mode. 1: CBC of block 1 done. Writing 1 clears this flag bit. When a batch of data is encrypted or

Bit	Name	Attribute	Reset Value	Description
				decrypted by CBC, this bit must be cleared by writing 1 to ensure that the next batch of data is encrypted or decrypted correctly. This bit is valid only in CBC mode, and does not work in ECB mode.
1	KEY_DONE	R	0x0	AES key expansion done bit: 0: AES key expansion undone 1: AES key expansion done. Writing 1 clears this flag bit.
0	CRYPT_DONE	R	0x0	AES algorithm done bit: 0: AES algorithm undone 1: AES algorithm done. Writing 1 clears this flag bit. An interrupt (if enabled) will be generated at the same time as this bit is set; writing 1 to this bit clears the interrupt.

14.4.6 Data Output Register (AES_DATAOUT)

Offset address: 0x14

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:0	DATAOUT	R	0x00000000	The data output register is used for storing the result of encryption and decryption, which is read-only.

14.5 Operation Procedure

The encryption and decryption process of AES module is shown below:

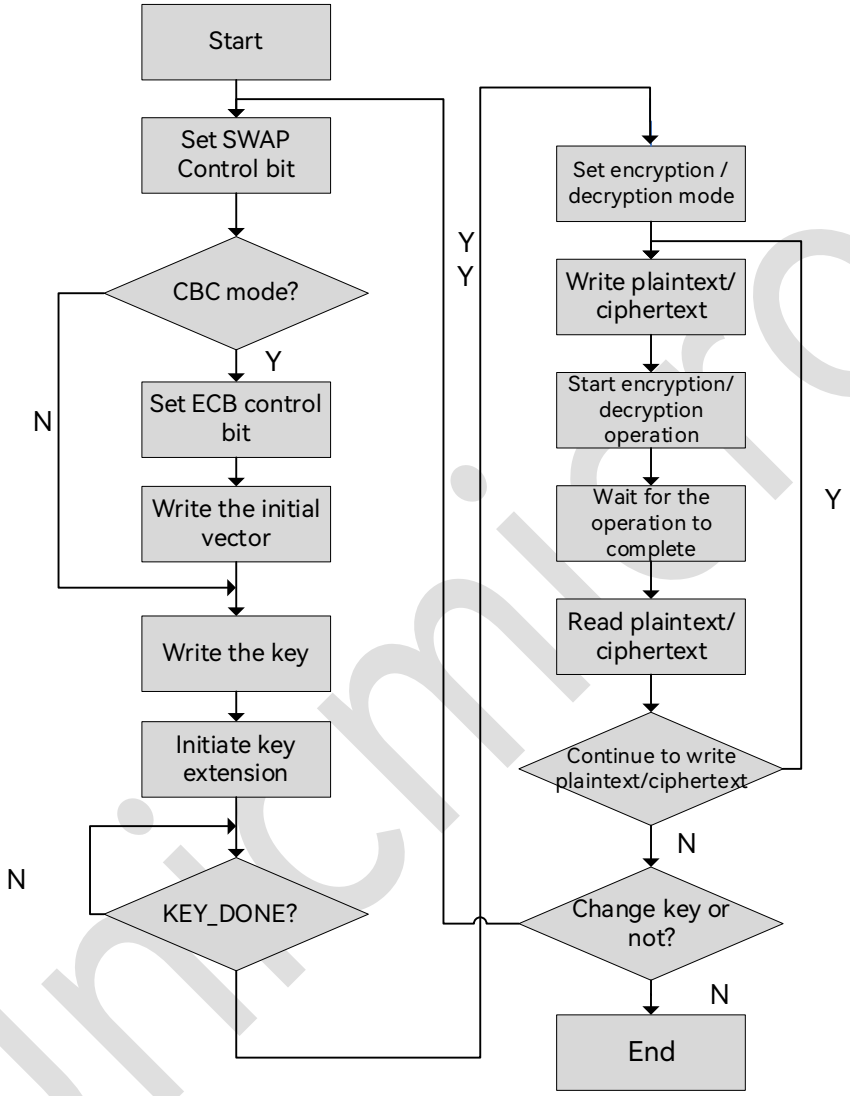


Figure 14-1: AES Encryption and Decryption Flowchart

15 True Random Number Generator (RNG)

15.1 Overview

The RNG is capable of delivering true random numbers.

15.2 Main Features

The RNG delivers true random numbers from random seeds.

15.3 Functional Description

15.3.1 RNG

RNG is capable of delivering true random numbers from random seeds.

15.4 Register Description

Register base address: 0x40B0_B000

The registers are listed below:

Table 15-1: List of RNG Registers

Offset Address	Name	Description
0x00	RNG_DATA	Data register
0x04	RNG_SEED	Random seed register
0x08	RNG_CR	Control register

15.4.1 Data Register (RNG_DATA)

Offset address: 0x00

Reset value: 0xXXXX XXXX (the data read is random)

Bit	Name	Attribute	Reset Value	Description
31:0	DATA	R/W	0xXXXX XXXX	Read data to obtain random number

15.4.2 Random Seed Register (RNG_SEED)

Offset address: 0x04

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:0	SEED	W	0x0	Reserved

15.4.3 Control Register (RNG_CR)

Offset address: 0x08

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:1	RSV	-	-	Reserved
0	EN	R/W	0x0	Enable random seed to automatically change with time: 1: enabled 0: disabled

15.5 Operation Procedure

1. Configure RNG_CR[0] to enable the random number generator.
2. Configure RNG_SEED[31: 0] to write a random seed.
3. Read random numbers from RNG_DATA[31:0], which can be read continuously.

16 Advanced Control Timer (TIM0 & TIM7)

16.1 Overview

The advanced-control timer consists of a 16-bit auto-reload counter driven by a programmable prescaler.

It may be used for a variety of purposes, including measuring the pulse lengths of input signals (input capture) or generating output waveforms (output compare, complementary PWM with dead-time insertion).

16.2 Main Features

- 16-bit up, down, up/down auto-reload counter
- 16-bit programmable prescaler allowing real-time adjustment of the counter clock division
- 4 independent channels for input capture, output compare, PWM generation and one-pulse output
- Complementary output with programmable dead-time insertion
- Support cascading with other timers
- Repetition counter to update the timer registers only after a given number of cycles of the counter
- Break inputs, break signal filtering and polarity selection, combinatorial configuration of break signals
- Interrupt or DMA event can be generated in the following cases:
  - Counter up/down overflow, counter initialization (triggered by software or hardware)

- Trigger event (counter start, stop, initialization, internal/external trigger)
- Input capture
- Output compare
- Break input

- Support incremental quadrature encoder and Hall sensor
- Trigger input for external clock

16.3 System Block Diagram

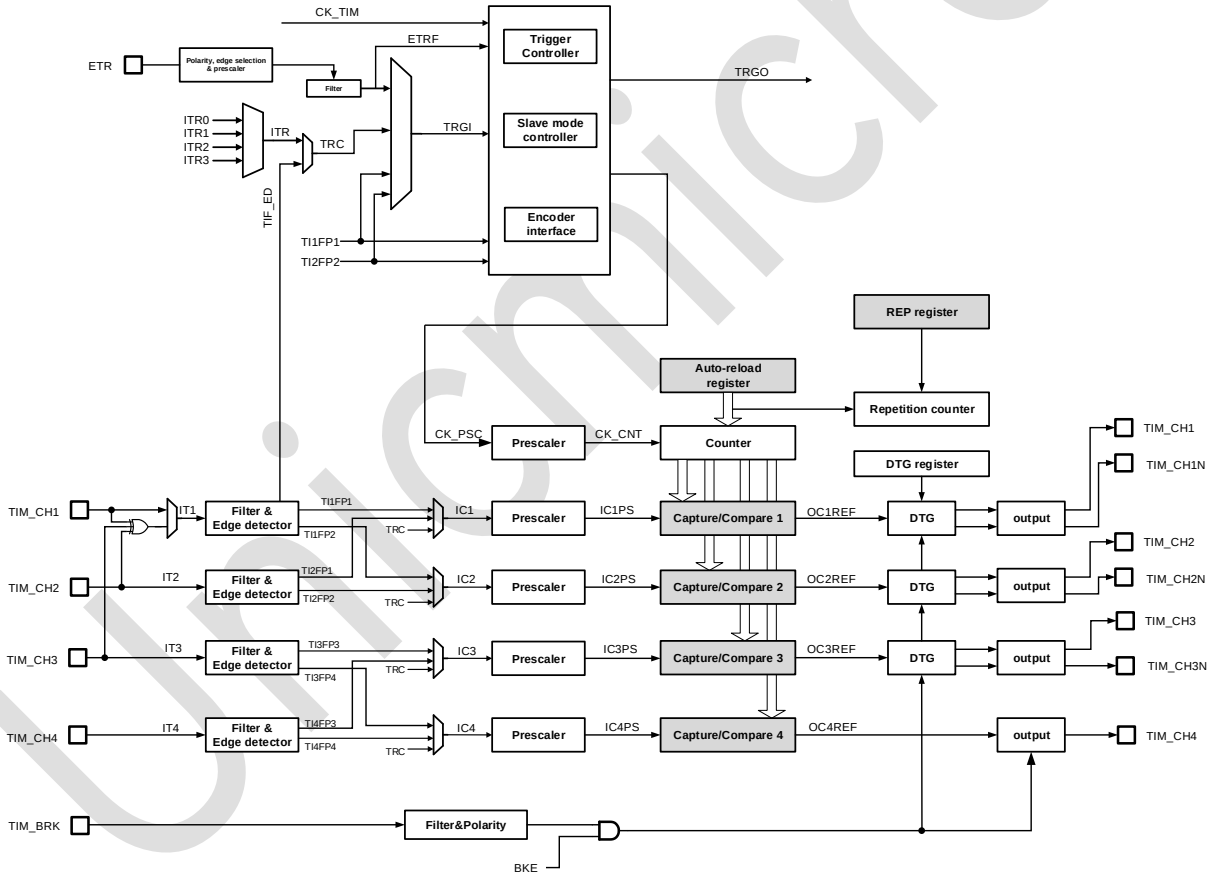


Figure 16-1: TIM0 & TIM7 System Block Diagram

16.4 Pin Description

Table 16-1: TIM0& TIM7 Pin Description

Function Pin	Alternate Function Pin	Direction	Functional Description
TIM0_BKIN	PA6, PB12, PA10, PB4	Input	Break input
TIM0_ETR	PA12, PA0, PD2	Input	External trigger input
TIM0_CH1	PA8, PA0, PB14, PA9, PB12	Input/output	Channel input capture / PWM output signal
TIM0_CH1N	PA7, PB13, PA1	Output	Reverse PWM output
TIM0_CH2	PA9, PA8	Input/output	Channel input capture / PWM output signal
TIM0_CH2N	PB0, PB14, PA6, PB6, PB15	Output	Reverse PWM output
TIM0_CH3	PA10, PA4, PB11	Input/output	Channel input capture / PWM output signal
TIM0_CH3N	PB1, PB15, PA9	Output	Reverse PWM output
TIM0_CH4	PA11, PA5, PA7, PB9, PB10	Input/output	Channel input capture / PWM output signal
TIM7_BKIN	PA6, PA2, PA9, PB5	Input	Break input
TIM7_ETR	PA0, PA15, PB7	Input	External trigger input
TIM7_CH1	PC6, PA2, PB14, PA10	Input/output	Channel input capture / PWM output signal
TIM7_CH1N	PA5, PA7, PA15, PB15, PC10	Output	Reverse PWM output
TIM7_CH2	PC7, PA3, PB15	Input/output	Channel input capture / PWM output signal
TIM7_CH2N	PB0, PB14, PC11	Output	Reverse PWM output
TIM7_CH3	PC8, PA4, PA6, PB4, PB8	Input/output	Channel input capture / PWM output signal
TIM7_CH3N	PB1, PB15, PB3, PB6, PC12	Output	Reverse PWM output
TIM7_CH4	PC9, PA5, PB5	Input/output	Channel input capture / PWM output signal

16.5 Timer Interconnection

Table 16-2: Timer Interconnection

TIMx Interconnection								
Source (TIMx Trigger Output)	**Inputs (TIMx Trigger Input)**							
TIM0_TRGO	-	TIM1_ITR0	TIM2_ITR0	TIM3_ITR0	-	TIM7_ITR0	-	-
TIM1_TRGO	TIM0_ITR1	-	TIM2_ITR1	TIM3_ITR1	TIM4_ITR0	TIM7_ITR1	TIM8_ITR2	TIM9_ITR0
TIM2_TRGO	TIM0_ITR2	TIM1_ITR2	-	TIM3_ITR2	TIM4_ITR1	-	-	TIM9_ITR1
TIM3_TRGO	TIM0_ITR3	TIM1_ITR3	TIM2_ITR3	-	TIM4_ITR2	TIM7_ITR2	TIM8_ITR0	TIM9_ITR2
TIM4_TRGO	TIM0_ITR0	-	TIM2_ITR2	-	-	TIM7_ITR3	TIM8_ITR1	-
TIM7_TRGO	-	TIM1_ITR1	-	TIM3_ITR3	TIM4_ITR3	-	-	-
TIM8_TRGO	-	-	-	-	-	-	-	TIM9_ITR3
TIM9_TRGO	-	-	-	-	-	-	TIM8_ITR3	-

16.6 Functional Description

16.6.1 Time-base Unit

The main block of the time-base unit is a 16-bit counter with its related auto-reload register. The counter can count up, down, or both up and down. The counter clock can be divided by a 16-bit prescaler.

The counter, the auto-reload register and the prescaler register can be written or read by software, which is true even when the counter is running.

The time-base unit includes:

- Counter register (TIM_CNT)
- Prescaler register (TIM_PSC)
- Auto-reload register (TIM_ARR)
- Repetition counter register (TIM_RCR)

The auto-reload register is preloaded, which is controlled by the auto-reload preload enable (ARPE) bit in the register. When ARPE = 0, write to the ARR register, and the written data is directly transferred to the shadow register. When ARPE = 1, the data written to the TIM_ARR

register is transferred to the shadow register when an update event (TIM_CNT overflow or underflow) occurs. The update event of ARR can also be actively triggered by software via register operation.

The counter TIM_CNT is clocked by the prescaler output TIM_PSC, which is enabled only when the counter enable bit (CEN) in the register is set. When CNT = ARR, this round of counting is over and the update event is sent.

TIM_PSC is a synchronous prescaler that can divide the counter clock frequency by any factor between 1 and 65536. The PSC register is also buffered, and overwriting PSC does not actually overwrite the shadow register unless a new update event occurs. Thus the PSC register can be changed in real time on the fly, and the new prescaler ratio is taken into account at the next update event.

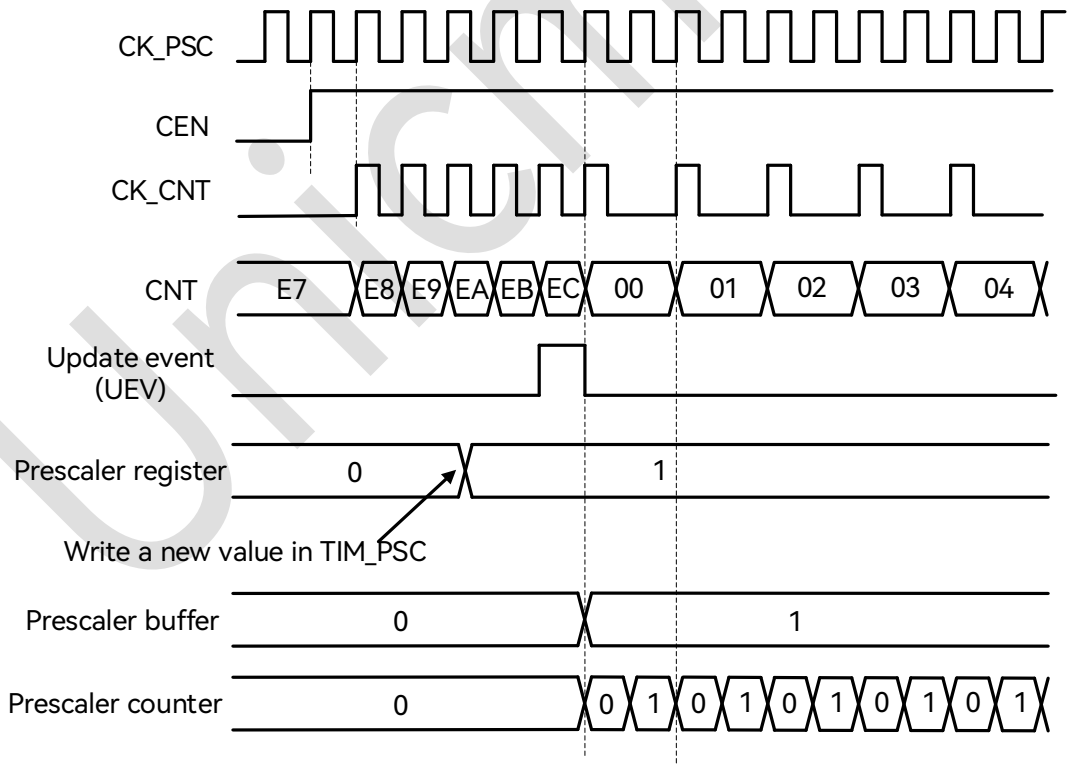


Figure 16-2: Counter Timing Diagram with Prescaler Division Changing from 1 to 2

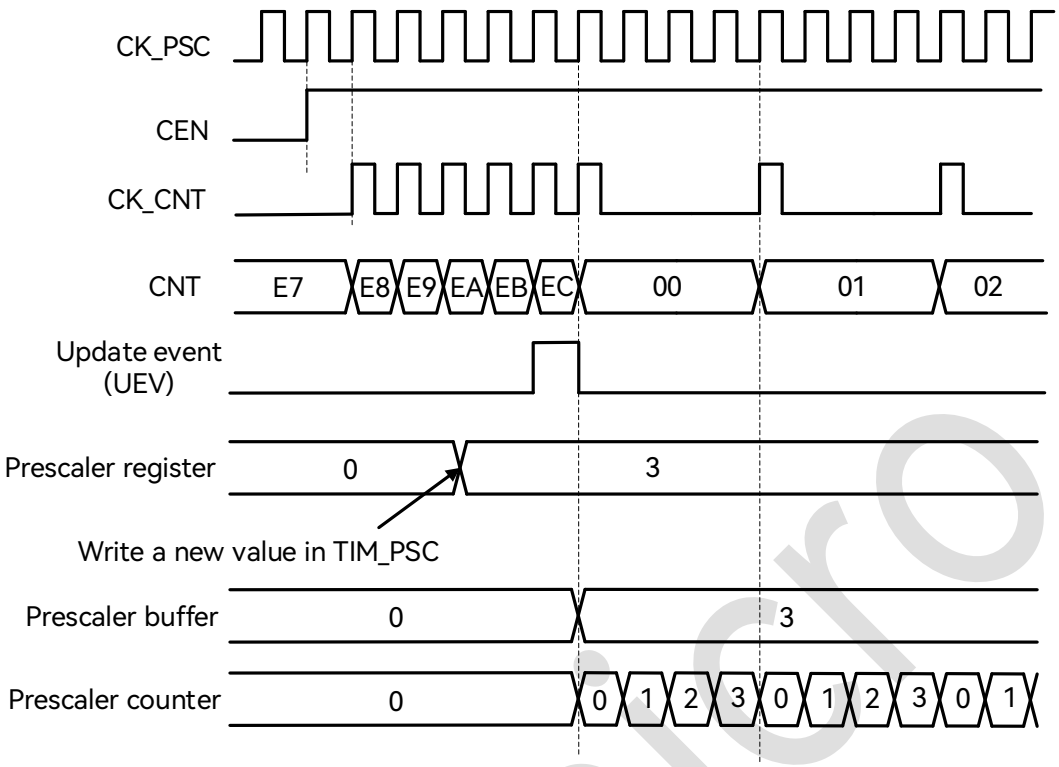


Figure 16-3: Counter Timing Diagram with Prescaler Division Changing from 1 to 4

16.6.2 Counter Operation Mode

The counter supports up-counting mode, down-counting mode and center-aligned mode.

16.6.2.1 Up-counting Mode

In up-counting mode, the counter counts from 0 to the auto-reload value, i.e. CNT = ARR, generating an overflow event, and then restarts counting from 0.

If the repetition counter is enabled, the counter repeats the above process a number of times (RCR + 1) as defined in RCR before generating an underflow event.

The software can directly trigger an update event by setting the UG bit in the register, at which time the CNT and the prescaler registers are automatically cleared. Whether setting the UG register triggers UIF (update interrupt flag) is determined by the setting of the URS register.

The update event can be disabled by setting the UDIS bit in the register to avoid updating the shadow register while writing new values in the preload registers.

When an update event occurs, the following registers are updated and the UIF bit is set:

- The repetition counter register RCR is reloaded with the content of TIM_RCR register.
- The auto-reload shadow register ARR is reloaded with the content of TIM_ARR register.
- The prescaler shadow register PSC is reloaded with the content of TIM_PSC register.

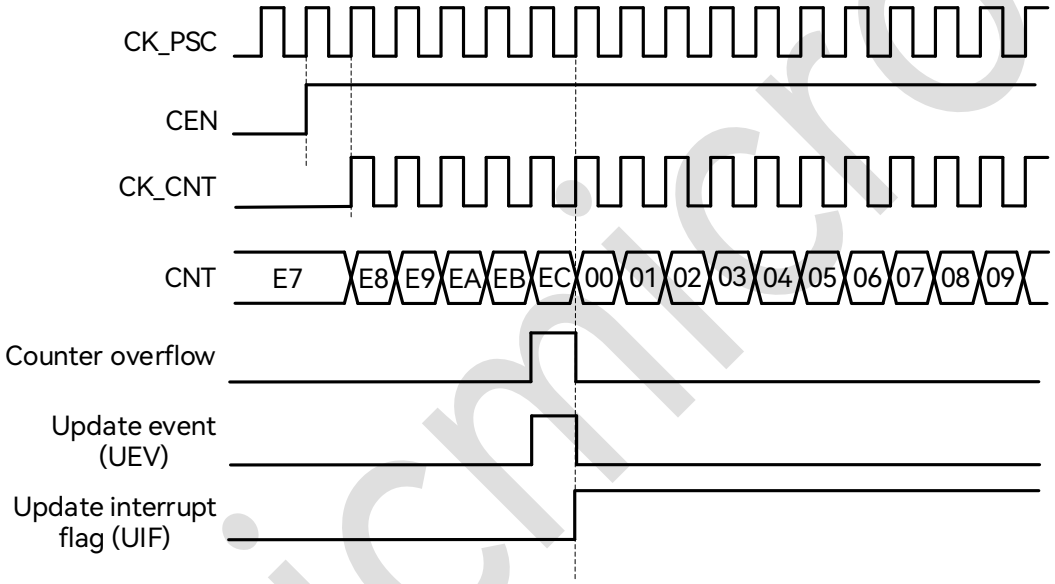


Figure 16-4: Up-counting Waveform Diagram, Internal Clock not Divided

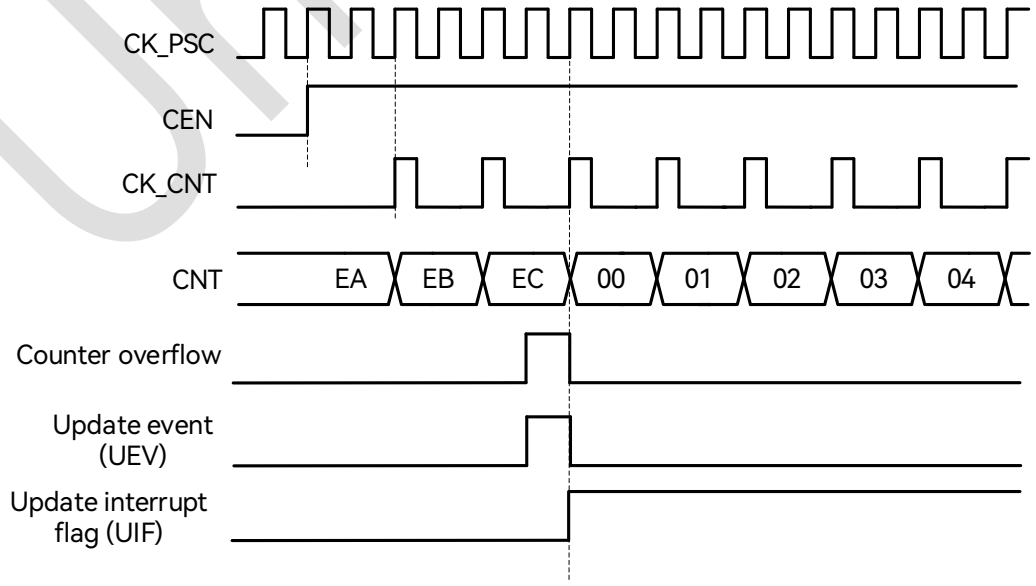


Figure 16-5: Up-counting Waveform Diagram, Internal Clock Divided by 2

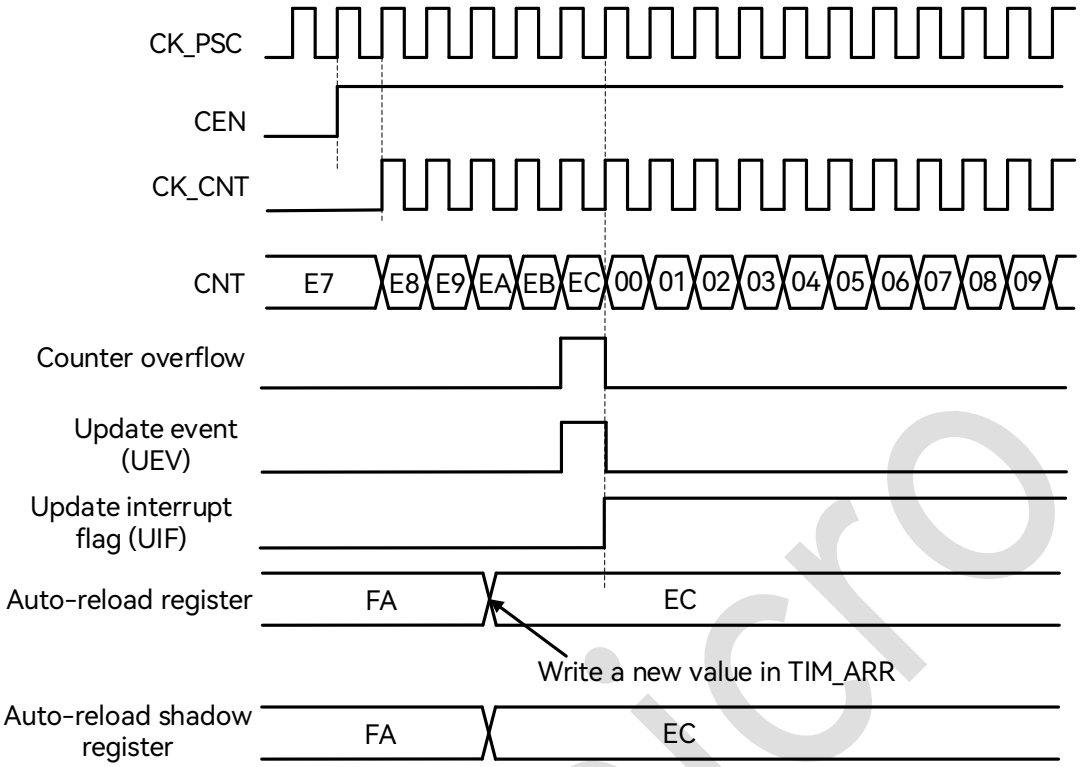


Figure 16-6: Counter Timing Diagram, Update Event when ARPE = 0 (TIM_ARR not Preloaded)

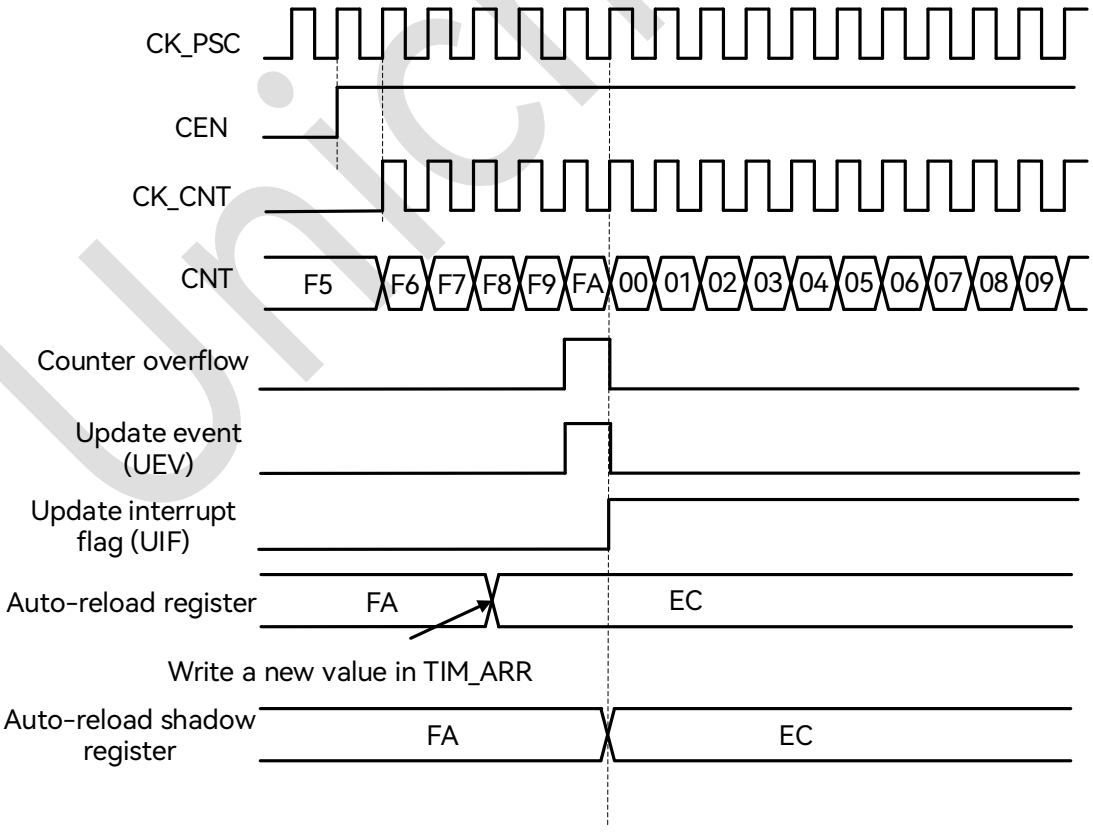


Figure 16-7: Counter Timing Diagram, Update Event when ARPE = 1 (TIM_ARR Preloaded)

16.6.2.2 Down-counting Mode

In down-counting mode, the counter counts from the auto-reload value down to 0, generating an underflow event, and then restarts counting from the auto-reload value.

If the repetition counter is enabled, the counter repeats the above process a number of times (RCR + 1) as defined in RCR before generating an underflow event.

The software can directly trigger an update event by setting the UG bit in the register, at which time the CNT and the prescaler registers are automatically cleared. Whether setting the UG register triggers UIF (update interrupt flag) is determined by the setting of the URS register.

The update event can be disabled by setting the UDIS bit in the register to avoid updating the shadow register while writing new values in the preload registers.

When an update event occurs, the following registers are updated and the UIF bit is set:

- The repetition counter register RCR is reloaded with the content of TIM_RCR register.
- The auto-reload shadow register ARR is reloaded with the content of TIM_ARR register.
- The prescaler shadow register PSC is reloaded with the content of TIM_PSC register.

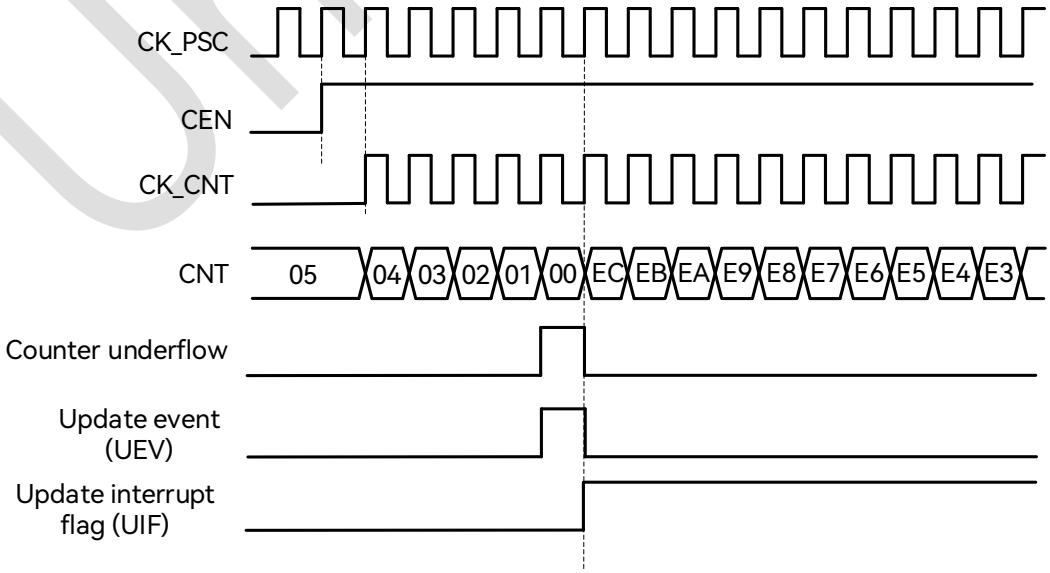


Figure 16-8: Down-counting Waveform Diagram, Internal Clock not Divided

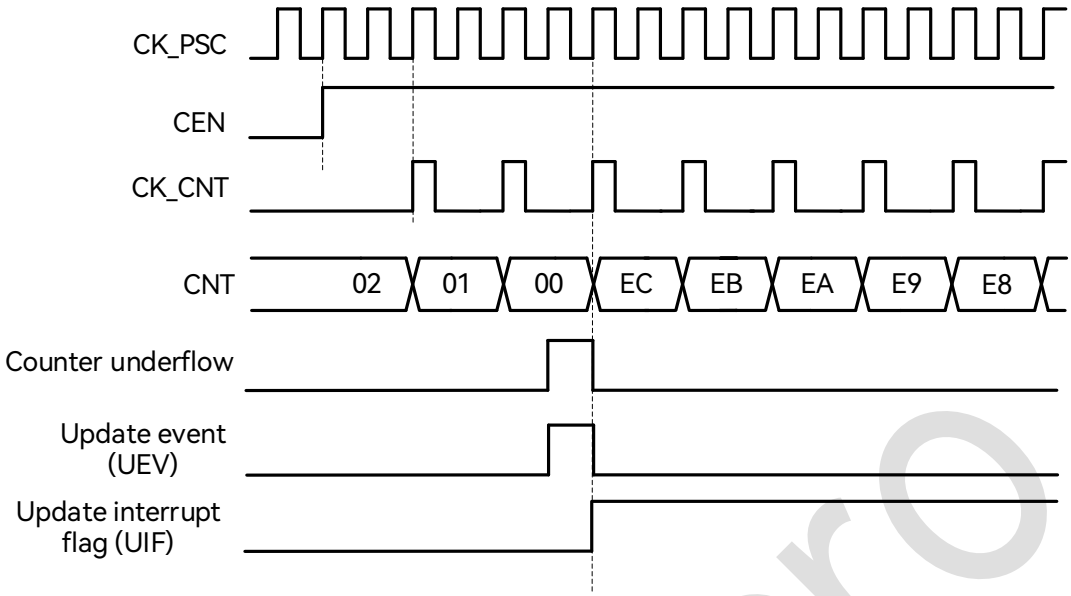


Figure 16-9: Down-counting Waveform Diagram, Internal Clock Divided by 2

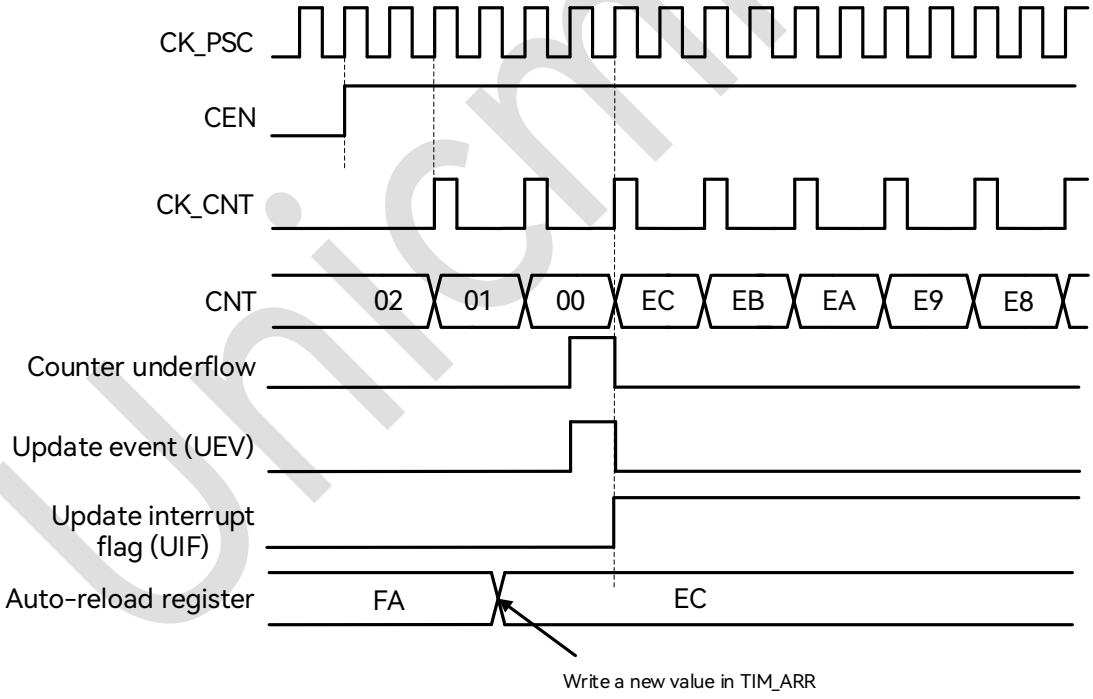


Figure 16-10: Down-counting Waveform Diagram, Internal Clock Divided by 2

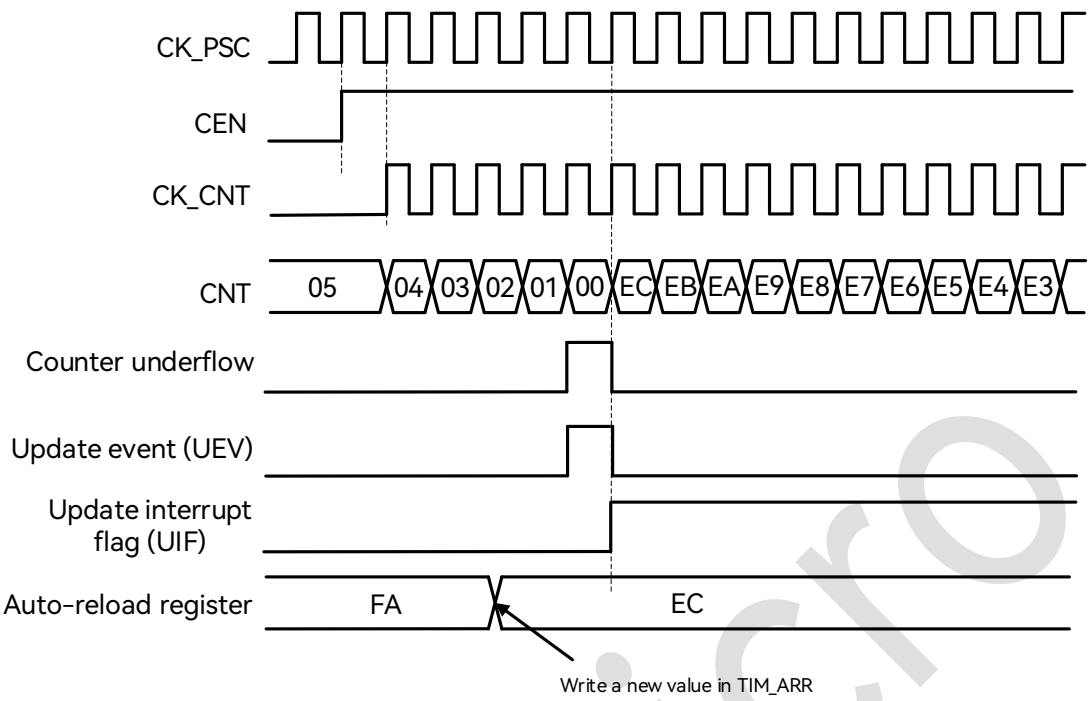


Figure 16-11: Down-counting Waveform Diagram, Update Event when Repetition Counter is not Used

16.6.2.3 Center-aligned Counting Mode

In center-aligned mode, the counter counts from 0 to the auto-reload value - 1 and generates a counter overflow event, then counts from the auto-reload value down to 1 and generates a counter underflow event, and then restarts counting from 0.

The CMS[1:0] bits are used for enabling the center-aligned mode and selecting the output compare mode herein. The center-aligned mode is active when CMS! = 00. The output compare interrupt flag of channels configured in output is set when: the counter counts down (CMS = 01), the counter counts up (CMS = 10), the counter counts up and down (CMS = 11).

In this mode, the DIR direction bit in the register cannot be written by software. It is updated by hardware and gives the current direction of the counter.

The counter updates the shadow registers of ARR, PSC and RCR at each counter overflow and underflow.

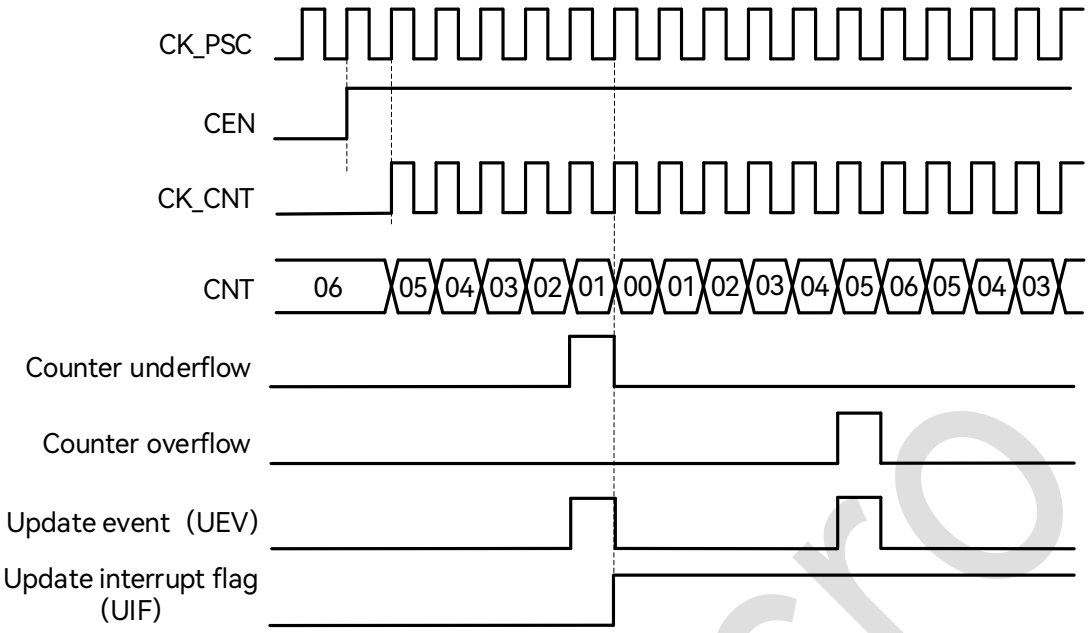


Figure 16-12: Center-aligned Counter Timing Diagram, TIM_PSC = 0, TIM_ARR = 0x6

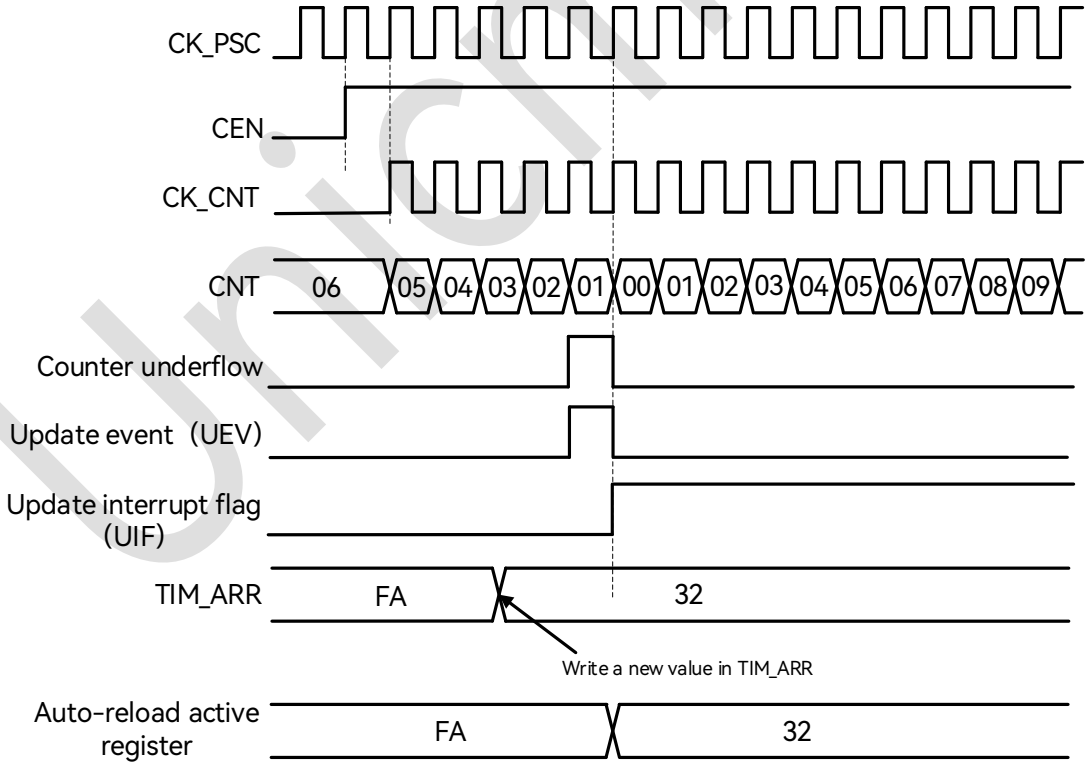


Figure 16-13: Counter Timing Diagram, Update Event with ARPE = 1 (Counter Underflow)

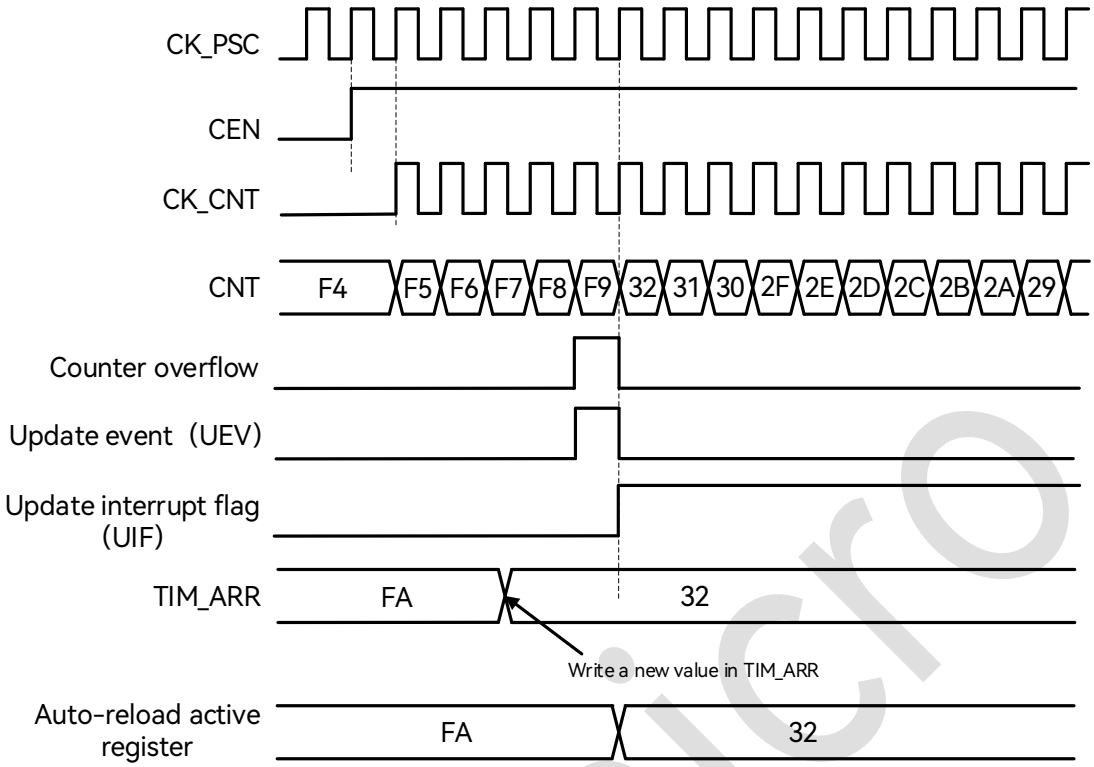


Figure 16-14: Counter Timing Diagram, Update Event with ARPE = 1 (Counter Overflow)

16.6.3 Repetition Counter

The update event is generated only when the repetition counter has reached zero at counter overflow or underflow. This means that the data are transferred from the preload registers of ARR, PSC and CCR (compare/capture registers in output compare mode) to the shadow registers every N + 1 counter overflows/underflows, where N is the value in the repetition counter register RCR.

The repetition counter is decremented:

- at each counter overflow in up-counting mode;
- at each counter underflow in down-counting mode;
- at each counter overflow / underflow in center-aligned mode.

Note: When the update event is generated by software or by hardware through the slave mode controller, it occurs immediately whatever the value of RCR is and the repetition counter is reloaded with the content of the TIM_RCR register.

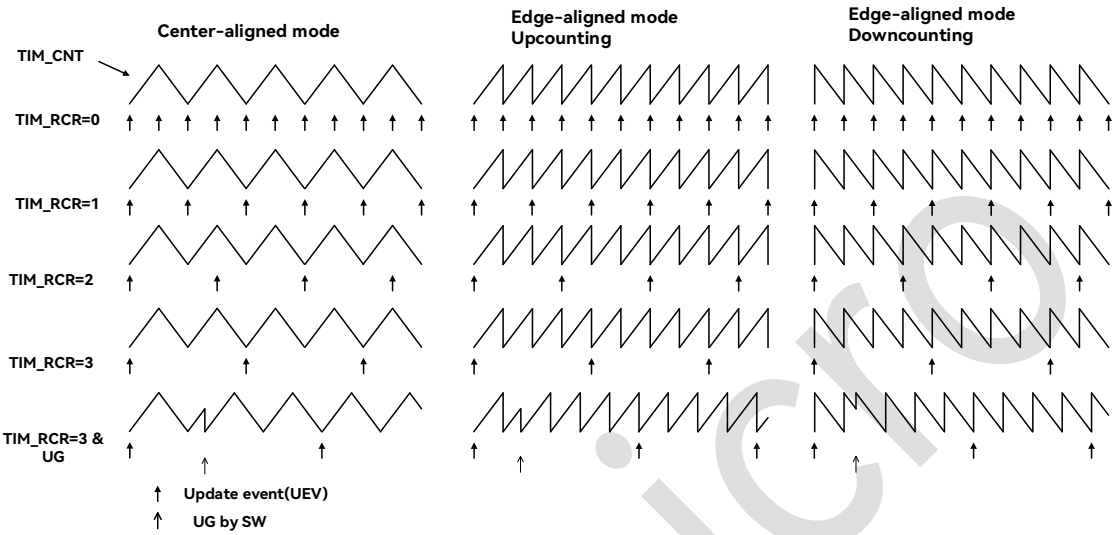


Figure 16-15: Examples of Updating Rate in Different Modes and TIM_RCR Register Setting

16.6.4 Preload Register

The following functional registers support the preload function:

- Auto-reload register TIM_ARR
- Prescaler register TIM_PSC (preload function cannot be disabled)
- Channel control register TIM_CCR
- CCxE and CCxNE control register
- OCxM control register

The preload function can be enabled or disabled by software for all of the above registers except TIM_PSC.

Registers with preload function contain two sets of physical entities:

- Shadow register: the register being used by the actual timer
- Preload register: the register accessible to software

When the preload function is disabled, the register with preload function has the following characteristics:

- The preload register can be accessed and overwritten by software in real time.
- The shadow register is updated synchronously with the preload register.

If the preload function is enabled, then:

- All software operations access the preload register.
- At the occurrence of update event, the content of all preload registers will be synchronously transferred to the corresponding shadow registers.

16.6.5 Counter Clock

The counter clock can be provided by the following clock sources:

- Internal clock: Timerx_clk
- External clock mode 1: external input pin TIx
- External clock mode 2: external trigger input ETR
- Internal trigger inputs (ITRx): using the trigger output (TRGO) of one timer as the counter clock

16.6.5.1 Internal Clock Source

If the slave mode controller is disabled (SMS = 000), then the CEN, DIR and UG bits are controlled by software. After the UG bit is set and the update signal is synchronized by CLK_PSC, the counter value is reinitialized.

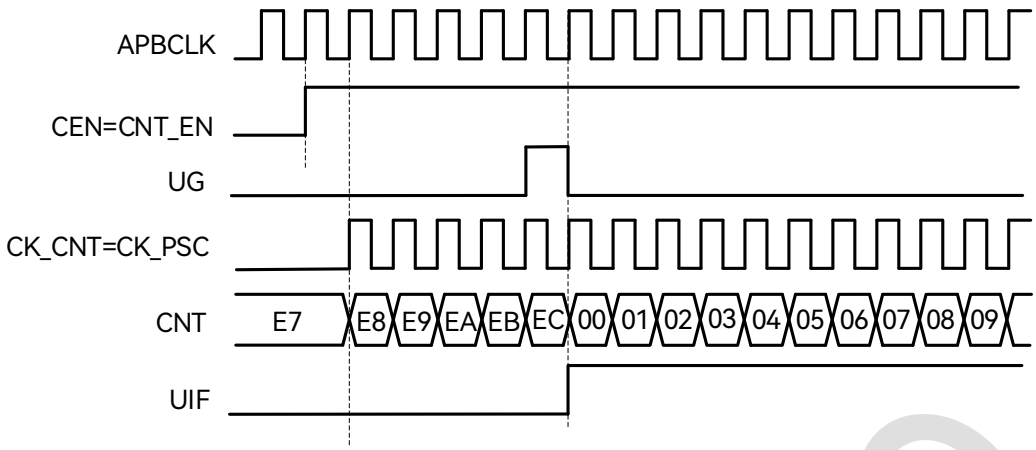


Figure 16-16: Timing Diagram in Internal Clock Source Mode, Clock Divided by 1

16.6.5.2 External Clock Source Mode 1

In this mode, the external pin input signal is directly used as the counter clock when SMS = 11, and the counter can count at each rising or falling edge on a selected input.

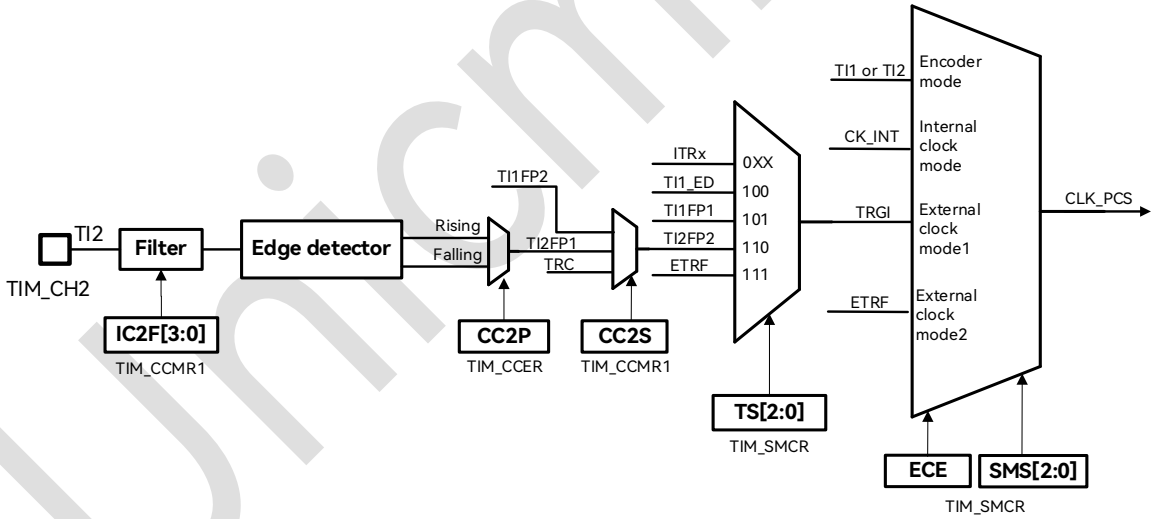


Figure 16-17: External Clock Connection Diagram

The external input signal will be synchronized with the internal clock before triggering the counter counting, and the TIF flag will be set by the valid edge on the input.

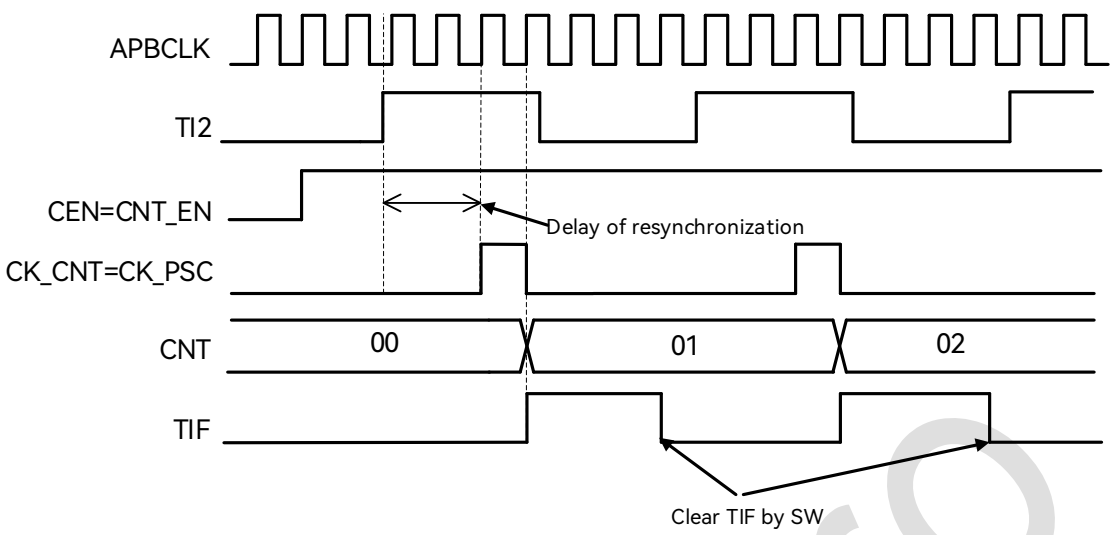


Figure 16-18: Timing Diagram in External Clock Source Mode 1

When counting with an external clock, the internal clock (timerx_clk) shall still be enabled so that TIM can use timerx_clk to synchronize and filter the external input clock. In external clock mode 1, the external input clock is first subject to filtering and edge selection to obtain a valid counting edge, which is input to the prescaler module as the valid operating clock (CLK_PSC).

The external clock synchronization adopts a simple two-stage flip-flop structure, so in order to avoid metastability, the external input clock width is required to be at least 2 timerx_clk cycles.

In this mode, only the inputs of channels 1 and 2 can be used as clock inputs, and the required configuration is as follows:

- In GPIO module, configure the corresponding pin as TIM_CH2.
- Turn off the channel enable bit, and configure TIM_CCER[4] = 0 to ensure the success of subsequent channel configuration.
- Select the input channel by setting TIM_CCMR1[9:8] = 01, with IC2 mapped on TI2.
- Select the active counting edge to be rising edge or falling edge by setting TIM_CCER[5] = 0.

- Configure the input filter duration by writing the IC2F[3:0] bits in the TIM_CCMR1 register (if no filter is required, keep IC2F = 0000).
- Enable the external clock source mode 1 by setting TIM_SMCR[2:0] = 111.
- Select TI2 as the trigger input source by setting TIM_SMCR[6:4] = 110.
- Enable the channel by setting TIM_CCER[4] = 1.
- Enable the counter by setting TIM_CR1[0] = 1.

The following diagram shows an example of typical external clock source mode 1:

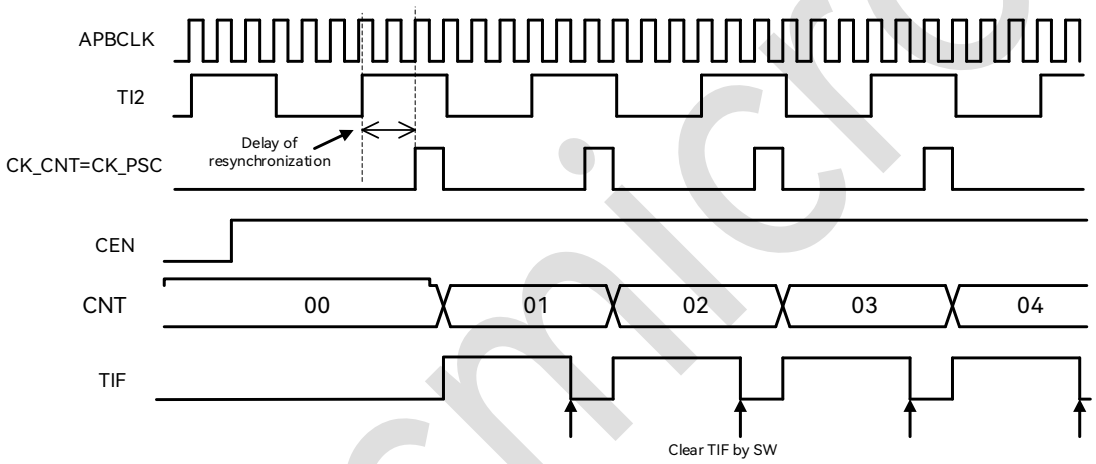


Figure 16-19: Timing Diagram in External Clock Source Mode 1

16.6.5.3 External Clock Source Mode 2

In this mode, the counter counts at each rising edge or falling edge (double-edge not supported) on the external trigger input TIM_ETR.

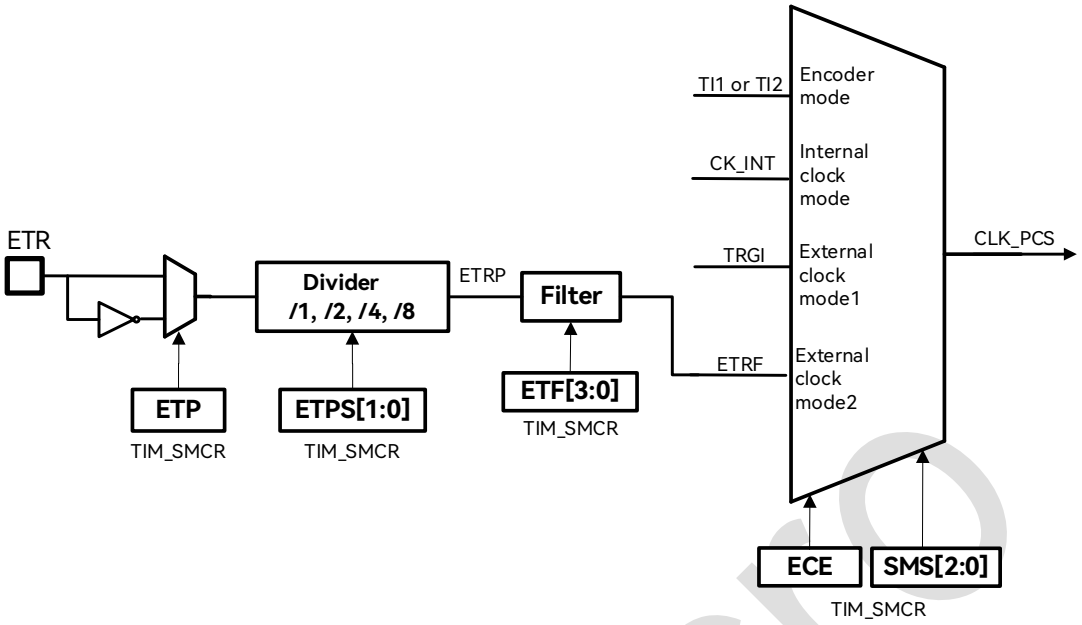


Figure 16-20: External Trigger Input Block Diagram

The following diagram shows the counter counting each 2 rising edges on ETR. The delay between the rising edge of ETR and the actual clock of the counter is due to the resynchronization of the internal clock.

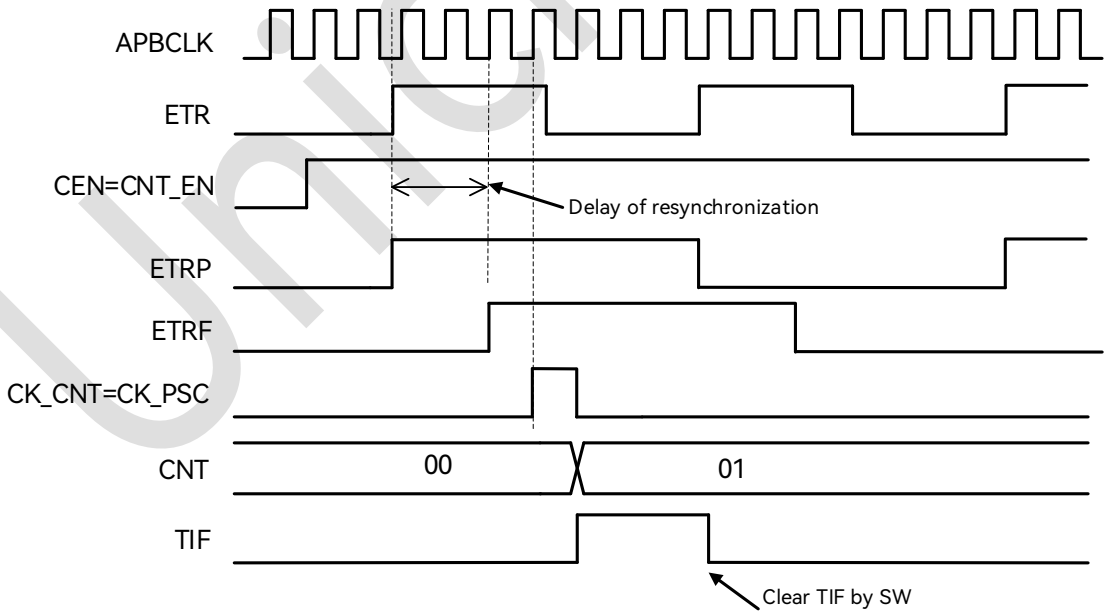


Figure 16-21: Timing 1 Diagram in External Clock Source Mode 2

The main difference from external clock source mode 1 is that the ETR input is directly divided and then filtered to generate CK_PSC clock, which means that the application scenarios where

the ETR input frequency is higher than timerx_clk can be supported, in which case the ETR input shall be pre-divided first before it is used to drive the counter.

The configuration required for this mode is as follows:

- In GPIO module, configure the corresponding pin as TIM_ETR.
- Select the ETP edge by setting TIM_SMCR[15] = 0.
- Set the ETR division ratio by writing TIM_SMCR.ETPS[1:0] = 01.
- Configure the input filter duration by setting TIM_SMCR.ETF[3:0] = 0000.
- Set the ECE register and enable the external clock source mode 2 by setting TIM_SMCR[14] = 1 and TIM_SMCR[2:0] = 000.
- Enable the counter by setting TIM_CR1[0] = 1.

The following diagram shows an example of typical external clock source mode 2:

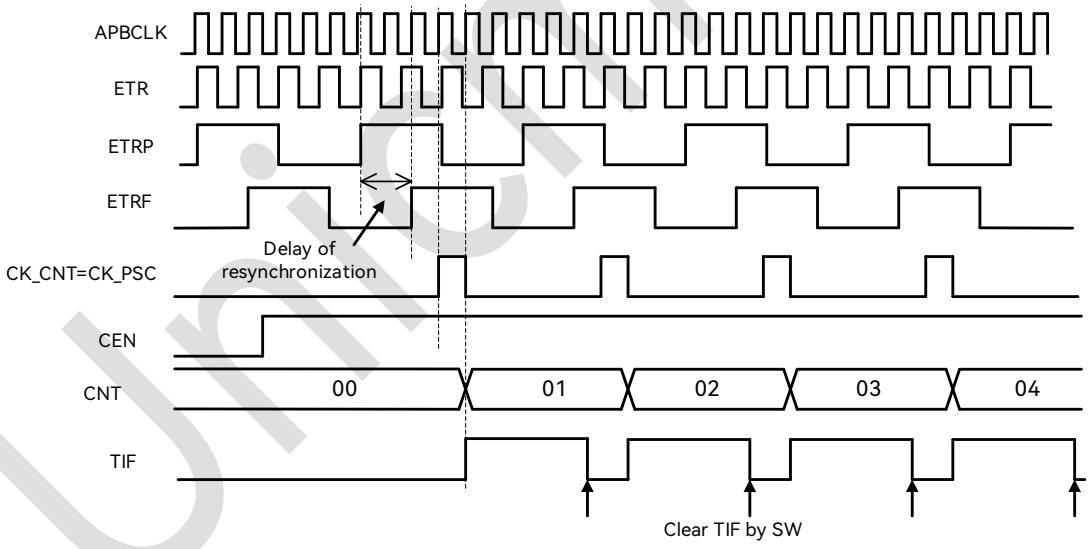


Figure 16-22: Timing 2 Diagram in External Clock Source Mode 2

In external clock source mode 2, TIM can still be configured as slave mode: for example, ETR input is used for counting while TRGO of another timer is used as the trigger signal, and the reset counter restarts counting at the arrival of trigger event.

16.6.6 Internal Trigger Signal (ITRx)

TIM supports four internal trigger inputs, which can be used for counting trigger or internal signal capture. For internal signal capture, it is required to configure TS to 000–011 for selecting ITR0–ITR3, and configure CCxS to 11 for selecting TRC (selected by four ITRx inputs) as the capture signal.

Each ITR input supports 4 internal signal extensions configured by the ITRxSEL register.

16.6.7 Capture / Compare Channels

TIM consists of 4 capture/compare channels, each of which is built around a capture/compare register TIM_CCR (including a shadow register), an input stage for capture and an output stage for compare.

The input stage samples the corresponding TIx input to generate a filtered signal TIxF. Then, an edge detection with polarity selection generates a signal (TIxFPx), which can be used as trigger input for counting or as the capture command and is prescaled before being captured.

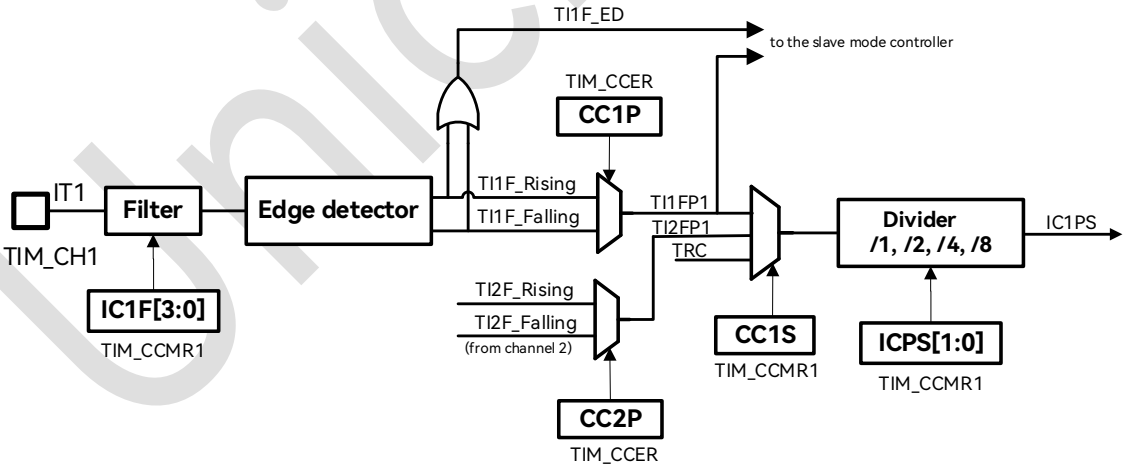


Figure 16-23: Capture/Compare Channel (Example: Channel 1 Input Stage)

The output stage generates an output reference signal OCxREF, which is fixed to be active high and acts as the reference input to the final output circuit. Wherein, channels 1–3 support

complementary output and dead-time insertion, while channel 4 is relatively simple and does not support complementary output.

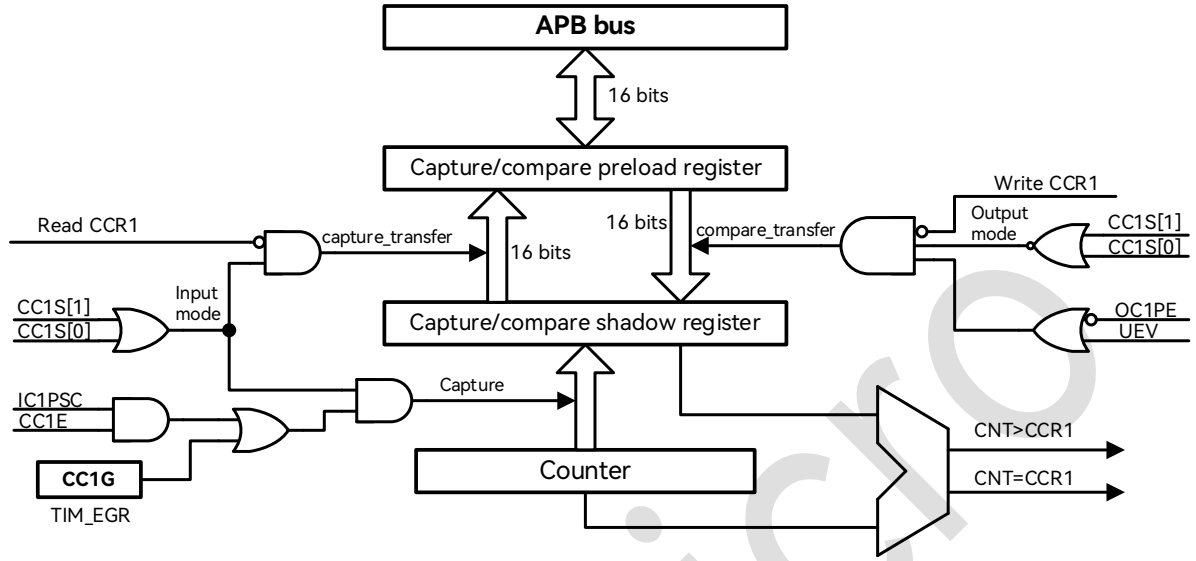


Figure 16-24: Capture/Compare Channel 1 Main Circuit

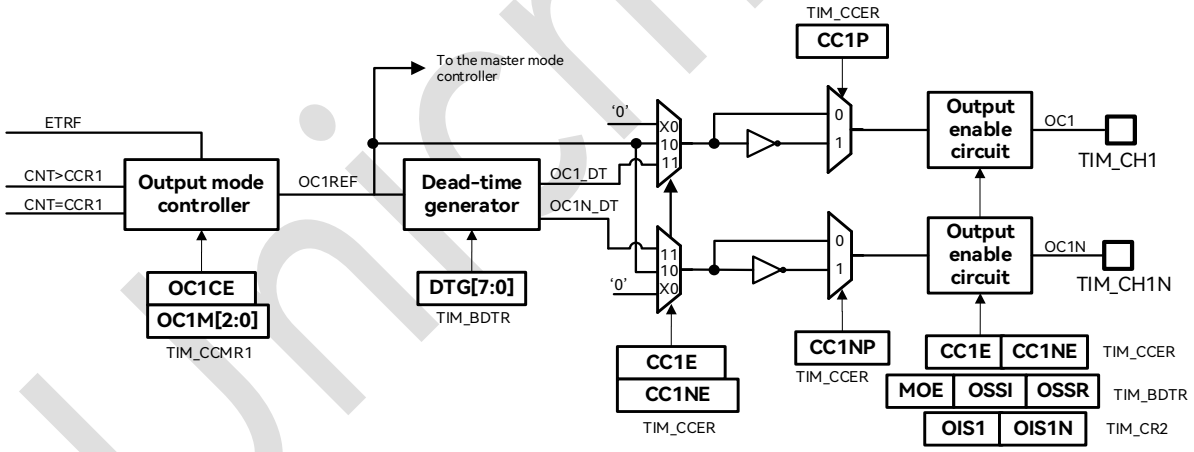


Figure 16-25: Output Stage of Capture/Compare Channel (Channel 1–3)

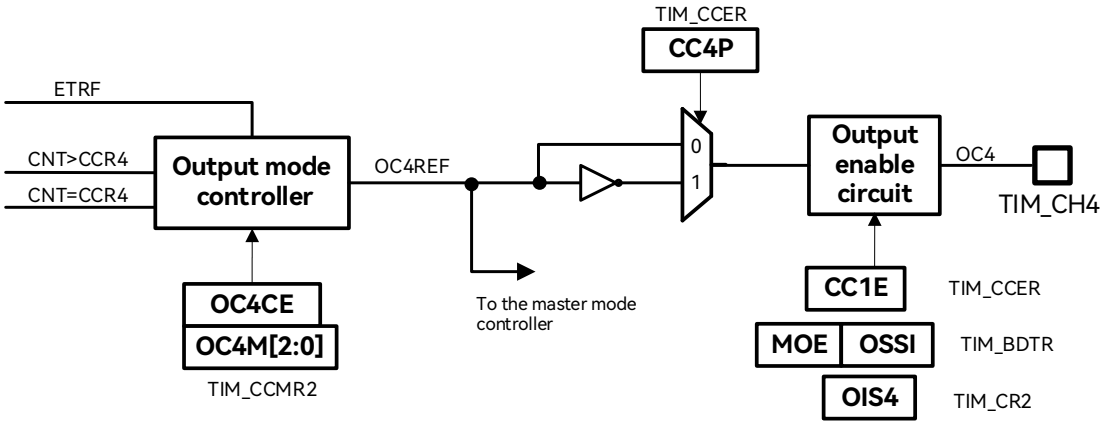


Figure 16-26: Output Stage of Capture/Compare Channel (Channel 4)

The capture/compare block is made of one preload register and one shadow register. Write and read always access the preload register. In capture mode, captures are actually done in the shadow register, which is copied into the preload register. In compare mode, the content of the preload register is copied into the shadow register for comparison with the counter.

16.6.8 Input Capture Mode

When the expected level transition is detected by the ICx signal, a capture is triggered and the current counter value is latched into CCR. At the same time, the CCxIF interrupt flag is set and a corresponding interrupt or a DMA request can be triggered. If a capture occurs while the CCxIF flag is already high, then the over-capture flag CCxOF is set (the last capture value in CCR is overwritten). CCxIF can be cleared by software or automatically cleared by reading the captured data stored in the CCR register. CCxOF is cleared when it is written with 1.

The input capture of PWM signals can be realized through the cooperation of two or more channels. For example, to calculate the period and duty cycle of an input signal, input the signal from TI1 pin, and take the rising edge and falling edge of the filtered signal inside the chip to obtain TI1FP1 and TI1FP2 respectively. Input TI1FP1 into capture channel 1 and TI1FP2 into channel 2, allowing channel 1 to capture the rising edge of the input signal and channel 2 to capture the falling edge of the input signal. After the capture interrupt occurs periodically,

the software can calculate the period and duty cycle of the input signal through the values of CCR1 and CCR2 registers.

To capture the counter value to the TIM_CCR1 register on the rising edge of the TI1 input, the configuration steps are as follows:

1. In GPIO module, configure the corresponding pin as TIM_CH1.
2. Disable the channel by setting TIM_CCER[0] = 0 to ensure the success of subsequent channel configuration.
3. Select the input channel by setting TIM_CCMR1[1:0] = 01, with IC1 mapped on TI1.
4. Select the active counting edge to be rising edge or falling edge by setting TIM_CCER[1].
5. Configure the input filter duration by setting the IC1F[3:0] bits in the TIM_CCMR1 register.
6. Configure the input prescaler by setting the IC1PS[1:0] bits in the TIM_CCMR1 register.
7. Enable the channel by setting TIM_CCER[0] = 1.

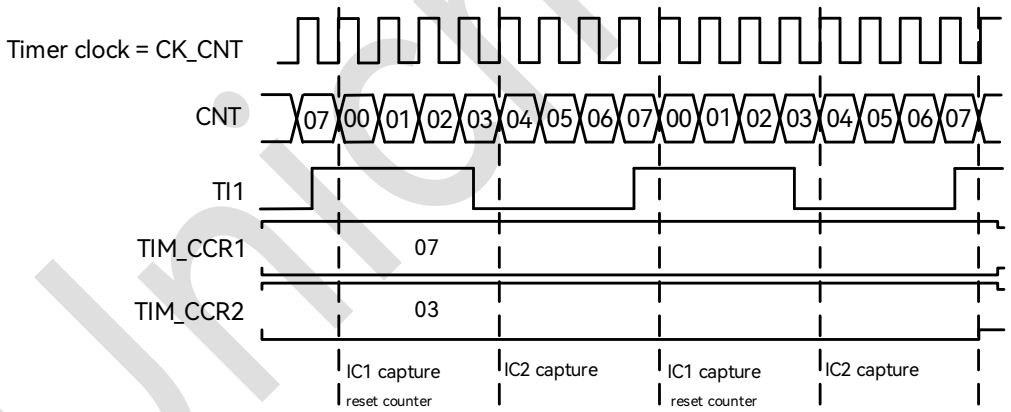


Figure 16-27: PWM Input Capture Mode Timing Diagram

The following settings are required for PWM input capture:

1. In GPIO module, configure the corresponding pin as TIM_CH1.
2. Disable the channel by setting TIM_CCER[0] = 0 and TIM_CCER[4] = 0 to ensure the success of subsequent channel configuration.
3. Select the input channel, with the two signals IC1 and IC2 mapped on the same TI1 input, and configure TIM_CCMR1[1:0] = 01 and TIM_CCMR1[9:8] = 01.

4. Select the active counting edge, with the two signals IC1 and IC2 active on edges with opposite polarities, and configure TIM_CCER[1] = 0 and TIM_CCER[5] = 1.
5. Configure the input filter duration by setting the IC1F[3:0] and IC2F[3:0] bits in the TIM_CCMR1 register.
6. Configure the input prescaler by setting the IC1PS[1:0] and IC2PS[1:0] bits in the TIM_CCMR1 register.
7. Select the trigger input source by setting TIM_SMCR.TS[2:0] = 101.
8. Set the slave mode controller to reset mode, and configure TIM_SMCR.SMS[2:0] = 100.
9. Enable the channel, and configure TIM_CCER[0] = 1 and TIM_CCER[4] = 1.

16.6.9 Forced Output Mode

In output compare mode, the OCxREF signal can be forced to active or inactive level directly by software, independently of any comparison between the CCR and the counter.

The OCxREF signal can be directly forced to be active (OCxREF is always active high) by software writing OCxM = 101, and forced to be inactive (low level) by writing OCxM = 100. Anyway, the comparison between CCR and the counter is still performed.

16.6.10 Output Compare Mode

In output compare mode, when a match is found between the capture/compare register CCR and the counter, the OCxREF can be set to be active, inactive or can toggle on match. At the same time, the interrupt flag is also set and DMA requests can be sent.

The output compare can also be used to output a pulse signal of a specific width (in one-pulse mode).

Procedure:

1. Select the counter clock (internal, external, prescaler).

2. Write the desired data in the ARR and CCR registers.
3. Set the interrupt enable bit and DMA enable bit as required.
4. Select the output mode.
5. Enable the counter.

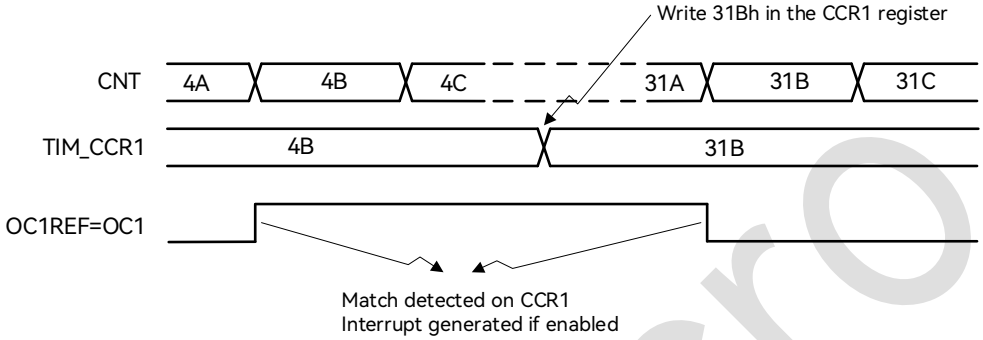


Figure 16-28: Output Compare Mode, Toggle on OC1

The CCR register can be updated at any time by software to control the output waveform, provided that the preload register is not enabled. Otherwise, the CCR shadow register is only updated with the content of the preload register at the next update event.

16.6.11 PWM Output

PWM mode allows you to generate a pulse width modulation signal with a frequency determined by the value of the ARR register and a duty cycle determined by the value of the CCR register.

The polarity of the output signal is software programmable using the CCxP bit in the register. In PWM mode, CNT and CCR registers are always compared. The timer is able to generate PWM in edge-aligned mode or center-aligned mode.

16.6.11.1 PWM Edge-aligned Mode

In up-counting mode, when it is configured in PWM mode 1, the OCxREF signal is high as long as CNT < CCR, otherwise it is low. And OCxREF will be held at 1 if CCR > ARR while held at 0 if CCR is 0.

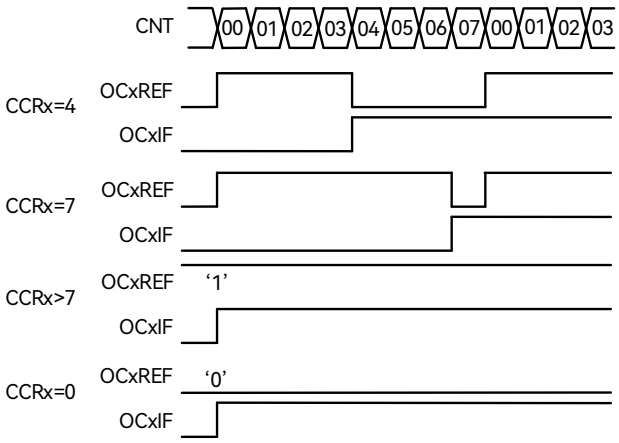


Figure 16-29: Edge-aligned PWM Waveform (ARR = 7)

In down-counting mode, the definition of OCxREF level is the same as that in up-counting mode.

16.6.11.2 PWM Center-aligned Mode

The definition of OCxREF level is the same as that in edge-aligned mode. The figure below is an example.

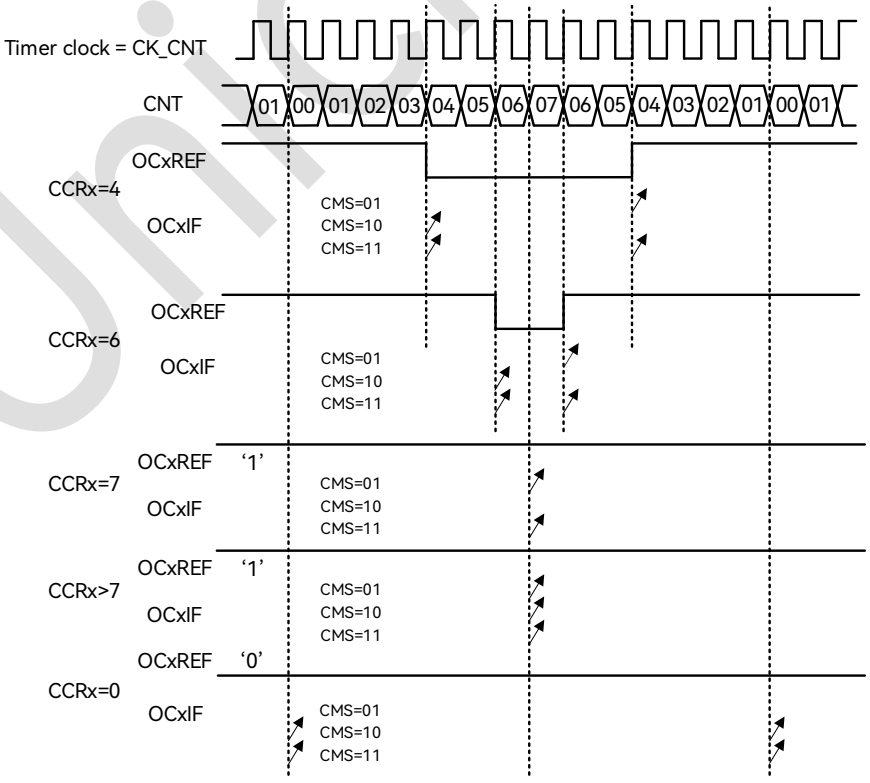


Figure 16-30: Center-aligned PWM Waveform (ARR = 7)

When start counting in center-aligned mode, the initial counting direction is determined by the DIR bit in the register, and in the subsequent process, the DIR bit is directly controlled by hardware. The safest way to use center-aligned mode is to generate an update by setting the UG bit in the register just before starting the counter and not to overwrite the counter while it is running.

16.6.12 Asymmetric PWM Mode

Asymmetric mode allows two center-aligned PWM signals to be generated with a programmable phase shift. In this mode, a pair of CCR registers can be used to determine the duty cycle and phase. One register controls the PWM during up-counting, and the other during down-counting, so that PWM is adjusted every half PWM cycle:

OC1REFC or OC2REFC is controlled by CCR1 and CCR2; OC3REFC or OC4REFC is controlled by CCR3 and CCR4.

Asymmetric PWM mode can be selected by writing 1110 or 1111 in the OCxM[3:0] bits in the CCMR register.

When a channel is used as asymmetric PWM channel, its complementary channel is recommended to be configured in the opposite PWM mode. For instance, one in asymmetric PWM mode 1 and the other in normal PWM mode 2.

When a given channel is used as asymmetric PWM channel, its complementary channel can also be used.

16.6.13 Combined PWM Mode

Combined PWM mode allows two edge or center-aligned PWM signals to be generated with programmable delay and phase shift between respective pulses. In this mode, a pair of CCR registers can be used to determine the duty cycle and phase. The resulting signals are made

of an OR or AND logical combination of two reference PWMs. Wherein OC1REFC or OC2REFC is controlled by CCR1 and CCR2; OC3REFC or OC4REFC is controlled by CCR3 and CCR4.

Combined PWM mode can be selected by writing 1100 or 1101 in the OCxM[3:0] bits in the CCMR register.

When a channel is used as combined PWM channel, its complementary channel is recommended to be configured in the opposite PWM mode. For instance, one in combined PWM mode 1 and the other in normal PWM mode 2.

When a given channel is used as combined PWM channel, its complementary channel can also be used.

16.6.14 Complementary Output and Dead-time Insertion

Channels 1–3 of TIM support complementary output and dead-time insertion. The DTG[7:0] bits in the register are used to set the dead-time delay (valid for all channels at the same time). The output signal OCx is in phase with the reference signal OCxREF, with the rising edge delayed relative to the reference rising edge. The output signal OCxN is in the opposite phase with the reference signal OCxREF, with the rising edge delayed relative to the reference falling edge.

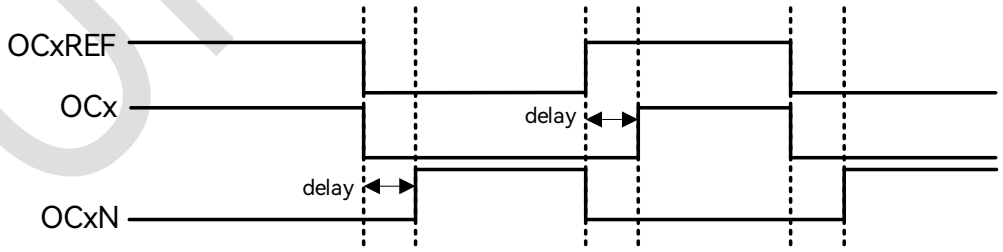


Figure 16-31: Waveform of Complementary Output with Dead-time Insertion

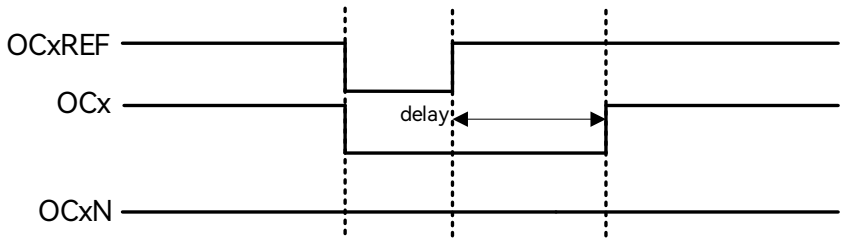


Figure 16-32: Dead-time Waveform with Delay Greater than Negative Pulse

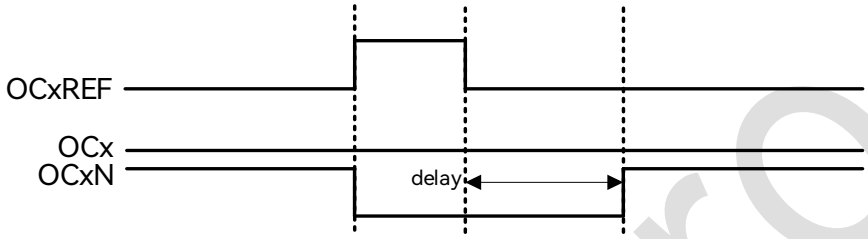


Figure 16-33: Dead-time Waveform with Delay Greater than Positive Pulse

16.6.15 Break Function

The break function can be activated using an external break signal or a clock fault signal.

When a break occurs:

- The output enable register is cleared asynchronously, and the output can be forced to inactive, idle or reset state (selected by the OSSI bit).
- Each output channel is driven with the level programmed in the OISx bit in the register.
- When complementary outputs are enabled, the outputs are asynchronously set to the inactive state and reset state, and the dead-time insertion circuit starts to work, driving the output to the level defined by OISx and OIXN after the dead time.
- The break status flag is set. An interrupt or DMA request can be triggered according to the configuration.
- If the automatic output is enabled (AOE = 1), the MOE bit will be automatically set at the next update event; otherwise, MOE will remain 0 until it is reset by software.

Note: The BRK signal is active on level, thus MOE cannot be set while BRK remains active. In the meantime, the break flag BIF cannot be cleared.

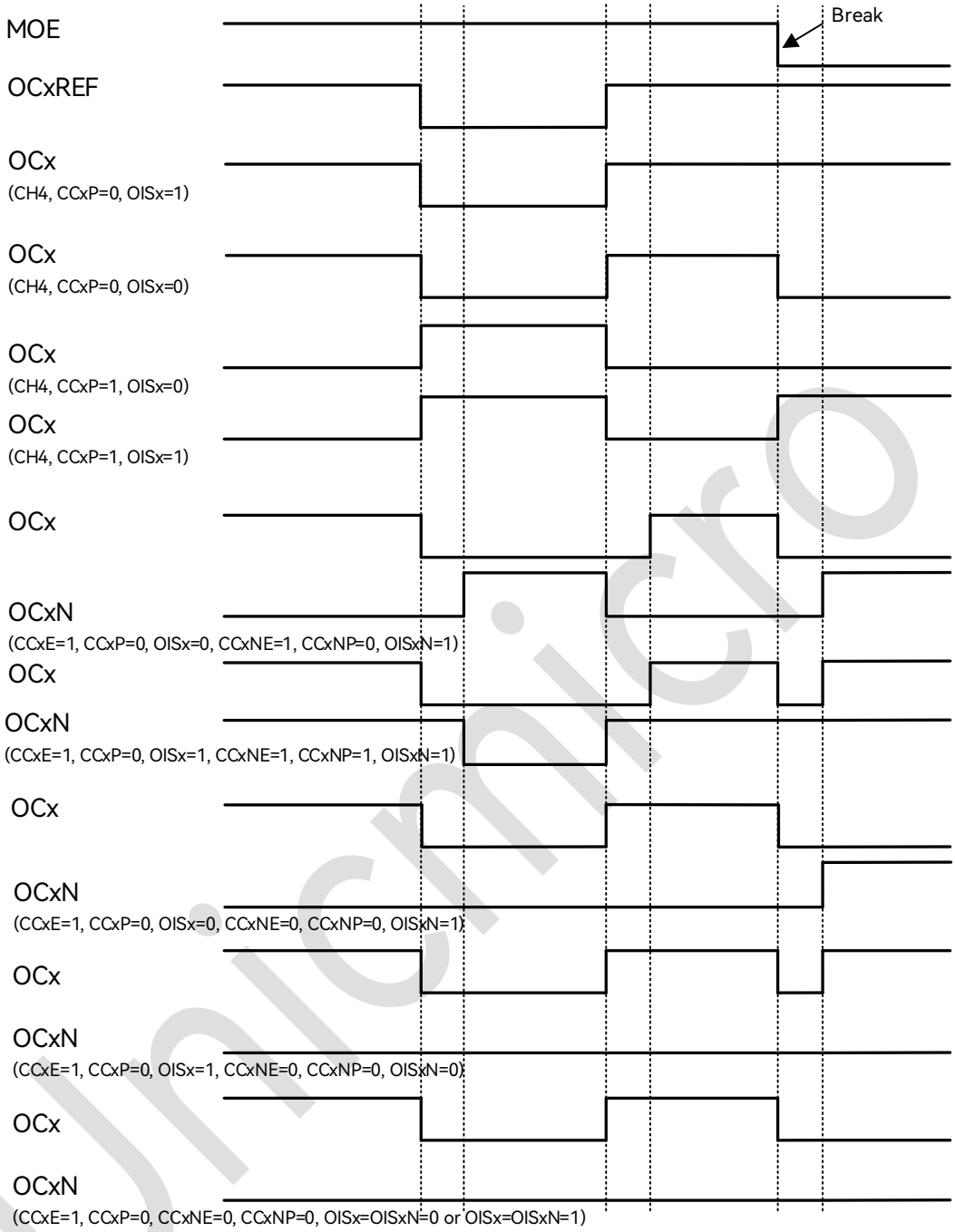


Figure 16-34: Various Output Behavior in Response to A Break Event

16.6.16 6-step PWM Generation

When complementary outputs are used on a channel, preload bits are available on the OCxM, CCxE and CCxNE bits. The preload bits are transferred to the shadow bits at the COM commutation event. Thus one can program in advance the configuration for the next step and

update the configuration of all the channels at the same time. COM can be generated by software by setting the COM bit in the TIM_EGR register or by hardware (on TRGI rising edge).

A flag is set when the COM event occurs, which can generate an interrupt or a DMA request.

The figure below describes the behavior of the OCx and OCxN outputs when a COM event occurs, in 3 different examples of programmed configurations.

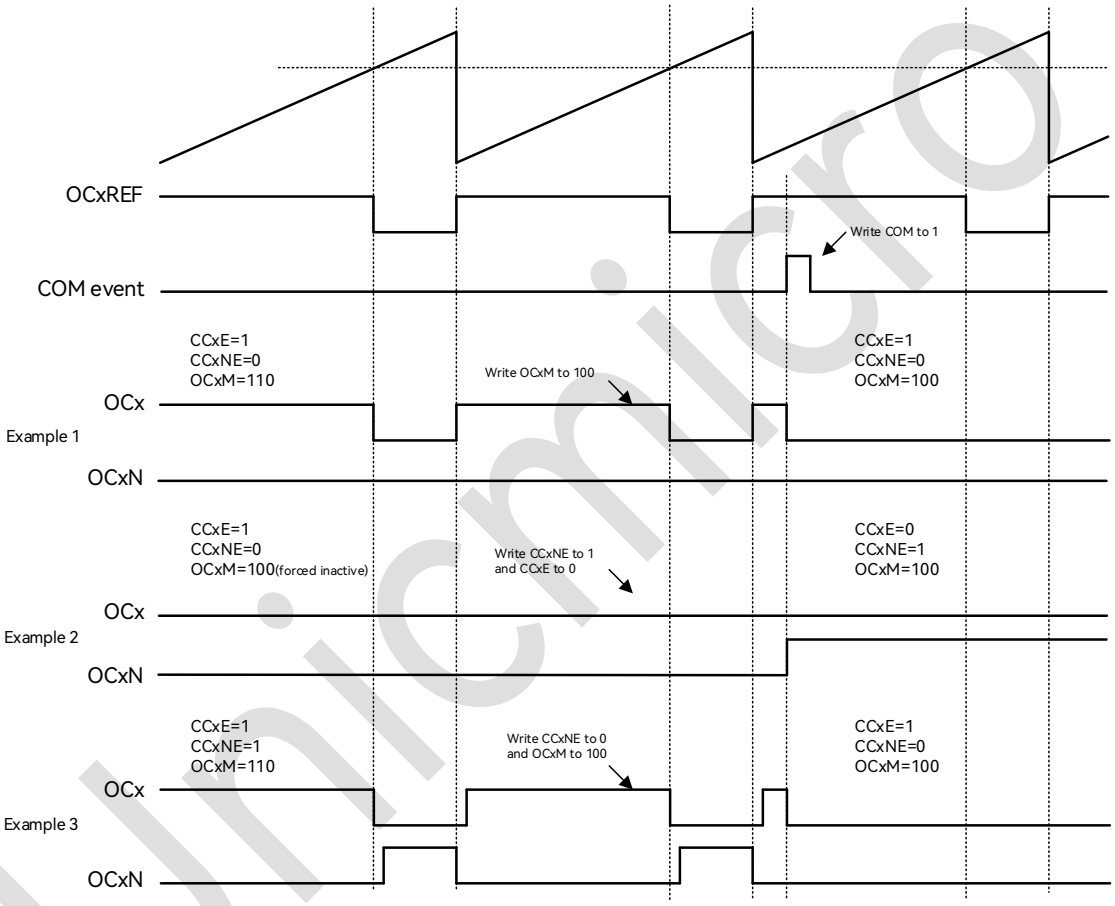


Figure 16-35: 6-step PWM Generation, COM Example (OSSR = 1)

16.6.17 One-pulse Output Mode

One-pulse output mode is a particular case of the compare output mode, which allows the counter to generate a pulse with a programmable length after a programmable delay following the occurrence of an event.

Different from other output modes, the counter will stop automatically at the next update event. A pulse can be correctly generated only if the compare value is different from the counter initial value. In up-counting, it is required that CNT < CCR ≤ ARR; in down-counting, it is required that CNT > CCR.

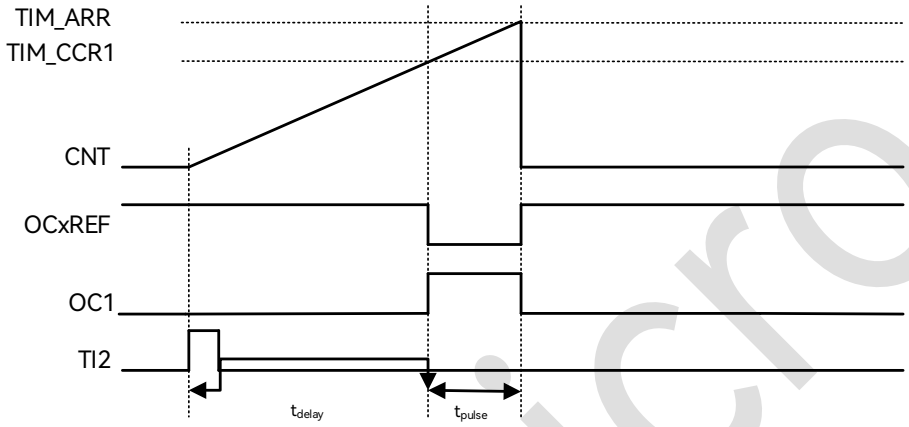


Figure 16-36: Timing Diagram of One-pulse Mode

In the above figure, TI2 input is used as the counter trigger signal. When the count value reaches CCR, the OCxREF outputs a low level. Once the counter counts up to ARR, the OCxREF signal returns to a high level, and the counter rolls back to 0, stopping the counting process.

The configuration for realizing the above function of TI2 as an input trigger is as follows:

1. In GPIO module, configure the corresponding pin as TIM_CH2.
2. Disable the channel by setting TIM_CCER[4] = 0 to ensure the success of subsequent channel configuration.
3. Select the input channel by setting TIM_CCMR1[9:8] = 01.
4. Select the active counting edge by setting TIM_CCER[5] = 0.
5. Select TI2FP2 as the trigger input source by setting TIM_SMCR.TS[2:0] = 110.
6. Set the slave mode controller to trigger mode by setting TIM_SMCR.SMS[2:0] = 110, with TI2FP2 for activating the counter.
7. Enable the channel by setting TIM_CCER[4] = 1.

The configuration for realizing the above function of OC1 as an output is as follows:

1. In GPIO module, configure the corresponding pin as TIM_CH1.
2. Disable the channel by setting TIM_CCER[0] = 0 to ensure the success of subsequent channel configuration.
3. Select the output channel by setting TIM_CCMR1[1:0] = 00.
4. Select the active counting edge by setting TIM_ CCMR1[6:4] = 111, in PWM mode 2.
5. Enable the channel by setting TIM_CCER[0] = 1.

Special settings for generating OPM waveform timing:

1. t_{delay} is determined by the value of TIM_CCR1.
2. t_{pulse} is determined by the difference between TIM_ARR and TIM_CCR1 (TIM_ARR – TIM_CCR1).
3. Configure to one-pulse mode by setting TIM_CR1[3] = 1.

16.6.18 Clearing OCxREF Signal on External Event

OCxREF is active at high level, and it can be pulled down directly until the next update event by applying a high level to the external ETR pin. This function can only be used in output compare and PWM modes, and does not work in forced mode. Enabling this function requires setting OCxCE to 1.

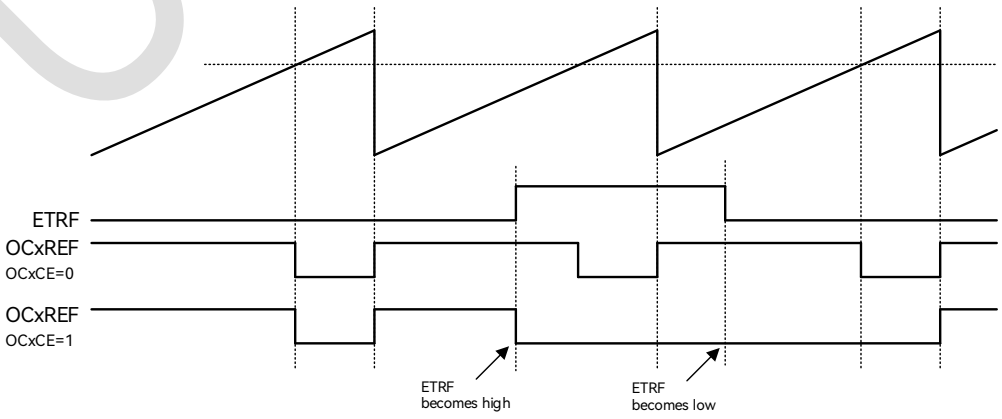


Figure 16-37: Timing Diagram of ETR Signal Clearing OCxREF of TIM

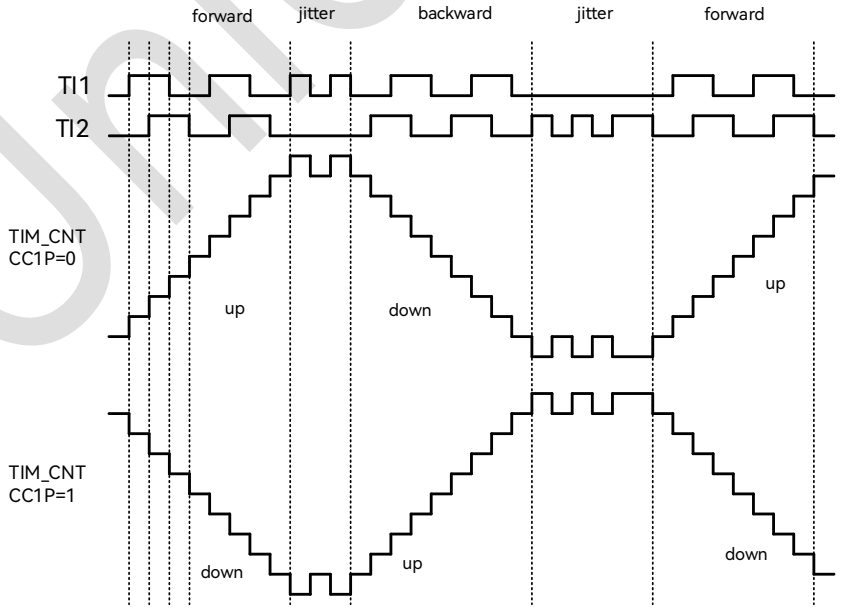
16.6.19 Encoder Interface Mode

The encoder interface mode involves two external input signals. The TIM determines whether to count up or down according to the edge of one signal relative to the level of the other signal. The following table shows the relationship between the counting mode and the two inputs:

Table 16-3: Counting Direction versus Encoder Signals

Active Edge	Level on Opposite Signal (TI1 for TI2, TI2 for TI1)	TI1 Signal		TI2 Signal	
		Rising	Falling	Rising	Falling
Counting on TI1 only	High	Down	Up	No count	No count
	Low	Up	Down	No count	No count
Counting on TI2 only	High	No count	No count	Up	Down
	Low	No count	No count	Down	Up
Counting on TI1 and TI2	High	Down	Up	Up	Down
	Low	Up	Down	Down	Up

For example, when the counter is counting on TI1, it will count down if TI2 is sampled as high level on the rising edge of TI1, and count up if TI2 is sampled as high level on the falling edge of TI1.



Example of counter operation in encoder interface mode

Figure 16-38: Example of Counter Operation in Encoder Interface Mode

The input channels in encoder interface mode shall be set as follows:

1. In GPIO module, configure the corresponding pins with TIM_CH1 and TIM_CH2 functions.
2. Disable the channel by setting TIM_CCER[0] = 0 and TIM_CCER[4] = 0 to ensure the success of subsequent channel configuration.
3. Select the input channel by setting TIM_CCMR1[1:0] = 01 and TIM_CCMR1[9:8] = 01.
4. Select the active counting edge by setting TIM_CCER[1] = 0 and TIM_CCER[5] = 0.
5. Set the slave mode controller to encoder mode 3 by setting TIM_SMCR.SMS[2:0] = 011.
6. Enable the channel by setting TIM_CCER[0] = 1 and TIM_CCER[4] = 1.

16.6.20 TIM Slave Mode

When TIM is used as a slave (triggered by an external event), it can be configured to operate in three modes: reset mode, gated mode, and trigger mode.

16.6.20.1 Reset Mode

In this mode, all the preload registers in TIM will be reinitialized in response to an event on a trigger input, and the counter will restart from 0. The following figure shows that the counter behaves normally until rising edge is detected on TI1 input, at which time the counter is cleared and restarts from 0.

The configuration in the following figure example is as follows:

1. In GPIO module, configure the corresponding pin as TIM_CH1.
2. Disable the channel by setting TIM_CCER[0] = 0 to ensure the success of subsequent channel configuration.
3. Select the input channel by setting TIM_CCMR1[1:0] = 01.
4. Select the active counting edge by setting TIM_CCER[1] = 0.
5. Select TI1FP1 as the trigger input source by setting TIM_SMCR.TS[2:0] = 101.
6. Configure the slave mode controller to reset mode by setting TIM_SMCR.SMS[2:0] = 100.

7. Enable the channel by setting TIM_CCER[0] = 1.
8. Enable the counter by setting TIM_CR1[0] = 1.

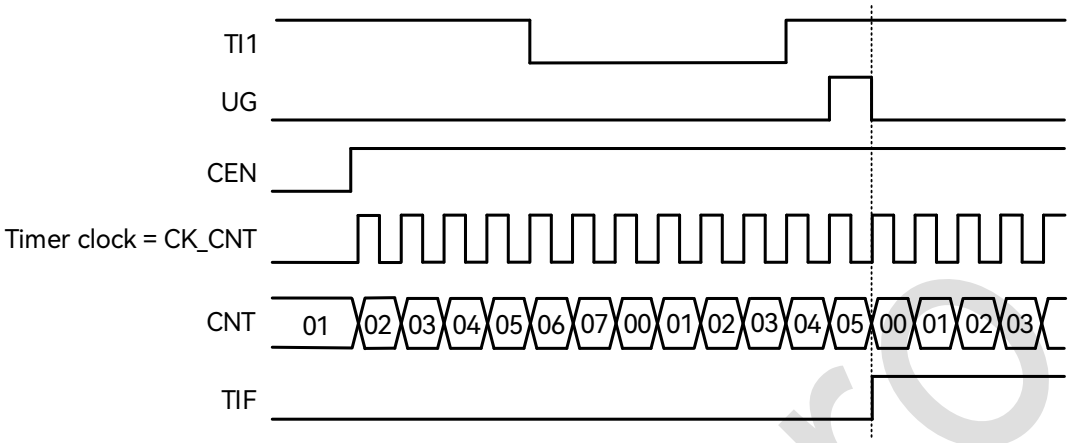


Figure 16-39: Timing Diagram in Reset Mode

16.6.20.2 Gated Mode

In this mode, the counter can be enabled depending on the level of a selected input. The interrupt flag is triggered whenever a level shift causes the counter to start or stop counting.

The configuration in the following figure example is as follows:

1. In GPIO module, configure the corresponding pin as TIM_CH1.
2. Disable the channel by setting TIM_CCER[0] = 0 to ensure the success of subsequent channel configuration.
3. Select the input channel by setting TIM_CCMR1[1:0] = 01.
4. Select the active counting edge by setting TIM_CCER[1] = 0.
5. Select TI1FP1 as the trigger input source by setting TIM_SMCR.TS[2:0] = 101.
6. Configure the slave mode controller to gated mode by setting TIM_SMCR.SMS[2:0] = 101.
7. Enable the channel by setting TIM_CCER[0] = 1.
8. Enable the counter by setting TIM_CR1[0] = 1.

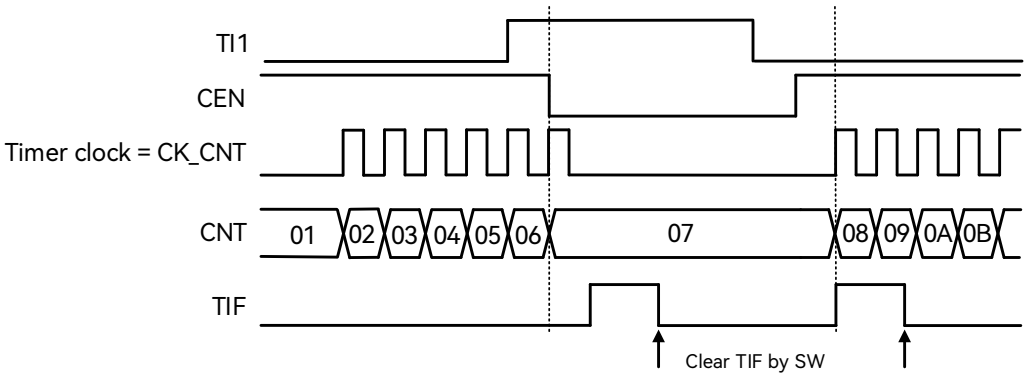


Figure 16-40: Timing Diagram in Gated Mode

16.6.20.3 Trigger Mode

The counter can start in response to an event on a selected input.

The configuration in the following figure example is as follows:

1. In GPIO module, configure the corresponding pin as TIM_CH1.
2. Disable the channel by setting TIM_CCER[0] = 0 to ensure the success of subsequent channel configuration.
3. Select the input channel by setting TIM_CCMR1[1:0] = 01.
4. Select the active counting edge by setting TIM_CCER[1] = 0.
5. Select TI1FP1 as the trigger input source by setting TIM_SMCR.TS[2:0] = 101.
6. Configure the slave mode controller to trigger mode by setting TIM_SMCR.SMS[2:0] = 110.
7. Enable the channel by setting TIM_CCER[0] = 1.

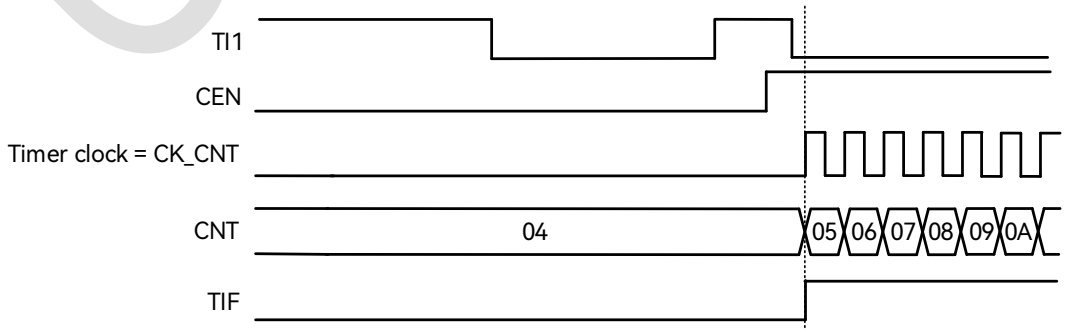


Figure 16-41: Timing Diagram in Trigger Mode

16.6.20.4 External Clock Mode 2 + Trigger Mode

In this mode, ETR can be set as the counting clock, while another external input is used as a trigger signal to start the counter. For instance, the counter begins counting on the rising edge of the ETR input after detecting the rising edge of TI1.

The configuration in the following figure example is as follows:

1. In GPIO module, configure the corresponding pins as TIM_CH1 and TIM_ETR.
2. Select the ETP edge by setting TIM_SMCR[15] = 0.
3. Set the ETR division ratio by setting TIM_SMCR.ETPS[1:0] = 01.
4. Configure the input filter duration by setting TIM_SMCR.ETF[3:0] = 0000.
5. Set the ECE register and enable the external clock mode 2 by setting TIM_SMCR[14] = 1.
6. Disable the channel by setting TIM_CCER[0] = 0 to ensure the success of subsequent channel configuration.
7. Select the input channel by setting TIM_CCMR1[1:0] = 01.
8. Select the active counting edge by setting TIM_CCER[1] = 0.
9. Select TI1FP1 as the trigger input source by setting TIM_SMCR.TS[2:0] = 101.
10. Configure the slave mode controller to trigger mode by setting TIM_SMCR.SMS[2:0] = 110.
11. Enable the channel by setting TIM_CCER[0] = 1.

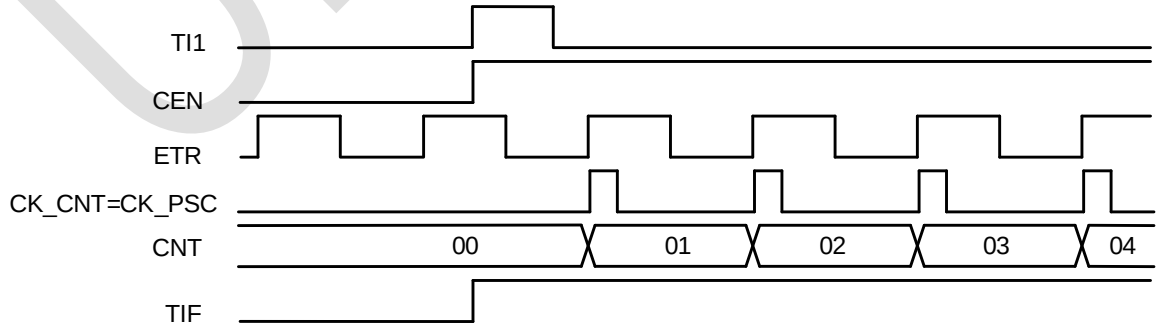


Figure 16-42: Timing Diagram in External Clock Mode 2 + Trigger Mode

16.6.21 Timer Synchronization

Timers can be cascaded together through trigger events to achieve synchronization or cascading operation.

A timer can utilize four internal trigger inputs, allowing the trigger signal output from one timer to connect to the internal trigger input of other timers.

16.6.22 DMA Access

TIM supports seven types of DMA requests, including four CC channel requests, an external trigger request, an update event request, and a COM trigger request.

Each CC channel generates a DMA request, which is used to transfer the content of CCRx to RAM in capture mode, and to write the data in RAM to CCRx in compare mode.

In addition, DMA requests can also be generated from external trigger event, software trigger event and COM trigger event, and at the occurrence of these requests, DMA burst transfer will be started to write data to one or more registers within TIM or to read one or more register values from TIM.

Table 16-4: Seven DMA Requests Supported by TIM

DMA Request	DMA Access Object	Single-transfer Length
TIM_CH1	Read DMAR	DBL
	Write DMAR	
TIM_CH2	Read DMAR	DBL
	Write DMAR	
TIM_CH3	Read DMAR	DBL
	Write DMAR	
TIM_CH4	Read DMAR	DBL
	Write DMAR	
TIM_TRIG	Read DMAR	DBL
	Write DMAR	
TIM_UEV	Read DMAR	DBL
	Write DMAR	

DMA Request	DMA Access Object	Single-transfer Length
TIM_COM	Read DMAR	DBL
	Write DMAR	

16.6.23 DMA Burst

TIM supports DMA and DMA-burst access. A DMA request can be generated at a specific event, so as to write the capture result in CCR to RAM or write the content of one or more registers in RAM to the preload register in TIM.

DMA-burst allows to generate multiple successive DMA requests upon a single event. The main purpose is to update the content of multiple registers in a row each time a given timer event is triggered, thus making it possible to dynamically modify the output waveform in real time.

The DMA controller destination is unique and must be directed to the virtual register TIM_DMAR. On a given timer event, the timer launches a sequence of DMA requests (burst). Each DMA write access to the TIM_DMAR register will be redirected to the actual function register by TIM.

The DBL bits in the register set the DMA burst length, and the DBA bits define the base address for DMA access to TIM (an offset starting from the address of the TIM_CR register).

In DMA-burst mode, all DMA access shall be directed to the virtual register DMAR, and TIM automatically accumulates the internal offset address according to the access. The DBA bits in the register are used to specify the destination address of the first DMA transfer within TIM, while the DBL bits are used to specify the burst length.

16.6.24 Input XOR Function

The input XOR function allows the input signals from channels 1−3 to be XORed together and then connected to the input of the filtering and edge detection circuit of channel 1, which can

be used for input capture or triggering on channel 1. It supports single-edge interrupt trigger for the three-input XOR but does not support dual-edge interrupt trigger.

The TI1S bit in the TIM_CR2 register is used to select whether the input to channel 1 comes from the XOR of the three channel inputs.

16.6.25 Debug Mode

When the CPU enters debug mode, the timer can either stop or continue working, and its behavior is defined by registers in the chip system.

When the timer is stopped during debugging, its output will be disabled (MOE is cleared). Depending on the register configuration, the output signal can be forced to be inactive or controlled by the GPIO module.

16.7 Register Description

TIM0 register base address: 0x4700_4000

TIM7 register base address: 0x4700_5000

The registers are listed below:

Table 16-5: List of Registers Regarding Advanced-control Timers TIM0 & TIM7

Offset Address	Name	Description
0x00	TIM_CR1	Control register 1
0x04	TIM_CR2	Control register 2
0x08	TIM_SMCR	Slave mode control register
0x0C	TIM_DIER	DMA and interrupt enable register
0x10	TIM_SR	Status register
0x14	TIM_EGR	Event generation register
0x18	TIM_CCMR1	Capture/compare mode register 1
0x1C	TIM_CCMR2	Capture/compare register 2
0x20	TIM_CCER	Capture/compare enable register
0x24	TIM_CNT	Counter register
0x28	TIM_PSC	Prescaler register

Offset Address	Name	Description
0x2C	TIM_ARR	Auto-reload register
0x30	TIM_RCR	Repetition counter register
0x34	TIM_CCR1	Capture/compare register 1
0x38	TIM_CCR2	Capture/compare register 2
0x3C	TIM_CCR3	Capture/compare register 3
0x40	TIM_CCR4	Capture/compare register 4
0x44	TIM_BDTR	Break and dead-time control register
0x48	TIM_DCR	DMA control register
0x4C	TIM_DMAR	DMA access register

Registers are detailed in the following sections.

16.7.1 Control Register 1 (TIM_CR1)

Offset address: 0x00

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:10	RSV	-	-	Reserved
9:8	CKD	R/W	0x0	Dead time and digital filter clock frequency division register (division ratio relative to CK_INT): 00: $t_{DTS} = t_{CK\_INT}$ 01: $t_{DTS} = 2 * t_{CK\_INT}$ 10: $t_{DTS} = 4 * t_{CK\_INT}$ 11: reserved, prohibited
7	ARPE	R/W	0x0	Auto-reload preload enable: 0: ARR not preloaded 1: ARR preloaded
6:5	CMS	R/W	0x0	Counter alignment mode selection: 00: edge-aligned mode 01: center-aligned mode 1; output compare interrupt flags are set only when the counter is counting down. 10: center-aligned mode 2; output compare interrupt flags are set only when the counter is counting up. 11: center-aligned mode 3; output compare interrupt flags are set both when the counter is counting up or

Bit	Name	Attribute	Reset Value	Description
				down.
4	DIR	R/W	0x0	Counting direction register: 0: count up 1: count down Note: This register is read-only when the timer is configured in center-aligned mode or encoder mode.
3	OPM	R/W	0x0	One-pulse output mode: 0: the counter does not stop at the occurrence of update event. 1: the counter stops at the occurrence of update event (CEN cleared automatically).
2	URS	R/W	0x0	Update request source: 0: an update interrupt or DMA request will be generated by any of the following events: ● Counter overflow/underflow ● Software setting the UG bit ● Update generated from slave mode controller 1: an update interrupt or DMA request will be generated only at counter overflow or underflow.
1	UDIS	R/W	0x0	Update disable: 0: update event enabled The update event can be generated by any of the following events: ● Counter overflow/underflow ● Software setting the UG bit ● Update generated from slave mode controller 1: update event disabled, shadow register not updated The counter and the prescaler will be reinitialized if the UG bit is set or if the slave mode controller receives a hardware reset.
0	CEN	R/W	0x0	Counter enable: 0: disabled 1: enabled Note: The external trigger mode can automatically set the CEN bit.

16.7.2 Control Register 2 (TIM_CR2)

Offset address: 0x04

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:15	RSV	-	-	Reserved
14	OIS4	R/W	0x0	OC4 output idle state: 0: OC4 = 0 (after a dead-time if complementary output is enabled) when MOE = 0 1: OC4 = 1 (after a dead-time if complementary output is enabled) when MOE = 0
13	OIS3N	R/W	0x0	OC3N output idle state: 0: OC3N = 0 after a dead-time when MOE = 0 1: OC3N = 1 after a dead-time when MOE = 0 [For OC1−3N, (CCxP‖CCxNP) = 1 is required to make OCxN = 1.]
12	OIS3	R/W	0x0	OC3 output idle state: 0: OC3 = 0 (after a dead-time if complementary output is enabled) when MOE = 0 1: OC3 = 1 (after a dead-time if complementary output is enabled) when MOE = 0 [For OC1−3, (CCxP‖CCxNP) is required to make OCx = 1.]
11	OIS2N	R/W	0x0	OC2N output idle state: 0: OC2N = 0 after a dead-time when MOE = 0 1: OC2N = 1 after a dead-time when MOE = 0 [For OC1−3N, (CCxP‖CCxNP) = 1 is required to make OCxN = 1.]
10	OIS2	R/W	0x0	OC2 output idle state: 0: OC2 = 0 (after a dead-time if complementary output is enabled) when MOE = 0 1: OC2 = 1 (after a dead-time if complementary output is enabled) when MOE = 0 [For OC1−3, (CCxP‖CCxNP) is required to make OCx = 1.]
9	OIS1N	R/W	0x0	OC1N output idle state:

Bit	Name	Attribute	Reset Value	Description
				0: OC1N = 0 after a dead-time when MOE = 0 1: OC1N = 1 after a dead-time when MOE = 0 [For OC1−3N, (CCxP‖CCxNP) = 1 is required to make OCxN = 1.]
8	OIS1	R/W	0x0	OC1 output idle state: 0: OC1 = 0 (after a dead-time if complementary output is enabled) when MOE = 0 1: OC1 = 1 (after a dead-time if complementary output is enabled) when MOE = 0 [For OC1−3, (CCxP‖CCxNP) is required to make OCx = 1.]
7	TI1S	R/W	0x0	T1 input selection: 0: T1 input from CH1 pin 1: T1 input from XOR combination of CH1, CH2 and CH3 pins
6:4	MMS	R/W	0x0	Master mode selection, selecting the TRGO trigger mode: 000: reset—TRGO is generated by the UG bit in the EGR register. 001: enable—TRGO is generated by the counter enable signal, including the CEN control bit and external trigger. 010: update—TRGO is generated by the update event. 011: compare pulse—TRGO is generated when an input capture or compare event occurs that sets CC1F to 1. 100: compare—TRGO is generated by OC1REF. 101: compare—TRGO is generated by OC2REF. 110: compare—TRGO is generated by OC3REF. 111: compare—TRGO is generated by OC4REF.
3	CCDS	R/W	0x0	Capture/compare DMA selection: 0: CCx DMA request sent when CCx event occurs 1: CCx DMA requests sent when update event occurs
2	CCUS	R/W	0x0	Capture/compare control update selection, selecting the method by which the preloaded

Bit	Name	Attribute	Reset Value	Description
				capture/compare control bits are updated: 0: by setting the COMG bit to 1 only 1: by setting the COMG bit to 1 or when the TRGI signal occurs
1	RSV	-	-	Reserved
0	CCPC	R/W	0x0	Capture/compare preload control: 0: CCxE, CCxNE and OCxM bits are not preloaded. 1: CCxE, CCxNE and OCxM bits are preloaded, after having been written, they are updated only when a commutation event (COM) occurs. Note: This bit acts only on channels that have a complementary output.

16.7.3 Slave Mode Control Register (TIM_SMCR)

Offset address: 0x08

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:16	RSV	-	-	Reserved
15	ETP	R/W	0x0	External trigger polarity: 0: active at high level or rising edge 1: active at low level or falling edge
14	ECE	R/W	0x0	Enable clock enable: 0: external clock mode 2 disabled 1: external clock mode 2 enabled; the counter is clocked by any active edge on the ETRF signal.
13:12	ETPS	R/W	0x0	External trigger prescaler: The frequency of external trigger signal ETRP must be at most 1/4 of TIM clock frequency. A prescaler can be enabled to reduce ETRP frequency when inputting fast external clocks. 00: prescaler off 01: frequency divided by 2 10: frequency divided by 4 11: frequency divided by 8

Bit	Name	Attribute	Reset Value	Description
11:8	ETF	R/W	0x0	External trigger filter frequency and length selection: 0000: no filter 0001: $f_{SAMPLING} = f_{CK\_INT}$, N = 2 0010: $f_{SAMPLING} = f_{CK\_INT}$, N = 4 0011: $f_{SAMPLING} = f_{CK\_INT}$, N = 8 0100: $f_{SAMPLING} = f_{DTS}$ / 2, N = 6 0101: $f_{SAMPLING} = f_{DTS}$ / 2, N = 8 0110: $f_{SAMPLING} = f_{DTS}$ / 4, N = 6 0111: $f_{SAMPLING} = f_{DTS}$ / 4, N = 8 1000: $f_{SAMPLING} = f_{DTS}$ / 8, N = 6 1001: $f_{SAMPLING} = f_{DTS}$ / 8, N = 8 1010: $f_{SAMPLING} = f_{DTS}$ / 16, N = 5 1011: $f_{SAMPLING} = f_{DTS}$ / 16, N = 6 1100: $f_{SAMPLING} = f_{DTS}$ / 16, N = 8 1101: $f_{SAMPLING} = f_{DTS}$ / 32, N = 5 1110: $f_{SAMPLING} = f_{DTS}$ / 32, N = 6 1111: $f_{SAMPLING} = f_{DTS}$ / 32, N = 8
7	MSM	R/W	0x0	Master/slave mode selection: 0: no action 1: the effect of an event on the trigger input (TRGI) is delayed to allow a perfect synchronization between the current timer and its slaves (through TRGO).
6:4	TS	R/W	0x0	Trigger selection for selecting the trigger source to be used to synchronize the counter: 000: internal trigger 0 (ITR0) 001: internal trigger 1 (ITR1) 010: internal trigger 0 (ITR2) 011: internal trigger 0 (ITR3) 100: TI1 edge detector (TI1F_ED) 101: filtered timer input 1 (TI1FP1) 110: filtered timer input 2 (TI2FP2) 111: external trigger input (ETRF) Note: These bits can be changed only when the slave mode is disabled (i.e. SMS = 000).
3	RSV	-	-	Reserved

Bit	Name	Attribute	Reset Value	Description
2:0	SMS	R/W	0x0	Slave mode selection: 000: slave mode disabled—if CEN is enabled, then the prescaler is clocked directly by the internal clock. 001: encoder mode 1—counter counts up/down on TI2FP1 edge depending on TI1FP2 level. 010: encoder mode 2—counter counts up/down on TI1FP2 edge depending on TI2FP1 level. 011: encoder mode 3—counter counts up/down on both TI1FP1 and TI2FP2 edges depending on the level of other inputs. 100: reset mode—rising edge of the selected trigger input (TRGI) reinitializes the counter and generates an update of the registers. 101: gated mode—the counter clock is enabled when TRGI is high, and stops as soon as it becomes low. 110: trigger mode—the counter starts at the rising edge of TRGI (but it is not reset). 111: external clock mode 1—rising edges of TRGI directly clock the counter.

16.7.4 DMA / Interrupt Enable Register (TIM_DIER)

Offset address: 0x0C

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:20	RSV	-	-	Reserved
19	CC4OF_DISABLE	R/W	0x0	CC4OF interrupt enable: 0: enabled 1: disabled
18	CC3OF_DISABLE	R/W	0x0	CC3OF interrupt enable: 0: enabled 1: disabled
17	CC2OF_DISABLE	R/W	0x0	CC2OF interrupt enable:

Bit	Name	Attribute	Reset Value	Description
				0: enabled 1: disabled
16	CC1OF_DISABLE	R/W	0x0	CC1OF interrupt enable: 0: enabled 1: disabled
15	RSV	-	-	Reserved
14	TDE	R/W	0x0	External trigger DMA request enable: 0: in slave mode, external trigger DMA request disabled 1: in slave mode, external trigger DMA request enabled (can be used to automatically update the preload register)
13	COMDE	R/W	0x0	COM DMA request enable: 0: COM DMA request disabled 1: COM DMA request enabled
12	CC4DE	R/W	0x0	Capture/compare channel 4 DMA request enable: 0: CC4 DMA request disabled 1: CC4 DMA request enabled
11	CC3DE	R/W	0x0	Capture/compare channel 3 DMA request enable: 0: CC3 DMA request disabled 1: CC3 DMA request enabled
10	CC2DE	R/W	0x0	Capture/compare channel 2 DMA request enable: 0: CC2 DMA request disabled 1: CC2 DMA request enabled
9	CC1DE	R/W	0x0	Capture/compare channel 1 DMA request enable: 0: CC1 DMA request disabled 1: CC1 DMA request enabled
8	UDE	R/W	0x0	Update DMA request enable: 0: update DMA request disabled 1: update DMA request enabled
7	BIE	R/W	0x0	Break interrupt enable: 0: break interrupt disabled

Bit	Name	Attribute	Reset Value	Description
				1: break interrupt enabled
6	TIE	R/W	0x0	Trigger interrupt enable: 0: trigger interrupt disabled 1: trigger interrupt enabled
5	COMIE	R/W	0x0	COM interrupt enable: 0: COM interrupt disabled 1: COM interrupt enabled
4	CC4IE	R/W	0x0	Capture/compare channel 4 interrupt enable: 0: CC4 interrupt disabled 1: CC4 interrupt enabled
3	CC3IE	R/W	0x0	Capture/compare channel 3 interrupt enable: 0: CC3 interrupt disabled 1: CC3 interrupt enabled
2	CC2IE	R/W	0x0	Capture/compare channel 2 interrupt enable: 0: CC2 interrupt disabled 1: CC2 interrupt enabled
1	CC1IE	R/W	0x0	Capture/compare channel 1 interrupt enable: 0: CC1 interrupt disabled 1: CC1 interrupt enabled
0	UIE	R/W	0x0	Update interrupt enable: 0: update interrupt disabled 1: update interrupt enabled

16.7.5 Status Register (TIM_SR)

Offset address: 0x10

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:13	RSV	-	-	Reserved
12	CC4OF	R/W0C	0x0	Capture/compare channel 4 overcapture flag: This bit is valid only when the corresponding channel is configured in input capture mode. This flag bit is set by hardware and cleared by software via writing it to 0. 0: no overcapture has been detected.

Bit	Name	Attribute	Reset Value	Description
				1: a new capture occurs while CC4IF flag is 1.
11	CC3OF	R/W0C	0x0	Capture/compare channel 3 overcapture flag: This flag is set by hardware only when the corresponding channel is configured in input capture mode. This flag bit is set by hardware and cleared by software via writing it to 0. 0: no overcapture has been detected. 1: a new capture occurs while CC3IF flag is 1.
10	CC2OF	R/W0C	0x0	Capture/compare channel 2 overcapture flag: This flag is set by hardware only when the corresponding channel is configured in input capture mode. This flag bit is set by hardware and cleared by software via writing it to 0. 0: no overcapture has been detected. 1: a new capture occurs while CC2IF flag is 1.
9	CC1OF	R/W0C	0x0	Capture/compare channel 1 overcapture flag: This flag is set by hardware only when the corresponding channel is configured in input capture mode. This flag bit is set by hardware and cleared by software via writing it to 0. 0: no overcapture has been detected. 1: a new capture occurs while CC1IF flag is 1.
8	RSV	-	-	Reserved
7	BIF	R/W0C	0x0	Break interrupt flag is set by hardware and cleared by software via writing it to 0.
6	TIF	R/W0C	0x0	Trigger interrupt flag is set by hardware and cleared by software via writing it to 0.
5	COMIF	R/W0C	0x0	COM interrupt flag is set by hardware and cleared by software via writing it to 0.
4	CC4IF	R/W0C	0x0	Capture/compare channel 4 interrupt flag: If channel CC4 is configured as output: the CC4IF flag is set when the counter matches the compare value. It is cleared by software via writing it to 0. If channel CC4 is configured as input: this flag is set by hardware on a capture. It is cleared by software via writing it to 0 or automatically cleared by software

Bit	Name	Attribute	Reset Value	Description
				reading TIM_CCR4.
3	CC3IF	R/W0C	0x0	Capture/compare channel 3 interrupt flag: If channel CC3 is configured as output: the CC3IF flag is set when the counter matches the compare value. It is cleared by software via writing it to 0. If channel CC3 is configured as input: this flag is set by hardware on a capture. It is cleared by software via writing it to 0 or automatically cleared by software reading TIM_CCR3.
2	CC2IF	R/W0C	0x0	Capture/compare channel 2 interrupt flag: If channel CC2 is configured as output: the CC2IF flag is set when the counter matches the compare value. It is cleared by software via writing it to 0. If channel CC2 is configured as input: this flag is set by hardware on a capture. It is cleared by software via writing it to 0 or automatically cleared by software reading TIM_CCR2.
1	CC1IF	R/W0C	0x0	Capture/compare channel 1 interrupt flag: If channel CC1 is configured as output: the CC1IF flag is set when the counter matches the compare value. It is cleared by software via writing it to 0. If channel CC1 is configured as input: this flag is set by hardware on a capture. It is cleared by software via writing it to 0 or automatically cleared by software reading TIM_CCR1.
0	UIF	R/W0C	0x0	Update interrupt flag is set by hardware and cleared by software via writing it to 0. UIF is set and the shadow register is updated at the following events: ● Counter overflow occurs if repetition counter = 0 and UDIS = 0. ● The counter is reinitialized by software setting the UG bit if URS = 0 and UDIS = 0. ● The counter is reinitialized by a trigger event if URS = 0 and UDIS = 0.

16.7.6 Event Generation Register (TIM_EGR)

Offset address: 0x14

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:8	RSV	-	-	Reserved
7	BG	W	0x0	This bit is set by software to generate a break event, and it is automatically cleared by hardware.
6	TG	W	0x0	This bit is set by software to generate a trigger event, and it is automatically cleared by hardware.
5	COMG	W	0x0	This flag bit is set by software and cleared by hardware via writing it to 1.
4	CC4G	W	0x0	Capture/compare channel 4 generation If channel CC4 is configured as output, CC4IF flag is set, and corresponding interrupt and DMA request will be sent if enabled. If channel CC4 is configured as input, the current counter value is captured in TIM_CCR4 register, the CC4IF flag is set, and corresponding interrupt and DMA request will be sent if enabled.
3	CC3G	W	0x0	Capture/compare channel 3 generation If channel CC3 is configured as output, CC3IF flag is set, and corresponding interrupt and DMA request will be sent if enabled. If channel CC3 is configured as input, the current counter value is captured in TIM_CCR3 register, the CC3IF flag is set, and corresponding interrupt and DMA request will be sent if enabled.
2	CC2G	W	0x0	Capture/compare channel 2 generation If channel CC2 is configured as output, CC2IF flag is set, and corresponding interrupt and DMA request will be sent if enabled. If channel CC2 is configured as input, the

Bit	Name	Attribute	Reset Value	Description
				current counter value is captured in TIM_CCR2 register, the CC2IF flag is set, and corresponding interrupt and DMA request will be sent if enabled.
1	CC1G	W	0x0	Capture/compare channel 1 generation If channel CC1 is configured as output, CC1IF flag is set, and corresponding interrupt and DMA request will be sent if enabled. If channel CC1 is configured as input, the current counter value is captured in TIM_CCR1 register, the CC1IF flag is set, and corresponding interrupt and DMA request will be sent if enabled.
0	UG	W	0x0	Update generation: This bit can be set by software to generate an update event, and is automatically cleared by hardware. When the UG bit is set by software, the counter is reinitialized, the shadow register is updated, and the prescaler counter is cleared.

16.7.7 Capture/Compare Mode Register 1 (TIM_CCMR1)

Offset address: 0x18

Reset value: 0x0000 0000

Note: This register can be used for output compare mode or for input capture mode.

- Output compare mode

Bit	Name	Attribute	Reset Value	Description
31:25	RSV	-	-	Reserved
24	OC2M3	R/W	0x0	Form the 4-bit OC2M with bits 14:12
23:17	RSV	-	-	Reserved
16	OC1M3	R/W	0x0	Form the 4-bit OC1M with bits 6:4
15	OC2CE	R/W	0x0	Output compare 2 clear enable: 0: OC2REF is not affected by ETRF input. 1: OC2REF is automatically cleared once a high level is detected on ETRF input.

Bit	Name	Attribute	Reset Value	Description
14:12	OC2M2_0	R/W	0x0	Output compare 2 mode configuration These bits define the behavior of the output reference signal OC2REF. 0000: the comparison between the output compare register CCR2 and the counter CNT has no effect on the outputs. 0001: set OC2REF high when CCR2 = CNT (falling edge) 0010: set OC2REF low when CCR2 = CNT (falling edge) 0011: toggle OC2REF when CCR2 = CNT (falling edge) 0100: force OC2REF low (inactive) 0101: force OC2REF high (active) 0110: PWM mode 1 In up-counting, OC2REF is set high when CNT < CCR2, otherwise it is set low. In down-counting, OC2REF is set low when CNT ≥ CCR2, otherwise it is set high. 0111: PWM mode 2 In up-counting, OC2REF is set low when CNT < CCR2, otherwise it is set high. In down-counting, OC2REF is set high when CNT ≥ CCR2, otherwise it is set low. 1100: combined PWM mode 1—OC2REF has the same behavior as in PWM mode 1, and OC2REFC is the logical OR between OC2REF and OC1REF. 1101: combined PWM mode 2—OC2REF has the same behavior as in PWM mode 2, and OC2REFC is the logical AND between OC2REF and OC1REF. 1110: asymmetric PWM mode 1—OC2REF has the same behavior as in PWM mode 1. OC2REFC outputs OC2REF when the counter is counting up, and OC1REF when it is counting down.

Bit	Name	Attribute	Reset Value	Description
				1111: asymmetric PWM mode 2—OC2REF has the same behavior as in PWM mode 2. OC2REFC outputs OC2REF when the counter is counting up, and OC1REF when it is counting down.
11	OC2PE	R/W	0x0	Output compare 2 preload enable: 0: preload register on CCR2 disabled; CCR2 can be written directly. 1: preload register on CCR2 enabled; read/write operations access the preload register; the preload value is shifted to the shadow register at each update event.
10	OC2FE	R/W	0x0	Output compare 2 fast enable: 0: fast disabled, the trigger input will not affect the comparison output. 1: fast enabled, the trigger input will immediately change OC2REF to the output when the comparison values match, regardless of the actual current comparison. This function acts only if the channel is configured in PWM1 or PWM2 mode.
9:8	CC2S	R/W	0x0	Capture/compare channel 2 selection: 00: CC2 channel is configured as output. 01: CC2 channel is configured as input, IC2 is mapped on TI2. 10: CC2 channel is configured as input, IC2 is mapped on TI1. 11: CC2 channel is configured as input, IC2 is mapped on TRC. Note: CC2S bits are writable only when the channel is OFF (CC2E = 0).
7	OC1CE	R/W	0x0	Output compare 1 clear enable: 0: OC1REF is not affected by ETRF input. 1: OC1REF is automatically cleared once a high level is detected on ETRF input.
6:4	OC1M	R/W	0x0	Output compare 1 mode: these bits define the behavior of the output reference signal

Bit	Name	Attribute	Reset Value	Description
				OC1REF. 0000: the comparison between the output compare register CCR1 and the counter CNT has no effect on the outputs. 0001: set OC1REF high when CCR1 = CNT (falling edge) 0010: set OC1REF low when CCR1 = CNT (falling edge) 0011: toggle OC1REF when CCR1 = CNT (falling edge) 0100: force OC1REF low (inactive) 0101: force OC1REF high (active) 0110: PWM mode 1 In up-counting, OC1REF is set high when CNT < CCR1, otherwise it is set low. In down-counting, OC1REF is set low when CNT ≥ CCR1, otherwise it is set high. 0111: PWM mode 2 In up-counting, OC1REF is set low when CNT < CCR1, otherwise it is set high. In down-counting, OC1REF is set high when CNT ≥ CCR1, otherwise it is set low. 1100: combined PWM mode 1—OC1REF has the same behavior as in PWM mode 1, and OC1REFC is the logical OR between OC1REF and OC2REF. 1101: combined PWM mode 2—OC1REF has the same behavior as in PWM mode 2, and OC1REFC is the logical AND between OC1REF and OC2REF. 1110: asymmetric PWM mode 1—OC1REF has the same behavior as in PWM mode 1. OC1REFC outputs OC1REF when the counter is counting up, and OC2REF when it is counting down. 1111: asymmetric PWM mode 2—OC1REF has the same behavior as in PWM mode 2.

Bit	Name	Attribute	Reset Value	Description
				OC1REFC outputs OC1REF when the counter is counting up, and OC2REF when it is counting down.
3	OC1PE	R/W	0x0	Output compare 1 preload enable: 0: preload register disabled; CCR1 can be written directly. 1: preload register on CCR1 enabled; read/write operations access the preload register; the preload value is shifted to the shadow register at each update event.
2	OC1FE	R/W	0x0	Output compare 1 fast enable: 0: fast disabled, the trigger input will not affect the comparison output. 1: fast enabled, the trigger input will immediately change OC1REF to the output when the comparison values match, regardless of the actual current comparison. This function acts only if the channel is configured in PWM1 or PWM2 mode.
1:0	CC1S	R/W	0x0	Capture/compare channel 1 selection: 00: CC1 channel is configured as output. 01: CC1 channel is configured as input, IC1 is mapped on TI1. 10: CC1 channel is configured as input, IC1 is mapped on TI2. 11: CC1 channel is configured as input, IC1 is mapped on TRC. Note: CC1S bits are writable only when the channel is OFF (CC1E = 0).

- Input Capture Mode

Bit	Name	Attribute	Reset Value	Description
31:16	RSV	-	-	Reserved
15:12	IC2F	R/W	0x0	Input capture 2 filter
11:10	IC2PSC	R/W	0x0	Input capture 2 prescaler
9:8	CC2S	R/W	0x0	Capture/compare channel 2 selection:

Bit	Name	Attribute	Reset Value	Description
				00: CC2 channel is configured as output. 01: CC2 channel is configured as input, IC2 is mapped on TI2. 10: CC2 channel is configured as input, IC2 is mapped on TI1. 11: CC2 channel is configured as input, IC2 is mapped on TRC. Note: CC2S bits are writable only when the channel is OFF (CC2E = 0).
7:4	IC1F	R/W	0x0	Input capture 1 filter: This bit-field defines the frequency used to sample TI1 input and the length of the digital filter applied to TI1. 0000: no filter, sampling is done at f_{DTS} 0001: $f_{SAMPLING} = f_{CK\_INT}$, N = 2 0010: $f_{SAMPLING} = f_{CK\_INT}$, N = 4 0011: $f_{SAMPLING} = f_{CK\_INT}$, N = 8 0100: $f_{SAMPLING} = f_{DTS} / 2$, N = 6 0101: $f_{SAMPLING} = f_{DTS} / 2$, N = 8 0110: $f_{SAMPLING} = f_{DTS} / 4$, N = 6 0111: $f_{SAMPLING} = f_{DTS} / 4$, N = 8 1000: $f_{SAMPLING} = f_{DTS} / 8$, N = 6 1001: $f_{SAMPLING} = f_{DTS} / 8$, N = 8 1010: $f_{SAMPLING} = f_{DTS} / 16$, N = 5 1011: $f_{SAMPLING} = f_{DTS} / 16$, N = 6 1100: $f_{SAMPLING} = f_{DTS} / 16$, N = 8 1101: $f_{SAMPLING} = f_{DTS} / 32$, N = 5 1110: $f_{SAMPLING} = f_{DTS} / 32$, N = 6 1111: $f_{SAMPLING} = f_{DTS} / 32$, N = 8
3:2	IC1PSC	R/W	0x0	Input capture 1 prescaler: 00: no prescaler 01: capture is done once every 2 events 10: capture is done once every 4 events 11: capture is done once every 8 events The IC1PSC register is reset when CC1E = 0.
1:0	CC1S	R/W	0x0	Capture/compare channel 1 selection: 00: CC1 channel is configured as output.

Bit	Name	Attribute	Reset Value	Description
				01: CC1 channel is configured as input, IC1 is mapped on TI1. 10: CC1 channel is configured as input, IC1 is mapped on TI2. 11: CC1 channel is configured as input, IC1 is mapped on TRC. Note: CC1S bits are writable only when the channel is OFF (CC1E = 0).

16.7.8 Capture/Compare Mode Register 2 (TIM_CCMR2)

Offset address: 0x1C

Reset value: 0x0000 0000

Note: This register can be used for output compare mode or for input capture mode.

- Output compare mode

Bit	Name	Attribute	Reset Value	Description
31:25	RSV	-	-	Reserved
24	OC4M3	R/W	0x0	Form the 4-bit OC4M[3:0] with bits [14:12]
23:17	RSV	-	-	Reserved
16	OC3M3	R/W	0x0	Form the 4-bit OC3M[3:0] with bits [6:4]
15	OC4CE	R/W	0x0	Output compare 4 clear enable: 0: OC4REF is not affected by ETRF input. 1: OC4REF is automatically cleared once a high level is detected on ETRF input.
14:12	OC4M2_0	R/W	0x0	Output compare 4 mode: these bits define the behavior of the output reference signal OC4REF. 0000: the comparison between the output compare register CCR4 and the counter CNT has no effect on the outputs. 0001: set OC4REF high when CCR4 = CNT. 0010: set OC4REF low when CCR4 = CNT. 0011: toggle OC4REF when CCR4 = CNT. 0100: force OC4REF low (inactive)

Bit	Name	Attribute	Reset Value	Description
				0101: force OC4REF high (active) 0110: PWM mode 1 In up-counting, OC4REF is set high when CNT < CCR4, otherwise it is set low. In down-counting, OC4REF is set low when CNT > CCR4, otherwise it is set high. 0111: PWM mode 2 In up-counting, OC4REF is set low when CNT < CCR4, otherwise it is set high. In down-counting, OC4REF is set high when CNT > CCR4, otherwise it is set low. 1100: combined PWM mode 1—OC4REF has the same behavior as in PWM mode 1, and OC4REFC is the logical OR between OC4REF and OC3REF. 1101: combined PWM mode 2—OC4REF has the same behavior as in PWM mode 2, and OC4REFC is the logical AND between OC4REF and OC3REF. 1110: asymmetric PWM mode 1—OC4REF has the same behavior as in PWM mode 1. OC4REFC outputs OC4REF when the counter is counting up, and OC3REF when it is counting down. 1111: asymmetric PWM mode 2—OC4REF has the same behavior as in PWM mode 2. OC4REFC outputs OC4REF when the counter is counting up, and OC3REF when it is counting down.
11	OC4PE	R/W	0x0	Output compare 4 preload enable: 0: preload register disabled; CCR4 can be written directly. 1: preload register on CCR4 enabled; read/write operations access the preload register; the preload value is shifted to the shadow register at each update event.

Bit	Name	Attribute	Reset Value	Description
10	OC4FE	R/W	0x0	Output compare 4 fast enable: 0: fast disabled, the trigger input will not affect the comparison output. 1: fast enabled, the trigger input will immediately change OC4REF to the output when the comparison values match, regardless of the actual current comparison. This function acts only if the channel is configured in PWM1 or PWM2 mode.
9:8	CC4S	R/W	0x0	Capture/compare channel 4 selection: 00: CC4 channel is configured as output. 01: CC4 channel is configured as input, IC4 is mapped on TI4. 10: CC4 channel is configured as input, IC4 is mapped on TI3. 11: CC4 channel is configured as input, IC4 is mapped on TRC. Note: CC4S bits are writable only when the channel is OFF (CC4E = 0).
7	OC3CE	R/W	0x0	Output compare 3 clear enable: 0: OC3REF is not affected by ETRF input. 1: OC3REF is automatically cleared once a high level is detected on ETRF input.
6:4	OC3M2_0	R/W	0x0	Output compare 3 mode: these bits define the behavior of the output reference signal OC3REF. 0000: the comparison between the output compare register CCR3 and the counter CNT has no effect on the outputs. 0001: set OC3REF high when CCR3 = CNT. 0010: set OC3REF low when CCR3 = CNT. 0011: toggle OC3REF when CCR3 = CNT. 0100: force OC3REF low (inactive) 0101: force OC3REF high (active) 0110: PWM mode 1 In up-counting, OC3REF is set high when CNT <

Bit	Name	Attribute	Reset Value	Description
				CCR3, otherwise it is set low. In down-counting, OC3REF is set low when CNT > CCR3, otherwise it is set high. 0111: PWM mode 2 In up-counting, OC3REF is set low when CNT < CCR3, otherwise it is set high. In down-counting, OC3REF is set high when CNT > CCR3, otherwise it is set low. 1100: combined PWM mode 1—OC3REF has the same behavior as in PWM mode 1, and OC3REFC is the logical OR between OC3REF and OC4REF. 1101: combined PWM mode 2—OC3REF has the same behavior as in PWM mode 2, and OC3REFC is the logical AND between OC3REF and OC4REF. 1110: asymmetric PWM mode 1—OC3REF has the same behavior as in PWM mode 1. OC3REFC outputs OC3REF when the counter is counting up, and OC4REF when it is counting down. 1111: asymmetric PWM mode 2—OC3REF has the same behavior as in PWM mode 2. OC3REFC outputs OC3REF when the counter is counting up, and OC4REF when it is counting down.
3	OC3PE	R/W	0x0	Output compare 3 preload enable: 0: preload register disabled; CCR3 can be written directly. 1: preload register on CCR3 enabled; read/write operations access the preload register; the preload value is shifted to the shadow register at each update event.
2	OC3FE	R/W	0x0	Output compare 3 fast enable: 0: fast disabled, the trigger input will not affect the comparison output.

Bit	Name	Attribute	Reset Value	Description
				1: fast enabled, the trigger input will immediately change OC3REF to the output when the comparison values match, regardless of the actual current comparison. This function acts only if the channel is configured in PWM1 or PWM2 mode.
1:0	CC3S	R/W	0x0	Capture/compare channel 3 selection: 00: CC3 channel is configured as output. 01: CC3 channel is configured as input, IC3 is mapped on TI3. 10: CC3 channel is configured as input, IC3 is mapped on TI4. 11: CC3 channel is configured as input, IC3 is mapped on TRC. Note: CC3S bits are writable only when the channel is OFF (CC3E = 0).

- Input Capture Mode

Bit	Name	Attribute	Reset Value	Description
31:16	RSV	-	-	Reserved
5:12	IC4F	R/W	0x0	Input capture 4 filter: this bit-field defines the frequency used to sample TI4 input and the length of the digital filter applied to TI4: 0000: no filter, sampling is done at f_{DTS} 0001: $f_{SAMPLING} = f_{CK\_INT}$, N = 2 0010: $f_{SAMPLING} = f_{CK\_INT}$, N = 4 0011: $f_{SAMPLING} = f_{CK\_INT}$, N = 8 0100: $f_{SAMPLING} = f_{DTS} / 2$, N = 6 0101: $f_{SAMPLING} = f_{DTS} / 2$, N = 8 0110: $f_{SAMPLING} = f_{DTS} / 4$, N = 6 0111: $f_{SAMPLING} = f_{DTS} / 4$, N = 8 1000: $f_{SAMPLING} = f_{DTS} / 8$, N = 6 1001: $f_{SAMPLING} = f_{DTS} / 8$, N = 8 1010: $f_{SAMPLING} = f_{DTS} / 16$, N = 5 1011: $f_{SAMPLING} = f_{DTS} / 16$, N = 6 1100: $f_{SAMPLING} = f_{DTS} / 16$, N = 8

Bit	Name	Attribute	Reset Value	Description
				1101: $f_{SAMPLING} = f_{DTS} / 32$, N = 5 1110: $f_{SAMPLING} = f_{DTS} / 32$, N = 6 1111: $f_{SAMPLING} = f_{DTS} / 32$, N = 8
11:10	IC4PSC	R/W	0x0	Input capture 4 prescaler: 00: no prescaler 01: capture is done once every 2 events 10: capture is done once every 4 events 11: capture is done once every 8 events The IC4PSC register is reset when CC4E = 0.
9:8	CC4S	R/W	0x0	Capture/compare channel 4 selection: 00: CC4 channel is configured as output. 01: CC4 channel is configured as input, IC4 is mapped on TI4. 10: CC4 channel is configured as input, IC4 is mapped on TI3. 11: CC4 channel is configured as input, IC4 is mapped on TRC. Note: CC4S bits are writable only when the channel is OFF (CC4E = 0).
7:4	IC3F	R/W	0x0	Input capture 3 filter: this bit-field defines the frequency used to sample TI3 input and the length of the digital filter applied to TI3: 0000: no filter, sampling is done at f_{DTS} 0001: $f_{SAMPLING} = f_{CK\_INT}$, N = 2 0010: $f_{SAMPLING} = f_{CK\_INT}$, N = 4 0011: $f_{SAMPLING} = f_{CK\_INT}$, N = 8 0100: $f_{SAMPLING} = f_{DTS} / 2$, N = 6 0101: $f_{SAMPLING} = f_{DTS} / 2$, N = 8 0110: $f_{SAMPLING} = f_{DTS} / 4$, N = 6 0111: $f_{SAMPLING} = f_{DTS} / 4$, N = 8 1000: $f_{SAMPLING} = f_{DTS} / 8$, N = 6 1001: $f_{SAMPLING} = f_{DTS} / 8$, N = 8 1010: $f_{SAMPLING} = f_{DTS} / 16$, N = 5 1011: $f_{SAMPLING} = f_{DTS} / 16$, N = 6 1100: $f_{SAMPLING} = f_{DTS} / 16$, N = 8 1101: $f_{SAMPLING} = f_{DTS} / 32$, N = 5 1110: $f_{SAMPLING} = f_{DTS} / 32$, N = 6

Bit	Name	Attribute	Reset Value	Description
				1111: $f_{SAMPLING} = f_{DTS} / 32$, N = 8
3:2	IC3PSC	R/W	0x0	Input capture 3 prescaler: 00: no prescaler 01: capture is done once every 2 events 10: capture is done once every 4 events 11: capture is done once every 8 events The IC3PSC register is reset when CC3E = 0.
1:0	CC3S	R/W	0x0	Capture/compare channel 3 selection: 00: CC3 channel is configured as output. 01: CC3 channel is configured as input, IC3 is mapped on TI3. 10: CC3 channel is configured as input, IC3 is mapped on TI4. 11: CC3 channel is configured as input, IC3 is mapped on TRC. Note: CC1S bits are writable only when the channel is OFF (CC1E = 0).

16.7.9 Capture/Compare Enable Register (TIM_CCER)

Offset address: 0x20

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:14	RSV	-	-	Reserved
13	CC4P	R/W	0x0	Capture/compare 4 output polarity: **CC4 channel configured as output:** 0: OC4 is OC4REF. 1: OC4 is the inverse OC4REF. **CC4 channel configured as input:** 0: non-inverted mode: capture is done on the rising edge of IC4. 1: inverted mode: capture is done on the falling edge of IC4.
12	CC4E	R/W	0x0	Capture/compare 4 output enable: CC4 channel configured as output: 0: OC4 not active

Bit	Name	Attribute	Reset Value	Description
				1: OC4 active CC4 channel configured as input: 0: capture disabled 1: capture enabled
11	CC3NP	R/W	0x0	Capture/compare 3 complementary output polarity: **CC3 channel configured as output:** 0: OC3N is the inverse OC3REF. 1: OC3N is OC3REF. **CC3 channel configured as input:** Used in conjunction with CC3P
10	CC3NE	R/W	0x0	Capture/compare 3 complementary output enable: 0: off—OC3N is not active. 1: on—OC3N signal is output.
9	CC3P	R/W	0x0	Capture/compare 3 output polarity: **CC3 channel configured as output:** 0: OC3 is OC3REF. 1: OC3 is the inverse OC3REF. **CC3 channel configured as input:** In conjunction with CC3NP, {CC3NP,CC3P} bits select the polarity of IC3: 00: non-inverted mode: capture is done on the rising edge of IC3. 01: inverted mode: capture is done on the falling edge of IC3. 10: reserved 11: non-inverted mode: capture is done on both the rising edge and falling edge of IC3. This configuration must not be used for encoder mode.
8	CC3E	R/W	0x0	Capture/compare 3 output enable: refer to CC1E description
7	CC2NP	R/W	0x0	Capture/compare 2 complementary output polarity: **CC2 channel configured as output:** 0: OC2N is the inverse OC2REF.

Bit	Name	Attribute	Reset Value	Description
				1: OC2N is OC2REF. **CC2 channel configured as input:** Used in conjunction with CC2P
6	CC2NE	R/W	0x0	Capture/compare 2 complementary output enable: 0: off—OC2N is not active. 1: on—OC2N signal is output.
5	CC2P	R/W	0x0	Capture/compare 2 output polarity: **CC2 channel configured as output:** 0: OC2 is OC2REF. 1: OC2 is the inverse OC2REF. **CC2 channel configured as input:** In conjunction with CC2NP, {CC2NP,CC2P} bits select the polarity of IC2: 00: non-inverted mode: capture is done on the rising edge of IC2. 01: inverted mode: capture is done on the falling edge of IC2. 10: reserved 11: non-inverted mode: capture is done on both the rising edge and falling edge of IC2. This configuration must not be used for encoder mode.
4	CC2E	R/W	0x0	Capture/compare 2 output enable: **CC2 channel configured as output:** 0: OC2 not active 1: OC2 active **CC2 channel configured as input:** 0: capture disabled 1: capture enabled
3	CC1NP	R/W	0x0	Capture/compare 1 complementary output polarity: **CC1 channel configured as output:** 0: OC1N is the inverse OC1REF. 1: OC1N is OC1REF. **CC1 channel configured as input:** In conjunction with CC1P, select the polarity of

Bit	Name	Attribute	Reset Value	Description
				IC2.
2	CC1NE	R/W	0x0	Capture/compare 1 complementary output enable: 0: off—OC1N is not active. 1: on—OC1N signal is output.
1	CC1P	R/W	0x0	Capture/compare 1 output polarity: **CC1 channel configured as output:** 0: OC1 is OC1REF. 1: OC1 is the inverse OC1REF. **CC1 channel configured as input:** In conjunction with CC1NP, {CC1NP,CC1P} bits select the polarity of IC1: 00: non-inverted mode: capture is done on the rising edge of IC1. 01: inverted mode: capture is done on the falling edge of IC1. 10: reserved 11: non-inverted mode: capture is done on both the rising edge and falling edge of IC1. This configuration must not be used for encoder mode.
0	CC1E	R/W	0x0	Capture/compare 1 output enable: **CC1 channel configured as output:** 0: OC1 not active 1: OC1 active **CC1 channel configured as input:** 0: capture disabled 1: capture enabled

As shown in the table below, MOE is the total output enable bit of the timer, OSSI is the off_state selection bit in IDLE state (MOE = 0), and OSSR is the off_state selection bit in RUN state (MOE = 1).

Table 16-6: State Correspondence between Control Register and Complementary Output Channel

Control Register					Output State	
MOE	**OSSI**	**OSSR**	**CCxE**	**CCxNE**	**OCx Output State**	**OCxN Output State**
1	X	0	0	0	Output disabled (not driven by TIM) OCx = 0, OCx_EN = 0	Output disabled (not driven by TIM) OCxN = 0, OCxN_EN = 0
		0	0	1	Output disabled (not driven by TIM) OCx = 0, OCx_EN = 0	OCxREF XOR CCxNP, OCxN_EN = 1
		0	1	0	OCxREF XOR CCxP, OCx_EN = 1	Output disabled (not driven by TIM) OCxN = 0, OCxN_EN = 0
		0	1	1	OCREF + polarity + dead-time OCx_EN = 1	Complementary to OCREF + polarity + dead-time OCxN_EN = 1
		1	0	0	Output disabled (not driven by TIM) OCx = CCxP, OCx_EN = 0	Output disabled (not driven by TIM) OCxN = CCxNP, OCxN_EN = 0
		1	0	1	Off-state (output enabled with inactive state) OCx = CCxP, OCx_EN = 1	OCxREF XOR CCxNP, OCxN_EN = 1
		1	1	0	OCxREF XOR CCxP OCx_EN = 1	Off-state (output enabled with inactive state) OCxN = CCxNP, OCxN_EN = 1
		1	1	1	OCREF + polarity + dead-time OCx_EN = 1	Complementary to OCREF + polarity + dead-time OCxN_EN = 1

<table>
<tr><th colspan="5">Control Register</th><th colspan="2">Output State</th></tr>
<tr><th>MOE</th><th>OSSI</th><th>OSSR</th><th>CCxE</th><th>CCxNE</th><th>OCx Output State</th><th>OCxN Output State</th></tr>
<tr><td rowspan="6">0</td><td>0</td><td rowspan="6">X</td><td>0</td><td>0</td><td>Output disabled (not driven by TIM)
OCx = CCxP,
OCx_EN = 0</td><td>Output disabled (not driven by TIM)
OCxN = CCxNP,
OCxN_EN = 0</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td colspan="2" rowspan="3">Output disabled (not driven by TIM)
Asynchronously: OCx = CCxP, OCx_EN = 0, OCxN = CCxNP, OCxN_EN = 0
If the clock is present: OCx = OISx, OCxN = OISxN after a dead-time.</td></tr>
<tr><td>0</td><td>1</td><td>0</td></tr>
<tr><td>0</td><td>1</td><td>1</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>Output disabled (not driven by TIM)
OCx = CCxP,
OCx_EN = 0</td><td>Output disabled (not driven by TIM)
OCxN = CCxNP,
OCxN_EN = 0</td></tr>
<tr><td rowspan="3">1</td><td>0</td><td>1</td><td colspan="2" rowspan="3">Off-state (output enabled with inactive state)
Asynchronously: OCx = CCxP, OCx_EN = 1, OCxN = CCxNP, OCxN_EN = 1
If the clock is present: OCx = OISx, OCxN = OISxN after a dead-time.</td></tr>
<tr><td>1</td><td>0</td></tr>
<tr><td>1</td><td>1</td></tr>
</table>

16.7.10 Counter Register (TIM_CNT)

Offset address: 0x24

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:16	RSV	-	-	Reserved
15:0	CNT	R/W	0x0	Counter value

16.7.11 Prescaler Register (TIM_PSC)

Offset address: 0x28

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:16	RSV	-	-	Reserved

Bit	Name	Attribute	Reset Value	Description
15:0	PSC	R/W	0x0	Counter clock (CK_CNT) prescaler value: $f_{CK\_CNT} = f_{CK\_PSC} / (PSC[15:0] + 1)$ This is a preload register whose content are transferred into the shadow register at each update event.

16.7.12 Auto-reload Register (TIM_ARR)

Offset address: 0x2C

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:16	RSV	-	-	Reserved
15:0	ARR	R/W	0x0	Auto-reload value at counter overflow: This is a preload register whose content are transferred into the shadow register at each update event.

16.7.13 Repetition Counter Register (TIM_RCR)

Offset address: 0x30

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:8	RSV	-	-	Reserved
7:0	REP	R/W	0x0	Repetition counter value If REP is not 0, the repetition counter counts down at each update condition, and an update event is triggered once REP = 0.

16.7.14 Capture/Compare Register 1 (TIM_CCR1)

Offset address: 0x34

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:16	RSV	-	-	Reserved
15:0	CCR1	R/W	0x0	Capture/compare channel 1 register

Bit	Name	Attribute	Reset Value	Description
				If channel CC1 is configured as output: This is a preload register containing the value to be compared to the counter and signaled on OC1 output. **If channel CC1 is configured as input:** CCR1 is the counter value transferred by the last input capture event, at this point the CCR1 register is read-only.

16.7.15 Capture/Compare Register 2 (TIM_CCR2)

Offset address: 0x38

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:16	RSV	-	-	Reserved
15:0	CCR2	R/W	0x0	Capture/compare channel 2 register: **If channel CC2 is configured as output:** This is a preload register containing the value to be compared to the counter and signaled on OC2 output. **If channel CC2 is configured as input:** CCR2 is the counter value transferred by the last input capture event, at this point the CCR2 register is read-only.

16.7.16 Capture/Compare Register 3 (TIM_CCR3)

Offset address: 0x3C

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:16	RSV	-	-	Reserved
15:0	CCR3	R/W	0x0	Capture/compare channel 3 register: **If channel CC3 is configured as output:** This is a preload register containing the value to be compared to the counter and signaled on OC3 output.

Bit	Name	Attribute	Reset Value	Description
				If channel CC3 is configured as input: CCR3 is the counter value transferred by the last input capture event, at this point the CCR3 register is read-only.

16.7.17 Capture/Compare Register 4 (TIM_CCR4)

Offset address: 0x40

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:16	RSV	-	-	Reserved
15:0	CCR4	R/W	0x0	Capture/compare channel 4 register: **If channel CC4 is configured as output:** This is a preload register containing the value to be compared to the counter and signaled on OC4 output. **If channel CC4 is configured as input:** CCR4 is the counter value transferred by the last input capture event, at this point the CCR4 register is read-only.

16.7.18 Break and Dead-time Register (TIM_BDTR)

Offset address: 0x44

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:16	RSV	-	-	Reserved
15	MOE	R/W	0x0	Main output enable: This register controls the output enable of all channels, and the independent channel is enabled if the respective enable bits (i.e. CCxE and CCxNE) is set. MOE is set by software or automatically set by hardware if AOE = 1. It is cleared asynchronously by hardware as soon as the break input is active. 0: OC and OCN outputs disabled (with specific IO state

Bit	Name	Attribute	Reset Value	Description
				determined by OSSI) 1: OC and OCN outputs enabled (determined by the states of respective CCxE and CCxNE bits)
14	AOE	R/W	0x0	Automatic output enable: 0: MOE can be set only by software 1: MOE can be set by software or automatically set at the next update event
13	BKP	R/W	0x0	Break polarity: 0: break input is active low 1: break input is active high
12	BKE	R/W	0x0	Break enable: 0: break input disabled 1: break input enabled
11	OSSR	R/W	0x0	Off-state (of channel at CCx(N)E = 0) selection for run mode (MOE = 1): This bit is used only when MOE = 1 on channels with complementary output enabled. 0: when the output channel is disabled, OC and OCN do not drive IO, and OCxN always drives IO. 1: when the output channel is disabled, OC and OCN drive GPIO to be “inactive” (referring to OIS in CR2). *(If cfg_timx_break_ossi0_disout = 1 / TIMCFGR[0] / [1] = 0, it will have no effect and will directly cause OC4 not to drive.)”
10	OSSI	R/W	0x0	Off-state (of channel at CCx(N)E = 0) selection for idle mode (MOE = 0): This bit is used only when MOE = 0 on channels configured as outputs. 0: when the output channel is disabled, OC and OCN do not drive GPIO. ● If cfg_timx_break_ossi0_disout = 0 / TIMCFGR[0] / [1][12] = 1, it will keep driving. ● OCxN always drives IO. 1: when the output channel is disabled, OC and OCN are forced with idle level first, and then with “inactive level” (referring to OIS in CR2) after a dead-time.

Bit	Name	Attribute	Reset Value	Description
				*(MOE=0 will make OCx(N) = (|CCx(N)P） &&OISx(N), OC4 = OIS4)
9:8	LOCK	R/W	0x0	Lock configuration: 00: no bit is write-protected 01: lock level 1—DTG, OISx, OISxN, BKE, BKP and AOE bits can no longer be written. 10: lock level 2— lock level 1 + CCxP, CCxNP, OSSR and OSSI bits can no longer be written. 11: lock level 3— lock level 2 + OCxM and OCxPE bits can no longer be written when related channel is configured in output. Note: The LOCK register cannot be overwritten after being written with any value, and the write-protected register can only be overwritten after the TIM module is reset.
7:0	DTG	R/W	0x0	This bit-filed defines the duration of the dead-time inserted between the complementary outputs. DT corresponds to this duration. DTG[7:5] = 0xx: DT = DTG[7:0] * t_{DTS} DTG[7:5] = 10x: DT = (64 + DTG[5:0]) * 2 * t_{DTS} DTG[7:5] = 110: DT= (32 + DTG[4:0]) * 8 * t_{DTS} DTG[7:5] = 111: DT= (32 + DTG[4:0]) * 16 * t_{DTS}

16.7.19 DMA Control Register (TIM_DCR)

Offset address: 0x48

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:13	RSV	-	-	Reserved
12:8	DBL	R/W	0x0	DMA burst length: A read or write access to the TIM_DMAR register will trigger DMA transfer with a burst length of 1−18. 00000: 1 transfer 00001: 2 transfers

Bit	Name	Attribute	Reset Value	Description
				…… 10001: 18 transfers Others: invalid value, write prohibited
7:5	RSV	-	-	Reserved
4:0	DBA	R/W	0x0	DMA base address, defined as the offset address directed to the register: 00000: TIM_CR1 00001: TIM_CR2 00010: TIM_SMCR …… Note: When DBA + DBL exceeds the TIM register address range, the actual burst transfer stops automatically when it reaches the highest TIM register address, i.e., the burst length is shortened.

16.7.20 DMA Access Register (TIM_DMAR)

Offset address: 0x4C

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:0	DMAR	R/W	0x0	DMA register for burst access: When using DMA burst transfer, set the DMA channel peripheral address to TIM_DMAR. Accesses to this register will point to the register specified in TIM_DCR, and TIM will generate multiple DMA requests based on the DBL value.

16.8 Operation Procedure

16.8.1 Counting Mode

1. Enable the TIMx clock in RCM module.
2. Configure TIM_CR1[4] to set the counting direction.

3. Set TIM_CR1[7] to 1 to enable the auto-reload preload.
4. Configure TIM_PSC[15:0] to set the prescaler value.
5. Configure TIM_ARR[31:0] to set the auto-reload value.
6. Set TIM_RCR[7:0] to 0 to disable the repetition counting.
7. Set TIM_CR1[2] to 1 so that an update interrupt or DMA request will be generated only at counter overflow or underflow.
8. Set TIM_CR1[1] to 0 to enable the update event.
9. Set TIM_EGR[0] to 1 so that when the software sets UG, the counter will be reinitialized, the shadow register will be updated, and the prescaler counter will be cleared.
10. Set TIM_CR1[0] to 1 to enable the counter.
11. Set TIM_DIER[0] to 1 to enable the update event interrupt.

16.8.2 PWM Mode

1. Configure the GPIO alternate function to enable the TIMx clock in RCM module.
2. Configure TIM_CR1[4] to set the counting direction.
3. Set TIM_CR1[7] to 1 to enable the auto-reload preload.
4. Configure TIM_PSC[15:0] to set the prescaler value.
5. Configure TIM_ARR[31:0] to set the auto-reload value.
6. Set TIM_RCR[7:0] to 0 to disable the repetition counting.
7. According to the output channel, set TIM_CCMRx[1:0/9:8] to 0 to configure channel x as output.
8. Configure the OCxM[6:4/14:12] bits in TIM_CCMRx register to select PWM mode 1/2.
9. Configure TIM_CCER[1/3/5/7/9/11/13] to set the output polarity.
10. Set TIM_CCER[0/4/8/12] to 1 to enable channel X output.
11. Configure the main output enable bit TIM_BDTR[15] to 1 to enable OC and OCN outputs.

12. Set TIM_CR1[2] to 1 so that an update interrupt or DMA request will be generated only at counter overflow or underflow.
13. Set TIM_CR1[1] to 0 to enable the update event.
14. Set TIM_EGR[0] to 1 so that when the software sets TIM_EGR[0], the counter will be reinitialized, the shadow register will be updated, and the prescaler counter will be cleared.
15. Set TIM_CR1[0] to 1 to enable the counter.
16. Set TIM_DIER[0] to 1 to enable the update event interrupt.
17. Configure TIM_CCRx[31:0] to set the compare value of channel x.

16.8.3 Input Capture Mode

1. Configure the GPIO alternate function to enable the TIMx clock in RCM module.
2. Configure TIM_CR1[4] to set the counting direction.
3. Set TIM_CR1[7] to 1 to enable the auto-reload preload.
4. Configure TIM_PSC[15:0] to set the prescaler value.
5. Configure TIM_ARR[31:0] to set the auto-reload value.
6. Set TIM_RCR[7:0] to 0 to disable the repetition counting.
7. Configure TIM_CCMRx[1:0/9:8] to set channel CCx as input, and perform mapping as required.
8. Configure TIM_CCER[1/3/5/7/9/11/13] to set the capture polarity.
9. Configure TIM_CCMRx[7:4/15:12] to set the sampling frequency and filter length, generally set to 0.
10. Configure TIM_CCMRx[3:2/11:10] to set the prescaler value of input capture.
11. Set TIM_CCER[0/4/8/12] to 1 to enable the capture function.

12. Set TIM_EGR[0] to 1 so that when the software sets TIM_EGR[0], the counter will be reinitialized, the shadow register will be updated, and the prescaler counter will be cleared.
13. Set TIM_CR1[0] to 1 to enable the counter.
14. Set TIM_DIER[1/2/3/4] to 1 to enable the capture interrupt of channel x.

16.8.4 Complementary Output and Dead-time Insertion

Take 168 MHz as an example to calculate the cycle time: t_{DTS} = 5.95 ns

- DT = (0–127) * 5.95 = 0–755.65 ns, DTG[7:5] = 0xx: DT = DTG[7:0] * t_{DTS}
- DT = (64 + (0–63)) * 2 * 5.95 = 761.6–1511.3 ns, DTG[7:5] = 10x: DT = (64 + DTG[5:0]) * 2 * t_{DTS}
- DT = (32 + (0–31)) * 8 * 5.95 = 1523.2–2998.8 ns, DTG[7:5] = 110: DT = (32 + DTG[4:0]) * 8 * t_{DTS}
- DT = (32 + (0–31)) * 16 * 5.95 = 3046.4–5997.6 ns, DTG[7:5] = 111: DT = (32 + DTG[4:0]) * 16 * t_{DTS}

Before initializing the PWM mode, add the following configurations:

1. Configure TIM_BDTR[7:0] to set the duration of the dead-time inserted between the complementary outputs.
2. Set TIM_CCER[2/6/10] to 1 to enable the complementary output.

16.8.5 Break Function

Before initializing the PWM mode, add the following configurations:

1. Configure TIM_BDTR[11] to set the off-state for run mode.
2. Configure TIM_BDTR[10] to set the off-state for idle mode.
3. Configure TIM_BDTR[13] to set the break signal polarity.
4. Configure TIM_CCER[1/5/9/13] to set the output polarity of OCx.
5. Configure TIM_CCER[3/7/11] to set the output polarity of OCxN.
6. Configure TIM_CR2[8/10/12/14] to set the OCx output state in idle mode.

7. Configure TIM_CR2[9/11/13] to set the OCxN output state in idle mode.
8. Configure TIM_BDTR[14] to set the setting mode of TIM_BDTR[15].
9. Set TIM_BDTR[12] to 1 to enable the break input.

16.8.6 Encoder Interface Mode

1. Configure the GPIO alternate function to enable the TIMx clock in RCM module.
2. Configure TIM_CR1[4] to set the counting direction.
3. Set TIM_CR1[7] to 1 to enable the auto-reload preload.
4. Configure TIM_PSC[15:0] to set the prescaler value.
5. Configure TIM_ARR[31:0] to set the auto-reload value.
6. Set TIM_RCR[7:0] to 0 to disable the repetition counting.
7. Set TIM_CCMR1[1:0] to 1 to configure channel CC1 as input with IC1 mapped on TI1.
8. Set TIM_CCMR1[9:8] to 1 to configure channel CC2 as input with IC2 mapped on TI2.
9. Configure TIM_CCER[1] and TIM_CCER[3] to set the capture polarity.
10. Configure TIM_CCER[1] and TIM_CCER[7] to set the capture polarity.
11. Configure TIM_CCMR1[7:4] to set the sampling frequency and filter length, generally set to 0.
12. Configure TIM_CCMR1[15:12] to set the sampling frequency and filter length, generally set to 0.
13. Configure TIM_SMCR[2:0] to set encoder mode 1/2/3.
14. Set TIM_CCER[0] to 1 to enable the capture function of channel 1.
15. Set TIM_CCER[4] to 1 to enable the capture function of channel 2.
16. Set TIM_EGR[0] to 1 so that when the software sets TIM_EGR[0], the counter will be reinitialized, the shadow register will be updated, and the prescaler counter will be cleared.
17. Set TIM_CR1[0] to 1 to enable the counter.

18. Set TIM_DIER[1] to 1 to enable capture interrupt of channel 1.

16.8.7 DMA Mode

- **In input capture mode, the channel capture value of TIMx is transferred to SRAM via DMA:**

  1. In input capture mode, before setting TIM_EGR[0] bit by software and enabling the counter, add the following configurations:
  2. Configure TIM_DCR[12:8] to set the DMA burst length.
  3. Configure TIM_DCR[4:0] to set the DMA base address. Generally, the base address here selects the capture/compare register corresponding to the capture channel.
  4. Set TIM_DIER[9/10/11/12] to 1 to enable the CCx DMA request.
  5. Set TIM_CR2[3] to 0 to enable the CCxDMA request generation at CCx event.
  6. For details on DMA controller configuration, please refer to chapter "11 Direct Memory Access Controller (DMA)".
  7. After initiating DMA transfer, when a capture event occurs on the channel, DMA will transfer the value stored in base address to SRAM.

- **In output compare mode, the value in SRAM is transferred via DMA to the compare register of TIMx:**

  1. In output compare mode, before setting TIM_EGR[0] bit by software and enabling the counter, add the following configurations:
  2. Configure TIM_DCR[12:8] to set the DMA burst length.
  3. Configure TIM_DCR[4:0] to set the DMA base address. Generally, the base address here selects the capture/compare register corresponding to the compare channel.
  4. Set TIM_DIER[9/10/11/12] to 1 to enable the CCx DMA request.
  5. Set TIM_CR2[3] to 0 to generate CCxDMA request at CCx event.

6. For details on DMA controller configuration, please refer to chapter "11 Direct Memory Access Controller (DMA)".
7. After initiating DMA transfer, when the counter value matches the compare value, DMA will transfer the value in SRAM to the base address.

17 General-purpose Timers (TIM1−TIM4 & TIM8−TIM9)

17.1 Overview

The general-purpose timers consist of a 16-bit auto-reload counter driven by a programmable prescaler. It may be used for a variety of purposes, including input capture, output compare and PWM.

17.2 Main Features

- 16-bit up, down, up/down auto-reload counter
- 16-bit programmable prescaler allowing real-time adjustment of the counter clock division
- 4 independent channels for input capture, output compare, PWM generation and one-pulse output
- Support cascading with other timers
- Interrupt or DMA event can be generated in the following cases:
  - Counter overflow/underflow, counter initialization (triggered by software or hardware)
  - Trigger event (counter start, stop, initialization, or count by internal/external trigger)
  - Input capture
  - Output compare
- Support incremental quadrature encoder and Hall sensor
- Trigger input for external clock

17.3 System Block Diagram

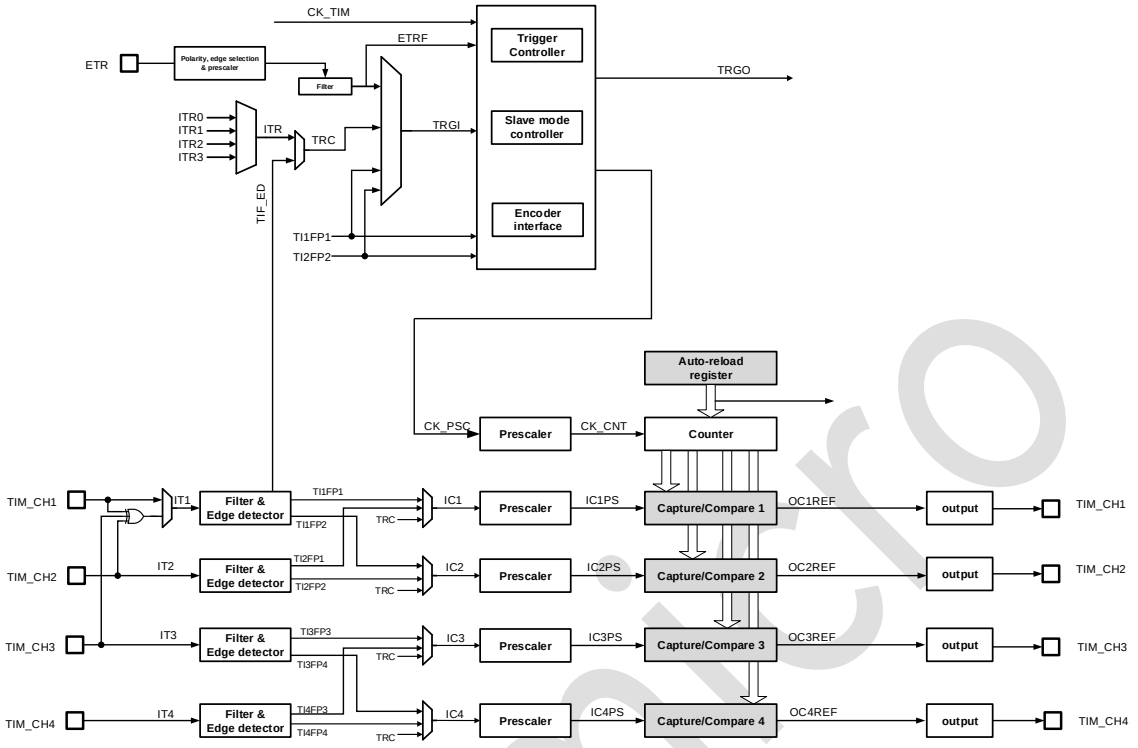


Figure 17-1: System Block Diagram of TIM1−TIM4 & TIM8−TIM9

17.4 Pin Description

Table 17-1: Pin Descriptions on TIM1−TIM4 & TIM8−TIM9

Function Pin	Alternate Function Pin	Direction	Functional Description
TIM1_ETR	PA0, PA5, PA15	Input/output	TIM1_ETR can be used as external trigger input, and cannot be used with TIM1_CH1 simultaneously.
TIM1_CH1	PA0, PA5, PA15, PD8	Input/output	TIM1_CH1 can be used as input capture or PWM output signal for channel 1. TIM1_ETR and TIM1_CH1 cannot be used simultaneously.
TIM1_CH2	PA1, PB3, PD9	Input/output	Channel input capture / PWM output signal

Function Pin	Alternate Function Pin	Direction	Functional Description
TIM1_CH3	PA2, PB10, PD10	Input/output	Channel input capture / PWM output signal
TIM1_CH4	PA3, PB11	Input/output	Channel input capture / PWM output signal
TIM2_ETR	PD2, PB2	Input	External trigger input
TIM2_CH1	PA6, PB4, PC6	Input/output	Channel input capture / PWM output signal
TIM2_CH2	PA7, PB5, PC7	Input/output	Channel input capture / PWM output signal
TIM2_CH3	PB0, PC8	Input/output	Channel input capture / PWM output signal
TIM2_CH4	PB1, PC9	Input/output	Channel input capture / PWM output signal
TIM3_ETR	PA11, PB10	Input	External trigger input
TIM3_CH1	PB6, PA6, PA7, PC4	Input/output	Channel input capture / PWM output signal
TIM3_CH2	PB7, PB1	Input/output	Channel input capture / PWM output signal
TIM3_CH3	PB8, PD14, PA4	Input/output	Channel input capture / PWM output signal
TIM3_CH4	PB9, PD15, PA5	Input/output	Channel input capture / PWM output signal
TIM4_CH1	PA0, PC0, PA4, PA7, PB1, PB12	Input/output	Channel input capture / PWM output signal
TIM4_CH2	PA1, PC1, PB13	Input/output	Channel input capture / PWM output signal
TIM4_CH3	PA2, PC2	Input/output	Channel input capture / PWM output signal
TIM4_CH4	PA3, PC3	Input/output	Channel input capture / PWM output signal
TIM8_ETR	PB8, PB15	Input	External trigger input
TIM8_CH1	PA2, PC12	Input/output	Channel input capture / PWM output signal
TIM8_CH2	PA3, PA1, PB9	Input/output	Channel input capture / PWM output signal

Function Pin	Alternate Function Pin	Direction	Functional Description
TIM8_CH3	PA6, PB10	Input/output	Channel input capture / PWM output signal (Note: If PWM output is required, TIM8_CH4 shall be enabled at the same time.)
TIM8_CH4	PB12, PB13	Input/output	Channel input capture / PWM output signal
TIM9_ETR	PC0, PC12	Input	External trigger input
TIM9_CH1	PB8, PC4, PB0, PB1	Input/output	Channel input capture / PWM output signal
TIM9_CH2	PA15, PC5	Input/output	Channel input capture / PWM output signal
TIM9_CH3	PB2, PC11, PB13	Input/output	Channel input capture / PWM output signal (Note: If PWM output is required, TIM9_CH4 shall be enabled at the same time.)
TIM9_CH4	PC10, PC11	Input/output	Channel input capture / PWM output signal

17.5 Functional Description

17.5.1 Time-base Unit

The main block of the time-base unit is a 16-bit counter with its related auto-reload register. The counter can count up, down, or both up and down. The counter clock can be divided by a 16-bit prescaler.

The counter, the auto-reload register and the prescaler register can be written or read by software, which is true even when the counter is running.

The time-base unit includes:

- Counter register (TIM_CNT)
- Prescaler register (TIM_PSC)
- Auto-reload register (TIM_ARR)

The auto-reload register is preloaded, which is controlled by the auto-reload preload enable (ARPE) bit in the register. When ARPE = 0, write to the ARR register, and the written data is directly transferred to the shadow register. When ARPE = 1, the data written to the ARR register is transferred to the shadow register when an update event (TIM_CNT overflow or underflow) occurs. The update event of ARR can also be actively triggered by software via register operation.

The counter TIM_CNT is clocked by the prescaler output TIM_PSC, which is enabled only when the counter enable bit (CEN) in the register is set. When CNT = ARR, this round of counting is over and the update event is sent.

TIM_PSC is a synchronous prescaler that can divide the counter clock frequency by any factor between 1 and 65536. The PSC register is also buffered, and overwriting PSC does not actually overwrite the shadow register unless a new update event occurs. Thus the PSC register can be changed in real time on the fly, and the new prescaler ratio is taken into account at the next update event.

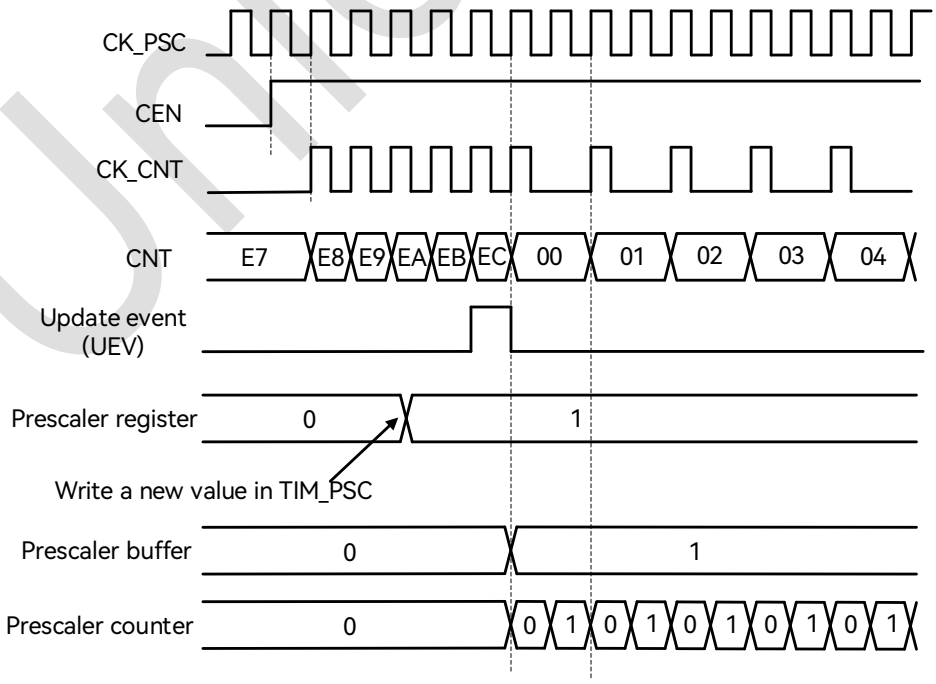


Figure 17-2: Counter Timing Diagram with Prescaler Division Changing from 1 to 2

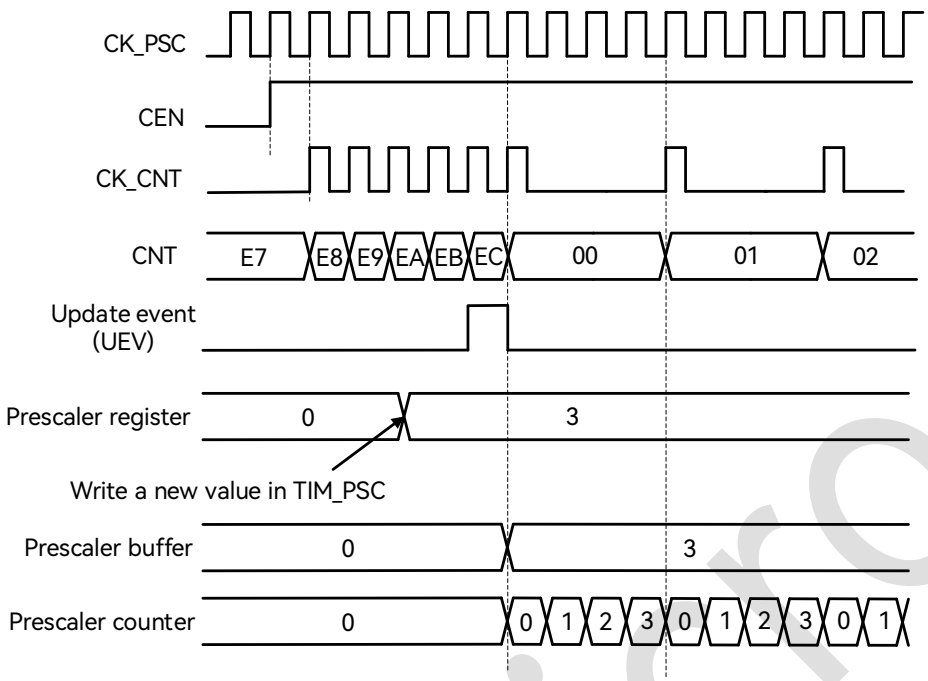


Figure 17-3: Counter Timing Diagram with Prescaler Division Changing from 1 to 4

17.5.2 Counter Operation Mode

The counter supports up-counting mode, down-counting mode and center-aligned mode.

17.5.2.1 Up-counting Mode

In up-counting mode, the counter counts from 0 to the auto-reload value, i.e. CNT = ARR, generating an overflow event, and then restarts counting from 0.

If the repetition counter is enabled, the counter repeats the above process a number of times (RCR + 1) as defined in RCR before generating an underflow event.

The software can directly trigger an update event by setting the UG bit in the register, at which time the CNT and the prescaler registers are automatically cleared. Whether setting the UG register triggers UIF (update interrupt flag) is determined by the setting of the URS register.

The update event can be disabled by setting the UDIS bit in the register to avoid updating the shadow register while writing new values in the preload registers.

When an update event occurs, the following registers are updated and the UIF bit is set:

- The auto-reload shadow register ARR is reloaded with the content of TIM_ARR register.
- The prescaler shadow register PSC is reloaded with the content of TIM_PSC register.

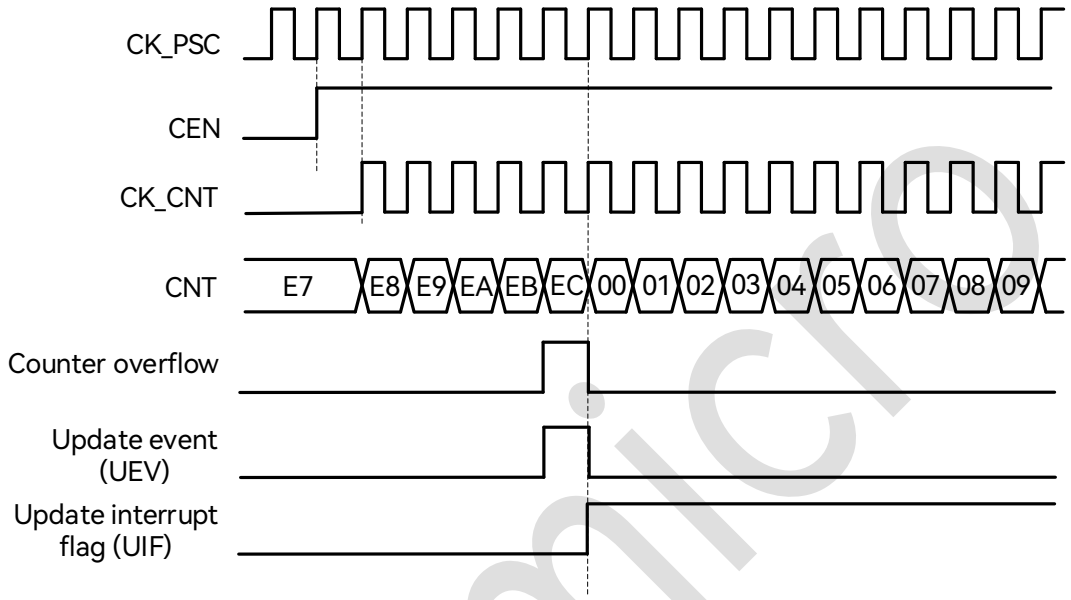


Figure 17-4: Up-counting Waveform Diagram, Internal Clock not Divided

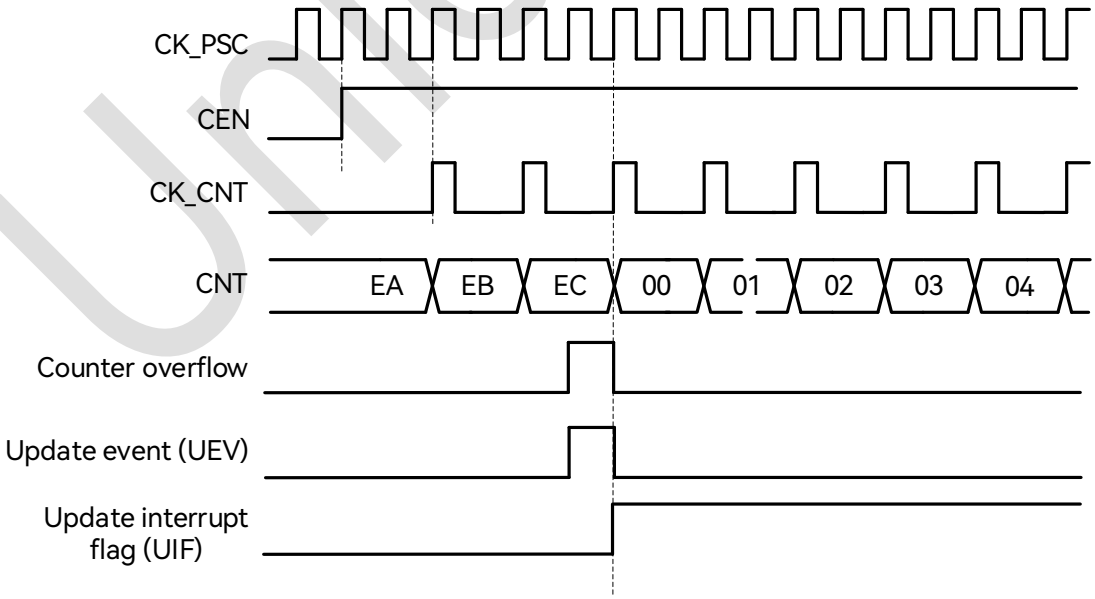


Figure 17-5: Up-counting Waveform Diagram, Internal Clock Divided by 2

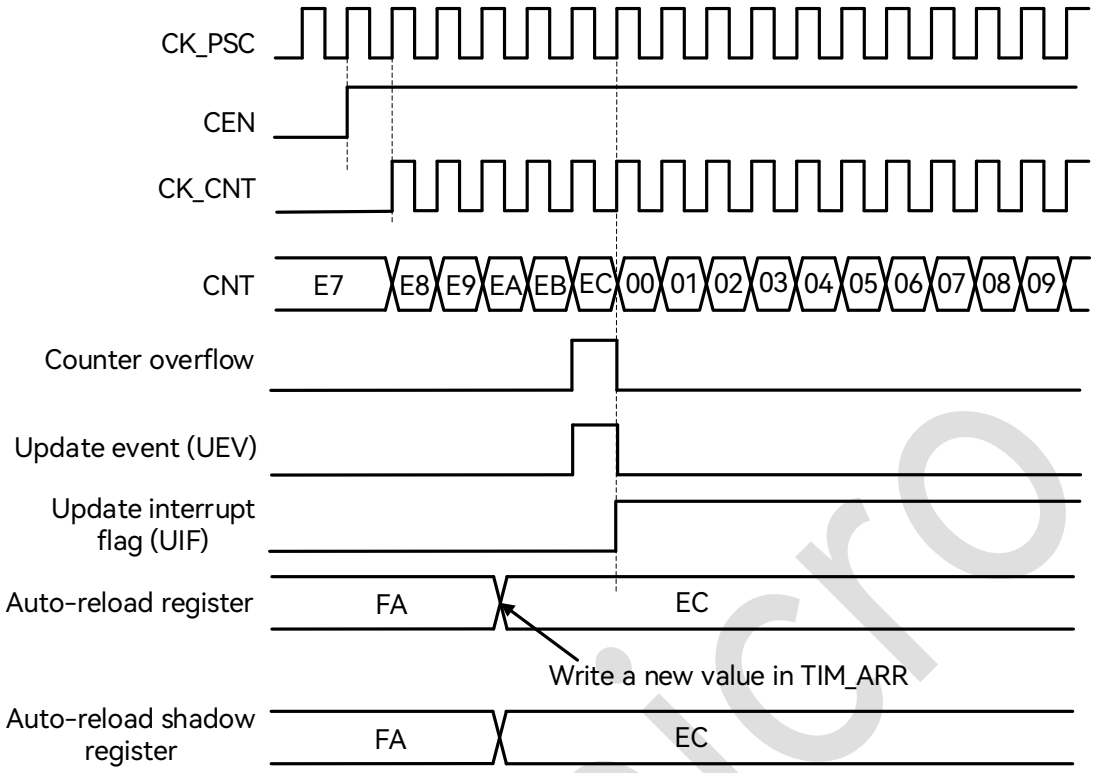


Figure 17-6: Counter Timing Diagram, Update Event when ARPE = 0 (TIM_ARR not Preloaded)

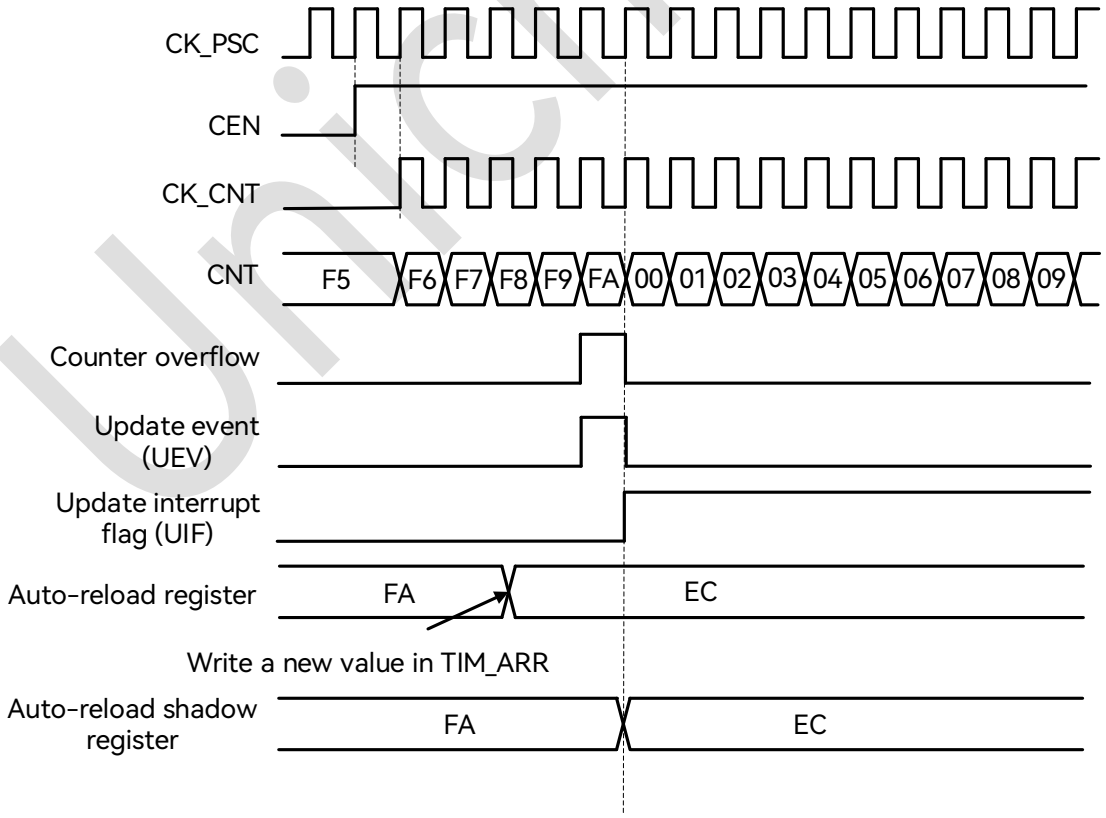


Figure 17-7: Counter Timing Diagram, Update Event when ARPE = 1 (TIM_ARR Preloaded)

17.5.2.2 Down-counting Mode

In down-counting mode, the counter counts from the auto-reload value down to 0, generating an underflow event, and then restarts counting from the auto-reload value.

If the repetition counter is enabled, the counter repeats the above process a number of times (RCR + 1) as defined in RCR before generating an underflow event.

The software can directly trigger an update event by setting the UG bit in the register, at which time the CNT and the prescaler registers are automatically cleared. Whether setting the UG register triggers UIF (update interrupt flag) is determined by the setting of the URS register.

The update event can be disabled by setting the UDIS bit in the register to avoid updating the shadow register while writing new values in the preload registers.

When an update event occurs, the following registers are updated and the UIF bit is set:

- The auto-reload shadow register ARR is reloaded with the content of TIM_ARR register.
- The prescaler shadow register PSC is reloaded with the content of TIM_PSC register.

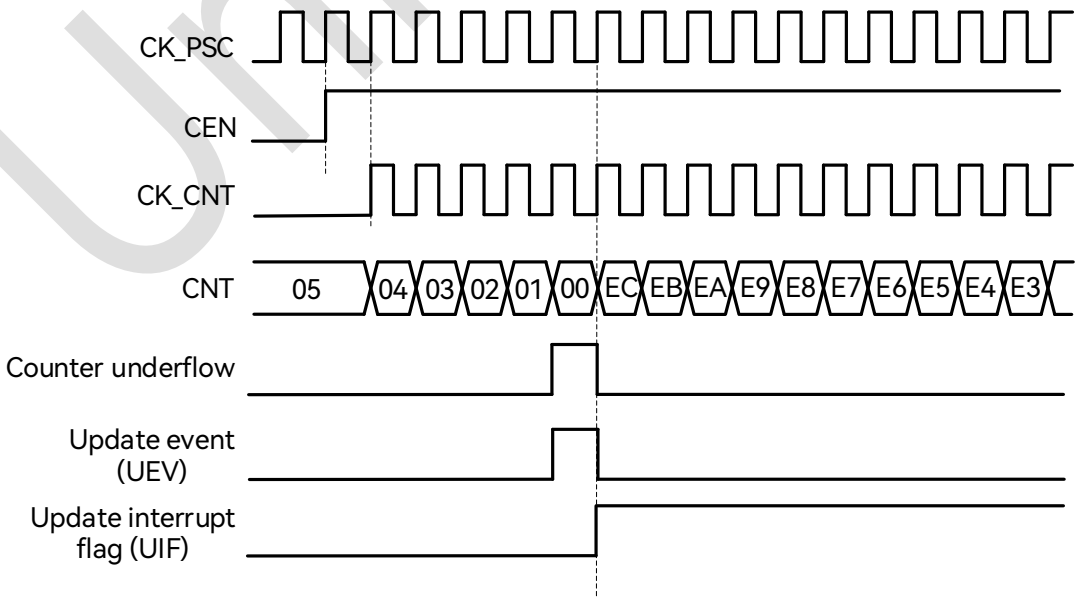


Figure 17-8: Down-counting Waveform Diagram, Internal Clock not Divided

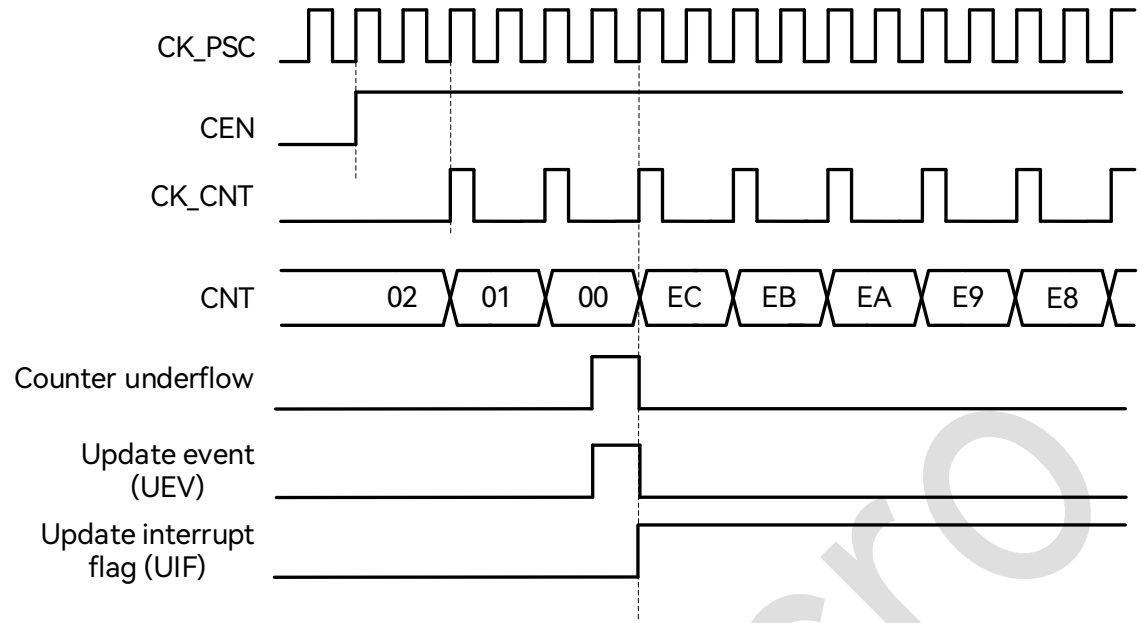


Figure 17-9: Down-counting Waveform Diagram, Internal Clock Divided by 2

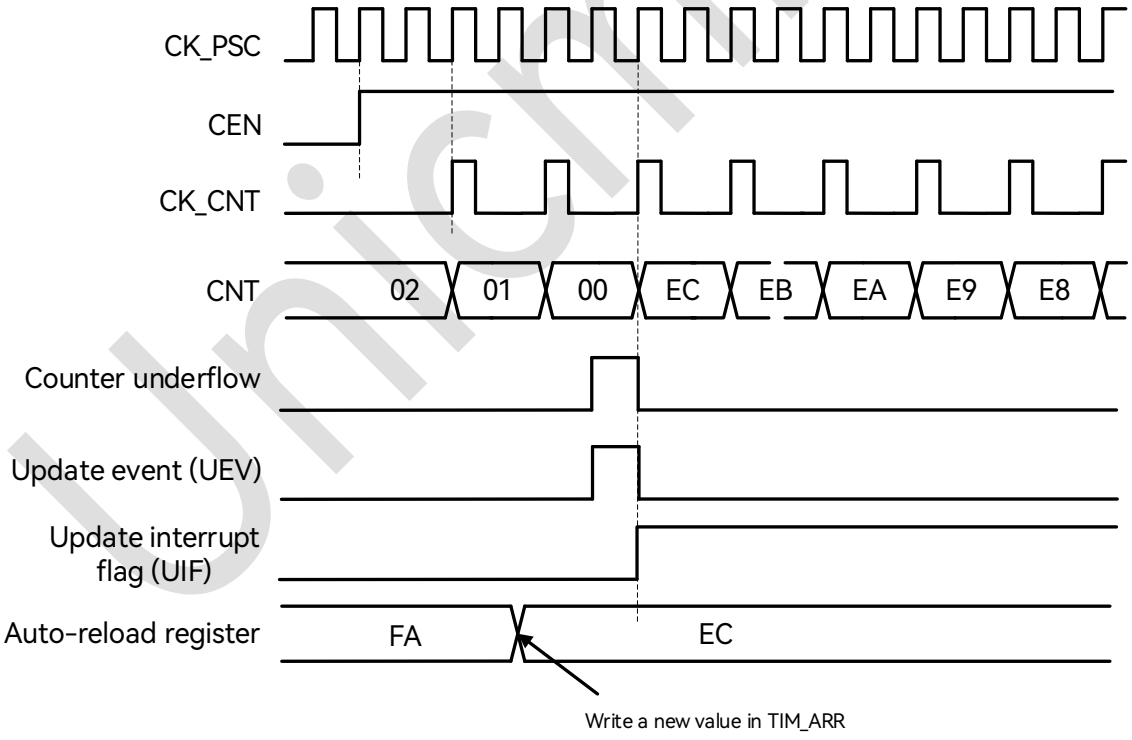


Figure 17-10: Down-counting Waveform Diagram, Internal Clock Divided by 2

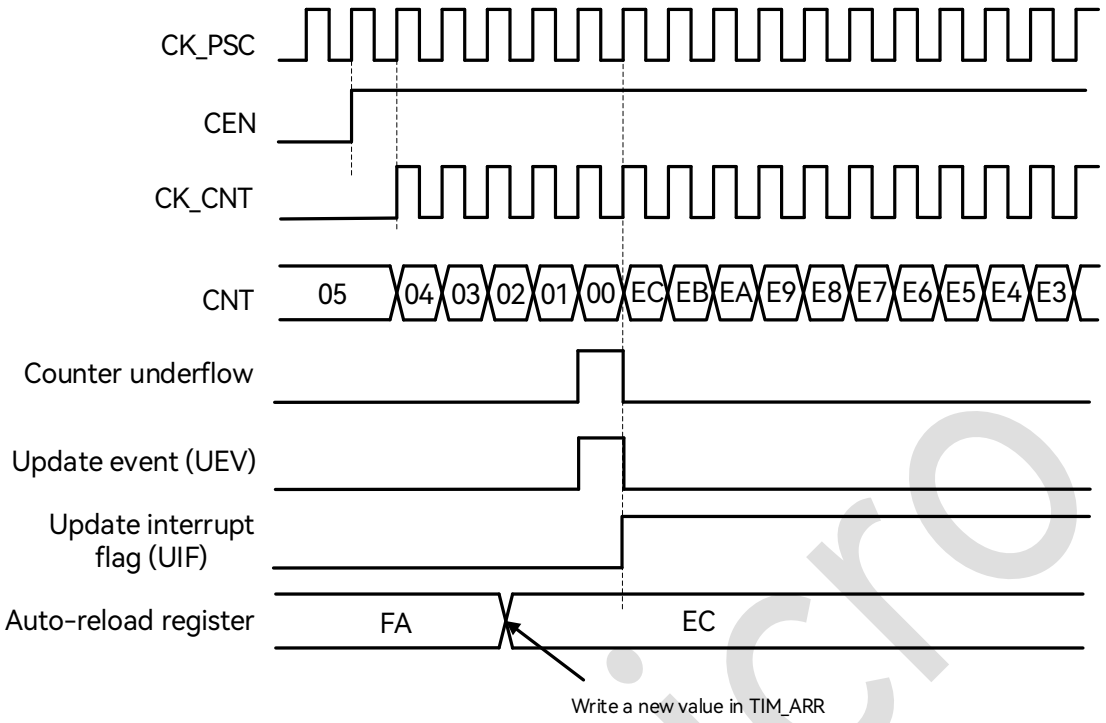


Figure 17-11: Down-counting Waveform Diagram, Update Event when Repetition Counter is not Used

17.5.2.3 Center-aligned Counting Mode

In center-aligned mode, the counter counts from 0 to the auto-reload value - 1 and generates a counter overflow event, then counts from the auto-reload value down to 1 and generates a counter underflow event, and then restarts counting from 0.

The CMS[1:0] bits in the register are used for enabling the center-aligned mode and selecting the output compare mode herein. The center-aligned mode is active when CMS! = 00. The output compare interrupt flag of channels configured in output is set when: the counter counts down (CMS = 01), the counter counts up (CMS = 10), the counter counts up and down (CMS = 11).

In this mode, the DIR direction bit in the register cannot be written by software. It is updated by hardware and gives the current direction of the counter.

The counter updates the shadow registers of ARR and PSC at each counter overflow and underflow.

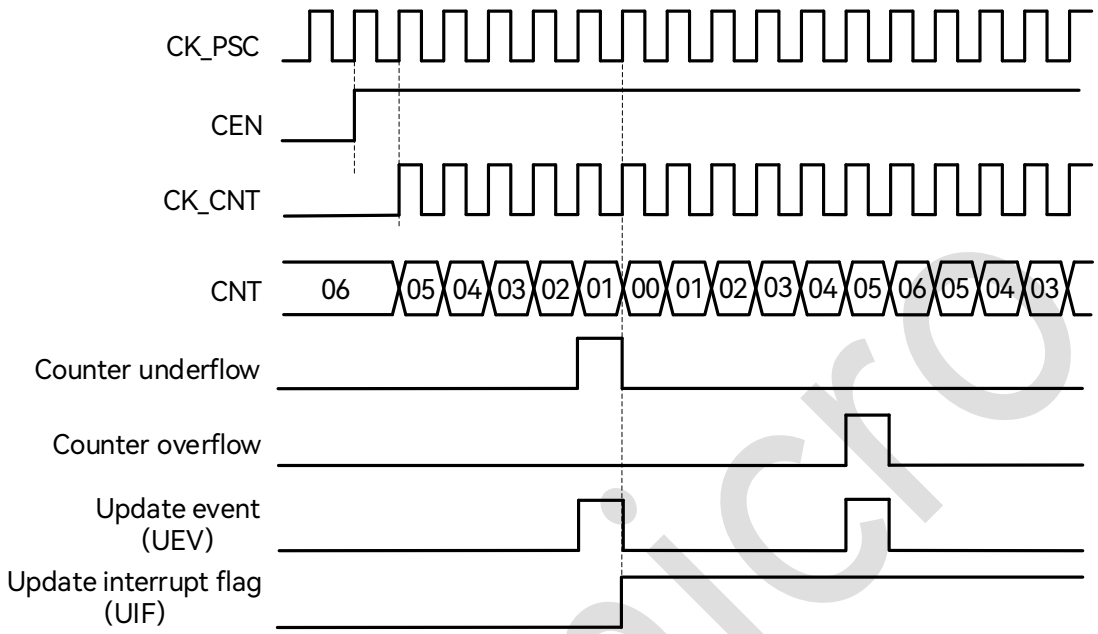


Figure 17-12: Center-aligned Counter Timing Diagram, TIM_PCS = 0, TIM_ARR = 0x6

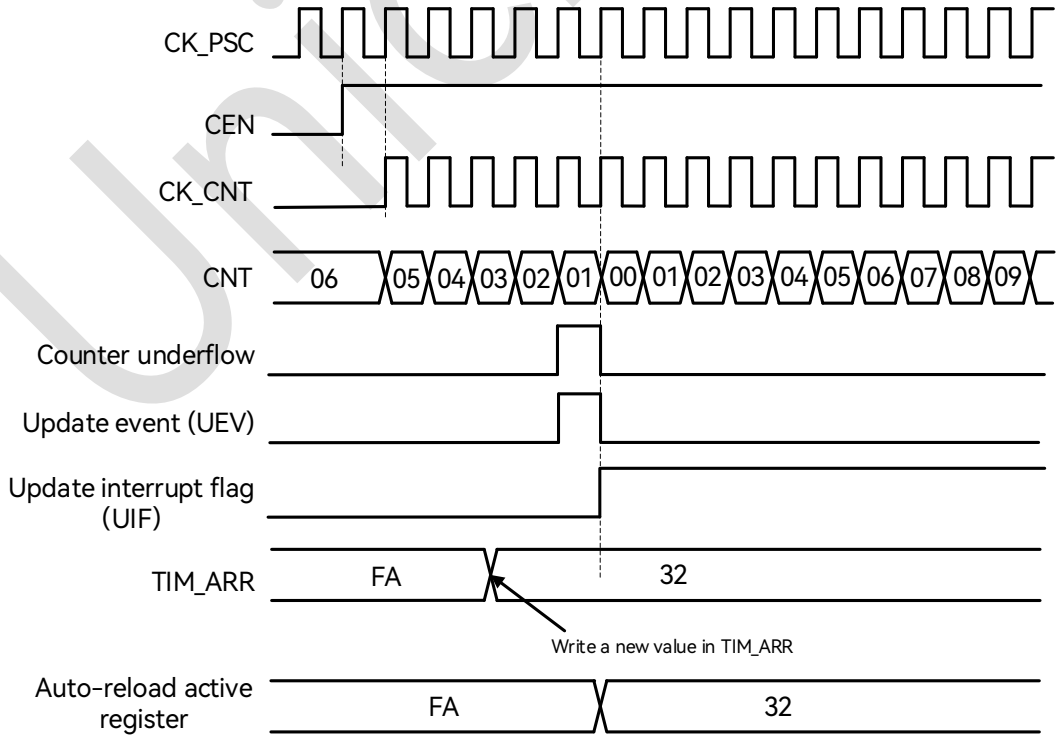


Figure 17-13: Counter Timing Diagram, Update Event with ARPE = 1 (Counter Underflow)

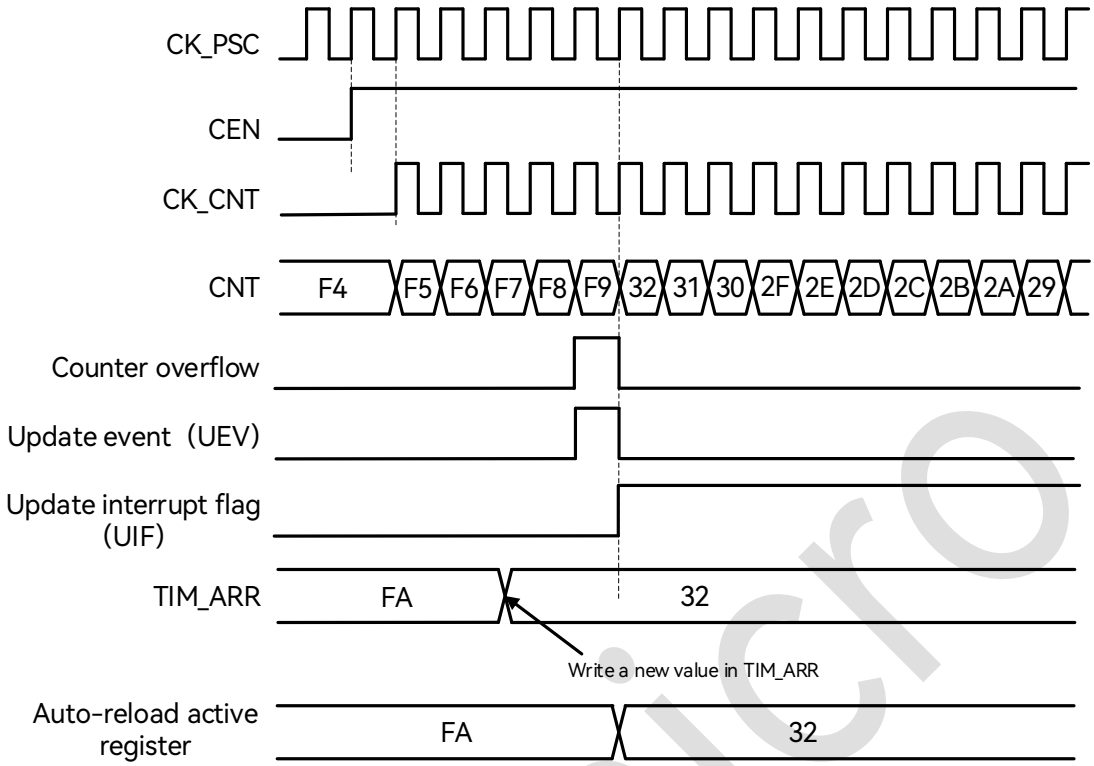


Figure 17-14: Counter Timing Diagram, Update Event with ARPE = 1 (Counter Overflow)

17.5.3 Preload Register

- The following functional registers support the preload function:
  - Auto-reload register TIM_ARR
  - Prescaler register TIM_PSC (preload function cannot be disabled)
  - Channel control register TIM_CCR
  - CCxE and CCxNE control register
  - OCxM control register

  The preload function can be enabled or disabled by software for all of the above registers except TIM_PSC.

- Registers with preload function contain two sets of physical entities:
  - Shadow register: the register being used by the actual timer

- Preload register: the register accessible to software

- When the preload function is disabled, the register with preload function has the following characteristics:
  - The preload register can be accessed and overwritten by software in real time.
  - The shadow register is updated synchronously with the preload register.
- If the preload function is enabled, then:
  - All software operations access the preload register.
  - At the occurrence of update event, the content of all preload registers will be synchronously transferred to the corresponding shadow registers.

17.5.4 Counter Clock

The counter clock can be provided by the following clock sources:

- Internal clock: Timerx_clk
- External clock mode 1: external input pin TIx
- External clock mode 2: external trigger input ETR
- Internal trigger inputs (ITRx): using the trigger output (TRGO) of one timer as the counter clock

17.5.4.1 Internal Clock Source

If the slave mode controller is disabled (SMS = 000), then the CEN, DIR and UG bits are controlled by software.

After the UG bit is set and the update signal is synchronized by CLK_PSC, the counter value is reinitialized.

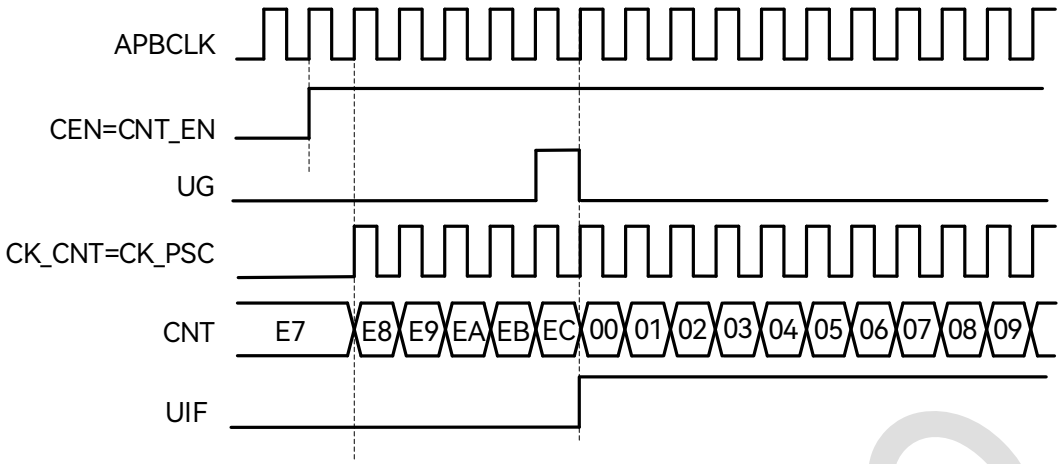


Figure 17-15: Timing Diagram in Internal Clock Source Mode, Clock Divided by 1

17.5.4.2 External Clock Source Mode 1

In this mode, the external pin input signal is directly used as the counter clock when SMS = 11, and the counter can count at each rising or falling edge.

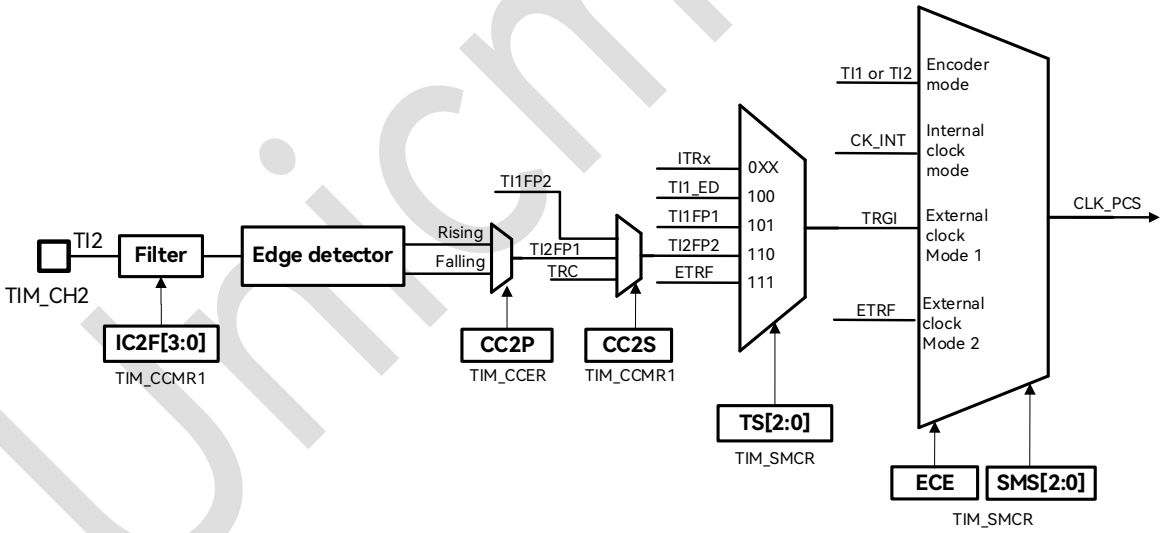


Figure 17-16: Example of External Clock Connection

The external input signal will be synchronized with the internal clock before triggering the counter counting, and the TIF flag will be set by the valid edge on the input.

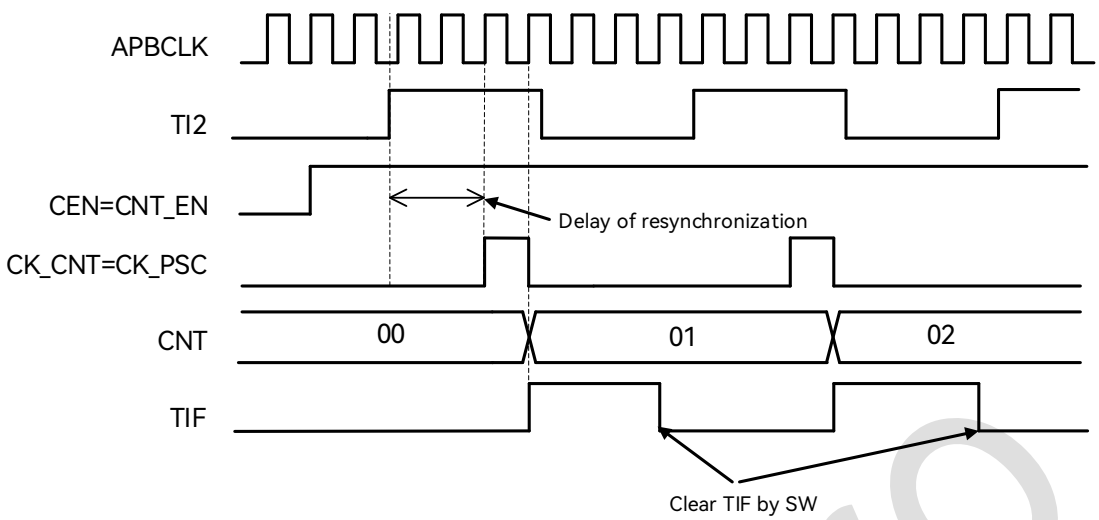


Figure 17-17: Timing Diagram in External Clock Source Mode 1

When counting with an external clock, the internal clock (timerx_clk) shall still be enabled so that TIM can use timerx_clk to synchronize and filter the external input clock. In external clock mode 1, the external input clock is first subject to filtering and edge selection to obtain a valid counting edge, which is input to the prescaler module as the valid operating clock (CLK_PSC).

The external clock synchronization adopts a simple two-stage flip-flop structure, so in order to avoid metastability, the external input clock width is required to be at least 2 timerx_clk cycles.

In this mode, only the inputs of channels 1 and 2 can be used as clock inputs, and the required configuration is as follows:

1. In GPIO module, configure the corresponding pin as TIM_CH2.
2. Disable the channel by setting TIM_CCER[4] = 0 to ensure the success of subsequent channel configuration.
3. Select the input channel by setting TIM_CCMR1[9:8] = 01, with IC2 mapped on TI2.
4. Select the active counting edge to be the rising edge or falling edge by setting TIM_CCER[5] = 0.

5. Configure the input filter duration by writing the IC2F[3:0] bits in the TIM_CCMR1 register (if no filter is required, keep IC2F = 0000).
6. Enable the external clock source mode 1 by setting TIM_SMCR[2:0] = 111.
7. Select TI2 as the trigger input source by setting TIM_SMCR[6:4] = 110.
8. Enable the channel by setting TIM_CCER[4] = 1.
9. Enable the counter by setting TIM_CR1[0] = 1.

The following diagram shows an example of typical external clock source mode 1:

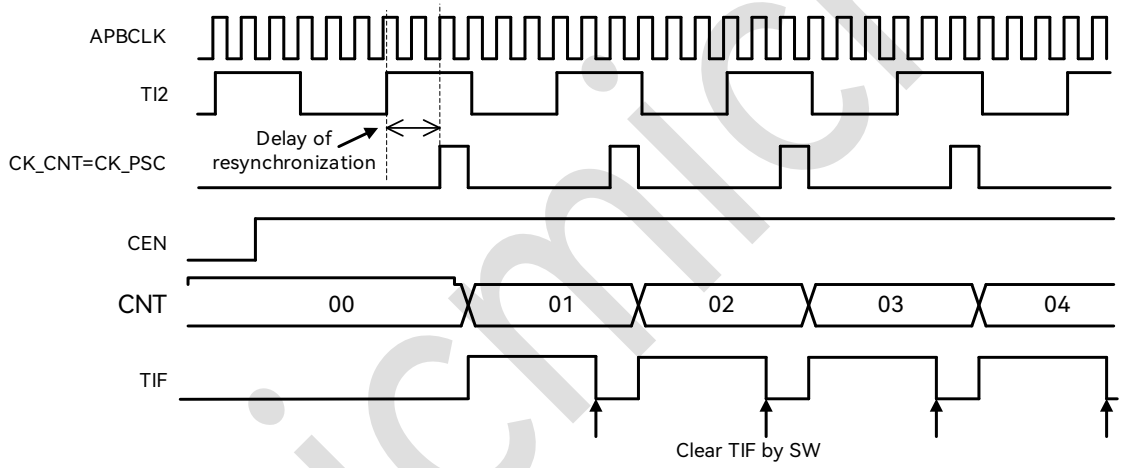


Figure 17-18: Timing Diagram in External Clock Source Mode 1

17.5.4.3 External Clock Source Mode 2

In this mode, the counter counts at each rising edge or falling edge (double-edge not supported) on the external trigger input TIM_ETR pin.

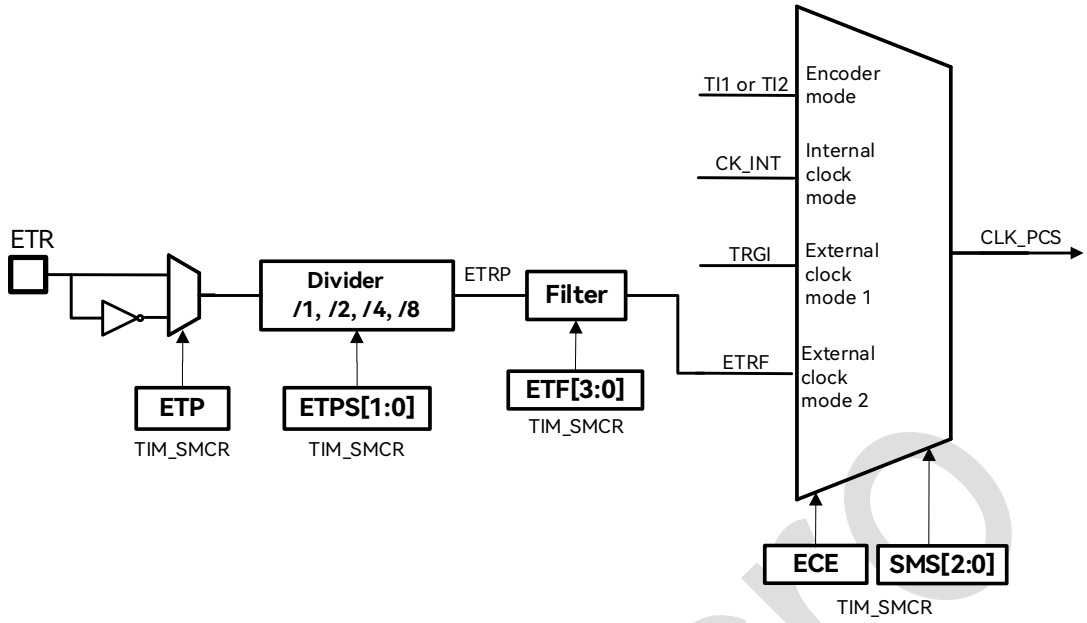


Figure 17-19: External Trigger Input Block Diagram

The following diagram shows the counter counting each 2 rising edges on ETR. The delay between the rising edge of ETR and the actual clock of the counter is due to the resynchronization of the internal clock.

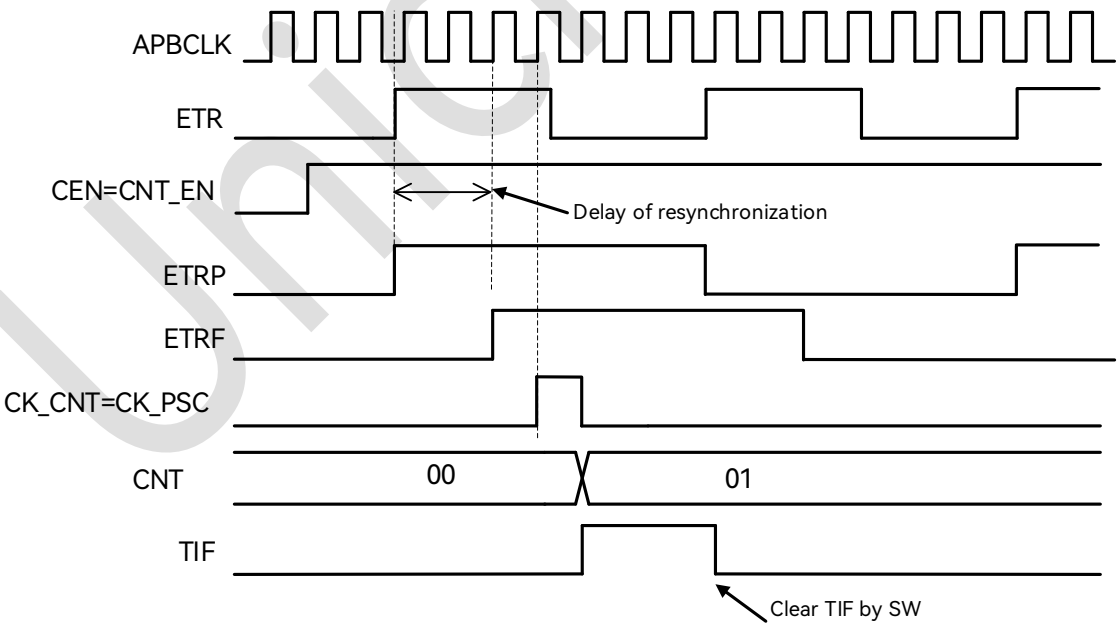


Figure 17-20: Timing 1 Diagram in External Clock Source Mode 2

The main difference from external clock source mode 1 is that the ETR input is directly divided and then filtered to generate CK_PSC clock, which mens that the application scenarios where

the ETR input frequency is higher than timerx_clk can be supported, in which case the ETR input shall be pre-divided first before it is used to drive the counter.

The configuration required for this mode is as follows:

1. In GPIO module, configure the corresponding pin as TIM_ETR.
2. Select the ETP edge by setting TIM_SMCR[15] = 0.
3. Set the ETR division ratio by setting TIM_SMCR.ETPS[1:0] = 01.
4. Configure the input filter duration by setting TIM_SMCR.ETF[3:0] = 0000.
5. Set the ECE register and enable the external clock source mode 2 by setting TIM_SMCR[14] = 1 and TIM_SMCR[2:0] = 000.
6. Enable the counter by setting TIM_CR1[0] = 1.

The following diagram shows an example of typical external clock source mode 2:

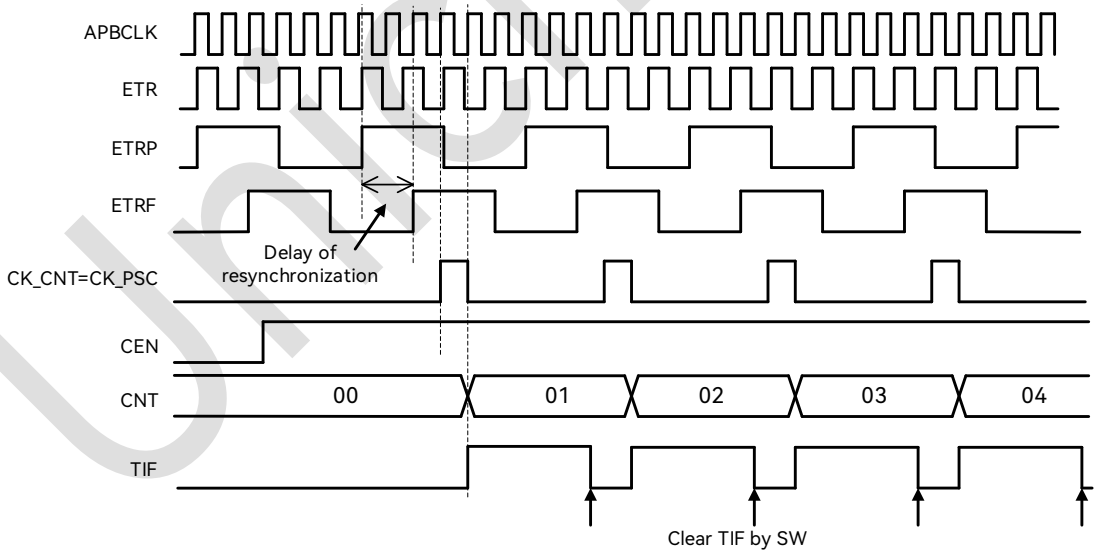


Figure 17-21: Timing 2 Diagram in External Clock Source Mode 2

In external clock source mode 2, TIM can still be configured as slave mode: for example, ETR input is used for counting while TRGO of another timer is used as the trigger signal, and the reset counter restarts counting at the arrival of trigger event.

17.5.5 Internal Trigger Signal (ITRx)

TIM supports four internal trigger inputs, which can be used for counting trigger or internal signal capture. For internal signal capture, it is required to configure TS to 000−011 for selecting ITR0−ITR3, and configure CCxS to 11 for selecting TRC as the capture signal.

Each ITR input supports 4 internal signal extensions configured by the ITRxSEL register.

17.5.6 Capture / Compare Channels

TIM consists of 4 capture/compare channels, each of which is built around a capture/compare register CCR (including a shadow register), an input stage for capture and an output stage for compare.

The input stage samples the corresponding TIx input to generate a filtered signal TIxF. Then, an edge detection with polarity selection generates a signal (TIxFPx), which can be used as trigger input for counting or as the capture command and is prescaled before being captured.

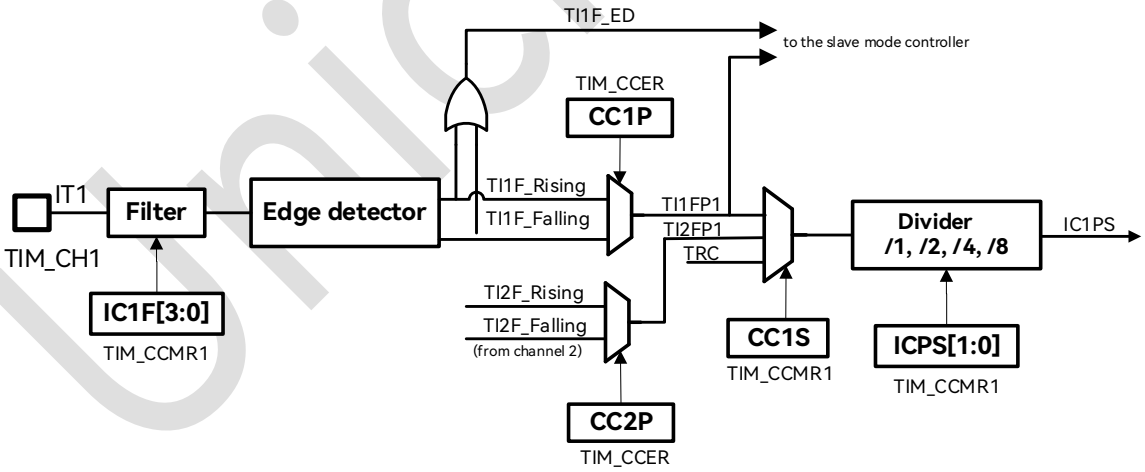


Figure 17-22: Capture/Compare Channel (Channel 1 Input Stage)

The output stage generates an output reference signal OCxREF, which is fixed to be active high and acts as the reference input to the final output circuit. Wherein, channels 1−3 support complementary output and dead-time insertion, while channel 4 is relatively simple and does not support complementary output.

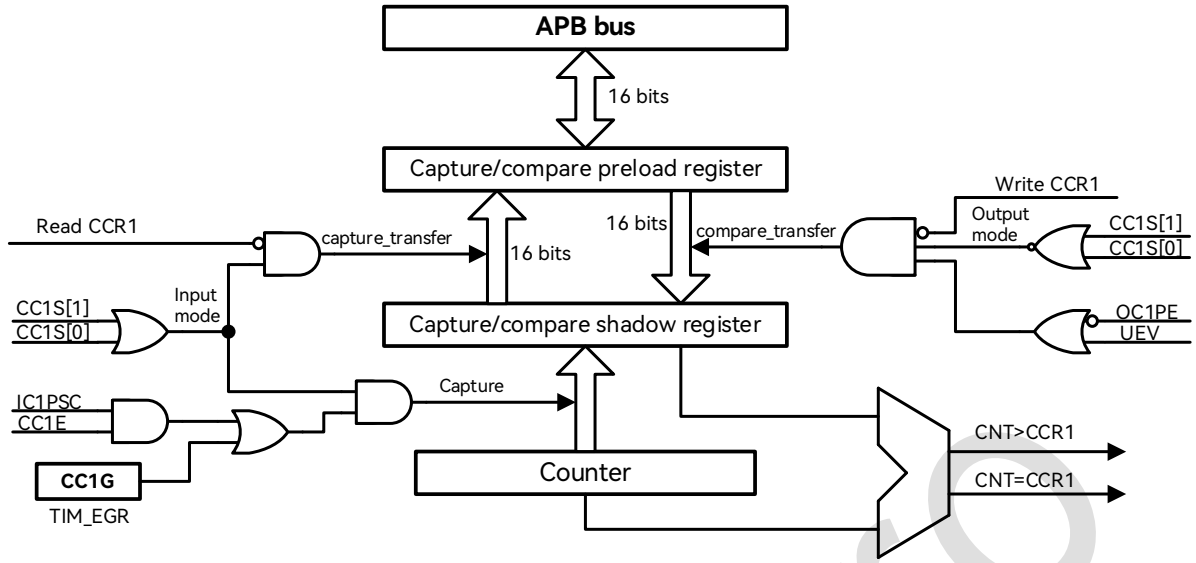


Figure 17-23: Capture/Compare Channel 1 Main Circuit

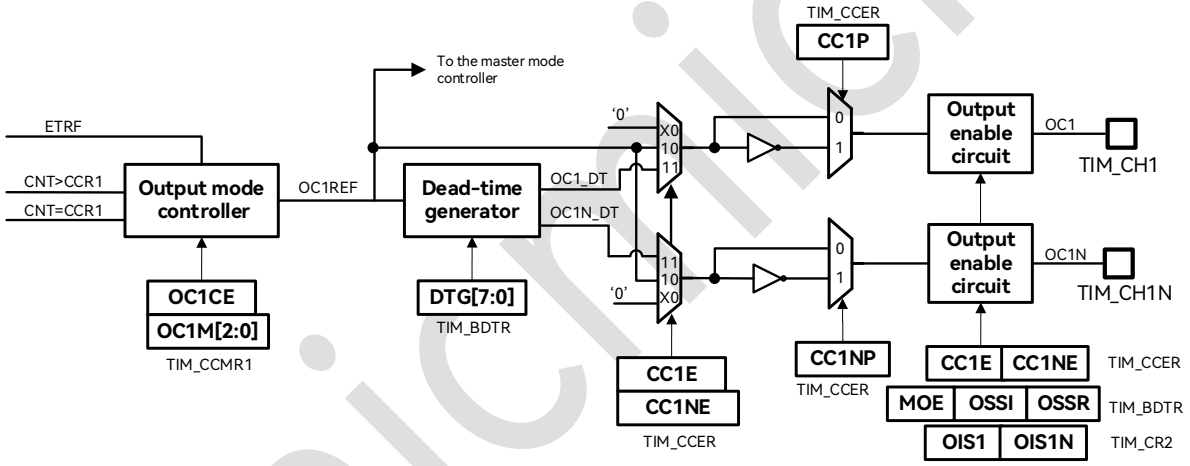


Figure 17-24: Output Stage of Capture/Compare Channel (Channels 1~3)

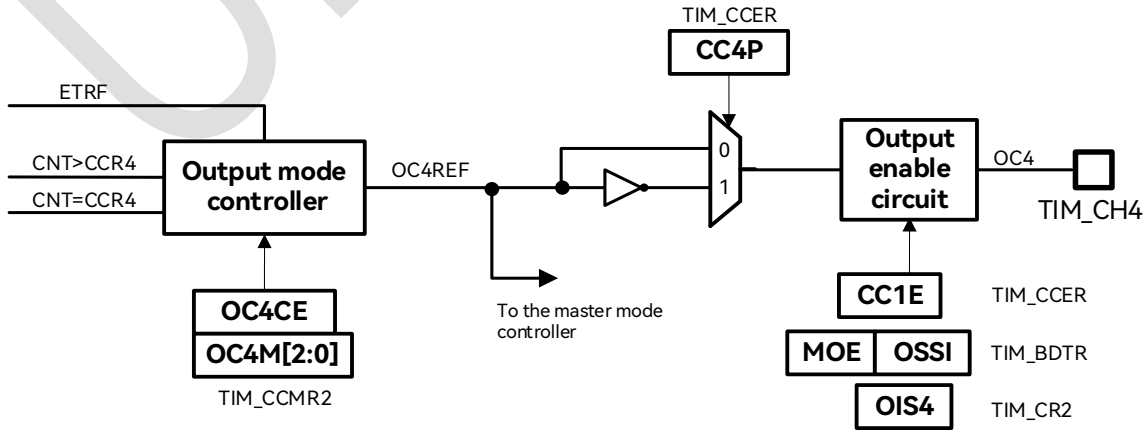


Figure 17-25: Output Stage of Capture/Compare Channel (Channel 4)

The capture/compare block is made of one preload register and one shadow register. Write and read always access the preload register. In capture mode, captures are actually done in the shadow register, which is copied into the preload register. In compare mode, the content of the preload register is copied into the shadow register for comparison with the counter.

17.5.7 Input Capture Mode

When the expected level transition is detected by the ICx signal, a capture is triggered, and the current counter value is latched into CCR. At the same time, the CCxIF interrupt flag is set and a corresponding interrupt or a DMA request can be triggered. If a capture occurs while the CCxIF flag is already high, then the over-capture flag CCxOF is set (the last capture value in CCR is overwritten). CCxIF can be cleared by software or automatically cleared by reading the CCR register. CCxOF is cleared by software writing it to 1.

The input capture of PWM signals can be realized through the cooperation of two or more channels. For example, to calculate the period and duty cycle of an input signal, input the signal from TI1 pin, and take the rising edge and falling edge of the filtered signal inside the chip to obtain TI1FP1 and TI1FP2 respectively. Input TI1FP1 into capture channel 1 and TI1FP2 into channel 2, allowing channel 1 to capture the rising edge of the input signal and channel 2 to capture the falling edge of the input signal. After the capture interrupt occurs periodically, the software can calculate the period and duty cycle of the input signal through the values of CCR1 and CCR2 registers.

To capture the counter value to the TIM_CCR1 register on the rising edge of the TI1 input, the configuration steps are as follows:

1. In GPIO module, configure the corresponding pin as TIM_CH1.
2. Disable the channel by setting TIM_CCER[0] = 0 to ensure the success of subsequent channel configuration.

3. Select the input channel by setting TIM_CCMR1[1:0] = 01, with IC1 mapped on TI1.

4. Select the active counting edge to be rising edge or falling edge by setting TIM_CCER[1].

5. Configure the input filter duration by writing the IC1F[3:0] bits in the TIM_CCMR1 register.

6. Configure the input prescaler by writing the IC1PS[1:0] bits in the TIM_CCMR1 register.

7. Enable the channel by setting TIM_CCER[0] = 1.

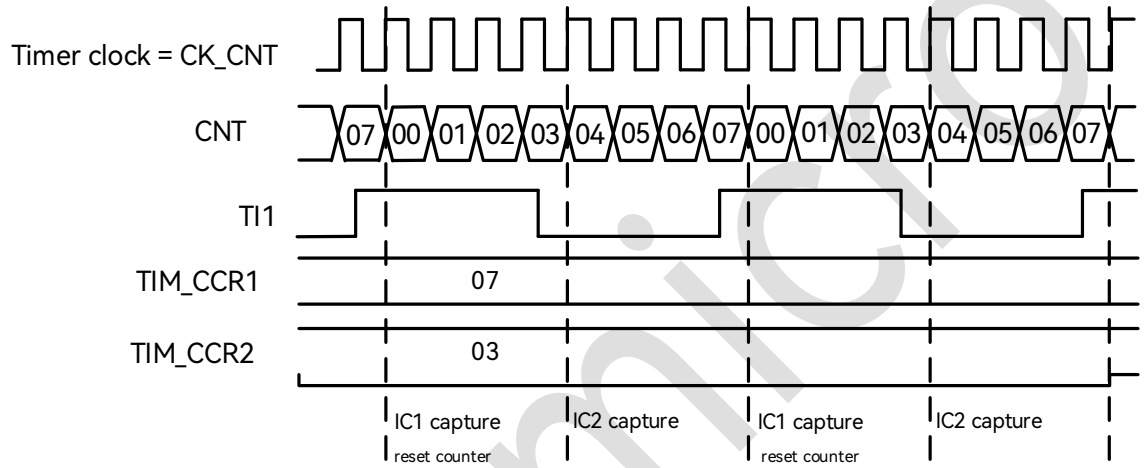


Figure 17-26: PWM Input Capture Mode Timing Diagram

The following settings are required for PWM input capture:

1. In GPIO module, configure the corresponding pin as TIM_CH1.

2. Disable the channel by setting TIM_CCER[0] = 0 and TIM_CCER[4] = 0 to ensure the success of subsequent channel configuration.

3. Select the input channel by setting TIM_CCMR1[1:0] = 01 and TIM_CCMR1[9:8] = 10, with the two signals IC1 and IC2 mapped on the same TI1 input.

4. Select the active counting edge by setting TIM_CCER[1] = 0 and TIM_CCER[5] = 1, with the two signals IC1 and IC2 active on edges with opposite polarities.

5. Configure the input filter duration by setting the IC1F[3:0] and IC2F[3:0] bits in the TIM_CCMR1 register.

6. Configure the input prescaler by setting the IC1PS[1:0] and IC2PS[1:0] bits in the TIM_CCMR1 register.
7. Select the trigger input source by setting TIM_SMCR[6:4][2:0] = 101.
8. Configure the slave mode controller to reset mode by setting TIM_SMCR.SMS[2:0] = 100.
9. Enable the channel by setting TIM_CCER[0] = 1 and TIM_CCER[4] = 1.

17.5.8 Software Force Output

In output compare mode, the OCxREF signal can be forced to active or inactive level directly by software, independently of any comparison between the CCR and the counter.

The OCxREF signal can be forced to be active (OCxREF is always active high) by writing OCxM = 101, and forced to be inactive (low level) by writing OCxM = 100. Anyway, the comparison between CCR and the counter is still performed.

17.5.9 Output compare mode

In output compare mode, when a match is found between the capture/compare register CCR and the counter, the OCxREF can be set to be active, inactive or to toggle on match. At the same time, the interrupt flag is also set and DMA requests can be sent (overwriting the configuration register).

The output compare can also be used to output a pulse signal of a specific width (in one-pulse mode).

Procedure:

1. Select the counter clock (internal, external, prescaler).
2. Write the desired data to the ARR and CCR registers.
3. Set the interrupt enable bit and DMA enable bit as required.

4. Select the output mode.

5. Enable the counter.

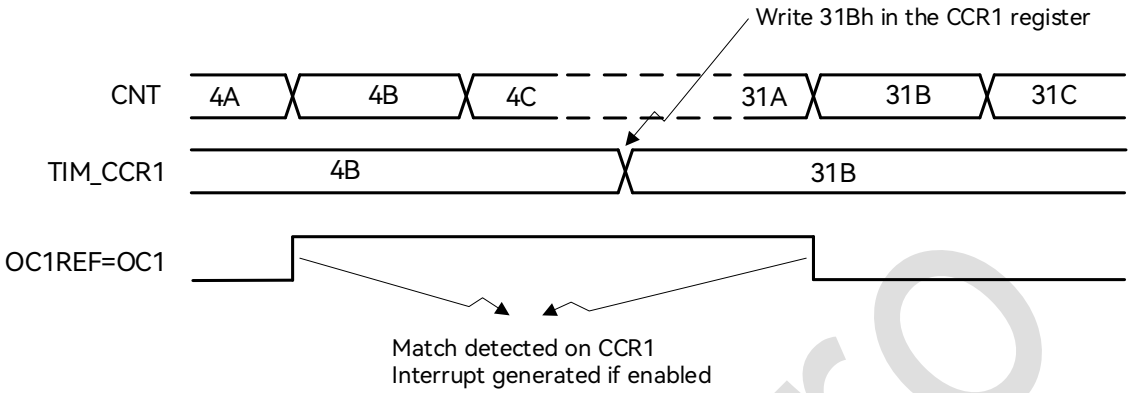


Figure 17-27: Output Compare Mode, Toggle on OC1

The CCR register can be updated at any time by software to control the output waveform, provided that the preload register is not enabled. Otherwise, the CCR shadow register is only updated with the content of the preload register at the next update event.

17.5.10 PWM Output

PWM mode allows you to generate a pulse width modulation signal with a frequency determined by the value of the ARR register and a duty cycle determined by the value of the CCR register.

The polarity of the output signal is software programmable using the CCxP bit in the register. In PWM mode, CNT and CCR registers are always compared. The timer is able to generate PWM in edge-aligned mode or center-aligned mode.

17.5.10.1 PWM Edge-aligned Mode

In up-counting mode, when it is configured in PWM mode 1, the OCxREF signal is high as long as CNT < CCR, otherwise it is low. And OCxREF will be held at 1 if CCR > ARR while held at 0 if CCR is 0.

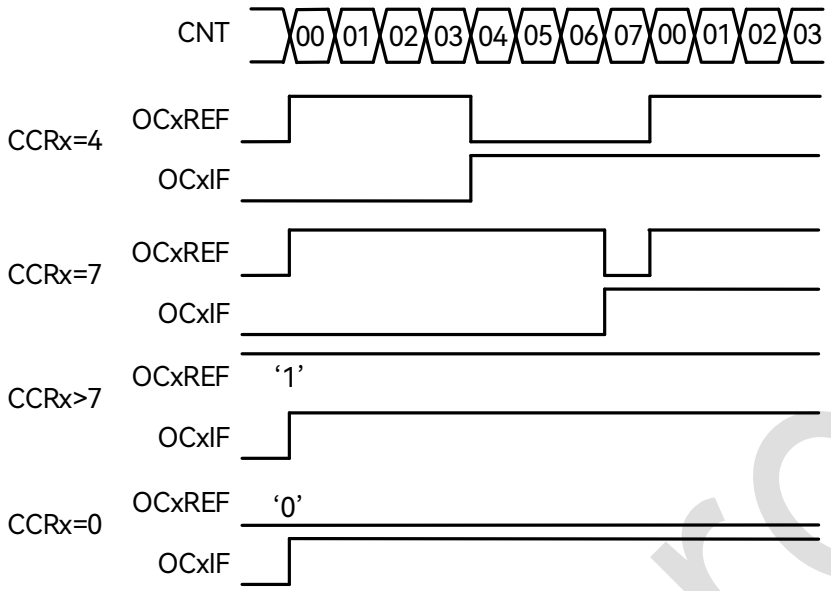


Figure 17-28: Edge-aligned PWM Waveform (ARR = 7)

In down-counting mode, the definition of OCxREF level is the same as that in up-counting mode.

17.5.10.2 PWM Center-aligned Mode

The definition of OCxREF level is the same as that in edge-aligned mode. The figure below is an example.

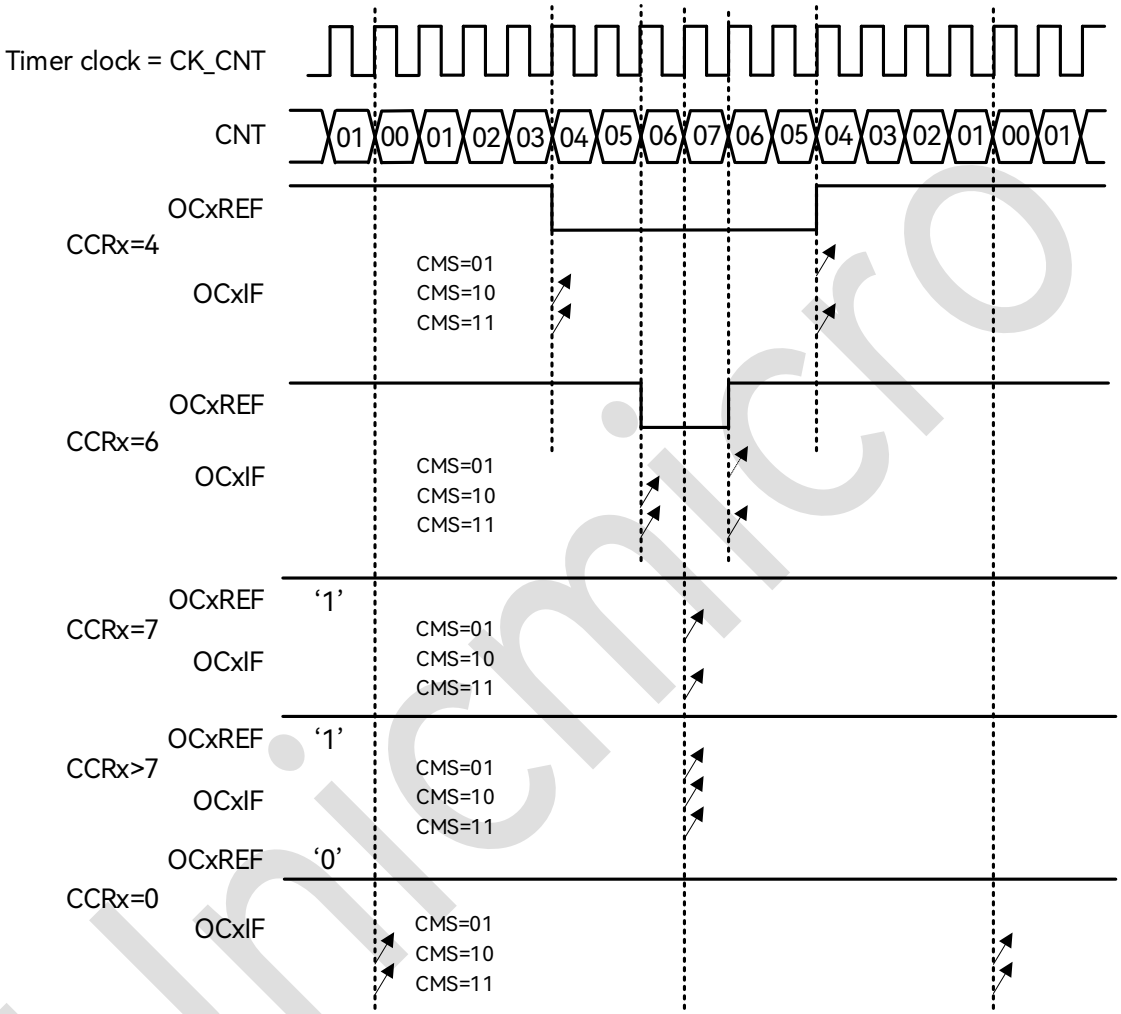


Figure 17-29: Center-aligned PWM Waveform (ARR = 7)

When start counting in center-aligned mode, the initial counting direction is determined by the DIR bit in the register, and in the subsequent process, the DIR bit is directly controlled by hardware. The safest way to use center-aligned mode is to generate an update by setting the UG bit in the register just before starting the counter and not to overwrite the counter while it is running.

17.5.11 One-pulse Output Mode

One-pulse output mode is a particular case of the compare output mode, which allows the counter to generate a pulse with a programmable length after a programmable delay following the occurrence of an event.

Different from other output modes, the counter will stop automatically at the next update event. A pulse can be correctly generated only if the compare value is different from the counter initial value. In up-counting, it is required that CNT < CCR ≤ ARR; in down-counting, it is required that CNT > CCR.

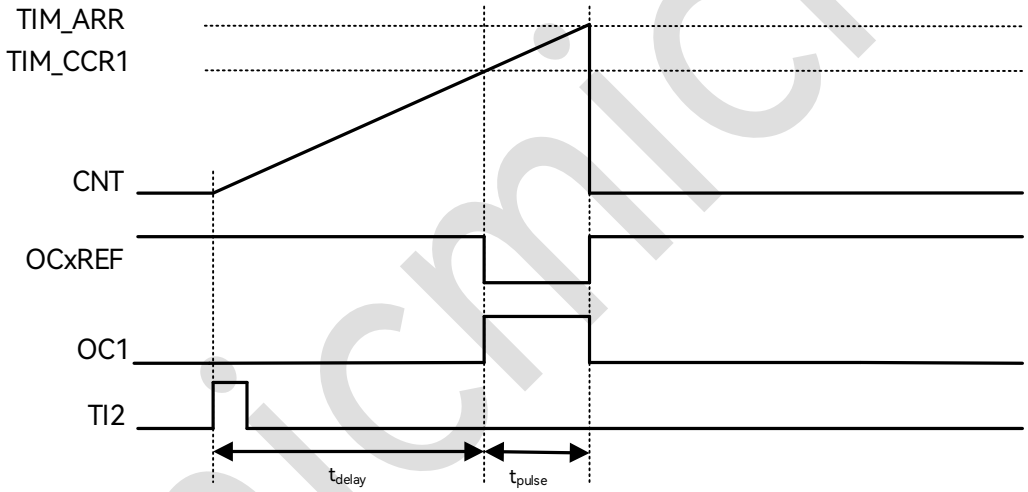


Figure 17-30: Timing Diagram of One-pulse Mode

In the above figure, TI2 input is used as the counter trigger signal. When the count value reaches CCR, the OCxREF outputs a low level. Once the counter counts up to ARR, the OCxREF signal returns to a high level, and the counter rolls back to 0, stopping the counting process.

- The configuration for realizing the above function of TI2 as an input trigger is as follows:

  1. In GPIO module, configure the corresponding pin as TIM_CH2.
  2. Disable the channel by setting TIM_CCER[4] = 0 to ensure the success of subsequent channel configuration.

3. Select the input channel by setting TIM_CCMR1[9:8] = 01.
4. Select the active counting edge by setting TIM_CCER[5] = 0.
5. Select TI2FP2 as the trigger input source by setting TIM_SMCR.TS[2:0] = 110.
6. Set the slave mode controller to trigger mode by setting TIM_SMCR.SMS[2:0] = 110, with TI2FP2 for activating the counter.
7. Enable the channel by setting TIM_CCER[4] = 1.

- The configuration for realizing the above function of OC1 as an output is as follows:
  1. In GPIO module, configure the corresponding pin as TIM_CH1.
  2. Disable the channel by setting TIM_CCER[0] = 0 to ensure the success of subsequent channel configuration.
  3. Select the output channel by setting TIM_CCMR1[1:0] = 00.
  4. Select the active counting edge by setting TIM_ CCMR1.OC1M = 111, in PWM mode 2.
  5. Enable the channel by setting TIM_CCER[0] = 1.
- Special settings for generating OPM waveform timing:
  1. t_{delay} is determined by the value of TIM_CCR1.
  2. t_{pulse} is determined by the difference between TIM_ARR and TIM_CCR1 (TIM_ARR−TIM_CCR1).
  3. Configure to one-pulse mode by setting TIM_CR1.OPM = 1.

17.5.12 Clearing OCxREF Signal on External Event

OCxREF is active at high level, and it can be pulled down directly until the next update event by applying a high level to the external ETR pin. This function can only be used in output

compare and PWM modes, and does not work in forced mode. Enabling this function requires setting OCxCE to 1.

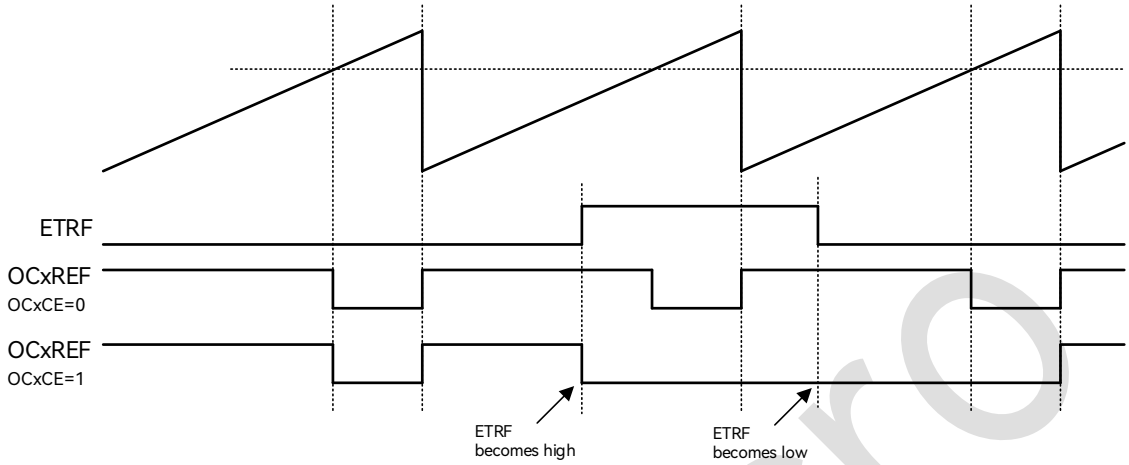


Figure 17-31: ETR Signal Clearing OCxREF of TIM

17.5.13 Encoder Interface Mode

The encoder interface mode involves two external input signals, and TIM determines whether to count up or down according to the level of the edge of one signal relative to the other signal. The following table shows the relationship between the counting mode and the two inputs:

Table 17-2: Counting Direction versus Encoder Signals

Active Edge	Level on Opposite Signal (TI1 for TI2, TI2 for TI1)	TI1 Signal		TI2 Signal	
		Rising	Falling	Rising	Falling
Counting on TI1 only	High	Down	Up	No count	No count
	Low	Up	Down	No count	No count
Counting on TI2 only	High	No count	No count	Up	Down
	Low	No count	No count	Down	Up
Counting on TI1 and TI2	High	Down	Up	Up	Down
	Low	Up	Down	Down	Up

For example, when the counter is counting on TI1, it will count down if TI2 is sampled as high level on the rising edge of TI1, and count up if TI2 is sampled as high level on the falling edge of TI1.

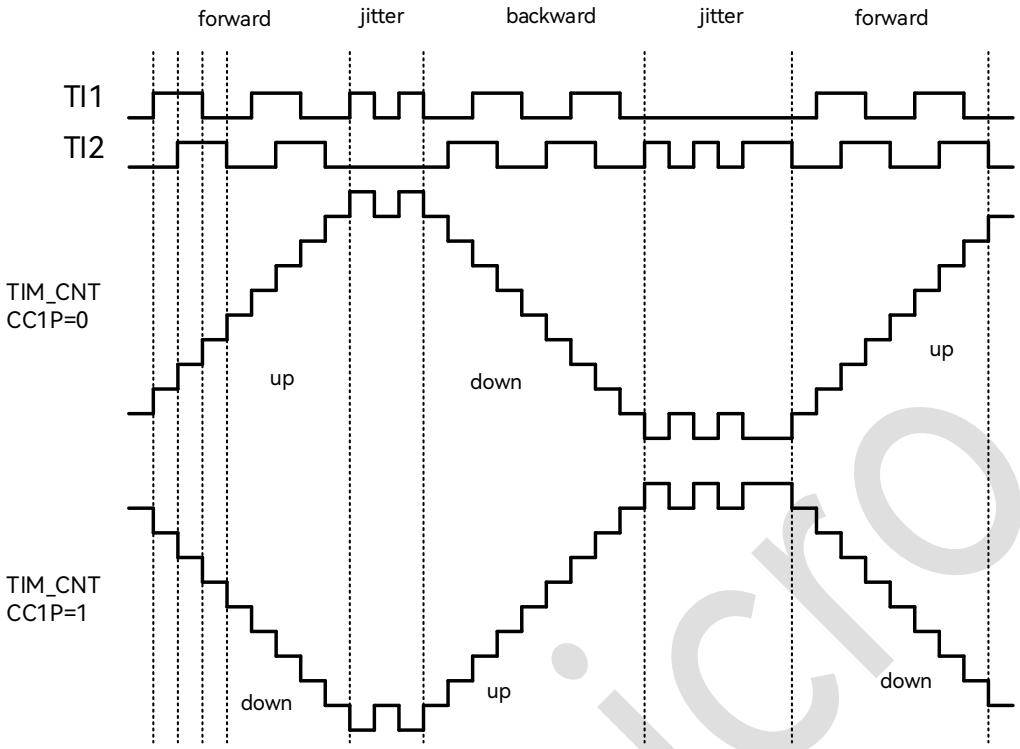


Figure 17-32: Example of Counter Operation in Encoder Interface Mode

The input channels in encoder interface mode shall be set as follows:

1. In GPIO module, configure the corresponding pins as TIM_CH1 and TIM_CH2 functions.
2. Disable the channel by setting TIM_CCER[0] = 0 and TIM_CCER[4] = 0 to ensure the success of subsequent channel configuration.
3. Select the input channel, and configure TIM_CCMR1[1:0] = 01 and TIM_CCMR1[9:8] = 01.
4. Select the active counting edge by setting TIM_CCER[1] = 0 and TIM_CCER[5] = 0.
5. Set the slave mode controller to encoder mode 3 by setting TIM_SMCR.SMS[2:0] = 011.
6. Enable the channel by setting TIM_CCER[0] = 1 and TIM_CCER[4] = 1.

17.5.14 TIM Slave Mode

When TIM is used as a slave (triggered by an external event), it can be configured to operate in three modes: reset mode, gated mode, and trigger mode.

17.5.14.1 Reset Mode

In this mode, all the preload registers in TIM will be reinitialized in response to an event on a trigger input, and the counter will restart from 0. The following figure shows that the counter behaves normally until rising edge is detected on TI1 input, at which time the counter is cleared and restarts from 0.

The configuration in the following figure example is as follows:

1. In GPIO module, configure the corresponding pin as TIM_CH1.
2. Disable the channel by setting TIM_CCER[0] = 0 to ensure the success of subsequent channel configuration.
3. Select the input channel by setting TIM_CCMR1[1:0] = 01.
4. Select the active counting edge by setting TIM_CCER[1] = 0.
5. Select TI1FP1 as the trigger input source by setting TIM_SMCR.TS[2:0] = 101.
6. Configure the slave mode controller to reset mode by setting TIM_SMCR.SMS[2:0] = 100.
7. Enable the channel by setting TIM_CCER[0] = 1.
8. Enable the counter by setting TIM_CR1[0] = 1.

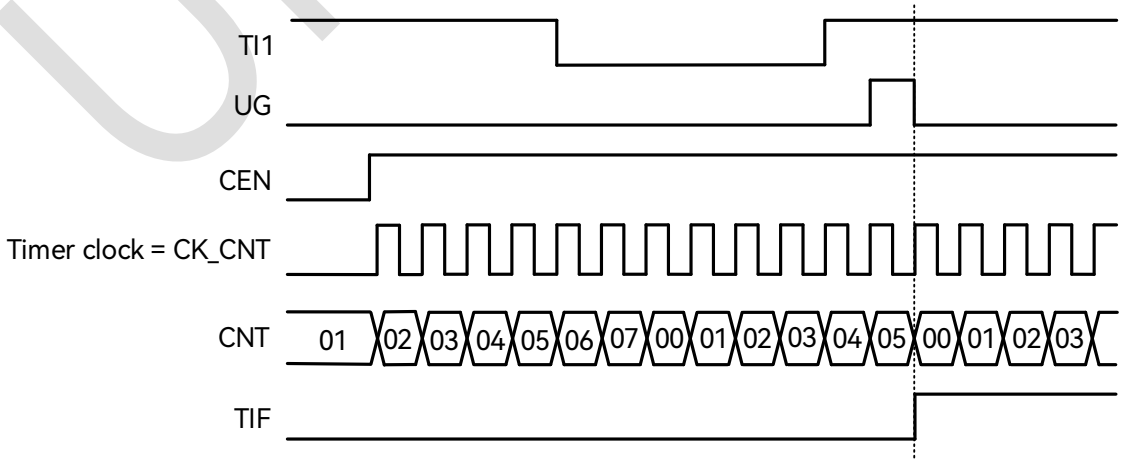


Figure 17-33: Timing Diagram in Reset Mode

17.5.14.2 Gated Mode

In this mode, the counter can be enabled depending on the level of a selected input. The interrupt flag is triggered whenever a level shift causes the counter to start or stop counting.

The configuration in the following figure example is as follows:

1. In GPIO module, configure the corresponding pin as TIM_CH1.
2. Disable the channel by setting TIM_CCER[0] = 0 to ensure the success of subsequent channel configuration.
3. Select the input channel by setting TIM_CCMR1[1:0] = 01.
4. Select the active counting edge by setting TIM_CCER[1] = 0.
5. Select TI1FP1 as the trigger input source by setting TIM_SMCR.TS[2:0] = 101.
6. Configure the slave mode controller to gated mode by setting TIM_SMCR.SMS[2:0] = 101.
7. Enable the channel by setting TIM_CCER[0] = 1.
8. Enable the counter by setting TIM_CR1[0] = 1.

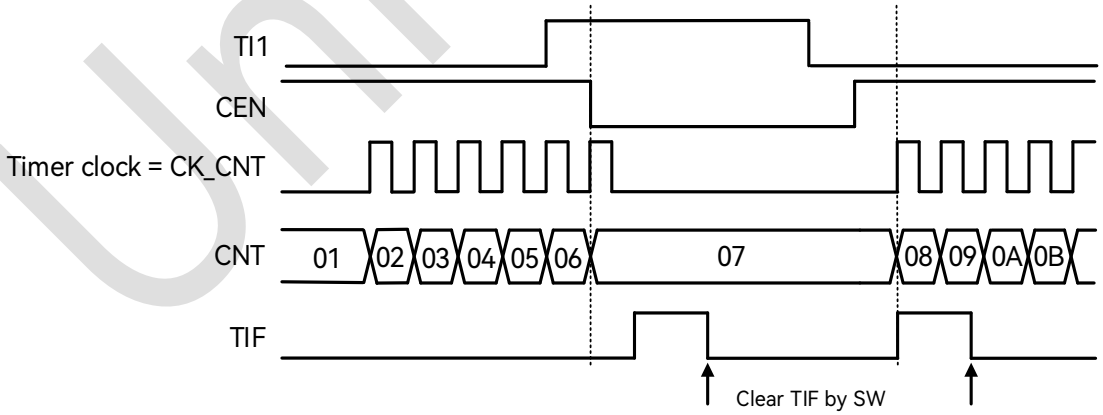


Figure 17-34: Timing Diagram in Gated Mode

17.5.14.3 Trigger Mode

The counter can start in response to an event on a selected input.

The configuration in the following figure example is as follows:

1. In GPIO module, configure the corresponding pin as TIM_CH1.
2. Disable the channel by setting TIM_CCER[0] = 0 to ensure the success of subsequent channel configuration.
3. Select the input channel by setting TIM_CCMR1[1:0] = 01.
4. Select the active counting edge by setting TIM_CCER[1] = 0.
5. Select TI1FP1 as the trigger input source by setting TIM_SMCR.TS[2:0] = 101.
6. Configure the slave mode controller to trigger mode by setting TIM_SMCR.SMS[2:0] = 110.
7. Enable the channel by setting TIM_CCER[0] = 1.

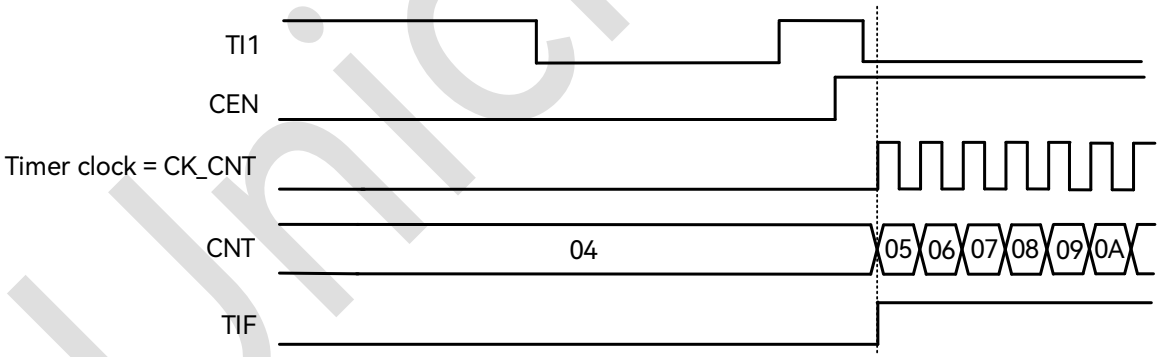


Figure 17-35: Timing Diagram in Trigger Mode

17.5.14.4 External Clock Mode 2 + Trigger Mode

In this mode, ETR can be set as the counting clock, while another external input is used as a trigger signal to start the counter. For instance, the counter begins counting on the rising edge of the ETR input after detecting the rising edge of TI1.

The configuration in the following figure example is as follows:

1. In GPIO module, configure the corresponding pins as TIM_CH1 and TIM_ETR.
2. Select the ETP edge by setting TIM_SMCR[15] = 0.
3. Set the ETR division ratio by setting TIM_SMCR.ETPS[1:0] = 01.
4. Configure the input filter duration by setting TIM_SMCR.ETF[3:0] = 0000.
5. Set the ECE register and enable the external clock mode 2 by setting TIM_SMCR[14] = 1.
6. Disable the channel by setting TIM_CCER[0] = 0 to ensure the success of subsequent channel configuration.
7. Select the input channel by setting TIM_CCMR1[1:0] = 01.
8. Select the active counting edge by setting TIM_CCER[1] = 0.
9. Select TI1FP1 as the trigger input source by setting TIM_SMCR.TS[2:0] = 101.
10. Configure the slave mode controller to trigger mode by setting TIM_SMCR.SMS[2:0] = 110.
11. Enable the channel by setting TIM_CCER[0] = 1.

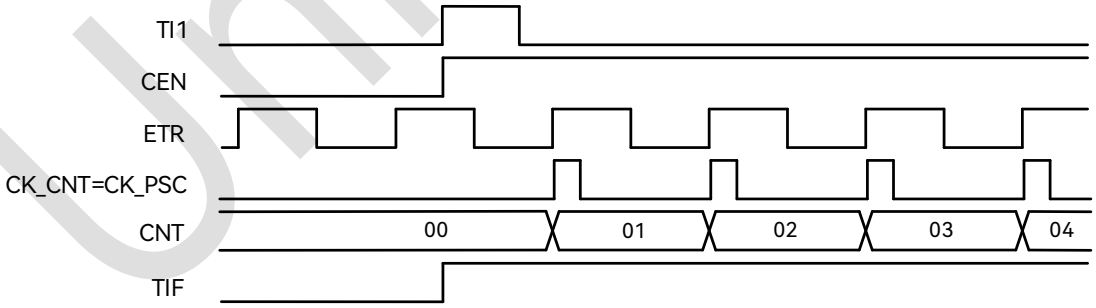


Figure 17-36: Timing Diagram in External Clock Mode 2 + Trigger Mode

17.5.15 Timer Synchronization

Timers can be cascaded together through trigger events to achieve synchronization or cascading operation. A timer can utilize four internal trigger inputs, allowing the trigger signal output from one timer to connect to the internal trigger input of other timers.

17.5.16 DMA Access

TIM supports 7 types of DMA requests, namely 4 CC channel requests, external trigger request, user software trigger request and COM trigger request.

Each CC channel generates a DMA request, which is used to transfer the content of CCRx to RAM in capture mode, and to write the data in RAM to CCRx in compare mode.

In addition, DMA requests can also be generated from external trigger event, software trigger event and COM trigger event, and at the occurrence of these requests, DMA burst transfer will be started to write data to one or more registers within TIM or to read one or more register values from TIM.

Table 17-3: Seven DMA Requests Supported by TIM

DMA Request	DMA Access Object	Single-transfer Length
TIM_CH1	Read DMAR	DBL
	Write DMAR	
TIM_CH2	Read DMAR	DBL
	Write DMAR	
TIM_CH3	Read DMAR	DBL
	Write DMAR	
TIM_CH4	Read DMAR	DBL
	Write DMAR	
TIM_TRIG	Read DMAR	DBL
	Write DMAR	
TIM_UEV	Read DMAR	DBL
	Write DMAR	
TIM_COM	Read DMAR	DBL
	Write DMAR	

17.5.17 DMA Burst

TIM supports DMA and DMA-burst access. A DMA request can be generated at a specific event, so as to write the capture result in CCR to RAM or write the content of one or more registers in RAM to the preload register in TIM.

DMA-burst allows to generate multiple successive DMA requests upon a single event. The main purpose is to update the content of multiple registers in a row each time a given timer event is triggered, thus making it possible to dynamically modify the output waveform in real time.

The DMA controller destination is unique and must be directed to the virtual register TIM_DMAR. On a given timer event, the timer launches a sequence of DMA requests (burst). Each DMA write access to the TIM_DMAR register will be redirected to the actual function register by TIM.

The DBL bits in the register set the DMA burst length, and the DBA bits define the base address for DMA access to TIM (an offset starting from the address of the TIM_CR register).

In DMA-burst mode, all DMA access shall be directed to the virtual register DMAR, and TIM automatically accumulates the internal offset address according to the access. The DBA bits in the register are used to specify the destination address of the first DMA transfer within TIM, while the DBL bits are used to specify the burst length.

17.5.18 Input XOR Function

The input XOR function allows the input signals from channels 1−3 to be XORed together and then connected to the input of the filtering and edge detection circuit of channel 1, which can be used for input capture or triggering on channel 1.

The TI1S bit in the TIM_CR2 register is used to select whether the input to channel 1 comes from the XOR of the three channel inputs.

17.5.19 Debug Mode

When the CPU enters debug mode, the timer can either stop or continue working, and its behavior is defined by registers in the chip system.

When the timer is stopped during debugging, its output will be disabled (MOE is cleared). Depending on the register configuration, the output signal can be forced to be inactive or controlled by the GPIO module.

17.6 Register Description

TIM1 register base address: 0x4700_A000

TIM2 register base address: 0x4700_A400

TIM3 register base address: 0x4700_A800

TIM4 register base address: 0x4700_AC00

TIM8 register base address: 0x4700_8000

TIM9 register base address: 0x4700_9000

The registers are listed below:

Table 17-4: General-purpose Timers (TIM1−TIM4 & TIM8−TIM9)

Offset Address	Name	Description
0x00	TIM_CR1	Control register 1
0x04	TIM_CR2	Control register 2
0x08	TIM_SMCR	Slave mode control register
0x0C	TIM_DIER	DMA and interrupt enable register
0x10	TIM_SR	Status register
0x14	TIM_EGR	Event generation register
0x18	TIM_CCMR1	Capture/compare mode register 1
0x1C	TIM_CCMR2	Capture/compare mode register 2
0x20	TIM_CCER	Capture/compare enable register

Offset Address	Name	Description
0x24	TIM_CNT	Counter register
0x28	TIM_PSC	Prescaler register
0x2C	TIM_ARR	Auto-reload register
0x34	TIM_CCR1	Capture/compare register 1
0x38	TIM_CCR2	Capture/compare register 2
0x3C	TIM_CCR3	Capture/compare register 3
0x40	TIM_CCR4	Capture/compare register 4
0x48	TIM_DCR	DMA control register
0x4C	TIM_DMAR	DMA access register

Registers are detailed in the following sections.

17.6.1 Control Register 1 (TIM_CR1)

Offset address: 0x00

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:10	RSV	-	-	Reserved
9:8	CKD	R/W	0x0	Dead time and digital filter clock frequency division register (division ratio relative to CK_INT): 00: $t_{DTS} = t_{CK\_INT}$ 01: $t_{DTS} = 2 * t_{CK\_INT}$ 10: $t_{DTS} = 4 * t_{CK\_INT}$ 11: reserved, prohibited
7	ARPE	R/W	0x0	Auto-reload preload enable: 0: ARR not preloaded 1: ARR preloaded
6:5	CMS	R/W	0x0	Counter alignment mode selection: 00: edge-aligned mode 01: center-aligned mode 1; output compare interrupt flags are set only when the counter is counting down. 10: center-aligned mode 2; output compare interrupt flags are set only when the counter is counting up. 11: center-aligned mode 3; output compare interrupt flags are set both when the counter is counting up or

Bit	Name	Attribute	Reset Value	Description
				down.
4	DIR	R/W	0x0	Counting direction register: 0: count up 1: count down Note: This register is read-only when the timer is configured in center-aligned mode or encoder mode.
3	OPM	R/W	0x0	One-pulse mode output: 0: the counter does not stop at the occurrence of update event. 1: the counter stops at the occurrence of update event (CEN cleared automatically).
2	URS	R/W	0x0	Update request source: 0: an update interrupt or DMA request will be generated by any of the following events: ● Counter overflow/underflow ● Software setting the UG bit ● Update generated from the slave mode controller 1: an update interrupt or DMA request will be generated only at counter overflow or underflow.
1	UDIS	R/W	0x0	Update disable: 0: update event enabled; the update event can be generated by any of the following events: ● Counter overflow/underflow ● Software setting the UG bit ● Update generated from the slave mode controller 1: update event disabled, shadow register not updated The counter and the prescaler will be reinitialized if the UG bit is set or if the slave mode controller receives a hardware reset.
0	CEN	R/W	0x0	Counter enable: 0: disabled 1: enabled Note: The external trigger mode can automatically set the CEN bit.

17.6.2 Control Register 2 (TIM_CR2)

Offset address: 0x04

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:8	RSV	-	-	Reserved
7	TI1S	R/W	0x0	T1 input selection: 0: T1 input from CH1 pin 1: T1 input from XOR combination of CH1, CH2 and CH3 pins
6:4	MMS	R/W	0x0	Master mode selection, selecting the TRGO trigger mode: 000: reset—TRGO is generated by the UG bit in the EGR register. 001: enable—TRGO is generated by the counter enable signal, including the CEN control bit and external trigger. 010: update—TRGO is generated by the update event. 011: compare pulse—TRGO is generated when an input capture or compare event occurs that sets CC1F to 1. 100: compare—TRGO is generated by OC1REF. 101: compare—TRGO is generated by OC2REF. 110: compare—TRGO is generated by OC3REF. 111: compare—TRGO is generated by OC4REF.
3	CCDS	R/W	0x0	Capture/compare DMA selection: 0: CCx DMA request sent when CCx event occurs 1: CCx DMA request sent when update event occurs
2:0	RSV	-	-	Reserved

17.6.3 Slave Mode Control Register (TIM_SMCR)

Offset address: 0x08

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:16	RSV	-	-	Reserved
15	ETP	R/W	0x0	External trigger polarity: 0: active at high level or rising edge 1: active at low level or falling edge
14	ECE	R/W	0x0	Enable clock enable: 0: external clock mode 2 disabled 1: external clock mode 2 enabled; the counter is clocked by any active edge on the ETRF signal.
13:12	ETPS	R/W	0x0	External trigger prescaler: The frequency of external trigger signal ETRP must be at most 1/4 of TIM clock frequency. A prescaler can be enabled to reduce ETRP frequency when inputting fast external clocks. 00: prescaler off 01: divided by 2 10: divided by 4 11: divided by 8
11:8	ETF	R/W	0x0	External trigger filter frequency and length selection: 0000: no filter 0001: $f_{SAMPLING} = f_{CK\_INT}$, N = 2 0010: $f_{SAMPLING} = f_{CK\_INT}$, N = 4 0011: $f_{SAMPLING} = f_{CK\_INT}$, N = 8 0100: $f_{SAMPLING} = f_{DTS} / 2$, N = 6 0101: $f_{SAMPLING} = f_{DTS} / 2$, N = 8 0110: $f_{SAMPLING} = f_{DTS} / 4$, N = 6 0111: $f_{SAMPLING} = f_{DTS} / 4$, N = 8 1000: $f_{SAMPLING} = f_{DTS} / 8$, N = 6 1001: $f_{SAMPLING} = f_{DTS} / 8$, N = 8 1010: $f_{SAMPLING} = f_{DTS} / 16$, N = 5 1011: $f_{SAMPLING} = f_{DTS} / 16$, N = 6

Bit	Name	Attribute	Reset Value	Description
				1100: $f_{SAMPLING} = f_{DTS} / 16$, N = 8 1101: $f_{SAMPLING} = f_{DTS} / 32$, N = 5 1110: $f_{SAMPLING} = f_{DTS} / 32$, N = 6 1111: $f_{SAMPLING} = f_{DTS} / 32$, N = 8
7	MSM	R/W	0x0	Master/slave mode selection: 0: no action 1: the effect of an event on the trigger input (TRGI) is delayed to allow a perfect synchronization between the current timer and its slaves (through TRGO).
6:4	TS	R/W	0x0	Trigger selection for selecting the trigger source to be used to synchronize the counter: 000: internal trigger 0 (ITR0) 001: internal trigger 1 (ITR1) 010: internal trigger 2 (ITR2) 011: internal trigger 3 (ITR3) 100: TI1 edge detector (TI1F_ED) 101: filtered timer input 1 (TI1FP1) 110: filtered timer input 2 (TI2FP2) 111: external trigger input (ETRF) Note: These bits can be changed only when the slave mode is disabled (i.e. SMS = 000).
3	RSV	-	-	Reserved
2:0	SMS	R/W	0x0	Slave mode selection: 000: slave mode disabled— if CEN is enabled, then the prescaler is clocked directly by the internal clock. 001: encoder mode 1— counter counts up/down on TI2FP1 edge depending on TI1FP2 level. 010: encoder mode 2— counter counts up/down on TI1FP2 edge depending on TI2FP1 level. 011: encoder mode 3— counter counts up/down on both TI1FP1 and TI2FP2 edges depending on the level of other inputs. 100: reset mode— rising edge of the selected trigger input (TRGI) reinitializes the counter and generates an update of the registers.

Bit	Name	Attribute	Reset Value	Description
				101: gated mode— the counter clock is enabled when TRGI is high, and stops as soon as it becomes low. 110: trigger mode— the counter starts at the rising edge of TRGI (but it is not reset). 111: external clock mode 1— rising edges of TRGI directly clock the counter.

17.6.4 DMA / Interrupt Enable Register (TIM_DIER)

Offset address: 0x0C

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:20	RSV	-	-	Reserved
19	CC4OF_DISABLE	R/W	0x0	CC4OF interrupt enable: 0: enabled 1: disabled
18	CC3OF_DISABLE	R/W	0x0	CC3OF interrupt enable: 0: enabled 1: disabled
17	CC2OF_DISABLE	R/W	0x0	CC2OF interrupt enable: 0: enabled 1: disabled
16	CC1OF_DISABLE	R/W	0x0	CC1OF interrupt enable: 0: enabled 1: disabled
15	RSV	-	-	Reserved
14	TDE	R/W	0x0	External trigger DMA request enable: 0: in slave mode, external trigger DMA request disabled 1: in slave mode, external trigger DMA request enabled (can be used to automatically update the preload register)
13	RSV	-	-	Reserved

Bit	Name	Attribute	Reset Value	Description
12	CC4DE	R/W	0x0	Capture/compare channel 4 DMA request enable: 0: CC4 DMA request disabled 1: CC4 DMA request enabled
11	CC3DE	R/W	0x0	Capture/compare channel 3 DMA request enable: 0: CC3 DMA request disabled 1: CC3 DMA request enabled
10	CC2DE	R/W	0x0	Capture/compare channel 2 DMA request enable: 0: CC2 DMA request disabled 1: CC2 DMA request enabled
9	CC1DE	R/W	0x0	Capture/compare channel 1 DMA request enable: 0: CC1 DMA request disabled 1: CC1 DMA request enabled
8	UDE	R/W	0x0	Update DMA request enable: 0: update DMA request disabled 1: update DMA request enabled
7	RSV	-	-	Reserved
6	TIE	R/W	0x0	Trigger interrupt enable: 0: trigger interrupt disabled 1: trigger interrupt enabled
5	RSV	-	-	Reserved
4	CC4IE	R/W	0x0	Capture/compare channel 4 interrupt enable: 0: CC4 interrupt disabled 1: CC4 interrupt enabled
3	CC3IE	R/W	0x0	Capture/compare channel 3 interrupt enable: 0: CC3 interrupt disabled 1: CC3 interrupt enabled
2	CC2IE	R/W	0x0	Capture/compare channel 2 interrupt enable: 0: CC2 interrupt disabled 1: CC2 interrupt enabled

Bit	Name	Attribute	Reset Value	Description
1	CC1IE	R/W	0x0	Capture/compare channel 1 interrupt enable: 0: CC1 interrupt disabled 1: CC1 interrupt enabled
0	UIE	R/W	0x0	Update interrupt enable: 0: update interrupt disabled 1: update interrupt enabled

17.6.5 Status Register (TIM_SR)

Offset address: 0x10

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:13	RSV	-	-	Reserved
12	CC4OF	R/W0C	0x0	Capture/compare channel 4 overcapture flag: This register is valid only when the corresponding channel is configured in input capture mode. This flag is set by hardware and cleared by software via writing it to 0. 0: no overcapture detected 1: a new capture occurs while CC4IF flag is 1
11	CC3OF	R/W0C	0x0	Capture/compare channel 3 overcapture flag: This register is valid only when the corresponding channel is configured in input capture mode. This flag is set by hardware and cleared by software via writing it to 0. 0: no overcapture detected 1: a new capture occurs while CC3IF flag is 1
10	CC2OF	R/W0C	0x0	Capture/compare channel 2 overcapture flag: This register is valid only when the corresponding channel is configured in input capture mode. This flag is set by hardware and cleared by software via writing it to 0. 0: no overcapture detected 1: a new capture occurs while CC2IF flag is 1
9	CC1OF	R/W0C	0x0	Capture/compare channel 1 overcapture flag:

Bit	Name	Attribute	Reset Value	Description
				This register is valid only when the corresponding channel is configured in input capture mode. This flag is set by hardware and cleared by software via writing it to 0. 0: no overcapture detected 1: a new capture occurs while CC1IF flag is 1
8:7	RSV	-	-	Reserved
6	TIF	R/W0C	0x0	Trigger interrupt flag is set by hardware and cleared by software via writing it to 0.
5	RSV	-	-	Reserved
4	CC4IF	R/W0C	0x0	Capture/compare channel 4 interrupt flag: If channel CC4 is configured as output: the CC4IF flag is set when the counter matches the compare value, and cleared by software via writing it to 0. If channel CC4 is configured as input: this flag is set by hardware on a capture, and cleared by software via writing it to 0 or automatically cleared by software reading TIM_CCR4.
3	CC3IF	R/W0C	0x0	Capture/compare channel 3 interrupt flag: If channel CC3 is configured as output: the CC3IF flag is set when the counter matches the compare value, and cleared by software via writing it to 0. If channel CC3 is configured as input: this flag is set by hardware on a capture, and cleared by software via writing it to 0 or automatically cleared by software reading TIM_CCR3.
2	CC2IF	R/W0C	0x0	Capture/compare channel 2 interrupt flag: If channel CC2 is configured as output: the CC2IF flag is set when the counter matches the compare value, and cleared by software via writing it to 0. If channel CC2 is configured as input: this flag is set by hardware on a capture, and cleared by software via writing it to 0 or automatically cleared by software reading TIM_CCR2.

Bit	Name	Attribute	Reset Value	Description
1	CC1IF	R/W0C	0x0	Capture/compare channel 1 interrupt flag: If channel CC1 is configured as output: the CC1IF flag is set when the counter matches the compare value, and cleared by software via writing it to 0. If channel CC1 is configured as input: this flag is set by hardware on a capture, and cleared by software via writing it to 0 or automatically cleared by software reading TIM_CCR1.
0	UIF	R/W0C	0x0	Update event interrupt flag is set by hardware and cleared by software via writing it to 0. UIF is set and the shadow register is updated at the following events: ● Counter overflow occurs if repetition counter = 0 and UDIS = 0. ● The counter is reinitialized by software setting the UG bit if URS = 0 and UDIS = 0. ● The counter is reinitialized by a trigger event if URS = 0 and UDIS = 0.

17.6.6 Event Generation Register (TIM_EGR)

Offset address: 0x14

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:7	RSV	-	-	Reserved
6	TG	W	0x0	This bit is set by software to generate a trigger event, and it is automatically cleared by hardware.
5	RSV	-	-	Reserved
4	CC4G	W	0x0	Capture/compare channel 4 generation If channel CC4 is configured as output, CC4IF flag is set, and corresponding interrupt and DMA request will be sent if enabled. If channel CC4 is configured as input, the current counter value is captured in TIM_CCR4 register, the CC4IF flag is set, and corresponding interrupt

Bit	Name	Attribute	Reset Value	Description
				and DMA request will be sent if enabled.
3	CC3G	W	0x0	Capture/compare channel 3 generation If channel CC3 is configured as output, CC3IF flag is set, and corresponding interrupt and DMA request will be sent if enabled. If channel CC3 is configured as input, the current counter value is captured in TIM_CCR3 register, the CC3IF flag is set, and corresponding interrupt and DMA request will be sent if enabled.
2	CC2G	W	0x0	Capture/compare channel 2 generation If channel CC2 is configured as output, CC2IF flag is set, and corresponding interrupt and DMA request will be sent if enabled. If channel CC2 is configured as input, the current counter value is captured in TIM_CCR2 register, the CC2IF flag is set, and corresponding interrupt and DMA request will be sent if enabled.
1	CC1G	W	0x0	Capture/compare channel 1 generation If channel CC1 is configured as output, CC1IF flag is set, and corresponding interrupt and DMA request will be sent if enabled. If channel CC1 is configured as input, the current counter value is captured in TIM_CCR1 register, the CC1IF flag is set, and corresponding interrupt and DMA request will be sent if enabled.
0	UG	W	0x0	This bit can be set by software to generate an update event, and is automatically cleared by hardware. When the UG bit is set by software, the counter is reinitialized, the shadow register is updated, and the prescaler counter is cleared.

17.6.7 Capture/Compare Mode Register 1 (TIM_CCMR1)

Offset address: 0x18

Reset value: 0x0000 0000

This register can be used for output compare mode or for input capture mode.

- Output compare mode

Bit	Name	Attribute	Reset Value	Description
31:16	RSV	-	-	Reserved
15	OC2CE	R/W	0x0	Output compare 2 clear enable, refer to OC1CE description
14:12	OC2M	R/W	0x0	Output compare 2 mode configuration, refer to OC1M description
11	OC2PE	R/W	0x0	Output compare 2 preload enable, refer to OC1PE description
10	OC2FE	R/W	0x0	Output compare 2 fast enable, refer to OC1FE description
9:8	CC2S	R/W	0x0	Capture/compare channel 2 selection: 00: CC2 channel is configured as output. 01: CC2 channel is configured as input, IC2 is mapped on TI2. 10: CC2 channel is configured as input, IC2 is mapped on TI1. 11: CC2 channel is configured as input, IC2 is mapped on TRC. Note: CC2S bits are writable only when the channel is OFF (CC2E = 0).
7	OC1CE	R/W	0x0	Output compare 1 clear enable: 0: OC1REF is not affected by ETRF input. 1: OC1REF is automatically cleared once a high level is detected on ETRF input.
6:4	OC1M	R/W	0x0	Output compare 1 mode: these bits define the behavior of the output reference signal OC1REF. 000: the comparison between the output compare register CCR1 and the counter CNT

Bit	Name	Attribute	Reset Value	Description
				has no effect on the outputs. 001: set OC1REF high when CCR1 = CNT (falling edge) 010: set OC1REF low when CCR1 = CNT (falling edge) 011: toggle OC1REF when CCR1 = CNT (falling edge) 100: force OC1REF low (inactive) 101: force OC1REF high (active) 110: PWM mode 1—in up-counting, OC1REF is set high when CNT < CCR1, otherwise it is set low; in down-counting, OC1REF is set low when CNT ≥ CCR1, otherwise it is set high. 111: PWM mode 2—in up-counting, OC1REF is set low when CNT < CCR1, otherwise it is set high; in down-counting, OC1REF is set high when CNT ≥ CCR1, otherwise it is set low.
3	OC1PE	R/W	0x0	Output compare 1 preload enable: 0: preload register on CCR1 disabled; CCR1 can be written directly. 1: preload register on CCR1 enabled; read/write operations access the preload register; the preload value is shifted to the shadow register at each update event.
2	OC1FE	R/W	0x0	Output compare 1 fast enable: 0: fast disabled, the trigger input will not affect the comparison output. 1: fast enabled, the trigger input will immediately change OC1REF to the output when the comparison values match, regardless of the actual current comparison. This function acts only if the channel is configured in PWM1 or PWM2 mode.
1:0	CC1S	R/W	0x0	Capture/compare channel 1 selection: 00: CC1 channel is configured as output. 01: CC1 channel is configured as input, IC1 is mapped on TI1.

Bit	Name	Attribute	Reset Value	Description
				10: CC1 channel is configured as input, IC1 is mapped on TI2. 11: CC1 channel is configured as input, IC1 is mapped on TRC. Note: CC1S bits are writable only when the channel is OFF (CC1E = 0).

- Input capture mode

Bit	Name	Attribute	Reset Value	Description
31:16	RSV	-	-	Reserved
15:12	IC2F	R/W	0x0	Input capture 2 filter
11:10	IC2PSC	R/W	0x0	Input capture 2 prescaler
9:8	CC2S	R/W	0x0	Capture/compare channel 2 selection: 00: CC2 channel is configured as output. 01: CC2 channel is configured as input, IC2 is mapped on TI2. 10: CC2 channel is configured as input, IC2 is mapped on TI1. 11: CC2 channel is configured as input, IC2 is mapped on TRC. Note: CC2S bits are writable only when the channel is OFF (CC2E = 0).
7:4	IC1F	R/W	0x0	Input capture 1 filter This bit-field defines the frequency used to sample TI1 input and the length of the digital filter applied to TI1. 0000: no filter, sampling is done at f_{DTS} 0001: $f_{SAMPLING} = f_{CK\_INT}$, N = 2 0010: $f_{SAMPLING} = f_{CK\_INT}$, N = 4 0011: $f_{SAMPLING} = f_{CK\_INT}$, N = 8 0100: $f_{SAMPLING} = f_{DTS} / 2$, N = 6 0101: $f_{SAMPLING} = f_{DTS} / 2$, N = 8 0110: $f_{SAMPLING} = f_{DTS} / 4$, N = 6 0111: $f_{SAMPLING} = f_{DTS} / 4$, N = 8 1000: $f_{SAMPLING} = f_{DTS} / 8$, N = 6 1001: $f_{SAMPLING} = f_{DTS} / 8$, N = 8

Bit	Name	Attribute	Reset Value	Description
				1010: $f_{SAMPLING} = f_{DTS} / 16$, N = 5 1011: $f_{SAMPLING} = f_{DTS} / 16$, N = 6 1100: $f_{SAMPLING} = f_{DTS} / 16$, N = 8 1101: $f_{SAMPLING} = f_{DTS} / 32$, N = 5 1110: $f_{SAMPLING} = f_{DTS} / 32$, N = 6 1111: $f_{SAMPLING} = f_{DTS} / 32$, N = 8
3:2	IC1PSC	R/W	0x0	Input capture 1 prescaler: 00: no prescaler 01: capture is done once every 2 events 10: capture is done once every 4 events 11: capture is done once every 8 events The IC1PSC register is reset when CC1E = 0.
1:0	CC1S	R/W	0x0	Capture/compare channel 1 selection: 00: CC1 channel is configured as output. 01: CC1 channel is configured as input, IC1 is mapped on TI1. 10: CC1 channel is configured as input, IC1 is mapped on TI2. 11: CC1 channel is configured as input, IC1 is mapped on TRC. Note: CC1S bits are writable only when the channel is OFF (CC1E = 0).

17.6.8 Capture/Compare Mode Register 2 (TIM_CCMR2)

Offset address: 0x1C

Reset value: 0x0000 0000

This register can be used for output compare mode or for input capture mode.

- Output compare mode

Bit	Name	Attribute	Reset Value	Description
31:16	RSV	-	-	Reserved
15	OC4CE	R/W	0x0	Output compare 4 clear enable: 0: OC4REF is not affected by ETRF input. 1: OC4REF is automatically cleared once a high level is detected on ETRF input.

Bit	Name	Attribute	Reset Value	Description
14:12	OC4M	R/W	0x0	Output compare 4 mode configuration: These bits define the behavior of the output reference signal OC4REF. 000: the comparison between the output compare register CCR4 and the counter CNT has no effect on the outputs. 001: set OC4REF high when CCR4 = CNT. 010: set OC4REF low when CCR4 = CNT. 011: toggle OC4REF when CCR4 = CNT. 100: force OC4REF low (inactive) 101: force OC4REF high (active) 110: PWM mode 1 ● In up-counting, OC4REF is set high when CNT < CCR4, otherwise it is set low. ● In down-counting, OC4REF is set low when CNT > CCR4, otherwise it is set high. 111: PWM mode 2 ● In up-counting, OC4REF is set low when CNT < CCR4, otherwise it is set high. ● In down-counting, OC4REF is set high when CNT > CCR4, otherwise it is set low.
11	OC4PE	R/W	0x0	Output compare 4 preload enable: 0: preload register on CCR4 disabled; CCR4 can be written directly. 1: preload register on CCR4 enabled; read/write operations access the preload register; the preload value is shifted to the shadow register at each update event.
10	OC4FE	R/W	0x0	Output compare 4 fast enable: 0: fast disabled, the trigger input will not affect the comparison output. 1: fast enabled, the trigger input will immediately change OC4REF to the output when the comparison values match, regardless

Bit	Name	Attribute	Reset Value	Description
				of the actual current comparison. This function acts only if the channel is configured in PWM1 or PWM2 mode.
9:8	CC4S	R/W	0x0	Capture/compare channel 4 selection: 00: CC4 channel is configured as output. 01: CC4 channel is configured as input, IC4 is mapped on TI4. 10: CC4 channel is configured as input, IC4 is mapped on TI3. 11: CC4 channel is configured as input, IC4 is mapped on TRC. Note: CC4S bits are writable only when the channel is OFF (CC4E = 0).
7	OC3CE	R/W	0x0	Output compare 3 clear enable: 0: OC3REF is not affected by ETRF input. 1: OC3REF is automatically cleared once a high level is detected on ETRF input.
6:4	OC3M	R/W	0x0	Output compare 3 mode: these bits define the behavior of the output reference signal OC3REF. 000: the comparison between the output compare register CCR3 and the counter CNT has no effect on the outputs. 001: set OC3REF high when CCR3 = CNT. 010: set OC3REF low when CCR3 = CNT. 011: toggle OC3REF when CCR3 = CNT. 100: force OC3REF low (inactive) 101: force OC3REF high (active) 110: PWM mode 1—in up-counting, OC3REF is set high when CNT < CCR3, otherwise it is set low; in down-counting, OC3REF is set low when CNT > CCR3, otherwise it is set high. 111: PWM mode 2—in up-counting, OC3REF is set low when CNT < CCR3, otherwise it is set high; in down-counting, OC3REF is set high when CNT > CCR3, otherwise it is set low.
3	OC3PE	R/W	0x0	Output compare 3 preload enable:

Bit	Name	Attribute	Reset Value	Description
				0: preload register on CCR3 disabled; CCR3 can be written directly. 1: preload register on CCR3 enabled; read/write operations access the preload register; the preload value is shifted to the shadow register at each update event.
2	OC3FE	R/W	0x0	Output compare 3 fast enable: 0: fast disabled, the trigger input will not affect the comparison output. 1: fast enabled, the trigger input will immediately change OC3REF to the output when the comparison values match, regardless of the actual current comparison. This function acts only if the channel is configured in PWM1 or PWM2 mode.
1:0	CC3S	R/W	0x0	Capture/compare channel 3 selection: 00: CC3 channel is configured as output. 01: CC3 channel is configured as input, IC3 is mapped on TI3. 10: CC3 channel is configured as input, IC3 is mapped on TI4. 11: CC3 channel is configured as input, IC3 is mapped on TRC. Note: CC3S bits are writable only when the channel is OFF (CC3E = 0).

- Input capture mode

Bit	Name	Attribute	Reset Value	Description
31:16	RSV	-	-	Reserved
15:12	IC4F	R/W	0x0	Input capture 4 filter: this bit-field defines the frequency used to sample TI4 input and the length of the digital filter applied to TI4: 0000: no filter, sampling is done at f_{DTS} 0001: $f_{SAMPLING} = f_{CK\_INT}$, N = 2 0010: $f_{SAMPLING} = f_{CK\_INT}$, N = 4 0011: $f_{SAMPLING} = f_{CK\_INT}$, N = 8 0100: $f_{SAMPLING} = f_{DTS} / 2$, N = 6

Bit	Name	Attribute	Reset Value	Description
				0101: $f_{SAMPLING} = f_{DTS} / 2$, N = 8 0110: $f_{SAMPLING} = f_{DTS} / 4$, N = 6 0111: $f_{SAMPLING} = f_{DTS} / 4$, N = 8 1000: $f_{SAMPLING} = f_{DTS} / 8$, N = 6 1001: $f_{SAMPLING} = f_{DTS} / 8$, N = 8 1010: $f_{SAMPLING} = f_{DTS} / 16$, N = 5 1011: $f_{SAMPLING} = f_{DTS} / 16$, N = 6 1100: $f_{SAMPLING} = f_{DTS} / 16$, N = 8 1101: $f_{SAMPLING} = f_{DTS} / 32$, N = 5 1110: $f_{SAMPLING} = f_{DTS} / 32$, N = 6 1111: $f_{SAMPLING} = f_{DTS} / 32$, N = 8
11:10	IC4PSC	R/W	0x0	Input capture 4 prescaler: 00: no prescaler 01: capture is done once every 2 events 10: capture is done once every 4 events 11: capture is done once every 8 events The IC4PSC register is reset when CC4E = 0.
9:8	CC4S	R/W	0x0	Capture/compare channel 4 selection: 00: CC4 channel is configured as output. 01: CC4 channel is configured as input, IC4 is mapped on TI4. 10: CC4 channel is configured as input, IC4 is mapped on TI3. 11: CC4 channel is configured as input, IC4 is mapped on TRC. Note: CC4S bits are writable only when the channel is OFF (CC4E = 0).
7:4	IC3F	R/W	0x0	Input capture 3 filter: this bit-field defines the frequency used to sample TI3 input and the length of the digital filter applied to TI3: 0000: no filter, sampling is done at f_{DTS} 0001: $f_{SAMPLING} = f_{CK\_INT}$, N = 2 0010: $f_{SAMPLING} = f_{CK\_INT}$, N = 4 0011: $f_{SAMPLING} = f_{CK\_INT}$, N = 8 0100: $f_{SAMPLING} = f_{DTS} / 4$, N = 6 0111: $f_{SAMPLING} = f_{DTS} / 4$, N = 8 1000: $f_{SAMPLING} = f_{DTS} / 8$, N = 6

Bit	Name	Attribute	Reset Value	Description
				1001: $f_{SAMPLING} = f_{DTS} / 8$, N = 8 1010: $f_{SAMPLING} = f_{DTS} / 16$, N = 5 1011: $f_{SAMPLING} = f_{DTS} / 16$, N = 6 1100: $f_{SAMPLING} = f_{DTS} / 16$, N = 8 1101: $f_{SAMPLING} = f_{DTS} / 32$, N = 5 1110: $f_{SAMPLING} = f_{DTS} / 32$, N = 6 1111: $f_{SAMPLING} = f_{DTS} / 32$, N = 8
3:2	IC3PSC	R/W	0x0	Input capture 3 prescaler: 00: no prescaler 01: capture is done once every 2 events 10: capture is done once every 4 events 11: capture is done once every 8 events The IC3PSC register is reset when CC3E = 0.
1:0	CC3S	R/W	0x0	Capture/compare channel 3 selection: 00: CC3 channel is configured as output. 01: CC3 channel is configured as input, IC3 is mapped on TI3. 10: CC3 channel is configured as input, IC3 is mapped on TI4. 11: CC3 channel is configured as input, IC3 is mapped on TRC. Note: CC3S bits are writable only when the channel is OFF (CC3E = 0).

17.6.9 Capture/Compare Enable Register (TIM_CCER)

Offset address: 0x20

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:14	RSV	-	-	Reserved
13	CC4P	R/W	0x0	Capture/compare 4 output polarity: **CC4 channel configured as output:** 0: OC4 is OC4REF. 1: OC4 is the inverse OC4REF. **CC4 channel configured as input:** 0: non-inverted mode: capture is done on the

Bit	Name	Attribute	Reset Value	Description
				rising edge of IC4. 1: inverted mode: capture is done on the falling edge of IC4.
12	CC4E	R/W	0x0	Capture/compare 4 output enable: **CC4 channel configured as output:** 0: OC4 not active 1: OC4 active **CC4 channel configured as input:** 0: capture disabled 1: capture enabled
11:10	RSV	-	-	Reserved
9	CC3P	R/W	0x0	Capture/compare 3 output polarity: **CC3 channel configured as output:** 0: OC3 is OC3REF. 1: OC3 is the inverse OC3REF. **CC3 channel configured as input:** 0: non-inverted mode: capture is done on the rising edge of IC3. 1: inverted mode: capture is done on the falling edge of IC3.
8	CC3E	R/W	0x0	Capture/compare 3 output enable: **CC3 channel configured as output:** 0: OC3 not active 1: OC3 active **CC3 channel configured as input:** 0: capture disabled 1: capture enabled
7:6	RSV	-	-	Reserved
5	CC2P	R/W	0x0	Capture/compare 2 output polarity: **CC2 channel configured as output:** 0: OC2 is OC2REF. 1: OC2 is the inverse OC2REF. **CC2 channel configured as input:** 0: non-inverted mode: capture is done on the rising edge of IC2. 1: inverted mode: capture is done on the falling edge of IC2.

Bit	Name	Attribute	Reset Value	Description
4	CC2E	R/W	0x0	Capture/compare 2 output enable: **CC2 channel configured as output:** 0: OC2 not active 1: OC2 active **CC2 channel configured as input:** 0: capture disabled 1: capture enabled
3:2	RSV	-	-	Reserved
1	CC1P	R/W	0x0	Capture/compare 1 output polarity: **CC1 channel configured as output:** 0: OC1 is OC1REF. 1: OC1 is the inverse OC1REF. **CC1 channel configured as input:** 0: non-inverted mode: capture is done on the rising edge of IC1. 1: inverted mode: capture is done on the falling edge of IC1.
0	CC1E	R/W	0x0	Capture/compare 1 output enable: **CC1 channel configured as output:** 0: OC1 not active 1: OC1 active **CC1 channel configured as input:** 0: capture disabled 1: capture enabled

17.6.10 Counter Register (TIM_CNT)

Offset address: 0x24

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:16	RSV	-	-	Reserved
15:0	CNT	R/W	0x0	Counter value

17.6.11 Prescaler Register (TIM_PSC)

Offset address: 0x28

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:16	RSV	-	-	Reserved
15:0	PSC	R/W	0x0	Counter clock (CK_CNT) prescaler value: $f_{CK\_CNT} = f_{CK\_PSC} / (PSC[15:0] + 1)$ This is a preload register whose content are transferred into the shadow register at each update event.

17.6.12 Auto-reload Register (TIM_ARR)

Offset address: 0x2C

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:16	RSV	-	-	Reserved
15:0	ARR	R/W	0x0	Auto-reload value at counter overflow: This is a preload register whose content are transferred into the shadow register at each update event.

17.6.13 Capture/Compare Register 1 (TIM_CCR1)

Offset address: 0x34

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:16	RSV	-	-	Reserved
15:0	CCR1	R/W	0x0	Capture/compare channel 1 register: **If channel CC1 is configured as output:** This is a preload register containing the value to be compared to the counter and signaled on OC1 output. **If channel CC1 is configured as input:** CCR1 is the counter value transferred by the

Bit	Name	Attribute	Reset Value	Description
				last input capture event, at this point the CCR1 register is read-only.

17.6.14 Capture/Compare Register 2 (TIM_CCR2)

Offset address: 0x38

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:16	RSV	-	-	Reserved
15:0	CCR2	R/W	0x0	Capture/compare channel 2 register: **If channel CC2 is configured as output:** This is a preload register containing the value to be compared to the counter and signaled on OC2 output. **If channel CC2 is configured as input:** CCR2 is the counter value transferred by the last input capture event, at this point the CCR2 register is read-only.

17.6.15 Capture/Compare Register 3 (TIM_CCR3)

Offset address: 0x3C

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:16	RSV	-	-	Reserved
15:0	CCR3	R/W	0x0	Capture/compare channel 3 register **If channel CC3 is configured as output:** This is a preload register containing the value to be compared to the counter and signaled on OC3 output. **If channel CC3 is configured as input:** CCR3 is the counter value transferred by the last input capture event, at this point the CCR3 register is read-only.

17.6.16 Capture/Compare Register 4 (TIM_CCR4)

Offset address: 0x40

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:16	RSV	-	-	Reserved
15:0	CCR4	R/W	0x0	Capture/compare channel 4 register **If channel CC4 is configured as output:** This is a preload register containing the value to be compared to the counter and signaled on OC4 output. **If channel CC4 is configured as input:** CCR4 is the counter value transferred by the last input capture event, at this point the CCR4 register is read-only.

17.6.17 DMA Control Register (TIM_DCR)

Offset address: 0x48

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:13	RSV	-	-	Reserved
12:8	DBL	R/W	0x0	DMA burst length: A read or write access to the TIM_DMAR register will trigger DMA transfer with a burst length of 1−18. 00000: 1 transfers 00001: 2 transfers 00010: 3 transfers …… 10001: 18 transfers Others: invalid value, write prohibited
7:5	RSV	-	-	Reserved
4:0	DBA	R/W	0x0	DMA base address, defined as the offset address directed to the register: 00000: TIM_CR1

Bit	Name	Attribute	Reset Value	Description
				00001: TIM_CR2 00010: TIM_SMCR …… Note: When DBA + DBL exceeds the TIM register address range, the actual burst transfer stops automatically when it reaches the highest TIM register address, i.e., the burst length is shortened.

17.6.18 DMA Access Register (TIM_DMAR)

Offset address: 0x4C

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:0	DMAR	R/W	0x0	DMA register for burst access: When using DMA burst transfer, set the DMA channel peripheral address to TIM_DMAR. Accesses to this register will point to the register specified in TIM_DCR, and TIM will generate multiple DMA requests based on the DBL value.

17.7 Operation Procedure

17.7.1 Counting Mode

1. Enable the TIMx clock in RCM module.
2. Configure TIM_CR1[4] to set the counting direction.
3. Set TIM_CR1[7] to 1 to enable the auto-reload preload.
4. Configure TIM_PSC[15:0] to set the prescaler value.
5. Configure TIM_ARR[31:0] to set the auto-reload value.
6. Set TIM_CR1[2] to 1 so that an update interrupt or DMA request will be generated only at counter overflow or underflow.

7. Set TIM_CR1[1] to 0 to enable the update event.
8. Set TIM_EGR[0] to 1 so that when the software sets TIM_EGR[0], the counter will be reinitialized, the shadow register will be updated, and the prescaler counter will be cleared.
9. Set TIM_CR1[0] to 1 to enable the counter.
10. Set TIM_DIER[0] to 1 to enable the update event interrupt.

17.7.2 PWM Mode

1. Configure the GPIO alternate function to enable the TIMx clock in RCM module.
2. Configure TIM_CR1[4] to set the counting direction.
3. Set TIM_CR1[7] to 1 to enable the auto-reload preload.
4. Configure TIM_PSC[15:0] to set the prescaler value.
5. Configure TIM_ARR[31:0] to set the auto-reload value.
6. According to the output channel, set TIM_CCMRx[1:0/9:8] to 0 to configure channel x as output.
7. Configure TIM_CCMRx[6:4/14:12] to set PWM mode 1/2.
8. Configure TIM_CCER[1/5/9/13] to set the output polarity.
9. Set TIM_CCER[0/4/8/12] to 1 to enable channel x output.
10. Set TIM_CR1[2] to 1 so that an update interrupt or DMA request will be generated only at counter overflow or underflow.
11. Set TIM_CR1[1] to 0 to enable the update event.
12. Set TIM_EGR[0] to 1 so that when the software sets TIM_EGR[0], the counter will be reinitialized, the shadow register will be updated, and the prescaler counter will be cleared.
13. Set TIM_CR1[0] to 1 to enable the counter.
14. Set TIM_DIER[0] to 1 to enable the update event interrupt.

15. Configure TIM_CCRx[31:0] to set the compare value of channel x.

17.7.3 Input Capture Mode

1. Configure the GPIO alternate function to enable the TIMx clock in RCM module.
2. Configure TIM_CR1[4] to set the counting direction.
3. Set TIM_CR1[7] to 1 to enable the auto-reload preload.
4. Configure TIM_PSC[15:0] to set the prescaler value.
5. Configure TIM_ARR[31:0] to set the auto-reload value.
6. Configure TIM_CCMRx[1:0/9:8] to set channel CCx as input, and perform mapping as required.
7. Configure TIM_CCER[1/5/9/13] to set the capture polarity.
8. Configure TIM_CCMRx[7:4/15:12] to set the sampling frequency and filter length, generally set to 0.
9. Configure TIM_CCMRx[3:2/11:10] to set the prescaler value of input capture.
10. Set TIM_CCER[0/4/8/12] to 1 to enable the capture function.
11. Set TIM_EGR[0] to 1 so that when the software sets TIM_EGR[0], the counter will be reinitialized, the shadow register will be updated, and the prescaler counter will be cleared.
12. Set TIM_CR1[0] to 1 to enable the counter.
13. Set TIM_DIER[1/2/3/4] to 1 to enable the capture interrupt of channel x.

17.7.4 DMA Mode

In input capture mode, the channel capture value of TIMx is transferred to SRAM via DMA:

1. In input capture mode, before setting TIM_EGR[0] bit by software and enabling the counter, add the following configurations:
2. Configure TIM_DCR[12:8] to set the DMA burst length.

3. Configure TIM_DCR[4:0] to set the DMA base address. Generally, the base address here selects the capture/compare register corresponding to the capture channel.
4. Set TIM_DIER[9/10/11/12] to 1 to enable the CCx DMA request.
5. Set TIM_CR2[3] to 0 to generate CCxDMA request at CCx event.
6. For details on DMA controller configuration, please refer to chapter "11 Direct Memory Access Controller (DMA)".
7. After initiating DMA transfer, when a capture event occurs on the channel, DMA will transfer the value stored in base address to SRAM.

In output compare mode, the value in SRAM is transferred via DMA to the compare register of TIMx:

1. In PWM mode, before setting TIM_EGR[0] bit by software and enabling the counter, add the following configurations:
2. Configure TIM_DCR[12:8] to set the DMA burst length.
3. Configure TIM_DCR[4:0] to set the DMA base address. Generally, the base address here selects the capture/compare register corresponding to the compare channel.
4. Set TIM_DIER[9/10/11/12] to 1 to enable the CCx DMA request.
5. Set TIM_CR2[3] to 0 to generate CCxDMA request at CCx event.
6. For details on DMA controller configuration, please refer to chapter "11 Direct Memory Access Controller (DMA)".
7. After the DMA transfer is started, when the counter value matches the compare value, DMA will transfer the value in SRAM to the base address.

18 General-purpose Timers (TIM14~TIM16)

18.1 Overview

The general-purpose timers consist of a 16-bit auto-reload counter driven by a programmable prescaler. It may be used for a variety of purposes, including input capture, output compare and PWM.

18.2 Main Features

- 16-bit up, down, up/down auto-reload counter
- 16-bit programmable prescaler allowing real-time adjustment of the counter clock division
- 1 independent channel for input capture, output compare and PWM generation
- Interrupt or DMA event can be generated in the following cases:
  - Counter overflow/underflow, counter initialization (triggered by software or hardware)
  - Input capture
  - Output compare

18.3 System Block Diagram

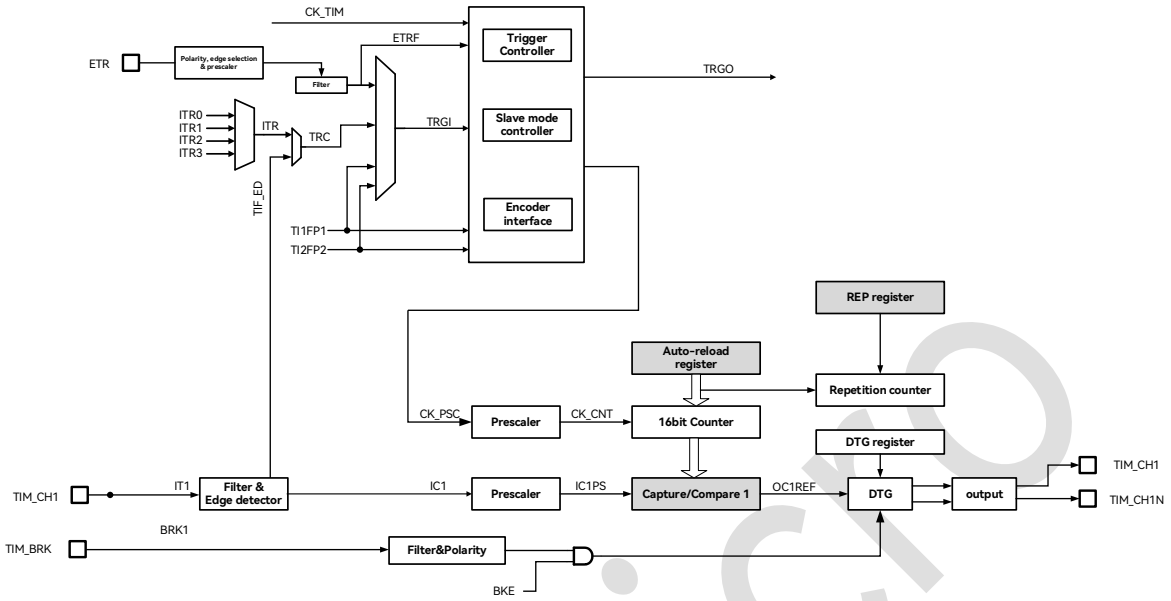


Figure 18-1: System Block Diagram of TIM14~TIM16

18.3.1 Encoder Interface Mode

1. Configure the GPIO alternate function to enable the TIMx clock in RCM module.
2. Configure TIM_CR[4] to set the counting direction.
3. Set TIM_CR1[7] to 1 to enable the auto-reload preload.
4. Configure TIM_PSC[15:0] to set the prescaler value.
5. Configure TIM_ARR[31:0] to set the auto-reload value.
6. Set TIM_CCMR1[1:0] to 1 to configure channel CC1 as input with IC1 mapped on TI1.
7. Set TIM_CCMR1[9:8] to 1 to configure channel CC2 as input with IC2 mapped on TI2.
8. Configure TIM_CCER[1] to set the capture polarity.
9. Configure TIM_CCER[5] to set the capture polarity.
10. Configure TIM_CCMR1[7:4] to set the sampling frequency and filter length, generally set to 0.
11. Configure TIM_CCMR1[15:12] to set the sampling frequency and filter length, generally set to 0.

12. Configure TIM_SMCR[2:0] to set encoder mode 1/2/3.
13. Set TIM_CCER[0] to 1 to enable the capture function of channel 1.
14. Set TIM_CCER[4] to 1 to enable the capture function of channel 2.
15. Set TIM_EGR[0] to 1 so that when the software sets TIM_EGR[0], the counter will be reinitialized, the shadow register will be updated, and the prescaler counter will be cleared.
16. Set TIM_CR1[0] to 1 to enable the counter.
17. Set TIM_DIER[1] to 1 to enable capture interrupt of channel 1.

18.4 Pin Description

Table 18-1: Pin Description on TIM14~TIM16

Function Pin	Alternate Function Pin	Direction	Functional Description
TIM14_BKIN	PA9, PB12, PC4	Input	Break input
TIM14_CH1	PA2, PB0, PB14, PD2	Input/output	Channel input capture / PWM output signal
TIM14_CH1N	PA1, PA3, PB13	Output	PWM output inversion
TIM15_BKIN	PB2, PB5, PC10	Input	Break input
TIM15_CH1	PA1, PA6, PB3, PB8, PC4	Input/output	Channel input capture / PWM output signal
TIM15_CH1N	PA15, PB6, PB15, PC5	Output	Reverse PWM output
TIM16_BKIN	PA10, PB4, PD2, PC10	Input	Break input
TIM16_CH1	PA7, PB9, PC11, PB6	Input/output	Channel input capture / PWM output signal
TIM16_CH1N	PB7, PC12	Output	Reverse PWM output

18.5 Functional Description

18.5.1 Time-base Unit

The main block of the time-base unit is a 16-bit counter with its related auto-reload register. The counter can count up, down, or both up and down. The counter clock can be divided by a 16-bit prescaler.

The counter, the auto-reload register and the prescaler register can be written or read by software, which is true even when the counter is running.

The time-base unit includes:

- Counter register (TIM_CNT)
- Prescaler register (TIM_PSC)
- Auto-reload register (TIM_ARR)

The auto-reload register is preloaded, which is controlled by the auto-reload preload enable (ARPE) bit in the register. When ARPE = 0, write to the ARR register, and the written data is directly transferred to the shadow register. When ARPE = 1, the data written to the ARR register is transferred to the shadow register when an update event (TIM_CNT overflow or underflow) occurs. The update event of ARR can also be actively triggered by software via register operation.

The counter TIM_CNT is clocked by the prescaler output TIM_PSC, which is enabled only when the counter enable bit (CEN) in the register is set. When CNT = ARR, this round of counting is over and the update event is sent.

TIM_PSC is a synchronous prescaler that can divide the counter clock frequency by any factor between 1 and 65536. The PSC register is also buffered, and overwriting PSC does not actually overwrite the shadow register unless a new update event occurs. Thus the PSC register can be

changed in real time on the fly, and the new prescaler ratio is taken into account at the next update event.

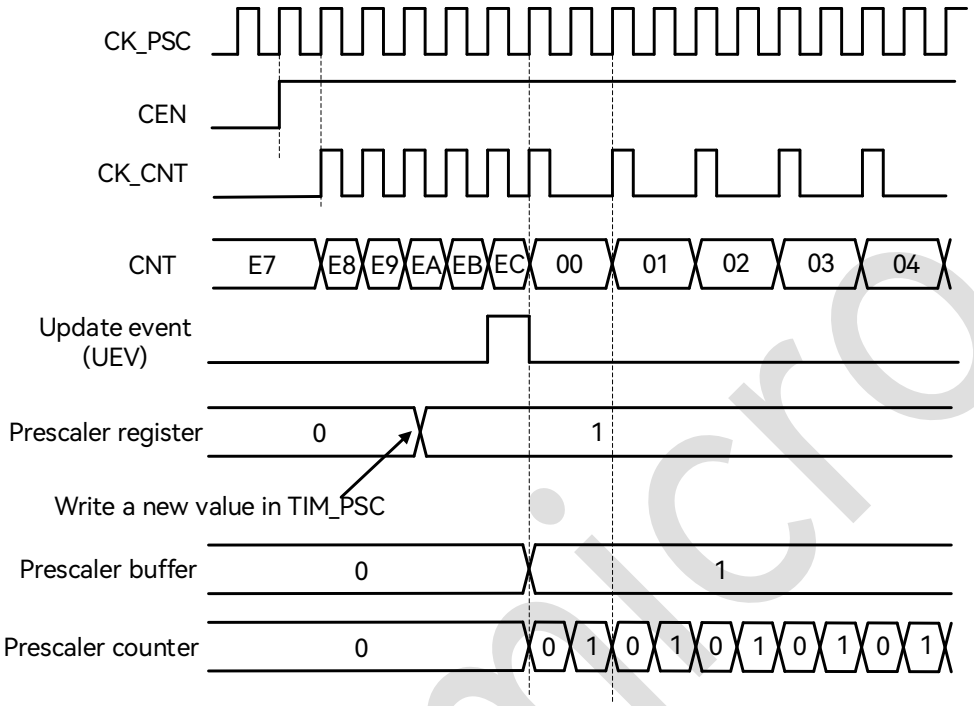


Figure 18-2: Counter Timing Diagram with Prescaler Division Changing from 1 to 2

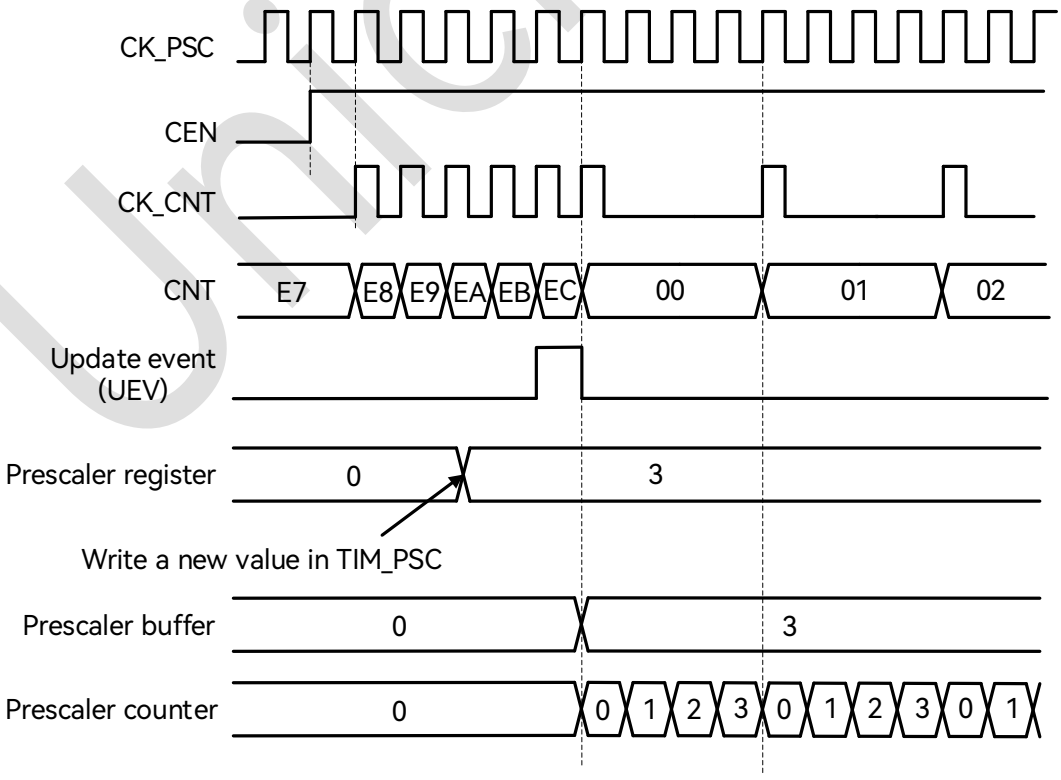


Figure 18-3: Counter Timing Diagram with Prescaler Division Changing from 1 to 4

18.5.2 Counter Operation Mode

The counter supports up-counting mode, down-counting mode and center-aligned mode.

18.5.2.1 Up-counting Mode

In up-counting mode, the counter counts from 0 to the auto-reload value, i.e. CNT = ARR, generating an overflow event, and then restarts counting from 0.

If the repetition counter is enabled, the counter repeats the above process a number of times (RCR + 1) as defined in RCR before generating an underflow event.

The software can directly trigger an update event by setting the UG bit in the register, at which time the CNT and the prescaler registers are automatically cleared. Whether setting the UG register triggers UIF (update interrupt flag) is determined by the setting of the URS register.

The update event can be disabled by setting the UDIS bit in the register to avoid updating the shadow register while writing new values in the preload registers.

When an update event occurs, the following registers are updated and the UIF bit is set:

- The auto-reload shadow register ARR is reloaded with the content of TIM_ARR register.
- The prescaler shadow register PSC is reloaded with the content of TIM_PSC register.

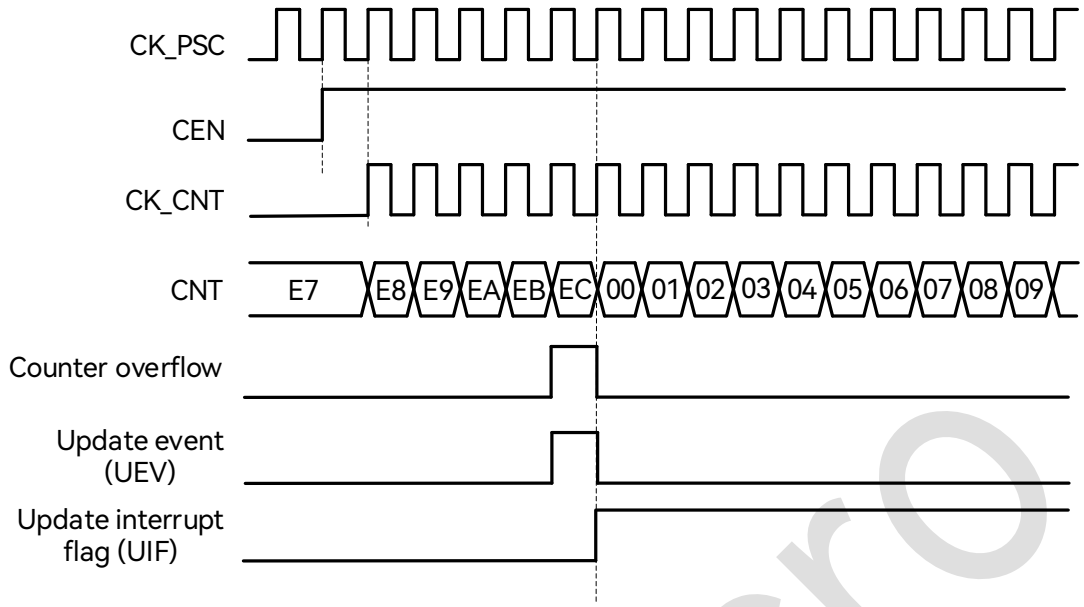


Figure 18-4: Up-counting Waveform Diagram, Internal Clock not Divided

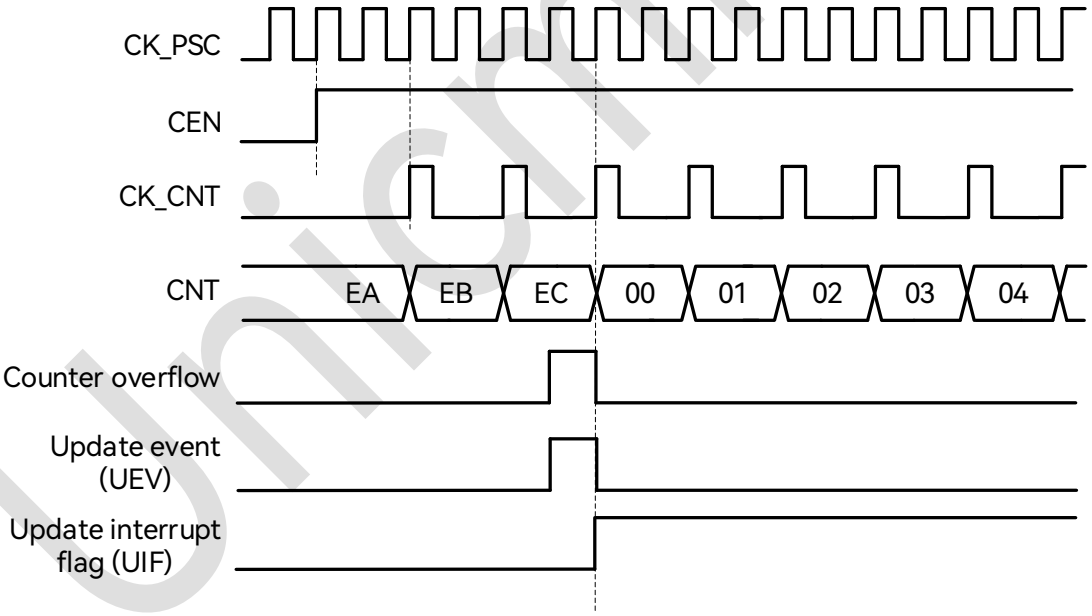


Figure 18-5: Up-counting Waveform Diagram, Internal Clock Divided by 2

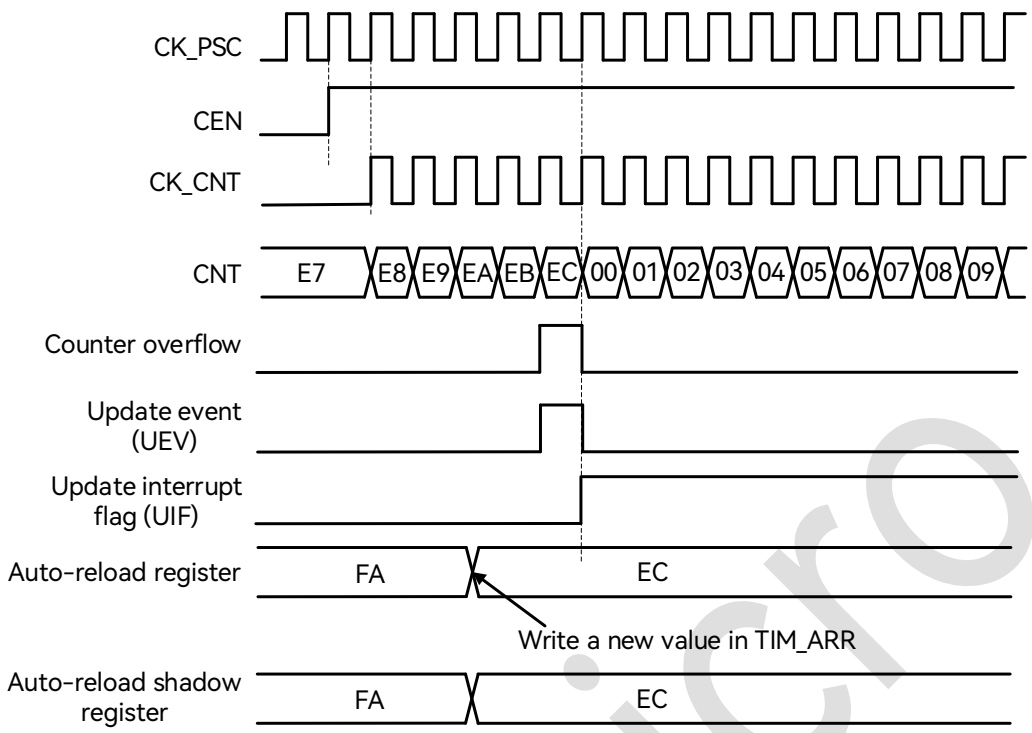


Figure 18-6: Counter Timing Diagram, Update Event when ARPE = 0 (TIM_ARR not Preloaded)

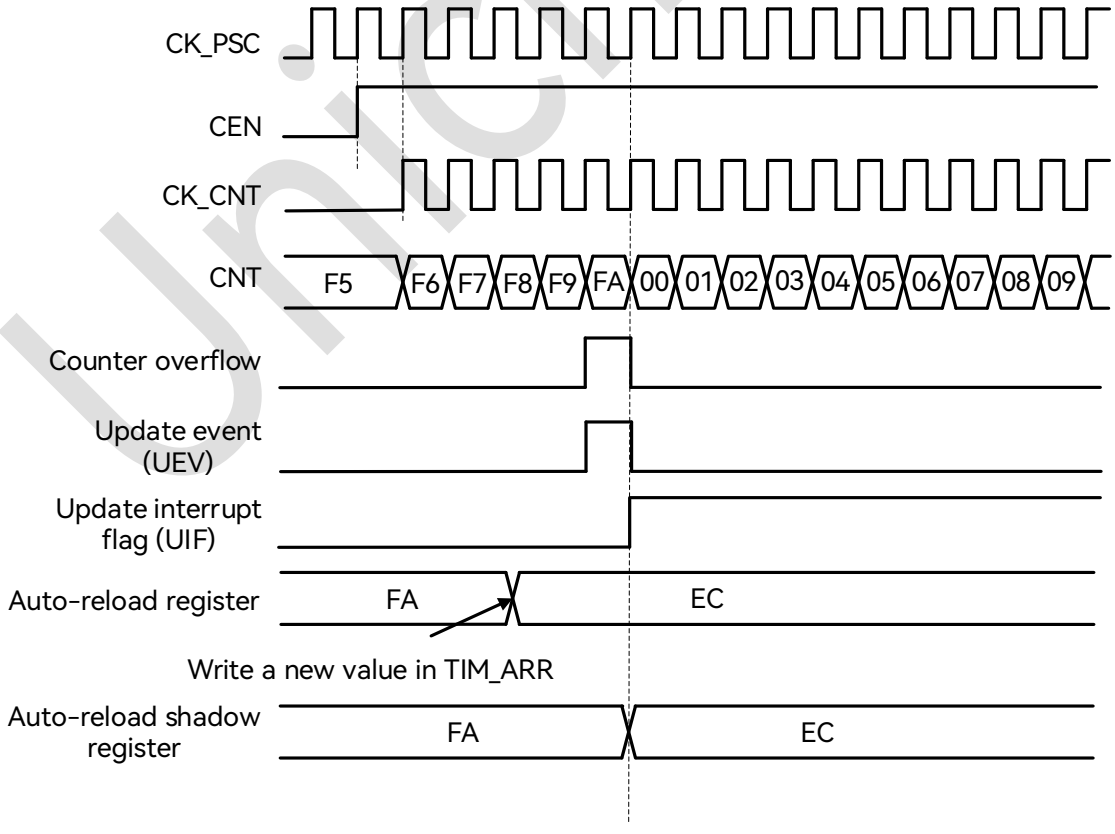


Figure 18-7: Counter Timing Diagram, Update Event when ARPE = 1 (TIM_ARR Preloaded)

18.5.2.2 Down-counting Mode

In down-counting mode, the counter counts from the auto-reload value down to 0, generating an underflow event, and then restarts counting from the auto-reload value.

If the repetition counter is enabled, the counter repeats the above process a number of times (RCR + 1) as defined in RCR before generating an underflow event.

The software can directly trigger an update event by setting the UG bit in the register, at which time the CNT and the prescaler registers are automatically cleared. Whether setting the UG register triggers UIF (update interrupt flag) is determined by the setting of the URS register.

The update event can be disabled by setting the UDIS bit in the register to avoid updating the shadow register while writing new values in the preload registers.

When an update event occurs, the following registers are updated and the UIF bit is set:

- The auto-reload shadow register ARR is reloaded with the content of TIM_ARR register.
- The prescaler shadow register PSC is reloaded with the content of TIM_PSC register.

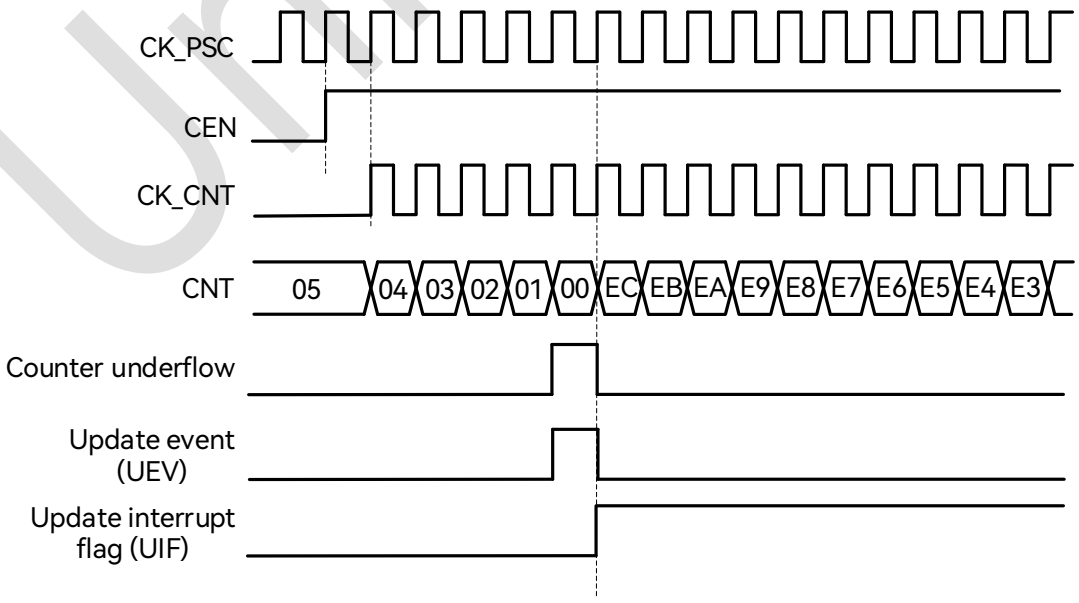


Figure 18-8: Down-counting Waveform Diagram, Internal Clock not Divided

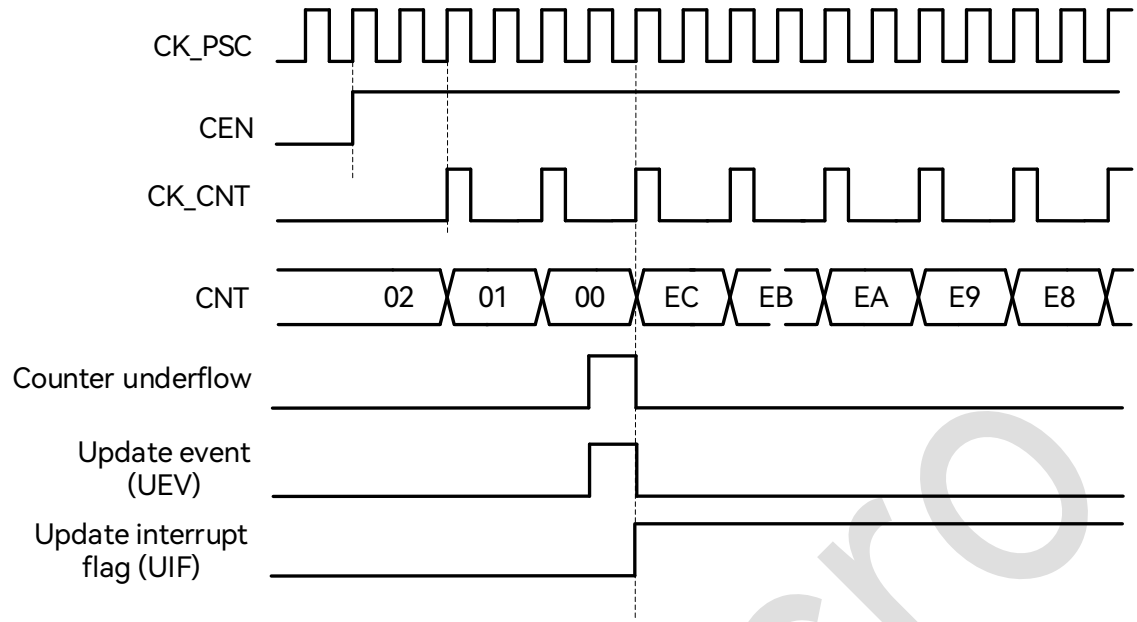


Figure 18-9: Down-counting Waveform Diagram, Internal Clock Divided by 2

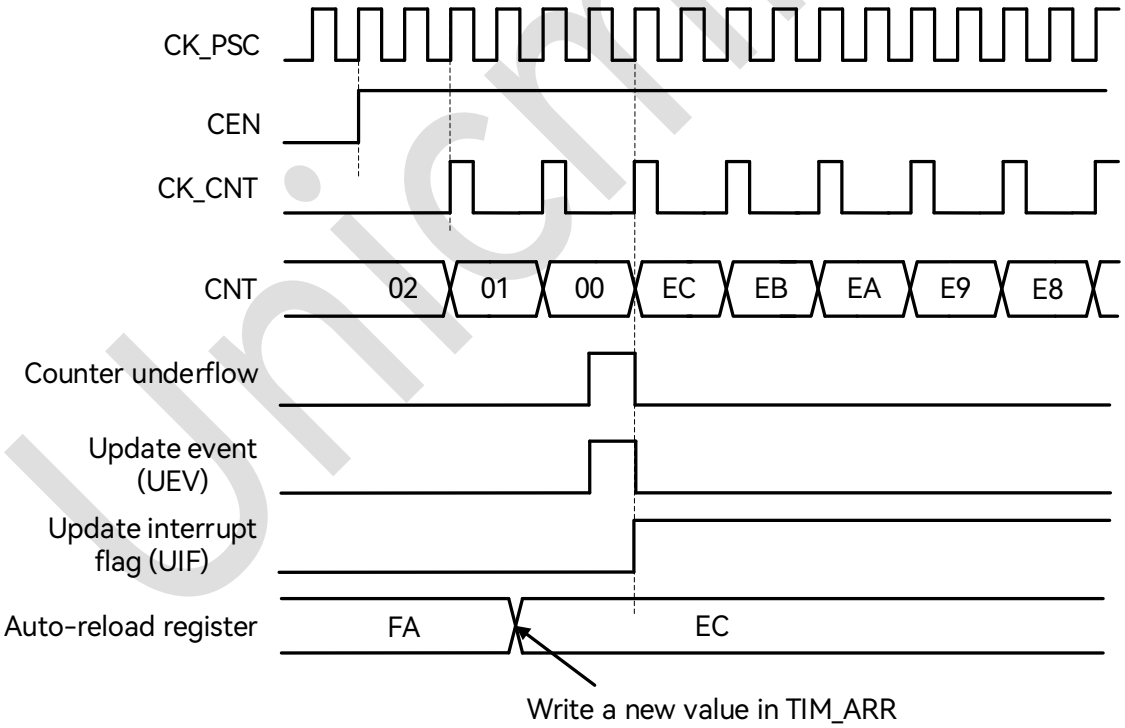


Figure 18-10: Down-counting Waveform Diagram, Internal Clock Divided by 2

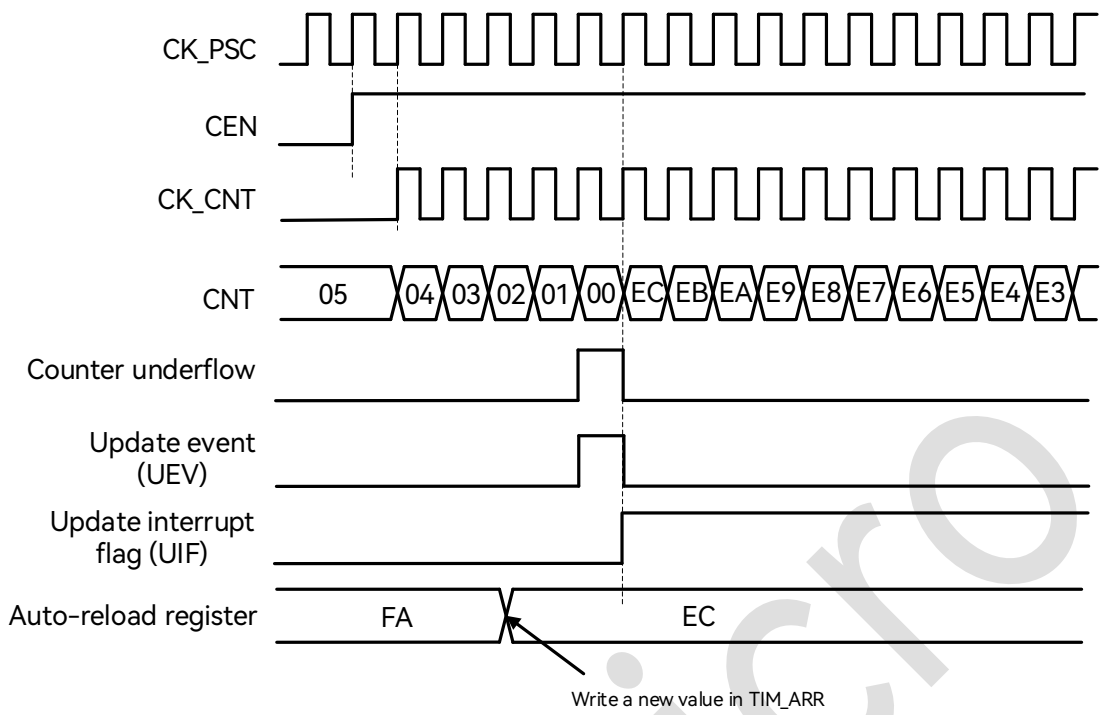


Figure 18-11: Down-counting Waveform Diagram, Update Event when Repetition Counter is not Used

18.5.2.3 Center-aligned Counting Mode

In center-aligned mode, the counter counts from 0 to the auto-reload value - 1 and generates a counter overflow event, then counts from the auto-reload value down to 1 and generates a counter underflow event, and then restarts counting from 0.

The CMS[1:0] bits in the register are used for enabling the center-aligned mode and selecting the output compare mode herein. The center-aligned mode is active when CMS! = 00. The output compare interrupt flag of channels configured in output is set when: the counter counts down (CMS = 01), the counter counts up (CMS = 10), the counter counts up and down (CMS = 11).

In this mode, the DIR direction bit in the register cannot be written by software. It is updated by hardware and gives the current direction of the counter.

The counter updates the shadow registers of ARR and PSC at each counter overflow and underflow.

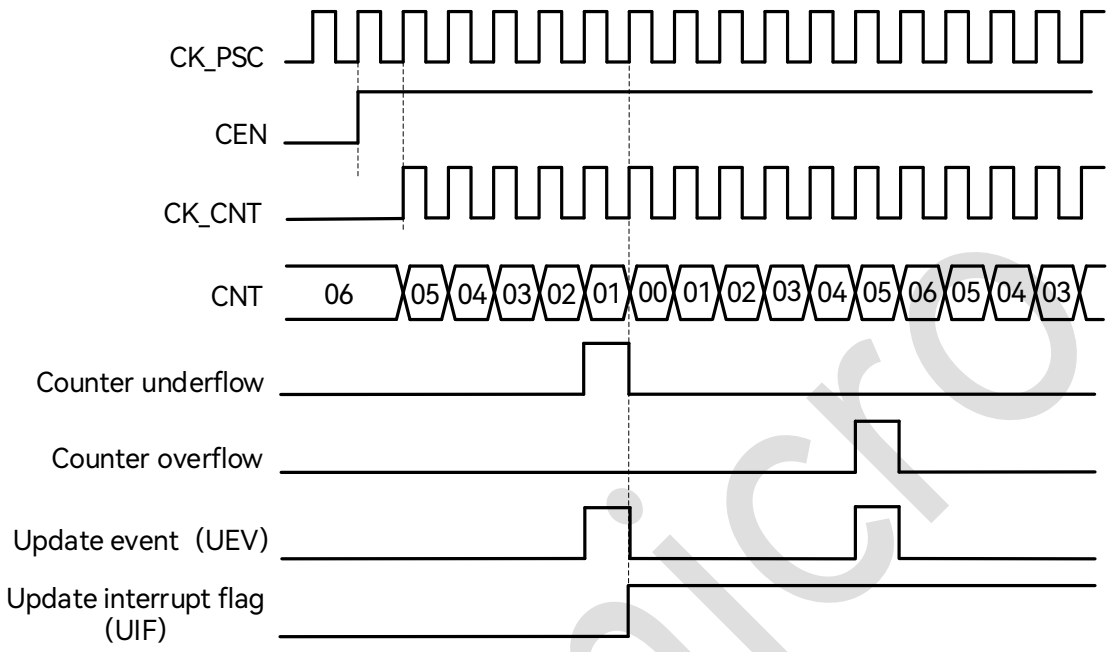


Figure 18-12: Center-aligned Counter Timing Diagram, TIM_PCS = 0, TIM_ARR = 0x6

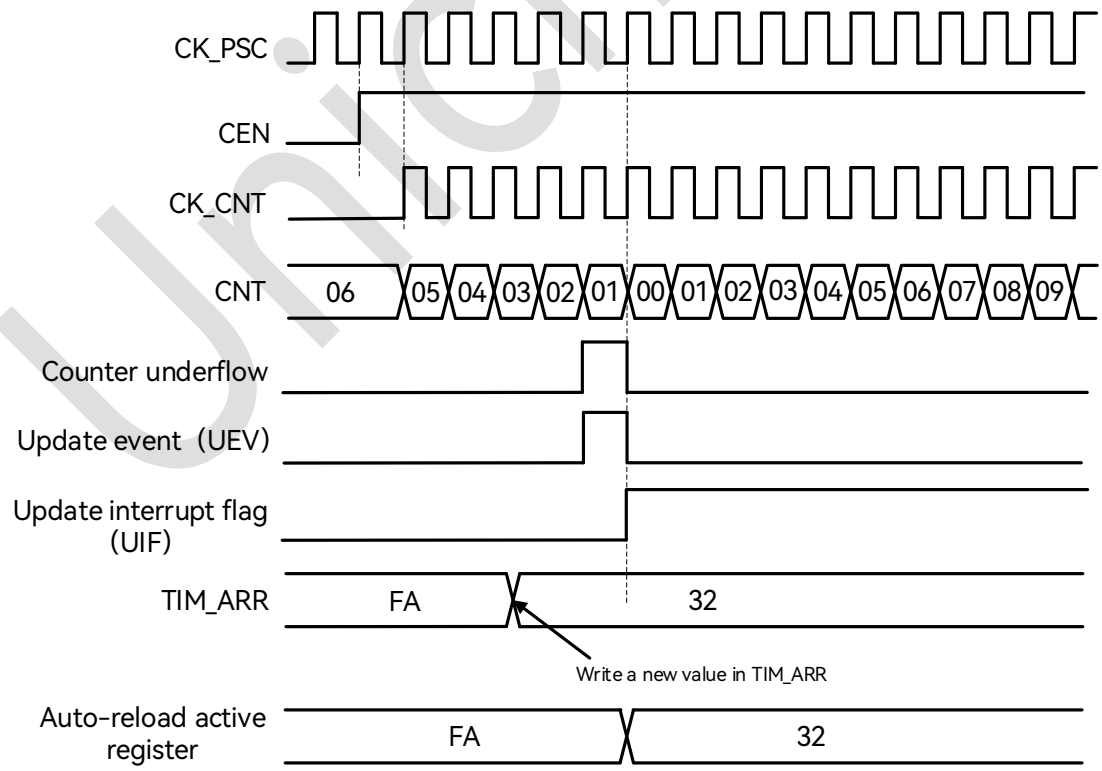


Figure 18-13: Counter Timing Diagram, Update Event with ARPE = 1 (Counter Underflow)

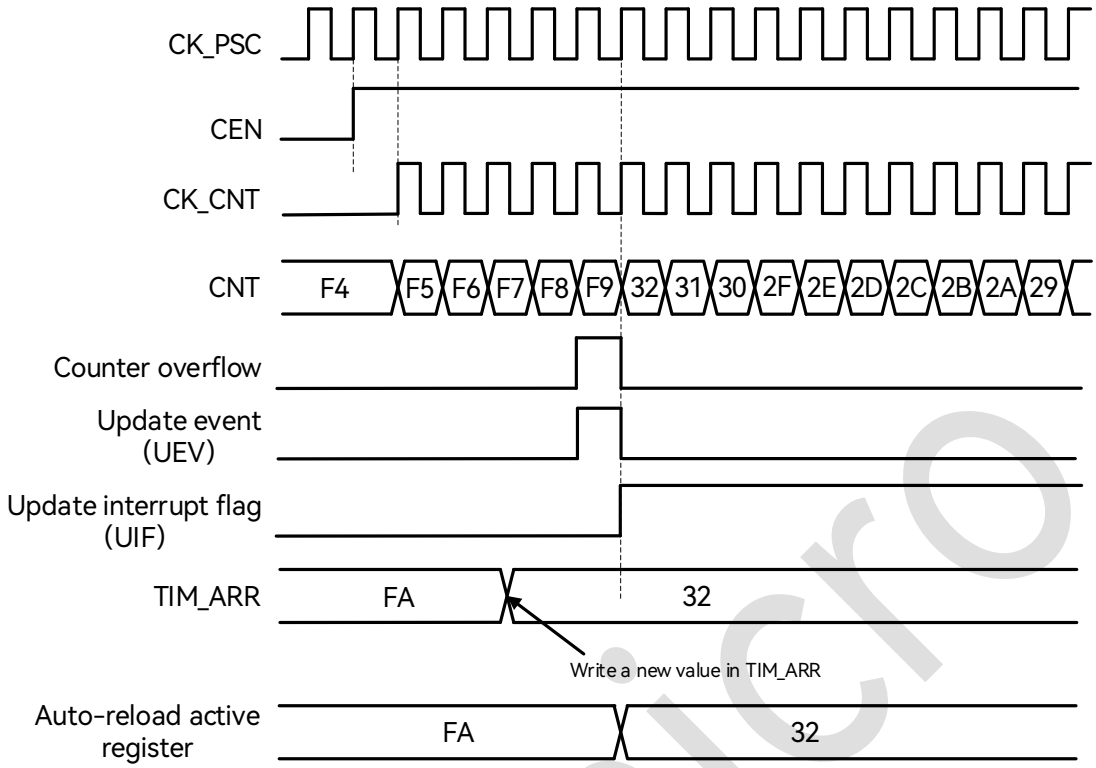


Figure 18-14: Counter Timing Diagram, Update Event with ARPE = 1 (Counter Overflow)

18.5.3 Preload Register

- The following functional registers support the preload function:
  - Auto-reload register TIM_ARR
  - Prescaler register TIM_PSC (preload function cannot be disabled)
  - Channel control register TIM_CCR
  - CCxE and CCxNE control register
  - OCxM control register

The preload function can be enabled or disabled by software for all of the above registers except TIM_PSC.

- Registers with preload function contain two sets of physical entities:
  - Shadow register: the register being used by the actual timer

- Preload register: the register accessible to software

- When the preload function is disabled, the register with preload function has the following characteristics:
  - The preload register can be accessed and overwritten by software in real time.
  - The shadow register is updated synchronously with the preload register.
- If the preload function is enabled, then:
  - All software operations access the preload register.
  - At the occurrence of update event, the content of all preload registers will be synchronously transferred to the corresponding shadow registers.

18.5.4 Counter Clock

The counter clock can be provided by the following clock sources:

- Internal clock: Timerx_clk
- External clock mode 1: external input pin TIx

18.5.4.1 Internal Clock Source

If the slave mode controller is disabled (SMS = 000), then the CEN, DIR and UG bits are controlled by software.

After the UG bit is set and the update signal is synchronized by CLK_PSC, the counter value is reinitialized.

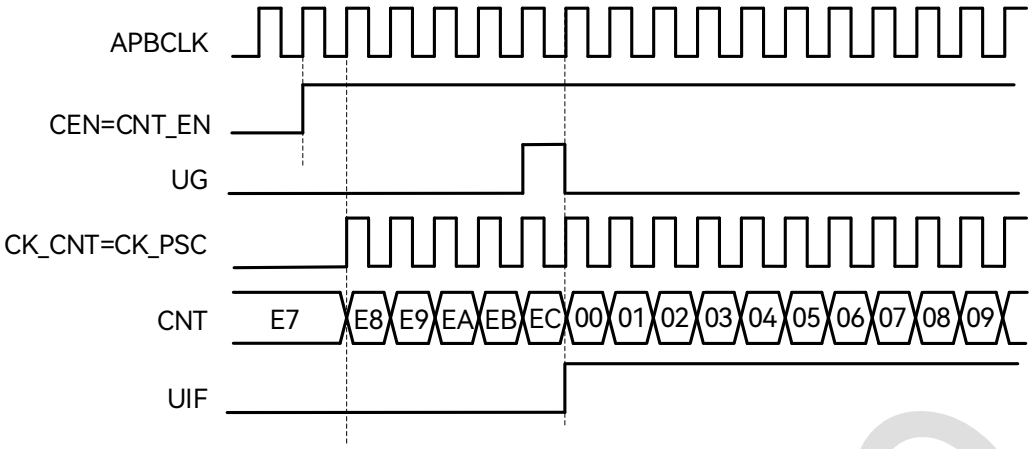


Figure 18-15: Timing Diagram in Internal Clock Source Mode, Clock Divided by 1

18.5.5 Capture / Compare Channels

TIM consists of one capture/compare channel, each of which is built around a capture/compare register CCR (including a shadow register), an input stage for capture and an output stage for compare.

The input stage samples the corresponding TIx input to generate a filtered signal TIxF. Then, an edge detection with polarity selection generates a signal (TIxFPx), which can be used as trigger input for counting or as the capture command and is prescaled before being captured.

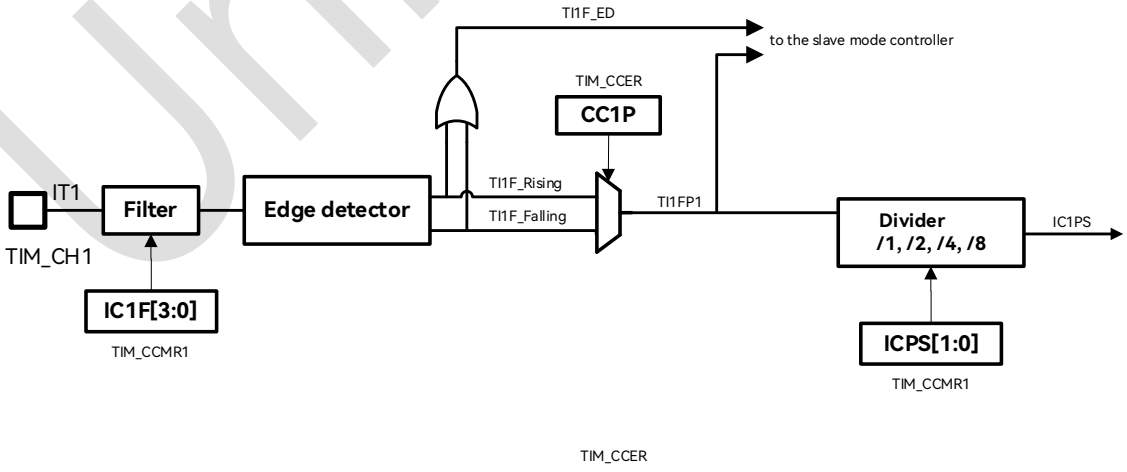


Figure 18-16: Capture/Compare Channel (Channel 1 Input Stage)

The output stage generates an output reference signal OCxREF, which is fixed to be active high and acts as the reference input to the final output circuit. Wherein, channels 1~3 support complementary output and dead-time insertion, while channel 4 is relatively simple and does not support complementary output.

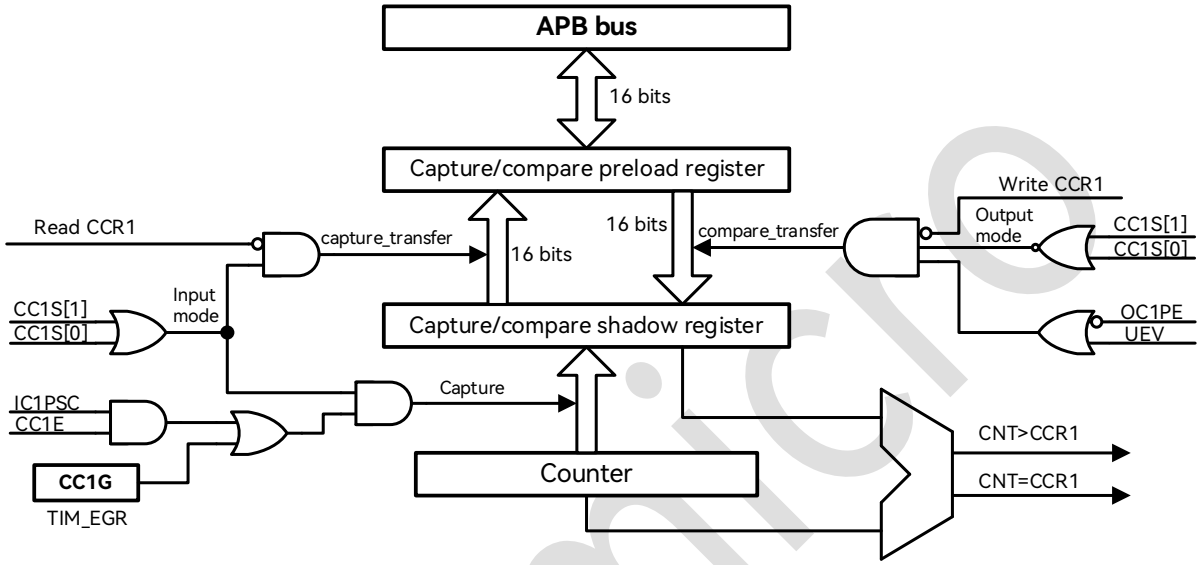


Figure 18-17: Capture/Compare Channel 1 Main Circuit

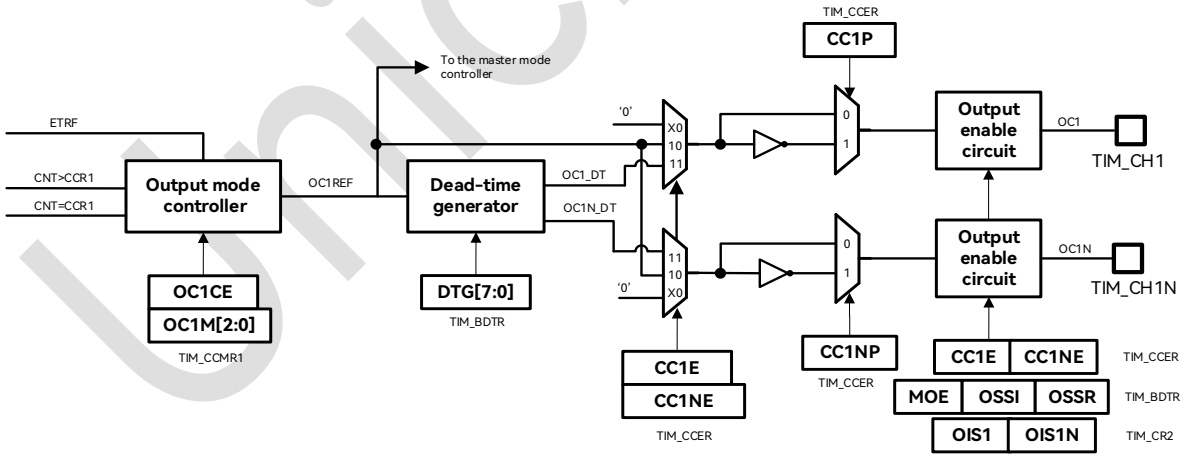


Figure 18-18: Output Stage of Capture/Compare Channel (Channel 1)

The capture/compare block is made of one preload register and one shadow register. Write and read always access the preload register. In capture mode, captures are actually done in

the shadow register, which is copied into the preload register. In compare mode, the content of the preload register is copied into the shadow register for comparison with the counter.

18.5.6 Input Capture Mode

When the expected level transition is detected by the ICx signal, a capture is triggered, and the current counter value is latched into CCR. At the same time, the CCxIF interrupt flag is set and a corresponding interrupt or a DMA request can be triggered. If a capture occurs while the CCxIF flag is already high, then the over-capture flag CCxOF is set (the last capture value in CCR is overwritten). CCxIF can be cleared by software or automatically cleared by reading the CCR register. CCxOF is cleared by software writing it to 1.

To capture the counter value to the TIM_CCR1 register on the rising edge of the TI1 input, the configuration steps are as follows:

1. In GPIO module, configure the corresponding pin as TIM_CH1.
2. Disable the channel by setting TIM_CCER[0] = 0 to ensure the success of subsequent channel configuration.
3. Select the input channel by setting TIM_CCMR1[1:0] = 01, with IC1 mapped on TI1.
4. Select the active counting edge to be rising edge or falling edge by setting TIM_CCER[1].
5. Configure the input filter duration by writing the IC1F[3:0] bits in the TIM_CCMR1 register.
6. Configure the input prescaler by writing the IC1PS[1:0] bits in the TIM_CCMR1 register.
7. Enable the channel by setting TIM_CCER[0] = 1.

18.5.7 Software Force Output

In compare output mode, the OCxREF signal can be forced to active or inactive level directly by software, independently of any comparison between the CCR and the counter.

The OCxREF signal can be forced to be active (OCxREF is always active high) by writing OCxM

= 101, and forced to be inactive (low level) by writing OCxM = 100. Anyway, the comparison between CCR and the counter is still performed.

18.5.8 Output Compare Mode

In output compare mode, when a match is found between the capture/compare register CCR and the counter, the OCxREF can be set to be active, inactive or to toggle on match. At the same time, the interrupt flag is also set and DMA requests can be sent (overwriting the configuration register).

The output compare can also be used to output a pulse signal of a specific width (in one-pulse mode).

Procedure:

1. Select the counter clock (internal, external, prescaler).
2. Write the desired data to the ARR and CCR registers.
3. Set the interrupt enable bit and DMA enable bit as required.
4. Select the output mode.
5. Enable the counter.

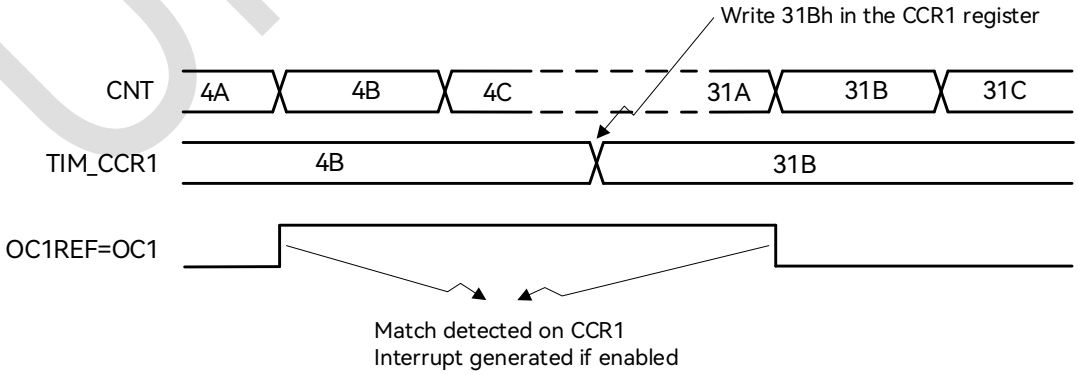


Figure 18-19: Output Compare Mode, Toggle on OC1

The CCR register can be updated at any time by software to control the output waveform,

provided that the preload register is not enabled. Otherwise, the CCR shadow register is only updated with the content of the preload register at the next update event.

18.5.9 PWM Output

PWM mode allows you to generate a pulse width modulation signal with a frequency determined by the value of the ARR register and a duty cycle determined by the value of the CCR register.

The polarity of the output signal is software programmable using the CCxP bit in the register. In PWM mode, CNT and CCR registers are always compared. The timer is able to generate PWM in edge-aligned mode or center-aligned mode.

18.5.9.1 PWM Edge-aligned Mode

In up-counting mode, when it is configured in PWM mode 1, the OCxREF signal is high as long as CNT < CCR, otherwise it is low. And OCxREF will be held at 1 if CCR > ARR while held at 0 if CCR is 0.

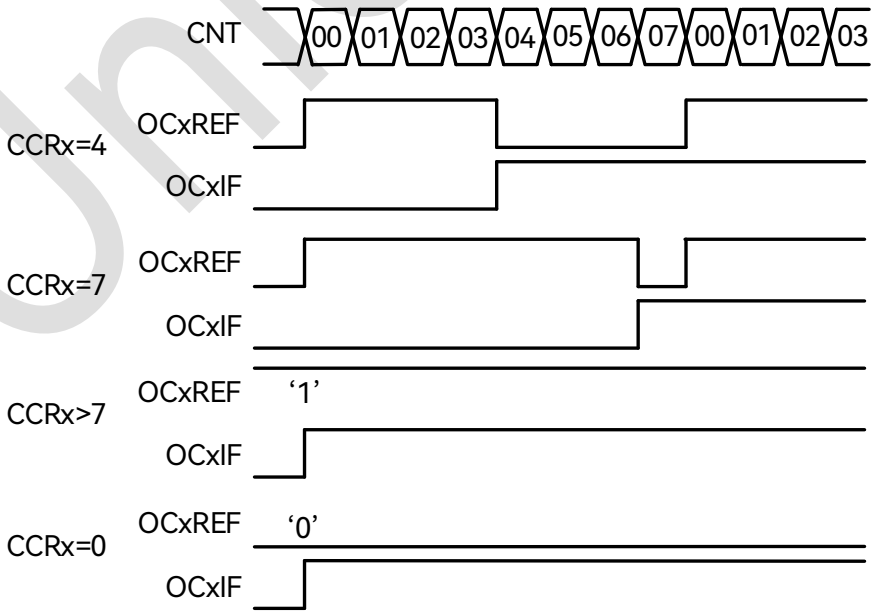


Figure 18-20: Edge-aligned PWM Waveform (ARR = 7)

In down-counting mode, the definition of OCxREF level is the same as that in up-counting mode.

18.5.9.2 PWM Center-aligned Mode

The definition of OCxREF level is the same as that in edge-aligned mode. The figure below is an example.

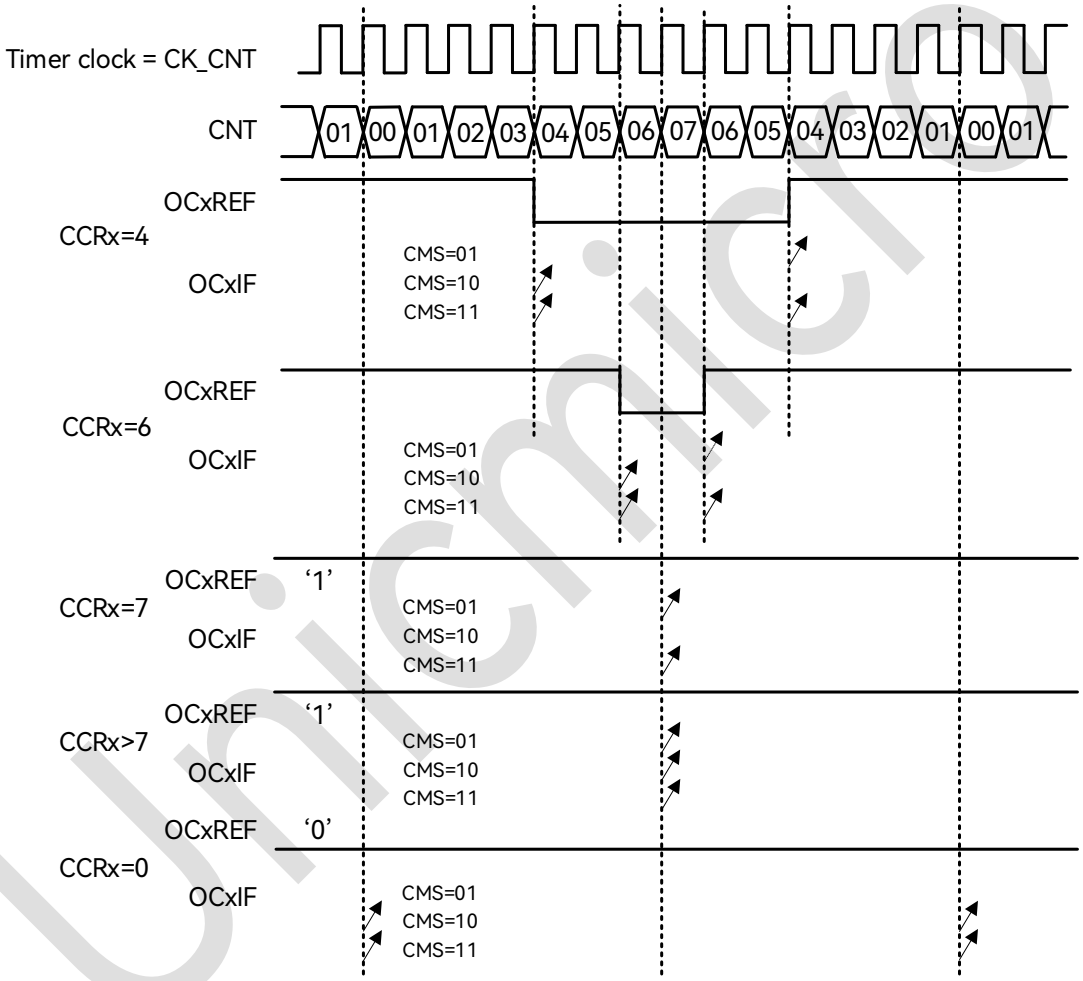


Figure 18-21: Center-aligned PWM Waveform (ARR = 7)

When start counting in center-aligned mode, the initial counting direction is determined by the DIR bit in the register, and in the subsequent process, the DIR bit is directly controlled by hardware. The safest way to use center-aligned mode is to generate an update by setting the UG bit in the register just before starting the counter and not to overwrite the counter while it is running.

18.5.10 DMA Access

TIM supports 4 types of DMA requests, namely CC channel request, external trigger request, user software trigger request and COM trigger request.

The DMA request generated by CC channel is used to transfer the content of CCRx to RAM in capture mode, and to write the data in RAM to CCRx in compare mode.

In addition, DMA requests can also be generated from external trigger event, software trigger event and COM trigger event, and at the occurrence of these requests, DMA burst transfer will be started to write data to one or more registers within TIM or to read one or more register values from TIM.

Table 18-2: Four DMA Requests Supported by TIM

DMA Request	DMA Access Object	Single-transfer Length
TIM_CH1	Read DMAR	DBL
	Write DMAR	
TIM_TRIG	Read DMAR	DBL
	Write DMAR	
TIM_UEV	Read DMAR	DBL
	Write DMAR	
TIM_COM	Read DMAR	DBL
	Write DMAR	

18.5.11 DMA Burst

TIM supports DMA and DMA-burst access. A DMA request can be generated at a specific event, so as to write the capture result in CCR to RAM or write the content of one or more registers in RAM to the preload register in TIM.

DMA-burst allows to generate multiple successive DMA requests upon a single event. The main purpose is to update the content of multiple registers in a row each time a given timer event is triggered, thus making it possible to dynamically modify the output waveform in real time.

The DMA controller destination is unique and must be directed to the virtual register TIM_DMAR. On a given timer event, the timer launches a sequence of DMA requests (burst). Each DMA write access to the TIM_DMAR register will be redirected to the actual function register by TIM.

The DBL bits in the register set the DMA burst length, and the DBA bits define the base address for DMA access to TIM (an offset starting from the address of the TIM_CR register).

In DMA-burst mode, all DMA access shall be directed to the DMAR virtual register, and TIM automatically accumulates the internal offset address according to the access. The DBA register is used to specify the destination address of the first DMA transfer within TIM, while the DBL register is used to specify the burst length.

18.5.12 Debug Mode

When the CPU enters debug mode, the timer can either stop or continue working, and its behavior is defined by registers in the chip system.

When the timer is stopped during debugging, its output will be disabled (MOE is cleared). Depending on the register configuration, the output signal can be forced to be inactive or controlled by the GPIO module.

18.6 Register Description

TIM14 register base address: 0x4700_7000

TIM15 register base address: 0x4700_7400

TIM16 register base address: 0x4700_7800

The registers are listed below:

Table 18-3: List of General-purpose Timers (TIM14~TIM16)

Offset Address	Name	Description
0x00	TIM_CR1	Control register 1
0x04	TIM_CR2	Control register 2
0x0C	TIM_DIER	DMA and interrupt enable register
0x10	TIM_SR	Status register
0x14	TIM_EGR	Event generation register
0x18	TIM_CCMR1	Capture/compare mode register 1
0x20	TIM_CCER	Capture/compare enable register
0x24	TIM_CNT	Counter register
0x28	TIM_PSC	Prescaler register
0x2C	TIM_ARR	Auto-reload register
0x34	TIM_CCR1	Capture/compare register 1
0x44	TIM_BDTR	Break and dead-time control register
0x48	TIM_DCR	DMA control register
0x4C	TIM_DMAR	DMA access register

Registers are detailed in the following sections.

18.6.1 Control Register 1 (TIM_CR1)

Offset address: 0x00

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:10	RSV	-	-	Reserved
9:8	CKD	R/W	0x0	Dead time and digital filter clock frequency division register (division ratio relative to CK_INT): 00: $t_{DTS} = t_{CK\_INT}$ 01: $t_{DTS} = 2 * t_{CK\_INT}$ 10: $t_{DTS} = 4 * t_{CK\_INT}$ 11: reserved, prohibited
7	ARPE	R/W	0x0	Auto-reload preload enable: 0: ARR not preloaded 1: ARR preloaded
6:5	CMS	R/W	0x0	Counter alignment mode selection:

Bit	Name	Attribute	Reset Value	Description
				00: edge-aligned mode 01: center-aligned mode 1; output compare interrupt flags are set only when the counter is counting down. 10: center-aligned mode 2; output compare interrupt flags are set only when the counter is counting up. 11: center-aligned mode 3; output compare interrupt flags are set both when the counter is counting up or down.
4	DIR	R/W	0x0	Counting direction register: 0: count up 1: count down Note: This register is read-only when the timer is configured in center-aligned mode or encoder mode.
3	OPM	R/W	0x0	One-pulse mode output: 0: the counter does not stop at the occurrence of update event. 1: the counter stops at the occurrence of update event (CEN cleared automatically).
2	URS	R/W	0x0	Update request source: 0: an update interrupt or DMA request will be generated by any of the following events: ● Counter overflow/underflow ● Software setting the UG bit ● Update generated from the slave mode controller 1: an update interrupt or DMA request will be generated only at counter overflow or underflow.
1	UDIS	R/W	0x0	Update disable: 0: update event enabled; the update event can be generated by any of the following events: ● Counter overflow/underflow ● Software setting the UG bit ● Update generated from the slave mode controller 1: update event disabled, shadow register not updated The counter and the prescaler will be reinitialized if the UG bit is set or if the slave mode controller receives a hardware reset.

Bit	Name	Attribute	Reset Value	Description
0	CEN	R/W	0x0	Counter enable: 0: disabled 1: enabled Note: The external trigger mode can automatically set the CEN bit.

18.6.2 Control Register 2 (TIM_CR2)

Offset address: 0x04

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:8	RSV	-	-	Reserved
9	OIS1N	R/W	0x0	Output idle state (OC1N output): 0: OC1N = 0 after a dead-time when MOE = 0 1: OC1N = 1 after a dead-time when MOE = 0; for OC1N, (CC1P‖CC1NP) = 1 is required to make OC1N = 1.
8	OIS1	R/W	0x0	Output idle state (OC1 output): 0: OC1 = 0 (after a dead-time if complementary output is enabled) when MOE = 0 1: OC1 = 1 (after a dead-time if complementary output is enabled) when MOE = 0; for OC1, (CC1P‖CC1NP) is required to make OC1 = 1.
7:4	RSV	-	-	Reserved
3	CCDS	R/W	0x0	Capture/compare DMA selection: 0: CCx DMA request sent when CCx event occurs 1: CCx DMA request sent when update event occurs
2:1	RSV	-	-	Reserved
0	CCPC	R/W	0x0	Capture/compare preload control: 0: CCxE, CCxNE and OCxM bits are not preloaded. 1: CCxE, CCxNE and OCxM bits are preloaded; after having been written, they are updated only when a commutation event (COM) occurs. Note: This bit acts only on channels that have a complementary output.

18.6.3 DMA / Interrupt Enable Register (TIM_DIER)

Offset address: 0x0C

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:17	RSV	-	-	Reserved
16	CC1OF_DISABLE	R/W	0x0	CC1OF interrupt enable: 0: enabled 1: disabled
15	RSV	-	-	Reserved
14	TDE	R/W	0x0	External trigger DMA request enable: 0: in slave mode, external trigger DMA request disabled 1: in slave mode, external trigger DMA request enabled (can be used to automatically update the preload register)
13:10	RSV	-	-	Reserved
9	CC1DE	R/W	0x0	Capture/compare channel 1 DMA request enable: 0: CC1 DMA request disabled 1: CC1 DMA request enabled
8	UDE	R/W	0x0	Update DMA request enable: 0: update DMA request disabled 1: update DMA request enabled
7	RSV	-	-	Reserved
6	TIE	R/W	0x0	Trigger interrupt enable: 0: trigger interrupt disabled 1: trigger interrupt enabled
5:2	RSV	-	-	Reserved
1	CC1IE	R/W	0x0	Capture/compare channel 1 interrupt enable: 0: CC1 interrupt disabled 1: CC1 interrupt enabled
0	UIE	R/W	0x0	Update interrupt enable: 0: update interrupt disabled 1: update interrupt enabled

18.6.4 Status Register (TIM_SR)

Offset address: 0x10

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:10	RSV	-	-	Reserved
9	CC1OF	R/W0C	0x0	Capture/compare channel 1 overcapture flag: This register is valid only when the corresponding channel is configured in input capture mode. This flag bit is set by hardware and cleared by software via writing it to 0. 0: no overcapture detected 1: a new capture occurs while CC1IF flag is 1.
8:7	RSV	-	-	Reserved
6	TIF	R/W0C	0x0	Trigger interrupt flag is set by hardware and cleared by software via writing it to 0.
5:2	RSV	-	-	Reserved
1	CC1IF	R/W0C	0x0	Capture/compare channel 1 interrupt flag: If channel CC1 is configured as output: the CC1IF flag is set when the counter matches the compare value, and cleared by software via writing it to 0. If channel CC1 is configured as input: this flag is set by hardware on a capture, and cleared by software via writing it to 0 or automatically cleared by software reading TIM_CCR1.
0	UIF	R/W0C	0x0	Update event interrupt flag is set by hardware and cleared by software via writing it to 0. UIF is set and the shadow register is updated at the following events: -Counter overflow occurs if repetition counter = 0 and UDIS = 0. -The counter is reinitialized by software setting the UG bit if URS = 0 and UDIS = 0. -The counter is reinitialized by a trigger event if URS = 0 and UDIS = 0.

18.6.5 Event Generation Register (TIM_EGR)

Offset address: 0x14

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:7	RSV	-	-	Reserved
6	TG	W	0x0	This bit can be set by software to generate a trigger event, and it is automatically cleared by hardware.
5:2	RSV	-	-	Reserved
1	CC1G	W	0x0	Capture/compare channel 1 generation: If channel CC1 is configured as output, CC1IF flag is set, and corresponding interrupt and DMA request will be sent if enabled. If channel CC1 is configured as input, the current counter value is captured in TIM_CCR1 register, the CC1IF flag is set, and corresponding interrupt and DMA request will be sent if enabled.
0	UG	W	0x0	This bit can be set by software to generate an update event, and is automatically cleared by hardware. When the software sets UG, the counter will be reinitialized, the shadow register will be updated, and the prescaler counter will be cleared.

18.6.6 Capture/Compare Mode Register 1 (TIM_CCMR1)

Offset address: 0x18

Reset value: 0x0000 0000

This register can be used for output compare mode or for input capture mode.

- Output compare mode

Bit	Name	Attribute	Reset Value	Description
31:8	RSV	-	-	Reserved
7	OC1CE	R/W	0x0	Output compare 1 clear enable: 0: OC1REF is not affected by ETRF input.

Bit	Name	Attribute	Reset Value	Description
				1: OC1REF is automatically cleared once a high level is detected on ETRF input.
6:4	OC1M	R/W	0x0	Output compare 1 mode: these bits define the behavior of the output reference signal OC1REF. 000: the comparison between the output compare register CCR1 and the counter CNT has no effect on the outputs. 001: set OC1REF high when CCR1 = CNT (falling edge) 010: set OC1REF low when CCR1 = CNT (falling edge) 011: toggle OC1REF when CCR1 = CNT (falling edge) 100: force OC1REF low (inactive) 101: force OC1REF high (active) 110: PWM mode 1—in up-counting, OC1REF is set high when CNT < CCR1, otherwise it is set low; in down-counting, OC1REF is set low when CNT ≥ CCR1, otherwise it is set high. 111: PWM mode 2—in up-counting, OC1REF is set low when CNT < CCR1, otherwise it is set high; in down-counting, OC1REF is set high when CNT ≥ CCR1, otherwise it is set low.
3	OC1PE	R/W	0x0	Output compare 1 preload enable: 0: preload register on CCR1 disabled; CCR1 can be written directly. 1: preload register on CCR1 enabled; read/write operations access the preload register; the preload value is shifted to the shadow register at each update event.
2	OC1FE	R/W	0x0	Output compare 1 fast enable: 0: fast disabled, the trigger input will not affect the comparison output. 1: fast enabled, the trigger input will immediately change OC1REF to the output when the comparison values match, regardless of the actual current comparison.

Bit	Name	Attribute	Reset Value	Description
				This function acts only if the channel is configured in PWM1 or PWM2 mode.
1	RSV	-	-	Reserved
0	CC1S	R/W	0x0	Capture/compare channel 1 selection: 0: CC1 channel is configured as output. 1: CC1 channel is configured as input, IC1 is mapped on TI1. Note: CC1S bits are writable only when the channel is OFF (CC1E = 0).

- Input capture mode

Bit	Name	Attribute	Reset Value	Description
31:8	RSV	-	-	Reserved
7:4	IC1F	R/W	0x0	Input capture 1 filter: this bit-field defines the frequency used to sample TI1 input and the length of the digital filter applied to TI1: 0000: no filter, sampling is done at f_{DTS} 0001: $f_{SAMPLING} = f_{CK\_INT}$, N = 2 0010: $f_{SAMPLING} = f_{CK\_INT}$, N = 4 0011: $f_{SAMPLING} = f_{CK\_INT}$, N = 8 0100: $f_{SAMPLING} = f_{DTS} / 2$, N = 6 0101: $f_{SAMPLING} = f_{DTS} / 2$, N = 8 0110: $f_{SAMPLING} = f_{DTS} / 4$, N = 6 0111: $f_{SAMPLING} = f_{DTS} / 4$, N = 8 1000: $f_{SAMPLING} = f_{DTS} / 8$, N = 6 1001: $f_{SAMPLING} = f_{DTS} / 8$, N = 8 1010: $f_{SAMPLING} = f_{DTS} / 16$, N = 5 1011: $f_{SAMPLING} = f_{DTS} / 16$, N = 6 1100: $f_{SAMPLING} = f_{DTS} / 16$, N = 8 1101: $f_{SAMPLING} = f_{DTS} / 32$, N = 5 1110: $f_{SAMPLING} = f_{DTS} / 32$, N = 6 1111: $f_{SAMPLING} = f_{DTS} / 32$, N = 8
3:2	IC1PSC	R/W	0x0	Input capture 1 prescaler: 00: no prescaler 01: capture is done once every 2 events 10: capture is done once every 4 events 11: capture is done once every 8 events

Bit	Name	Attribute	Reset Value	Description
				The IC1PSC register is reset when CC1E = 0.
1	RSV	-	-	Reserved
0	CC1S	R/W	0x0	Capture/compare channel 1 selection: 0: CC1 channel is configured as output. 1: CC1 channel is configured as input, IC1 is mapped on TI1. Note: CC1S bits are writable only when the channel is OFF (CC1E = 0).

18.6.7 Capture/Compare Enable Register (TIM_CCER)

Offset address: 0x20

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:4	RSV	-	-	Reserved
3	CC1NP	R/W	0x0	Capture/compare 1 complementary output polarity: CC1 channel configured as output: 0: OC1N is the inverse OC1REF. 1: OC1N is OC1REF. When CC1 channel is configured as input, it is used in conjunction with CC1P to select the polarity of IC1/IC2.
2	CC1NE	R/W	0x0	Capture/compare 1 complementary output enable: 0: disabled—OC1N is not active. 1: enabled—OC1N signal is output.
1	CC1P	R/W	0x0	Capture/compare 1 output polarity: **CC1 channel configured as output:** 0: OC1 is OC1REF. 1: OC1 is the inverse OC1REF. **CC1 channel configured as input:** 0: non-inverted mode: capture is done on the rising edge of IC1. 1: inverted mode: capture is done on the falling edge of IC1.

Bit	Name	Attribute	Reset Value	Description
0	CC1E	R/W	0x0	Capture/compare 1 output enable: **CC1 channel configured as output:** 0: OC1 not active 1: OC1 active **CC1 channel configured as input:** 0: capture disabled 1: capture enabled

18.6.8 Counter Register (TIM_CNT)

Offset address: 0x24

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:16	RSV	-	-	Reserved
15:0	CNT	R/W	0x0	Counter value

18.6.9 Prescaler register (TIM_PSC)

Offset address: 0x28

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:16	RSV	-	-	Reserved
15:0	PSC	R/W	0x0	Counter clock (CK_CNT) prescaler value: $f_{CK\_CNT} = f_{CK\_PSC} / (PSC[15:0] + 1)$ This is a preload register whose content are transferred into the shadow register at each update event.

18.6.10 Auto-reload Register (TIM_ARR)

Offset address: 0x2C

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:16	RSV	-	-	Reserved

Bit	Name	Attribute	Reset Value	Description
15:0	ARR	R/W	0x0	Auto-reload value at counter overflow: This is a preload register whose content are transferred into the shadow register at each update event.

18.6.11 Capture/Compare Register 1 (TIM_CCR1)

Offset address: 0x34

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:16	RSV	-	-	Reserved
15:0	CCR1	R/W	0x0	Capture/compare channel 1 register **If channel CC1 is configured as output:** This is a preload register containing the value to be compared to the counter and signaled on OC1 output. **If channel CC1 is configured as input:** CCR1 is the counter value transferred by the last input capture event, at this point the CCR1 register is read-only.

18.6.12 Break and Dead-time Register (TIM_BDTR)

Offset address: 0x44

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:16	RSV	-	-	Reserved
15	MOE	R/W	0x0	Main output enable: This register controls the output enable of all channels, and the independent channel is enabled if the respective enable bits (i.e. CCxE and CCxNE) are set. This bit is set by software or automatically set by hardware if AOE = 1. It is cleared asynchronously by hardware as soon as the break input is active. 0: OC and OCN outputs disabled (with specific

Bit	Name	Attribute	Reset Value	Description
				IO state determined by OSSI) 1: OC and OCN outputs enabled (determined by the states of respective CCxE and CCxNE bits)
14	AOE	R/W	0x0	Automatic output enable: 0: MOE can be set only by software 1: MOE can be set by software or automatically set at the next update event
13	BKP	R/W	0x0	Break polarity: 0: break input is active low 1: break input is active high
12	BKE	R/W	0x0	Break enable: 0: break input disabled 1: break input enabled
11	OSSR	R/W	0x0	Off-state (of channel at CCx(N)E = 0) selection for run mode (MOE = 1): This bit is used only when MOE = 1 on channels with complementary output enabled. 0: when the output channel is disabled, OC and OCN do not drive IO, and OCxN always drives IO. 1: when the output channel is disabled, OC and OCN drive GPIO to be “inactive” (referring to OIS in CR2). *(If cfg_timx_break_ossi0_disout = 1 / TIMCFGR[0] / [1] = 0, it will have no effect and will directly cause OC4 not to drive.)”
10	OSSI	R/W	0x0	Off-state (of channel at CCx(N)E = 0) selection for idle mode (MOE = 0): This bit is used only when MOE = 0 on channels configured as outputs. 0: when the output channel is disabled, OC and OCN do not drive GPIO. ● If cfg_timx_break_ossi0_disout = 0 / TIMCFGR[0] / [1][12] = 1, it will keep driving. ● OCxN always drives IO. 1: when the output channel is disabled, OC and OCN are forced with idle level first, and then with “inactive level” (referring to OIS in CR2)

Bit	Name	Attribute	Reset Value	Description
				after a dead-time. *(MOE=0 will make OCx(N) = (|CCx(N)P） &&OISx(N), OC4 = OIS4)
9:8	LOCK	R/W	0x0	Lock configuration: 00: no bit is write-protected 01: lock level 1—DTG, OISx, OISxN, BKE, BKP and AOE bits can no longer be written. 10: lock level 2— lock level 1 + CCxP, CCxNP, OSSR and OSSI bits can no longer be written. 11: lock level 3— lock level 2 + OCxM and OCxPE bits can no longer be written when related channel is configured in output. Note: The LOCK register cannot be overwritten after being written with a value other than 00, and the write-protected register can only be overwritten after the TIM module is reset.
7:0	DTG	R/W	0x0	This bit-filed defines the duration of the dead-time inserted between the complementary outputs. DT corresponds to this duration. DTG[7:5] = 0xx: DT = DTG[7:0] * t_{DTS} DTG[7:5] = 10x: DT = (64 + DTG[5:0]) * 2 * t_{DTS} DTG[7:5] = 110: DT= (32 + DTG[4:0]) * 8 * t_{DTS} DTG[7:5] = 111: DT= (32 + DTG[4:0]) * 16 * t_{DTS}

18.6.13 DMA Control Register (TIM_DCR)

Offset address: 0x48

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:13	RSV	-	-	Reserved
12:8	DBL	R/W	0x0	DMA burst length A read or write access to the TIM_DMAR register will trigger DMA transfer with a burst length of 1–18. 00000: 1 transfer 00001: 2 transfers

Bit	Name	Attribute	Reset Value	Description
				00010: 3 transfers …… 10001: 18 transfers Others: invalid value, write prohibited
7:5	RSV	-	-	Reserved
4:0	DBA	R/W	0x0	DMA base address, defined as the offset address directed to the register: 00000: TIM_CR1 00001: TIM_CR2 00010: TIM_SMCR …… Note: When DBA + DBL exceeds the TIM register address range, the actual burst transfer stops automatically when it reaches the highest TIM register address, i.e., the burst length is shortened.

18.6.14 DMA Access Register (TIM_DMAR)

Offset address: 0x4C

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:0	DMAR	R/W	0x0	DMA register for burst access When using DMA burst transfer, set the DMA channel peripheral address to TIM_DMAR. Accesses to this register will point to the register specified in TIM_DCR, and TIM will generate multiple DMA requests based on the DBL value.

18.7 Operation Procedure

18.7.1 Counting Mode

1. Enable the TIMx clock in RCM module.
2. Configure TIM_CR1[4] to set the counting direction.
3. Set TIM_CR1[7] to 1 to enable the auto-reload preload.
4. Configure TIM_PSC[15:0] to set the prescaler value.
5. Configure TIM_ARR[31:0] to set the auto-reload value.
6. Set TIM_CR1[2] to 1 so that an update interrupt or DMA request will be generated only at counter overflow or underflow.
7. Set TIM_CR1[1] to 0 to enable the update event.
8. Set TIM_EGR[0] to 1 so that when the software sets TIM_EGR[0], the counter will be reinitialized, the shadow register will be updated, and the prescaler counter will be cleared.
9. Set TIM_CR1[0] to 1 to enable the counter.
10. Set TIM_DIER[0] to 1 to enable the update event interrupt.

18.7.2 PWM Mode

1. Configure the GPIO alternate function to enable the TIMx clock in RCM module.
2. Configure TIM_CR1[4] to set the counting direction.
3. Set TIM_CR1[7] to 1 to enable the auto-reload preload.
4. Configure TIM_PSC[15:0] to set the prescaler value.
5. Configure TIM_ARR[31:0] to set the auto-reload value.
6. According to the output channel, set TIM_CCMRx[1:0/9:8] to 0 to configure channel x as output.
7. Configure TIM_CCMRx[6:4/14:12] to set PWM mode 1/2.

8. Configure TIM_CCER[1/5/9/13] to set the output polarity.
9. Set TIM_CCER[0/4/8/12] to 1 to enable channel x output.
10. Set TIM_CR1[2] to 1 so that an update interrupt or DMA request will be generated only at counter overflow or underflow.
11. Set TIM_CR1[1] to 0 to enable the update event.
12. Set TIM_EGR[0] to 1 so that when the software sets TIM_EGR[0], the counter will be reinitialized, the shadow register will be updated, and the prescaler counter will be cleared.
13. Set TIM_CR1[0] to 1 to enable the counter.
14. Set TIM_DIER[0] to 1 to enable the update event interrupt.
15. Configure TIM_CCRx[31:0] to set the compare value of channel x.

18.7.3 Input Capture Mode

1. Configure the GPIO alternate function to enable the TIMx clock in RCM module.
2. Configure TIM_CR1[4] to set the counting direction.
3. Set TIM_CR1[7] to 1 to enable the auto-reload preload.
4. Configure TIM_PSC[15:0] to set the prescaler value.
5. Configure TIM_ARR[31:0] to set the auto-reload value.
6. Configure TIM_CCMRx[1:0/9:8] to set channel CCx as input, and perform mapping as required.
7. Configure TIM_CCER[1/5/9/13] to set the capture polarity.
8. Configure TIM_CCMRx[7:4/15:12] to set the sampling frequency and filter length, generally set to 0.
9. Configure TIM_CCMRx[3:2/11:10] to set the prescaler value of input capture.
10. Set TIM_CCER[0/4/8/12] to 1 to enable the capture function.

11. Set TIM_EGR[0] to 1 so that when the software sets TIM_EGR[0], the counter will be reinitialized, the shadow register will be updated, and the prescaler counter will be cleared.
12. Set TIM_CR1[0] to 1 to enable the counter.
13. Set TIM_DIER[1/2/3/4] to 1 to enable the capture interrupt of channel x.

18.7.4 DMA Mode

In input capture mode, the channel capture value of TIMx is transferred to SRAM via DMA:

1. In input capture mode, before setting TIM_EGR[0] bit by software and enabling the counter, add the following configurations:
2. Configure TIM_DCR[12:8] to set the DMA burst length.
3. Configure TIM_DCR[4:0] to set the DMA base address. Generally, the base address here selects the capture/compare register corresponding to the capture channel.
4. Set TIM_DIER[9/10/11/12] to 1 to enable the CCx DMA request.
5. Set TIM_CR2[3] to 0 to generate CCxDMA request at CCx event.
6. For details on DMA controller configuration, please refer to chapter "11 Direct Memory Access Controller (DMA)".
7. After initiating DMA transfer, when a capture event occurs on the channel, DMA will transfer the value stored in base address to SRAM.

In output compare mode, the value in SRAM is transferred via DMA to the compare register of TIMx.

1. In PWM mode, before setting TIM_EGR[0] bit by software and enabling the counter, add the following configurations:
2. Configure TIM_DCR[12:8] to set the DMA burst length.
3. Configure TIM_DCR[4:0] to set the DMA base address. Generally, the base address here selects the capture/compare register corresponding to the compare channel.

4. Set TIM_DIER[9/10/11/12] to 1 to enable the CCx DMA request.
5. Set TIM_CR2[3] to 0 to generate CCxDMA request at CCx event.
6. For details on DMA controller configuration, please refer to chapter "11 Direct Memory Access Controller (DMA)".
7. After the DMA transfer is started, when the counter value matches the compare value, DMA will transfer the value in SRAM to the base address.

19 Basic Timer (TIM5)

19.1 Overview

The basic timer consists of a 16-bit auto-reload counter driven by a programmable prescaler.

19.2 Main Features

- 16-bit auto-reload upcounter
- 16-bit programmable prescaler allowing real-time adjustment of the counter clock frequency division
- Interrupt generation on the following event:
  - Counter overflow, counter initialization (triggered by software or hardware)

19.3 Functional Description

19.3.1 Time-base Unit

The main block of the time-base unit is a 16-bit counter with its related auto-reload register. The counter clock can be divided by a 16-bit prescaler.

The counter, the auto-reload register and the prescaler register can be written or read by software, which is true even when the counter is running.

The time-base unit includes:

- Counter register (TIM_CNT)
- Prescaler register (TIM_PSC)
- Auto-reload register (TIM_ARR)

The auto-reload register is preloaded, which is controlled by the auto-reload preload enable (ARPE) bit in the register. When ARPE = 0, write to the ARR register, and the written data is

directly transferred to the shadow register. When ARPE = 1, the data written to the ARR register is transferred to the shadow register when an update event (TIM_CNT overflow or underflow) occurs. The update event of ARR can also be generated by software.

The counter TIM_CNT is clocked by the prescaler output TIM_PSC, which is enabled only when the counter enable bit (CEN) in the register is set. When CNT = ARR, this round of counting is over and the update event is sent.

TIM_PSC is a synchronous prescaler that can divide the counter clock frequency by any factor between 1 and 65536. The PSC register is also buffered, and overwriting PSC does not actually overwrite the shadow register unless a new update event occurs. Thus the PSC register can be changed in real time on the fly, and the new prescaler ratio is taken into account at the next update event.

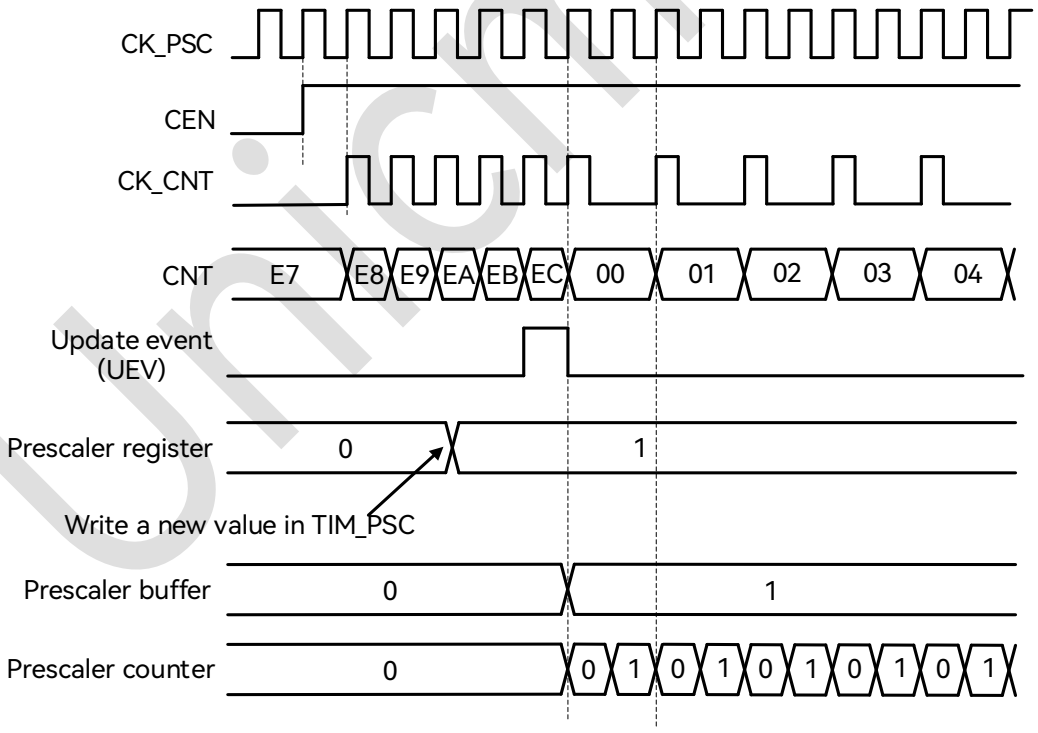


Figure 19-1: Counter Timing Diagram with Prescaler Division Changing from 1 to 2

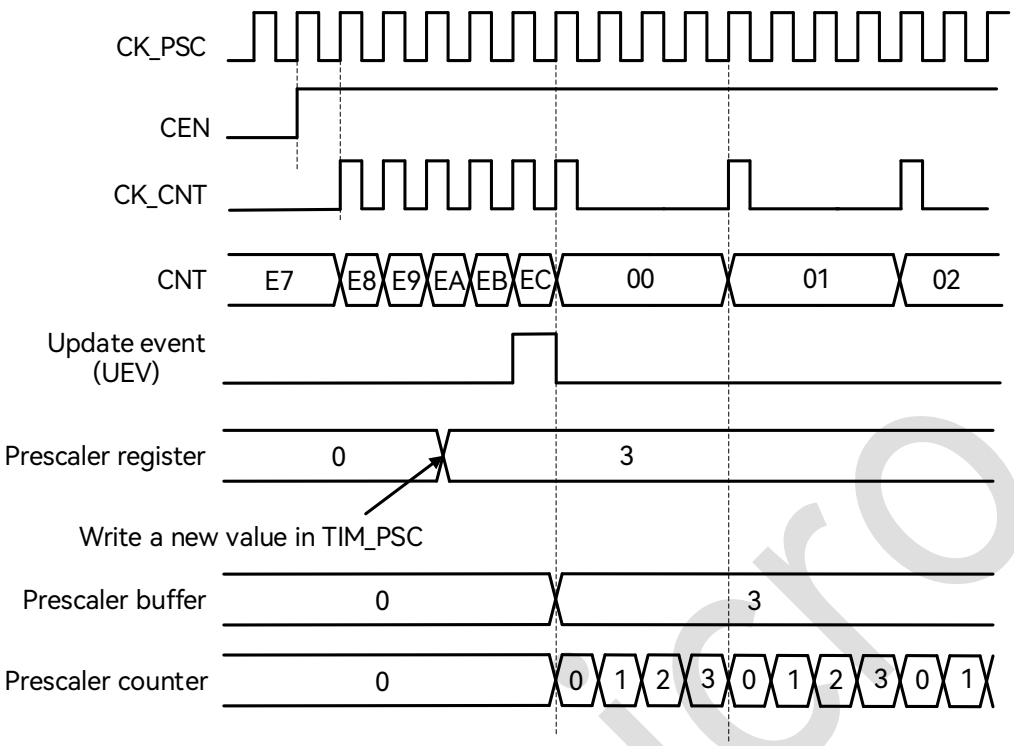


Figure 19-2: Counter Timing Diagram with Prescaler Division Changing from 1 to 4

19.3.2 Counter Operation Mode

The basic timer only supports up-counting mode.

In up-counting mode, the counter counts from 0 to the auto-reload value, i.e. CNT = ARR, generating an overflow event, and then restarts counting from 0.

If the repetition counter is enabled, the counter repeats the above process a number of times (RCR + 1) as defined in RCR before generating an underflow event.

The software can directly trigger an update event by setting the UG bit in the register, at which time the CNT and the prescaler registers are automatically cleared. Whether setting the UG register triggers UIF (update interrupt flag) is determined by the setting of the URS register.

The update event can be disabled by setting the UDIS bit in the register to avoid updating the shadow register while writing new values in the preload registers.

When an update event occurs, the following registers are updated and the UIF bit is set:

- The auto-reload shadow register ARR is reloaded with the content of TIM_ARR register.
- The prescaler shadow register PSC is reloaded with the content of TIM_PSC register.

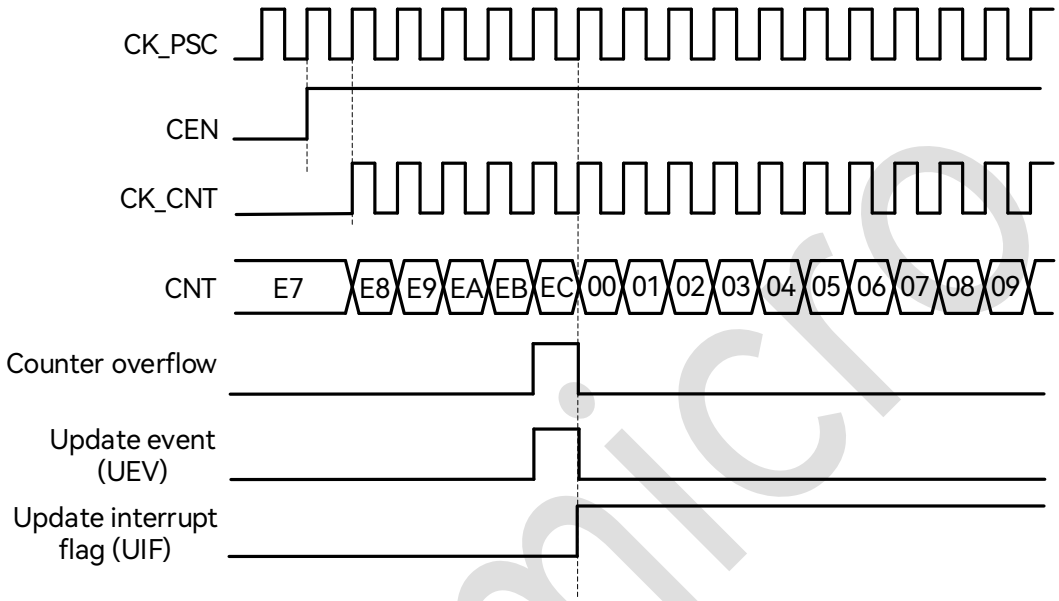


Figure 19-3: Up-counting Waveform Diagram, Internal Clock not Divided

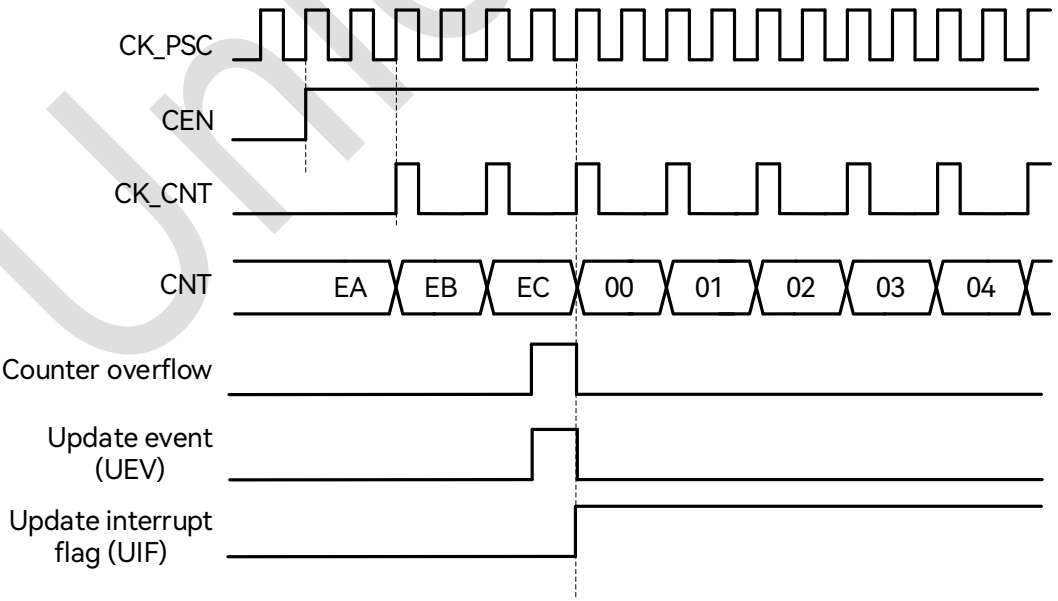


Figure 19-4: Up-counting Waveform Diagram, Internal Clock Divided by 2

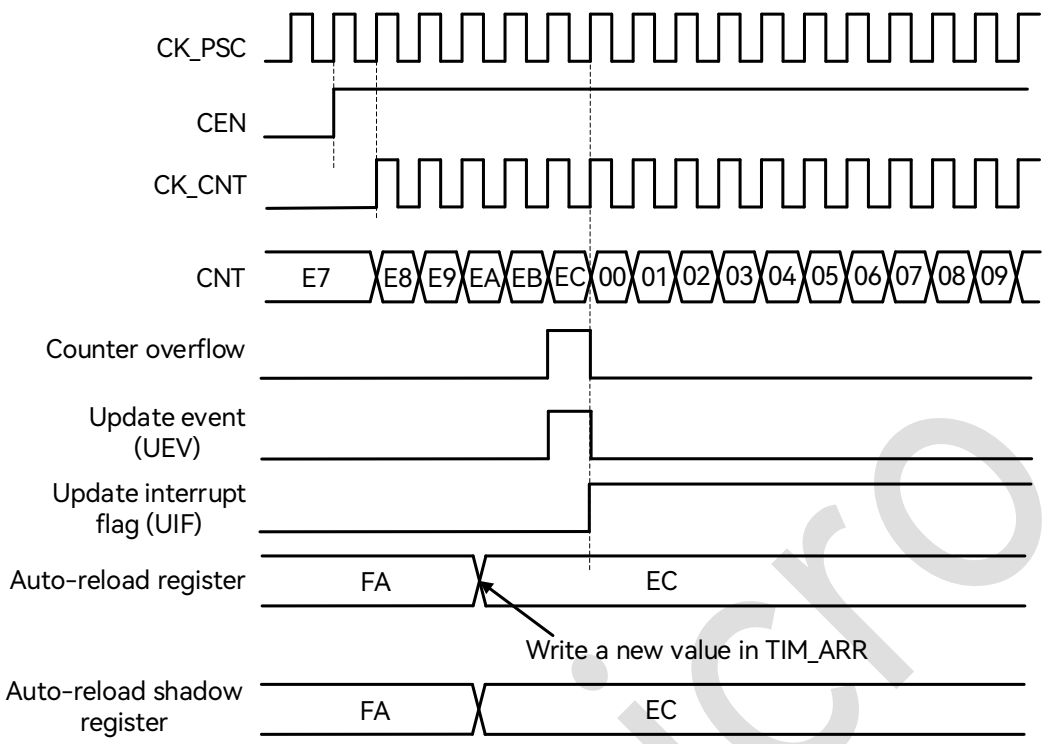


Figure 19-5: Counter Timing Diagram, Update Event when ARPE = 0 (TIM_ARR not Preloaded)

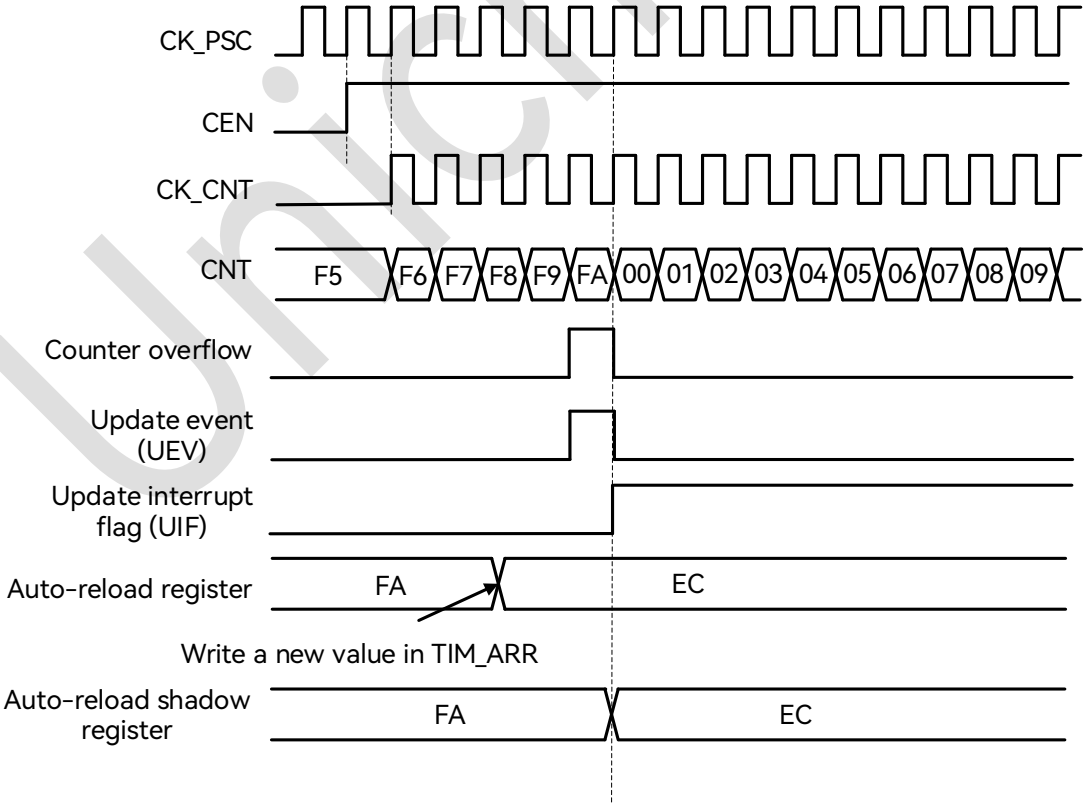


Figure 19-6: Counter Timing Diagram, Update Event when ARPE = 1 (TIM_ARR Preloaded)

19.3.3 Preload Register

- The following functional registers support the preload function:
  - Auto-reload register TIM_ARR
  - Prescaler register TIM_PSC (preload function cannot be disabled)

The preload function can be enabled or disabled by software for all of the above registers except TIM_PSC.

- Registers with preload function contain two sets of physical entities:
  - Shadow register: the register being used by the actual timer
  - Preload register: the register accessible to software
- When the preload function is disabled, the register with preload function has the following characteristics:
  - The preload register can be accessed and overwritten by software in real time.
  - The shadow register is updated synchronously with the preload register.
- If the preload function is enabled, then:
  - All software operations access the preload register.
  - At the occurrence of update event, the content of all preload registers will be synchronously transferred to the corresponding shadow registers.

19.3.4 Counter Clock

The counter clock can be provided by the following clock sources:

Internal clock: Timerx_clk

19.3.4.1 Internal Clock Source

If the slave mode controller is disabled (SMS = 000), then the CEN, DIR and UG bits are controlled by software.

After the UG bit is set and the update signal is synchronized by CLK_PSC, the counter value is reinitialized.

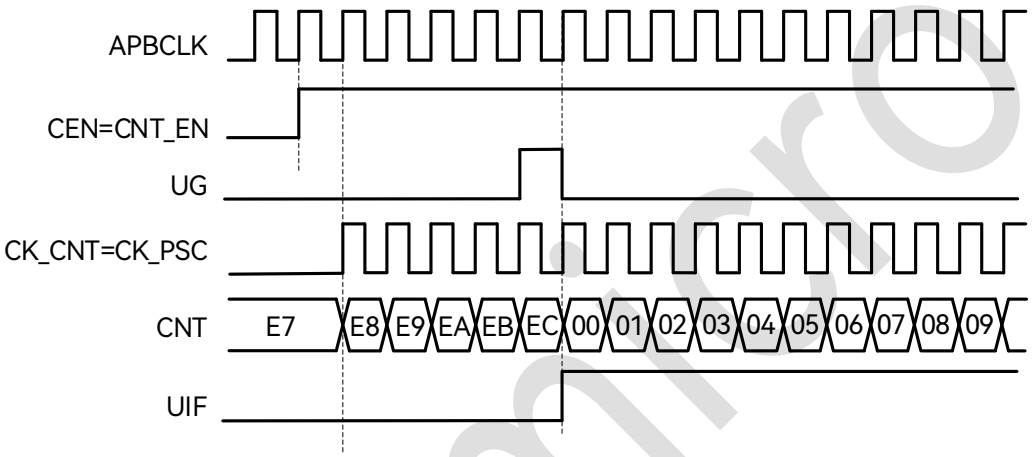


Figure 19-7: Timing Diagram in Internal Clock Source Mode, Clock Divided by 1

19.3.5 Debug Mode

When the CPU enters debug mode, the timer can either stop or continue working, and its behavior is defined by registers in the chip system.

When the timer is stopped during debugging, its output will be disabled (MOE is cleared). Depending on the register configuration, the output signal can be forced to be inactive or controlled by the GPIO module.

19.4 Register Description

TIM5 register base address: 0x40B0_C000

The registers are listed below:

Table 19-1: List of TIM5 Registers

Offset Address	Name	Description
0x00	TIM_CR1	Control register 1
0x04	TIM_CR2	Control register 2
0x0C	TIM_DIER	DMA and interrupt enable register
0x10	TIM_SR	Status register
0x14	TIM_EGR	Event generation register
0x24	TIM_CNT	Counter register
0x28	TIM_PSC	Prescaler register
0x2C	TIM_ARR	Auto-reload register

Registers are detailed in the following sections.

19.4.1 Control Register 1 (TIM_CR1)

Offset address: 0x00

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:8	RSV	-	-	Reserved
7	APRE	R/W	0x0	Auto-reload preload enable: 0: ARR not preloaded 1: ARR preloaded
6:4	RSV	-	-	Reserved
3	OPM	R/W	0x0	One-pulse mode output: 0: the counter does not stop at the occurrence of update event. 1: the counter stops at the occurrence of update event (CEN cleared automatically).
2	URS	R/W	0x0	Update request source: 0: an update interrupt or DMA request will be generated by any of the following events: ● Counter overflow/underflow

Bit	Name	Attribute	Reset Value	Description
				● Software setting the UG bit ● Update generated from the slave mode controller 1: an update interrupt or DMA request will be generated only at counter overflow or underflow.
1	UDIS	R/W	0x0	Update disable: 0: update event enabled; the update event can be generated by any of the following events: ● Counter overflow/underflow ● Software setting the UG bit 1: update event disabled, shadow register not updated; the counter and the prescaler will be reinitialized if the UG bit is set or if the slave mode controller receives a hardware reset.
0	CEN	R/W	0x0	Counter enable: 0: disabled 1: enabled Note: The external trigger mode can automatically set the CEN bit.

19.4.2 Control Register 2 (TIM_CR2)

Offset address: 0x04

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:7	RSV	-	-	Reserved
6:4	MMS	R/W	0x0	Master mode selection, selecting the TRGO trigger mode: 000: reset—TRGO is generated by the UG bit in the EGR register. 001: enable—TRGO is generated by the counter enable signal, including the CEN control bit and external trigger. 010: update—TRGO is generated by the update event. 011: compare pulse—TRGO is generated when an

Bit	Name	Attribute	Reset Value	Description
				input capture or compare event occurs that sets CC1F to 1. 100: compare—TRGO is generated by OC1REF. 101: compare—TRGO is generated by OC2REF. 110: compare—TRGO is generated by OC3REF. 111: compare—TRGO is generated by OC4REF.
3:0	RSV	-	-	Reserved

19.4.3 DMA / Interrupt Enable Register (TIM_DIER)

Offset address: 0x0C

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:9	RSV	-	-	Reserved
8	UDE	R/W	0x0	Update DMA request enable: 0: update DMA request disabled 1: update DMA request enabled
7:1	RSV	-	-	Reserved
0	UIE	R/W	0x0	Update interrupt enable: 0: update interrupt disabled 1: update interrupt enabled

19.4.4 Status Register (TIM_SR)

Offset address: 0x10

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:1	RSV	-	-	Reserved
0	UIF	R/W0C	0x0	Update event interrupt flag is set by hardware and cleared by software via writing it to 0. UIF is set and the shadow register is updated at the following events: ● Counter overflow occurs if repetition counter = 0 and UDIS = 0. ● The counter is reinitialized by software

Bit	Name	Attribute	Reset Value	Description
				setting the UG bit if URS = 0 and UDIS = 0. ● The counter is reinitialized by a trigger event if URS = 0 and UDIS = 0.

19.4.5 Event Generation Register (TIM_EGR)

Offset address: 0x14

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:1	RSV	-	-	Reserved
0	UG	W	0x0	This bit can be set by software to generate an update event, and is automatically cleared by hardware. When the software sets UG, the counter will be reinitialized, the shadow register will be updated, and the prescaler counter will be cleared.

19.4.6 Counter Register (TIM_CNT)

Offset address: 0x24

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:16	RSV	-	-	Reserved
15:0	CNT	R/W	0x0	Counter value

19.4.7 Prescaler Register (TIM_PSC)

Offset address: 0x28

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:16	RSV	-	-	Reserved
15:0	PSC	R/W	0x0	The counter clock (CK_CNT) frequency is: $f_{CK\_CNT} = f_{CK\_PSC} / (PSC[15:0] + 1)$ This is a preload register whose content are transferred into the shadow register at each

Bit	Name	Attribute	Reset Value	Description
				update event.

19.4.8 Auto-reload Register (TIM_ARR)

Offset address: 0x2C

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:16	RSV	-	-	Reserved
15:0	ARR	R/W	0x0	Auto-reload value at counter overflow: This is a preload register whose content are transferred into the shadow register at each update event.

19.5 Operation Procedure

19.5.1 Counter Startup

1. Enable the TIMx clock in RCM module.
2. Configure TIM_ARR[31:0] to set the auto-reload value.
3. Configure TIM_PSC[15:0] to set the prescaler value.
4. Set TIM_EGR[0] to 1 to generate a software update event and update the shadow register.
5. Set TIM_SR[0] to 0 to clear the update interrupt flag.
6. Set TIM_CR1[0] to 1 to enable the counter.

20 Low-power Timer (LPTIM0–LPTIM1)

20.1 Overview

LPTIM is a 16-bit low-power timer/counter module running in Always-on power domain. By selecting suitable clock source, LPTIM is able to keep running in various low-power modes with extremely low power consumption. LPTIM can be used as an external pulse counter in low-power mode even with no internal clock source. Also, in combination with an external input trigger signal, LPTIM is able to realize timeout wake-up from low-power modes.

This chip is provided with two low-power timers: LPTIM0 and LPTIM1. LPTIM0/1 are capable of waking up the system from sleep and stop modes through interrupts. After entering Standby0 mode, LPTIM0 can wake the system up through counter overflow interrupt, compare match interrupt, and external pulse counting interrupt (all interrupts generated by LPTIM0 can wake it up from Standby0). In contrast, LPTIM1 only supports wakeup from Standby0 through internal counter overflow interrupt and compare match interrupt; it does not support wakeup by external pulse counting interrupt.

20.2 Main Features

- 16-bit upcounter
- 3-bit asynchronous prescaler with 8 possible dividing factors (1, 2, 4, 8, 16, 32, 64, 128)
- Selectable clock source:
  - Internal clock sources: LSCLK (RCL or XTL), CLK1Hz, APB0 (PCLK0)
  - External clock source: LPTIMx_IN
- 16-bit compare register
- 16-bit destination register
- Selectable software / hardware input trigger

- Configurable input polarity
- External pulse counting with no clock source
- Externally triggered timeout wakeup from low-power modes
- Timed wakeup
- 16-bit PWM

20.3 System Block Diagram

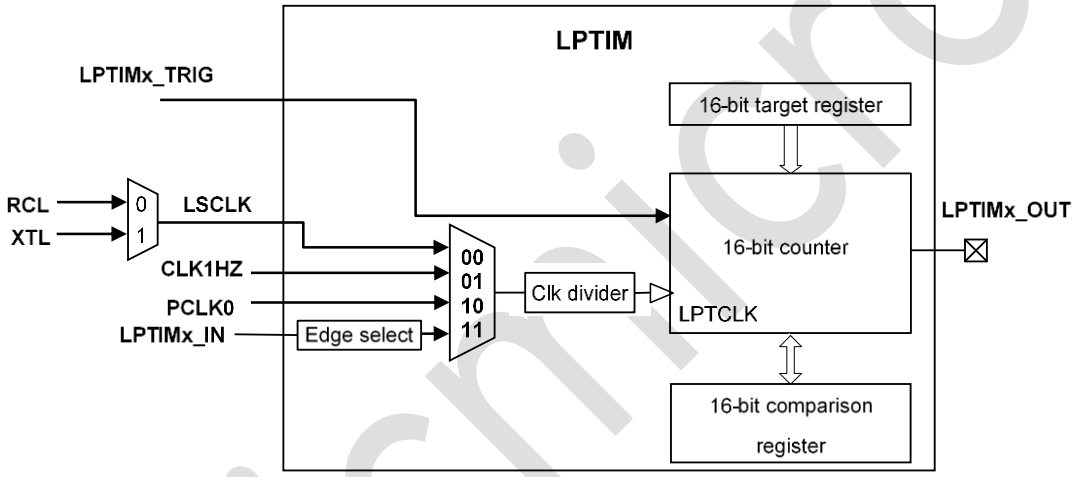


Figure 20-1: System Block Diagram

20.4 Pin Description

Table 20-1: LPTIM Pin Description

Function Pin	Alternate Function Pin	Direction	Functional Description
LPTIM0_IN	PC0, PA5, PB5	Input	LPTIM signal input (for external asynchronous pulse counting mode)
LPTIM0_TRIG	PC3, PB6	Input	LPTIM clock input (input capture)
LPTIM0_OUT	PC1, PA4, PB2	Output	LPTIM signal output (PWM)
LPTIM1_IN	PC8, PD10	Input	LPTIM signal input (for external asynchronous pulse counting mode)
LPTIM1_TRIG	PC9	Input	LPTIM clock input (input capture)
LPTIM1_OUT	PB7, PD2	Output	LPTIM signal output (PWM)

Note: In Standby0 Mode, LPTIM0 can wake up the system through PC0 and PC3 pins; while

LPTIM1 can only wake up by internal timing, but not by external pins of PD10/PD11.

20.5 Functional Description

20.5.1 General-purpose Timer

LPTIM can operate using either internal or external clocks, and after enabling there is a synchronization process of two counting clock cycles before operation begins.

20.5.2 Pulse-trigger Counting

LPTIM works with an internal clock, sampling the external asynchronous trigger signal, and can count the rising edge, falling edge, or both edges of the trigger signal. When the number of pulses reaches the set comparison value or overflows, the corresponding interrupt flag bit will be set to 1. There is a synchronization process of two counting clock cycles before and after enabling.

20.5.3 External Asynchronous Pulse Counting

LPTIM directly uses the external input pulse as the working clock, with polarity configurable to rising edge or falling edge. Once enabled, it starts working immediately without a synchronization process.

20.5.4 Timeout Mode

LPTIM operates with an internal or external clock, sampling the external asynchronous trigger signal. The counter will be started at the first sampling of the trigger signal, and will be cleared and restarted if another trigger signal is sampled after startup. If there is no new trigger before the counter overflows, an overflow interrupt will be generated, the counting will be stopped, and the enable will be cleared. There is a synchronization process of two counting clock cycles after enabling.

20.5.5 Counting Mode

The LPTIM features two counting modes:

Continuous counting mode: The counter keeps running after being enabled until it is disabled. Upon reaching the target value, it returns to 0 to restart counting, generating an overflow interrupt.

One-shot counting mode: The counter counts to the target value upon being triggered, then resets to 0 and automatically stops, generating an overflow interrupt. As the overflow signal and the LPTEN enable signal are in different clock domains, disabling the enable signal is implemented by asynchronous reset and synchronous release.

20.5.6 Externally Triggered Timeout Wakeup

LPTIM can be enabled by an external trigger signal or by software. In timeout mode, the first valid edge of the external trigger input will start the counter, while the subsequent trigger signal will clear the counter. If there is no effective trigger signal before the counter reaches the comparison value, a timeout interrupt will be generated to wake up the MCU.

The valid edge of the external trigger signal can be configured via registers, and since the external trigger signal is considered asynchronous input, there is a delay of at least two counting clock cycles for sampling and judgment.

20.5.7 16-bit PWM

After enabling PWM mode, LPTIM starts counting from 0x0000. The output goes high when the count value is equal to the comparison value, and goes low when the count value is equal to the final value. The PWM period is determined by the target value register, and the duty cycle is determined by the compare value register.

20.6 Register Description

LPTIM0 register base address: 0x40B0_9000

LPTIM1 register base address: 0x40B0_A000

The registers are listed below:

Table 20-2: List of LPTIM Registers

Offset Address	Name	Description
0x00	LPTIM_CFG	Configuration register
0x04	LPTIM_CNT	Counter register
0x08	LPTIM_CMP	Compare value register
0x0C	LPTIM_TARGET	Target value register
0x10	LPTIM_IE	Interrupt enable register
0x14	LPTIM_IF	Interrupt status register
0x18	LPTIM_CTRL	Control register

Registers are detailed in the following sections.

20.6.1 Configuration Register (LPTIM_CFG)

Offset address: 0x00

Reset value: 0x0000 0200

Bit	Name	Attribute	Reset Value	Description
31:16	RSV	-	-	Reserved
15	FLTEN	R/W	0x0	External trigger signal filter enable: When enabled, external trigger signals with a holding time shorter than two counting clock cycles will be filtered out. 0: filter disabled 1: filter enabled
14:13	RSV	-	-	Reserved
12:10	DIVSEL	R/W	0x0	Counter clock division selection: 000: divided by 1 001: divided by 2 010: divided by 4 011: divided by 8 100: divided by 16

Bit	Name	Attribute	Reset Value	Description
				101: divided by 32 110: divided by 64 111: divided by 128
9:8	CLKSEL	R/W	0x2	Counter clock source selection: 00: LSCLK (low-speed system clock RCL or XTL) selected as counter clock 01: RCLP selected as counter clock 10: PCLK of LPTIM selected as counter clock 11: LPTIM (external pin selected by SYSCFG->SYSCTRL1[11] or CLK_1Hz) selected as counter clock
7	EDGESEL	R/W	0x0	LPTIM input edge selection: 0: LPTIM counts at rising edge 1: LPTIM counts at falling edge
6:5	TRIGCFG	R/W	0x0	External trigger edge selection: 00: rising edge for external trigger input 01: falling edge for external trigger input 10/11: rising/falling edge for external trigger input
4	POLARITY	R/W	0x0	Counter clock polarity selection: 0: positive waveform, that is, when the first count value = the comparison value, the rising edge of the output waveform is generated. 1: negative waveform, that is, when the first count value = the comparison value, the falling edge of the output waveform is generated.
3	PWM	R/W	0x0	Pulse width modulation mode: 0: periodic square wave output 1: PWM output
2	MODE	R/W	0x0	Counting mode: 0: continuous counting mode: the counter keeps running after being triggered until it is disabled. Upon reaching the target value, the counter returns to 0 to restart counting, generating an overflow interrupt.

Bit	Name	Attribute	Reset Value	Description
				1: one-shot counting mode: after being triggered, the counter counts to the target value, then returns to 0 and automatically stops, generating an overflow interrupt.
1:0	TMODE	R/W	0x0	Operation mode selection: 00: general timer mode with waveform output 01: pulse-trigger counting mode 10: external asynchronous pulse counting mode 11: timeout mode

20.6.2 Counter Register (LPTIM_CNT)

Offset address: 0x04

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:16	RSV	-	-	Reserved
15:0	CNT	R	0x0	Counter value

20.6.3 Compare Value Register (LPTIM_CMP)

Offset address: 0x08

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:16	RSV	-	-	Reserved
15:0	CMP	R/W	0x0	Compare value register

20.6.4 Target Value Register (LPTIM_TARGET)

Offset address: 0x0C

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:16	RSV	-	-	Reserved
15:0	TARGET	R/W	0x0	Target value

20.6.5 Interrupt Enable Register (LPTIM_IE)

Offset address: 0x10

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:3	RSV	-	-	Reserved
2	TRIGIE	R/W	0x0	External trigger valid edge interrupt enable: 1: enabled 0: disabled
1	OVIE	R/W	0x0	Counter overflow interrupt enable: 1: enabled 0: disabled
0	COMPIE	R/W	0x0	Compare match interrupt enable: 1: interrupt enabled when the counter value matches the compare value 0: interrupt disabled when the counter value matches the compare value

20.6.6 Interrupt Flag Register (LPTIM_IF)

Offset address: 0x14

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:3	RSV	-	-	Reserved
2	TRIGIF	R/W1C	0x0	**External trigger valid edge interrupt flag:** 1: a valid edge detected on the external trigger input 0: no valid edge detected on the external trigger input
1	OVIF	R/W1C	0x0	**Counter overflow interrupt flag:** 1: counter overflow interrupt generated 0: no counter overflow interrupt generated
0	COMPIF	R/W1C	0x0	**Compare match interrupt flag:** 1: interrupt generated when the counter value matches the compare value 0: no interrupt generated

20.6.7 Control Register (LPTIM_CTRL)

Offset address: 0x18

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:1	RSV	-	-	Reserved
0	LPTEN	R/W	0x0	**LPTIM enable:** 1: counter counting enabled 0: counter counting disabled

20.7 Operation Procedure

20.7.1 Enabling LPTIM Clock

1. Write 0xABCD to the PMU_CPR register to enable its write operation.
2. Configure PMU_FCCR[3:2] to enable the LPTIM0/1 controller clock.
3. Configure PMU_FRCR[3:2] and release the LPTIM0/1 controller reset.
4. Write 0x459E to the PMU_CPR register to end its write operation.

20.7.2 LPTIM Counting

1. Enable the LPTIM clock.
2. Configure the LPTIM_CFG register, and write 0 to clear previous configurations.
3. Configure LPTIM_CFG[12:10] to set the clock division factor, configure LPTIM_CFG[9:8] to set the LPTIM clock source, configure LPTIM_CFG[2] to set the LPTIM counting mode, and configure LPTIM_CFG[1:0] to set the LPTIM operating mode (general timer mode with waveform output).
4. Configure LPTIM_TARGET [15:0] to set the target count value.
5. Configure LPTIM_IE[1] to enable the LPTIM overflow interrupt.
6. Configure LPTIM_CTRL[0] to enable LPTIM.

20.7.3 LPTIM PWM Output

1. Enable the LPTIM clock.
2. Configure the pin where LPTIM_OUT is located to enable its clock and multiplex its function to LPTIM_OUT.
3. Configure the LPTIM_CFG register, and write 0 to clear previous configurations.
4. Configure LPTIM_CFG[12:10] to set the clock division factor, configure LPTIM_CFG[9:8] to set the LPTIM clock source, configure LPTIM_CFG[2] to set the LPTIM counting mode, configure LPTIM_CFG[1:0] to set the LPTIM operating mode (general timer mode with waveform output), and set the pulse width modulation mode of PWM bit to PWM output.
5. Configure LPTIM_CMP [15:0] to set the count compare value.
6. Configure LPTIM_TARGET [15:0] to set the target count value.
7. Configure LPTIM_IE[0] to enable the LPTIM compare match interrupt.
8. Configure LPTIM_CTRL[0] to enable LPTIM.

20.7.4 LPTIM External Trigger Count Interrupt

1. Enable the LPTIM clock.
2. Configure the pin where LPTIM_TRIGGRT is located to enable its clock and multiplex its function to LPTIM_TRIGGER.
3. Configure the LPTIM_CFG register, and write 0 to clear previous configurations.
4. Configure LPTIM_CFG[12:10] to set the clock division factor, configure LPTIM_CFG[9:8] to set the LPTIM clock source, configure LPTIM_CFG[2] to set the LPTIM counting mode, configure LPTIM_CFG[1:0] to set the LPTIM operating mode (pulse-trigger counting mode), configure the TRIGCFG bit to set the valid edge for external trigger signal, and set FLTEN bit 1 to enable trigger signal filtering.
5. Configure LPTIM_TARGET [15:0] to set the target count value.
6. Configure LPTIM_IE[2] to enable the LPTIM trigger interrupt.

7. Configure LPTIM_CTRL[0] to enable LPTIM.

20.7.5 LPTIM Timeout Mode

1. Enable the LPTIM clock.
2. Configure the pin where LPTIM_TRIGGRT is located to enable its clock and multiplex its function to LPTIM_TRIGGER.
3. Configure the LPTIM_CFG register, and write 0 to clear previous configurations.
4. Configure LPTIM_CFG[12:10] to set the clock division factor, configure LPTIM_CFG[9:8] to set the LPTIM clock source, configure LPTIM_CFG[2] to set the LPTIM counting mode, configure LPTIM_CFG[1:0] to set the LPTIM operating mode (timeout mode), configure the TRIGCFG bit to set the valid edge for external trigger signal, and set FLTEN bit 1 to enable trigger signal filtering.
5. Configure the LPTM_TARGET register to set the target count value.
6. Configure the OVIE bit of LPTIM_IE register to enable the LPTIM overflow interrupt.
7. Configure the LPTEN bit of LPTIM_CTRL register to enable LPTIM.

20.7.6 LPTIM External Clock Source Count Overflow Interrupt

1. Enable the LPTIM clock.
2. Configure the pin where LPTIM_IN is located to enable its clock and multiplex its function to LPTIM_IN.
3. Configure the LPTIM_CFG register, and write 0 to clear previous configurations.
4. Configure the CLKSEL, DIVSEL and TMODE bits of the LPTIM_CFG register to correspondingly set the LPTIM clock source (with the CLKSEL bit set to 11), the clock division factor, and the LPTIM operating mode (external asynchronous pulse counting mode), and configure the EDGESEL bit to set the counting edge of the external clock source.

5. Configure LPTIM_TARGET [15:0] to set the target count value.
6. Configure LPTIM_IE[1] to enable the LPTIM overflow interrupt.
7. Configure LPTIM_CTRL[0] to enable LPTIM.

21 Independent Watchdog Timer (IWDT)

21.1 Overview

The watchdog timer can generate an interrupt or reset when the counter reaches the given timeout value. It can be used to regain control when the system fails to respond as expected due to software errors or external device failures.

21.2 Main Features

- 32-bit downcounter with programmable load
- Independent watchdog timer enabled
- Interrupt generation logic with interrupt masking
- Lockout register for software runaway protection
- Software boot function: reset enabling / disabling in IWDT control register

21.3 Register Description

IWDT register base address: 0x40B0_6000

The registers are listed below:

Table 21-1: List of IWDT Registers

Offset Address	Name	Description
0x00	IWDT_LOAD	Load register
0x04	IWDT_CNT	Counter register
0x08	IWDT_CTRL	Control register
0x0C	IWDT_CLR	Clear register
0x10	IWDT_INTRAW	Raw interrupt status register
0x14	IWDT_MINTS	Interrupt status register
0x18	IWDT_STALL	Clock division register
0x1C	IWDT_LOCK	Counter lock register

Registers are detailed in the following sections.

21.3.1 Load Register (IWDT_LOAD)

Offset address: 0x00

Reset value: 0x0000 3FFF

Bit	Name	Attribute	Reset Value	Description
31:0	LOAD	R/W	0x0000 3FFF	IWDT initial load value

21.3.2 Counter Register (IWDT_CNT)

Offset address: 0x04

Reset value: 0x0000 3FFF

Bit	Name	Attribute	Reset Value	Description
31:0	CNT	R	0x0000 3FFF	Counter value in IWDT If the clock source frequency is 32,768 Hz, the default overflow time is about 0.5 s.

21.3.3 Control Register (IWDT_CTRL)

Offset address: 0x08

Reset value: 0x8000 000C

Bit	Name	Attribute	Reset Value	Description
31	WRC	R	0x1	The setting of the IWDT load value or writing to the IWDT_CTRL register will take effect after a certain delay (of 3 to 4 non-divided IWDTCLK cycles) when writing to the IWDT_LOAD or IWDT_CTRL registers. During this delay, the status of the setting bit may not yet be effective. 0: the setting bit is not yet effective. 1: the setting bit is effective.
30:4	RSV	-	-	Reserved
3	HWIWDT_EN	R/W	0x1	Hardware enable bit for starting IWDT in EFC: 0: disabled 1: enabled

Bit	Name	Attribute	Reset Value	Description
				Note: When enabled, if the IWDT is started from the NVR area of EFC, the system will automatically begin watchdog counting after power-up.
2	RST_MODE	R/W	0x1	IWDT overflow reset mode selection (without interrupt enabled): 0: overflow reset shall be counted twice for the first time 1: overflow reset shall be counted once
1	RSTEN	R/W	0x0	IWDT overflow reset enable: 0: disabled 1: enabled
0	INTEN	R/W	0x0	IWDT interrupt enable: 0: disabled 1: enabled Note: Overflow interrupt has higher priority than overflow reset.

Notes:

When Rst_mode is set to 0 without enabling interrupt, the overflow reset shall be counted twice for the first time.

1. If the load register (IWDT_LOAD) is configured before the first overflow, it will reset the count twice. After that, if the load register is configured before the first overflow, it will reset the count twice. However, if the load register is configured once after the first overflow, subsequent configurations of the load register will only reset the overflow count once.
2. After the first overflow, if the load register (IWDT_LOAD) is configured, then each time the load register is configured before the next overflow, it will only reset the overflow count once.

21.3.4 Clear Register (IWDT_CLR)

Offset address: 0x0C

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:0	CLR_CARRY	W	0x0	Writing any value to this register will clear the IWDT overflow status, thus clearing the interrupt and reset.

21.3.5 Raw Interrupt Status Register (IWDT_INTRAW)

Offset address: 0x10

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:1	RSV	-	-	Reserved
0	INTRAW	R	0x0	Raw interrupt register, without interrupt enable masking: 0: no overflow occurred in IWDT 1: overflow occurred in IWDT

21.3.6 Interrupt Status Register (IWDT_MINTS)

Offset address: 0x14

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:1	RSV	-	-	Reserved
0	INTMS	R	0x0	IWDT interrupt flag bit: 0: no interrupt occurred in IWDT 1: interrupt occurred in IWDT

21.3.7 Clock Division Register (IWDT_STALL)

Offset address: 0x18

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:16	CLK_DIV	R/W	0x0	IWDT counter clock division value: 0x0: no frequency division 0x1: divided by 2 0x2: divided by 3 …. 0xFFFF: 0xFFFF+ divided by 1
15:9	RSV	-	-	Reserved
8	STALL	R/W	0x0	Enable bit for IWDT not counting when the chip is in HALT state: 0: counter stop function disabled in HALT state 1: counter stop function enabled in HALT state
7:0	RSV	-	-	Reserved

21.3.8 Counter Lock Register (IWDT_LOCK)

Offset address: 0x1C

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:0	LOCK	R/W	0x0	IWDT lock function enable: It is not locked by default. When the LOCK function is enabled, all WDT registers except this one are not writable. Write any value except 0x1ACCE551 to this register can enable IWDT LOCK function, while write 0x1ACCE551 to this register can clear the LOCK function. Read this register: returning 1 indicates a locked state, while returning 0 indicates an unlocked state.

21.4 Operation Procedure

21.4.1 Enabling IWDT Clock

1. Write 0xABCD to the PMU_CPR register to enable its write operation.
2. Configure PMU_FCCR[1] to enable the IWDT controller clock.
3. Configure PMU_FRCR[1] to release the IWDT controller reset.
4. Write 0x459E to the PMU_CPR register to end its write operation.

21.4.2 IWDT Load Value Setting

1. Write 0x1ACCE551 to the IWDT_LOCK register to unlock the register.
2. Configure IWDT_LOAD[31:0] to set the IWDT counter load value.
3. Wait for IWDT_CTRL[31] to be set, i.e., the setting takes effect.
4. Write any value except 0x1ACCE551 to the IWDT_LOCK register to lock the register.

21.4.3 IWDT Counter Overflow Interrupt

1. Enable the IWDT clock.
2. Write 0x1ACCE551 to the IWDT_LOCK register to unlock the register.
3. Configure IWDT_STALL[31:16] to set the IWDT clock division.
4. Initialize the IWDT counter overflow interrupt by setting IWDT_CTRL[1] to 1 to enable the IWDT interrupt.
5. Wait for IWDT_CTRL[31] to be set, i.e., the setting takes effect.
6. Configure IWDT_LOAD[31:0] to set the IWDT counter load value.
7. Wait for IWDT_CTRL[31] to be set, i.e., the setting takes effect.

8. Write any value except 0x1ACCE551 to the IWDT_LOCK register to lock the register.

21.4.4 IWDT Counter Overflow Reset

1. Enable the IWDT clock.
2. Write 0x1ACCE551 to the IWDT_LOCK register to unlock the register.
3. Configure IWDT_STALL[31:16] to set the IWDT clock division.
4. Configure IWDT_CTRL[2] to set the reset condition to require either one complete count or two complete counts for the first reset.
5. Wait for IWDT_CTRL[31] to be set, i.e., the setting takes effect.
6. Configure IWDT_CTRL[1] to enable the IWDT overflow reset.
7. Wait for IWDT_CTRL[31] to be set, i.e., the setting takes effect.
8. Write IWDT_LOAD[31:0] to set the IWDT counter load value.
9. Wait for IWDT_CTRL[31] to be set, i.e., the setting takes effect.
10. Write any value except 0x1ACCE551 to the IWDT_LOCK register to lock the register.

22 System Window Watchdog (WWDT)

22.1 Overview

The system window watchdog is a watchdog running synchronously with CPU, aiming at monitoring the running status of CPU in real time, so that it can reset CPU in the case of abnormal operation to avoid unpredictable consequences.

To ensure synchronization and real-time performance, the WWDT operates using the PCLK clock, with an internal prescaler circuit to generate a synchronized count enable signal.

22.2 Main Features

- Counter clock PCLK0 (APB0 PCLK)
- Up-counting mode: the counter counts from 0 to the overflow time.
- The overflow time can be selected as 1/ 4/ 16/ 64/ 128/ 256/ 512/ 1024/ 2048/ 4096/ 8192/ 16384/ 32768/ 65536 times of 4096 PCLK0 cycles.
- The window period is defined as the period when the counter is greater than or equal to 50% of the overflow time.
- With early warning interrupt capability, an interrupt will occur when the count reaches 75% of the overflow time.

22.3 Functional Description

WWDT is disabled by default after the chip is reset, and the software needs to write 0x5A to the control register (WWDT_CTRL) to activate WWDT. After WWDT is activated, if the software writes 0xAC to WWDT control register (WWDT_CTRL) during the window period, the counter will be cleared. Once WWDT is enabled, it cannot be disabled again except by a reset.

The WWDT operates using PCLK with a built-in 4096 times prescaler circuit. The overflow time can be selected as 1/4/16/64/128/256/512/1024/2048/4096/8192/16384/32768/65536 times of 4096 PCLK0 cycles. The overflow time length is calculated as follows:

$$t_{WWDT} = f_{PCLK} \times 4096 \times N_{CFG}$$

The following table shows some calculation examples:

Table 22-1: WWDT Counter Overflow Time Calculation

PCLK0 Frequency (f_{PCLK})	Overflow Length Configuration (N_{CFG})	Overflow Time (t_{WWDT}) (ms)
42 MHz	1	0.097523
	4	0.390095
	16	1.560381
	64	6.241524
	128	12.483047
	256	24.966095
	512	49.93219
	1024	99.864381
	2048	199.728762
	4096	399.457524
	8192	798.915048
	16384	1597.830096
	32768	3195.660192
	65536	6391.320384

WWDT can only be cleared during the window period, otherwise a reset will be triggered directly. The enable window is the second half of the counter cycle, and the software should pay attention to query the count value before clearing the watchdog.

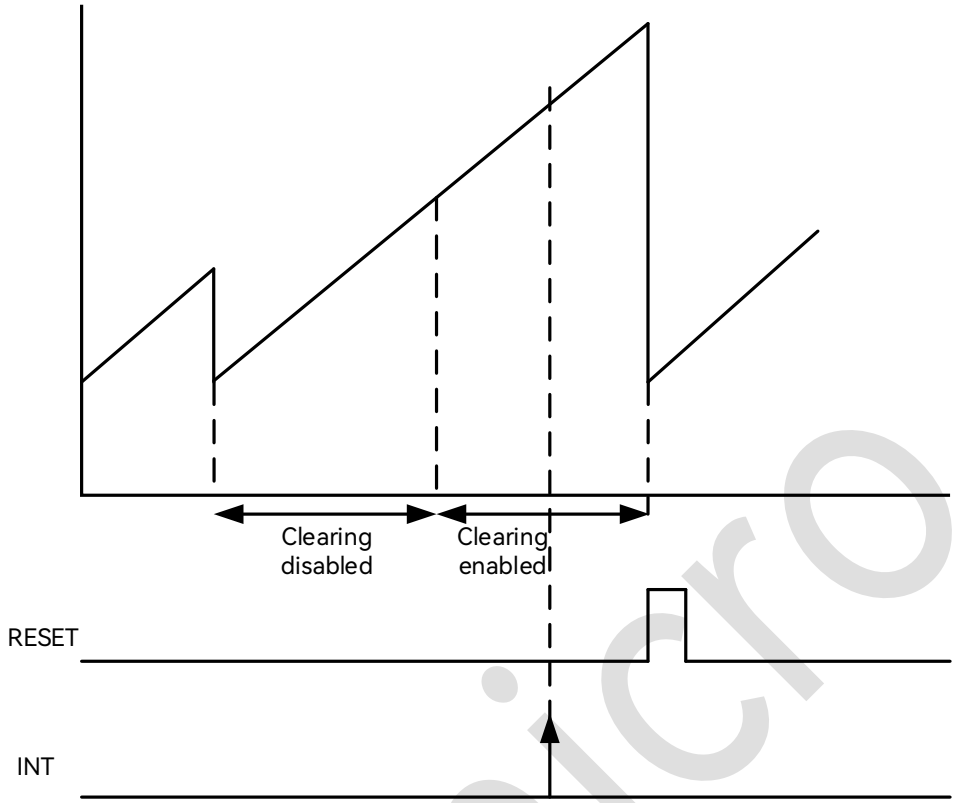


Figure 22-1: WWDT Counter Refresh Timing Diagram

WWDT will generate a CPU reset when any of the following events occurs:

- Counter overflow
- Write a value other than 0xAC to the WWDT control register (which can be used to trigger CPU software reset).
- Write 0xAC to the WWDT control register during window close period.

An early warning interrupt will be triggered when the counter reaches 75% of the overflow time.

22.4 Register Description

WWDT register base address: 0x40B0_5000

The registers are listed below:

Table 22-2: List of WWDT Registers

Offset Address	Name	Description
0x00	WWDT_CTRL	Control register
0x04	WWDT_CFG	Configuration register
0x08	WWDT_CNT	Counter register
0x0C	WWDT_IE	Interrupt enable
0x10	WWDT_IF	Interrupt flag register
0x14	DIV_CNT	PCLK prescaler counter register

22.4.1 Control Register (WWDT_CTRL)

Offset address: 0x00

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:8	RSV	-	-	Reserved
7:0	WWDT_CTRL	W	0x0	CPU writing 0x5A to this bit will start WWDT. After starting WWDT, CPU writing 0xAC to this bit will clear the counter. After starting WWDT, CPU writing a value other than 0xAC to this bit will reset the system.

22.4.2 Configuration Register (WWDT_CFG)

Offset address: 0x04

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:4	RSV	-	-	Reserved
3:0	WWDT_CFG	R/W	0x0	Configure watchdog overflow time: 0000: $t_{PCLK} \times 4096 \times 1$ 0001: $t_{PCLK} \times 4096 \times 4$

Bit	Name	Attribute	Reset Value	Description
				0010: t_{PCLK} × 4096 × 16 0011: t_{PCLK} × 4096 × 64 0100: t_{PCLK} × 4096 × 128 0101: t_{PCLK} × 4096 × 256 0110: t_{PCLK} × 4096 × 512 0111: t_{PCLK} × 4096 × 1024 1000: t_{PCLK} × 4096 × 2048 1001: t_{PCLK} × 4096 × 4096 1010: t_{PCLK} × 4096 × 8192 1011: t_{PCLK} × 4096 × 16384 1100: t_{PCLK} × 4096 × 32768 1101: t_{PCLK} × 4096 × 65536 1110: reserved 1111: reserved

22.4.3 Counter Register (WWDT_CNT)

Offset address: 0x08

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:16	RSV	-	-	Reserved
15:0	WWDT_CNT	R	0x0	WWDT counter register value, which the software can query to know the WWDT timing progress.

22.4.4 Interrupt Enable Register (WWDT_IE)

Offset address: 0x0C

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:1	RSV	-	-	Reserved
0	WWDT_IE	R/W	0x0	WWDT interrupt enable: 0: disabled 1: enabled

22.4.5 Interrupt Flag Register (WWDT_IF)

Offset address: 0x10

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:1	RSV	-	-	Reserved
0	WWDT_IF	R/W1C	0x0	WWDT 75% timing interrupt flag: 0: no interrupt generated 1: interrupt flag is set

22.4.6 PCLK Prescaler Counter Register (DIV_CNT)

Offset address: 0x14

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:12	RSV	-	-	Reserved
11:0	DIV_CNT	R	0x0	This register is the PCLK prescaler counter. The Wwdt_cnt will increment by 1 after every 4096 PCLK cycles.

22.5 Operation Procedure

22.5.1 Interrupt Mode

1. Write 0xA5A55A5A to RCM_RCMPR register to enable its write operation.
2. Configure RCM_APB0CKENR[10] to enable the WWDT controller clock.
3. Configure RCM_APB0RSTR[10] to release the WWDT controller reset.
4. Write 0xFFFFFFFF to RCM_RCMPR register to end its write operation.
5. Configure WWDT_CFG[3:0] to set the overflow time of WWDT counter.
6. Initialize the WWDT interrupt, and configure WWDT_IE[0] to enable WWDT interrupt.
7. Write 0x5A to the WWDT_CTRL register to start the WWDT counter.

8. The window period is defined as the period when the counter is greater than or equal to 50% of the overflow time, and the following operations are performed in different time periods:

   A. When the count value is less than 50% of the overflow time, the watchdog is fed, and the system is reset directly.

   B. When the count value is greater than 50% of the overflow time, the watchdog is fed, the WWDT counter is cleared, and the counting restarts.

   C. When the count value reaches 75% of the overflow time, a WWDT warning interrupt is triggered.

   D. When the count value exceeds the overflow time, the system generates a reset.

9. Wait for the counter register (WWDT_CNT) to count to 75% of the overflow time, an interrupt is triggered. If the configuration register (WWDT_CFG) is configured with TPCLK × 4096 × 1, then based on the count value in the PCLK prescaler counter register (DIV_CNT), the interrupt is triggered when it counts to 0xC00.

10. The interrupt flag register (WWDT_IF) is cleared by writing 1 to it.

22.5.2 WWDT Clear Counter

Write 0xAC to the WWDT_CTRL register to clear the counter.

23 Real-time Clock (RTC)

23.1 Overview

The battery backup unit (BBU) is a set of functional modules that includes a real-time clock (RTC) with a calendar, an alarm, a periodic wake-up source, a tamper detector, and an 80-byte backup register.

23.2 Main Features

The BBU supports the following features:

- The backup system is always running.
  - Real-time clock (RTC) module
  - Tamper detector
  - 20-word (i.e., 80-byte) backup register
- Programmable alarm capable of generating interrupts

23.3 Block Diagram of Modules with System Connection

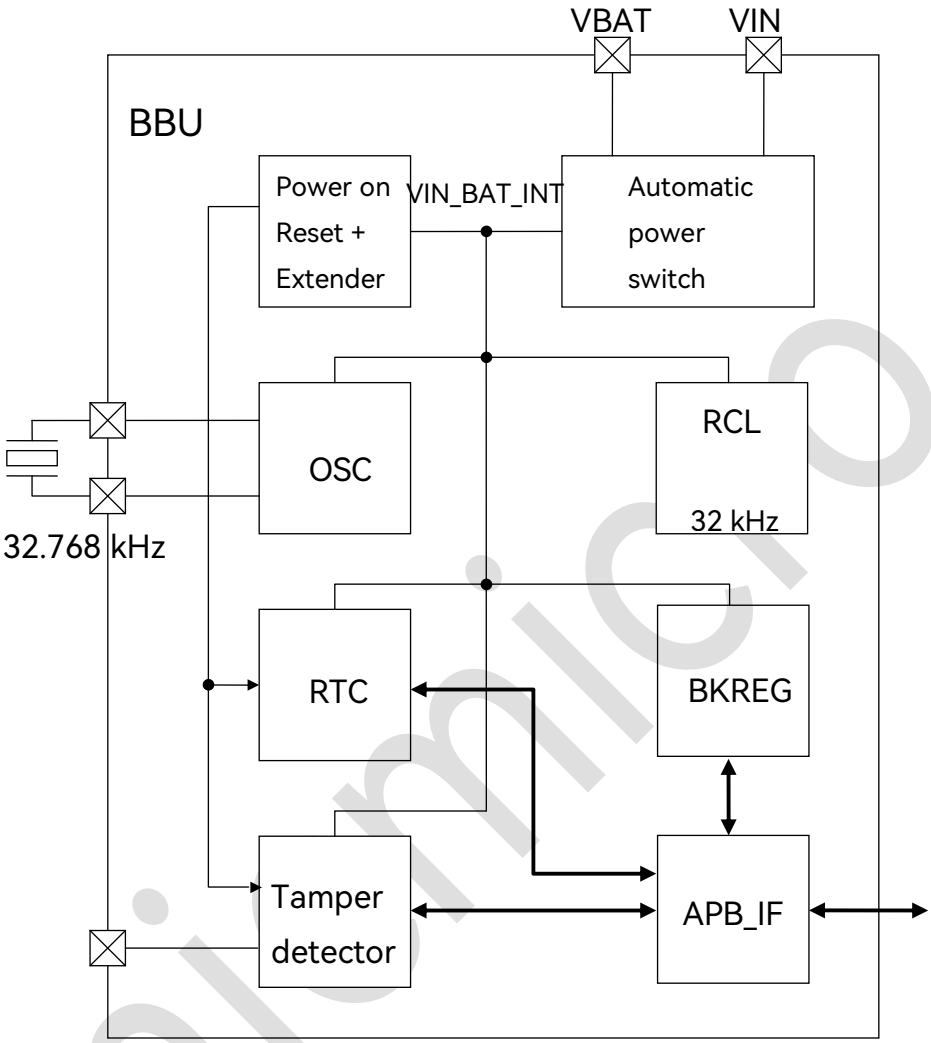


Figure 23-1 : BBU Module with Analog Module

Signal boundaries cross different power domains require isolation and level shifter units, as shown in Figure 23-1.

23.4 Module Description

23.4.1 RTC

23.4.1.1 Time and Calendar Counters

The RTC module is capable of providing real-time date and time information that can be accessed by the core and peripheral devices.

Date and time counters:

- Centisecond (1% of a second) counter (CENTISEC)
- Second counter (SECOND)
- Minute counter (MINUTE)
- Hour counter (HOUR)
- Week counter (WEEK)
- Day counter (DAY)
- Month counter (MONTH)
- Year counter (YEAR and CENTURY)

YEAR stands for the last two digits of the year. CENTURY indicates the 2000s (21st century) or 2100s (22nd century).

Each register is coded in BCD (binary-coded decimal) format.

23.4.1.2 Centisecond Counter (CENTISEC)

This counter continuously increments from 00 to 99 every second. It increments once every 326~329 low-speed clock cycles, depending on the second period after calibration. The jitter in one second is controlled within only one HCLK cycle.

23.4.1.3 Time Counter (SECOND, MINUTE and HOUR)

The time counter behaves as follows:

SECOND: When the second counter counts from 59 to 00 (carry), the minute counter will be incremented by 1.

MINUTE: When the minute counter counts from 59 to 00 (carry), the hour counter will be incremented by 1.

HOUR: The hour counter consists of H20_PA (1 bit) and HOUR19 (5 bits).

- In 12-hour format, the day counter is incremented by 1 when [H20_PA, HOUR19] is incremented from [1,11] (11:00 p.m.) to [0,12] (12:00 a.m.).
- In 24-hour format, the day counter is incremented by 1 when [H20_PA, HOUR19] is incremented from [1,3] (23:00) to [0,0] (00:00).

By setting the HOUR12_24 bit in the RTC_TIME register, the hour format can be selected from either the 12-hour or 24-hour format.

- 0: 12-hour format
- 1: 24-hour format

As shown in the table below, HOUR is represented by two registers: HOUR19 and H20_PA, with the value of HOUR19 encoded in BCD format. The time format depends on the clock system settings. In the 12-hour clock mode, H20_PA indicates AM or PM. In the 24-hour clock mode, H20_PA represents the hour of 20 plus.

Table 23-1: HOUR Register Coding

Time	24-hour Format		12-hour Format	
	H20_PA	HOUR19	H20_PA	HOUR19
0 a.m.	0	0	0	12
1 a.m.	0	1	0	1
2 a.m.	0	2	0	2
3 a.m.	0	3	0	3
4 a.m.	0	4	0	4
5 a.m.	0	5	0	5
6 a.m.	0	6	0	6
7 a.m.	0	7	0	7
8 a.m.	0	8	0	8
9 a.m.	0	9	0	9
10 a.m.	0	10	0	10
11 a.m.	0	11	0	11
0 p.m.	0	12	1	12

Time	24-hour Format		12-hour Format	
	H20_PA	HOUR19	H20_PA	HOUR19
1 p.m.	0	13	1	1
2 p.m.	0	14	1	2
3 p.m.	0	15	1	3
4 p.m.	0	16	1	4
5 p.m.	0	17	1	5
6 p.m.	0	18	1	6
7 p.m.	0	19	1	7
8 p.m.	1	0	1	8
9 p.m.	1	1	1	9
10 p.m.	1	2	1	10
11 p.m.	1	3	1	11

23.4.1.4 Date Counter

Week counter (WEEK)

The counter value ranges from 0 to 6, and a 7 is treated as 0 when entered (coding rules: 0 or 7 = Sunday, 1 = Monday, ..., 5 = Friday, 6 = Saturday). The relationship between the WEEK counter and the actual day of the week is user-defined.

Calendar counter (DAY, MONTH, YEAR, CENTURY)

The automatic calendar function provides calendar numbers as shown below:

- DAY: the counter ranges from:
  - 01 to 31: in January, March, May, July, August, October, December
  - 01 to 30: in April, June, September, November
  - 01 to 29: in February of a leap year
  - 01 to 28: in February of a non-leap year

When the day counter jumps back to 01, the month counter will carry forward.

- MONTH: it ranges from 01 to 12, and jumping back to 01 will carry it to the year counter.

- YEAR: it ranges from 00 to 99, and jumping back to 00 will clear the century counter.
- CENTURY:
  - When the century counter is 0, YEAR = 04, 08, …, 92, 96 are leap years.
  - When the century counter is 1, YEAR = 00, 04, 08, …, 92, 96 are leap years.
  - The leap year can be correctly identified in the range from 2000 to 2399.

23.4.1.5 Time and Date Setting

To prevent accidental carry when writing the time and date counters, all timers will be frozen until the end of the write operation.

The time/date setting operation should follow the following sequence:

1. Write 1 to RTC_WRSTA, and the counter will stop timing.
2. Complete the write operation as soon as possible. **Note that any non-existing time or date settings will be ignored.**
3. Write 1 to RTC_WRSTP and start timing from this point.

23.4.1.6 Time and Date Reading

Before reading the time and date registers, RTC needs to copy the real-time values of the time/date registers into the shadow reading register. This isolation ensures the integrity of the data being read, preventing changes to the time/date values due to time updates. The time/date read operation shall follow the sequence below:

1. Write 1 to RTC_RDSTA.
2. Complete the read operation as soon as possible.
3. Write 1 to RTC_RDSTP.

23.4.1.7 Alarm Clock

The BBU provides two programmable alarms: alarm 1 and alarm 2.

The registers starting with ALM1_ are two alarm configuration registers ALM1_TIME and ALM1_DATE, and an alarm enable register ALM1_EN.

The ALM1_EN register controls the status of the alarm. When all bits of the ALM1_EN register are 0, alarm 1 is disabled. For each bit, 1 indicates that the alarm is only enabled when the corresponding register matches, while 0 indicates that the alarm is enabled regardless of the corresponding register value. The alarm flag and interrupt will be generated when the time increases to match the enabled time set by the ALM1_DATE and ALM1_TIME registers.

Example of alarm 1 setting is shown in the table below:

Table 23-2: Example of Alarm 1 Setting

ALM1 Date/Time Registers								ALM1_EN Register								Condition of Alarm 1
ALM1_YEAR	ALM1_MONTH	ALM1_DAY	ALM1_WEEK	ALM1_HOUR	ALM1_MINUTE	ALM1_SECOND	ALM1_CENTISEC	ALM1_EN_YEAR	ALM1_EN_MON	ALM1_EN_DAY	ALM1_EN_WEEK	ALM1_EN_HOUR	ALM1_EN_MIN	ALM1_EN_SEC	ALM1_EN_CS	
2	3	4	5	6	7	8	9	0	0	0	0	0	0	0	0	Disabled
2	3	4	5	6	7	8	9	0	0	0	0	0	1	1	0	07:08 in every hour
2	3	4	5	6	7	8	9	1	1	1	0	1	1	1	0	06:07:08 on March 4, 2002
2	3	4	5	6	7	8	9	0	0	1	0	1	1	1	0	06:07:08 on the 4th of every month
2	3	4	5	6	7	8	9	0	0	0	1	1	1	1	0	06:07:08 on every Friday

Alarm 2 generates periodic interrupts. Depending on the register settings, the interrupt interval varies from every minute to every 1/128th of a second.

Alerts can be checked by registers RTC_INT_RAW and RTC_INT_STA.

Registers RTC_INT_EN and RTC_INT_CLR control the output of the above interrupts to the CPU, and the interrupt status retains its value until it is cleared by APB access.

23.4.1.8 Oscillator Calibration

As shown in the figure below, this module can calibrate the input frequency of the low-speed clock to within ±5 ppm of the nominal 32.768 kHz. The calibration function changes the maximum value of the clock counter from 32768 to the specified value. The calibration step is ±1 cycle, and the maximum calibration range is from -128 cycles to +127 cycles.

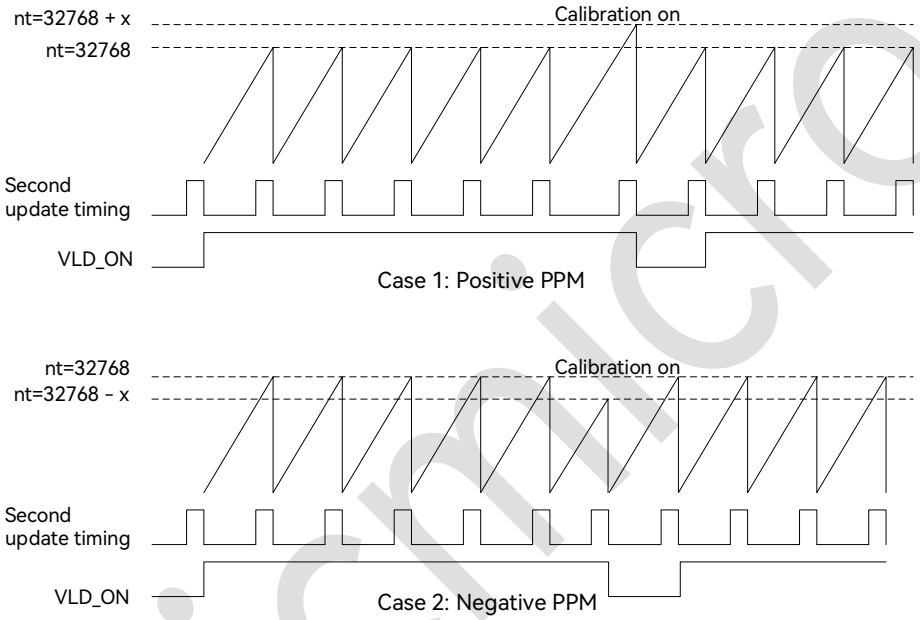


Figure 23-2: Oscillator Calibration

The 8-bit trim register (TRIM_VALUE[7:0], or TRIM[7:0] for short) in RTC_TRIM register stores a signed value that can adjust the maximum value of the low-speed clock counter. TRIM[7] is its symbol, where “0” represents a non-negative value and “1” represents a negative value. When TRIM[7] is “1”, TRIM[6:0] is considered a binary complement.

The TRIM_MODE register defines the frequency at which this alignment is performed.

- 0x0: once every 60 seconds (whenever SECOND = 00)
- 0x1: once every 30 seconds (whenever SECOND = 00 or 30)
- 0x2: once every 15 seconds (whenever SECOND = 00, 15, 30 or 45)
- 0x3: once every 6 seconds (whenever SECOND = 00, 06, 12, 18, 24, 30, 36, 42, 48 or 54)

This compensation scheme ensures that the minute refresh point is independent of the oscillator frequency accuracy.

Examples of oscillator calibration calculations are listed in Table 23-3 and Table 23-6.

TRIM_MODE = 0x0

Table 23-3 : Example of Oscillator Calibration Calculation with TRIM_MODE = 0x0

Binary TRIM[7:0]	Compensation Value	Total Clock Cycle in 60 Seconds	[ppm]
10000000	-128	32768 * 59 + 32640 * 1	-65.1
11000000	-64	32768 * 59 + 32704 * 1	-32.2
11111111	-1	32768 * 59 + 32767 * 1	-0.5
00000000	0	32768 * 60	0.0
00000001	1	32768 * 59 + 32769 * 1	0.5
00111111	63	32768 * 59 + 32831 * 1	32
01111111	127	32768 * 59 + 32895 * 1	64.6

TRIM_MODE = 0x1

Table 23-4 : Example of Oscillator Calibration Calculation with TRIM_MODE = 0x1

Binary TRIM[7:0]	Compensation Value	Total Clock Cycle in 30 Seconds	[ppm]
10000000	-128	32768 * 29 + 32640 * 1	-130.2
11000000	-64	32768 * 29 + 32704 * 1	-65.1
11111111	-1	32768 * 29 + 32767 * 1	-1
00000000	0	32768 * 30	0.0
00000001	1	32768 * 29 + 32769 * 1	1
00111111	63	32768 * 29 + 32831 * 1	64.6
01111111	127	32768 * 29 + 32895 * 1	129.2

TRIM_MODE = 0x2

Table 23-5 : Example of Oscillator Calibration Calculation with TRIM_MODE = 0x2

Binary TRIM[7:0]	Compensation Value	Total Clock Cycle in 15 Seconds	[ppm]
10000000	-128	32768 * 14 + 32640 * 1	-244.2
11000000	-64	32768 * 14+ 32704 * 1	-122
11111111	-1	32768 * 14 + 32767 * 1	-1.9
00000000	0	32768 * 15	0.0
00000001	1	32768 * 14 + 32769 * 1	1.9
00111111	63	32768 * 14 + 32831 * 1	120.2
01111111	127	32768 * 14 + 32895 * 1	242.2

TRIM_MODE = 0x3

Table 23-6 : Example of Oscillator Calibration Calculation with TRIM_MODE = 0x3

Binary TRIM[7:0]	Compensation Value	Total Clock Cycle in 6 Seconds	[ppm]
10000000	-128	32768 * 5 + 32640 * 1	-651
11000000	-64	32768 * 5 + 32704 * 1	-325.5
11111111	-1	32768 * 5 + 32767 * 1	-5.1
00000000	0	32768 * 6	0.0
00000001	1	32768 * 5 + 32769 * 1	5.1
00111111	63	32768 * 5 + 32831 * 1	320.4
01111111	127	32768 * 5 + 32895 * 1	646

For a given target ppm value, TRIM and TRIM_MODE shall be calculated correctly to make the actual ppm as close to the target as possible. It is also recommended that adjustments be performed more frequently at smaller steps by selecting a higher TRIM_MODE setting whenever possible.

The output signal VLD_ON_O is set to 1 for tunning the clock calibration. VLD_ON_O outputs a high level for 6, 15, 30 or 60 seconds according to the VLD_MODE setting, which facilitates timing measurements using an external reference clock.

23.4.2 Tamper Detector

The tamper input pin can be used to monitor specific locations in the system. A change in the state of the tamper input may indicate that a tamper event has occurred. The event to be monitored can be configured as either input (rising or falling or double) edge(s) detection or (high or low) level detection. In the case of level detection, the debounce circuit can filter out pulses shorter than 6 milliseconds (the duration is configurable).

When tampering is detected, the backup system behaves as follows:

- Store the timestamp (year, month, date, hour, minute and second) into one of three sets of registers. These registers always retain information about the last three tamper events, which are accessible by the ARM core in normal power consumption mode.

- The maximum number of tamper events detected is 63.
- If enabled, an interrupt request is sent to the CPU. In battery backup mode, the logic circuits are powered down and therefore any interrupt requests are ignored.

23.4.3 Backup Register

The backup register is organized with 32 bits as one word, with a total space of 20 words.

It can be accessed by ARM CPU through APB interface.

23.5 Register Description

RTC register base address: 0x40B0_0000

The registers are listed below:

Table 23-7: List of RTC Registers

Offset Address	Name	Description
0x00	RTC_TIME	RTC time register
0x04	RTC_DATE	RTC date register
0x08	RTC_ACCESS	RTC counter read-write control register
0x10	RTC_TRIM	RTC calibration control register
0x14	RTC_TEST	RTC incremental debugging function register
0x20	RTC_ALM1TIME	Alarm 1 time setting register
0x24	RTC_ALM1DATE	Alarm 1 date setting register
0x28	RTC_ALM1EN	Alarm 1 enable register
0x2C	RTC_ALM2SETTING	Alarm 2 setting register
0x30	RTC_TAMPCTRL	Tamper detector control and configuration register
0x34	RTC_TAMPCNT	Tamper count record register
0x38	RTC_TAMP3TIME	Third new tamper event timestamp register
0x3C	RTC_TAMP3DATE	Third new tamper event datestamp register
0x40	RTC_TAMP2TIME	Second new tamper event timestamp register
0x44	RTC_TAMP2DATE	Second new tamper event datestamp register
0x48	RTC_TAMP1TIME	Latest tamper event timestamp register
0x4C	RTC_TAMP1DATE	Latest tamper event datestamp register
0x50	RTC_INTEN	RTC interrupt enable register

Offset Address	Name	Description
0x54	RTC_INTRAW	RTC raw interrupt status register
0x58	RTC_INTSTA	RTC interrupt valid status register
0x5C	RTC_INTCLR	RTC interrupt status clear register
0x60	RTC_BKREG0	Backup register, 20 words or 80 bytes in total
0x64	RTC_BKREG1	
…	…	
0xAC	RTC_BKREG19	

Note: Since all registers of RTC are in the system low-speed (32 kHz) clock domain, reading or writing any register requires two low-speed clock cycles.

23.5.1 RTC Time Register (RTC_TIME)

Offset address: 0x00

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31	HOUR12_24	R/W	0x0	Setting of hour format: 0: 12-hour format 1: 24-hour format
30	RSV	-	-	Reserved
29	H20_PA	R/W	0x0	In the 12-hour format, H20_PA indicates AM or PM. 0: AM 1: PM In the 24-hour format, H20_PA represents the hour of 20 plus. For details, please refer to section 23.4.1.3, “Hour Format”.

Bit	Name	Attribute	Reset Value	Description
28: 24	HOUR19	R/W	0x0	The value of HOUR19 is encoded in BCD (binary-coded decimal) format. The hour format depends on HOUR12_24. In 12-hour format, when [H20_PA, HOUR19] counts from [1, 0x11] (11:00 p.m.) to [0, 0x12] (12:00 a.m.), the day counter will be incremented. In 24-hour format, when [H20_PA, HOUR19] counts from [1, 3] (23:00) to [0, 0] (00:00), the day counter will be incremented.
23	RSV	-	-	Reserved
22:16	MINUTE	R/W	0x0	The value of the minute counter is encoded in BCD format, within the range from 0x00 to 0x59. When it counts from 0x59 to 0x00, the HOUR counter will increment.
15	RSV	-	-	Reserved
14:8	SECOND	R/W	0x0	The value of the second counter is encoded in BCD format, within the range from 0x00 to 0x59. When it counts from 0x59 to 0x00, the MINUTE counter will increment.
7:0	CENTISEC	R	0x0	The value of the centisecond counter is encoded in BCD format, within the range from 0x00 to 0x99. When it counts from 0x99 to 0x00, the SECOND counter will increment.

Notes:

1. The RTC_ACCESS register shall be written before and after reading and writing this register.
2. Invalid time value written to RTC_TIME will be ignored. The validity of values written to the HOUR, MINUTE and SECOND registers is judged separately, and an invalid write to one register will not affect the writing of valid values to the other registers.

23.5.2 RTC Date Register (RTC_DATE)

Offset address: 0x04

Reset value: 0x8001 0106

Bit	Name	Attribute	Reset Value	Description
31	CENTURY	R/W	0x1	Century counter When it becomes 0, it will no longer increment with the year counter. 0: 2100 s to 2300 s (22nd to 24th century) 1: 2000 s (21st century)
30	RSV	-	-	Reserved
29:22	YEAR	R/W	0x0	This register represents the last two digits of the decimal year. This value is encoded in BCD format, within the range from 0x00 to 0x99. When the year counter counts from 0x99 to 0x00, the century counter will become 0. When CENTURY = 0, the years 04, 08, …, 92 and 96 are leap years, while the year 00 is a non-leap year. When CENTURY = 1, the years 00, 04, 08, …, 92 and 96 are leap years.
21	RSV	-	-	Reserved
20:16	MONTH	R/W	0x1	The value of the month counter is encoded in BCD format, within the range from 0x1 to 0x12. When it counts from 0x12 to 0x1, the year counter will increment.
15:14	RSV	-	-	Reserved
13:8	DAY	R/W	0x1	The value of the day counter is encoded in BCD format, within the range from: 0x1 to 0x31, in January, March, May, July, August, October and December. 0x1 to 0x30, in April, June, September and November. 0x1 to 0x29, in February of a leap year. 0x1 to 0x28, in February of a normal year. When the day counter counts from the end of the month to 0x1, the month counter will be incremented.

Bit	Name	Attribute	Reset Value	Description
7:3	RSV	-	-	Reserved
2:0	WEEK	R/W	0x6	The week counter ranges from 0 to 6. Wherein, 0 means Sunday, 1 means Monday, and so on. The relationship between the week counter and the date is user-defined.

Notes:

1. The RTC_ACCESS register shall be written before and after reading and writing this register.
2. Invalid date values (combination of date and month) written to RTC_DATE will be ignored.

23.5.3 RTC Counter Read-write Control Register (RTC_ACCESS)

Offset address: 0x08

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:4	RSV	-	0x0	Reserved
3	RTC_WRSTP	W	0x0	Writing 1 to this bit indicates that CPU has finished writing to the RTC_TIME and RTC_DATE registers. After this bit is written with 1, the CENTISEC counter will continue to count.
2	RTC_WRSTA	W	0x0	Writing 1 to this bit indicates that CPU has started writing to the RTC_TIME and RTC_DATE registers. After this bit is written with 1, the CENTISEC counter will retain its written value until 1 is written to RTC_WRSTP.
1	RTC_RDSTP	W	0x0	Writing 1 to this register indicates that CPU has finished reading from the RTC_TIME and RTC_DATE registers. After this register is written with 1, the shadow registers RTC_TIME and RTC_DATE will latch

Bit	Name	Attribute	Reset Value	Description
				the values of the time and date counters.
0	RTC_RDSTA	W	0x0	Writing 1 to this register indicates that CPU has started reading from the RTC_TIME and RTC_DATE registers. After this register is written with 1, the shadow registers RTC_TIME and RTC_DATE will continuously change with the time and date counters.

23.5.4 RTC Calibration Control Register (RTC_TRIM)

Offset address: 0x10

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:14	RSV	-	-	Reserved
13	VLD_EN	R/W	0x0	Validation enable: 0: validation disabled 1: validation enabled; the VLD_ON_O pulse is set periodically.
12:11	VLD_MODE	R/W	0x0	Setting of validation duration for VLD_ON_O: 0: VLD_ON_O remains high for 60 s. 1: VLD_ON_O remains high for 30 s. 2: VLD_ON_O remains high for 15 s. 3: VLD_ON_O remains high for 6 s.
10	TRIM_EN	R/W	0x0	Calibration enable: 0: calibration disabled 1: calibration enabled
9:8	TRIM_MODE	R/W	0x0	Calibration frequency setting: 0: once every 60 s 1: once every 30 s 2: once every 15 s 3: once every 6 s

Bit	Name	Attribute	Reset Value	Description
7:0	TRIM_VALUE	R/W	0x0	Signed compensation constant, indicating the compensation amount used to align the clock error.

23.5.5 RTC Counter Increment Function Test Register (RTC_TEST)

The increment function of RTC counter is used for testing or debugging.

Offset address: 0x14

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31	RTC_TEST_EN	R/W	0x0	Increment function enable: 0: normal operation, with increment function disabled 1: increment function enabled
30:5	RSV	-	-	Reserved
4	MON_CNT_UP	W	0x0	Month counter increment enable: 0: no action 1: both the month counter and the second counter are incremented.
3	DAY_CNT_UP	W	0x0	Day counter increment enable: 0: no action 1: both the day counter and the second counter are incremented.
2	HR_CNT_UP	W	0x0	Hour counter increment enable: 0: no action 1: both the hour counter and the second counter are incremented.
1	MIN_CNT_UP	W	0x0	Minute counter increment enable: 0: no action 1: both the minute counter and the second counter are incremented.
0	SEC_CNT_UP	W	0x0	Second counter increment enable: 0: no action 1: the second counter is incremented.

Notes:

1. The RTC_TEST_EN bit must be set to 1 before setting any other increment function bits. When the RTC_TEST_EN bit switches from 0 to 1, any simultaneous write operations to other increment function bits will be invalid.
2. Every time the increment function is enabled, the second counter will count up. If multiple increment bits are set in a single write operation, the second counter will still increment by 1.
3. If the time or date counters generate a carry during incrementing, it will cause a cascading increment of the higher-level counters.
4. If the day counter is incremented, the week counter will also be incremented. However, if the month counter is incremented and the day counter is not incremented, the week counter will remain unchanged.

23.5.6 Alarm 1 Time Setting Register (RTC_ALM1TIME)

Offset address: 0x20

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31	ALM1_HOUR12_24	R/W	0x0	Hour format setting for alarm 1: 0: 12-hour format 1: 24-hour format
30	RSV	-	-	Reserved
29	ALM1_H20_PA	R/W	0x0	Hour setting for alarm 1: The hour format follows the setting of ALM1_HOUR12_24.
28:24	ALM1_HOUR19	R/W	0x0	
23	RSV	-	-	Reserved
22:16	ALM1_MINUTE	R/W	0x0	Minute setting for alarm 1
15	RSV	-	-	Reserved
14:8	ALM1_SECOND	R/W	0x0	Second setting for alarm 1
7:0	ALM1_CENTISEC	R/W	0x0	Centisecond setting for alarm 1

23.5.7 Alarm 1 Date Setting Register (RTC_ALM1DATE)

Offset address: 0x24

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:30	RSV	-	-	Reserved
29:22	ALM1_YEAR	R/W	0x0	Year setting for alarm 1
21	RSV	-	-	Reserved
20:16	ALM1_MONTH	R/W	0x0	Month setting for alarm 1
15:14	RSV	-	-	Reserved
13:8	ALM1_DAY	R/W	0x0	Date setting for alarm 1
7:3	RSV	-	-	Reserved
2:0	ALM1_WEEK	R/W	0x0	Week setting for alarm 1

23.5.8 Alarm 1 Setting Enable Register (RTC_ALM1EN)

Offset address: 0x28

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31	ALM1_EN	R/W	0x0	Alarm 1 enable: 0: alarm 1 disabled 1: alarm 1 enabled
30:8	RSV	-	-	Reserved
7	ALM1_EN_YEAR	R/W	0x0	Year condition control bit for alarm 1: 0: condition not applied 1: condition applied
6	ALM1_EN_MON	R/W	0x0	Month condition control bit for alarm 1: 0: condition not applied 1: condition applied
5	ALM1_EN_DAY	R/W	0x0	Date condition control bit for alarm 1: 0: condition not applied 1: condition applied
4	ALM1_EN_WEEK	R/W	0x0	Week condition control bit for alarm 1: 0: condition not applied 1: condition applied

Bit	Name	Attribute	Reset Value	Description
3	ALM1_EN_HOUR	R/W	0x0	Hour condition control bit for alarm 1: 0: condition not applied 1: condition applied
2	ALM1_EN_MIN	R/W	0x0	Minute condition control bit for alarm 1: 0: condition not applied 1: condition applied
1	ALM1_EN_SEC	R/W	0x0	Second condition control bit for alarm 1: 0: condition not applied 1: condition applied
0	ALM1_EN_CS	R/W	0x0	Centisecond condition control bit for alarm 1: 0: condition not applied 1: condition applied

23.5.9 Alarm 2 Setting Register (RTC_ALM2SETTING)

Offset address: 0x2C

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31	ALM2_EN	R/W	0x0	Alarm 2 enable: 0: alarm 2 disabled 1: alarm 2 enabled
30:4	RSV	-	-	Reserved
3:0	ALM2_SETTING	R/W	0x0	Alarm 2 interrupt output cycle: 0: no output 1: 1 s 2: 1/2 s 3: 1/4 s 4: 1/8 s 5: 1/16 s 6: 1/32 s 7: 1/64 s 8: 1/128 s 9: 1 min Others: 1 s

23.5.10 Tamper Detector Control and Configuration Register (RTC_TAMPCTRL)

Offset address: 0x30

Reset value: 0x0000 0030

Bit	Name	Attribute	Reset Value	Description
31:7	RSV	-	-	Reserved
6	TAMP_CNT_CLR	W	0x0	Clear bit of the tamper counter Write 1 to this bit to clear the tamper counter TAMP_CNT.
5:4	TAMP_DBNC	R/W	0x3	Debounce time for TAMPER_IN: 0: no debounce 1: 2 ms 2: 4 ms 3: 6 ms
3:1	TAMP_EDGE	R/W	0x0	Events to be detected on the TAMPER_IN pin: 0: rising edge 1: falling edge 2: both edges 3: high level lasting longer than the set value of TAMP_DBNC 4: low level lasting longer than the set value of TAMP_DBNC Others: tamper detector disabled
0	TAMP_EN	R/W	0x0	Tamper detector enable bit for the TAMPER_IN pin: 0: tamper detector disabled 1: tamper detector enabled

23.5.11 Tamper Count Record Register (RTC_TAMPCNT)

Offset address: 0x34

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:6	RSV	-	-	Reserved

Bit	Name	Attribute	Reset Value	Description
5:0	TAMP_CNT	R	0x0	Tamper event counter: 0x00: no tamper event 0x01–0x3E: number of tamper events 0x3F: 63 or more tamper events

23.5.12 Third New Tamper Event Timestamp Register (RTC_TAMP3TIME)

Offset address: 0x38

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:30	RSV	-	-	Reserved
29	TAMP3_H20_PA	R	0x0	Hour of the third new tamper event The hour format follows the setting of HOUR12_24.
28:24	TAMP3_HOUR19	R	0x0	
23	RSV	-	-	Reserved
22:16	TAMP3_MINUTE	R	0x0	Minute of the third new tamper event
15	RSV	-	-	Reserved
14:8	TAMP3_SECOND	R	0x0	Second of the third new tamper event
7:0	RSV	-	-	Reserved

23.5.13 Third New Tamper Event Datestamp Register (RTC_TAMP3DATE)

Offset address: 0x3C

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:30	RSV	-	-	Reserved
29:22	TAMP3_YEAR	R	0x0	Year of the third new tamper event
21	RSV	-	-	Reserved
20:16	TAMP3_MONTH	R	0x0	Month of the third new tamper event
15:14	RSV	-	-	Reserved
13:8	TAMP3_DAY	R	0x0	Date of the third new tamper event
7:0	RSV	-	-	Reserved

23.5.14 Second New Tamper Event Timestamp Register (RTC_TAMP2TIME)

Offset address: 0x40

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:30	RSV	-	-	Reserved
29	TAMP2_H20_PA	R	0x0	Hour of the second new tamper event The hour format follows the setting of HOUR12_24.
28:24	TAMP2_HOUR19	R	0x0	
23	RSV	-	-	Reserved
22:16	TAMP2_MINUTE	R	0x0	Minute of the second new tamper event
15	RSV	-	-	Reserved
14:8	TAMP2_SECOND	R	0x0	Second of the second new tamper event
7:0	RSV	-	-	Reserved

23.5.15 Second New Tamper Event Datestamp Register (RTC_TAMP2DATE)

Offset address: 0x44

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:30	RSV	-	-	Reserved
29:22	TAMP2_YEAR	R	0x0	Year of the second new tamper event
21	RSV	-	-	Reserved
20:16	TAMP2_MONTH	R	0x0	Month of the second new tamper event
15:14	RSV	-	-	Reserved
13:8	TAMP2_DAY	R	0x0	Date of the second new tamper event
7:0	RSV	-	-	Reserved

23.5.16 Latest Tamper Event Timestamp Register (RTC_TAMP1TIME)

Offset address: 0x48

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:30	RSV	-	-	Reserved
29	TAMP1_H20_PA	R	0x0	Hour of the latest tamper event The hour format follows the setting of HOUR12_24.
28:24	TAMP1_HOUR19	R	0x0	
23	RSV	-	-	Reserved
22:16	TAMP1_MINUTE	R	0x0	Minute of the latest tamper event
15	RSV	-	-	Reserved
14:8	TAMP1_SECOND	R	0x0	Second of the latest tamper event
7:0	RSV	-	-	Reserved

23.5.17 Latest Tamper Event Datestamp Register (RTC_TAMP1DATE)

Offset address: 0x4C

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:30	RSV	-	-	Reserved
29:22	TAMP1_YEAR	R	0x0	Year of the latest tamper event
21	RSV	-	-	Reserved
20:16	TAMP1_MONTH	R	0x0	Month of the latest tamper event
15:14	RSV	-	-	Reserved
13:8	TAMP1_DAY	R	0x0	Date of the latest tamper event
7:0	RSV	-	-	Reserved

23.5.18 RTC Interrupt Enable Register (RTC_INTEN)

Offset address: 0x50

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:3	RSV	-	-	Reserved
2	TAMP_INT_EN	R/W	0x0	Tamper interrupt enable: 0: tamper interrupt disabled 1: tamper interrupt enabled
1	ALM2_INT_EN	R/W	0x0	Alarm 2 interrupt enable: 0: alarm 2 interrupt disabled 1: alarm 2 interrupt enabled
0	ALM1_INT_EN	R/W	0x0	Alarm 1 interrupt enable: 0: alarm 1 interrupt disabled 1: alarm 1 interrupt enabled

23.5.19 RTC Raw Interrupt Status Register (RTC_INTRAW)

Offset address: 0x54

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:3	RSV	-	-	Reserved
2	TAMP_INT_RAW	R	0x0	Tamper raw interrupt status: 0: not triggered 1: triggered
1	ALM2_INT_RAW	R	0x0	Alarm 2 raw interrupt status: 0: not triggered 1: triggered
0	ALM1_INT_RAW	R	0x0	Alarm 1 raw interrupt status: 0: not triggered 1: triggered

23.5.20 Interrupt Status Register after RTC Enable (RTC_INTSTA)

Offset address: 0x58

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:3	RSV	-	-	Reserved
2	TAMP_INT_STA	R	0x0	Interrupt status after tamper is enabled: 0: not triggered 1: triggered
1	ALM2_INT_STA	R	0x0	Interrupt status after alarm 2 is enabled: 0: not triggered 1: triggered
0	ALM1_INT_STA	R	0x0	Interrupt status after alarm 1 is enabled: 0: not triggered 1: triggered

23.5.21 RTC Interrupt Status Clear Register (RTC_INTCLR)

Offset address: 0x5C

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:3	RSV	-	-	Reserved
2	TAMP_INT_CLR	W	0x0	Tamper interrupt status clear: 0: no action 1: tamper interrupt status cleared
1	ALM2_INT_CLR	W	0x0	Alarm 2 interrupt status clear: 0: no action 1: alarm 2 interrupt status cleared
0	ALM1_INT_CLR	W	0x0	Alarm 1 interrupt status clear: 0: no action 1: alarm 1 interrupt status cleared

23.5.22 Backup Register (RTC_BKREG_n)

Backup register: 20 words (i.e., 80 bytes) in total

Offset address: 0x60 + 0x4 × n, wherein n = 0, 1, ..., 19

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:0	BKREG_n	R/W	0x0	Backup register n (n = 0, 1, ..., 19)

23.6 Operation Procedure

23.6.1 RTC Clock Enabling

1. Write 0xABCD to the PMU_CPR register to enable its write operation.
2. Configure PMU_FCCR[6] to enable the RTC controller clock.
3. Configure PMU_FRCR[6] to release the RTC controller reset.
4. Configure PMU_SASR[0] to set the external VBAT power supply (that is, select the external VBAT power supply).
5. Write 0x459E to the PMU_CPR register to end its write operation.

23.6.2 RTC Time Reading

1. Enable the RTC clock.
2. Start initializing the RTC date and time by writing 0x4 to the RTC_ACCESS register to enable RTC write access.
3. Write the date and time to the RTC_DATE and RTC_TIME registers.
4. End the initialization of the RTC date and time by writing 0x8 to the RTC_ACCESS register to finish RTC writing.
5. Write 0x1 to the RTC_ACCESS register to update the date and time.
6. Write 0x2 to the RTC_ACCESS register to save the current date and time.
7. Read the RTC_DATE and RTC_TIME registers to obtain the date and time.

23.6.3 RTC Alarm 1 Setting

1. Enable the RTC clock.
2. Start initializing the RTC date and time by writing 0x4 to the RTC_ACCESS register to enable RTC write access.
3. Write the date and time to the RTC_DATE and RTC_TIME registers.
4. End the initialization of the RTC date and time by writing 0x8 to the RTC_ACCESS register to finish RTC writing.
5. Set the alarm 1 timing by writing the date and time to the RTC_ALM1DATE and RTC_ALM1TIME registers.
6. Initialize the alarm 1 interrupt by configuring RTC_INTEN[0] to enable the alarm 1 interrupt.
7. Enable the corresponding condition application bit in the RTC_ALM1EN register by configuring RTC_ALM1EN[31] to enable alarm 1.

23.6.4 RTC Alarm 2 Setting

1. Enable the RTC clock.
2. Start initializing the RTC date and time by writing 0x4 to the RTC_ACCESS register to enable RTC write access.
3. Write the date and time to the RTC_DATE and RTC_TIME registers.
4. End the initialization of the RTC date and time by writing 0x8 to the RTC_ACCESS register to finish RTC writing.
5. Configure RTC_ALM2SETTING[3:0] to set the interrupt output period for alarm 2.
6. Initialize the alarm 2 interrupt by configuring RTC_INTEN[1] to enable the alarm 2 interrupt.
7. Configure RTC_ALM2SETTING[31] to enable alarm 2.

23.6.5 RTC Backup Register

1. Write 0xABCD to the PMU_CPR register to enable its write operation.
2. Configure PMU_SASR[0] to set the external VBAT power supply (that is, select the external VBAT power supply).
3. Write 0x459E to the PMU_CPR register to end its write operation.
4. Write data to the RTC_BKREGn registers (one backup register stores 4 bytes, totaling 80 bytes).
5. With the external VBAT power supply connected, VDDH is powered off first and then re-powered on to read the backup register again, which is consistent with that before power off.

23.6.6 RTC Tamper Detection

1. Configure GPIOC13 pin as an input.
2. Write 0xABCD to the PMU_CPR register to enable its write operation.
3. Configure PMU_PDWKCR[3] to set the RTC_TAMPER wakeup event to active.
4. Write 0x459E to the PMU_CPR register to end its write operation.
5. Enable the RTC clock.
6. Start initializing the RTC date and time by writing 0x4 to the RTC_ACCESS register to enable RTC write access.
7. Write the date and time to the RTC_DATE and RTC_TIME registers.
8. End the initialization of the RTC date and time by writing 0x8 to the RTC_ACCESS register to finish RTC writing.
9. Configure RTC_TAMPCTRL[3:1] to set the levels of the TAMPER_IN pin to be monitored.
10. Set RTC_TAMPCTRL[6] to 1 to clear the tamper counter value.
11. Set RTC_INTEN[2] to 1 to enable the tamper interrupt.
12. Set RTC_TAMPCTRL[0] to 1 to enable the tamper detection.

23.6.7 RTC Tamper Counter Value Clearing

Set RTC_TAMP_CTRL[6] to 1 to clear the tamper counter value.

23.6.8 RTC Calibration

1. Enable the RTC clock.
2. Start initializing the RTC date and time by writing 0x4 to the RTC_ACCESS register to enable RTC write access.
3. Write the date and time to the RTC_DATE and RTC_TIME registers.
4. End the initialization of the RTC date and time by writing 0x8 to the RTC_ACCESS register to finish RTC writing.
5. Configure RTC_TRIM[9:8] and RTC_TRIM[7:0] to set the calibration frequency and compensation constant, and set RTC_TRIM[10] to 1 to enable the calibration function.
6. Configure PC5 to alternate function 0, which can output a parity level.

23.6.9 RTC Increment Test

1. Enable the RTC clock.
2. Clear the RTC_TEST register and configure it to increment the corresponding counter.
3. Write 0x1 to the RTC_ACCESS register to update the date and time.
4. Write 0x2 to the RTC_ACCESS register to save the current date and time.
5. Read the RTC_DATE and RTC_TIME registers to obtain the date and time.
6. Repeat steps 2 to 5 to observe the corresponding counter increment.

24 CANFD Bus Controller (CANFD)

24.1 Overview

The CAN (Controller Area Network) controller can be used in the fields of automotive electronics and industrial control, supporting the CAN FD protocol and is downward compatible with the CAN 2.0A/B protocols.

24.2 Main Features

- Compliant with ISO 11898-1:2015 specification
- Support CAN and CAN FD formats
- Data frames of up to 64 bytes
- Flexible data rates
- Data rates up to 8 Mbps
- Hardware message filtering (single/dual filters)
- Transmission and reception via DMA supported
- 3 transmit buffers
- 256-byte RX FIFO and 256-byte TX FIFO
- Overload frame generated at FIFO overflow
- Protocol exception event detection
- Normal mode and listening mode
- Up to 32 configurable receive filters
- Transmitter delay compensation, with the length of up to three data bits
- Single transmission
- Transmission can be aborted
- Readable error counter

24.3 Pin Description

Table 24-1: CAN Pin Description

Function Pin	Alternate Function Pin	Direction	Functional Description
CAN0_TX	PA3, PA12, PB7, PB9, PB13, PC6, PC8, PC10, PC12	Output	Transmitting data
CAN0_RX	PA2, PA4, PA5, PA11, PB6, PB8, PB12, PC7, PC9, PC11, PD2	Input	Receiving data

24.4 Functional Description

24.4.1 Standard Frame Memory Buffer Layout

Messages to be transmitted/received are stored in the transmit/receive buffer, where the transmit buffer can hold one message at a time, while the receive FIFO buffer can hold multiple messages.

The memory address cannot be accessed directly and can only be obtained indirectly through the TXBUFn and RXBUFn registers. The layout of the transmit/receive buffers for a standard frame is as follows:

offset	31 30 29 28 27 26 25 24	23 22 21	20	19	18 17 16	15 14 13 12 11 10 9 8	7	6	5	4	3 2 1 0
0x000	DATA 1 (DLC<1 then free)	ID[2:0]	RSVD	ESI	RSVD	ID[10:3]	IDE	RRS/RTR	FDF	BRS	DLC
0x001	DATA 5 (DLC<5 then free)	DATA 4 (DLC<4 then free)				DATA 3 (DLC<3 then free)		DATA 2 (DLC<2 then free)			
0x002	DATA 9 (DLC<9 then free)	DATA 8 (DLC<8 then free)				DATA 7 (DLC<7 then free)		DATA 6 (DLC<6 then free)			
0x003	DATA 13 (DLC<10 then free)	DATA 12 (DLC<9 then free)				DATA 11 (DLC<9 then free)		DATA 10 (DLC<9 then free)			
0x004	DATA 17 (DLC<11 then free)	DATA 16 (DLC<10 then free)				DATA 15 (DLC<10 then free)		DATA 14 (DLC<10 then free)			
0x005	DATA 21 (DLC<11 then free)	DATA 20 (DLC<11 then free)				DATA 19 (DLC<11 then free)		DATA18 (DLC<11 then free)			
0x006	DATA 25 (DLC<12 then free)	DATA 24 (DLC<11 then free)				DATA 23 (DLC<11 then free)		DATA 22 (DLC<11 then free)			
...	...	...				...		...			
0x00F	DATA 61 (DLC<15 then free)	DATA 60 (DLC<15 then free)				DATA 59 (DLC<15 then free)		DATA 58 (DLC<15 then free)			
0x010	FIDX	DATA 64 (DLC<15 then free)				DATA 63 (DLC<15 then free)		DATA 62 (DLC<15 then free)			

IDE (identifier extension bit), which specifies whether it is a standard frame or an extended frame:

0: standard frame

1: extended frame

RTR (remote transmission request bit) / RRS (remote request substitute bit), which specify whether it is a standard CAN data frame or a remote frame. For CAN FD format, this bit indicates the bus status at the RRS position:

0: data frame

1: remote frame

Note: There is no remote frame in CAN FD, where the explicit status is sent at the RRS position.

FDF (FD frame format):

0: classic CAN mode

1: CAN FD format

BRS (bit rate switch, valid only in CAN FD mode):

0: frame without bit rate switching

1: frame with bit rate switching

ESI (error status indicator, valid only in CAN FD mode):

0: active error node

1: passive error node

DLC: data length code

ID: CAN identifier

DATA (1...64): data bytes (7-MSB, 0-LSB)

Note: The number of data bytes transmitted is defined by the DLC field; for the transmission in classic CAN frame format, if DLC > 8, only 8 bytes of data are transmitted; if RTR is set, no data bytes are transmitted regardless of the DLC field.

FIDX[5:0]: The index of the matching received filter, which is located after the last data byte.

RSVD: reserved

24.4.2 Extended Frame Memory Buffer Layout

offset	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x000	ID[12:5]								ID[20:13]								ID[28:21]								IDE	SRR/RTR	FDF	BRS	DLC			
0x001	DATA 3 (DLC<3 then free)								DATA 2 (DLC<2 then free)								DATA 1 (DLC<1 then free)								ID[4:0]					RRS/RTR	ESI	RSVD
0x002	DATA 7 (DLC<9 then free)								DATA 6 (DLC<8 then free)								DATA 5 (DLC<7 then free)								DATA 4 (DLC<6 then free)							
0x003	DATA 11 (DLC<10 then free)								DATA 10 (DLC<9 then free)								DATA 9 (DLC<9 then free)								DATA 8 (DLC<9 then free)							
0x004	DATA 15 (DLC<11 then free)								DATA 14 (DLC<10 then free)								DATA 13 (DLC<10 then free)								DATA 12 (DLC<10 then free)							
0x005	DATA 19 (DLC<11 then free)								DATA 18 (DLC<11 then free)								DATA 17 (DLC<11 then free)								DATA16 (DLC<11 then free)							
...	...								...								...								...							
0x00F	DATA 59 (DLC<15 then free)								DATA 58 (DLC<15 then free)								DATA 57 (DLC<15 then free)								DATA 56 (DLC<15 then free)							
0x010	DATA 64 (DLC<15 then free)								DATA 62 (DLC<15 then free)								DATA 61 (DLC<15 then free)								DATA 60 (DLC<15 then free)							
0x011	free								free								FIDX								DATA 64 (DLC<15 then free)							

IDE (identifier extension bit), which specifies whether it is a standard frame or an extended frame:

0: standard frame

1: extended frame

RRS (remote request substitute bit) /RTR, which transmits an implicit level in the extended format.

RTR (remote transmission request bit) / RRS (remote request substitute bit), which specify whether it is a standard CAN data frame or a remote frame. For CAN FD format, this bit indicates the bus status at the RRS position:

0: data frame

1: remote frame

Note: There is no remote frame in CAN FD, where the explicit status is sent at the RRS position.

FDF (FD frame format):

0: classic CAN mode

1: CAN FD format

BRS (bit rate switch, valid only in CAN FD mode):

0: frame without bit rate switching

1: frame with bit rate switching

ESI (error status indicator, valid only in CAN FD mode):

0: active error node

1: passive error node

DLC: data length code

ID: CAN identifier

DATA (1...64): data bytes (7-MSB, 0-LSB)

Note: The number of data bytes transmitted is defined by the DLC field; for the transmission in classic CAN frame format, if DLC > 8, only 8 bytes of data are transmitted; if RTR is set, no data bytes are transmitted regardless of the DLC field.

FIDX[5:0]: The index of the matching received filter, which is located after the last data byte.

RSVD: reserved

24.4.3 TX Buffer Handling

The DCAN has three TX buffers, each with a size of 72 bytes (18 words of 32 bits), allowing each message specified by classic CAN and CAN FD to be stored with an effective payload of up to 64 bytes. The format of the message to be transmitted can be individually configured for each TX buffer. The table below describes the frame configuration:

FDCON		TX Buffer Layout		Frame Format
FDEN	BRSEN	FDF	BRS	
0	-	-	-	Classic CAN, with an effective payload of 8 bytes
1	0	0	-	Classic CAN, with an effective payload of 8 bytes
1	0	1	-	CAN FD without bit rate switching, with an effective payload of 64 bytes
1	1	0	-	Classic CAN, with an effective payload of 8 bytes
1	1	1	0	CAN FD without bit rate switching, with an effective payload of 64 bytes
1	1	1	1	CAN FD with bit rate switching, with an effective payload of 64 bytes

TX buffers have hardware priority, with TXB0 having the highest priority and being sent first, while TXB2 the lowest priority and sent last. When DCAN sends a message from one TX buffer, access to the other buffers is allowed to check which TX buffer is locked or released, as specified by the TBS bit in the SR register of the TX buffer, using the TXBSEL register.

24.4.4 Acceptance Filtering

The CAN module can have up to 32 receive filters, each with its own set of acceptance mask registers (AMR), acceptance code registers (ACR), and a mode selector (AFM). Access to the ACR, AMR, and AFM for a specified filter is accomplished indirectly through the ACFSEL.SEL register.

Only when the identifier bit of the received message is equal to the predefined bit in the acceptance code register can the RX filter in CAN controller pass the received message to RX FIFO. The acceptance filter consists of the acceptance code registers (ACR3:ACR0) and the acceptance mask registers (AMR3:AMR0). The AFM bit of the mode register (CAN_MR) can be set for single/dual filters. In the configuration of single filter, the filter is 4 bytes long. If the received data is in standard frame mode, the first two bytes, including the arbitration bit, RTR bit and data bit (the data byte is not mandatory), can be received. All individual bit comparison must be signaled to indicate successful reception of the data; if the received data is in extended frame format, the arbitration bit and RTR bit will be received. For bits not defined in the format, the filter will not perform comparisons.

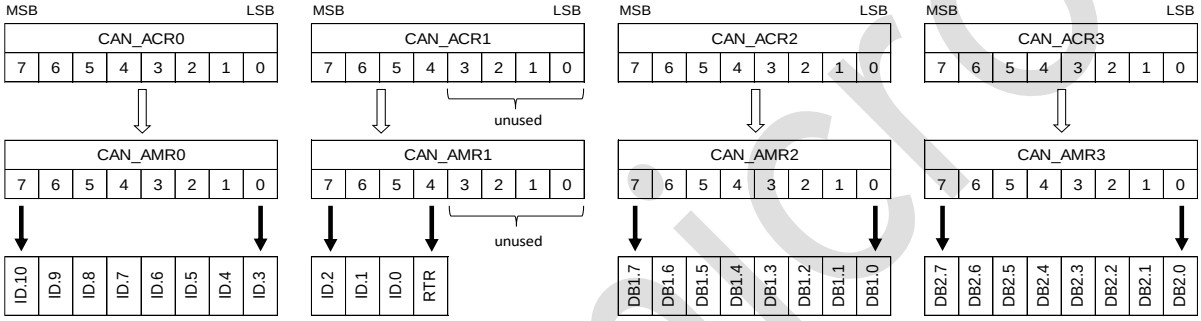
In dual-filter configuration, two shorter filters are defined. The received data will be compared against both filters to decide whether the data shall be stored in the RX FIFO. If at least one acceptance filter comparison is successful, the received data will be stored in FIFO. For standard format frames, the first filter will compare the arbitration, the RTR bit and the first data byte. The second filter will only compare the arbitration and RTR bits. If the first filter

does not filter the data byte, the lower four bits of AMR1 and AMR3 shall be set to 1 (do not compare this bit).

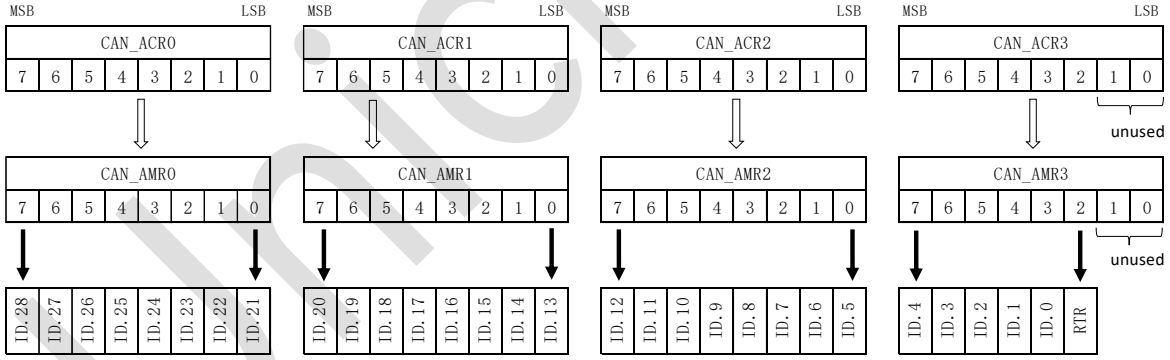
The bit formats corresponding to different filter settings and different arbitration lengths (11 bits for standard frame / 29 bits for extended frame) are shown below:

Single filter:

1. Standard frame

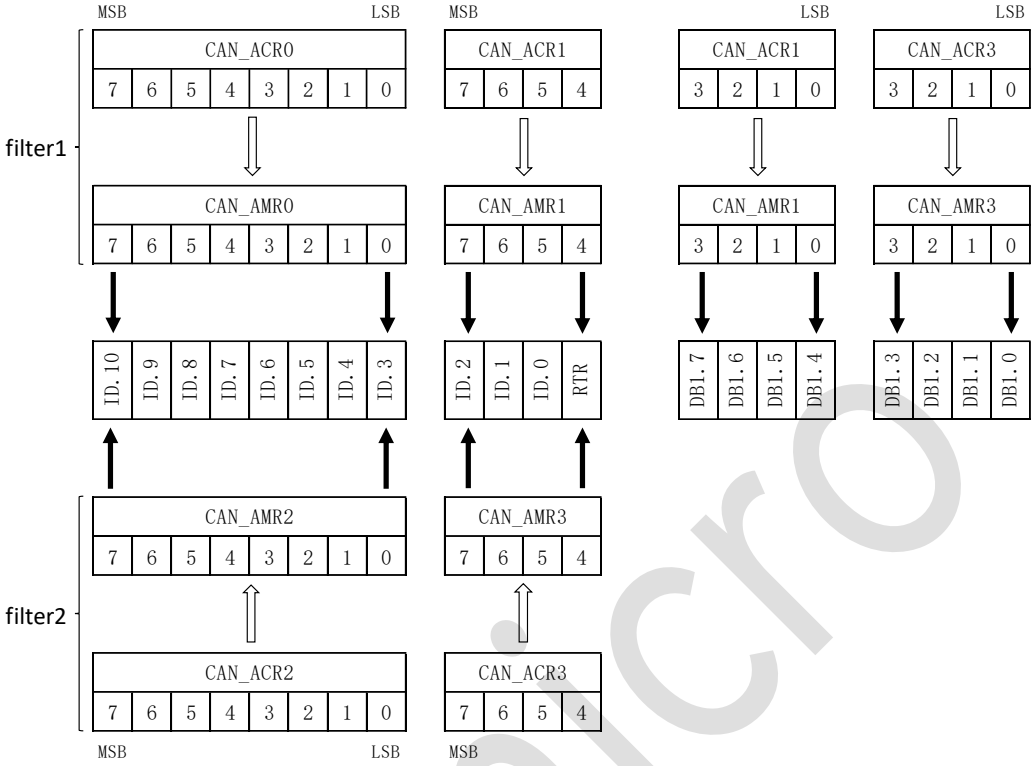


2. Extended frame

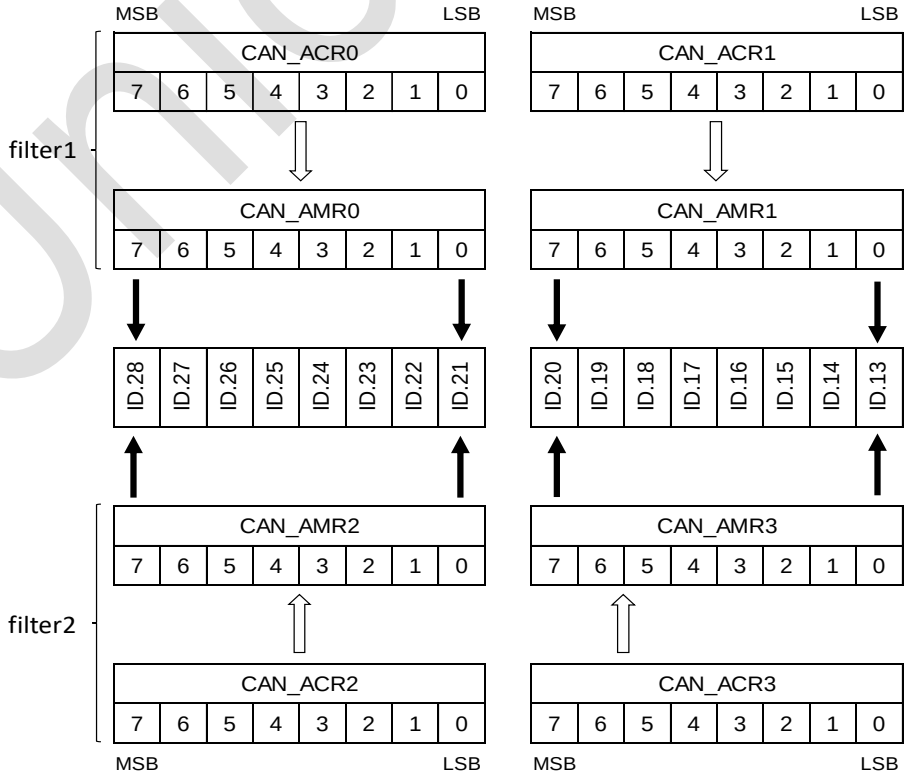


Dual-filter:

In standard mode, upon receiving data, the system will compare the first received data byte with the first filter ID including the RTR bit, or with the second filter ID also including the RTR bit.

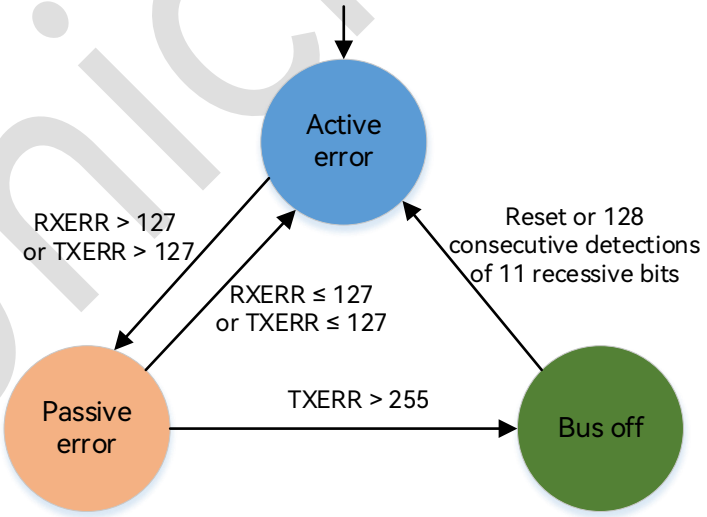


If an extended frame is received in dual-filter mode, both filters will only compare the first two bytes of the extended identifier range.



24.4.5 Error Counter Analysis

Based on the values of the error counters, the DCAN controller can operate in three possible error states: active error, passive error, or bus off. If both error counters are between 0 and 127, the CAN controller is in an active error state. In this state, an active error flag (6 dominant bits) is generated when the error condition is met. If either error counter exceeds 127 (ranging from 128 to 255), the DCAN changes to a passive error state. In this case, a passive error flag (6 recessive bits) is generated upon detecting an error condition. If the transmission error counter exceeds 255, the bus off state is reached. In this state, the reset request bit is automatically set, and the DCAN cannot affect the bus. When in bus off mode, the DCAN can only recover from this state through a “Reset Request = 0” command from the host controller. This will initiate the bus off recovery procedure, during which the transmission error counter counts 128 bus idle signals. At the end of this period, if both error counters are at 0, the device will revert to an active error state.



24.4.6 Data Overflow

Data overflow occurs when the RX FIFO is full and new incoming messages are being received. It signals the host CPU by setting the data overflow status flag in the status register. Data

overflow results in a data overflow interrupt request. When the DCAN controller runs into a data overflow state, it means that the host controller is severely overloaded as it does not have enough time to read all received messages from the RX buffer in time. The data overflow signal indicates a loss of data, which may lead to inconsistencies in the system. In general, the system should be designed to ensure that incoming messages are transmitted and processed fast enough to avoid data overflow. If data overflow cannot be avoided, an exception handler specific to the application should be implemented in the host controller.

24.4.7 Arbitration Lost Capture

The DCAN controller can identify the exact position of the CAN bit stream where arbitration loss occurs and immediately generate an “arbitration loss interrupt.” Additionally, the bit number is captured in the arbitration loss capture register (CAN_ALC). Once the host controller reads the contents of this register, the capture function for the next arbitration loss is activated. This feature allows the DCAN to monitor each CAN bus access. All instances of unsuccessful arbitration can be identified for diagnostics or during system configuration.

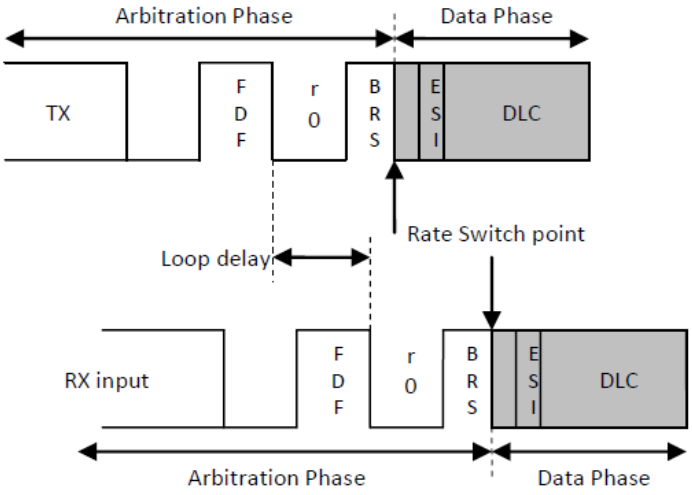
24.4.8 Transceiver Delay Compensation

When the BRS bit is set in the message, the CAN FD protocol allows frames to be transmitted and received at a higher bit rate during the data phase than during the arbitration phase. During frame transmission, the transmitter compares the transmitted bits with the currently received bits. When DCAN is switched to the data bit time, it can become shorter than the loop delay of the transceiver. This limitation can affect the correct comparison of transmitted and received bits.

DCAN supports the optional transceiver delay compensation (TDC) mechanism, which allows shorter data bit time than the transmitter delay. It defines a secondary sample point (SSP) where the transmitted bits are correctly compared with the received bits. SSP is the sum of the

measurement delay from TXD output to RXD input and the transmitter delay compensation offset (TDCO). TDCO is used to adjust the position of SSP from the bit edge to the mid-position. Note that the value of TDCO cannot be greater than the bit duration of the data phase. The actual transmitter compensation value (SSPP) is set in the CAN_SSPP register, whose contents are cleared in reset mode and updated each time a CAN FD frame is transmitted with transceiver delay compensation enabled.

Enable the transceiver delay compensation (TDC) by setting the TDCE bit in the CAN_TDCR register. Within each CAN FD frame, when arbitration is determined and before the bit rate is switched, the transmitter starts measuring from the edge of the FDF to r0. This delay is measured during the DCAN system clock cycle by a counter that starts at the falling edge between FD and r0 on TXD and stops when this falling edge is seen on RXD. A configurable offset TDCO is added to the measured delay value to place the SSP position in the middle of the bit time.



24.4.9 Restricted Operation Mode

In restricted operation mode, the DCAN can send and receive frames and give acknowledge to valid frames, but it does not send active error frames or overload frames. In case of an error condition or overload condition, the DCAN does not send a dominant bit, but instead waits for

a bus idle condition to resynchronize itself to the CAN communication. When an error is detected, the error counter does not increment. The restricted operation mode can be enabled by setting the REOM bit in the FDCON register.

24.4.10 Disabled Automatic Retransmission Mode

When the DAR bit in the FDCON register is set, the DCAN will not retransmit frames that have lost arbitration or have been disturbed by errors during transmission. When an error is detected during frame transmission, the error cause will be reported, and an active or passive flag will be sent. After that, the DCAN enters the idle state.

24.4.11 Sleep Mode

CAN enters sleep mode when the following conditions are met:

- CAN is in normal operation.
- There are no pending transmission requests.
- There are no pending interrupts.
- The wake-up prescaler (CAN_WUPDIV) has a non-zero value.
- The CAN bus is in an idle state.

When the SLEEP bit in the MR register is set to 1, the CAN goes from normal mode (RM=0) to sleep mode. If the SLEEP bit is set during CAN bus activation or while transmitting a frame, the DCAN will wait until the bus is idle before entering sleep mode. The SLEEP bit in the MR register indicates the true state of the core.

The DCAN enters normal mode (wakes up) under the following conditions:

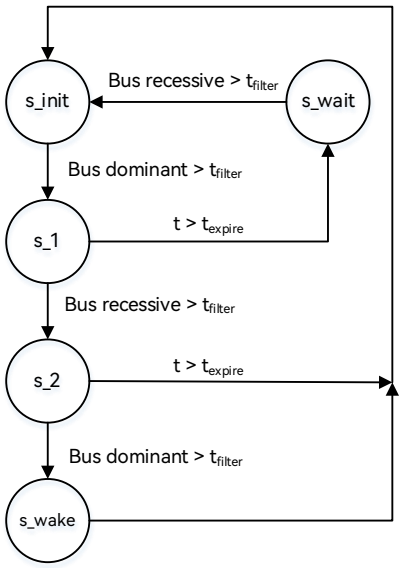
- Hardware reset
- SLEEP bit in MR register cleared

- Wake-up mode detected

24.4.12 Wake-up Unit

When DCAN is in sleep mode, the wake-up unit is enabled, and the module monitors the RXD input to detect the correct wake-up method. The wake-up module consists of two programmable counters (a 16-bit CAN_WUPFT counter and a 20-bit CAN_WUPET counter), a slope detector and a state machine. To ensure a wide range of measurement intervals, the counter triggers the prescaler (CAN_WUPPRE) by an 8-bit clock. The CAN_WUPFT counter is used to specify the minimum duration of stable states (dominant or recessive) on the CAN bus, while the CAN_WUPET counter is used to determine the maximum detection time for the wake-up mode.

Once the DCAN enters sleep mode, the wake-up detection unit will enter the s_init state. The module will remain in this state as long as the bus is recessive and the dominant level is shorter than t_{filter}. When the dominant level is longer than t_{filter}, the module jumps to s_1 state where it waits for the recessive level. If the recessive level is shorter than t_{filter}, it will be ignored. If the wake-up module detects that the recessive level is longer than t_{filter}, it will jump to s_2 state and wait for the dominant level to be longer than t_{filter}. If the dominant level is long enough, the wake-up mode is detected, DCAN is woken up, and no message can be received or sent until 11 consecutive bits (no bus sequence) are detected.



24.4.13 Listening Mode

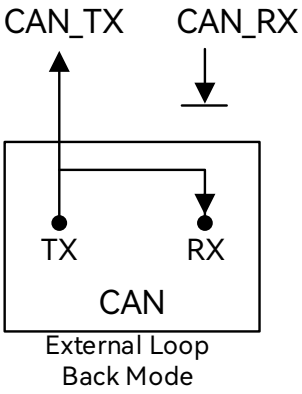
In listening mode, even if a message is successfully received, the CAN controller does not issue an ACK to the CAN bus; the error counter stops at the current value. Listening mode is primarily used for automatically detecting bit rates without interfering with network traffic, or for the design of CAN bus analyzers.

24.4.14 Test Mode

When LBEN = 1 in the CAN_WUPTEST register, CAN enters the test mode.

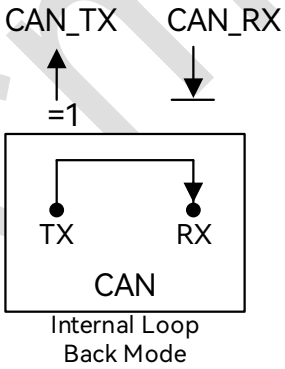
24.4.14.1 External Loop Back Mode

When LBEN=1 and TXC=0 in the CAN_WUPTEST register, CAN enters the external loop back mode. This mode is provided for hardware self-test. CAN ignores acknowledge errors (recessive bit sampled in the acknowledge slot of a data / remote frame) in loop back mode. CAN goes from TX output to RX input, ignoring the actual value of the CAN_RX input pin, and can monitor the transmitted information on the CAN_TX pin.



24.4.14.2 Internal Loop Back Mode

When the LBEN and TXC bits of the CAN_WUPTEST register are set, CAN enters the internal loop back mode. In this mode, the CAN_RX pin is disconnected from the CAN, and the CAN_TX pin is held in a recessive state.



24.5 List of Registers

CAN register base address: 0x40B0_8000

The registers are listed below:

Table 24-2: List of CAN Registers

Offset Address	Name	Description	
0x00	CAN_CONFIG0	0x00	CAN_MR: mode register
		0x01	CAN_CMR: command register
		0x02	CAN_SR: status register

Offset Address	Name	Description	
		0x03	CAN_ISR: interrupt status / acknowledge register
0x04	CAN_CONFIG1	0x04	CAN_IMR: interrupt mask register
		0x05	CAN_RMC: receive data count register
		0x06	CAN_BTR0: bus timing register 0
		0x07	CAN_BTR1: bus timing register 1
0x08	CAN_TXBUF	TX buffer register	
0x0C	CAN_RXBUF	RX buffer register	
0x10	CAN_ACR	Acceptance code register	
0x14	CAN_AMR	Acceptance mask register	
0x18	CAN_ERRCR	0x18	CAN_ECC: error code capture register
		0x19	CAN_RXERR: receive error count register
		0x1A	CAN_TXERR: transmit error count register
		0x1B	CAN_ALC: arbitration lost capture register
0x1C	CAN_NBT	Nominal bit timing register	
0x20	CAN_DBTCR	0x22–0x20	CAN_DBT: data bit timing register
		0x23	CAN_SSPP: secondary sample point position register
0x24	CAN_CONFIG2	0x24	CAN_FDCON: FD control register
		0x25	CAN_FDSTA: FD status register
		0x26	CAN_DPERR: error count register in data phase
		0x27	CAN_APERR: error count register in arbitration phase
0x28	CAN_WUPTEST	0x28	CAN_TEST: test register
		0x2A–0x29	CAN_WUPRE: wake-up timer prescaler register
		0x2B	CAN_WUPFT: wake-up filter time register
0x2C	CAN_WUPET	Wake-up expiration time register	
0x30	CAN_CONFIG3	0x31–0x30	CAN_RXDCNT: receive buffer data count register
		0x32	CAN_TXSCNT: transmit buffer space count register
		0x33	CAN_TDCR: transceiver delay compensation register

Offset Address	Name	Description	
0x34	CAN_IRCONFIG	0x34	CAN_EISR: extended interrupt status / acknowledge register
		0x35	CAN_EIMR: extended interrupt mask register
		0x37–0x36	CAN_RXFTO: RX buffer timeout register
0x38	CAN_RTCONFIG	0x38	CAN_ACFSEL: receive filter selection register
		0x39	CAN_TXBSEL: transmit buffer selection register
		0x3A	CAN_TXBSTA: transmit buffer status register

24.5.1 Configuration Register 0 (CAN_CONFIG0)

CAN_MR register (CAN_CONFIG0[7:0]), mode register

CAN_CMR register (CAN_CONFIG015: 8]), command register

CAN_SR register (CAN_CONFIG0[23:16]), status register

CAN_ISR register (CAN_CONFIG0[31:24]), interrupt status / acknowledge register

Offset address: 0x00

Reset value: 0x0020 0004

Bit	Name	Attribute	Reset Value	Description
31	WUI	R/W	0x0	Wake-up interrupt status bit: This bit is set when CAN wakes up from sleep mode or clears the sleep bit in the CAN_MR register. Write 1 to clear this interrupt.
30	ALI	R/W	0x0	Arbitration lost interrupt status bit: When an arbitration lost is detected during message transmission and CAN becomes the receiver, this bit is set, and reading ALC register can check which bit in the arbitration phase is lost. Write 1 to clear this interrupt.
29	EWI	R/W	0x0	Error warning interrupt status bit: This error warning interrupt bit is set when the ES or BS bit in SR register changes. Therefore, it can be used to detect if CAN enters or exits the bus-off state. Write 1 to clear this interrupt.

Bit	Name	Attribute	Reset Value	Description
28	EPI	R/W	0x0	Error passive interrupt status bit: This bit is set when the CAN bus controller reaches or exits the error passive level (i.e., the status changes from active to passive or vice versa). Write 1 to clear this interrupt.
27	RI	R/W	0x0	RX interrupt status bit: CAN sets this bit to 1 when there is at least one CAN frame data in RX FIFO. After reading the message, CPU must write the RI bit to 1 (message read acknowledgment) to decrement the count of RX message counter (RMC), which does not automatically decrement.
26	TI	R/W	0x0	TX interrupt status bit: The TX interrupt bit is set upon successful transmission. The write pointer can be reset to TX RAM by clearing the TI bit (via writing 1) before writing a new data frame.
25	BEI	R/W	0x0	Bus error interrupt status bit: Set BEI when CAN encounters a bus error in the course of sending or receiving messages. Write 1 to clear this interrupt.
24	DOI	R/W	0x0	RX data overflow interrupt status bit: DOI is set when an RX FIFO overflow occurs. Write 1 to clear this interrupt.
23	RBS	R	0x0	RX FIFIO status: 1: at least one message in FIFO 0: no message in FIFO
22	DSO	R	0x	Data overflow status: 1: RX FIFO overflow triggers an interrupt (if enabled). 0: no overflow has occurred since the last data overflow clearing.
21	TBS	R	0x1	TX buffer status: 1: TX buffer can be written by CPU. 0: TX buffer is locked. A message is being sent or waiting to be sent. If CPU tries to write to the TX buffer in the locked state (TBS = 0), the

Bit	Name	Attribute	Reset Value	Description
				written data is not accepted.
20	EP	R	0x0	Error passive status: 1: CAN is in error passive state. 0: CAN is not in error passive state.
19	RS	R	0x0	RX status bit: 1: CAN is receiving. 0: CAN is not in receive state.
18	TS	R	0x0	TX status bit: 1: CAN is transmitting. 0: CAN is not in transmit state.
17	ES	R	0x0	Error status bit: 1: at least one CAN error counter reaches the error warning limit of 96. 0: normal status
16	BS	R	0x0	Bus status bit: 1: off-line state The CAN controller is in reset mode and the error warning interrupt is triggered (if enabled). The TX error counter is set to 127, and the RX error counter is set to 0. CAN will remain in reset mode until CPU clears the RM bit. After this operation, CAN will wait for the occurrence of 128 bus idle signals (11 consecutive recessive bits), and the TX error counter will count down. Then the BS bit is cleared, the error counter is reset, and the error warning interrupt is triggered (if enabled). 0: normal state; frame transmission and reception can be performed.
15	RXFIFORST	W	0x0	RX FIFO reset
14:13	CMD2	R/W	0x0	TXB2 transmit command: Write: 00: invalid 01: transfer aborted 10: transfer request 11: single-transfer request Read:

Bit	Name	Attribute	Reset Value	Description
				00: idle 01: abort pending 10: transfer pending 11: abort command pending in current transmitting
12:11	CMD1	R/W	0x0	TXB1 transmit command: Write: 00: invalid 01: transfer aborted 10: transfer request 11: single-transfer request Read: 00: idle 01: abort pending 10: transfer pending 11: abort command pending in current transmitting
10:9	CMD0	R/W	0x0	TXB0 transmit command: Write: 00: invalid 01: transfer aborted 10: transfer request 11: single-transfer request Read: 00: idle 01: abort pending 10: transfer pending 11: abort command pending in current transmitting
8	RSV	-	-	Reserved
7	SLEEP	R/W	0x0	Sleep mode: 1: CAN enters sleep mode. 0: CAN exits sleep mode.
6	DMA	R/W	0x0	Enable DMA mode
5:3	RT	R/W	0x0	RX FIFO trigger depth: 000: 1 001: 4

Bit	Name	Attribute	Reset Value	Description
				010: 8 100: 32 101: 64 Note: These bits are valid only in DMA mode.
2	RM	R/W	0x1	Reset mode set bit: 1: CAN works in reset mode. 0: CAN works in other modes. No data transmission and reception are performed in reset mode, which is used for some hardware configurations (e.g. some registers can only be written in reset mode). After reset mode, it can enter listening mode or normal mode.
1	LOM	R/W	0x0	Listening mode set bit: 1: if RM = 0, CAN enters listening mode. 0: if RM = 0, CAN enters normal mode. This bit can only be set in reset mode.
0	AFM	R/W	0x0	Hardware filter selection bit: 1: use single filter 0: use double filters This bit can only be set in reset mode.

Note: In the case of a bus error or arbitration lost, the single-transfer request does not perform retransmission of the lost frames.

24.5.2 Configuration Register 1 (CAN_CONFIG1)

CAN_IMR register (CAN_CONFIG1[7:0]), interrupt mask register

CAN_RMC register (CAN_CONFIG1[15:8]), receive data count register

CAN_BTR register (CAN_CONFIG1[31:16]), bus timing register

Offset address: 0x04

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31	SAM	R/W	0x0	Bus level sampling number selection bit:

Bit	Name	Attribute	Reset Value	Description
				1: sample the bus level three times (for medium/low-speed buses) 0: sample the bus level once (for high-speed bus)
30:28	TSEG2	R/W	0x0	Number of clock cycles for time segment 2: $t_{TSEG2} = t_{SCLK} \times (4 \times TSEG2.2 + 2 \times TSEG2.1 + TSEG2.0 + 1)$
27:24	TSEG1	R/W	0x0	Number of clock cycles for time segment 1: $t_{TSEG1} = t_{SCLK} \times (8 \times TSEG1.3 + +4 \times TSEG1.2 + 2 \times TSEG1.1 + TSEG1.0 + 1)$
23:22	SJW	R/W	0x0	Synchronization jump width: $t_{SJW} = t_{SCLK} \times (2 \times SJW.1 + SJW.0 + 1)$ In order to compensate the phase shift between clock oscillators of different CAN bus controllers, the bit period must be shortened or extended accordingly. SJW defines the maximum number of clock cycles for a resynchronization to change one bit period. During resynchronization, the hardware synchronizes with the RX signal by increasing "1 + SJW" t_{SCLK} in PBS1 segment or decreasing "1–(1+SJW)" t_{SCLK} in PBS2 segment.
21:16	BRP	R/W	0x0	Baud rate prescaler value: $t_{SCLK} = 2 \times t_{CLK} (32 \times BRP.5 + 16 \times BRP.4 + 8 \times BRP.3 + 4 \times BRP.2 + 2 \times BRP.1 + BRP.0 + 1)$ Wherein, $t_{CLK} = 1 / f_{PCLK}$
15:8	RMC	R	0x0	This read-only register holds the number of frames stored in the RX FIFO. This value is incremented each time a frame is successfully received and decremented by clearing the RI interrupt.
7	WUIM	R/W	0x0	Wakeup interrupt enable bit: 1: WUIM enabled 0: WUIM disabled
6	ALIM	R/W	0x0	Arbitration lost interrupt enable bit: enable the CAN transmitter to trigger an interrupt

Bit	Name	Attribute	Reset Value	Description
				when an arbitration lost is detected during transmission and it becomes a CAN receiver: 1: ALI enabled 0: ALI disabled
5	EWIM	R/W	0x0	Error warning interrupt enable bit: enable to trigger an interrupt when the status of the BS or ES bit of CAN_SR register changes: 1: EWI enabled 0: EWI disabled
4	EPIM	R/W	0x0	Error passive interrupt enable bit: enable to trigger an interrupt when the CAN controller enters or exits error passive mode: 1: EPI enabled 0: EPI disabled
3	RIM	R/W	0x0	RX interrupt enable bit: 1: RI enabled 0: RI disabled
2	TIM	R/W	0x0	TX interrupt enable bit: 1: TI enabled 0: TI disabled
1	BEIM	R/W	0x0	Bus error interrupt enable bit: enable to trigger an interrupt when a bus error occurs during CAN transmission or reception: 1: BEI enabled 0: BEI disabled
0	DOIM	R/W	0x0	RX data overflow interrupt enable bit: 1: DOI enabled 0: DOI disabled

Note: The calculation formula and illustration for CAN baud rate are as follows:

$$\mathbf{baud} = \frac{\mathbf{fpclk}}{2*(BRP+1)(TSEG1+TSEG2+3)}$$

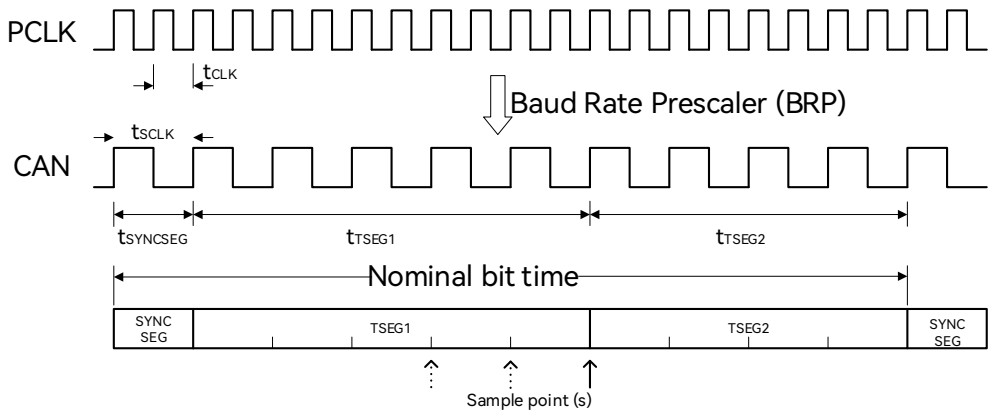


Figure 24-1: Illustration of CAN Baud Rate Calculation

24.5.3 Transmit Buffer Register (CAN_TXBUF)

Offset address: 0x08

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:0	TXBUF	W	0x0	The transmit buffer register is used to write the CAN frames to be transmitted over CAN network. Writing to this register performs automatic increment of the internal write pointer. By writing TI bit in ISR register, the write pointer can be reset to the address 0h of the TX memory.

24.5.4 Receive Buffer Register (CAN_RXBUF)

Offset address: 0x0C

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:0	RXBUF	R	0x0	The receive buffer register is used to read CAN frames received from the CAN network. Reading this register will automatically increment the read address pointer of the internal FIFO (increment after reading).

After receiving a frame of CAN data, the RMC register count is increased by 1, and the CAN controller will write data into RX FIFO one by one. RBS will be set when a 32-bit data is written. After a frame of data is written, the RI flag bit is set.

24.5.5 Acceptance Code Register (CAN_ACR)

The CAN has multiple filters that can be specifically accessed using ACFSEL.SEL. The contents of the acceptance code register for a specified filter can only be read or written through the ACRx register when the CAN is in reset mode.

Offset address: 0x10

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:0	ACR3_0	R/W	0x0	The acceptance code register contains the arbitration bit of the message to be received, and the corresponding acceptance mask register defines the bits to be compared and the irrelevant bits.

24.5.6 Acceptance Mask Register (CAN_AMR)

The AMR register contains a bit mask that defines the bits to be compared. The CAN has multiple filters that can be specifically accessed using ACFSEL.SEL. The contents of the acceptance mask register for a specified filter can only be read or written through the AMRx register when the CAN is in reset mode.

Offset address: 0x14

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:0	AMR3_0	R/W	0x0	The acceptance mask register defines the bits to be compared and the irrelevant bits. Set the corresponding bit to 1 indicating not to compare the corresponding bit in the ACR register.

24.5.7 Error Configuration Register (CAN_ERRCR)

The CAN_ECC read-only register (CAN_ERRCR[7:0]) holds the error code regarding the last bus error that occurred on the CAN network. This register is read-only. The CAN core will not update this register until the previous bus error is acknowledged (by acknowledging the bus error interrupt).

CAN_RXERR register (CAN_ERRCR[15:8]), receive error count register

CAN_TXERR register (CAN_ERRCR[23:16]), transmit error count register

CAN_ALC register (CAN_ERRCR[31:24]), arbitration lost capture register

Offset address: 0x18

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:29	RSV	-	-	Reserved
28:24	ALC	R	0x0	Arbitration lost position
23:16	TXERR	R	0x0	Lower 8 bits of the current value of TX error counter. If a bus-off event occurs, the TX error counter will be initialized to 127 to calculate the minimum protocol-defined time (128 occurrences of the bus idle signal). Read TXERR during this time can obtain information about the bus-off recovery state.
15:8	RXERR	R	0x0	Current value of the RX error counter. If a bus shutdown event occurs, the RX error counter will be initialized to 0.
7	RXWRN	R	0x0	Set this bit to 1 when the RXERR counter value is greater than or equal to 96.
6	TXWRN	R	0x0	Set this bit to 1 when the TXERR counter value is greater than or equal to 96.
5	EDIR	R	0x0	Direction of data transmission at the occurrence of an error: 0: transmitting 1: receiving
4	ACKER	R	0x0	Set this bit to 1 when an ACK error occurs.

Bit	Name	Attribute	Reset Value	Description
3	FRMER	R	0x0	Set this bit to 1 when a frame format error occurs.
2	CRCER	R	0x0	Set this bit to 1 when an CRC error occurs.
1	STFER	R	0x0	Set this bit to 1 when a stuffing error occurs.
0	BER	R	0x0	Set this bit to 1 when a bit error occurs.

The CAN controller is able to determine the exact in-frame location of the arbitration lost. Immediately thereafter an “arbitration lost interrupt” will be generated. In addition, the number of bits is captured in the arbitration lost capture register. Once the master controller reads the contents of this register, the capture function will be activated for the next arbitration lost. This feature allows CAN to monitor each CAN bus access. For diagnostics or during system configuration, each case of unsuccessful arbitration can be determined.

ALC[4]	ALC[3]	ALC[2]	ALC[1]	ALC[0]	Decimal	Description
0	0	0	0	0	0	Arbitration lost in **ID28 / 10**
0	0	0	0	1	1	Arbitration lost in **ID27 / 9**
0	0	0	1	0	2	Arbitration lost in **ID26 / 8**
0	0	0	1	1	3	Arbitration lost in **ID25 / 7**
0	0	1	0	0	4	Arbitration lost in **ID24 / 6**
0	0	1	0	1	5	Arbitration lost in **ID23 / 5**
0	0	1	1	0	6	Arbitration lost in **ID22 / 4**
0	0	1	1	1	7	Arbitration lost in **ID21 / 3**
0	1	0	0	0	8	Arbitration lost in **ID20 / 2**
0	1	0	0	1	9	Arbitration lost in **ID19 / 1**
0	1	0	1	0	10	Arbitration lost in **ID18 / 0**
0	1	0	1	1	11	Arbitration lost in **SRTR / RTR**
0	1	1	0	0	12	Arbitration lost in **IDE**
0	1	1	0	1	13	Arbitration lost in **ID17***
0	1	1	1	0	14	Arbitration lost in **ID16***
0	1	1	1	1	15	Arbitration lost in **ID15***
1	0	0	0	0	16	Arbitration lost in **ID14***
1	0	0	0	1	17	Arbitration lost in **ID13***
1	0	0	1	0	18	Arbitration lost in **ID12***
1	0	0	1	1	19	Arbitration lost in **ID11***
1	0	1	0	0	20	Arbitration lost in **ID10***

ALC[4]	ALC[3]	ALC[2]	ALC[1]	ALC[0]	Decimal	Description
1	0	1	0	1	21	Arbitration lost in **ID9***
1	0	1	1	0	22	Arbitration lost in **ID8***
1	0	1	1	1	23	Arbitration lost in **ID7***
1	1	0	0	0	24	Arbitration lost in **ID6***
1	1	0	0	1	25	Arbitration lost in **ID5***
1	1	0	1	0	26	Arbitration lost in **ID4***
1	1	0	1	1	27	Arbitration lost in **ID3***
1	1	1	0	0	28	Arbitration lost in **ID2***
1	1	1	0	1	29	Arbitration lost in **ID1***
1	1	1	1	0	30	Arbitration lost in **ID0***
1	1	1	1	1	31	Arbitration lost in **RTR**

Note*: Only for extended frames.

24.5.8 Nominal Bit Timing Register (CAN_NBT)

When the EXTBT bit in the FD control register (FDCON) is set, the nominal bit timing register is used to configure the bit time for the arbitration phase of the CAN FD frame. This register can only be accessed when the CAN is in reset mode. The CAN bit time can be programmed within a range of 3 to 385 time units. According to the formula, the CAN time quantum can be programmed in the range from 1 to 1024 clock cycles.

$$t_q = (NBRP + 1) \times t_{clk}$$

Offset address: 0x1C

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:25	NSJM	R/W	0x0	Synchronization jump width in arbitration phase
24:18	NSEG2	R/W	0x0	Time segment after sample point in arbitration phase
17:10	NSEG1	R/W	0x0	Time segment before sample point in arbitration phase
9:0	NBRP	R/W	0x0	Baud rate prescaler in arbitration phase

24.5.9 Data Bit Timing Configuration Register (CAN_DBTCR)

CAN_DBT register (CAN_DBTCR[23:0]) is the data bit timing register.

When the FDEN bit in the FD control register (FDCON) is set, the data bit timing register is used to configure the bit time for the data phase of the CAN FD frame. The bit time of the data phase must be shorter than or equal to that of the arbitration phase. This register can only be accessed when the CAN is in reset mode. The CAN bit time in data phase can be programmed within a range of 3 to 81 time units. According to the formula, the data CAN time quantum in data phase can be programmed in the range from 1 to 1024 clock cycles.

$$t_q = (DBRP + 1) \times t_{clk}$$

CAN_SSPP register (CAN_DBTCR[31: 24]) is the secondary sample point position register. This read-only register contains information about the secondary sample point, which is defined as the sum of the measured transceiver loop delay and the TDCO value. The transceiver loop delay measures the time from the change edge of the transmitted FDF to r0, to the corresponding receive edge in each transmitted CAN FD frame.

Offset address: 0x20

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31	RSV	-	-	Reserved
30:24	SSPP	R	0x0	Secondary sample point position
23:20	NSJM	R/W	0x0	Synchronization jump width in data phase
19:16	DSEG2	R/W	0x0	Time segment after sample point in data phase
15:10	DSEG1	R/W	0x0	Time segment before sample point in data phase
9:0	DBRP	R/W	0x0	Baud rate prescaler in data phase

Note: When the FDEN bit is set, the CAN baud rate calculation formula is as follows:

$$\mathbf{baud} = \frac{\mathbf{fpclk}}{(DBRP+1)(DSEG1+DSEG2+3)}$$

24.5.10 Configuration Register 2 (CAN_CONFIG2)

CAN_FDCON register (CAN_CONFIG2[7:0]) is the FD control register.

The FDCON register contains the control bits for CAN FD mode. This register can only be accessed when the CAN is in reset mode.

CAN_FDSTA register (CAN_CONFIG2[15:8]) is the FD status register.

This read-only register stores the current state of CAN controller and the protocol information detected in CAN FD frame format. The error indication will be updated only when CAN is running in FD mode (FDEN is set). These bits are automatically cleared upon successful reception or transmission of a CAN FD frame.

CAN_DPERR register (CAN_CONFIG2[23:16]) is the error count register in data phase.

This read-only register holds the current value of the counter, which is incremented when the transmitter and receiver detect an error in the data phase of a CAN FD frame. The counter stops counting when it reaches 255, and the DLO interrupt flag is set when a DPERR error occurs again. The value of this register can be reset by writing the DPERR register.

CAN_APERR register (CAN_CONFIG2[31:24]) is the error count register in arbitration phase.

This read-only register holds the current value of the counter, which is incremented when the transmitter and receiver detect an error in the arbitration phase of a CAN FD frame. The counter stops counting when it reaches 255, and the ALO interrupt flag is set when an APERR error occurs again. The value of this register can be reset by writing the APERR register.

Offset address: 0x24

Reset value: 0x00

Bit	Name	Attribute	Reset Value	Description
31:24	APERR	R/W*	0x0	Error counter in arbitration phase

Bit	Name	Attribute	Reset Value	Description
23:16	DPERR	R/W*	0x0	Error counter in data phase
15:14	STATE	R	0x0	DCAN operation state: 00: integrated—waiting for 11 recessive bits after reset or bus disconnection 01: idle—waiting for frame start 10: receiver—node operating as receiver 11: transmitter—node operating as transmitter
13	RSV	-	-	Reserved
12	PEE	R	0x0	Protocol exception: A recessive state is detected at the “res” position, and the system enters the bus integrating state.
11	STFERR	R	0x0	Stuffing error: When BRS is set, a stuffing error occurs in the data phase of the CAN FD frame.
10	FRMERR	R	0x0	Form error: When BRS is set, the fixed-format bit field contains at least one illegal bit in the data phase of the CAN FD frame.
9	CRCERR	R	0x0	CRC error: The calculated CRC differs from the CRC received in the CAN FD frame.
8	BITERR	R	0x0	Bit error: This bit is set when there is an inconsistency between the transmitted and received bits in the CAN FD frame.
7	RSV	-	-	Reserved
6	PED	R/W	0x0	Protocol exception disabled: 0: When a recessive state at the “res” position is detected on the CAN bus, the system will enter the bus integrating state. 1: A recessive state at the “res” position on the CAN bus is treated as a form error.
5	REOM	R/W	0x0	Restricted operation mode: 0: Send (active or passive) error frames or overload frames when conditions are met. 1: The node does not send (active or passive) error

Bit	Name	Attribute	Reset Value	Description
				frames or overload frames. When an error or overload condition is met, it does not send dominant bits, but waits for the bus to be idle.
4	DAR	R/W	0x0	Automatic retransmission enable: 0: Automatic retransmission enabled 1: Automatic retransmission disabled
3	ISO	R/W	0x0	ISO CAN format selection: 0: The frame format conforms to Bosch CAN FD specification. 1: The frame format conforms to ISO 11898-1: 2015.
2	EXTBT	R/W	0x0	Bit time prescaler in arbitration phase: 0: Configure the bit time of the arbitration phase using the CAN_BTR0 and CAN_BTR1 registers. 1: Configure the bit time of the arbitration phase using the CAN_NBT register.
1	BRSEN	R/W	0x0	Bit rate switching in data phase: 0: No bit rate switches within the CAN FD frame. 1: Bit rate switches from the nominal bit rate of the arbitration phase to the bit rate of the data phase.
0	FDEN	R/W	0x0	FD frame format: 0: Classic CAN frame 1: CAN FD frame

Note*: Writing any value clears the corresponding register.

24.5.11 Test and Wake-up Configuration Register (CAN_WUPTEST)

CAN_TEST register (CAN_WUPTEST[7: 0]) is the test register. It contains the control bits for loop back mode. This register can only be accessed when the CAN is in reset mode. When the CAN is in loop back mode (LBEN = 1), the RXD input is disconnected from the CAN core. If the acceptance filtering is passed, the transmitted frame is received and saved in the FIFO.

CAN_WUPDIV register (CAN_WUPTEST[15: 8]) is the wake-up timer prescaler register.

The CAN clock is divided by (WUPDIV + 1) to generate the toggle signals for the WUPFT and WUPET counters. This register can only be accessed when the CAN is in reset mode. Calculate the timer trigger interval through the following formula:

$$t = \frac{WUPDIV + 1}{f_{clk}}$$

CAN_WUPFT register (CAN_WUPTEST[31:16]) is the wake-up filter time register.

The set value of this 16-bit register must be greater than 0 compared to the actual value of the filter counter. When the counter value is greater than or equal to the WUPFT register, the state detected on the CAN bus is valid. This register can only be accessed when the CAN is in reset mode.

Offset address: 0x28

Reset value: 0x00

Bit	Name	Attribute	Reset Value	Description
31:16	WUPFT	R/W	0x0	Wake-up filter time
15:8	WUPDIV	R/W	0x0	Wake-up timer prescaler
7:2	RSV	-	-	Reserved
1	TXC	R/W	0x0	TXD terminal state selection in loop back mode: 0: The transmission frame appears on the TXD terminal. 1: The TXD terminal remains recessive (TXD = 1).
0	LBEN	R/W	0x0	Enable bit for loop back mode

24.5.12 Wake-up Expiration Time Register (CAN_WUPET)

This is a 20-bit register that is compared with the actual value of the expiration counter. When this counter value is less than the WUPET register value, the detected wake-up is valid. This register can only be accessed when the CAN is in reset mode.

CAN_WUPETL

Offset address: 0x2C

Reset value: 0x00

Bit	Name	Attribute	Reset Value	Description
31:20	RSV	-	-	Reserved
19:0	WUPET	R/W	0x0	Wake-up expiration time

24.5.13 Configuration Register 3 (CAN_CONFIG3)

CAN_RXDCNT register (CAN_CONFIG3[15:0]), the receive buffer data count register, is a 16-bit read-only register that saves the count of data in RX FIFO. Upon successful reception of a message, it is incremented by the length of the message stored in RX FIFO (4-byte alignment). After reading the RXBUFn register, this register is decremented. The counter stops counting when the RX FIFO data count reaches its maximum value. The value of this register is cleared after a hardware reset or in reset mode.

CAN_TXSCNTR register (CAN_CONFIG3[23:16]), the transmit buffer space count register, is a read-only register that indicates the available space (in bytes) in the TX BUFFER, specified by an 8-bit value. The counter decrements each time the TXBUFn register is written to. After a message is successfully sent, the counter is set to the maximum available value. The size of the TX BUFFER is fixed at 128 bytes.

CAN_TDCR register (CAN_CONFIG3[31:23]) is the transceiver delay compensation register used to configure the transceiver delay compensation feature, which is available during frame transmission in CAN FD mode. This register can only be accessed when the DCAN is in reset mode.

Offset address: 0x30

Reset value: 0x0012 0000

Bit	Name	Attribute	Reset Value	Description
31	TDCEN	R/W	0x0	Transceiver delay compensation enable
30:24	TDCO	R/W	0x0	Offset value added to the transceiver loop delay
23:21	RSV	-	-	Reserved
20:16	TXSCNT	R	0x12	TX FIFO space counter
15:0	RXDCNT	R	0x0	RX FIFO data counter

24.5.14 Extended Interrupt and Receive Buffer Configuration Register (CAN_IRCONFIG)

CAN_EISR register (CAN_IRCONIFG[7:0]) is the extended interrupt status and acknowledge register. This register notifies the host of important events in the CAN core either through interrupts or polling (when interrupts are disabled).

CAN_EIMR register (CAN_IRCONFIG[15:8]) is the extended interrupt mask register that used to mask interrupts. Writing 1 to the bit in the EIMR register enables the corresponding interrupt, while clearing the bit disables the interrupt. By default, all interrupts are disabled.

CAN_RXFTO register (CAN_IRCONFIG[31:16]) is a 16-bit register that specifies the maximum invalid time for the RX FIFO when there is at least one data frame. The value in this register is compared to a counter that increments at each bit time during the arbitration phase. When these values are equal, an RXTO interrupt will be generated. The value of this register is cleared after a hardware reset or in reset mode.

Offset address: 0x34

Reset value: 0x00

Bit	Name	Attribute	Reset Value	Description
31:16	RXFTO	R/W	0x0	RX FIFO timeout
15:12	RSV	-	-	Reserved
11	DLOM	R/W	0x0	Mask DPERR overflow interrupt
10	ALOM	R/W	0x0	Mask APERR overflow interrupt
9	RXTOM	R/W	0x0	Mask RX FIFO timeout interrupt
8	RXFTM	R/W	0x0	Mask RX FIFO trigger interrupt

Bit	Name	Attribute	Reset Value	Description
7:4	RSV	-	-	Reserved
3	DLO	R	0x0	DPERR counter overflow
2	ALO	R	0x0	APERR counter overflow
1	RXTO	R/W	0x0	RX FIFO timeout: This flag is set when there has been no write or read operation on RX FIFO within the user-defined time (as specified in the RXFTO register), and at least one data item is present in RX FIFO during this period. Writing a 1 to this bit clears it.
0	RXFT	R/W	0x0	RX FIFO reaches the set trigger value: This flag is set when RX FIFO reaches the set trigger value (as defined by RT[2:0] in the MR register). Writing a 1 to this bit clears it.

24.5.15 Receive Filter Selection and Transmit Buffer Configuration Register (CAN_RTCONFIG)

The CAN_ACFSEL register (CAN_RTCONFIG[7:0]) is used to select the receive filter. This register can only be accessed when the CAN is in reset mode.

The CAN_TXBSEL register (CAN_RTCONFIG[15:8]) is used to select access to the TX buffer. Each write to the TXBSEL register resets the TX buffer write address.

CAN_TXBSTA register (CAN_RTCONFIG[23:16]) is the transmit buffer status register.

Offset address: 0x38

Reset value: 0x00

Bit	Name	Attribute	Reset Value	Description
31:23	RSV	-	-	Reserved
22:20	TXBAD	R	0x0	TX buffer abort completion: Each TX buffer has its own abort completion flag. These bits are set when the transmission is executed. When an abort request is set during

Bit	Name	Attribute	Reset Value	Description
				transmission pending, these bits will be set at the end of the transmission, regardless of whether the transmission is successful or not. In the case where an abort request is set before the transmission starts from the specified TX buffer, the TXBAD bit is immediately set. This bit is cleared by writing to the CMR register to request a new transmission.
19	RSV	-	-	Reserved
18:16	TXBTD	R	0x0	TX buffer transmission completion: Each TX buffer has its own transmission completion flag, which is set upon successful completion of a transmission from the corresponding TX buffer. This bit is cleared by writing to the CMR register to request a new transmission.
15:10	RSV	-	-	Reserved
9:8	SELTX	R/W	0x0	TX buffer selection: 00: TX buffer 0 01: TX buffer 1 10: TX buffer 2
7:5	RSV	-	-	Reserved
4:0	SEL	R/W	0x0	Select filter

24.6 Operation Procedure

24.6.1 Transmit CAN Data Frame

1. Turn on the CAN clock, release the reset, and multiplex the CAN function pins.
2. Configure the bus timing registers CAN_BTR0/CAN_BTR1.
3. Clear the error flag bit/interrupt flag bit in CAN_ISR register.
4. Configure the interrupt enable register CAN_IMR to enable TI interrupt (optional).
5. Configure the mode register CAN_MR[1] to enter the normal mode.

6. Configure the TX buffer register CAN_TXBUF, write the contents of CAN data frames according to the defined format in sequence, and write 32 bits of data at a time.
7. Configure the command register CAN_CMR[2] to start transmitting.
8. Wait for the status register CAN_SR[5] being set to 1 (if TI interrupt is enabled, here the trigger of TI interrupt can be waited), then the data is transmitted.

24.6.2 Receive CAN Data Frame

1. Turn on the CAN clock, release the reset, and multiplex the CAN function pins.
2. Configure the bus timing registers CAN_BTR0/CAN_BTR1.
3. Clear the error flag bit/interrupt flag bit in CAN_ISR register.
4. Configure the interrupt enable register CAN_IMR to enable RI interrupt (optional).
5. Configure the RX filter and set CAN_MR[0] to 1 if a single filter is used. The CAN_ACR register configures what the user needs to filter, and the CAN_AMR register selects the bits that need to be compared with those of the CAN_ACR register. If no comparison is required, all bits of the CAN_AMR register shall be set to 1.
6. Configure the mode register CAN_MR[1] to enter the normal mode.
7. Wait for the status register CAN_SR[7] being set to 1 (if RI interrupt is enabled, here the trigger of RI interrupt can be waited), read the data in RX buffer register CAN_RXBUF several times until all the data are retrieved.

25 Inter-integrated Circuit Interface (I2C0)

25.1 Overview

The purpose of the I2C module is to facilitate the read and write operations by CPU to the slave devices connected on the I2C bus. When the CPU performs a write operation to the slave device, it configures the configuration registers of the I2C module through the bus, then sends control information and operands to the data communication registers of the I2C module. After parsing the command, the I2C module transmits the data from its data channel registers to the slave device via the I2C bus. Once the transmission is complete, the final status is fed back to the CPU via an interrupt. The process for the CPU to read data from the slave device is similar to that of the write operation.

25.2 Main Features

- Dual-line I2C serial interface
- Operating modes: master receive/transmit, slave receive/transmit
- Standard mode (100 Kb/s), fast mode (400 Kb/s) and fast mode+ (1 Mb/s)
- 7-bit and 10-bit addressing
- Broadcast addressing
- Interrupt polling

25.3 Pin Description

Table 25-1: I2C0 Pin Description

Function Pin	Alternate Function Pin	Direction	Functional Description
I2C0_SDA	PA5, PA10, PA14, PB4, PB5, PB7, PB9, PC1, PC5, PC7, PD11	Input/output	Data
I2C0_SCL	PA4, PA9, PA15, PB6, PB8, PC0, PC4, PC6, PD10	Input/output	Clock

25.4 Functional Description

25.4.1 SDA Holding Time

It meets the I2C protocol specification requirements.

25.5 Register Description

I2C0 register base address: 0x4700_3000

The registers are listed below:

Table 25-2: List of I2C0 Registers

Offset Address	Name	Description
0x00	I2C0_CR	I2C configuration register
0x04	I2C0_CLR	I2C configuration clear register
0x08	I2C0_STAT	I2C status register
0x0C	I2C0_DATA	I2C data register
0x10	I2C0_CCR	I2C baud rate configuration register
0x14	I2C0_SAD0	I2C slave address register 0
0x18	I2C0_SADM0	I2C slave address mask register 0
0x1C	I2C0_XSAD0	I2C slave extended address register
0x20	I2C0_XSADM0	I2C slave extended address mask register
0x24	I2C0_SRST	I2C reset register
0x28	I2C0_SAD1	I2C slave address register 1
0x2C	I2C0_SADM1	I2C slave address mask register 1
0x30	I2C0_SAD2	I2C slave address register 2
0x34	I2C0_SADM2	I2C slave address mask register 2
0x38	I2C0_SAD3	I2C slave address register 3
0x3c	I2C0_SADM3	I2C slave address mask register 3

25.5.1 I2C Configuration Register (I2C0_CR)

Offset address: 0x00

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:9	RSV	-	-	Reserved
8	GCAVAL	R	0x0	General Call address flag bit: 0: General Call address not received 1: General Call address received
7	IEN	R/W	0x0	I2C interrupt enable: 1: enabled 0: invalid
6	ENAB	R/W	0x0	I2C interface enable: 1: enabled 0: invalid
5	STA	R/W	0x0	Start signal enable: 1: The I2C module enters the master mode and sends a START signal when the bus is idle. When this bit is set while the I2C is in master mode, a RESTART signal will be sent. When this bit is set while the I2C is in slave mode, the I2C module will enter master mode and send a START signal after completing the current data transmission and the bus is released and idle. After sending the START signal, this bit is automatically cleared; writing 0 to this bit has no effect.
4	STP	R/W	0x0	Stop signal enable: 1: send stop signal When this bit is set while the I2C is in master mode, a STOP signal will be sent. When this bit is set while the I2C is in slave mode, the I2C module will interpret it as having received a STOP signal and will not send a STOP signal on the bus.

Bit	Name	Attribute	Reset Value	Description
				If both the STA and STP bits are set simultaneously, the I2C module will first send a STOP signal (in master mode) and then send a START signal. After sending the STOP signal, this bit is automatically cleared; writing 0 to this bit has no effect.
3	IFLG	R/W	0x0	Interrupt flag bit: This bit will be set when the I2C_STAT register is in any state other than 0xf8. Writing to the CLR_IFLG bit of the I2C_CLR register will clear this bit to 0. When the STP bit is set to 1, that is, after the STOP signal is sent, this bit will also be cleared to 0.
2	AAK	R/W	0x0	Answer flag enable: 0: NACK (no response in slave mode) 1: ACK When AAK = 1, the I2C module will respond with ACK in the following cases: 1. The address in the slave address register is received. 2. A general call address is received after enabling the GC bit. 3. A byte of data is received in master/slave mode. After setting this bit to 1, writing to the CLR_AAK bit of the I2C_CLR register will clear this bit to 0; writing 0 to this bit has no effect.
1	SLAV10M	R	0x0	Flag bit that the received data matches the data in the extended address register in slave mode: 0: The data does not match. 1: The data is matched.
0	SLAV7M	R	0x0	Flag bit that the received data matches the data in the address register in slave mode: 0: The data does not match. 1: The data is matched.

25.5.2 I2C Configuration Clear Register (I2C0_CLR)

Offset address: 0x04

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:8	RSV	-	-	Reserved
7	CLR_IEN	W	0x0	I2C interrupt enable clear register: 1: interrupt enable cleared Writing 0 has no effect on the bit value.
6	CLR_ENAB	W	0x0	I2C enable clear register: 1: I2C interface disabled, I2C does not perform address matching and ignores the information on the SCL/SDA lines. Writing 0 has no effect on the bit value.
5	CLR_STA	W	0x0	Start flag clear register: 1: clear the TX start flag Writing 0 has no effect on the bit value.
4	RSV	-	-	Reserved
3	CLR_IFLG	W	0x0	Interrupt flag clear register: 1: clear the interrupt flag Writing 0 has no effect on the bit value.
2	CLR_AAK	W	0x0	Acknowledge flag clear register: 1: clear the acknowledge flag Writing 0 has no effect on the bit value.
1:0	RSV	-	-	Reserved

25.5.3 I2C Status Register (I2C0_STAT)

Offset address: 0x08

Reset value: 0x0000 00F8

Bit	Name	Attribute	Reset Value	Description
31:8	RSV	-	-	Reserved
7:0	STA	R	0xF8	I2C status register

Meanings of different codes in the status field of I2C_STAT register:

Status Code	I2C Bus and Hardware Status
0x00	Due to the occurrence of illegal start or stop conditions, bus error will occur on the master or the selected slave; 0x00 will also occur when external interference causes I2C to enter an undefined state.
0x08	The START flag has been sent.
0x10	The RESTART flag has been sent.
0x18	The slave address + header Write have been sent and an ACK is received.
0x20	The slave address + header Write have been sent and a NAK is received.
0x28	In master mode, the data in I2C_DATA has been sent and an ACK has been received.
0x30	In master mode, the data in I2C_DATA has been sent and a NAK is received.
0x38	Arbitration loss (address or data byte)
0x40	The slave address + header Read have been sent and an ACK is received.
0x48	The slave address + header Read have been sent and a NAK is received.
0x50	In master mode, the data byte has been received and an ACK has been sent.
0x58	In master mode, the data byte has been received and a NAK has been sent.
0x60	Its own slave address + header Write have been received and an ACK has been sent.
0x68	In master mode, an arbitration loss is detected, its own slave + header Write have been received, and an ACK has been sent.
0x70	The universal call address (0x00) has been received, and an ACK has been sent.
0x78	In master mode, an arbitration loss is detected, the general call address has been received, and an ACK has been sent.
0x80	After receiving its own slave address, the data byte has been received and an ACK has been returned.
0x88	After receiving its own slave address, the data byte has been received and a NAK has been returned.
0x90	After receiving the general call address, the data byte has been received and an ACK has been returned.
0x98	After receiving the general call address, the data byte has been received and a NAK has been returned.
0xA0	In slave mode, a stop condition or a repeated start condition is received.
0xA8	Its own slave address + header Read have been received and an ACK has been sent.
0xB0	In master mode, an arbitration loss is detected, the general call address +

Status Code	I2C Bus and Hardware Status
	header Read have been received, and an ACK has been sent.
0xB8	In slave mode (AAK = 1), the data has been sent and an ACK has been received.
0xC0	In slave mode (AAK = 1), data have been sent and a NAK has been received.
0xC8	In slave mode (AAK = 0), the last data byte has been sent and an ACK has been received.
0xD0	In slave mode (AAK = 0), the last data byte has been sent and a NAK has been received.
0xE0	The second byte of the 10-bit slave address + header Write have been sent and an ACK is received.
0xE8	The second byte of the 10-bit slave address + header Write have been sent and a NAK is received.
0xF8	No relevant status information is available, IFLG = 0.

If an illegal start or stop signal appears on the I2C bus, it will enter the bus error status (status code 0x00). The user can clear the IFLG bit state by setting STP, and the I2C module will return to the idle state, and this operation will not send a STOP signal on the I2C bus.

25.5.4 I2C Data Register (I2C0_DATA)

Offset address: 0x0C

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:8	RSV	-	-	Reserved
7:0	DATA	R/W	0x0	I2C data register: In I2C TX mode, write TX data to this register. In I2C RX mode, read RX data from this register.

25.5.5 I2C Baud Rate Configuration Register (I2C0_CCR)

Offset address: 0x10

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:7	RSV	-	-	Reserved
6:4	CCRM	R/W	0x0	Baud rate configuration bit M

Bit	Name	Attribute	Reset Value	Description
3:0	CCRM	R/W	0x0	Baud rate configuration bit N

FOSCL = f_{SCL} = P_{CLK} / (2^M × (N + 1) × 10); wherein, FOSCL is the frequency of SCL output by I2C.

25.5.6 I2C Slave Address Register 0 (I2C0_SAD0)

Offset address: 0x14

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:8	RSV	-	-	Reserved
7:1	ADR0	R/W	0x0	I2C slave address 0
0	GC0	R/W	0x0	Broadcast address acknowledge enable: 0: disabled 1: enabled

25.5.7 I2C Slave Address Mask Register 0 (I2C0_SADM0)

Offset address: 0x18

Reset value: 0x0000 00FE

Bit	Name	Attribute	Reset Value	Description
31:8	RSV	-	-	Reserved
7:1	AMR0	R/W	0x7F	Address mask register 0 in I2C slave mode: Each bit corresponds to a bit in ADR0. When the corresponding bit is 1, compare the value of the corresponding bit in ADR0 in the case of I2C module being used as a slave. When the corresponding bit is 0, do not compare the value of the corresponding bit in ADR0 in the case of I2C module being used as a slave.
0	RSV	-	-	Reserved

25.5.8 10-bit I2C Slave Address Register (I2C0_XSAD0)

Offset address: 0x1C

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:11	RSV	-	-	Reserved
10:1	XADR	R/W	0x0	10-bit address bit in I2C slave mode
0	XGC	R/W	0x0	Broadcast address acknowledge enable in 10-bit addressing mode: 0: disabled 1: enabled

25.5.9 10-bit I2C Slave Address Mask Register (I2C0_XSADM0)

Offset address: 0x20

Reset value: 0x0000 01FE

Bit	Name	Attribute	Reset Value	Description
31:9	RSV	-	-	Reserved
8:1	XAMR	R/W	0xFF	10-bit address mask register 0 in I2C slave mode: Each bit corresponds to a bit in XADR. When the corresponding bit is 0, do not compare the value of the corresponding bit in XADR in the case of I2C module being used as a slave. When the corresponding bit is 1, compare the value of the corresponding bit in XADR in the case of I2C module being used as a slave.
0	RSV	-	-	Reserved

25.5.10 I2C Reset Register (I2C0_SRST)

Offset address: 0x24

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:0	SRST	W	0x0	Writing any value to this register resets the I2C module.

25.5.11 I2C Slave Address Register 1 (I2C0_SAD1)

Offset address: 0x28

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:8	RSV	-	-	Reserved
7:1	ADR1	R/W	0x0	I2C slave address 1
0	GC1	R/W	0x0	Broadcast address acknowledge enable: 0: disabled 1: enabled

25.5.12 I2C Slave Address Mask Register 1 (I2C0_SADM1)

Offset address: 0x2C

Reset value: 0x0000 00FE

Bit	Name	Attribute	Reset Value	Description
31:8	RSV	-	-	Reserved
7:1	AMR1	R/W	0x7F	Address mask register 1 in I2C slave mode: Each bit corresponds to a bit in ADR1. When the corresponding bit is 0, do not compare the value of the corresponding bit in ADR1 in the case of I2C module being used as a slave. When the corresponding bit is 1, compare the value of the corresponding bit in ADR1 in the case of I2C module being used as a slave.
0	RSV	-	-	Reserved

25.5.13 I2C Slave Address Register 2 (I2C0_SAD2)

Offset address: 0x30

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:8	RSV	-	-	Reserved
7:1	ADR2	R/W	0x0	I2C slave address 2

Bit	Name	Attribute	Reset Value	Description
0	GC2	R/W	0x0	Broadcast address acknowledge enable: 0: disabled 1: enabled

25.5.14 I2C Slave Address Mask Register 2 (I2C0_SADM2)

Offset address: 0x34

Reset value: 0x0000 00FE

Bit	Name	Attribute	Reset Value	Description
31:8	RSV	-	-	Reserved
7:1	AMR2	R/W	0x7F	Address mask register 2 in I2C slave mode: Each bit corresponds to a bit in ADR2. When the corresponding bit is 0, do not compare the value of the corresponding bit in ADR2 in the case of I2C module being used as a slave. When the corresponding bit is 1, compare the value of the corresponding bit in ADR2 in the case of I2C module being used as a slave.
0	RSV	-	-	Reserved

25.5.15 I2C Slave Address Register 2 (I2C0_SAD3)

Offset address: 0x38

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:8	RSV	-	-	Reserved
7:1	ADR3	R/W	0x0	I2C slave address 3
0	GC3	R/W	0x0	Broadcast address acknowledge enable: 0: disabled 1: enabled

25.5.16 I2C Slave Address Mask Register 3 (I2C0_SADM3)

Offset address: 0x3C

Reset value: 0x0000 00FE

Bit	Name	Attribute	Reset Value	Description
31:8	RSV	-	-	Reserved
7:1	AMR3	R/W	0x7F	Address mask register 3 in I2C slave mode: Each bit corresponds to a bit in ADR3. When the corresponding bit is 0, do not compare the value of the corresponding bit in ADR3 in the case of I2C module being used as a slave. When the corresponding bit is 1, compare the value of the corresponding bit in ADR3 in the case of I2C module being used as a slave.
0	RSV	-	-	Reserved

25.6 Operation Procedure

The interface can operate in one of the four following modes:

- Slave transmitter
- Slave receiver
- Master transmitter
- Master receiver

Note: The I2C interface module can only operate in either master mode or slave mode, but cannot work in both modes simultaneously. Therefore, it is important to ensure that the registers I2C0_CR[6] and I2C0_CR[0] cannot be set to 0 and 1 (or 1 and 0) respectively at the same time.

25.6.1 Master and Slave Initialization Configuration

- **Master initialization:**

  1. Multiplex the corresponding GPIO into I2C_SCL and I2C_SDA.
  2. Activate the I2C clock in the system register to run I2C.
  3. Write 1 to I2C0_SRST to reset the I2C module.
  4. Configure I2C0_CCR[3: 0] and [6: 4] to set the I2C communication rate (standard/fast/fast-plus).
  5. For slave mode, set I2C0_CR[2] to 1 and I2C0_CR[6] to 1.

- **Slave initialization:**

  1. Multiplex the corresponding GPIO into I2C_SCL and I2C_SDA.
  2. Activate the I2C clock in the system register to run I2C.
  3. Write 1 to the I2C reset register to reset the I2C module.
  4. Load its slave address into I2C0_SAD0[7:1] / I2C0_SAD1[7:1] / I2C0_SAD2[7:1] / I2C0_SAD3[7:1] / I2C0_XSAD[10:1], and set the address matching registers I2C0_SADM0[7:1] / I2C0_SADM1[7:1] / I2C0_SADM2[7:1] / I2C0_SADM3[7:] / I2C0_XSADM[10:1] / I2C0_SADx[0] to enable the address broadcast acknowledgment (if required).
  5. Set I2C0_CR[7] to 1 to enable I2C interrupt (if required).
  6. Set I2C0_CR[2] to 1 and set I2C0_CR[6] to enable master transmit function.
  7. Set the destination address of the slave to be operated on.

25.6.2 Master Transmitter

1. Write 1 to I2C0_CR[5] to send the START flag.

2. Wait for the value of I2C0_STAT[7:0] to become 0x08 (START flag has been sent); and this bit will be cleared automatically once the START flag is sent successfully.
3. Write SAL (7 bits/10 bits) + W(0) to I2C0_DATA[7:0].
4. Write 1 to I2C0_CLR[3] to clear the interrupt flag, then the master SCL line sends a clock signal and the SDA line sends SLA + W.
5. Wait for the value of I2C0_STAT[7:0] to become 0x18 (SLAVE address + header Write have been sent and an ACK has been received).
6. Write the data to be sent/the memory address of the slave device to be written to I2C0_DATA[7:0] (for 10-bit addressing, the second byte of device address + header Write are sent).
7. Receive the slave device status I2C0_STAT[7:0] = 0xE0, indicating that the second byte of address has been sent and an ACK has been received, continue to send data cyclically (this action occurs only for 10-bit addressing).
8. Write 1 to I2C0_CLR[3] to send the data.
9. Wait for the value of I2C0_STAT[7:0] to become 0x28 (data in I2C0_DATA has been sent and an ACK has been received).
10. Repeat steps 7−10 until all the data to be sent are sent.
11. Write 1 to I2C0_CR[4] to send the STOP flag, then the transmission is completed.

25.6.3 Master Receiver

1. Write 1 to I2C0_CR[5] to send the START flag.
2. Wait for the value of I2C0_STAT[7:0] to become 0x08 (START flag has been sent); and this bit will be cleared automatically once the START flag is sent successfully.

3. Write SLA (7 bits) + R(1) to I2C0_DATA[7:0] (this operation occurs for 7-bit addressing).
4. Write 1 to I2C0_CLR[3] to clear the interrupt flag, then the master SCL line sends a clock signal and the SDA line sends SLA + R (this operation occurs for 7-bit addressing).
5. Write the first byte of SLA + W(0) to I2C0_DATA[7:0] (this operation occurs for 10-bit addressing).
6. Wait for the value of I2C0_STAT[7:0] to become 0x18 (SLAVE address + header Write have been sent and an ACK has been received) (this operation occurs for 10-bit addressing).
7. Write the second segment of SLA to I2C0_DATA[7:0] (this operation occurs for 10-bit addressing).
8. Receive the slave device status I2C0_STAT[7:0] = 0xE0, indicating that the second byte of address has been sent and an ACK has been received (this action occurs only for 10-bit addressing).
9. Write 1 to I2C0_CR[5] to send the RESTART flag.
10. Write 1 to I2C0_CLR[3] to clear the interrupt flag, and wait for the value of I2C0_STAT[7:0] to become 0x10 (RESTART flag has been sent) (this operation occurs for 10-bit addressing).
11. Write the first byte of SLA + R(1) to I2C0_DATA[7:0] (this operation occurs for 10-bit addressing).
12. Wait for the value of I2C0_STAT[7:0] to become 0x40 (SLAVE address + header Read have been sent and an ACK has been received) (this operation occurs for 10-bit addressing).
13. Write 1 to I2C0_CR[2] to set the AAK bit of the I2C0_CR register.
14. Write 1 to I2C0_CLR[3] to start receiving the data.

15. Wait for the value of I2C0_STAT[7:0] to become 0x50 (the data byte has been received and an ACK has been sent), and read the data received in I2C0_DATA.
16. Repeat steps 14–15 until all the data are received.
17. Write 1 to I2C0_CR[4] to send the STOP flag, then the transmission is completed.

25.6.4 Slave Transmitter

Query mode:

1. Wait for the value of I2C0_STAT[7:0] to become 0x60 (its own slave address + header Write have been received and an ACK has been sent).
2. Write 1 to I2C0_CLR[3] to clear the interrupt flag, then the slave SCL line releases the clock signal and the SDA line receives the data.
3. Wait for the value of I2C0_STAT[7:0] to become 0x80 (after receiving its own slave address, the data byte has been received and an ACK has been returned).
4. Read the data in I2C0_DATA and save it to a variable.
5. Repeat steps 2–4 until all the data are received.
6. Wait for the value of I2C0_STAT[7:0] to become 0xA0 (a stop condition or a repeated start condition is received in slave mode).
7. Write 1 to I2C0_CLR[3] to clear the interrupt flag, then the slave SCL line releases the clock signal and the SDA line receives the data.

Interrupt mode:

1. Wait for the value of I2C0_STAT[7:0] to become 0x80 (after receiving its own slave address, the data byte has been received and an ACK has been returned), and an interrupt is generated.

2. In the interrupt service routine, read the data from I2C0_DATA and save it to a variable.
3. Write 1 to I2C0_CLR[3] to clear the interrupt flag, then the slave SCL line releases the clock signal and the SDA line receives the data.

25.6.5 Slave Receiver

1. Write 1 to the AAK bit of I2C0_CR register to enable acknowledgment.
2. Wait for the value of I2C0_STAT[7:0] to become 0xA8 (its own slave address + header Read have been received and an ACK has been sent).
3. Write the data to be sent to I2C0_DATA[7:0].
4. Write 1 to I2C0_CLR[3] to clear the interrupt flag, then the slave SCL line releases the clock signal.
5. Wait for the value of I2C0_STAT[7:0] to become 0xB8 (in slave mode (AAK = 1), the data has been sent and an ACK has been received).
6. Repeat step 5 until all the data are sent.
7. Write 1 to I2C0_CLR[3] to clear the interrupt flag, then the slave SCL line releases the clock signal.
8. Wait for the value of I2C0_STAT[7:0] to become 0xA0 (a stop condition or a repeated start condition is received in slave mode).
9. Write 1 to I2C0_CLR[3] to clear the interrupt flag, then the slave SCL line releases the clock signal.

26 Enhanced Inter-integrated Circuit Interface (I2C1 & I2C2)

26.1 Overview

The purpose of the I2C module is to facilitate the read and write operations by CPU to the slave devices connected on the I2C bus. When the CPU performs a write operation to the slave device, it configures the configuration registers of the I2C module through the bus, then sends control information and operands to the data communication registers of the I2C module. After parsing the command, the I2C module transmits the data from its data channel registers to the slave device via the I2C bus. Once the transmission is completed, the final status is fed back to the CPU via an interrupt. The process for the CPU to read data from the slave device is similar to that of the write operation.

26.2 Main Features

- Dual-line I2C serial interface
- Standard mode (100 Kb/s), fast mode (400 Kb/s) and fast mode+ (1 Mb/s)
- Master or slave mode
- 7-bit or 10-bit addressing mode
- Transmission in 7-bit or 10-bit combined addressing modes
- Bulk transfer mode
- 2-byte transmit and receive buffers
- Interrupt and polling operations
- Bit and byte waiting at all speeds
- Programmable SDA hold time
- Bus clearing

- DMA operation
- SMBus (system management bus) / PMBus (power management bus)
- SMBus slave detection and response to ARP commands
- Address resolution protocol (ARP)

26.3 Pin Description

Table 26-1: I2C1& I2C2 Pin Description

Function Pin	Alternate Function Pin	Direction	Functional Description
I2C1_SDA	PA2, PA8, PA12, PB11, PB12, PB13, PB14, PB15, PC9, PD14	Input/output	Data
I2C1_SCL	PA3, PA7, PA9, PA11, PB10, PB13, PB14, PD10, PD15	Input/output	Clock
I2C2_SDA	PA1, PA9, PA13, PB11, PC1, PC9, PD2, PD8	Input/output	Data
I2C2_SCL	PA0, PA8, PA14, PB10, PC0, PC12, PD9	Input/output	Clock

26.4 Functional Description

26.4.1 SMBus / PMBus

SMBus is used to provide predictable communication lines between the system and its devices, describing the device timeout definition and conditions thereof.

26.4.1.1 Bus Protocol

A typical SMBus features a set of commands by which data can be read and written. All commands are one byte in length, but the length of their parameters and return values can vary. According to the SMBus specification, the most significant bit (MSB) is transmitted first. For a given device, there are 11 command protocols, including: Quick Command, Send Byte, Receive Byte, Write Byte, Write Word, Read Byte, Read Word, Process Call, Block Read, Block Write, and Block Write-Block Read Process Call.

The SMBus protocol for message transactions is usually different from the I2C data transfer commands, but it is still possible to program the SMBus master for I2C data transfer. The following table describes the SMBus protocol generated by the TX FIFO command of I2C.

In SMBus master mode, all received data is available in RX FIFO. In SMBus slave mode, all bus protocol command codes and data bytes will be placed in RX FIFO, and the data bytes of read request must be sent using TX FIFO, similar to I2C mode.

Table 26-2: SMBus Protocol Usage Table

Protocol	Required TXFIFO Commands	DATA (I2C_DATA_CMD[7:0])	CMD (I2C_DATA_CMD[8])	STOP (I2C_DATA_CMD[9])	Remarks
Quick Command	1	Not applicable	Set R/W.	Set to 1.	Set bit 11 and bit 16 of I2C_TAR to 1.
Send Byte	1	Data byte	Set to 0.	Set to 1.	-
Receive Byte	1	Not applicable	Set to 1.	Set to 1.	-
Write Byte	2	Command code	Set to 0.	Set to 0.	-
		Data byte	Set to 0.	Set to 1.	-
Write Word	3	Command code	Set to 0.	Set to 0.	-
		Data byte low	Set to 0.	Set to 0.	-
		Data byte high	Set to 0.	Set to 1.	-
Read Byte	2	Command code	Set to 0.	Set to 0.	-
		Not applicable	Set to 1.	Set to 1.	-
Read Word	3	Command code	Set to 0.	Set to 0.	-
		Not applicable	Set to 1.	Set to 0.	-
		Not applicable	Set to 1.	Set to 1.	-
Process Call	5	Command code	Set to 0.	Set to 0.	-
		Data byte low	Set to 0.	Set to 0.	-
		Data byte high	Set to 0.	Set to 0.	-
		Not applicable	Set to 1.	Set to 0.	-
		Not applicable	Set to 1.	Set to 1.	-
Block Write	N + 1	Command code	Set to 0.	Set to 0.	-
		Data byte	Set to 0.	Set to 0.	-

Protocol	Required TXFIFO Commands	DATA (I2C_DATA_CMD[7:0])	CMD (I2C_DATA_CMD[8])	STOP (I2C_DATA_CMD[9])	Remarks
		N + 1) Data byte N	Set to 0.	Set to 1.	-
Block Read	N+1	Command code	Set to 0.	Set to 0.	-
		Not applicable	Set to 0.	Set to 0.	-
		N + 1) Not applicable	Set to 0.	Set to 1.	-
Block Write-Block Read Process Call	M + N + 1	Command code	Set to 0.	Set to 0.	-
		Data byte 1	Set to 0.	Set to 0.	-
		M + 1) Data byte M	Set to 0.	Set to 0.	-
		M + 2) Not applicable	Set to 1.	Set to 0.	-
		M + 3) Not applicable	Set to 1.	Set to 0.	-
		M + N + 1) Not applicable	Set to 1.	Set to 1.	-
SMBUS Host Notify Protocol	3	Data byte low	Set to 0.	Set to 0.	I2C_TAR is set to SMBus master address (0001 000).

The I2C slave receives via the Quick command only if SMBUS_SLAVE_QUICK_CMD_EN of I2C_CR is enabled. Whenever this bit is selected, the slave receives only the Quick command and no other bus protocols. The I2C slave issues a SMBUS_QUICK_DET interrupt upon receipt of the Quick command.

SMBus introduces a packet error checking mechanism by appending a PEC byte to the end of the bus protocol. This can be accomplished by adding additional commands (PEC bytes) while transmitting and by decoding them in software when receiving.

26.4.1.2 SMBus Address Resolution Protocol

SMBus slave address conflicts can be resolved by the master dynamically assigning a new unique address to each slave device. This feature allows devices to be “hot-swapped”.

SMBus introduces a 128-bit unique device ID (UDID) for each device in the system to isolate each device for address assignment. The high 96 bits of the UDID are set to 0, while the low 32 bits are controlled by the I2C_SMBUS_ARP_UDID_LSB register.

The SMUBS_PERSISTANT_SLV_ADDR_EN bit in the IC_CR register is used to indicate whether I2C supports persistent slave addresses.

The I2C master can issue generic and directed address resolution protocol (ARP) commands to assign dynamic addresses to slaves in the SMBus system.

Table 26-3: Table of SMBus ARP Commands Derived via TxFIFO Commands in I2C

ARP Command	Required TXFIFO Commands	Command / Data (I2C_DATA_CMD[7:0])	CMD Bit (I2C_DATA_CMD[8])	STOP bit (I2C_DATA_CMD[9])	Remarks
Prepare for ARP	2	Command = ‘00000001’	Set to 0.	Set to 0.	I2C_TAR[6:0] is set to SMBus default address (1100 001).
		PEC byte	Set to 0.	Set to 1.	
Reset device (general)	2	Command = ‘0000 0010’	Set to 0.	Set to 0.	I2C_TAR[6:0] is set to SMBus default address (1100 001).
		PEC byte	Set to 0.	Set to 1.	
Get UDID (general)	20	Command = ‘0000 0011’	Set to 0.	Set to 0.	1. I2C_TAR[6:0] is set to SMBus default address
		Not applicable	Set to 1.	Set to 0.	
		Not applicable	Set to 1.	Set to 0.	
		Not applicable	Set to 1.	Set to 0.	

ARP Command	Required TXFIFO Commands	Command / Data (I2C_DATA_CMD[7:0])	CMD Bit (I2C_DATA_CMD[8])	STOP bit (I2C_DATA_CMD[9])	Remarks
		PEC byte	Set to 1.	Set to 1.	(1100 001). 2. Perform 16 reads of the 128 UDID bytes. 3. The last read command accesses the slave address.
Assign address	20	Command = '0000 0100'	Set to 0.	Set to 0.	1. I2C_TAR[6:0] is set to SMBus default address (1100 001). 2. Perform 16 writes of the 128 UDID bytes. 3. The last write command accesses the slave address.
		Byte count = 17	Set to 0.	Set to 0.	-
		UDID byte 15	Set to 0.	Set to 0.	
		UDID byte 14	Set to 0.	Set to 0.	
		Assigned address	Set to 0.	Set to 0.	
		PEC byte	Set to 01.	Set to 1.	
Get UDID (directed)	19	Command = '0000 0011'	Set to 0.	Set to 0.	1. I2C_TAR[6:0] is set to SMBus default address
		{Slave address[6:0],1}	Set to 1.	Set to 0.	
		Not applicable	Set to 1.	Set to 0.	

ARP Command	Required TXFIFO Commands	Command / Data (I2C_DATA_CMD[7:0])	CMD Bit (I2C_DATA_CMD[8])	STOP bit (I2C_DATA_CMD[9])	Remarks
		Not applicable	Set to 1.	Set to 0.	(1100 001). 2. Perform 16 reads of the 128 UDID bytes. 3. The last read command accesses the slave address.
		PEC byte	Set to 1.	Set to 1.	
Reset device (directed)	2	Command = {slave address[6:0],0}	Set to 0.	Set to 0.	I2C_TAR[6:0] is set to SMBus default address (1100 001).
		PEC byte	Set to 0.	Set to 1.	

26.4.1.3 Performing ARP in Master Mode

When I2C is used as a SMBus master, the following steps shall be performed to assign a unique address to each slave device in order to resolve address conflicts:

1. After a reset or cold start, the SMBus master issues a “Prepare for ARP” command, indicating that the master will perform ARP to allocate dynamic addresses for all devices. The slave must refresh any pending master notification commands.
2. Upon receiving an ACK for the “Prepare to ARP” command, it indicates that there are devices in the system that support ARP. The master then issues a “Get UDID” command. A NACK signifies that either there are no devices supporting ARP or that all slaves currently have resolved addresses. In this case, the master must complete the operations outlined in step 8.
3. The I2C master issues a “Get UDID” command to receive the UDID information from the

slave for dynamic address allocation.

4. If the first three bytes of the “Get UDID” command are ACKed and the received byte count is 0x11, the master issues an “Assign Address” command. Otherwise, the master must complete the operations described in step 8 to indicate that ARP has been completed.
5. The master issues an “Assign Address” command to dynamically assign an address to the slave corresponding to the UDID received via the “Get UDID” command.
6. If the assigned address packet is ACKed, the master will remove the assigned address from the address pool and jump to step 3 to obtain the UDID of another slave device. If the address packet is NACKed, the master will not remove this address from the address pool and will then jump to step 3 to obtain the UDID of the same slave or another slave.
7. If the assigned address packet is ACKed, the master will store the assigned address along with the UDID feature of the device in the used address pool.
8. The master jumps to step 3 to issue the “Get UDID” command again to receive the UDIDs of other slaves. If a NACK is received for the “Get UDID,” the master jumps to step 9.
9. The I2C can switch to slave mode to detect requests from devices for master notification protocols.
10. If the I2C switches to slave mode and detects the master notification protocol, indicating that the slave is requesting a dynamic address, the master must perform ARP as described in step 11.
11. If in master mode, jump to step 3 to perform ARP; otherwise, jump to step 12.
12. The I2C switches to master mode and jumps to step 3 to perform ARP.

26.4.1.4 Performing ARP in Slave Mode

As a SMBus slave, the I2C performs the following tasks:

- Decode ARP commands and respond based on the internal status flags:

- SMBUS_SLAVE_ADDR_VALID and SMBUS_SLAVE_ADDR_RESOLVED of the I2C register.
- Generate and verify the PEC byte for ARP commands.
- An ACK will be generated for the PEC byte only if it matches the CRC value calculated from the received data. If not, NACK the PEC byte.

When another SMBus master on the bus issues an ARP command and the I2C device has participated in ARP, the I2C as a slave performs the following operations:

1. After a reset or cold start, the I2C slave checks if it supports a persistent slave address (PSA).
2. If the I2C has a PSA, it is indicated by the set address valid flag and set in the slave address register (I2C_SAR). If the flag is not set, proceed to step 4.
3. The I2C persistent slave stores the persistent address in I2C_SAR and sets the address valid flag to 1, with the address resolved flag set to 0.
4. The I2C non-persistent slave (non-PSA) clears the address valid and address resolved flags.
5. The I2C checks the slave address field in the received packet for the ARP default address to determine if it is an ARP command or a normal command. If it matches, proceed to step 6; otherwise, go to step 25.
6. If the I2C detects a packet sent to the SMBus device default address, it checks the command field to determine if it is the “Prepare to ARP” command. If it is, proceed to step 7; otherwise, go to step 8.
7. Upon receiving the “Prepare to ARP” command, the I2C acknowledges the packet and clears the address resolved flag to participate in the ARP process. The I2C then proceeds to step 5 and waits for another SMBus packet.
8. The I2C checks the command field to verify if the “Reset Device” command has been issued. If it is, proceed to step 9; otherwise, go to step 10.

9. Upon receiving the “Reset Device” command, the I2C acknowledges the packet and clears the address resolved and address valid flags (if it is non-PSA, I2C_CR[19]=0). The I2C then proceeds to step 5 and waits for another SMBus packet.
10. The device checks the command to verify if the “Assign Address” command has been issued. If it is, proceed to step 11; otherwise, go to step 13.
11. Upon receiving the “Assign Address” command, the I2C compares its UDID with the received bytes. If any byte does not match, the I2C will not acknowledge that byte and subsequent bytes. If all bytes in the UDID match, the device proceeds to step 12; otherwise, it goes to step 5 and waits for another SMBus packet.
12. After matching the UDID in step 11, DW_apb_i2c receives the slave address and uses it to set the I2C_SAR register. The I2C sets its address valid and address resolved flags, indicating it has received a dynamic address and will no longer respond to the “Get UDID” command unless it receives a “Prepare to ARP” or “Reset Device” command. The I2C now proceeds to step 5 and waits for another SMBus packet.
13. The I2C checks the command field to verify if the “Get UDID” command has been issued. If it is, proceed to step 14; otherwise, go to step 19.
14. Upon receiving the “Get UDID” command, the I2C checks its address resolved flag to determine if it must participate in the ARP process. If it is set, its address has been resolved by the ARP master, so the device proceeds to step 5 and waits for another SMBus packet. If the ARP flag has been cleared, proceed to step 15.
15. The I2C returns its UDID and monitors the SMBus data line for conflicts. If a conflict is detected at any time, the I2C sets the SLV_ARB_LOST bit and stops the transmission. It then continues with step 5 and waits for another SMBus packet. If no conflict is detected, the I2C proceeds to step 16.
16. The I2C checks its address valid (AV) flag to determine the value to return to the device slave address field. If the AV flag is set, proceed to step 17; otherwise, go to step 18.

17. When the AV flag is set, the current I2C_SAR is valid, so the device returns it to the device slave address field while monitoring the SMBus data line for conflicts. The I2C continues with step 5 and waits for another SMBus packet.。
18. If the AV flag is not set, the current slave address (I2C_SAR) is invalid. The I2C returns a value of FFh and monitors the SMBus data line for conflicts. If the ARP master receives a value of FFh, the device requires address allocation. The I2C continues with step 5 and waits for another SMBus packet.
19. The I2C may be receiving direct commands. If the address valid flag is set and the address matches the one in I2C_SAR, proceed to step 20; otherwise, continue with step 5 to wait for another SMBus packet.
20. If the address valid flag is set, check if the command is a “Reset Device” direct command. If it is, proceed to step 21; otherwise, go to step 22.
21. Upon receiving the “Reset Device” command, the I2C acknowledges the packet and clears the address resolved and address valid flags (if it is non-PSA, I2C_CR[19]=0). The I2C then proceeds to step 5 and waits for another SMBus packet.
22. The I2C checks if the received command is a “Directed Get UDID” command. If so, proceed to step 23 to return the UDID information; otherwise, go to step 24.
23. If the received command is a “Directed Get UDID” command, return the UDID information along with the current slave address, then go to step 5 and wait for another SMBus packet.
24. If the received command is not a “Directed Get UDID” command, the I2C has not received a valid ARP command, so it will NACK the command and continue with step 5 to wait for another SMBus packet.
25. If the address valid bit is set, proceed to step 26; otherwise, go to step 5 and wait for another SMBus packet. The received address is not the SMBus device default address,

and the packet may be the address for I2C core functions. The device checks its address valid bit to determine whether to respond.

26. When the address valid bit is set, the I2C has a valid slave address. It compares the received slave address with its own slave address; if they match, the I2C proceeds to step 27; otherwise, it goes to step 5 and waits for another SMBus packet.
27. The I2C receives the packet addressed to its core functions, acknowledges the packet, and processes it accordingly. The I2C then continues with step 5 and waits for another SMBus packet.

26.4.2 Bus Clearing

The I2C supports a bus clearing function that provides recovery of the data (SDA) and clock (SCL) lines when the clock or data lines are pulled low abnormally.

26.4.2.1 SDA Pull-down Recovery

If the SDA line is pulled down, the master will perform the following operations, as shown in Figures 26-1 and 26-2.

1. The master will send a maximum of 9 clock pulses to recover the low level of the bus within these 9 clock cycles.
2. If the SDA line recovers within these 9 clock pulses, the master will send a STOP condition to release the bus.
3. If the SDA line has not recovered by the end of the 9th clock pulse, the system will require a hardware reset.

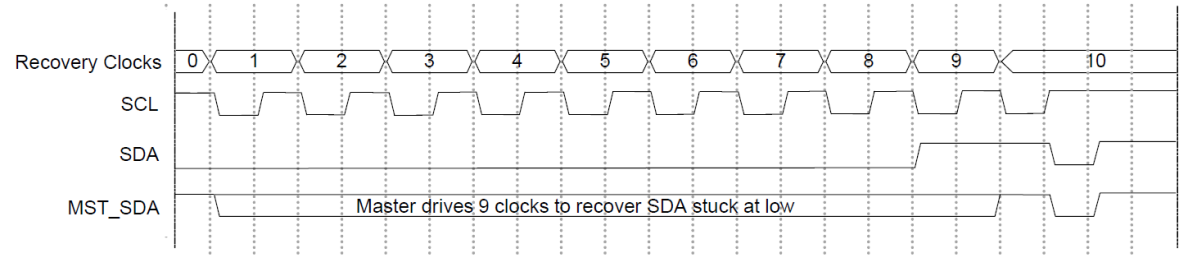


Figure 26-1: Recovery of SDA Using 9 SCL Clock Pulses

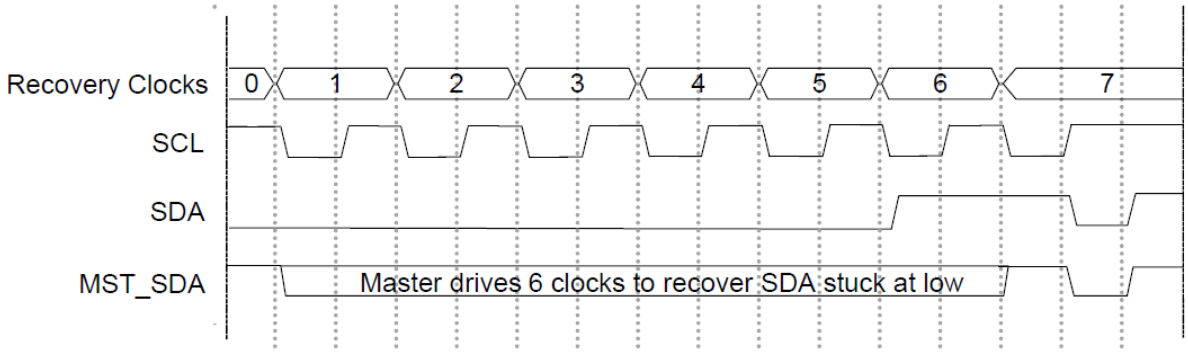


Figure 26-2: Recovery of SDA Using 6 SCL Clock Pulses

26.4.3 Configuring I2C SCLHCNT and SCLLCNT

When I2C operates as a master in standard mode (100 Kb/s), fast mode (400 Kb/s), or fast mode+ (1 Mb/s), the *CNT registers must be set up before I2C transmission to ensure I/O timing. The *CNT registers are as follows:

- I2C_SSSCLHCNT
- I2C_SSSCLLCNT
- I2C_FSSCLHCNT
- I2C_FSSCLLCNT

26.4.3.1 Minimum Setting Values for High and Low Levels

- The I2C_SSSCLLCNT and I2C_FSSCLLCNT registers must be greater than I2C_FSSPKLEN + 7.
- The I2C_SSSCLHCNT and I2C_FSSCLHCNT registers must be greater than I2C_FSSPKLEN + 5.

26.4.3.2 SCLHCNT and SCLLCNT Calculation Methods

The following describes how to calculate I2C_FSSCLHCNT and I2C_FSSCLLCNT in fast mode. Similarly, values for standard mode and fast mode+ can also be calculated.

- In fast mode (400 kb/s), the SCL period is 2.5 μs.
- For example, if the APB clock of I2C is 48 MHz, then the I2C_CLK period is 20.8 ns.
- According to the protocol, the minimum time for high and low levels of SCL is as follows:

  MIN_SCL_LOWtime_FS = 1300 ns

  MIN_SCL_HIGHtime_FS = 600 ns

  MIN_SCL_LOWtime_SS = 4700 ns

  MIN_SCL_HIGHtime_SS = 4000 ns

  MIN_SCL_LOWtime_FS+ = 500 ns

  MIN_SCL_HIGHtime_FS+ = 260 ns

$$\frac{SCL\_PERIOD\_FS}{I2C\_FSSCLHCNT + I2C\_FSSCLLCNT} = I2C\_CLK\_PERIOD$$

I2C_FSSCLLCNT x I2C_CLK_PERIOD ≥ MIN_SCL_LOWtime_FS

Substituting the SCL and I2C clock frequencies into the equation yields:

$$\frac{2500}{I2C\_FSSCLHCNT + I2C\_FSSCLLCNT} = 20.8$$

I2C_FSSCLLCNT × 20.8 ≥ 1300

Solving for I2C_FSSCLHCNT and I2C_FSSCLLCNT:

I2C_FSSCLHCNT + I2C_FSSCLLCNT = 120.9

I2C_FSSCLLCNT ≥ 62.5

Rounding I2C_FSSCLLCNT up gives I2C_FSSCLLCNT = 63. Consequently, I2C_FSSCLHCNT = 58.

26.4.4 SDA Hold Time

The I2C protocol specification requires that the minimum hold time for the SDA signal is 300 ns in standard mode and fast mode, and 0 ns in fast mode+.

Due to varying line delays encountered by each application, I2C includes a software-configurable I2C_SDA_HOLD register for dynamically adjusting the SDA hold time.

Bit[15:0] is used for the SDA hold time during master and slave transmission (from high to low).

Bit[23:16] is used for extended SDA transitions (if any), as long as the SCL on the receiving side is high (in either master mode or slave mode).

If different speed modes require different SDA hold times, the I2C_SDA_HOLD register must be rewritten when the speed mode changes. The I2C_SDA_HOLD register can only be written when I2C is disabled (I2C_EN[0] = 0).

26.4.4.1 SDA Timing During Reception

Speed Mode	Max. Value of RX_HOLD
Standard mode	I2C_SS_SCL_HCNT – I2C_FS_SPKLEN - 3
Standard mode or fast mode+	I2C_SS_SCL_HCNT – I2C_FS_SPKLEN - 3

26.4.4.2 SDA Timing During Transmission

The TX_HOLD register can be used to modify the timing of the SDA signal generated by I2C (ic_sda_oe). Each value in the TX_HOLD register represents one I2C clock cycle.

When I2C operates in master mode, the minimum hold time is one I2C clock cycle. Therefore, when TX_HOLD is set to 0, I2C will continue to drive SDA for one I2C clock cycle after SCL becomes 0. For other values in TX_HOLD:

- After SCL becomes 0, I2C will drive SDA for TX_HOLD * I2C clock cycles.

When I2C operates in slave mode, the minimum hold time is (I2C_FS_SPKLEN + 7) * I2C clock cycles. Therefore, when the value of TX_HOLD is less than I2C_FS_SPKLEN + 7, I2C will continue to drive SDA for (I2C_FS_SPKLEN + 7) * I2C clock cycles after SCL becomes 0.

- After SCL becomes 0, I2C will drive SDA for TX_HOLD * I2C clock cycles.

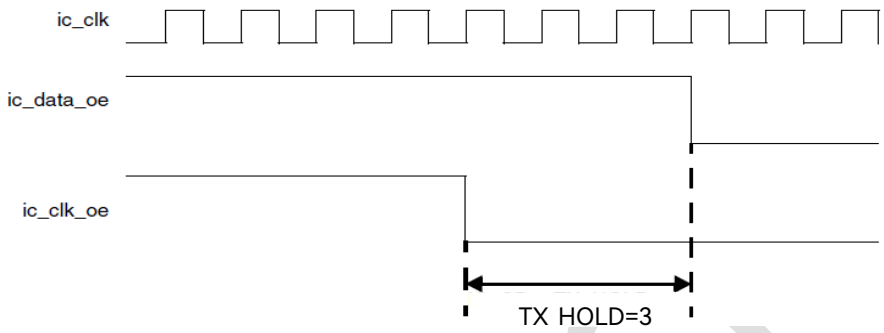


Figure 26-3: Timing Diagram for I2C Master Mode TX_HOLD = 3

26.4.5 Communication Using DMA

The I2C interface supports transmitting and receiving data using DMA. DMA transmission or DMA reception can be enabled individually by setting the corresponding bits in the DMA register. A DMA request is generated when the data register becomes empty during transmission or becomes full during reception. The DMA request must be responded to before the current byte transmission is complete.

- **DMA transmission**

  The DMA transmission mode can be activated by setting the TXEN bit in DMA register. After allocating a DMA channel for I2C, the DMA controller will load the data from the preset memory into the DR register when transmitting data.

- **DMA reception**

  The DMA reception mode can be activated by setting the RDMAE bit in DMA register. After allocating a DMA channel for I2C, the DMA controller will transfer the data from the DR register to the preset memory each time a data byte is received.

26.5 Register Description

I2C1 register base address: 0x40B0_E000

I2C2 register base address: 0x40B0_F000

The registers are listed below:

Table 26-4: List of I2C1 & I2C2 Registers

Offset Address	Name	Description
0x00	I2C_CR	I2C control register
0x04	I2C_TAR	I2C slave address access register
0x08	I2C_SAR	I2C slave address register
0x10	I2C_DATACMD	I2C data command register
0x14	I2C_SSSCLHCNT	I2C standard mode SCL high-level configuration register
0x18	I2C_SSSCLLCNT	I2C standard mode SCL low-level configuration register
0x1C	I2C_FSSCLHCNT	I2C fast mode SCL high-level configuration register
0x20	I2C_FSSCLLCNT	I2C fast mode SCL low-level configuration register
0x2C	I2C_ISR	I2C interrupt status register
0x30	I2C_INTMASK	I2C interrupt mask register
0x34	I2C_RAWISR	I2C raw interrupt status register
0x38	I2C_RXTL	I2C RX FIFO threshold register
0x3C	I2C_TXTL	I2C TX FIFO threshold register
0x40	I2C_CLR	I2C combined and independent interrupt clear register
0x44	I2C_CLRRXUNDER	I2C RX_UNDER interrupt clear register
0x48	I2C_CLRRXOVER	I2C RX_OVER interrupt clear register
0x4C	I2C_CLRTXOVER	I2C TX_OVER interrupt clear register
0x50	I2C_CLRRDREQ	I2C RD_REQ interrupt clear register
0x54	I2C_CLRTXABRT	I2C TX_ABRT interrupt clear register
0x58	I2C_CLRRXDONE	I2C RX_DONE interrupt clear register
0x5C	I2C_CLRACTIVITY	I2C ACTIVITY interrupt clear register
0x60	I2C_CLRSTOPDET	I2C STOP_DET interrupt clear register
0x64	I2C_CLRSTARTDET	I2C START_DET interrupt clear register

Offset Address	Name	Description
0x68	I2C_CLRGENCALL	I2C GEN_CALL interrupt clear register
0x6C	I2C_EN	I2C enable register
0x70	I2C_SR	I2C status register
0x74	I2C_TXFLR	I2C transmit buffer depth register
0x78	I2C_RXFLR	I2C receive buffer depth register
0x7C	I2C_SDAHOLD	I2C SDA hold time register
0x80	I2C_TXABRTSRC	I2C transmit fail interrupt source register
0x84	I2C_SLVDATANACKONLY	I2C slave data NACK generation register
0x88	I2C_DMACR	I2C DMA control register
0x8C	I2C_DMATDLR	I2C DMA transmit data level register
0x90	I2C_DMARDLR	I2C DMA receive data level register
0x94	I2C_SDASETUP	I2C SDA setup time register
0x98	I2C_ACKGENERALCALL	I2C broadcast call ACK register
0x9C	I2C_ENSR	I2C enable status register
0xA0	I2C_FSSPKLEN	I2C spike suppression limit register
0xA8	I2C_CLRRD	I2C RESTART_DET interrupt clear register
0xAC	I2C_SCLSLT	I2C SCL pull-low timeout register
0xB0	I2C_SDASLT	I2C SDA pull-low timeout register
0xB4	I2C_CLRSSD	I2C SCL pull-low detection interrupt clear register
0xBC	I2C_SMBUSCLOCKLOWSEXT	I2C SMBUS slave clock extension timeout register
0xC0	I2C_SMBUSCLOCKLOWMEXT	I2C SMBUS master clock extension timeout register
0xC4	I2C_SMBUSTHIGHMAXIDLECOUNT	I2C SMBUS maximum bus idle count register
0xC8	I2C_SMBUSISR	I2C SMBUS interrupt status register
0xCC	I2C_SMBUSINTMASK	I2C SMBUS interrupt mask register
0xD0	I2C_SMBUSRAWISR	I2C SMBUS raw interrupt status register
0xD4	I2C_CLRSMBUSISR	I2C SMBUS interrupt status clear register
0xDC	I2C_SMBUSUDIDLSB	I2C SMBUS ARP UDID LSB register

26.5.1 I2C Control Register (I2C_CR)

Offset address: 0x00

Reset value: 0x0000 0065

Bit	Name	Attribute	Reset Value	Description
31:20	RSV	-	-	Reserved
19	SMBUS_PERSISTANT_SLV_ADDR_EN	R/W	0x0	Enable the I2C as a persistent or non-persistent slave.
18	SMBUS_ARP_EN	R/W	0x0	Only for slave mode: Control whether I2C enables address resolution logic in SMBus mode.
17	SMBUS_SLAVE_QUICK_CMD_EN	R/W	0x0	Only for slave mode: 0: I2C receives all protocols except the QUICK command. 1: In SMBus mode, I2C receives only the QUICK command.
16:12	RSV	-	-	Reserved
11	BUS_CLEAR_FEATURE_CTRL	R/W	0x0	Only for master mode: 0: bus clearing function enabled 1: bus clearing function disabled
10	STOP_DET_IF_MASTER_ACTIVE	R/W	0x0	Only for master mode: 0: STOP_DET interrupt is generated regardless of whether the master is active or not. 1: STOP_DET interrupt is generated only when the master is active.
9	RX_FIFO_FULL_HLD_CTRL	R/W	0x0	Whether I2C holds the bus when the RX FIFO is full: 0: do not hold 1: hold
8	TX_EMPTY_CTRL	R/W	0x0	This bit controls the generation of the TX_EMPTY interrupt; please refer to the I2C_RAW_INTR_SR register for details.
7	STOP_DET_IFADDRESSED	R/W	0x0	Generation of STOP_DET interrupt in slave mode: 1: STOP_DET interrupt is generated only when there is an address match.

Bit	Name	Attribute	Reset Value	Description
				0: STOP_DET interrupt is generated regardless of whether there is an address match. This bit is applicable only in slave mode. Note: When the broadcast address is addressed, if this bit is set, the slave will not generate a STOP_DET interrupt. The STOP_DET interrupt is generated only when the sent address matches the slave address.
6	SLAVE_DISABLE	R/W	0x1	I2C slave enable: 0: slave enabled 1: slave disabled
5	RESTART_EN	R/W	0x1	In master mode, this bit controls whether to send a RESTART condition: 0: disabled 1: enabled When RESTART is disabled, the I2C interface as a master cannot perform the following functions: Sending a start byte Changing the transfer direction in combined format mode Read operation in 10-bit addressing format After replacing the RESTART condition, sending a stop condition followed by a start condition will result in splitting into multiple I2C transfers. If any of the above operations are executed, it will set bit 6 (TX_ABRT) of the

Bit	Name	Attribute	Reset Value	Description
				IC_RAW_INTR_STAT register.
4	ADDR10_MASTER	R	0x0	Addressing mode when I2C operates as a master: 0: 7-bit addressing mode 1: 10-bit addressing mode
3	ADDR10_SLAVE	R/W	0x0	In slave mode, this bit controls the response to 10-bit or 7-bit addressing: 0: 7-bit addressing; The I2C ignores the 10-bit addressing. For 7-bit addressing, only the lower 7 bits of the IC_SAR register are compared. 1: 10-bit addressing; The I2C only responds to 10-bit addressing, comparing the received address with the 10 bits of IC_SAR.
2:1	SPEED	R/W	0x2	These two bits control the operating speed mode of the I2C: This setting is only valid when the I2C is in master mode. 1: standard mode (0–100 Kb/s) 2: fast mode (100 Kb/s–400 Kb/s) or fast mode+ (400 Kb/s–1Mb/s)
0	MASTER_MODE	R/W	0x1	Master mode enable: 0: disabled 1: enabled

SLAVE_DISABLE(I2C_CR[6]) and MASTER_MODE(I2C_CR[0]) are configured as show in the following table.

Table 26-5: I2C Slave and Master Configuration Table

SLAVE_DISABLE (I2C_CR[6])	MASTER_MODE (I2C_CR[0])	State
0	0	Slave
0	1	Setting error
1	0	Setting error
1	1	Master

26.5.2 I2C Slave Address Access Register (I2C_TAR)

Offset address: 0x04

Reset value: 0x0000 0055

Bit	Name	Attribute	Reset Value	Description
31:17	RSV	-	-	Reserved
16	SMBUS_QUICK_CMD	R/W	0x0	When bit[11 (SPECIAL) is set to 1, this bit indicates whether the I2C executes the QUICK command.
15:13	RSV	-	-	Reserved
12	BITADDR10_MASTER	R/W	0x0	When the I2C operates as a master, it starts the transmission in either 7-bit or 10-bit addressing mode: 0: 7-bit addressing 1: 10-bit addressing
11	SPECIAL	R/W	0x0	This bit indicates whether the software is executing a special command (broadcast call or start byte command): 0: Ignore the 10th bit GC, normally use the ADDR bit. 1: Execute special I2C commands as described by the GC bit.
10	GC_OR_START	R/W	0x0	If bit 11 is set, this bit indicates whether the I2C is executing a broadcast call or a start byte command: 0: broadcast call address; only write operations can be performed when sending a broadcast call address. The

Bit	Name	Attribute	Reset Value	Description
				I2C interface remains in broadcast addressing mode until the value of SPECIAL (bit 11) is cleared. 1: start byte command
9:0	ADDR	R/W	0x55	Destination address of the main operation These bits can be ignored when sending a broadcast address. To generate a start byte command, the CPU only needs to write to these bits once.

26.5.3 I2C Slave Address Register (I2C_SAR)

Offset address: 0x08

Reset value: 0x0000 0055

Bit	Name	Attribute	Reset Value	Description
31:10	RSV	-	-	Reserved
9:0	ADDR	R/W	0x55	When the I2C is in slave mode, the slave address is saved. For 7-bit addressing mode, ADDR[6:0] is valid.

26.5.4 I2C Data Command Register (I2C_DATACMD)

Offset address: 0x10

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:12	RSV	-	-	Reserved
11	FIRST_DATA_BYTE	R	0x0	Whether it is the first data byte received after the address phase in master receiver or slave receiver mode: 1: yes 0: no
10	RESTART	W	0x0	Generation of RESTART signal before transmitting or receiving data:

Bit	Name	Attribute	Reset Value	Description
				1: If the RESTART_EN signal is “1”, a RESTART signal is generated before data is received or transmitted (depending on the value of CMD), regardless of whether the previous command changes the direction of data transmission. If the RESTART_EN signal is “0”, a START signal immediately follows the STOP signal. 0: If the RESTART_EN signal is “1”, a RESTART signal is generated only when the previous command changes the direction of data transmission. If the RESTART_EN signal is “0”, a START signal immediately follows the STOP signal.
9	STOP	W	0x0	Generation of STOP signal after transmitting or receiving data: 1: A STOP signal is generated after the current byte, regardless of whether the TXFIFO is empty. If the TXFIFO is not empty, the master immediately issues a new transfer and bus arbitration signal. 0: A STOP signal is not generated after the current byte, regardless of whether the TXFIFO is empty. The master continues the current transmission (transmitting or receiving data based on the value of CMD). If the TXFIFO is empty, the master will pull SCL low to suspend the bus until TXFIFO receives new data.
8	CMD	W	0x0	Performing read or write operation in master mode: 1: read 0: write This bit is used to distinguish between

Bit	Name	Attribute	Reset Value	Description
				read and write commands when a command is entered into TXFIFO. In RX mode, the bit write operation is ignored. In TX mode, 0 indicates that the data in DATA has been transmitted.
7:0	DATA	R/W	0x0	Data to be transmitted or received on the I2C bus: If a write operation is performed on DATA followed by a read, the read operation is invalid. However, when reading this register, the returned value is the data received by the I2C.

26.5.5 I2C Standard Mode SCL High-level Configuration Register (I2C_SSSCLHCNT)

Offset address: 0x14

Reset value: 0x0000 0028

Bit	Name	Attribute	Reset Value	Description
31:16	RSV	-	-	Reserved
15:0	SS_SCL_HCNT	R/W	0x28	SCL clock high-level cycle in I2C standard mode Note: This register is configurable between 6 and 65525, because the I2C interface uses a 16-bit counter whose value is equal to I2C_SSSCLHCNT + 10, indicating that the I2C bus is in an idle state.

26.5.6 I2C Standard Mode SCL Low-level Configuration Register (I2C_SSSCLLCNT)

Offset address: 0x18

Reset value: 0x0000 002F

Bit	Name	Attribute	Reset Value	Description
31:16	RSV	-	-	Reserved
15:0	SS_SCL_LCNT	R/W	0x2F	SCL clock low-level cycle in I2C standard mode The minimum value is 8.

26.5.7 I2C Fast Mode SCL High-level Configuration Register (I2C_FSSCLHCNT)

Offset address: 0x1C

Reset value: 0x0000 0006

Bit	Name	Attribute	Reset Value	Description
31:16	RSV	-	-	Reserved
15:0	FS_SCL_HCNT	R/W	0x6	SCL clock high-level cycle in I2C fast mode The minimum value is 6.

26.5.8 I2C Fast Mode SCL Low level Configuration Register (I2C_FSSCLLCNT)

Offset address: 0x20

Reset value: 0x0000 000D

Bit	Name	Attribute	Reset Value	Description
31:16	RSV	-	-	Reserved
15:0	FS_SCL_LCNT	R/W	0xD	SCL clock low-level cycle in I2C fast mode The minimum value is 8.

26.5.9 I2C Interrupt Status Register (I2C_ISR)

Offset address: 0x2C

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:15	RSV	-	-	Reserved
14	SCL_STUCK_AT_LOW	R	0x0	SCL pull-low interrupt status: 0: no interrupt 1: interrupt occurred
13	MST_ON_HOLD	R	0x0	Master bus holding interrupt status: 0: no interrupt 1: interrupt occurred
12	RESTART_DET	R	0x0	Restart detection interrupt status: 0: no interrupt 1: interrupt occurred
11	GEN_CALL	R	0x0	General call request received interrupt status: 0: no interrupt 1: interrupt occurred
10	START_DET	R	0x0	Start detection interrupt status, indicating whether the I2C bus interface has detected a START or RESTART condition: 0: no interrupt 1: interrupt occurred
9	STOP_DET	R	0x0	Stop detection interrupt status, indicating whether the I2C bus interface has detected a STOP condition: 0: no interrupt 1: interrupt occurred
8	ACTIVITY	R	0x0	Activity interrupt status, recording the I2C activity status until cleared: 0: no interrupt 1: interrupt occurred
7	RX_DONE	R	0x0	Reception complete interrupt status: 0: no interrupt 1: interrupt occurred
6	TX_ABRT	R	0x0	Transmission abort interrupt status: 0: no interrupt 1: interrupt occurred
5	RD_REQ	R	0x0	In slave mode, interrupt status for

Bit	Name	Attribute	Reset Value	Description
				other masters attempting to read data from the I2C interface: 0: no interrupt 1: interrupt occurred
4	TX_EMPTY	R	0x0	Interrupt status of TX FIFO reaching or falling below the threshold: 0: no interrupt 1: interrupt occurred
3	TX_OVER	R	0x0	TX FIFO overflow interrupt status: 0: no interrupt 1: interrupt occurred
2	RX_FULL	R	0x0	Interrupt status of RX FIFO reaching or exceeding the threshold: 0: no interrupt 1: interrupt occurred
1	RX_OVER	R	0x0	RX FIFO overflow interrupt status: 0: no interrupt 1: interrupt occurred
0	RX_UNDER	R	0x0	Data read overflow (i.e., CPU reading from FIFO when RX_FIFO is empty) interrupt status: 0: no interrupt 1: interrupt occurred

26.5.10 I2C Interrupt Mask Register (I2C_INTMASK)

Offset address: 0x30

Reset value: 0x0000 48FF

Bit	Name	Attribute	Reset Value	Description
31:15	RSV	-	-	Reserved
14	SCL_STUCK_AT_LOW	R/W	0x1	SCL pull-low interrupt mask: 0: masked 1: unmasked
13	MST_ON_HOLD	R/W	0x0	Master bus holding interrupt mask: 0: masked

Bit	Name	Attribute	Reset Value	Description
				1: unmasked
12	RESTART_DET	R/W	0x0	Restart detection interrupt mask: 0: masked 1: unmasked
11	GEN_CALL	R/W	0x1	General call request received interrupt mask: 0: masked 1: unmasked
10	START_DET	R/W	0x0	Start detection interrupt mask: 0: masked 1: unmasked
9	STOP_DET	R/W	0x0	Stop detection interrupt mask: 0: masked 1: unmasked
8	ACTIVITY	R/W	0x0	Activity interrupt mask: 0: masked 1: unmasked
7	RX_DONE	R/W	0x1	Reception complete interrupt mask: 0: masked 1: unmasked
6	TX_ABRT	R/W	0x1	Transmission abort interrupt mask: 0: masked 1: unmasked
5	RD_REQ	R/W	0x1	As a slave, interrupt mask of another I2C master requesting data reading: 0: masked 1: unmasked
4	TX_EMPTY	R/W	0x1	Interrupt mask of TX FIFO reaching or falling below the threshold: 0: masked 1: unmasked
3	TX_OVER	R/W	0x1	TX FIFO overflow interrupt mask: 0: masked 1: unmasked
2	RX_FULL	R/W	0x1	Interrupt mask of RX FIFO reaching or exceeding the threshold:

Bit	Name	Attribute	Reset Value	Description
				0: masked 1: unmasked
1	RX_OVER	R/W	0x1	RX FIFO overflow interrupt mask: 0: masked 1: unmasked
0	RX_UNDER	R/W	0x1	Data read overflow (i.e., CPU reading from FIFO when RX FIFO is empty) interrupt mask: 0: masked 1: unmasked

26.5.11 I2C Raw Interrupt Register (I2C_RAWISR)

Offset address: 0x34

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:15	RSV	-	-	Reserved
14	SCL_STUCK_AT_LOW	R	0x0	This bit indicating whether SCL remains low for I2C_clk cycles set in I2C_SCL_STUCK_AT_LOW_TIMOUT register. This bit in valid only when BUS_CLEAR_FEATURE_CTRL (I2C_CR[11]) = 1.
13	MST_ON_HOLD	R	0x0	This bit indicating whether the master holds the bus and the TX FIFO is empty.
12	RESTART_DET	R	0x0	This bit indicating whether the RESTART condition occurs in slave mode with the slave addressed.
11	GEN_CALL	R	0x0	Broadcast call: This bit is set when broadcast call address is received. It is cleared when the I2C interface is disabled or when the CPU reads the GC register. I2C will store the received data in receive buffer.

Bit	Name	Attribute	Reset Value	Description
10	START_DET	R	0x0	Start condition detection: Regardless of whether the I2C interface is operating in master or slave mode, this bit is set once a start or repeated start condition is detected on the I2C interface.
9	STOP_DET	R	0x0	Stop condition detection: Slave mode: ● If STOP_DET_IFADDRESSED (IC_CR[7]) = 1'b1, the STOP_DET interrupt is generated only when the slave is addressed. ● If STOP_DET_IFADDRESSED (IC_CR[7]) = 1'b0, the STOP_DET interrupt is generated regardless of whether the slave is addressed or not. Master mode: ● If STOP_DET_IF_MASTER_ACTIVE (IC_CR[10]) = 1'b1, the STOP_DET interrupt is generated only when the master is in an active state. ● If STOP_DET_IF_MASTER_ACTIVE (IC_CR[10]) = 1'b0, the STOP_DET interrupt is generated regardless of whether the master is in an active state or not.
8	ACTIVITY	R	0x0	I2C interface activation: this bit is used to capture the active state of the I2C module. Once set, it can only be cleared by one of the following four methods: ● Disabling the I2C interface ● Reading the I2C_CLR_ACTIVITY register ● Reading the I2C_CLR register ● System reset Once set, it can only be cleared by the above methods, even if the I2C is idle, this bit will remain high until cleared.

Bit	Name	Attribute	Reset Value	Description
7	RX_DONE	R	0x0	Slave transmission complete: When the I2C operates as a slave transmitter, this bit will be set if the master does not respond after transmitting a byte of data. This happens after the last byte of transmission, indicating the end of transmission.
6	TX_ABRT	R	0x0	Transmission abort: When the I2C interface is transmitting data, this bit is set if the data in the TX_FIFO cannot be completely sent out. Note: When any event in the I2C_TXABRTSRC register causes the transmission to be aborted, the I2C will flush/reset/clear the TX_FIFO. Once the I2C_CLRTXABRT register is read, the TX_FIFO will be able to receive new data from the APB bus.
5	RD_REQ	R	0x0	Read request: When the I2C operates as a slave, this bit is set if another master attempts to read data from the I2C interface. The I2C interface will keep the bus waiting (SCL = 0) until the interrupt is processed. This means that the I2C interface as a slave has been successfully addressed by another master and is required to transmit data. The processor must respond to this interrupt and then write data to the I2C_DATA_CMD register. This bit is set when the processor reads the RD_REQ register.
4	TX_EMPTY	R	0x0	Transmit buffer empty: The state of this bit depends on the TX_EMPTY_CTRL in the I2C_CR register: When TX_EMPTY_CTRL = 0, this bit is set if

Bit	Name	Attribute	Reset Value	Description
				the transmit buffer is less than or equal to the value set in the I2C_TXTL register. When TX_EMPTY_CTRL = 1, this bit is set if the transmit buffer is less than or equal to the value set in the I2C_TXTL register and the internal shift register is empty. This bit is automatically cleared by hardware when the transmit buffer is not empty.
3	TX_OVER	R	0x0	Transmit buffer overrun: This bit is set when the transmit buffer is full and the processor writes new data resulting in an overflow.
2	RX_FULL	R	0x0	Receive buffer full: This bit is set when the receive buffer reaches or exceeds the threshold in the IC_RXTL register. This bit is cleared by hardware when the receive buffer is below the threshold.
1	RX_OVER	R	0x0	Receive buffer overrun: This bit is set when the receive buffer is full and new data is received. In this case, the I2C interface will respond, but the new data will be lost.
0	RX_UNDER	R	0x0	Receive buffer underrun: This bit is set when the RX FIFO is empty and the I2C_DATA_CMD register is read.

26.5.12 I2C RX FIFO Threshold Register (I2C_RXTL)

Offset address: 0x38

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:1	RSV	-	-	Reserved

Bit	Name	Attribute	Reset Value	Description
0	RX_TL	R/W	0x0	RX FIFO threshold: Control the trigger of RX_FULL interrupt.

26.5.13 I2C TX FIFO Threshold Register (I2C_TXTL)

Offset address: 0x3C

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:1	RSV	-	-	Reserved
0	TX_TL	R/W	0x0	TX FIFO threshold: Control the trigger of TX_EMPTY interrupt.

26.5.14 I2C Combined and Independent Interrupt Clear Register (I2C_CLR)

Offset address: 0x40

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:1	RSV	-	-	Reserved
0	CLR_INTR	R	0x0	Reading this register will clear all combined interrupts, independent interrupts and the I2C_TXABRTSRC register. This bit does not clear the interrupts that can be cleared automatically by hardware, but clear only the interrupts that can be cleared by software.

26.5.15 I2C RX_UNDER Interrupt Clear Register (I2C_CLRRXUNDER)

Offset address: 0x44

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description

Bit	Name	Attribute	Reset Value	Description
31:1	RSV	-	-	Reserved
0	CLR_RX_UNDER	R	0x0	Reading this register clears the RX_UNDER interrupt (I2C_RAWISR[0]).

26.5.16 I2C RX_OVER Interrupt Clear Register (I2C_CLRRXOVER)

Offset address: 0x48

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:1	RSV	-	-	Reserved
0	CLR_RX_OVER	R	0x0	Reading this register clears the RX_OVER interrupt (I2C_RAWISR[1]).

26.5.17 I2C TX_OVER Interrupt Clear Register (I2C_CLRTXOVER)

Offset address: 0x4C

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:1	RSV	-	-	Reserved
0	CLR_TX_OVER	R	0x0	Reading this register clears the RX_OVER interrupt (I2C_RAWISR[3]).

26.5.18 I2C RD_REQ Interrupt Clear Register (I2C_CLRRDREQ)

Offset address: 0x50

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:1	RSV	-	-	Reserved
0	CLR_RD_REQ	R	0x0	Reading this register clears the RD_REQ interrupt (I2C_RAWISR[5]).

26.5.19 I2C TX_ABRT Interrupt Clear Register (I2C_CLRTXABRT)

Offset address: 0x54

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:1	RSV	-	-	Reserved
0	CLR_TX_ABRT	R	0x0	Reading this register clears the TX_ABRT interrupt (I2C_RAWISR[6]) and the I2C_TXABRTSRC register. It also releases the TX FIFO from the refresh/reset state in order to receive the written data.

26.5.20 I2C RX_DONE Interrupt Clear Register (I2C_CLRRXDONE)

Offset address: 0x58

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:1	RSV	-	-	Reserved
0	CLR_RX_DONE	R	0x0	Reading this register clears the RX_DONE interrupt (I2C_RAWISR[7]).

26.5.21 I2C ACTIVITY Interrupt Clear Register (I2C_CLRACTIVITY)

Offset address: 0x5C

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:1	RSV	-	-	Reserved

Bit	Name	Attribute	Reset Value	Description
0	CLR_ACTIVITY	R	0x0	If the I2C bus is inactive, read this register to clear the ACTIVITY interrupt (I2C_RAWISR[8]). If the I2C is still active, then the ACTIV interrupt will remain set. This bit is cleared by hardware when the I2C module is disabled or when the I2C bus is no longer active. The status of ACTIVITY in I2C_RAWISR can be obtained by reading this register.

26.5.22 I2C STOP_DET Interrupt Clear Register (I2C_CLRSTOPDET)

Offset address: 0x60

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:1	RSV	-	-	Reserved
0	CLR_STOP_DET	R	0x0	Reading this register clears the STOP interrupt (I2C_RAWISR[9]).

26.5.23 I2C START_DET Interrupt Clear Register (I2C_CLRSTARTDET)

Offset address: 0x64

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:1	RSV	-	-	Reserved
0	CLR_START_DET	R	0x0	Reading this register clears the START interrupt (I2C_RAWISR[10]).

26.5.24 I2C GEN_CALL Interrupt Clear Register (I2C_CLRGENCALL)

Offset address: 0x68

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:1	RSV	-	-	Reserved
0	CLR_GEN_CALL	R	0x0	Reading this register clears the GC interrupt (I2C_RAWISR[11]).

26.5.25 I2C Enable Register (I2C_EN)

Offset address: 0x6C

Reset value: 0x0000 0004

Bit	Name	Attribute	Reset Value	Description
31:17	RSV	-	-	Reserved
16	SMBUS_CLK_RESET	R/W	0x0	Release SMBus master clock reset in SMBus master mode
15:4	RSV	-	-	Reserved
3	SDA_STUCK_RECOVERY_ENABLE	R/W	0x0	This bit is used as part of the SDA recovery mechanism if SDA is pulled low at the TX_ABORT interrupt. This bit will be automatically cleared.
2	TX_CMD_BLOCK	R/W	0x1	In master mode: 0: Once the first data in the TX FIFO is available, I2C will automatically start transmitting data. 1: Even if there is data to be transmitted in the TX FIFO, the I2C data transmission will be prevented.
1	ABORT	R/W	0x0	I2C transmission abort: 0: abort did not occur or has already ended. 1: abort operation is in progress. When the I2C module is set to operate as a master, the I2C transmission can be aborted via software. Once set, it cannot be cleared immediately. After being

Bit	Name	Attribute	Reset Value	Description
				set, the I2C module control logic will generate a STOP condition and clear the transmit buffer after completing the current transmission, and a TX_ABRT interrupt will be generated after the abort operation. The ABORT bit will be cleared automatically after the abort operation.
0	ENABLE	R/W	0x0	I2C module enable: 0: I2C module disabled (transmit and receive buffer remain empty) 1: I2C module enabled

26.5.26 I2C Status Register (I2C_SR)

Offset address: 0x70

Reset value: 0x0000 0006

Bit	Name	Attribute	Reset Value	Description
31:19	RSV	-	-	Reserved
18	SMBUS_SLAVE_ADDR_RESOLVED	R	0x0	This bit is set when the slave address (I2C_SAR[6:0]) is resolved by the ARP master.
17	SMBUS_SLAVE_ADDR_VALID	R	0x0	This bit is set when the slave address (I2C_SAR[6:0]) is valid.
16	SMBUS_QUICK_CMD	R	0x0	This bit is set when the R/W bit of the QUICK command is received.
15:12	RSV	-	-	Reserved
11	SDA_STUCK_NOT_RECOVERED	R	0x0	This bit is set when SDA is still pulled low after the recovery mechanism.
10	SLV_HOLD_RX_FIFO_FULL	R	0x0	RX FIFO is full and additional bytes are received, and the bus remains in slave mode.
9	SLV_HOLD_TX_FI	R	0x0	TX FIFO is empty and the bus

Bit	Name	Attribute	Reset Value	Description
	FO_EMPTY			remains in slave mode during a read request. The bus is held until the data to be transmitted is read from TX FIFO.
8	MST_HOLD_RX_FIFO_FULL	R	0x0	RX FIFO is full and additional bytes are received, and the bus remains in master mode.
7	MST_HOLD_TX_FIFO_EMPTY	R	0x0	When the TX FIFO is empty and the stop bit is not set for the last byte, the master pauses the write transfer. When the master holds the bus because the TX FIFO is empty, this bit indicates that the bus is held and the previously transmitted command does not set the stop bit.
6	SLV_ACTIVITY	R	0x0	Slave state machine active status bit: 0: The slave state machine is in IDLE state, so the I2C slave part is inactive. 1: The slave state machine is not in IDLE state, so the I2C slave part is active.
5	MST_ACTIVITY	R	0x0	Master state machine active status bit: 0: The master state machine is in IDLE state, so the I2C master part is inactive. 1: The master state machine is not in IDLE state, so the I2C master part is active.
4	RFF	R	0x0	Receive buffer full: 0: receive buffer not full 1: receive buffer full
3	RFNE	R	0x0	Receive buffer not empty: 0: receive buffer empty

Bit	Name	Attribute	Reset Value	Description
				1: receive buffer not empty
2	TFE	R	0x1	Transmit buffer empty: 0: transmit buffer not empty 1: transmit buffer empty
1	TFNF	R	0x1	Transmit buffer not full: 0: transmit buffer full 1: transmit buffer not full
0	ACTIVITY	R	0x0	I2C bit activity status: the result of the OR of MST_ACTIVITY bit and SLV_ACTIVITY bit.

26.5.27 I2C Transmit Buffer Depth Register (I2C_TXFLR)

Offset address: 0x74

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
15:2	RSV	-	-	Reserved
1:0	CNT	R	0x0	Number of valid data in transmit buffer (0–2)

26.5.28 I2C Receive Buffer Depth Register (I2C_RXFLR)

Offset address: 0x78

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
15:2	RSV	-	-	Reserved
1:0	CNT	R	0x0	Number of valid data in receive buffer (0–2)

26.5.29 I2C SDA Hold Time Register (I2C_SDAHOLD)

Offset address: 0x7C

Reset value: 0x0000 0001

Bit	Name	Attribute	Reset Value	Description
31:24	RSV	-	-	Reserved

Bit	Name	Attribute	Reset Value	Description
23:16	RX_HOLD	R/W	0x0	When I2C is in receive mode, the SDA hold time is measured in units of APB1 clock cycles.
15:0	TX_HOLD	R/W	0x1	When I2C is in transmit mode, the SDA hold time is measured in units of APB1 clock cycles.

26.5.30 I2C Transmit Fail Interrupt Source Register (I2C_TXABRTSRC)

Offset address: 0x80

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:25	RSV	-	-	Reserved
24:23	TX_FLUSH_CNT	R	0x0	Represent the number of data in TX FIFO that are refreshed due to the TX_ABRT interrupt.
22:18	RSV	-	-	Reserved
17	ABRT_SDA_STUCK_AT_LOW	R	0x0	Master mode only The master detects that SDA has been pulled low.
16	ABRT_USER_ABRT	R	0x0	Master mode only The master detects that the transmission has been aborted.
15	ABRT_SLVRD_INTX	R	0x0	The value is set when the processor responds to a request from the slave mode to transfer data to the master, and the user sets I2C_data_CMD[8] (CMD) to 1.
14	ABRT_SLV_ARBLOST	R	0x0	The slave loses the bus while transmitting data to the master.
13	ABRT_SLVFLUSH_TXFIFO	R	0x0	The slave receives a read command, and then generates a TX_ABRT interrupt to refresh the values in the TX FIFO.
12	ARB_LOST	R	0x0	Master arbitration loss

Bit	Name	Attribute	Reset Value	Description
11	ABRT_MASTER_DIS	R	0x0	The user attempts to initialize master operation while the master mode is disabled.
10	ABRT_10B_RD_NORSTRT	R	0x0	Restart is disabled (I2C_CR[5] = 0), and the master sends a read command in 10-bit addressing mode.
9	ABRT_SBYTE_NORSTRT	R	0x0	Restart is disabled (I2C_CR[5] = 0), and the user attempts to send a start byte. To clear this bit, the source of ABRT_SYTE_NORSTRT must first be cleared. The Restart bit (IC_CON[5] = 1) must be enabled, or the SPECIAL bit (IC_TAR[11]) must be cleared, or the GC_or_START bit (IC_TAR[10]) must be cleared.
8	RSV	-	-	Reserved
7	ABRT_SBYTE_ACKDET	R	0x0	The master sends a start byte and receives an ACK (incorrect behavior).
6	RSV	-	-	Reserved
5	ABRT_GCALL_READ	R	0x0	The I2C master sends a broadcast and reads the post-broadcast data over the bus (I2C_DATA_CMD[9] = 1).
4	ABRT_GCALL_NOACK	R	0x0	The I2C master sends a broadcast, but the slave does not respond with an ACK.
3	ABRT_TXDATA_NOACK	R	0x0	Master mode only The master has received an ACK for the address phase but does not receive an ACK from the slave for the immediately sent data.
2	ABRT_10ADDR2_NOACK	R	0x0	The master does not receive an ACK from the slave after sending the second 10-bit address in 10-bit addressing mode.
1	ABRT_10ADDR1_NOACK	R	0x0	The master does not receive an ACK from the slave after sending the first 10-bit address in 10-bit addressing

Bit	Name	Attribute	Reset Value	Description
				mode.
0	ABRT_7B_ADDR_NOACK	R	0x0	The master does not receive an ACK from the slave after sending the address in 7-bit addressing mode.

26.5.31 I2C Slave Data NACK Generation Register (I2C_SLVDATANACKONLY)

Offset address: 0x84

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:1	RSV	-	-	Reserved
0	NACK	R/W	0x0	I2C as slave receiver: 0: normally generate NACK and ACK 1: generate NACK after receiving data

26.5.32 I2C DMA Control Register (I2C_DMACR)

Offset address: 0x88

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
15:2	RSV	-	-	Reserved
1	TDMAE	R/W	0x0	Transmit DMA enable: 0: transmit DMA disabled 1: transmit DMA enabled
0	RDMAE	R/W	0x0	Receive DMA enable: 0: receive DMA disabled 1: receive DMA enabled

26.5.33 I2C DMA Transmit Data Level Register (I2C_DMATDLR)

Offset address: 0x8C

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description

Bit	Name	Attribute	Reset Value	Description
15:1	RSV	-	-	Reserved
0	DMATDL	R/W	0x0	When TDMAE = 1, dma_tx_req is generated when the data in TX FIFO is equal to or less than the value of DMATDL.

26.5.34 I2C DMA Receive Data Level Register (I2C_DMARDLR)

Offset address: 0x90

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
15:1	RSV	-	-	Reserved
0	DMARDL	R/W	0x0	When RDMAE = 1, dma_rx_req is generated when the data in RX FIFO is equal to or greater than DMARDL + 1.

26.5.35 I2C SDA Setup Time Register (I2C_SDASETUP)

Offset address: 0x94

Reset value: 0x0000 0064

Bit	Name	Attribute	Reset Value	Description
31:8	RSV	-	-	Reserved
7:0	SDA_SETUP	R/W	0x64	SDA setup time If the required setup time for the SDA line is 1000 ns, and the APB1 clock frequency is 10 MHz, it is recommended that this register be set to 11. The minimum value of this register is 2.

26.5.36 I2C Broadcast Call ACK Register (I2C_ACKGENERALCALL)

Offset address: 0x98

Reset value: 0x0000 0001

Bit	Name	Attribute	Reset Value	Description
31:1	RSV	-	-	Reserved

Bit	Name	Attribute	Reset Value	Description
0	ACK_GEN_CALL	R/W	0x1	Broadcast call ACK: 1: Respond with ACK after receiving a broadcast call. 0: Does not respond after receiving a broadcast call, and does not generate an interrupt.

26.5.37 I2C Enable Status Register (I2C_ENSR)

Offset address: 0x9C

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:3	RSV	-	-	Reserved
2	SLV_RX_DATA_LOST	R	0x0	Slave data loss: 1: The I2C transmission has been terminated. Although the data byte responded with NACK, it has entered the I2C data transmission phase. 0: The I2C has been disabled, and the slave reception has not entered the data transfer phase.
1	SLV_DISABLED_WHILE_BUSY	R	0x0	The slave is disabled while busy.
0	IC_EN	R	0x0	I2C enable status: 0: disabled 1: enabled

26.5.38 I2C Spike Suppression Limit Register (2C_FSSPKLEN)

Offset address: 0xA0

Reset value: 0x0000 0001

Bit	Name	Attribute	Reset Value	Description
31:8	RSV	-	-	Reserved

Bit	Name	Attribute	Reset Value	Description
7:0	I2C_FS_SPKLEN	R/W	0x1	Before performing any I2C bus transactions, this register must be configured to ensure stable operation. Writing is valid only when I2C_EN[0] = 0.

26.5.39 I2C RESTART_DET Interrupt Clear Register (I2C_CLRRD)

Offset address: 0xA8

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:1	RSV	-	-	Reserved
0	CLR_RESTART_DET	R	0x0	Reading this register clears the RESTART interrupt (I2C_RAW_ISR[10]).

26.5.40 I2C SCL Pull-low Timeout Register (I2C_SCLSALT)

Offset address: 0xAC

Reset value: 0xFFFF FFFF

Bit	Name	Attribute	Reset Value	Description
31:0	I2C_SCL_STUCK_LOW_TIME OUT	R/W	0xFFFFFFFF	An interrupt is generated when SCL is detected to be held low for I2C_clk cycles specified by this register.

26.5.41 I2C SDA Pull-low Timeout Register (I2C_SDASALT)

Offset address: 0xB0

Reset value: 0xFFFF FFFF

Bit	Name	Attribute	Reset Value	Description
31:0	IC_SDA_STUCK_LOW_TIME OUT	R/W	0xFFFFFFFF	Provided that Sda_stuck_recovery_en (bit 3 of I2C_ENABLE) is enabled, when SDA is detected to be held low for I2C_clk cycles specified by this register, the recovery of the SDA line is initiated.

26.5.42 I2C SCL Pull-low Detection Interrupt Clear Register (I2C_CLRSSD)

Offset address: 0xB4

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:1	RSV	-	-	Reserved
0	CLR_SCL_STUCK	R	0x0	Reading this register clears the SCL pull-low detection interrupt (I2C_RAW_ISR[14]).

26.5.43 I2C SMBUS Slave Clock Extension Timeout Register (I2C_SMBUSCLOCKLOWSEXT)

Offset address: 0xBC

Reset value: 0xFFFF FFFF

Bit	Name	Attribute	Reset Value	Description
31:0	SMBUS_CLK_LOW_SEXT_TIMEOUT	R/W	0xFFFFFFFF	Used to detect the slave clock extension timeout in master mode, where the slave extends the time from start to stop.

26.5.44 I2C SMBUS Master Clock Extension Timeout Register (I2C_SMBUSCLOCKLOWMEXT)

Offset address: 0xC0

Reset value: 0xFFFF FFFF

Bit	Name	Attribute	Reset Value	Description
31:0	SMBUS_CLK_LOW_MEXT_TIMEOUT	R/W	0xFFFFFFFF	Used to detect the timeout for the extension of SMBus clock (SCL) from start to ACK, ACK to ACK, or ACK to stop in master mode.

26.5.45 I2C SMBUS Maximum Bus Idle Count Register (I2C_SMBUSTHIGHMAXIDLECOUNT)

Offset address: 0xC4

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:16	RSV	-	-	Reserved
15:0	SMBUS_THIGH_MAX_BUS_IDLE_CNT	R/W	0xFFFF	Used to set the required bus idle time period when a master device has been dynamically added to the bus and the state transitions on the SMBCLK or SMBDAT lines may not have been detected yet.

26.5.46 I2C SMBus Interrupt Status Register (I2C_SMBUSISR)

Offset address: 0xC8

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:9	RSV	-	-	Reserved
8	R_SLV_RX_PEC_NACK	R	0x0	Interrupt status for the slave generating NACK for the PEC byte from the slave ARP command: 0: no interrupt 1: interrupt occurred
7	R_ARP_ASSGN_ADDR_CMD_DET	R	0x0	Interrupt status for receiving an Assign Address ARP: 0: no interrupt 1: interrupt occurred
6	R_ARP_GET_UDID_CMD_DET	R	0x0	Interrupt status for receiving a general or directed Get UDID ARP command: 0: no interrupt 1: interrupt occurred
5	R_ARP_RST_CMD_DET	R	0x0	Interrupt status for receiving a general or directed Reset ARP

Bit	Name	Attribute	Reset Value	Description
				command: 0: no interrupt 1: interrupt occurred
4	R_ARP_PREPARE_CMD_DET	R	0x0	Interrupt status for receiving a Prepare to ARP command: 0: no interrupt 1: interrupt occurred
3	R_HOST_NOTIFY_MST_DET	R	0x0	Interrupt status for receiving a Host Notify command: 0: no interrupt 1: interrupt occurred
2	R_QUICK_CMD_DET	R	0x0	Interrupt status for receiving a Quick command: 0: no interrupt 1: interrupt occurred
1	R_MST_CLOCK_EXTND_TIMEOUT	R	0x0	Interrupt status for master transactions from start to stop (START to ACK, ACK to ACK, or ACK to STOP) exceeding I2C_SMBUS_CLOCK_LOW_MEXT time for each byte in the message: 0: no interrupt 1: interrupt occurred
0	R_SLV_CLOCK_EXTND_TIMEOUT	R	0x0	Interrupt status for slave transactions (start to stop) exceeding IC_SMBUS_CLOCK_LOW_SEXT time: 0: no interrupt 1: interrupt occurred

26.5.47 I2C SMBus Interrupt Mask Register (I2C_SMBUSINTMASK)

Offset address: 0xCC

Reset value: 0x0000 01FF

Bit	Name	Attribute	Reset Value	Description
31:9	RSV	-	-	Reserved

Bit	Name	Attribute	Reset Value	Description
8	M_SLV_RX_PEC_NACK	R/W	0x1	Interrupt mask for the slave generating NACK for the PEC byte from the slave ARP command: 0: masked 1: unmasked
7	M_ARP_ASSGN_ADDM_CMD_DET	R/W	0x1	Interrupt mask for receiving an Assign Address ARP: 0: masked 1: unmasked
6	M_ARP_GET_UDID_CMD_DET	R/W	0x1	Interrupt mask for receiving a general or directed Get UDID ARP command: 0: masked 1: unmasked
5	M_ARP_RST_CMD_DET	R/W	0x1	Interrupt mask for receiving a general or directed Reset ARP command: 0: masked 1: unmasked
4	M_ARP_PREPARE_CMD_DET	R/W	0x1	Interrupt mask for receiving a Prepare to ARP command: 0: masked 1: unmasked
3	M_HOST_NOTIFY_MST_DET	R/W	0x1	Interrupt mask for receiving a Host Notify command: 0: masked 1: unmasked
2	M_QUICK_CMD_DET	R/W	0x1	Interrupt mask for receiving a Quick command: 0: masked 1: unmasked

Bit	Name	Attribute	Reset Value	Description
1	M_MST_CLOCK_EXTND_TIMEOUT	R/W	0x1	Interrupt mask for master transactions from start to stop (START to ACK, ACK to ACK, or ACK to STOP) exceeding I2C_SMBUS_CLOCK_LOW_MEXT time for each byte in the message: 0: masked 1: unmasked
0	M_SLV_CLOCK_EXTND_TIMEOUT	R/W	0x1	Interrupt mask for slave transactions (start to stop) exceeding IC_SMBUS_CLOCK_LOW_SEXT time: 0: masked 1: unmasked

26.5.48 I2C SMBus RAW Interrupt Status Register (I2C_SMBUSRAWISR)

Offset address: 0xD0

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:9	RSV	-	-	Reserved
8	SLV_RX_PEC_NACK	R	0x0	The slave generates a NACK for the PEC byte from the slave ARP command.
7	ARP_ASSGN_ADDCMD_DET	R	0x0	An Assign Address ARP command is received.
6	ARP_GET_UDID_CMD_DET	R	0x0	A general or directed Get UDID ARP command is received.
5	ARP_RST_CMD_DET	R	0x0	A general or directed Reset ARP command is received.
4	ARP_PREPARE_CMD_DET	R	0x0	A Prepare to ARP command is received.
3	HOST_NOTIFY_MST_DET	R	0x0	A Host Notify command is received.
2	QUICK_CMD_DET	R	0x0	A Quick command is received.
1	MST_CLOCK_EXTN	R	0x0	Master transactions from start to

Bit	Name	Attribute	Reset Value	Description
	D_TIMEOUT			stop (START to ACK, ACK to ACK, or ACK to STOP) exceed the I2C_SMBUS_CLOCK_LOW_MEXT time for each byte in the message. This bit is valid only when I2C_CR[0] = 1.
0	SLV_CLOCK_EXTND_TIMEOUT	R	0x0	Slave transactions (start to stop) exceed the IC_SMBUS_CLOCK_LOW_SEXT time. This bit is valid only when I2C_CR[0] = 1.

26.5.49 I2C SMBus Interrupt Status Clear Register (I2C_CLRSMBUSISR)

Offset address: 0xD4

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:9	RSV	-	-	Reserved
8	CLR_SLV_RX_PEC_NACK	W	0x0	Writing 1 to this bit clears the interrupt status for the slave generating NACK for the PEC byte from the slave ARP command.
7	CLR_ARP_ASSGN_ADDCLR_CMD_DET	W	0x0	Writing 1 to this bit clears the interrupt status for receiving an Assign Address ARP.
6	CLR_ARP_GET_UDID_CMD_DET	W	0x0	Writing 1 to this bit clears the interrupt status for receiving a general or directed Get UDID ARP command.
5	CLR_ARP_RST_CMD_DET	W	0x0	Writing 1 to this bit clears the interrupt status for receiving a general or directed Reset ARP command.
4	CLR_ARP_PREPAR	W	0x0	Writing 1 to this bit clears the

Bit	Name	Attribute	Reset Value	Description
	E_CMD_DET			interrupt status for receiving a Prepare to ARP command.
3	CLR_HOST_NOTIFY_MST_DET	W	0x0	Writing 1 to this bit clears the interrupt status for receiving a Host Notify command.
2	CLR_QUICK_CMD_DET	W	0x0	Writing 1 to this bit clears the interrupt status for receiving a Quick command.
1	CLR_MST_CLOCK_EXTND_TIMEOUT	W	0x0	Writing 1 to this bit clears the interrupt status for master transactions from start to stop (START to ACK, ACK to ACK, or ACK to STOP) exceeding I2C_SMBUS_CLOCK_LOW_MEXT time for each byte in the message.
0	CLR_SLV_CLOCK_EXTND_TIMEOUT	W	0x0	Writing 1 to this bit clears the interrupt status for slave transactions (start to stop) exceeding IC_SMBUS_CLOCK_LOW_SEXT time.

26.5.50 I2C SMBus ARP UDID LSB register (I2C_ SMBUSUDIDLSB)

Offset address: 0xDC

Reset value: 0xFFFF FFFF

Bit	Name	Attribute	Reset Value	Description
31:0	IC_SMBUS_ARP_UDID_LSB	R/W	0xFFFFFFFF	Under the address resolution protocol, the unique device identifier of LSB 32-bit slave is stored.

26.6 Operation Procedure

The interface can operate in one of the four following modes:

- Slave transmitter
- Slave receiver

- Master transmitter
- Master receiver

Note: The I2C interface module can only operate in either master mode or slave mode, but cannot work in both modes simultaneously. Therefore, it is important to ensure that the registers I2C_CR[6] and I2C_CR[0] cannot be set to 0 and 1 (or 1 and 0) respectively at the same time.

26.6.1 Master and Slave Initialization Configuration

Master initialization:

1. Configure the GPIO alternate function to enable the I2Cx clock in RCM module.
2. Set I2C_EN[0] to 0 to disable the I2C module.
3. Set I2C_CR[6] to 1 to disable the slave.
4. Configure I2C_CR[4] to set the master address format to 7-bit / 10-bit.
5. Configure I2C_CR[2:1] to set the I2C operating speed.
6. To ensure communication timing, the value of CNT shall be configured before I2C transmission. In standard mode, configure I2C_SSSCLHCNT[15:0] and I2C_SSSCLLCNT[15:0]. In fast mode, configure I2C_FSSCLHCNT[15:0] and I2C_FSSCLLCNT[15:0].
7. Configure I2C_RXTL[7:0] and I2C_TXTL[7:0] to set the thresholds of RX FIFO and TX FIFO, which are set to 0 here.
8. Set I2C_CR[0] to 1 to enable the master.
9. Set I2C_EN[0] to 1 to enable the I2C module.

Slave initialization:

1. Configure the GPIO alternate function to enable the I2Cx clock in RCM module.
2. Set I2C_EN[0] to 0 to disable the I2C module.
3. Configure I2C_CR[3] to set the slave address format to 7-bit / 10-bit.
4. Configure I2C_SAR[9:0] and write the slave address according to the set slave address format.
5. Set I2C_CR[6] to 0 to enable the slave.
6. Set I2C_CR[7] to 1 to generate a STOP_DET interrupt only when the address matches.
7. Set I2C_CR[0] to 0 to disable the master.
8. Configure I2C_RXTL[7:0] and I2C_TXTL[7:0] to set the thresholds of RX FIFO and TX FIFO, which are set to 0 here.
9. Set I2C_EN[0] to 1 to enable the I2C module.

26.6.2 Master Transmitter

1. Configure I2C_TAR[9:0] to set the destination address of the slave to be operated on.
2. Set I2C_DATACMD[8] to 0 to specify a write operation. At the same time, write the data to be transmitted into I2C_DATACMD[7:0].
3. After each operation on the I2C_DATACMD register, read I2C_RAWISR[4] until the read status value is 1, indicating that the data in the transmit buffer has been transmitted (the condition for setting I2C_RAWISR[4] is determined by I2C_TXTL[7:0]).
4. To transmit a single data byte, it is also required to set I2C_DATACMD[9] to 1 in step 2 to generate a STOP signal after transmitting the current byte.
5. To transmit multiple data bytes, repeat steps 2 and 3. When transmitting the last data byte, set I2C_DATACMD[8] to 0 to set the write operation, and set I2C_DATACMD[9] to 1

to generate a STOP signal after transmitting the current byte. At the same time, write the data to be transmitted into I2C_DATACMD[7:0].

26.6.3 Master Receiver

1. Configure I2C_TAR[9:0] to set the destination address of the slave to be operated on.
2. Set I2C_DATACMD[8] to 1 to specify a read operation.
3. Read I2C_RAWISR[2] until the read status value is 1, indicating that the receive buffer has received one data byte (the condition for setting I2C_RAWISR[2] is determined by I2C_RXTL[7:0]). At this point, read the data from I2C_DATACMD[7:0].
4. To receive a single data byte, it is also required to set I2C_DATACMD[9] to 1 in step 2 to generate a STOP signal after reading the current byte.
5. To receive multiple data bytes, repeat steps 2 and 3. When receiving the last data byte, set I2C_DATACMD[8] to 1 to set the read operation, and set I2C_DATACMD[9] to 1 to generate a STOP signal after reading the current byte.

26.6.4 Slave Transmitter

1. Read I2C_RAWISR[5] until the read status value is 1, indicating that the master is attempting to read data from this slave.
2. Set I2C_DATACMD[8] to 0 to specify a write operation. At the same time, write the data to be transmitted into I2C_DATACMD[7:0].
3. Read I2C_RAWISR[4] until the read status value is 1, indicating that the data in the transmit buffer has been transmitted (the condition for setting I2C_RAWISR[4] is determined by I2C_TXTL[7:0]).
4. To transmit multiple data bytes, repeat steps 2 and 3.

5. After transmitting all data, the RD_REQ state shall be cleared by setting I2C_CLRRDREQ[0].

26.6.5 Slave Receiver

1. Read I2C_RAWISR[2] until the read status value is 1, indicating that the receive buffer has received one data byte (the condition for setting I2C_RAWISR[2] is determined by I2C_RXTL[7:0]).
2. At this point, read the data from I2C_DATACMD[7:0].
3. To read multiple data bytes, repeat steps 1 and 2.

27 Inter-IC Sound (I2S) Interface

27.1 Overview

The Inter-IC Sound Bus (I2S) is a standardized communication interface developed by Philips for use in many (super) large-scale IC-based systems, especially in many digital stereo audio systems.

Main features:

- Master and slave modes
- Simplex transmitting, simplex receiving, duplex transceiving
- Philips standard, left-justified and right-justified standards, PCM (with short and long frame) standard
- Configurable audio channel length (bits): 16 or 32
- Configurable audio data length (bits): 8, 16, 24 or 32
- Stereo / mono audio data available in non-PCM mode
- The stereo audio data can be transmitted first via left channel or first via right channel.
- The mono audio data can be transmitted via left channel or right channel.
- In non-PCM mode, the polarity of WS is optional.
- In non-PCM mode, the moment of switching between SD and WS is optional on the rising or falling edge of SCK; while in PCM mode, it is fixed on the rising edge of SCK.
- Built-in 8-word TX FIFO and RX FIFO (1 word = 32 bits), each can store 16 pieces of 16-bit data or 8 pieces of 32-bit data.

- MCLK generated by PLL is required when audio block is defined as Master at 256 times the audio sampling rate (Fs), which is typically 8 / 11.025 / 16 / 22.05 / 24 / 32 / 44.1 / 48 / 96 / 192 kHz.
- If interrupt is enabled, the following conditions will trigger an interrupt:
  - TX FIFO with enough space
  - RX FIFO with enough data
  - TX FIFO under-load
  - RX FIFO overflow
- DMA operation

27.2 Pin Description

Table 27-1: I2S Pin Description

Function Pin	Alternate Function Pin	Direction	Functional Description
I2S_MCLK	PA0, PA6, PC4, PC6	Input/output	The audio master clock can be generated either from an internal audio PLL or from an external input.
I2S_WS	PA8, PB9, PB12	Input/output	Word select
I2S_CK	PA5, PB3, PB10, PB13	Input/output	Bit clock
I2S_SD	PA7, PB5, PB15, PC3	Input/output	Audio data input/output
I2S_EXTSD	PB4, PB14, PC2, PC11	Input	Full-duplex audio data input

27.3 Functional Description

27.3.1 TX FIFO

The TX FIFO is an 8-word deep FIFO used to store the data written by CPU or DMA into the "Write Data Register" (I2S_WR).

27.3.2 RX FIFO

The RX FIFO is an 8-word deep FIFO used to store the data received from the external I2S module. The CPU or DMA reads data in RX FIFO through the “Read Data Register” (I2S_RD).

27.3.3 16-bit Channel Control

This module is used to control 16-bit long channels. It sends audio data from the TX FIFO to the external I2S device through a port and receives audio data from the external I2S device to store it in the RX FIFO via another port.

In the I2S protocol, the SCK (serial clock, also known as bit clock) is used to guide the shift register to shift data by one bit during each SCK cycle. The WS signal (word select, also known as left-right clock or LRCLK) indicates which channel of data is being transmitted.

In master mode, the SCK is generated by dividing the MCLK (master clock) from the PLL module. This control module counts the edges of the synchronized SCK and generates the WS signal.

In slave mode, both SCK and WS come from the external I2S master device. This control module counts the edges of the synchronized SCK and adjusts the SCK count based on the edges of the synchronized WS.

When I2S_GCR[9] is set to 1, the transmit operation begins. When the “Transmitter Shift Register” is empty, it fetches data from the TX FIFO and shifts the data to the serial port.

When I2S_GCR[8] is set to 1, the receive operation begins. Input data from the serial port is placed in the “Receiver Shift Register”. Once all bits of the data are shifted in, the data is transferred to the RX FIFO.

27.3.4 32-bit Channel Control

This module is used to control 32-bit long audio channels. It functions similarly to the control logic for 16-bit channels, and the logic is also analogous. The main difference lies in the SCK frequency and some control signals that manage the transmission timing of dual-channel data from TX FIFO or to RX FIFO.

27.3.5 Port Synchronization

In master mode, this module generates the SCK by dividing MCLK from the PLL. To maintain the sampling rate as 1/256 of the MCLK frequency, the SCK frequency is 1/8 of MCLK for 16-bit channels and 1/4 of MCLK for 32-bit channels. For 16-bit channels, the frequency of PCLK must be greater than 3/4 times MCLK. For 32-bit channels, the frequency of PCLK must be greater than 3/2 times MCLK. In slave mode, SCK comes directly from the external I2S device. The frequency of PCLK must be greater than three times SCK.

The following table summarizes the requirements for PCLK frequency in different modes.

Table 27-2: PCLK Frequency Requirements for I2S in Different Modes

Master Mode		
16-bit channel	$f_{SCK} = 1/8 \times f_{MCLK}$	$f_{PCLK} > 3/4 \times f_{MCLK}$
32-bit channel	$f_{SCK} = 1/4 \times f_{MCLK}$	$f_{PCLK} > 3/2 \times f_{MCLK}$
Slave Mode		
$f_{PCLK} > 3 \times f_{SCK}$		

27.3.6 Interface Timing

I2S (Inter-IC Sound) is a bus standard developed by Philips. According to the Philips standard, SD (serial audio data) is delayed by one SCK (serial clock or bit clock) cycle compared to WS (word select, or left-right clock). If SD is shorter than the channel, zeros are padded after SD in the channel. By default, SD toggles on the falling edge of SCK and samples on the rising edge of SCK.

The MSB-justified (left-justified) and LSB-justified (right-justified) standards were developed later. In these standards, SD and WS are transmitted synchronously. If SD is shorter than the channel, in the MSB-justified standard, zeros are padded after SD in the channel; while in the LSB-justified standard, zeros are padded before SD in the channel. By default, SD toggles on the falling edge of SCK and samples on the rising edge of SCK.

In the PCM standard, SD is delayed by one SCK cycle compared to WS. There are two frame synchronization formats in the PCM standard: short frame synchronization and long frame synchronization. In the short frame synchronization format, when the least significant bit of SD (including the zeros padded after the valid bits of SD) is being transmitted, WS remains high for one SCK cycle. In the long frame synchronization format, when the high 13 bits of SD are being transmitted, WS remains high for 13 SCK cycles. Note that WS does not represent word select (left/right clock), because there is no distinction between the left channel and the right channel in the PCM standard. If SD is shorter than the channel, zeros are padded after SD in the channel. Specifically, SD toggles on the rising edge of SCK and samples on the falling edge of SCK, which is not configurable.

Figure 27-1 shows the I2S interface timing for a 16-bit channel, and Figure 27-2 shows the I2S interface timing for a 32-bit channel. Note that “sck_cnt” is an inaccessible internal counter and is not a signal on the I2S port.

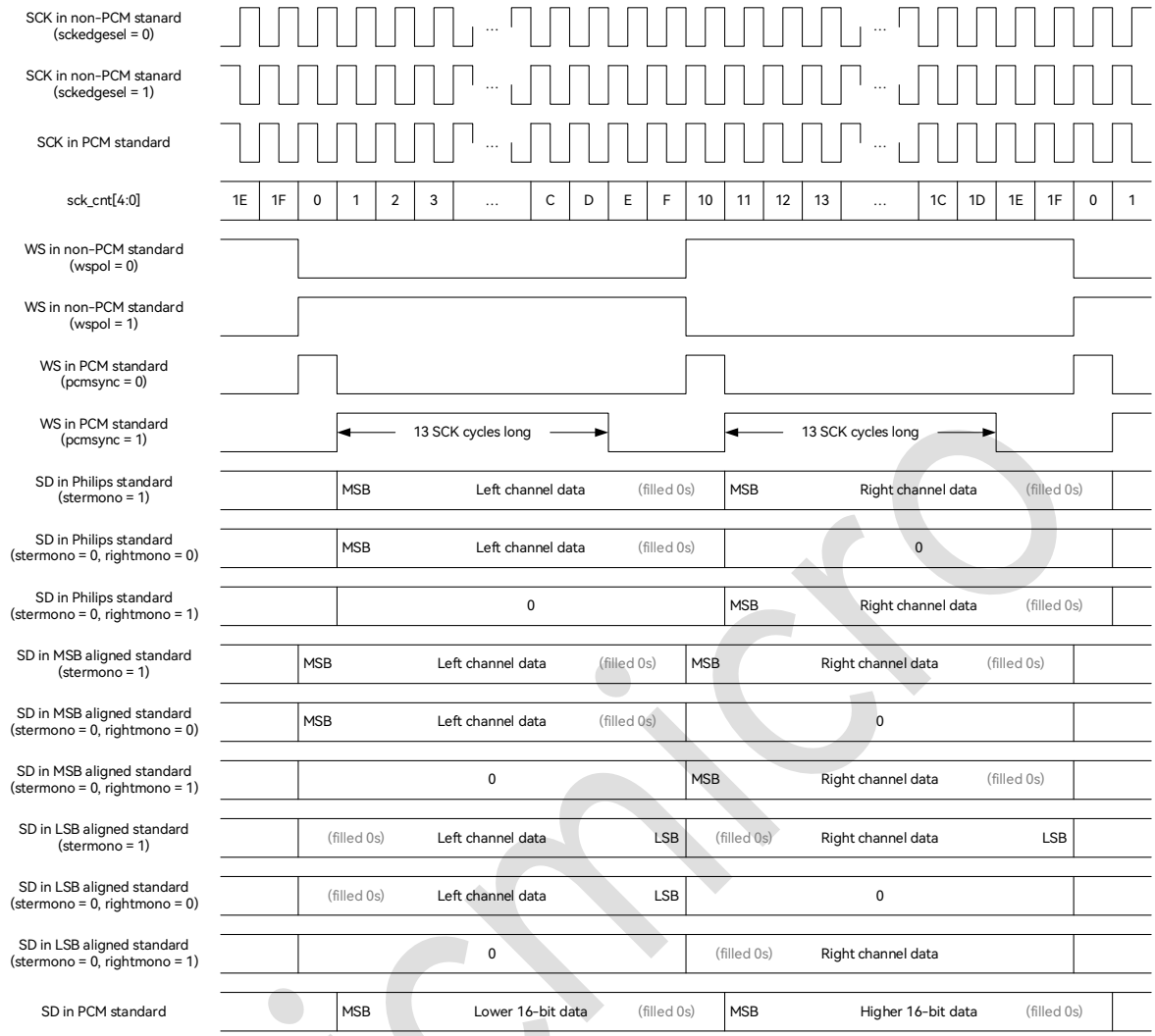


Figure 27-1: I2S Signal Waveform for 16-bit Channel

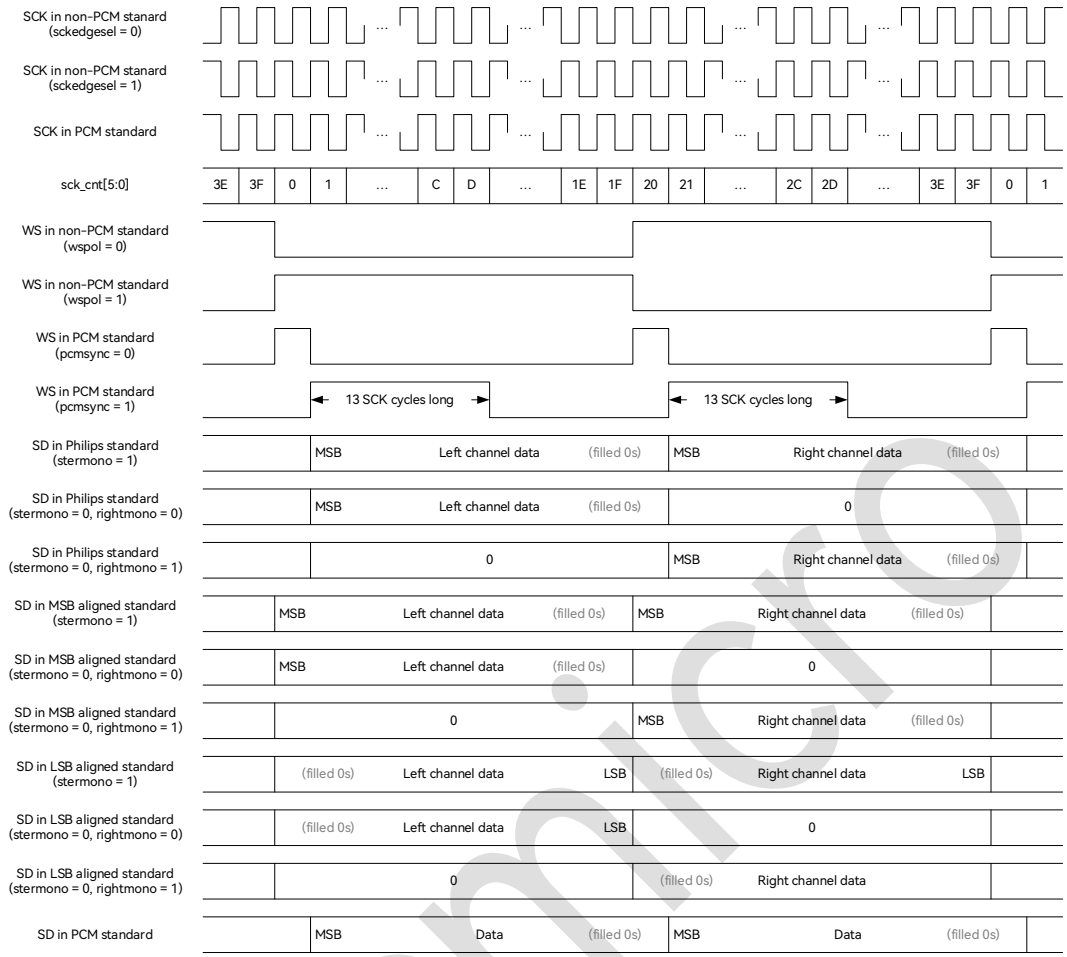


Figure 27-2: I2S Signal Waveform for 32-bit Channel

27.4 Register Description

I2S register base address: 0x4700_B000

The registers are listed below:

Table 27-3: List of I2S Registers

Offset Address	Name	Description
0x00	I2S_WR	I2S write data register
0x04	I2S_RD	I2S read data register
0x08	I2S_CSR	I2S current status register
0x0C	I2S_GCR	I2S global control register
0x10	I2S_DFR	I2S data format register
0x14	I2S_ISR	I2S interrupt status register
0x18	I2S_IER	I2S interrupt enable register
0x1C	I2S_ICR	I2S interrupt clear register

27.4.1 I2S Write Data Register (I2S_WR)

Offset address: 0x00

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:0	I2S_WR	W/R	0x0	Audio data written to TX FIFO: ● For data in a 16-bit channel: One word of data contains two 16-bit mono data (with the 16 low bits transmitted first) or a pair of stereo data (with the 16 low bits transmitted first; by default, the left channel data is transmitted first, but it is possible to configure which data is transferred first via I2S_DFR[8]). ● For data in a 32-bit channel: One word of data contains either a single 32-bit mono data or stereo data.

Note: The data in TXFIFO must be MSB-justified.

1. For data in a 16-bit channel:

   If the data is 8 bits long, it must be stored in bits [31:24] or [15:8], while bits [23:16] and [7:0] are ignored.

2. For data in a 32-bit channel:

   - If the data is 8 bits long, it must be stored in bits [31:24], while bits [23:0] are ignored.
   - If the data is 16 bits long, it must be stored in bits [31:16], while bits [15:0] are ignored.
   - If the data is 24 bits long, it must be stored in bits [31:8], while bits [7:0] are ignored.

27.4.2 I2S Read Data Register (I2S_RD)

Offset address: 0x04

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:0	I2S_RD	R	0x0	Audio data read from RX FIFO: For data in a 16-bit channel: One word of data contains two 16-bit mono data (with the 16 low bits transmitted first) or a pair of stereo data (with the 16 low bits transmitted first; by default, the left channel data is transmitted first, but it is possible to configure which data is transferred first via I2S_DFR[8] "Right_first"). For data in a 32-bit channel: One word of data contains either a single 32-bit mono data or stereo data.

Note: The data in RX FIFO is always MSB-justified.

1. For data in a 16-bit channel:

   If the data is 8 bits long, it will be stored in bits [31:24] or [15:8], while bits [23:16] and [7:0] will remain zero.

2. For data in a 32-bit channel:

   - If the data is 8 bits long, it will be stored in bits [31:24], while bits [23:0] will remain zero.
   - If the data is 16 bits long, it will be stored in bits [31:16], while bits [15:0] will remain zero.
   - If the data is 24 bits long, it will be stored in bits [31:8], while bits [7:0] will remain zero.

27.4.3 I2S Current Status Register (I2S_CSR)

Offset address: 0x08

Reset value: 0x0000 0105

Bit	Name	Attribute	Reset Value	Description
31:16	RSV	-	-	Reserved
15:12	RXFIFO_LEVEL	R	0x0	RX FIFO data count: These bits indicate how many words of data are in the RX FIFO. The range is from 0x0 to 0x8.
11	RSV	-	-	Reserved
10	RXFIFO_TRIG	R	0x0	RX FIFO data available status: This bit is set to 1 when the RX FIFO has received enough data (reaching the data amount specified by I2S_GCR[1:0]). 0: RX FIFO without enough data 1: RX FIFO with enough data
9	RXFIFO_FULL	R	0x0	RX FIFO full status: 0: RX FIFO not full 1: RX FIFO full
8	RXFIFO_EMPTY	R	0x1	RX FIFO empty status: 0: RX FIFO not empty 1: RX FIFO empty
7:4	TXFIFO_LEVEL	R	0x0	TX FIFO data count: These bits indicate how many words of data are in the TX FIFO. The range is from 0x0 to 0x8.
3	RSV	-	-	Reserved
2	TXFIFO_TRIG	R	0x1	TX FIFO availability: This bit is set to 1 when the TX FIFO has sufficient empty space (reaching the number of vacancies specified by I2S_GCR[5:4]). 0: TX FIFO without sufficient empty space 1: TX FIFO with sufficient empty space

Bit	Name	Attribute	Reset Value	Description
1	TXFIFO_FULL	R	0x0	TX FIFO full status: 0: TX FIFO not full 1: TX FIFO full
0	TXFIFO_EMPTY	R	0x1	TX FIFO empty status: 0: TX FIFO not empty 1: TX FIFO empty

27.4.4 I2S Global Control Register (I2S_GCR)

Offset address: 0x0C

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:15	RSV	-	-	Reserved
15	SD_DIR	W/R	0x0	SD pin direction selection: 0: input 1: output In full-duplex mode, the SD pin must be set to output.
14	I2S_EN	W/R	0x0	I2S enable: 0: I2S disabled 1: I2S enabled
13	DMA_MODE	W/R	0x0	DMA access enable: 0: CPU reads/writes TX FIFO and RX FIFO 1: DMA reads/writes TX FIFO and RX FIFO
12	INTEN	W/R	0x0	I2S interrupt enable: 0: I2S interrupt disabled 1: I2S interrupt enabled
11	MST_MODE	W/R	0x0	Master/slave mode selection: In master mode, I2S outputs SCK and WS; in slave mode, I2S inputs SCK and WS. 0: slave mode 1: master mode

Bit	Name	Attribute	Reset Value	Description
10	FILLDATASEL	W/R	0x0	Data padding selection when TX FIFO is underrun: 0: transmit empty packet data (all zeros) 1: transmit the previous data
9	TXEN	W/R	0x0	Transmit control logic and TX FIFO enable: 0: transmit disabled 1: transmit enabled
8	RXEN	W/R	0x0	Receive control logic and RX FIFO enable: 0: receive disabled 1: receive enabled
7:5	RSV	-	-	Reserved
4	TXFIFO_WTMK	W/R	0x0	TX FIFO watermark (number of triggers) selection: 0: TX FIFO not full (There is 1 word or more empty bits in TX FIFO.) 1: 4 words or less data in TX FIFO (There are 4 words or more empty bits in TX FIFO.)
3:1	RSV	-	-	Reserved
0	RXFIFO_WTMK	W/R	0x0	RX FIFO watermark (number of triggers) selection: 0: RX FIFO not empty 1: 4 words or more data in RX FIFO

27.4.5 I2S Data Format Register (I2S_DFR)

Offset address: 0x10

Reset value: 0x0000 0024

Bit	Name	Attribute	Reset Value	Description
31:11	RSV	-	-	Reserved

Bit	Name	Attribute	Reset Value	Description
10	PCM_SYNC	W/R	0x0	PCM frame synchronization format selection: This bit is valid only in PCM standard. 0: short frame synchronization (WS stays high for 1 SCK cycle) 1: long frame synchronization (WS stays high for 13 SCK cycles)
9	RIGHT_MONO	W/R	0x0	Left/right mono selection: This bit is valid only for mono data. 0: select left channel data for transmission or reception 1: select right channel data for transmission or reception
8	RIGHT_FIRST	W/R	0x0	Stereo audio priority channel selection: This bit is used to select which channel of data is transmitted or received first and is valid only for stereo data. 0: left channel data first 1: right channel data first
7	SCK_EDGESEL	W/R	0x0	Selection of SCK edge where data switching occurs: This bit is not valid for the PCM standard. Note: In the PCM standard, SD and WS switch on the rising edge of SCK. 1: SD and WS switch on the rising edge of SCK 0: SD and WS switch on the falling edge of SCK (recommended)
6	WSPOL	W/R	0x0	WS polarity selection: This bit is not valid for the PCM standard, as it does not concern left or right channels; WS is used for frame synchronization. 0: WS low indicates left channel (used in Philips standard) 1: WS high indicates left channel (used in MSB- and LSB-justified standards)

Bit	Name	Attribute	Reset Value	Description
5	STER_MONO	W/R	0x1	Stereo or mono data selection: This bit is not valid for the PCM standard, as it does not concern left or right channels. In fact, the control logic of the PCM standard is designed for stereo data. 0: mono data 1: stereo data
4	CHANLEN32	W/R	0x0	Channel length selection: 0: 16 bits 1: 32 bits
3:2	DATA_LENSEL	R	0x1	Audio data length selection: A 16-bit channel supports 8-bit and 16-bit data, while a 32-bit channel supports all four data types. 0x0: 8 bits 0x1: 16 bits (default) 0x2: 24 bits 0x3: 32 bits
1:0	I2S_STD	W/R	0x0	I2S standard selection: 0X0: Philips standard 0x1: MSB-justified (left-justified) standard 0x2: LSB-justified (right-justified) standard 0x3: PCM standard

27.4.6 I2S Interrupt Status Register (I2S_ISR)

Offset address: 0x14

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:5	RSV	-	-	Reserved
4	FRAME_ERR_INTF	R	0x0	Frame error interrupt flag: In slave mode, this bit is set when WS switches at unexpected time. Frame errors may lead to errors and omissions in SD, as if a frame error occurs, the SCK counter will

Bit	Name	Attribute	Reset Value	Description
				automatically correct itself. However, this sudden jump in the SCK counter may cause many control signal errors. 0: no frame error has occurred. 1: frame error has occurred.
3	UNDERRUN_INTF	R	0x0	TX FIFO underrun error interrupt flag: This bit is set when the TX FIFO is empty after transmit is enabled. The data transmitted can be 0x0 or previously transmitted data. 0: no TX FIFO underrun error has occurred. 1: TX FIFO underrun error has occurred.
2	RXOERR_INTF	R	0x0	RX FIFO overflow error interrupt flag: This bit is set when an attempt is made to write a newly arrived data word into a full RX FIFO. In this case, the newly arrived data will be lost, and the receiving process will continue. 0: no RX FIFO overflow error has occurred. 1: RX FIFO overflow error has occurred.
1	RX_INTF	R	0x0	RX FIFO data sufficient interrupt flag: This bit is set when the RX FIFO has received enough data (reaching the data amount specified by I2S_GCR[1:0]). 0: RX FIFO has insufficient data. 1: RX FIFO has sufficient data.
0	TX_INTF	R	0x0	TX FIFO space sufficient interrupt flag: This bit is set when the TX FIFO has sufficient empty space (reaching the number of vacancies specified by I2S_GCR[5:4]). 0: TX FIFO has insufficient empty space. 1: TX FIFO has sufficient empty space.

Note: This interrupt status register is the raw interrupt status register, which cannot be masked by the interrupt enable register but can be cleared by the interrupt clear register.

27.4.7 I2S Interrupt Enable Register (I2S_IER)

Offset address: 0x18

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:5	RSV	-	-	Reserved
4	FRAME_ERR_INTEN	W/R	0x0	Frame error interrupt enable: 0: disabled 1: enabled
3	UNDERRUN_INTEN	W/R	0x0	TX FIFO underrun error interrupt enable: 0: disabled 1: enabled
2	RXOERR_INTEN	W/R	0x0	RX FIFO overflow error interrupt enable: 0: disabled 1: enabled
1	RX_INTEN	W/R	0x0	RX FIFO data sufficient interrupt enable: 0: disabled 1: enabled
0	TX_INTEN	W/R	0x0	TX FIFO space sufficient interrupt enable: 0: disabled 1: enabled

27.4.8 I2S Interrupt Clear Register (I2S_ICR)

Offset address: 0x1C

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:5	RSV	-	-	Reserved
4	FRAME_ERR_INTCLR	W	0x0	Frame error interrupt clear: 0: not cleared 1: cleared
3	UNDERRUN_INTCLR	W	0x0	TX FIFO underrun error interrupt clear: 0: not cleared 1: cleared

Bit	Name	Attribute	Reset Value	Description
2	RXOERR_INTCLR	W	0x0	RX FIFO overflow error interrupt clear: 0: not cleared 1: cleared
1	RX_INTCLR	W	0x0	RX FIFO data sufficient interrupt clear: 0: not cleared 1: cleared
0	TX_INTCLR	W	0x0	RX FIFO space sufficient interrupt clear: 0: not cleared 1: cleared

27.5 Operation Procedure

1. Configure the I2S_DFR register to select the I2S standard and set the channel length, data format, etc.
2. Configure the I2S_IER register to enable interrupts.
3. Configure the I2S_GCR register to enable I2S and interrupts, but do not enable transmission (I2S_GCR[9]) or reception (I2S_GCR[8]) yet. Also select master/slave mode, CPU or DMA access mode, and FIFO watermark, etc.
4. If DMA access mode is selected, please configure the DMA controller correctly.
5. Set I2S_GCR[15] to choose the SD pin as input or output (must be set as output for full-duplex transmission).
6. Set I2S_GCR[9] / I2S_GCR[8] to start data transmission.
7. If CPU access mode is selected, write the data to be transmitted to the I2S_WR register whenever there is sufficient space in the TX FIFO, and/or read the received data from the I2S_RD register whenever there is enough data in the RXFIFO. The status of TX FIFO and RX FIFO can be checked through the I2S_CSR and I2S_ISR registers, or can be obtained via interrupt requests.

8. If DMA access mode is selected, the DMA controller will write the data to be transmitted to the I2S_WR register via DMA handshake signals, and/or read the received data from the I2S_RD register. The status of TXFIFO and RXFIFO can be checked at any time through the I2S_CSR and I2S_ISR registers.

9. Regardless of whether CPU access mode or DMA access mode is selected, disable I2S by clearing I2S_GCR[14] after the transmission is complete.

28 Universal Asynchronous Receiver Transmitter (UART0 & UART2 & UART3)

28.1 Overview

Universal asynchronous receiver/transmitter (hereinafter referred to as UART) is a widely used serial communication interface that supports full duplex communication. UART is to send the data transmitted in parallel in memory or processor to the UART receiver of peripherals in series, or to receive the serial data of UART peripherals and convert them into parallel data for the processor. It supports serial communication with external interface devices.

28.2 Main Features

- Standard asynchronous communication data formats
  - 1 start bit
  - 1 parity bit (odd or even) or no parity bit
  - 1 stop bit
  - Bytes transmitted sequentially from LSB to MSB
- 8-bit 4-level RX FIFO
- Programmable baud rate (adjustable according to frequency divider parameter)
- Data communication and error handling interrupts
- Status bit can be accessed by either query or interrupt
- Flags of FIFO non-empty, half-full, full, overflow
- Flags of transmission data error and parity error
- Transmission at common baud rates such as 9600 bps, 19200 bps and 115200 bps
- Self-test mode, that is, receive the data transmitted by itself

28.3 Pin Description

Table 28-1: UART0 & UART2 & UART3 Pin Description

Function Pin	Alternate Function Pin	Direction	Functional Description
UART0_TX	PA9, PB6, PA4, PB8	Output	Transmitting data
UART0_RX	PA10, PB7, PA5	Input	Receiving data
UART2_TX	PB10, PC10, PB4, PB8, PD8	Output	Transmitting data
UART2_RX	PB11, PC11, PB5, PB9, PD9	Input	Receiving data
UART3_TX	PA0, PC10, PB0, PB14	Output	Transmitting data
UART3_RX	PA1, PC11, PB1, PB15, PD10	Input	Receiving data

28.4 Functional Description

The UART can be configured to transmit or receive data at any desired baud rate based on user requirements.

28.4.1 Configurable Baud Rate

The UART supports the configuration of any baud rate for data transmission and reception, which is primarily controlled by the BRPL and BRPH registers, according to the following formula:

$$baud = \frac{fclk}{\{BRPH,BRPL\}}$$

For example, to obtain a baud rate of 9600 kbps with the system clock fclk of 96 MHz:

UART_BRP = 96,000,000 ÷ 9,600 = 10,000 = 0x2710

i.e., UART_BRPH = 0x27, UART_BRPL = 0x10.

28.4.2 UART Transmit Mode

In UART transmit mode, parallel data can be converted into serial data for transmitting. When using the UART for data transmission, it is necessary to first configure the baud rate, enable or disable parity checking, and specify the type of parity checking. Interrupt-trigger mode can

also be enabled. The UART can only transmit one byte of data at a time and includes error checking functionality for transmitted data. It also supports parity bit functionality, allowing the receiver to check and validate the received data.

28.4.3 UART Receive Mode

When using the UART for data reception, any baud rate can be configured for data reception, and parity checking can be used to verify if any errors occurred during data transmission. Additionally, an RX FIFO is provided, capable of storing a maximum of 4 bytes of data.

28.5 Register Description

UART0 register base address: 0x4700_F000

UART2 register base address: 0x40B0_3000

UART3 register base address: 0x4700_D000

Table 28-2: List of UART0/2/3 Registers

Offset Address	Name	Description
0x00	UART_ISR	UART interrupt status register
0x04	UART_IER	UART interrupt enable register
0x08	UART_CR	UART control register
0x0C	UART_TDR	UART transmit data register
0x0C	UART_RDR	UART receive data register
0x10	UART_BRPL	UART baud rate parameter low register
0x14	UART_BRPH	UART baud rate parameter high register

Registers are detailed in the following sections.

28.5.1 Interrupt Status Register (UART_ISR)

Offset address: 0x00

Reset value: 0x00

Description: The interrupt status register mainly consists of the FIFO non-empty flag, half-full flag, full flag, overflow flag, transmission complete flag, and parity error flag. These flags reflect the status of the data transmission process.

Bit	Name	Attribute	Reset Value	Description
7:6	RSV	-	-	Reserved
5	FIFO_NE	R/W0C	0x0	FIFO non-empty flag: 0: FIFO empty 1: FIFO not empty This bit will be automatically cleared when FIFO is empty.
4	FIFO_HF	R/W0C	0x0	FIFO half-full flag: 0: FIFO not half-full 1: FIFO half-full When reading data in FIFO, this bit is cleared automatically.
3	FIFO_FU	R/W0C	0x0	FIFO full flag: 0: FIFO not full 1: FIFO full When reading data in FIFO, this bit is cleared automatically. This bit can also be cleared by software via writing 0 to it.
2	FIFO_OV	R/W0C	0x0	RX FIFO receive overflow error: 0: no receive overflow error 1: receive overflow error occurred
1	TXEND	R/W0C	0x0	UART transmission complete flag: 0: transmission is not complete 1: transmission is complete
0	TRE	R/W0C	0x0	UART transmission/reception parity error flag: 0: no parity error upon completion of UART transmission/reception 1: parity error occurred upon completion of UART transmission/reception

28.5.2 Interrupt Enable Register (UART_IER)

Offset address: 0x04

Reset value: 0x00

Description: The interrupt enable register primarily enables the FIFO non-empty interrupt, half-full interrupt, full interrupt, overflow interrupt, transmission complete interrupt, and parity error interrupt.

Bit	Name	Attribute	Reset Value	Description
7:6	RSV	-	-	Reserved
5	FIFO_EN	R/W	0x0	FIFO non-empty interrupt enable: 0: disabled 1: enabled
4	FIFO_HFEN	R/W	0x0	FIFO half-full interrupt enable: 0: disabled 1: enabled
3	FIFO_FUEN	R/W	0x0	FIFO full interrupt enable: 0: disabled 1: enabled
2	FIFO_OVEN	R/W	0x0	RX FIFO receive overflow interrupt enable: 0: disabled 1: enabled
1	TXEND_EN	R/W	0x0	UART transmission complete interrupt enable: 0: disabled 1: enabled
0	TRE_EN	R/W	0x0	UART TX/RX parity error interrupt enable: 0: disabled 1: enabled

28.5.3 Control Register (UART_CR)

Offset address: 0x08

Reset value: 0x00

Description: The control register primarily controls UART transmission enable, parity enable and type, FIFO data clear, and self-test mode.

Bit	Name	Attribute	Reset Value	Description
7:5	RSV	-	-	Reserved
4	UART_LB	R/W	0x0	UART self-test mode enable: 0: disabled 1: enabled
3	UART_PD	R/W	0x0	Parity check enable: 0: with parity check 1: no parity check
2	FLUSH	R/W	0x0	Clear data and pointer in UART RX FIFO: 0: not clear 1: clear
1	TRS	R/W	0x0	UART data transmission enable: 0: data transmission disabled 1: data transmission enabled
0	ODD_EN	R/W	0x0	Parity mode selection: 0: even parity 1: odd parity

28.5.4 Transmit Data Register (UART_TDR)

Offset address: 0x0C

Reset value: 0x00

Bit	Name	Attribute	Reset Value	Description
7:0	UART_DATA	W	0x0	Store the data to be transmitted.

28.5.5 Receive Data Register (UART_RDR)

Offset address: 0x0C

Reset value: 0x00

Bit	Name	Attribute	Reset Value	Description
7:0	UART_DATA	R	0x0	Store the data received.

28.5.6 Baud Rate Parameter Low-order Register (UART_BRPL)

Offset address: 0x10

Reset value: 0x74

Bit	Name	Attribute	Reset Value	Description
7:0	UART_BRPL	R/W	0x74	The baud rate parameter registers UART_BPRH and UART_BPRL constitute a 16-bit frequency divider.

28.5.7 Baud Rate Parameter High-order Register (UART_BRPH)

Offset address: 0x14

Reset value: 0x01

Bit	Name	Attribute	Reset Value	Description
7:0	UART_BRPH	R/W	0x01	The baud rate parameter registers UART_BPRH and UART_BPRL constitute a 16-bit frequency divider.

28.6 Operation Procedure

28.6.1 UART Initialization

1. Enable the clocks for the corresponding GPIO pins and configure the pins for UART_TX and UART_RX alternate functions.
2. Configure the system configuration register for the UART module clock.
3. Configure the UART_BRPH[7:0] and UART_BRPL[7:0] registers to set the UART baud rate.
4. Configure the UART_CR register to set UART parity and enable UART data transmission.
5. Configure UART_IER register to enable the corresponding UART interrupts.

28.6.2 UART Transmission Process

1. Before transmitting data, the software can configure the baud rate parameter and parity type.
2. Set UART_CR[1] to 1 to enable transmission.
3. Write the first data byte to the UART_TDR register.
4. Query the transmission complete flag UART_ISR[1], if UART1_ISR[1] = 1, the current data transmission is completed.
5. Write the next data byte to UART_TDR.

28.6.3 UART Reception Process

1. Before transmitting data, the software can configure the baud rate parameter and parity type.
2. For data reception, query the UART_ISR[5] flag or wait for an interrupt. If UART_ISR[5] = 1, it means the RX FIFO is not empty; then read the data from the UART_RDR. The corresponding flag will be automatically cleared after reading the data.
3. If there is an error in data reception, wait for the interrupt or query the UART_ISR register flag bit to determine the error type and execute corresponding error handling, after which the software clears the error flag bit.
4. Continue to receive data.

29 Enhanced Universal Asynchronous Receiver Transmitter (UART1)

29.1 Overview

Universal asynchronous receiver/transmitter (hereinafter referred to as UART) is a widely used serial communication interface that supports full duplex communication. UART is to send the data transmitted in parallel in memory or processor to the UART receiver of peripherals in series, or to receive the serial data of UART peripherals and convert them into parallel data for the processor. It supports serial communication with external interface devices.

29.2 Main Features

- 16-byte hardware FIFO
- Baud rate supporting integer-N and fractional-N
- CTS / RTS flow control
- Error start bit detection
- Frame interrupt detection
- Circuit-break detection
- Setting of data bit width (5–9 bits) and number of stop bits (1 bit, 1.5 bits and 2 bits)
- Fixed check, parity check or no parity check for data
- IrDA 1.0 protocol with the baud rate ranging from 9.6 k to 115.2 k
- DMA operation

29.3 Pin Description

Table 29-1: UART1 Pin Description

Function Pin	Alternate Function Pin	Direction	Functional Description
UART1_TX	PA2, PA6, PA14, PC0 PA9, PA10, PA11, PB10	Output	Transmitting data
UART1_RX	PA3, PA7, PA13, PC1, PD2 PA10, PA9, PA12, PB11	Input	Receiving data
UART1_CTS	PA0, PB3, PB13, PA2	Input	Hardware flow control mode transmitting enable signal
UART1_RTS	PA1, PB5, PB14, PA3	Output	Hardware flow control mode transmitting request signal

29.4 Functional Description

The UART can be configured to transmit or receive data at any desired baud rate based on user requirements.

29.4.1 Configurable Baud Rate

UART supports the configuration of any baud rate for data transmission and reception, which is primarily configured by the DLL register, DLH register and DLF register. The calculation process is as follows:

Baud rate is the desired baud rate and f_{clk} is the clock frequency,

A is the integer part of $\frac{fclk}{16*Baud\ rate}$, B is the fractional part of $\frac{fclk}{16*Baud\ rate}$,

then A is configured by the DLH and DLL registers; B is configured by the DLF register.

29.4.2 UART Transmission Mode

In UART transmission mode, parallel data can be converted into serial data for transmission. When using UART for data transmission, it is possible to configure the data size, baud rate, whether parity checking is enabled, and the type of parity.

29.4.3 UART Reception Mode

When using the UART for data reception, any baud rate can be configured for data reception, and parity checking can be used to verify if any errors occurred during data transmission.

29.4.4 IrDA Mode

UART is compatible with the IrDA 1.0 physical layer protocol, with a maximum transmission baud rate of 115.2K Baud. The data format is fixed as 1 start bit + 8 data bits + 1 stop bit, with no parity bit.

29.5 Register Description

UART1 register base address: 0x4700_E000

Table 29-2: List of UART1 Registers

Offset Address	Name	Description
0x00	UART1_RBR	Receive buffer register
0x00	UART1_THR	Transmit buffer register
0x00	UART1_DLL	Baud rate division low-order register
0x04	UART1_DLH	Baud rate division high-order register
0x04	UART1_IER	Interrupt enable register
0x08	UART1_IIR	Interrupt status register
0x08	UART1_FCR	FIFO control register
0x0C	UART1_LCR	LINE control register
0x10	UART1_MCR	Flow control register
0x14	UART1_LSR	LINE status register
0x18	UART1_MSR	Flow status register
0x20	UART1_LPDLL	Low-power baud rate division low-order register
0x24	UART1_LPDLH	Low-power baud rate division high-order register
0x7C	UART1_USR	Status register
0x80	UART1_TFL	TX FIFO data count register
0x84	UART1_RFL	RX FIFO data count register
0xA4	UART1_HTX	Suspend TX transmission
0xA8	UART1_DMASA	DMA software acknowledgment
0xC0	UART1_DLF	Fractional frequency division register

Offset Address	Name	Description
0xC4	UART1_RAR	Receive address matching register
0xC8	UART1_TAR	Transmit address matching register
0xCC	UART1_LCRE	LINE control extension register

Registers are detailed in the following sections.

29.5.1 Receive Buffer Register (UART1_RBR)

Offset address: 0x00

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:9	RSV	-	-	Reserved
8:0	RBR	R	0x0	The receive data register can store data received in UART mode or SIR mode. The data in this register is valid only when the DR bit of the UART_LSR register is set. This field serves as the entry for the RX FIFO and can only be accessed when the DLAB bit of the UART_LCR is 0.

29.5.2 Transmit Buffer Register (UART1_THR)

Offset address: 0x00

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:9	RSV	-	-	Reserved
8:0	THR	W	0x0	The transmit data register can store data to be transmitted in UART mode or SIR mode. Data can only be written to this register when the THRE bit of the UART_LSR register is set. This field serves as the entry for the TX FIFO and can only be accessed when the DLAB bit of the UART_LCR is 0.

29.5.3 Baud Rate Division Low-order Register (UART1_DLL)

Offset address: 0x00

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:8	RSV	-	-	Reserved
7:0	DLL	R/W	0x0	Baud rate configuration register (low): This field can only be accessed when the DLAB bit of the UART_LCR is 1 and the UART is in an idle state (when the BUSY bit of the UART_USR is 0). The calculation formula for the integer part of baud rate is: baud rate = f_{clk} / (16 * {DLH, DLL}) Note: If there is a fractional division, the DLF must be configured before configuring DLL.

29.5.4 Baud Rate Division High-order Register (UART1_DLH)

Offset address: 0x04

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:8	RSV	-	-	Reserved
7:0	DLH	R/W	0x0	Baud rate configuration register (high): This field can only be accessed when the DLAB bit of the UART_LCR is 1 and the UART is in an idle state (when the BUSY bit of the UART_USR is 0). The calculation formula for the integer part of baud rate is: baud rate = f_{clk} / (16 * {DLH, DLL}) Note: If there is a fractional division, the DLF must be configured before configuring DLL.

29.5.5 Interrupt Enable Register (UART1_IER)

Offset address: 0x04

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:8	RSV	-	-	Reserved
7	PTIME	R/W	0x0	THRE interrupt enable: this field is only accessible when the DLAB bit of UART_LCR is 0. 1: THRE interrupt enabled 0: THRE interrupt disabled
6:3	RSV	-	-	Reserved
2	ELSI	R/W	0x0	Receiver LINE status interrupt enable: this field is only accessible when the DLAB bit of UART_LCR is 0. 1: LINE interrupt enabled 0: LINE interrupt disabled
1	ETBEI	R/W	0x0	TX FIFO empty interrupt enable: this field is only accessible when the DLAB bit of UART_LCR is 0. 1: TX FIFO empty interrupt enabled 0: TX FIFO empty interrupt disabled
0	ERBFI	R/W	0x0	Receive data available interrupt enable and timeout interrupt (FIFO enabled) enable: this field is only accessible when the DLAB bit of UART_LCR is 0. 1: RX FIFO non-empty interrupt and timeout interrupt enabled 0: RX FIFO non-empty interrupt and timeout interrupt disabled

29.5.6 Interrupt Status Register (UART1_IIR)

Offset address: 0x08

Reset value: 0x0000 0001

Bit	Name	Attribute	Reset Value	Description
31:8	RSV	-	-	Reserved

Bit	Name	Attribute	Reset Value	Description
7:6	FIFOSE	R	0x0	FIFO enable flag: 11: FIFO enabled 00: FIFO disabled
5:4	RSV	-	-	Reserved
3:0	IID	R	0x01	Status ID: 0000: reserved 0001: no interrupt 0010: TX FIFO empty 0100: RX FIFO not empty 0110: receiver LINE empty 0111: busy 1100: timeout status; after enabling FIFO and RX FIFO non-empty interrupts, if there is at least one data in RX FIFO and the CPU has not read the FIFO within 4 UART frames, this field will be set to the timeout interrupt status. Others: reserved

Note: The interrupt status in this register will be cleared upon reading.

29.5.7 FIFO Control Register (UART1_FCR)

Offset address: 0x08

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:8	RSV	-	-	Reserved
7:6	RT	W	0x0	RX FIFO non-empty interrupt setting: when the number of data in FIFO is greater than or equal to the corresponding status of this setting, the RX FIFO non-empty interrupt bit is set as: 00: 1 frame of data 01: 4 frames of data 10: 8 frames of data 11: 14 frames of data
5:4	TET	W	0x0	TX FIFO empty interrupt setting: when the number of

Bit	Name	Attribute	Reset Value	Description
				data in FIFO is smaller than or equal to the corresponding status of this setting, the TX FIFO empty interrupt bit is set as: 00: FIFO empty 01: 2 frames of data 10: 4 frames of data 11: 8 frames of data
3	RSV	-	-	Reserved
2	XFIFOR	W	0x0	This bit can only be written as 0; writing 1 is invalid.
1	RFIFOR	W	0x0	This bit can only be written as 0; writing 1 is invalid.
0	FIFOE	W	0x0	FIFO enable: 1: FIFO enabled 0: FIFO disabled Changing the value of this bit will reset both the RX FIFO and TX FIFO.

29.5.8 LINE Control Register (UART1_LCR)

Offset address: 0x0C

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:8	RSV	-	-	Reserved
7	DLAB	R/W	0x0	UART_DLL and UART_DLH register access setting bits: can be written only when UART is idle: 1: UART_DLL and UART_DLH can be accessed at offset addresses 0x0 and 0x4 respectively. 0: UART_RBR / UART_THR can be accessed at offset address 0x0, and UART_IER can be accessed at offset address 0x4.
6	BC	R/W	0x0	Break control bit: used to generate a break condition transmitted to the receiving device. In UART mode, it will hold TX low, while in SIR mode, it will continuously send positive pulses on TX. 1: enable break

Bit	Name	Attribute	Reset Value	Description
				0: disable break Note: When there is still data that has not been transmitted, it will not be triggered until the transmission is completed.
5	SEPS	R/W	0x0	Forced setting of parity bit: writable only when UART is idle: 1: When SEPS, PEN and EPS are all set to 1, the parity bits checked for transmitting and receiving are 0; when SEPS and PEN are 1, and EPS is 0, the parity bits checked for transmitting and receiving are 1; when PEN is 0, there are no parity bits for transmitting and receiving. 0: The forced setting function of parity bit is disabled.
4	EPS	R/W	0x0	Parity selection bit: writable only when UART is idle: 1: even parity 0: odd parity
3	PEN	R/W	0x0	Parity enable bit: writable only when UART is idle: 1: parity enabled 0: parity disabled
2	STOP	R/W	0x0	Stop bit length setting: writable only when UART is idle: 1: when DLS = 00, 1.5 bits STOP; otherwise, 2 bits STOP. 0: 1 bit STOP
1:0	DLS	R/W	0	UART frame data length setting: writable only when UART is idle: 00: 5 bits 01: 6 bits 10: 7 bits 11: 8 bits

29.5.9 Flow Control Register (UART1_MCR)

Offset address: 0x10

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:8	RSV	-	-	Reserved
7	DMAE	R/W	0	DMA transfer mode selection bit:

Bit	Name	Attribute	Reset Value	Description
				0: DMA transfer mode disabled 1: DMA transfer mode enabled Note: To communicate with an external DMA controller, this bit must be set.
6	SIRE	R/W	0	SIR (IrDA) mode selection bit: 0: IrDA SIR mode disabled 1: IrDA SIR mode enabled In this mode, a frame of data consists of 1 start bit, 8 data bits and 1 stop bit, and cannot be modified through the LCR register (when using this mode, the lower four bits of the LCR must not be configured before enabling this bit).
5	AFCE	R/W	0	Automatic flow control mode selection bit: 1: CTS/RTS automatic flow control enabled 0: CTS/RTS automatic flow control disabled
4	LB	R/W	0	UART self-diagnostic mode In UART mode, the data on the sout line remains high, while the serial data output is internally looped back to the sin line, allowing the UART TX output waveform to be sent back to the same UART RX. In SIR mode, the data on the sir_out line remains low, allowing the UART TX output waveform to be inverted and sent back to the same UART RX. 1: LOOP mode enabled 0: LOOP mode disabled Note: The TX to RX operation is performed internally, so the TX waveform is not actually output.
3:2	RSV	-	-	Reserved
1	RTS	R/W	0	RTS interface software control bit: 1: RTS request output valid 0: RTS request output invalid Note: When LB mode is enabled (MCR[4] = 1), RTS will become invalid.
0	RSV	-	-	Reserved

29.5.10 LINE Status Register (UART1_LSR)

Offset address: 0x14

Reset value: 0x0000 0060

Bit	Name	Attribute	Reset Value	Description
31:9	RSV	-	-	Reserved
8	ADDR_RCVD	R	0x0	Flag of whether the received data is address or data in 9-bit data mode: 1: the received data is address information. 0: the received data is data information. Read this register and clear it.
7	RFE	R	0x0	RX FIFO error flag: 1: at least one data in RX FIFO has parity error, UART frame format error, or FIFO break. 1: there is no error in the data in RX FIFO. When the erroneous data in RX FIFO is the next data to be read and there are no errors in the other data in RX FIFO, reading this register clears it to 0.
6	TEMT	R	0x01	Transmission complete flag: 1: transmission completed, both TX FIFO and shift register are empty. 0: transmission not completed
5	THRE	R	0x01	When PTIME and FIFO are both enabled, the TX FIFO full flag is controlled by the thresholds set in UART_FCR[5:4], and THRE interrupt will no longer be controlled. 1: TX FIFO full 0: TX FIFO not full Otherwise, TX FIFO empty flag: 1: TX FIFO empty 0: TX FIFO not empty
4	BI	R	0x0	Break interrupt flag: 1: break signal received

Bit	Name	Attribute	Reset Value	Description
				0: break signal not received In UART mode, this bit is set whenever the serial input (sin) remains in a logic “0” state for a duration exceeding the sum of the start time + data bits + parity bit + stop bits. In infrared mode, this bit is set whenever the serial input (SIR_in) remains in a continuous logic “0” state for a duration exceeding the sum of the start time + data bits + parity bit + stop bits. Reading this register or the RBR register will clear this bit to 0.
3	FE	R	0x0	Frame format error flag: 1: frame format error 0: no frame format error Reading this register or the RBR register will clear this bit to 0.
2	PE	R	0x0	Parity error flag: 1: parity error 0: no parity error Reading this register or the RBR register will clear this bit to 0.
1	OE	R	0x0	RX FIFO overflow flag: 1: RX FIFO overflow 0: no RX FIFO overflow Reading this register clears it to 0.
0	DR	R	0x0	The RFNE function of the UART_USR register is redundant when the FIFO is enabled, corresponding to the non-empty flag of RBR register or RX FIFO: 1: RX FIFO not empty 0: RX FIFO empty

29.5.11 Flow Status Register (UART1_MSR)

Offset address: 0x18

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:5	RSV	-	-	Reserved
4	CTS	R	0x0	CTS flag bit: 1: There is CTS request. 0: There is no CTS request.
3:0	RSV	-	-	Reserved

29.5.12 Low-power Baud Rate Division Low-order Register (UART1_LPDLL)

Offset address: 0x20

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:8	RSV	-	-	Reserved
7:0	LPDLL	R/W	0x0	Under low power conditions, the low byte of the baud rate division register must be set for a baud rate of 115200. This register is only effective when configured in SIR low power receive mode. This field is accessible only when the DLAB bit of UART_LCR is set to 1. The calculation formula is: Low baud rate = f_{clk} / (16 * {LPDLH, LPDLL})

29.5.13 Low-power Baud Rate Division High-order Register (UART1_LPDLH)

Offset address: 0x24

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:8	RSV	-	-	Reserved

Bit	Name	Attribute	Reset Value	Description
7:0	LPDLH	R/W	0x0	Under low power conditions, the high byte of the baud rate division register must be set for a baud rate of 115200. This register is only effective when configured in SIR low power receive mode. This field is accessible only when the DLAB bit of UART_LCR is set to 1. This bit is used to configure SIR mode receive detection. The calculation formula is: Low baud rate = f_{clk} / (16 * {LPDLH, LPDLL})

29.5.14 Status Register (UART1_USR)

Offset address: 0x7C

Reset value: 0x0000 0006

Bit	Name	Attribute	Reset Value	Description
31:5	RSV	-	-	Reserved
4	RFF	R	0x0	RX FIFO full flag: 1: RX FIFO full 0: RX FIFO not full
3	RFNE	R	0x0	RX FIFO non-empty flag: 1: RX FIFO not empty 0: RX FIFO empty
2	TFE	R	0x01	TX FIFO empty flag: 1: TX FIFO empty 0: TX FIFO not empty
1	TFNF	R	0x01	TX FIFO non-full flag: 1: TX FIFO not full 0: TX FIFO full
0	BUSY	R	0x0	1: UART is transmitting. 0: UART is idle.

29.5.15 TX FIFO Data Count Register (UART1_TFL)

Offset address: 0x80

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:5	RSV	-	-	Reserved
4:0	TFL	R	0x0	Data bit count in TX FIFO.

29.5.16 RX FIFO Data Count Register (UART1_RFL)

Offset address: 0x84

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:5	RSV	-	-	Reserved
4:0	RFL	R	0x0	Data bit count in RX FIFO.

29.5.17 Suspend TX Transmission (UART1_HTX)

Offset address: 0xA4

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:1	RSV	-	-	Reserved
0	Halt TX	R/W	0x0	This register is used to pause transmission, allowing the master to fill the TX FIFO: 0: TX suspending disabled 1: TX suspending enabled

29.5.18 DMA Software Acknowledgment (UART1_DMASA)

Offset address: 0xA8

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:1	RSV	-	-	Reserved
0	DMASA	W	0x0	When the transmission needs to be terminated due to an error, this register is used to execute a DMA software ACK, and this bit will be automatically cleared.

29.5.19 Fractional Frequency Division Register (UART1_DLF)

Offset address: 0xC0

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:6	RSV	-	-	Reserved
5:0	DLF	R/W	0x0	Fractional frequency division register: the fractional part of baud rate is DLF/64.

29.5.20 Receive Address Matching Register (UART1_RAR)

Offset address: 0xC4

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:8	RSV	-	-	Reserved
7:0	RAR	R/W	0x0	Receive address matching register

29.5.21 Transmit Address Matching Register (UART1_TAR)

Offset address: 0xC8

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:8	RSV	-	-	Reserved
7:0	TAR	R/W	0x0	Transmit address matching register

29.5.22 LINE Control Extension Register (UART1_LCRE)

Offset address: 0xCC

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:4	RSV	-	-	Reserved
3	TRANSMIT_MODE	R/W	0x0	9-bit transmit mode selection, writable only in UART idle state: 1: TX FIFO is of 9 bits, and the address and data

Bit	Name	Attribute	Reset Value	Description
				indication flags come from TX FIFO. 0: TX FIFO is of 8 bits, the transmitted address is determined by the SEND_ADDR and UART_TAR bits, and the transmitted data comes from TX FIFO.
2	SEND_ADDR	R/W	0x0	Transmit address matching enable bit: 1: in 9-bit mode, when the 9th bit in the UART frame is 1 (address indication), UART will transmit the data in the UART_TAR register. 0: in 9-bit mode, when the 9th bit in the UART frame is 0 (data indication), UART will transmit the data in FIFO. This bit will be automatically cleared after the transmission is completed.
1	ADDR_MATCH	R/W	0x0	Receive data address matching mode enable: 1: address matching mode enabled 0: address matching mode disabled This bit is valid only when the DLS_E bit is 1.
0	DLS_E	R/W	0x0	This bit is writable only when UART is idle: 1: 9-bit mode enabled 0: frame format determined by the DLS bit

29.6 Operation Procedure

29.6.1 UART Initialization

1. Enable the clocks for the corresponding GPIO pins and configure the pins for UART_TX and UART_RX alternate functions.
2. Configure UART module clock of the system configuration register.
3. Configure the UART1_DLF register to set the fractional frequency division.
4. Set UART1_LCR[7] to 1, and configure the UART1_DLH and UART1_DLL registers to set the UART baud rate.

5. Configure the UART1_LCR register to set the parity, number of stop bits, data frame length, and set DLAB to 0.
6. Configure the UART1_FCR register to enable FIFO.
7. Configure the UART1_IER register to enable the corresponding UART interrupts.

29.6.2 UART Transmission Process

1. Before transmitting data, the software can configure the baud rate parameter, parity type and data frame format.
2. Set UART1_LCR [7] to 0.
3. Write the first data byte to the UART1_THR register.
4. Query the transmission complete flag UART1_LSR[6], if UART1_LSR[6] = 1, the current data transmission is completed.
5. Write the next data byte to UART1_THR.

29.6.3 UART Reception Process

1. Before transmitting data, the software can configure the baud rate parameter, parity type and data frame format.
2. For data reception, query the UART1_USR flag or wait for an interrupt. If UART1_USR[3] = 1, it means the RX FIFO is not empty; then read the data from UART1_RBR, after which the corresponding flag will be automatically cleared.
3. If there is an error in data reception, wait for the interrupt or query the UART1_LSR register flag bit to determine the error type and execute corresponding error handling, after which the software clears the error flag bit.
4. Continue to receive data.

30 Low-power Universal Asynchronous Receiver Transmitter (LPUART0 & LPUART1)

30.1 Overview

LPUART is a low-power UART with communication up to 9600 baud/s under a 32-kHz clock and up to 115200 baud/s under the APB0 clock.

30.2 Main Features

- Asynchronous data transfer
- Standard UART frame format
  - 1 start bit
  - Programmable data word length: 7 or 8 bits
  - Odd parity bit, even parity bit or no parity bit
  - Configurable stop bit: 1 bit or 2 bits
- From 300 baud/s to 9600 baud/s using a 32.768 kHz XTL or a 32 kHz RCL clock source
- Up to 115200 Hz under the APB0 clock source
- Programmable data polarity
- Interrupt flags
  - Receive buffer full
  - Receive buffer overflow
  - Receive frame format error

- Receive parity bit error
- Start detection
- Data matching
- Transmit buffer empty
- Transmission complete flag

- Wakeup on low-power mode
  - Wakeup on RXD falling edge interrupt
  - Wakeup on start bit detection
  - Wakeup on 1-byte data receiving
  - Wakeup on 1-byte data matching
- An external low-speed clock must be used for meeting the clock accuracy.

30.3 System Block Diagram

30.3.1 Block Diagram

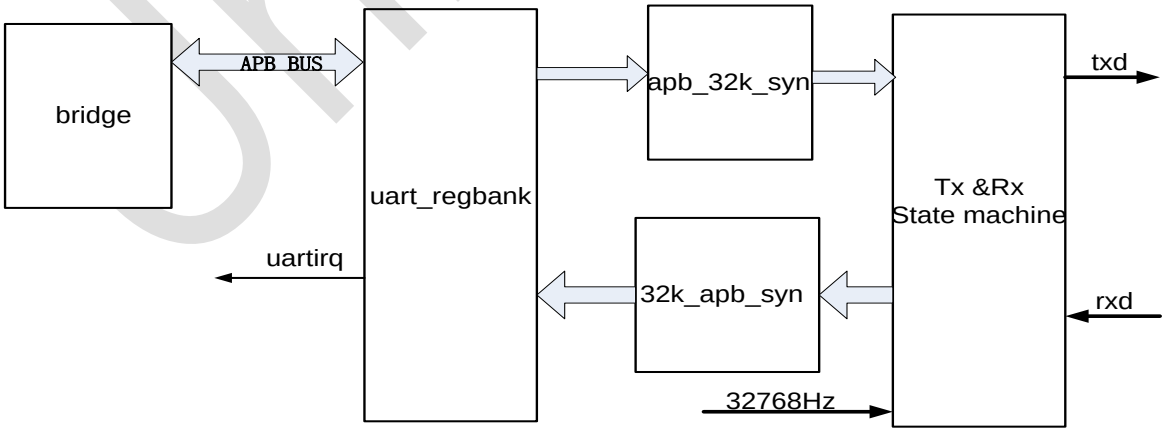


Figure 30-1: LPUART Block Diagram

30.3.2 Interface Timing

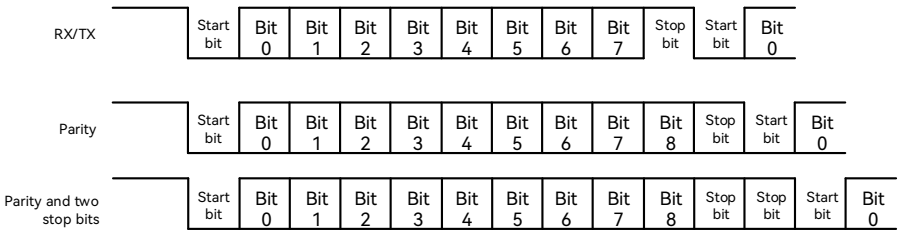


Figure 30-2: LPUART Interface Timing Diagram

30.3.3 Reception Timing

Since the LPUART clock is not an integer multiple of the baud rate, using a fixed division factor would introduce cumulative errors. To ensure accurate sampling, reception alternates between 3 and 4 division factors, ensuring that each bit is sampled once at its midpoint. Whether each bit is divided by 3 or 4 is controlled by the MCTL register. For example:

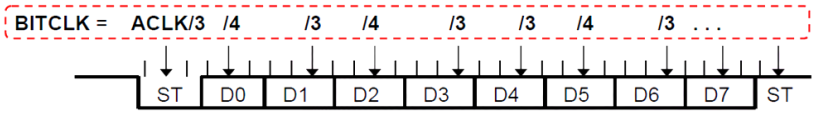


Figure 30-3: LPUART Reception Timing Diagram

30.3.4 Transmission Timing

Similar to LPUART reception, since the LPUART clock is not an integer multiple of the baud rate, using a fixed division factor would also introduce cumulative errors during transmission. Therefore, a 3 and 4 division alternating scheme is used for transmission as well. Whether each bit is divided by 3 or 4 is controlled by the MCTL register. For example:

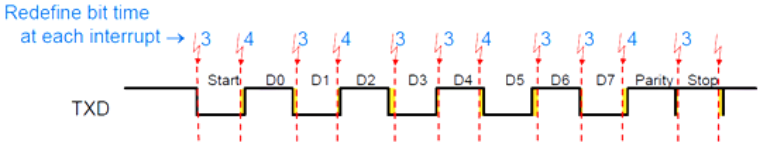


Figure 30-4: LPUART Transmission Timing Diagram

30.4 Pin Description

Table 30-1: LPUART Pin Description

Function Pin	Alternate Function Pin	Direction	Functional Description
LPUART0_TX	PA4, PB11, PA1, PB6, PB10, PC4, PC12, PC10	Output	Transmitting data
LPUART0_RX	PC2, PB10, PA0, PA3, PB7, PB11, PC5, PC11	Input	Receiving data
LPUART1_TX	PC4, PC10, PD8, PB2, PB12, PB0, PC12	Output	Transmitting data
LPUART1_RX	PC5, PC11, PD9, PD10, PB0, PA8, PB8, PB13, PA2	Input	Receiving data

Note: The system can be awakened from standby mode through the PC2 pin. The PB10 pin cannot be used in standby mode.

30.5 Functional Description

30.5.1 Transmission Mode

LPUART can convert parallel data into serial data for transmission. The format of the transmitted data frame can be configured to include 1 start bit, 7 or 8 data bits, odd parity, even parity, or no parity bit, and 1 or 2 stop bits. The transmission baud rate ranges from 300 to 9600.

30.5.2 Reception Mode

LPUART can convert serial data into parallel data for reception and supports waking the chip from low-power mode through LPUART receive events. It can also receive data while in Sleep/Stop mode. The transmission baud rate ranges from 300 to 9600 Hz.

30.6 Register Description

LPUART0 register base address: 0x40B0_7000

LPUART1 register base address: 0x40B0_1000

Table 30-2: List of LPUART Registers

Offset Address	Name	Description
0x00	LPUART_RXD	Receive data register
0x04	LPUART_TXD	Transmit data register
0x08	LPUART_STA	Status register
0x0C	LPUART_CON	Control register
0x10	LPUART_IF	Interrupt flag register
0x14	LPUART_BAUD	Baud rate register
0x18	LPUART_EN	Transmit/receive enable register
0x1C	LPUART_COMPARE	Data matching register
0x20	LPUART_MODU	Baud rate modulation control register

Registers are detailed in the following sections.

30.6.1 Receive Data Register (LPUART_RXD)

Offset address: 0x00

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:8	RSV	-	-	Reserved
7:0	LPURXD	R	0x0	Receive data register

30.6.2 Transmit Data Register (LPUART_TXD)

Offset address: 0x04

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:8	RSV	-	-	Reserved
7:0	LPUTXD	W	0x0	Transmit data register

30.6.3 Status Register (LPUART_STA)

Offset address: 0x08

Reset value: 0x0000 00C0

Bit	Name	Attribute	Reset Value	Description
31:8	RSV	-	-	Reserved
7	TC	R	0x1	The transmission complete flag is set when the transmission of a frame of data is completed and the TX buffer is empty. 1: transmission completed 0: transmission not completed
6	TXE	R	0x1	TX buffer empty flag, can be set by hardware and automatically cleared by software via writing data to it. 1: buffer empty 0: buffer not empty
5	START	R/W	0x0	Start bit detection flag, can be cleared by writing 1.
4	PERR	R/W	0x0	Parity error bit, can be cleared by writing 1.
3	FERR	R/W	0x0	Frame format error bit, can be cleared by writing 1.
2	RXOV	R/W	0x0	Receive buffer overflow bit, can be cleared by writing 1.
1	RXF	R	0x0	Receive buffer full bit, can be cleared by reading the LPUDATA register.
0	MATCH	R/W	0x0	Data matching flag, indicating that the data in receive buffer matches the compare register, can be cleared by writing 1.

30.6.4 Control Register (LPUART_CON)

Offset address: 0x0C

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:13	RSV	-	-	Reserved
12	TXPOL	R/W	0x0	Data transmission polarity: 0: non-inverted 1: inverted
11	TCIE	R/W	0x0	TX complete interrupt enable:

Bit	Name	Attribute	Reset Value	Description
				0: disabled 1: enabled
10	TXIE	R/W	0x0	TX buffer empty interrupt enable: 0: TX buffer empty interrupt disabled 1: TX buffer empty interrupt enabled
9	NEDET	R/W	0x0	Enable bit for sampling at falling edge: 0: do not use the falling edge of the 32k clock to detect the start bit. 1: use the falling edge of the 32k clock to detect the start bit.
8	PAREN	R/W	0x0	Parity bit enable: 0: data frame without parity bit 1: data frame with parity bit
7	PTYP	R/W	0x0	Parity type: 0: even parity 1: odd parity
6	SL	R/W	0x0	Stop bit length: 0: 1 bit 1: 2 bits
5	DL	R/W	0x0	Data length: 0: 8 bits 1: 7 bits
4	RXPOL	R/W	0x0	Receive polarity: 0: non-inverted 1: inverted
3	ERRIE	R/W	0x0	Error interrupt enable: 0: disabled 1: enabled
2	RXIE	R/W	0x0	Receive interrupt enable: 0: disabled 1: enabled
1:0	RXEV	R/W	0x0	Configure to determine under which of the following events to provide the CPU with a receive interrupt: 00: start bit detects a wake-up 01: 1-byte data reception completed 10: received data matching succeeded

Bit	Name	Attribute	Reset Value	Description
				11: a wake-up detected at falling edge

30.6.5 Interrupt Flag Register (LPUART_IF)

Offset address: 0x10

Reset value: 0x0000 0004

Bit	Name	Attribute	Reset Value	Description
31:4	RSV	-	-	Reserved
3	TC_IF	R/W1C	0x0	Transmission complete interrupt flag: 1: interrupt generated upon transmitting a frame of data 0: no interrupt generated
2	TX_IF	R/W1C	0x1	Transmit buffer empty interrupt flag: 1: interrupt generated after transmit buffer is empty 0: no interrupt generated
1	RXNEG_IF	R/W1C	0x0	RXD falling edge interrupt flag: 1: interrupt generated 0: no interrupt generated
0	RX_IF	R/W1C	0x0	Reception complete interrupt flag: 1: interrupt generated upon receiving a frame of data 0: no interrupt generated

30.6.6 Baud Rate Register (LPUART_BAUD)

Offset address: 0x14

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:5	RSV	-	-	Reserved
4	CLK_SEL	R/W	0	UART baud rate clock source selection: 0: default 32k low frequency clock (XTL/RCL) 1: system high frequency clock source selected
3	RSV	-	-	Reserved
2:0	BAUD	R/W	0x0	Baud rate (bps):

Bit	Name	Attribute	Reset Value	Description
				000: 9600 001: 4800 010: 2400 011: 1200 100: 600 101/110/111: 300

30.6.7 Transmit/Receive Enable Register (LPUART_EN)

Offset address: 0x18

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:2	RSV	-	-	Reserved
1	TXEN	R/W	0x0	Transmit enable: 0: LPUART transmission disabled 1: LPUART transmission enabled After the CPU writes 1 to enable, this register shall be read repeatedly until 1 is read before the next operation.
0	RXEN	R/W	0x0	Receive enable: 0: LPUART reception disabled 1: LPUART reception enabled After the CPU writes 1 to enable, this register shall be read repeatedly until 1 is read before the next operation.

30.6.8 Data Matching Register (LPUART_CMPARE)

Offset address: 0x1C

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:8	RSV	-	-	Reserved
7:0	COMPARE	R/W	0x0	Compare the data, if RXEV = 10, the reception complete interrupt will be triggered when the data in the receive buffer matches COMPARE.

30.6.9 Modulation Control Register (LPUART_MODU)

Offset address: 0x20

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:12	RSV	-	-	Reserved
11:0	MCTL	R/W	0x0	Modulation control signal for each bit of LPUART

30.7 Operation Procedure

30.7.1 Initialization

1. Configure the corresponding GPIO pins to be multiplexed as LPUART_TX and LPUART_RX.
2. Configure the PMU function clock control register PMU_FCCR[0] to enable the LPUART clock.
3. Configure the baud rate register (LPUBAUD) to set the baud rate.
4. Configure the modulation control register (MCTL) to select the appropriate modulation parameters according to the baud rate.
5. Configure the control register (LPUCON) to set the data frame format, parity, and stop bit length.

30.7.2 Reception Process

1. Configure the receive enable register LPUART_EN[0]=1 to enable data reception.
2. Configure the control register LPUART_CON[1:0] to enable the receive interrupt.
3. Wait for the receive interrupt to trigger, then read the data from the receive data register (LPURXD).
4. Write 1 to the interrupt flag register LPUART_IF[0] to clear the receive complete interrupt flag.
5. Continue receiving data by repeating steps 2, 3 and 4.

30.7.3 Transmission Process

1. Configure the transmit/receive enable register LPUART_EN[1]=1 to enable data transmission.
2. Write the data to be transmitted into the transmit data register (LPUTXD).
3. Wait for the status flag register LPUART_STA[6] to be set to 1.
4. Continue transmitting data by repeating steps 2 and 3.

30.7.4 Suggested Configuration for Modulation Control Register

The modulation control register MCTL shall be reasonably configured according to the different communication baud rates. The suggested configuration parameters are as follows:

Table 30-3: MCTL Configuration Parameter Table for LPUART_MODU

Baud	MCTL											
	Bit 0 (Start)	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7	Bit 8	Bit 9	Bit 10	Bit 11
9600	0	1	0	0	1	0	1	0	1	0	0	1
4800	1	1	0	1	1	1	1	1	0	1	1	1
2400	1	1	0	1	1	0	1	1	0	1	1	0
1200	0	1	0	0	1	0	0	1	0	0	1	0
600	0	1	1	0	1	0	1	1	0	1	1	0
300	0	1	0	0	0	0	1	0	0	0	0	1

The above parameter table assumes that LPUART operates at an exact 32.768 kHz clock. Working with RCLP will introduce additional errors and may require fine-tuning the baud rate modulation scheme to obtain better communication results.

30.7.5 Using the System High-frequency Clock Source

1. Configure LPUART_BUAD[4]=1 to select the system high-frequency clock for LPUART.
2. Configure RCM_CFGR2[9]=1 to enable the LPUART system high-frequency clock.
3. Configure RCM_CFGR2[8:0] as the division factor LPUART_DIV, so the input clock for LPUART, LPUART_CLK, is: LPUART_CLK = PCLK0 / (LPUART_DIV + 1).
4. Changing LPUART_CLK will subsequently change the output baud rate. For example, if LPUART outputs a baud rate of 9600 with RCL as the clock, if LPUART_CLK is twice that of RCL, the LPUART will output a baud rate of 19200 with the same configuration when using LPUART_CLK as the clock source.

31 Universal Synchronous / Asynchronous Receiver Transmitter (USART6 & USART7)

31.1 Overview

USART provides a complete full-duplex universal synchronous / asynchronous serial link. To ensure the highest standard, the data frame format can be programmed in a wide range (data length, parity, number of stop bit, etc.) This receiver implements detection of parity error, frame error and overflow error. Receiver timeout allows handling of frames with variable lengths, and transmitter time protection facilitates communication with slow remote devices. Multiprocessor communication can also be supported by address bit processing in reception and transmission.

31.2 Main Features

- Programmable baud rate generator
- 5-bit to 9-bit full-duplex synchronous or asynchronous serial communication
  - Configurable stop bits: 1, 1.5 or 2 in asynchronous mode; and 1 or 2 in synchronous mode
  - Parity generation and error detection
  - Detection of frame error and overflow error
  - MSB first or LSB first for data transfer
  - Break generation and detection
  - 8 or 16 times of oversampling receiver frequency
  - Receiver timeout and transmitter time protection
  - Multipoint mode with address generation and detection

- IrDA modulation / demodulation
  - Communication rate up to 115.2 kbps
- SPI mode
  - Master or slave
  - Programmable serial clock phase and polarity
  - The frequency of SPI serial clock (SCK) is up to MCK/6 of internal clock frequency.
- LIN mode
  - Complying with LIN1.3 and LIN2.0 protocols
  - Master or slave
  - Handling frames of up to 256 data bytes
  - The response data length can be configured by identifier or defined automatically.
  - Self-synchronization in slave node configuration
  - Automatic processing and verification of “Synch Break” and “Synch Field”
  - “Synch Break” will be detected even if it is partially overlapped with data bytes.
  - Automatic identifier parity calculation, sending and verification
  - Parity sending and verification can be disabled
  - Automatic checksum calculation, sending and verification
  - Checksum sending and verification can be disabled
  - Supporting both “Classic” and “Enhanced” checksum types
  - Complete LIN error checking and reporting
- DMA operation

31.3 Pin Description

Table 31-1: USART6 & USART7 Pin Description

Function Pin	Alternate Function Pin	Direction	Functional Description
USART6_CTS	PA0, PC0, PA15	Input	SPI slave mode chip select input (hardware flow control mode transmitting enable signal in UART mode is not supported)
USART6_RTS	PA1, PC1, PB3	Output	SPI master mode chip select output (hardware flow control mode transmitting request signal in UART mode is not supported)
USART6_CK	PA4, PC4, PA14, PB1	Input/output	Clock signal
USART6_TX	A2, PC2, PA8, PA14, PC0, PB4, PD14	Input/output	Transmitting data
USART6_RX	PA3, PC3, PA0, PA15, PB0, PB5, PC1, PD15	Input/output	Receiving data
USART7_CTS	PB6, PC5, PA6, PB13	Input	SPI slave mode chip select input (hardware flow control mode transmitting enable signal in UART mode is not supported)
USART7_RTS	PB7, PC9, PB1, PB14	Output	SPI master mode chip select output (hardware flow control mode transmitting request signal in UART mode is not supported)
USART7_CK	PB3, PC8, PA5, PB0, PB12, PC12	Input/output	Clock signal
USART7_TX	PA7, PB4, PC6, PB10, PB11, PC4, PC10	Input/output	Transmitting data
USART7_RX	PB5, PC7, PA5, PA6, PB10, PB11, PC5, PC11	Input/output	Receiving data

31.4 Functional Description

31.4.1 Baud Rate Generator

The clock source for the baud rate generator can be selected by configuring the USCLKS field in the mode register (USART_MR):

- Master clock
- Divided master clock
- External clock, valid on the SCK pin

The baud rate generator is based on a 16 prescaler and is programmed using the CD field in the baud rate generator register (USART_BRGR). If 0 is written to CD, the baud rate generator does not generate any clock. If 1 is written to CD, the prescaler is bypassed.

If the external SCK clock is selected, the duration of the low and high levels of the provided signal must be greater than the master clock (MCK) period. In USART mode, the signal provided on SCK must be at least 4.53 times lower than MCK, or 6 times lower in SPI mode.

31.4.1.1 Asynchronous Mode

In asynchronous mode, the clock is first divided by the CD bits in USART_BRGR. The generated clock is provided as the sampling clock to the receiver, and then divided by 16 or 8 according to the OVER bit in the USART_MR register.

The baud rate is calculated using the following formula:

$$\text{Baud rate} = \frac{\text{Selected clock source}}{(8\ \ (2-Over)\ \ CD)}$$

31.4.1.2 Fractional Baud Rate in Asynchronous Mode

The previously defined baud rate generator is limited by the fact that the output frequency varies only in integer multiples of the reference frequency. One solution to this problem is to integrate a high-resolution fractional N clock generator, where the fractional part is programmed by the FP bit in USART_BRGR. If FP is not 0, the fractional part is valid. This feature is only effective in standard USART mode. The fractional baud rate is calculated using the following formula:

$$\text{Baud rate} = \frac{\text{Selected clock source}}{(8\ \ (2-Over)\ \ \ (CD+\frac{FP}{8})\ \ )}$$

31.4.1.3 Synchronous Mode or SPI Mode

In synchronous mode, the clock is set by the CD bits in the USART_BRGR register:

$$\text{Baud rate} = \frac{\text{Selected clock source}}{CD}$$

In synchronous mode, if the external clock (USCLKS = 3) is selected, the clock is provided by the USART SCK pin signal; therefore, no prescaling is needed, and the value in USART_BRGR is invalid. The external clock frequency must be less than 1/3 of the system frequency. For the master in synchronous mode (USCLKS = 0 or 1, CLKO = 1), the maximum frequency limit for the receiver SCK is MCK / 3.

Whether the external clock or the internal clock divider (MCK/DIV) is selected, if the user wants to guarantee a 50:50 duty cycle for the signal on the SCK pin, the value of the CD field must be set to an even number. If the internal clock MCK is selected, even if the value of the CD field is odd, the baud rate generator will ensure a 50:50 duty cycle on the SCK pin.

31.4.2 Receiver and Transmitter Control

After a reset, the receiver is disabled. The user must enable the receiver by setting the control register USART_CR[4]. However, the receive register can be programmed before the receiver clock is enabled.

After a reset, the transmitter is also disabled. The user must enable the transmitter by setting the control register USART_CR[6]. However, the transmit register can be programmed before the transmitter clock is enabled.

The receiver and transmitter can be enabled together or separately.

At any time, the software can set USART_CR[2] and USART_CR[3] to reset the receiver or transmitter of USART. A software reset clears the status flags and resets the internal state machine, but the user-configured registers remain unchanged. Communication will immediately stop whether it is receiving or transmitting.

The user can also individually disable receiving or transmitting by setting USART_CR[5] and USART_CR[6]. If the receiver is disabled during character reception, the USART will wait until the current character reception is complete before stopping reception. If it is disabled during transmission, the USART will wait until the current character and the character stored in the USART_THR register have been transmitted. If time protection is set, it can be processed properly.

31.4.3 Synchronous and Asynchronous Modes

31.4.3.1 Transmitter Operation

The transmitter performs the same operations in both synchronous and asynchronous modes (synchronous = 0 or synchronous = 1). A start bit, up to 9 data bits, an optional parity bit, and up to two stop bits are sequentially output on the TXD pin at the falling edge of the serial

clock. The number of data bits is determined by the CHRL field and the MODE9 bit in the USART_MR register. If the MODE9 bit is set, the data bits will be 9 bits regardless of the CHRL field settings. The parity bit is configured by the PAR field in USART_MR and can be set for odd parity, even parity, space parity, mark parity, or no parity. The MSBF field configures which bit is sent first: writing 1 transmits the most significant bit first, while writing 0 transmits the least significant bit first. The number of stop bits is determined by the NBSTOP field. In asynchronous mode, 1.5 stop bits are supported.

- **Character transmitting**

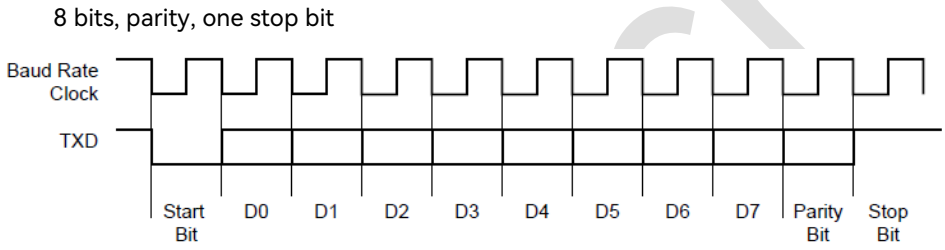


Figure 31-1: USART Character Transmitting

Characters are transmitted by writing to the transmit holding register (USART_THR). The transmitter has two status bits in the channel status register (USART_CSR): TXRDY (transmit ready) indicates that USART_THR is empty, and TXEMPTY indicates that all characters written to USART_THR have been processed. When the current character has been processed, and the last character written to USART_THR is transferred to the transmitter shift register, USART_THR becomes empty, and TXRDY goes high.

When the transmitter is disabled, both TXRDY and TXEMPTY are low. When TXRDY is low, writing characters to USART_THR is invalid, and the data written is lost.

- **Transmitter status**

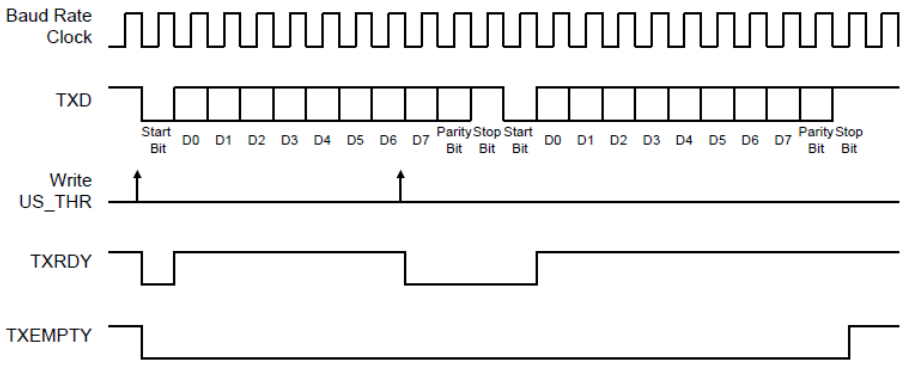


Figure 31-2: USART Transmitter Status

31.4.3.2 Manchester Encoding

When using Manchester encoding, characters transmitted via USART are encoded in the Bi-phase Manchester Encoding II format. To use this mode, the MAN bit field in the USART_MR register must be set to 1. Depending on the polarity configuration, a logic level (0 or 1) is encoded as a transition from 0 to 1 or from 1 to 0 for transmission. Thus, level transitions always occur at the midpoint of each bit time. Although it occupies more (twice as much) bandwidth than the original NRZ signal, it allows for better error control since the expected input must change at half a bit clock time. An example of a Manchester encoding sequence: with the default polarity encoder, the byte 0xB1 or 10110001 would be encoded as 1001101001010110.

- **NRZ code to Manchester code**

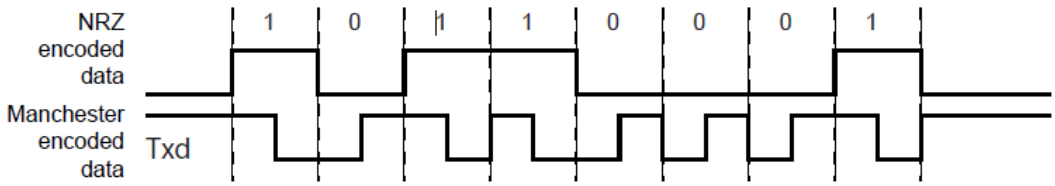


Figure 31-3: NRZ Code to Manchester Code

Manchester encoded characters can also be encapsulated by adding a configurable preamble signal and a frame start delimiter style. Depending on the configuration, the preamble signal is a training sequence composed of a predefined pattern, with a programmable length of 1 to 15

bit times. If the length of the preamble signal is set to 0, no preamble signal waveform will be generated. Various sequences can be selected for the preamble signal mode: ALL_ONE, ALL_ZERO, ONE_ZERO or ZERO_ONE, which are written into the TX_PP field of the USART_MAN register, while TX_PL is used to set the length of the preamble signal. The following figure illustrates and defines the valid modes. For increased flexibility, the encoding scheme can be configured using the TX_MPOL field of the USART_MAN register. If TX_MPOL is set to 0 (default), logic 0 is encoded by a transition from 0 to 1, and logic 1 is encoded by a transition from 1 to 0. If TX_MPOL is set to 1, logic 1 is encoded by a transition from 0 to 1, and logic 0 is encoded by a transition from 1 to 0.

- **Preamble signal mode with default polarity**

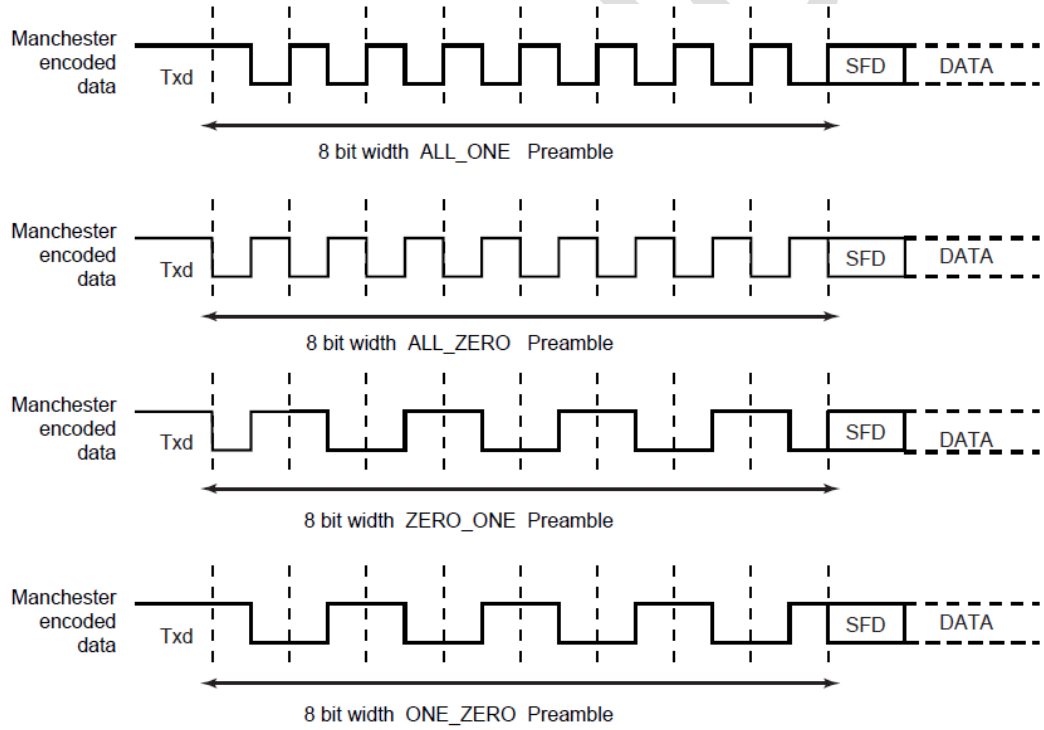


Figure 31-4: Preamble Signal Mode

A frame start delimiter can be configured using the ONEBIT bit in the USART_MR register, consisting of a user-defined pattern to indicate the start of valid data. Figure 31-4 presents these patterns. If the frame start delimiter, i.e., the start bit, is a bit (ONEBIT set to 1), detecting a Manchester encoded logic 0 indicates that a new character is being transmitted

on the serial line. If the frame start delimiter is a synchronization pattern, or a synchronization (ONEBIT set to 0) symbol, a new character is considered to start when a sequence of 3 bit times is transmitted serially on the line. When the transition occurs in the middle of the second bit time, the synchronization symbol waveform itself is an invalid Manchester waveform. There are two different synchronization modes: command synchronization symbol and data synchronization symbol. The command synchronization symbol uses a high level for 1.5 bit times to represent a '1'; then it switches to a low level for 1.5 bit times to indicate a second '1'. If the MODSYNC bit in the USART_MR register is set to 1, then the next character will be a command synchronization symbol; if set to 0, the next character will be a data synchronization symbol. When using DMA, the MODSYNC field can be updated by modifying a character in memory. To allow this mode, the VAR_SYNC field in the USART_MR register must be set to 1. This way, the MODSYNC in USART_MR is ignored, and the synchronization symbol is configured using the TXSYNH field in USART_THR. The USART character format will be modified to include the synchronization symbol information.

- **Frame start delimiter**

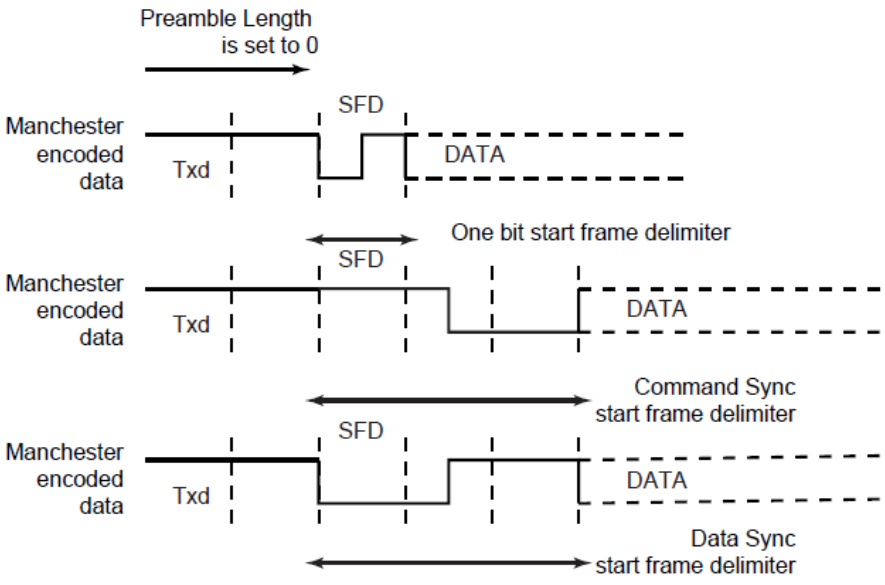


Figure 31-5: Frame Start Delimiter

31.4.3.3 Drift Compensation

Drift compensation is effective only in 16x oversampling mode. A hardware repair system allows greater clock drift. This hardware system can be enabled by setting the DRIFT bit in USART_MAN to 1. If the edge (either rising or falling) of RXD is aligned with the expected edge of the 16x clock cycle, it is considered normal operation with no correction. If an RXD event occurs within 2 to 4 clock cycles before the expected edge, the current cycle is shortened by one clock cycle. If an RXD event occurs within 2 to 3 clock cycles after the expected edge, the current cycle is extended by one clock cycle. These intervals are treated as drift, and corrections are performed automatically.

- **Bit synchronization**

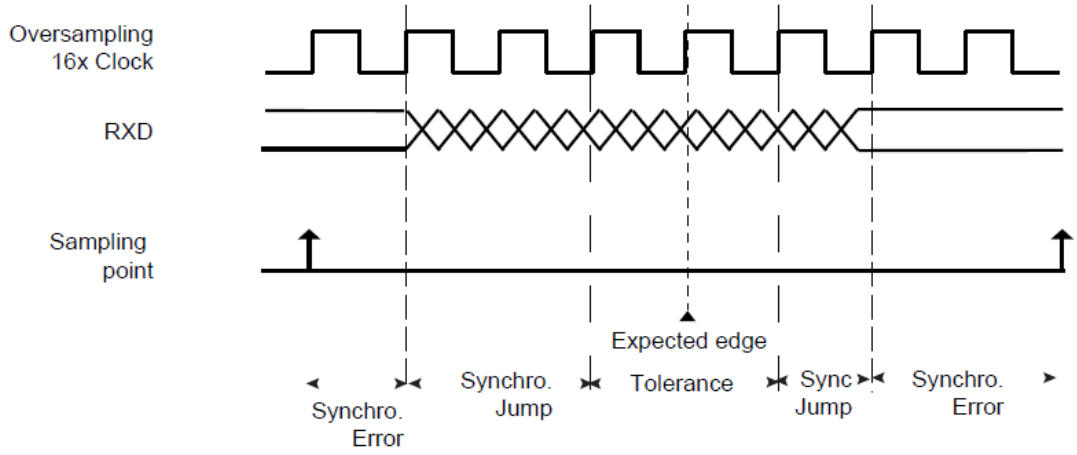


Figure 31-6: Bit Synchronization Diagram

31.4.3.4 Asynchronous Receiver

When the USART operates in asynchronous mode (SYNC = 0), the receiver oversamples the RXD input line. The oversampling frequency is set to either 16 or 8 times the baud rate, as configured by the OVER bit in USART_MR.

The receiver samples the RXD line. If the sample values are all 0 within 0.5 bit time, it indicates that the start bit is detected, after which data bits, parity bits and stop bits are sampled at the bit rate.

If the oversampling frequency is 16 times the baud rate (OVER = 0), a continuous sequence of 8 samples yielding 0 indicates the detection of a start bit. Subsequently, every 16 sampling clock cycles, the subsequent data bits, parity bits and stop bits are sampled in sequence. If the oversampling frequency is 8 times the baud rate (OVER = 1), a continuous sequence of 4 samples yielding 0 indicates the detection of a start bit. Then, every 8 sampling clock cycles, the data bits, parity bits and stop bits are sampled in sequence.

The receiver sets the data bit count, first transmitted bit and parity mode fields to be the same as those of the transmitter, specifically CHRL, MODE9, MSBF and PAR. The number of stop bits is irrelevant for the receiver, as it only acknowledges 1 stop bit regardless of the value in the NBSTOP field. Thus, resynchronization can occur between the transmitter and the receiver. Additionally, after detecting the stop bit, the receiver begins searching for a new start bit, enabling resynchronization even when the transmitter has only 1 stop bit.

The following figures illustrate the start bit detection and character reception when the USART operates in asynchronous mode.

- **Asynchronous start bit detection**

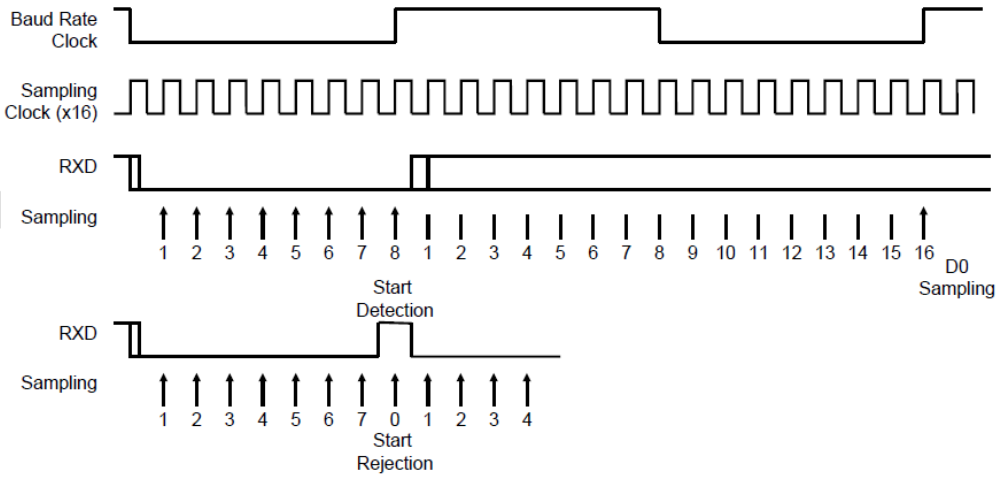


Figure 31-7: Asynchronous Start Bit Detection

- **Character reception**

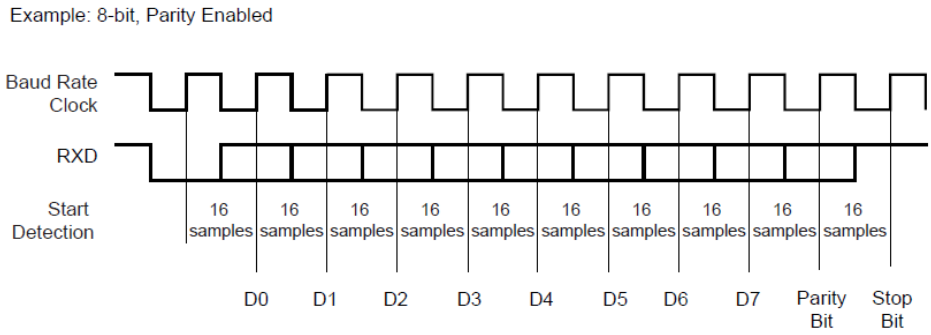


Figure 31-8: Character Reception

31.4.3.5 Manchester Decoder

When the MAN field in the USART_MR register is set to 1, the Manchester decoder is enabled. The decoder performs detection of the preamble signal and frame start delimiter. One input line is specifically designated for the Manchester encoded data input.

An optional preamble signal sequence can be defined, with its length also being user-defined and completely independent of the transmitter. The length of the preamble signal sequence can be configured by setting the RX_PL bit in the USART_MAN register. If the length is set to 0, preamble signal detection is disabled. Additionally, the input stream polarity can be set through the RX_MPOL field in the USART_MAN register. Depending on application requirements, the preamble signal mode can be defined by the RX_PP field in USART_MAN to match. Unlike the preamble signal, the frame start delimiter is shared in the Manchester encoder/decoder. Therefore, if ONEBIT is set to 1, only 0 Manchester encoding can be detected and treated as a valid frame start delimiter. If ONEBIT is set to 0, only the synchronous mode can be detected and treated as a valid frame start delimiter.

The decoder operates by detecting transitions in the input stream. If RXD is sampled low within 1/4 of a bit time, it is considered the detection of a start bit, as illustrated in the following figure. The sampling pulse rejects device requests.

- **Asynchronous start bit detection**

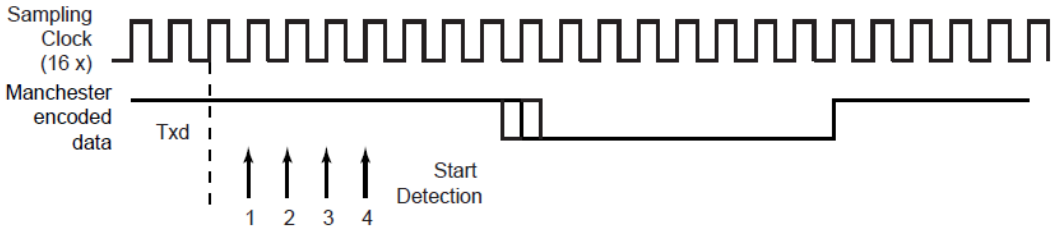


Figure 31-9: Asynchronous Start Bit Detection

The receiver is activated and begins detecting the preamble signal and frame separator, sampling data at 1/4 and 3/4 of the cycle. If a valid preamble signal or frame start delimiter is detected, the receiver will continue decoding using the same synchronization clock. If the data stream does not match a valid mode or valid frame start delimiter, the receiver will resynchronize at the next edge. The minimum time threshold for estimating bit values is 3/4 bit time. If a valid frame start delimiter is detected followed by a valid preamble signal (if used), the input stream will be decoded into NRZ encoded data and sent for USART processing. Figure 31-10 illustrates a case of mismatch in Manchester encoding mode. When the input data stream is sent to the USART, the receiver can also detect violations of Manchester encoding. Violations occur when there is a missing level transition in the bit intervals. In this case, the MANE flag in the USART_CSR register is set to 1. The MANE flag can be cleared by setting the RSTSTA bit in the control register USART_CR to 1. The following figures depict an example of Manchester error detection during data transmission.

- **Preamble mismatch**

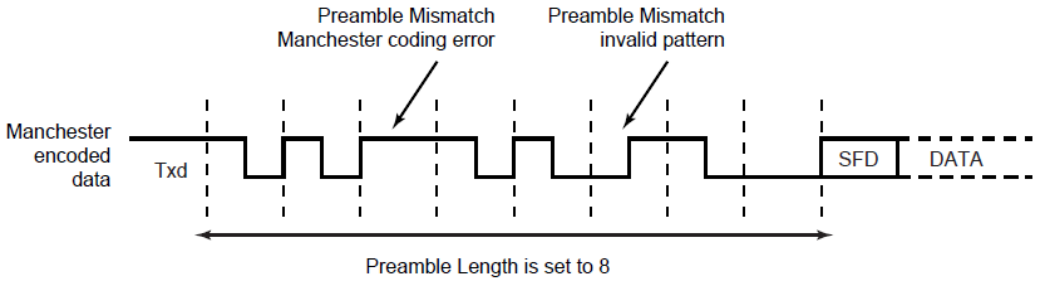


Figure 31-10 : Preamble Mismatch

- **Manchester error flag**

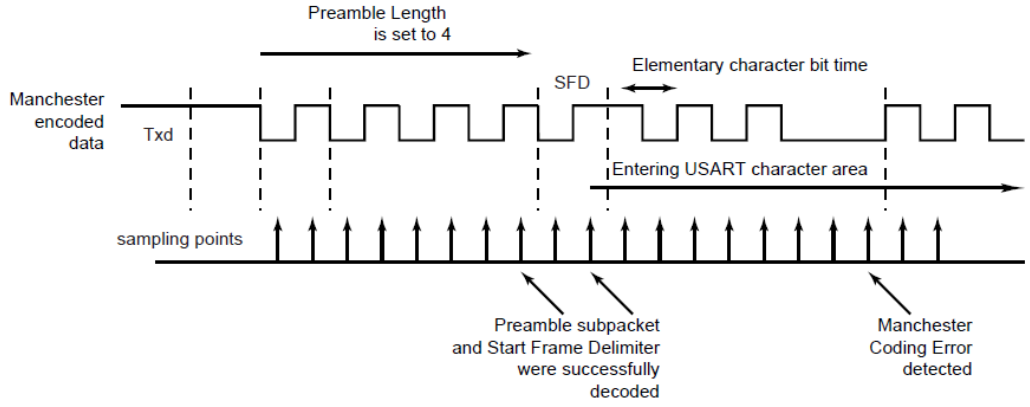


Figure 31-11: Manchester Error Flag

When the frame start delimiter is in synchronous mode (ONEBIT set to 0), command synchronization symbol and data synchronization symbol are supported. If a valid synchronization is detected, the received character is written into the RXCHR field of the USART_RHR register, while RXSYNH is updated. When the received character is a command, RXCHR is set to 1, and when the received character is data, RXCHR is set to 0. This mechanism alleviates and simplifies Direct Memory Access (DMA) since the character already contains its own synchronization field in the same register.

Since the decoder is configured to operate in single-pole mode, the first bit of the frame must be a level transition from 0 to 1.

31.4.3.6 Wireless Interface: Application of Manchester Encoding in USART

This section describes low-data-rate RF transmission systems and their integration with Manchester encoded USART. These systems are based on ICs for transmitters and receivers, supporting ASK and FSK modulation schemes.

The goal of the system is to achieve wireless full-duplex character transmission using two different frequency carriers. All configurations are illustrated in the following figure:

- **Manchester Encoded Character RF Transmission**

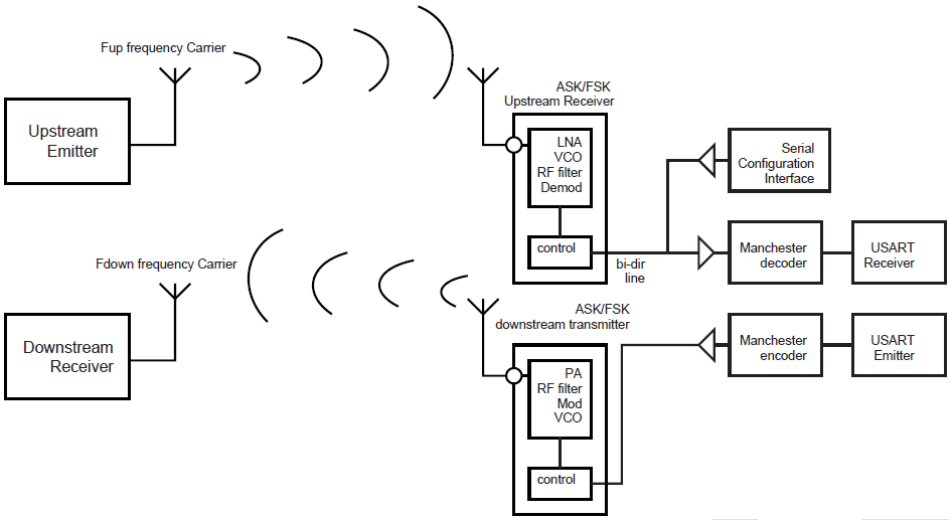


Figure 31-12: Manchester Encoded Character RF Transmission

The USART module is configured as a Manchester encoder/decoder. In the downlink communication, the Manchester-encoded characters are serially transmitted to the RF transmitter. This may also include user-defined preamble signals and a frame start delimiter. Typically, the preamble signal is used by the RF receiver to distinguish between valid data generated by the transmitter and noise signals. The Manchester data stream is then modulated. The following figure provides an example of an ASK modulation scheme. When the ASK modulator receives a logic high level, the power amplifier (PA) is activated, and an RF signal is transmitted at the downlink frequency. When the ASK modulator receives a logic low (0) level, the RF signal is turned off. If the FSK modulator is activated, it uses two different frequencies to transmit data. When sending a logic 1, the modulator outputs an RF signal at frequency F0; when sending a logic 0, the frequency switches to F1, as shown in the figures below.

For the receiving end, another carrier frequency is used. The RF receiver performs bit detections to detect the demodulated data stream. If a valid pattern is detected, the receiver switches to receiving mode, and the demodulated data stream is sent to the Manchester decoder. Due to bit detection within the RF IC, the amount of data transmitted to the

controller can be reduced by a user-defined number of bits. The length of the Manchester preamble signal is defined according to the configuration of the RF IC.

- **ASK modulator output**

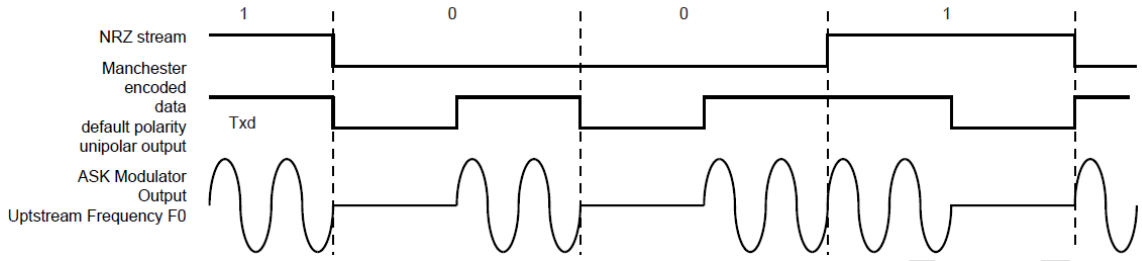


Figure 31-13: ASK Modulator Output

- **FSK modulator output**

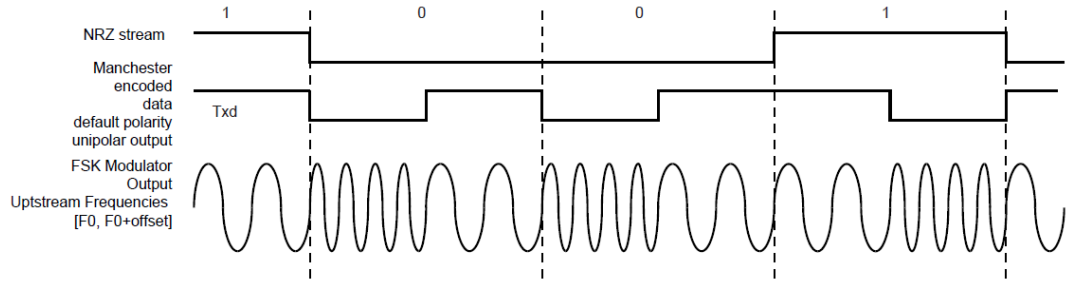


Figure 31-14: FSK Modulator Output

31.4.3.7 Synchronous Receiver

In synchronous mode (SYNC = 1), the receiver samples the RXD signal on the rising edge of each baud rate clock. If a low level is detected, it is determined to be a start bit; subsequently, all data bits, parity bits, and stop bits are sampled, after which the receiver will continue to wait for the next start bit. Synchronous mode provides high-speed transmission capability.

The configuration of fields and bits remains the same as in asynchronous mode. The following figure illustrates the character reception timing in synchronous mode.

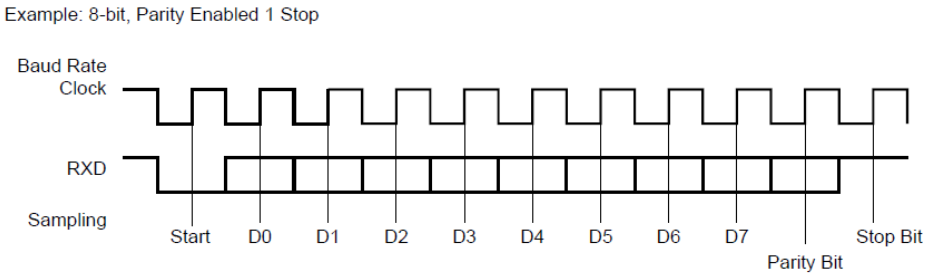


Figure 31-15: Character Reception Timing in Synchronous Mode

31.4.3.8 Receiver Operation

When a character reception is complete, it is transferred to the receive holding register (USART_RHR), and the RXRDY bit in the status register (USART_CSR) is set high. When RXRDY is set, indicating that a character has been received, the OVRE (overflow error) bit is set. The last character is transferred to USART_RHR, overwriting the previous character. Writing a 1 to the RSTSTA (reset status) bit in the control register (USART_CR) can clear the OVRE bit.

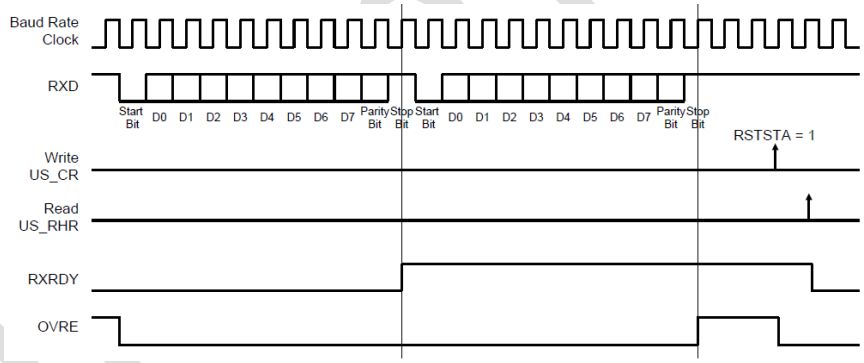


Figure 31-16: Receiver Status

31.4.3.9 Parity

By setting the PAR field in the mode register (USART_MR), the USART can support five different parity modes. The PAR field can also enable multidrop mode, as detailed in Section 31.4.3.10 Multidrop Mode. It supports parity bit generation and error detection.

If even parity is selected, when the number of ones sent by the transmitter is even, the parity bit generated by the parity bit generator is 0; when the number of ones is odd, the parity bit

generated is 1. Correspondingly, the receiver parity bit detector counts the received ones, and if the calculated parity bit differs from the sampled parity bit, an even parity error is reported. If odd parity is selected, when the number of ones sent by the transmitter is even, the parity bit generator produces a parity bit of 1; when the number of ones is odd, it produces a parity bit of 0. Correspondingly, the receiver parity bit detector also counts the received ones, and if the calculated parity bit differs from the sampled parity bit, an odd parity error is reported. When using mark parity, the parity bit generated by the parity bit generator is always 1 for all characters; if the sampled parity bit in the receiver is 0, the receiver parity bit detector reports a parity error. When using space parity, the parity bit generated is always 0 for all characters; if the sampled parity bit in the receiver is 1, the receiver parity bit detector reports a parity error. If parity is disabled, the transmitter does not generate a parity bit, and the receiver does not report parity errors.

The following table provides examples of the parity bits corresponding to the character 0x41 (ASCII character “A”) under different USART configurations. Since there are two bits set to 1, a “1” is added for odd parity and a “0” for even parity.

Table 31-2: Example of USART Parity Bits

Character	Hexadecimal	Binary	Parity Bit	Parity Mode
A	0x41	0100 0001	1	Odd
A	0x41	0100 0001	0	Even
A	0x41	0100 0001	1	Mark
A	0x41	0100 0001	0	Space
A	0x41	0100 0001	No	No

When the receiver detects a parity error, it sets the PARE (parity error) bit in the channel status register (USART_CSR). Writing a 1 to the RSTSTA bit in the control register (USART_CR) clears the PARE bit. The timing for setting and clearing the parity bit is shown in the diagram below.

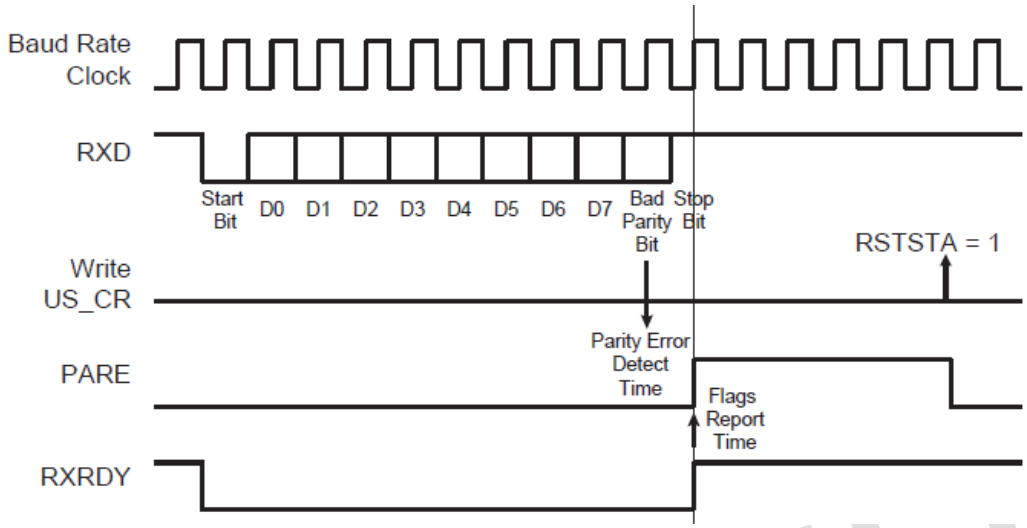


Figure 31-17: Parity Error Timing Diagram

31.4.3.10 Multidrop Mode

If the PAR field in the mode register (USART_MR) is programmed to 0x6 or 0x7, the USART will operate in multidrop mode. This mode distinguishes between data characters and address characters. When the parity bit is 0, data is sent; when the parity bit is 1, an address is sent.

When the USART is configured for multidrop mode, if the parity bit is high, the receiver will set the parity error bit PARE; when the SENDA bit in the control register is set to 1, the transmitter can also send characters when the parity bit is high.

To handle parity errors, writing a 1 to the RSTSTA bit in the control register to clear the PARE bit. When SENDA is written to the USART_CR, the transmitter sends an address byte (parity bit set). In this case, the next byte written to USART_THR will be sent as an address. If the SENDA command is not issued, any character written to USART_THR will be sent normally (with the parity bit set to 0).

31.4.3.11 Transmitter Time Protection

The time protection feature allows USART to connect with slow remote devices.

The time protection function allows an idle state to be inserted between two characters on the

transmitter TXD line. This idle state is actually a long stop bit.

The duration of idle state is programmed by the TG field in the transmitter time protection register (USART_TTGR). If the programmed value of this field is zero, no time protection will be generated. Otherwise, after each stop bit is sent, TXD will remain high for the number of bit periods specified in TG.

The behavior of the TXRDY and TXEMPTY status bits can be altered by the time protection, as shown in the diagram below. TXRDY will be set high only after the start bit of the next character is sent. Even if a character has been written to USART_THR, TXRDY will remain 0 during the time protection period. Since the time protection is part of the current transmission, TXEMPTY will remain low until the time protection phase is complete.

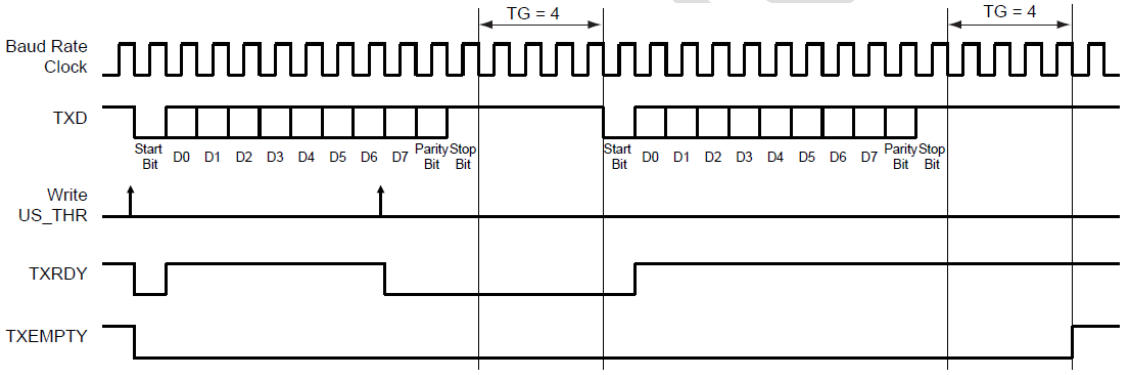


Figure 31-18: Timing Diagram of Time Protection Operation

The maximum time protection periods for the transmitter at different baud rates are shown in the following table.

Table 31-3: Maximum Time Protection Period

Baud Rate (bit/s)	Bit Time (μs)	Time Protection Length (ms)
1200	833	212.50
9600	104	26.56
14400	69.4	17.71
19200	52.1	13.28
28800	34.7	8.85
33400	29.9	7.63
56000	17.9	4.55

Baud Rate (bit/s)	Bit Time (μs)	Time Protection Length (ms)
57600	17.4	4.43
115200	8.7	2.21

31.4.3.12 Receiver Timeout

The receiver timeout supports the handling of variable-length frames. The receiver can detect the idle state on the RXD line; if a timeout is detected, the TIMEOUT bit in the channel status register (USART_CSR) will be set high, generating an interrupt to notify the driver that the frame has ended.

By programming the TO field in the receiver timeout register (USART_RTOR), the timeout delay period (the time the receiver waits for a new character) can be set. If the TO field is set to 0, the receiver timeout is disabled, and no timeout will be detected; and the TIMEOUT bit in the USART_CSR register will remain 0. Otherwise, the value of TO is loaded into a 16-bit counter. This counter decrements with each bit period and is reloaded upon the receipt of a new character. If the counter reaches 0, the TIMEOUT bit in the status register will be set high. The user can:

- Stop the counter clock until a new character is received. This can be achieved by writing 1 to the STTTO (start timeout) bit in the control register (USART_CR). In this way, the idle state on the RXD line will not trigger a timeout before the reception of a character, thus avoiding the need to handle an interrupt before receiving a character and allowing waiting for the next idle state on the RXD line after the frame has been received.
- Generate an interrupt if no character is received. This can be accomplished by writing 1 to the RETTO (reload and start timeout) bit in USART_CR. If RETTO is executed, the counter will start counting down from the TO value. The resulting periodic interrupts can be used to handle user-defined timeouts, such as when no key is pressed on the keyboard.

If STTTO is executed, the counter clock stops before the first character is received. The idle state of the RXD before the frame start does not provide a timeout. This prevents periodic interrupts and allows waiting for the frame to end when the RXD line is detected to be idle.

If RETTO is executed, the counter starts counting down from the TO value. The resulting periodic interrupts can be used to handle user-defined timeouts, such as when there is no input from the keyboard.

Table 31-4: Maximum Timeout Periods at Certain Standard Baud Rates

Baud Rate (bit/s)	Bit Time (μs)	Timeout Length (ms)
600	1667	109225
1200	833	54613
2400	417	27306
4800	208	13653
9600	104	6827
14400	69	4551
19200	52	3413
28800	35	2276
33400	30	1962
56000	18	1170
57600	17	1138
200000	5	328

31.4.3.13 Frame Error

The receiver can detect frame errors. A frame error is indicated when the stop bit of the received character is detected as 0. Frame errors may occur when the receiver and transmitter are out of synchronization.

Frame errors are represented by the FRAME bit in the USART_CSR register. When a frame error is detected, the FRAME bit is set in the middle of the stop bit time. The error can be cleared by writing 1 to the RSTSTA bit in the USART_CR register.

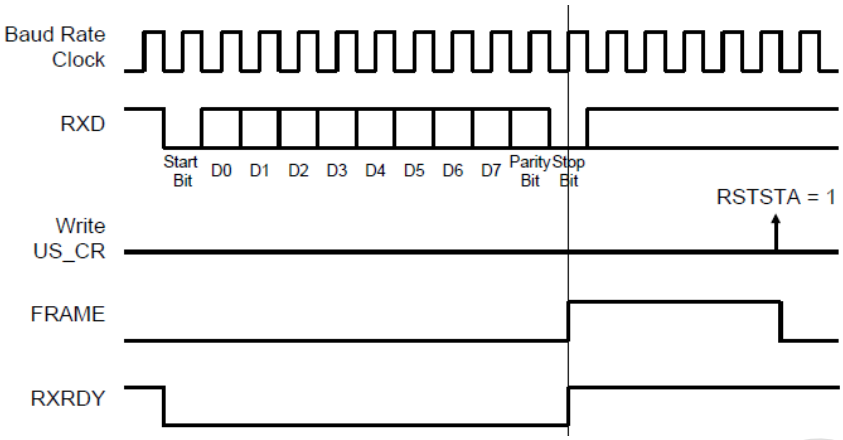


Figure 31-19: Timing Diagram of Frame Error Status

31.4.3.14 Transmission Break

The user can request the transmitter to generate a break condition on the TXD line. This causes the TXD line to be low for at least one complete character time, similar to transmitting a character of 0x00 with both the parity bit and stop bit as 0. Regardless, the transmitter guarantees that the TXD line will be low for one complete character transmission time until the user requests to remove the break condition.

Writing 1 to the STTBRK bit in the USART_CR register will send a break. This can be executed at any time, even if the transmitter is empty (no characters in the shift register or USART_THR) or a character is currently being transmitted. If a break request occurs while a character is being shifted out, the character transmission shall be completed before the TXD line goes low.

Once the STTBRK command is required, other STTBRK commands shall be ignored until the break is completed.

To remove the break condition, write 1 to the STPBRK bit in the USART_CR register. If STPBRK is requested before the end of the minimum break duration (one character, including start, data, parity, and stop bits), the transmitter guarantees that the break condition is completed.

The transmitter treats a break as a character; thus, the STTBRK and STPBRK commands will only be considered when the TXRDY in the USART_CSR register is 1. Similar to processing a

character, the TXRDY and TXEMPTY bits will be cleared when the break condition is activated.

Writing 1 to both the STTBRK and STPBRK bits in the USART_CR register will lead to unpredictable results. All STPBRK command requests that do not have preceding STTBRK commands will be ignored. Bytes written to the transmit holding register that have not been initiated will be ignored while a break is pending.

After the break condition, the transmitter will bring the TXD line back high (1) within a maximum of 12 bit times. Therefore, the transmitter ensures that the remote receiver correctly detects the end of the break and the start bit of the next character. If the time guarantee value exceeds 12, the TXD line will remain high during the time guarantee period.

After this period of holding the TXD line high, the transmitter resumes normal operation.

The following figure illustrates the effect of the start break (STTBRK) and stop break (STPBRK) commands on the TXD line.

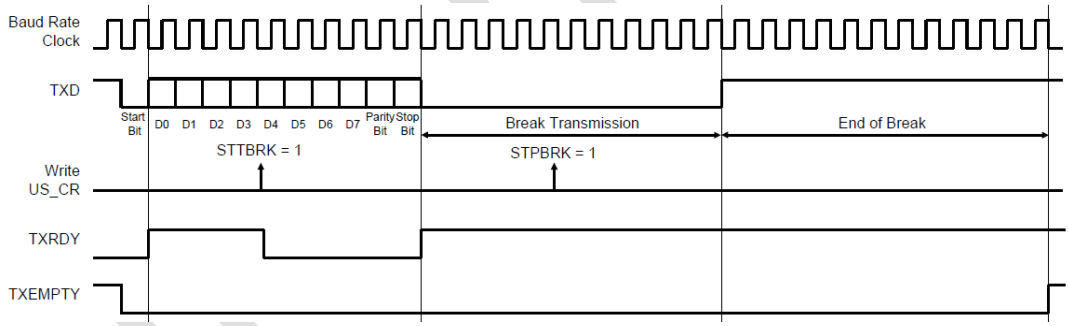


Figure 31-20: Transmission of Break

31.4.3.15 Reception Break

When all received data, parity and stop bits are low, the receiver detects a break condition. This is similar to detecting a frame with data of 0x00 and FRAME as low (indicating a frame error).

Upon detecting a low-level stop bit, the receiver sets the RXBRK bit in the USART_CSR register, which can be cleared by writing 1 to the RSTSTA bit in the USART_CR register.

A break end is detected when at least 2/16 of the bit period is high in asynchronous mode, or when a sample value is high in synchronous mode. The break end can also be achieved by setting the RXBRK bit.

31.4.4 IrDA Mode

The IrDA mode of USART supports half-duplex point-to-point wireless communication. It includes a modulator and demodulator for seamless connection with infrared transceivers, as shown in the figure below. The modulator and demodulator are compatible with the IrDA specification version 1.1, supporting data transmission speeds ranging from 2.4 kb/s to 115.2 kb/s.

By writing 0x8 to the USART_MODE field in the USART_MR register, the USART_IrDA mode can be enabled. The demodulation filter can be configured via the IrDA filter register (USART_IF). The USART transmitter and receiver operate in normal asynchronous mode, and all parameters are accessible. Note that both the modulator and demodulator are activated.

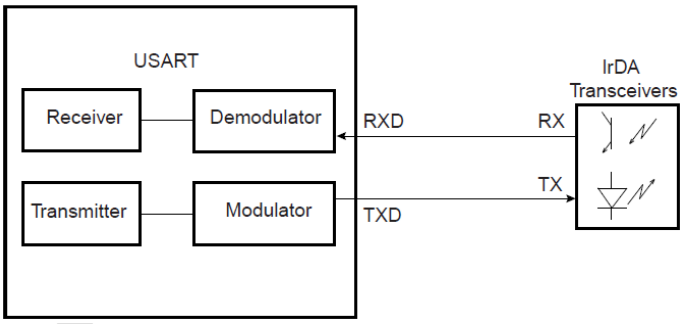


Figure 31-21: Connection with IrDA Transceiver

The receiver and transmitter must be enabled or disabled based on the transmission direction. To receive IrDA signals, the following configurations must be made:

- Disable TX and enable RX.
- Configure the GPIO port as TXD and set its output to 0 (to avoid LED illumination), disabling internal pull-up (to reduce power consumption).
- Receive data.

31.4.4.1 IrDA Modulation

When the baud rate is less than or equal to 115.2 kb/s, the RZI modulation scheme is used. A "0" is represented by a light pulse lasting for 3/16 of a bit period. The table below provides the durations of some signal pulses.

Table 31-5: IrDA Pulse Duration

Baud Rate	Pulse Duration (3/16)
2.4 kb/s	78.13 µs
9.6 kb/s	19.53 µs
19.2 kb/s	9.77 µs
38.4 kb/s	4.88 µs
57.6 kb/s	3.26 µs
115.2 kb/s	1.63 µs

The following figure shows an example of character transmission.

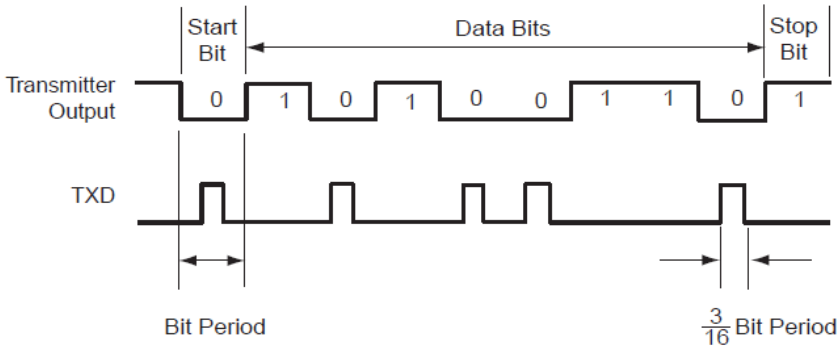


Figure 31-22: IrDA Modulation

31.4.4.2 IrDA Baud Rate

The following table gives some examples of CD values, baud rate errors, and pulse durations. Note that the maximum acceptable error is ±1.87%.

Table 31-6: IrDA Baud Rate Error

Peripheral Clock	Baud Rate	CD	Baud Rate Error	Pulse Time
3686400	115200	2	0.00%	1.63
20000000	115200	11	1.38%	1.63
32768000	115200	18	1.25%	1.63

Peripheral Clock	Baud Rate	CD	Baud Rate Error	Pulse Time
40000000	115200	22	1.38%	1.63
3686400	57600	4	0.00%	3.26
20000000	57600	22	1.38%	3.26
32768000	57600	36	1.25%	3.26
40000000	57600	43	0.93%	3.26
3686400	38400	6	0.00%	4.88
20000000	38400	33	1.38%	4.88
32768000	38400	53	0.63%	4.88
40000000	38400	65	0.16%	4.88
3686400	19200	12	0.00%	9.77
20000000	19200	65	0.16%	9.77
32768000	19200	107	0.31%	9.77
40000000	19200	130	0.16%	9.77
3686400	9600	24	0.00%	19.53
20000000	9600	130	0.16%	19.53
32768000	9600	213	0.16%	19.53
40000000	9600	260	0.16%	19.53
3686400	2400	96	0.00%	78.13
20000000	2400	521	0.03%	78.13
32768000	2400	853	0.04%	78.13

31.4.4.3 IrDA Demodulator

The demodulator is based on the IrDA receiver filter and includes an 8-bit down counter with values loaded from USART_IF. When a falling edge is detected on the RXD pin, the filter counter begins to count down at the rate of the master clock (MCK). If a rising edge is detected on the RXD pin, the counter stops and reloads the value from USART_IF. If the counter reaches 0 without detecting a rising edge, the input to the receiver is pulled low in one bit time.

The following figure illustrates the operation of the IrDA demodulator.

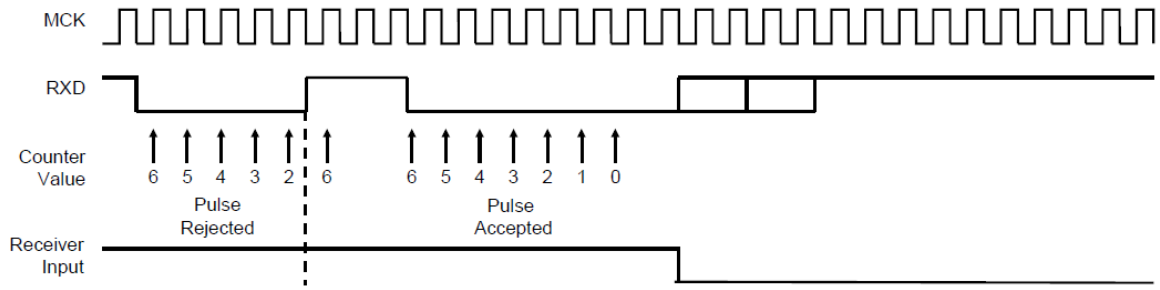


Figure 31-23: IrDA Demodulator Operation

Note that the field value of FI_DI_RATIO in USART_FIDI must be greater than 0 to ensure correct IrDA communication operation.

31.4.5 SPI Mode

The serial peripheral interface (SPI) mode is a synchronous serial data link that allows for communication with external devices in either master or slave mode.

It also enables inter-processor communication if an external processor is connected to the system.

The SPI is essentially a shift register that serially transmits data to other SPIs. During data transmission, one SPI system acts as the “master” controlling the data flow, while other SPIs operate as “slaves,” moving data in or out under the control of the master. Different CPUs can take turns acting as the master, and a single master can simultaneously transfer data to multiple slaves (multi-master protocols differ from single-master protocols, where only one CPU remains the master while others act as slaves). However, at any time, only one slave is allowed to write data to the master.

The SPI system selects a slave by the master sending the NSS signal. The USART in master mode of SPI can only connect to one slave because it can only generate one NSS signal.

The SPI system includes two data lines and two control lines:

- Master Out Slave In (MOSI): This data line is used for transmitting data from the master to the slave.
- Master In Slave Out (MISO): This data line is used for transmitting data from the slave to the master.
- Serial Clock (SCK): This control line is driven by the master to regulate data flow; the data baud rate can vary during transmission; one bit is transmitted per SCK cycle.
- Slave Select (NSS): This control line is used by the master to select or deselect the slave.

31.4.5.1 Operating Mode

The USART can operate in SPI master mode or SPI slave mode.

To configure the USART to operate in SPI master mode, write 0xE to the USART_MODE bit in the mode register. In this case, the SPI lines must be connected as follows:

- The output pin TXD drives the MOSI line.
- The MISO line drives the input pin RXD.
- The output pin SCK drives the SCK line.
- The output pin RTS drives the NSS line.

To configure the USART to operate in SPI slave mode, write 0xF to the USART_MODE bit in the mode register. In this case, the SPI lines must be connected as follows:

- The MOSI line drives the input pin RXD.
- The output pin TXD drives the MISO line.
- The SCK line drives the input pin SCK.
- The NSS line drives the input pin CTS. To avoid unpredictable behavior, once the SPI mode changes, the transmitter and receiver must be reset in software (in addition to the initialization after a hardware reset).

31.4.5.2 Baud Rate

In SPI mode, the baud rate generator operates similarly to that in USART synchronous mode. However, the following constraints must be adhered to:

- SPI master mode:
  - To generate the correct serial clock on the SCK pin, external clock SCK (USCLKS ≠ 0x3) must not be selected, and the CLKO bit in the mode register (USART_MR) must be set to 1.
  - The CD value must be greater than or equal to 4 for the transmitter and receiver to function properly.
  - If the internal clock division (MCK/DIV) is selected, the CD value must be set to an even number to ensure a 50:50 duty cycle on the SCK pin; if the internal clock (MCK) is chosen, the CD value can also be set to an odd number.
- SPI slave mode:
  - External clock (SCK) must be selected; the value of the USCLKS field in the mode register (USART_MR) is invalid; similarly, the value of USART_BRGR is also invalid since the clock is directly provided by the signal on the SCK pin of USART.
  - The frequency of the external clock (SCK) must not exceed 1/4 of the system clock frequency for the transmitter and receiver to operate properly.

31.4.5.3 Data Transmission

Up to 9 bits of data can be continuously transmitted on the TXD pin at the rising or falling edge (depending on CPOL and CPHA settings) of each programmable serial clock, without start bits, parity bits, or stop bits.

The number of data bits can be selected by setting the CHRL bit and the MODE9 bit in the mode register (USART_MR). If 9-bit data is chosen, only the MODE9 bit needs to be set; the CHRL field does not need to be considered. In SPI mode (whether in master or slave mode), the highest data bit is always transmitted first.

There are four combinations of clock polarity and phase for data transmission. The clock polarity is set by the CPOL bit in the mode register, and the clock phase is set by the CPHA bit. These two parameters determine at which clock edge to drive and sample the data. Each parameter has two states, resulting in four possible combinations. Therefore, a pair of master/slave must use the same parameter pair value to communicate. If multiple slaves are used, and each slave is fixed to a different configuration, the master must be reconfigured when communicating with different slaves.

Table 31-7: SPI Bus Protocol Modes

SPI Bus Protocol Mode	CPOL	CPHA
0	0	1
1	0	0
2	1	1
3	1	0

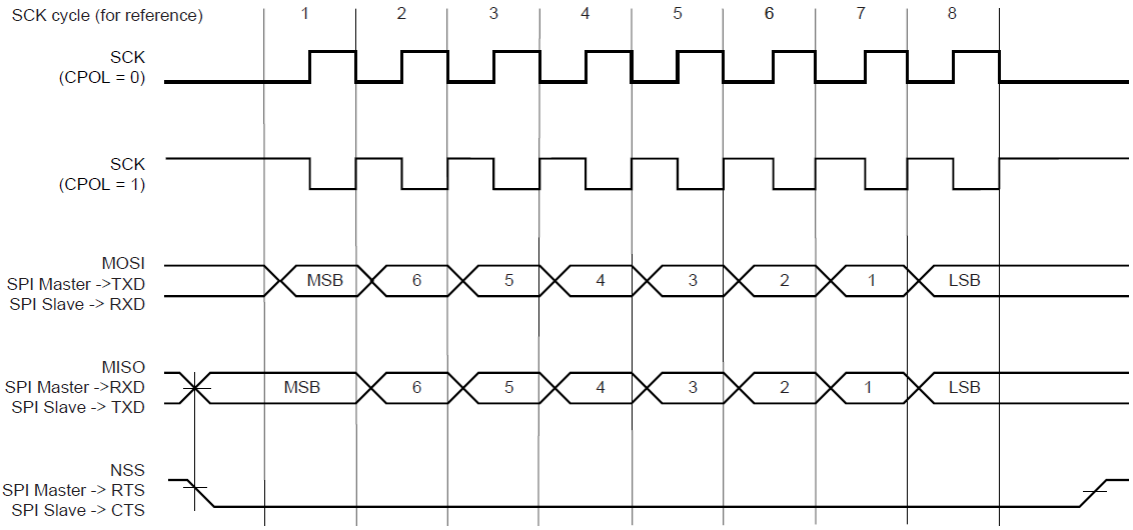


Figure 31-24: SPI Transmission Format (CPHA = 1, 8 bits per transmission)

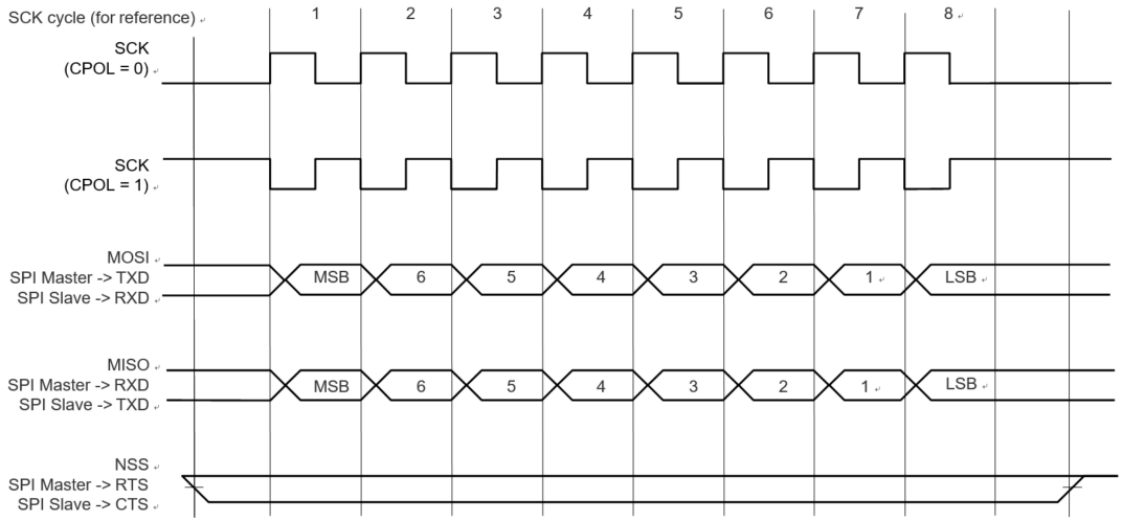


Figure 31-25: SPI Transmission Format (CPHA = 0, 8 bits per transmission)

31.4.5.4 Receiver and Transmitter Control

31.4.5.4.1 Character Transmission

Character transmission is performed by writing characters to the transmit holding register (USART_THR). If the USART operates in SPI master mode, additional conditions for transmitting characters can be imposed. When the receiver is not ready (i.e., the character has not been read), setting the INACK bit in the USART mode register (USART_MR) can prevent any character from being transmitted (even if the data has been written to USART_THR). If INACK is set to 0, characters will be sent regardless of the state of the receiver; if INACK is set to 1, the transmitter must wait for the data in the receive holding register to be read before transmitting data (indicated by the RXRDY flag being cleared), thus avoiding any overflow in the receiver (character loss).

The transmitter has two status bits in the channel status register (USART_CSR): TXRDY (transmit ready) indicates that USART_THR is empty, and TXENPTY indicates that all characters written to USART_THR have been processed. After the current character is processed, the last character written to USART_THR is sent to the shift register of the transmitter, and USART_THR is cleared, then TXRDY is set. When the transmitter is disabled,

both TXRDY and TXENPTY bits are set to 0. When TXRDY is 0, writing characters to USART_THR is ineffective, and the written characters will be lost.

If the USART operates in SPI slave mode and it is necessary to send a character when the transmit holding register (USART_THR) is empty, the UNRE (buffer data empty error) bit will be set. During this time, the TXD transmission line remains high. The UNRE bit can be cleared by writing 1 to the RSTSTA (reset status) bit in the control register (USART_CR).

In SPI master mode, before transmitting the most significant bit, a low-level signal is issued on the slave select line (NSS) for one bit time; after transmitting the least significant bit, NSS remains high for one bit time. Therefore, the slave select signal is always released between character transmissions, inserting at least a 3-bit time delay. However, to support the CSAAT mode (chip selection activated after transmission) for the slave device, the slave select line (NSS) can be forced low by setting the RTSEN bit in the control register (USART_CR) to 1. Only by setting the RTSDIS bit in the control register (USART_CR) to 1 can the slave select line (NSS) be released by pulling it high (for example, when all data has been transmitted to the slave device).

In SPI slave mode, the transmitter does not request to initialize character transmission on the falling edge of the slave select line (NSS), but only during the low level. However, the slave select line (NSS) must remain low for at least one bit time before the first serial clock cycle corresponding to the most significant bit.

31.4.5.4.2 Character Reception

When a character is fully received, it is transferred to the receive holding register (USART_RHR), and the RXRDY bit in the status register (USART_CSR) is set high. If a character is received while RXRDY is set, the OVER (overflow error) bit is set. The last character is transferred to USART_RHR, overwriting the current character. Writing 1 to the RSTSTA (reset status) bit in the control register (USART_CR) can clear the OVRE bit.

To ensure the normal operation of the receiver in SPI slave mode, the master device must ensure that there is at least a one-bit time delay between transmitting each character in the frame. The receiver does not request to initialize character reception on the falling edge of the slave select line (NSS), but only during the low level. However, the slave select line (NSS) must remain low for at least one bit time before the first serial clock cycle corresponding to the most significant bit.

31.4.5.4.3 Reception Timeout

Since the baud rate clock of the receiver is only available during data transmission in SPI mode, a timeout is not possible in this mode, regardless of the timeout value in the timeout register (USART_RTOR) (TO bit field value).

31.4.6 LIN Mode

The LIN mode provides connections for master nodes and slave nodes on the LIN bus.

LIN (local interconnect network) is a serial communication protocol that effectively supports the control of electromechanical nodes in distributed automotive applications.

The main features of the LIN bus include:

- Single master/multiple slaves concept
- Low-cost implementation based on general UART/SCI interface hardware, equivalent software, or pure state machines
- Slave nodes can achieve self-synchronization without a crystal oscillator or ceramic oscillator.
- Deterministic signal transmission
- Low-cost single-wire implementation
- Transmission rates of up to 20 kbps

LIN provides cost-effective bus communication without the need for CAN bandwidth and multifunctionality.

The LIN mode allows microprocessors to handle LIN frames with minimal actions.

31.4.6.1 Operation Mode

The USART can act as a LIN master node or a LIN slave node.

The node configuration is selected by setting the USART_MODE bit in the USART mode register (USART_MR):

- LIN master node (USART_MODE = 0xA)
- LIN slave node (USART_MODE = 0xB)

To avoid unpredictable behavior, any changes to the LIN node configuration must be made after performing a software reset on the transmitter and receiver (excluding the initialization of node configuration following a hardware reset).

31.4.6.2 Baud Rate Configuration

Refer to the section "31.4.1.1 Asynchronous Mode".

The baud rate is configured in the baud rate generator register (USART_BRGR).

- LIN master node: The baud rate is configured in the baud rate generator register (USART_BRGR).
- LIN slave node: The initial baud rate is configured in the baud rate generator register (USART_BRGR). When writing to USART_BRGR, this configuration is automatically copied to the LIN baud rate register (USART_LINBRR). After the synchronization process is complete, the baud rate is updated in USART_LINBRR.

31.4.6.3 Reception and Transmission Control

Refer to the section “31.4.2 Receiver and Transmitter Control”.

31.4.6.4 Character Transmission

Refer to the section “31.4.3.1 Transmitter Operation”.

31.4.6.5 Character Reception

Refer to the section “31.4.3.8 Receiver Operation”.

31.4.6.6 Message Header Transmission (Master Node Configuration)

All LIN frames start with a message header sent by the master node, which consists of the break field, sync field and identifier field.

Therefore, in the master node configuration, frame processing begins with sending the message header.

Once the identifier is written to the LIN identifier register (USARTINIR), the message header is transmitted. At the same time, the TXRDY flag goes low.

The break field, sync field and identifier field are automatically sent in sequence.

The break field consists of 13 dominant bits and 1 recessive bit; the sync field is the character 0x55, while the identifier corresponds to the character written to the LIN identifier register (USART_LINIR). The identifier parity bit is automatically calculated and sent.

When the identifier field is transmitted to the shift register of the transmitter, the TXRDY flag goes high.

Once the break field is sent, the LINBK flag in the channel status register (USART_CSR) is set. Similarly, once the identifier field is sent, the LINID flag in USART_CSR is set. These flags can be reset by writing 1 to the RSTSTA bit in the control register (USART_CR).

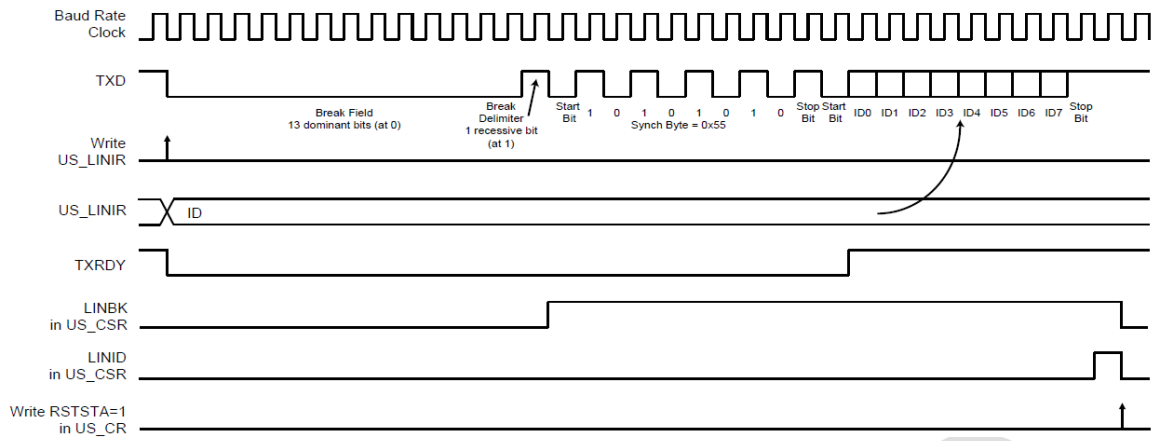


Figure 31-26: Message Transmission

31.4.6.7 Message Header Reception (Slave Node Configuration)

All LIN frames start with a message header sent by the master node, which consists of the break field, sync field and identifier field.

In the slave node configuration, frame processing begins with receiving the message header.

The USART uses an 11-bit break detection threshold at the actual baud rate. At any time, if 11 consecutive recessive bits are detected on the bus, the USART will detect a break field. As long as the break field is not detected, the USART remains idle and does not receive data.

When a break field is detected, the LINBK flag in the channel status register (USART_CSR) is set, and the USART expects the sync field character to be 0x55. This field is used to update the actual baud rate for synchronization. If the received sync character is not 0x55, an inconsistent sync field error is generated.

After receiving the sync field, the USART expects to receive the identifier field.

When the identifier field is received, the LINID flag in USART_CSR is set. At this point, the IDCHR field in the LIN identifier register (USART_LINIR) is updated with the received character. The identifier parity bit can be automatically calculated and checked.

If the entire message header is not received within the maximum length time specified by $t_{Header\_maximum}$, the LINHTE error flag in USART_CSR will be set.

Writing 1 to the RSTSTA bit in the control register (USART_CR) resets the LINID, LINBK and LINHTE flags.

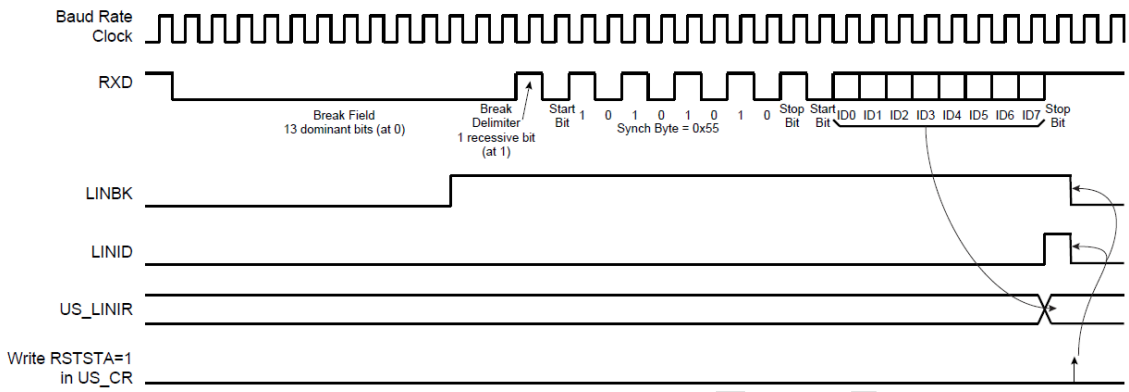


Figure 31-27: Message Reception

31.4.6.8 Slave Node Synchronization

Synchronization is completed solely in the slave node configuration. This step is based on the timing measurement between the falling edges of the sync field. The distances of the falling edges are 2, 4, 6 and 8 bit times.

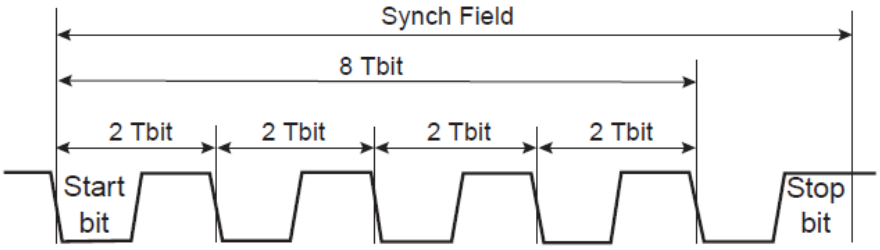


Figure 31-28: Sync Field

The timing measurement is performed by a 19-bit counter based on the sampling clock.

When the start bit of the sync field is detected, the counter is reset. The counter then increments during the next 8 Tbits period of the sync field. At the end of this 8 Tbits, the counter stops. At this point, the high 16 bits of the counter (the value divided by 8) are given

to the new clock divider (LINCD), and the low 3 bits (the remainder) are provided to the new fractional part (LINFP).

Upon receiving the sync field, the clock divider (CD) and the fractional part (FP) in the baud rate generator register (USART_BRGR) will be updated.

If the sampled synchronization character is not equal to 0x55, the error flag LINISFE in the channel status register (USART_CSR) will be set. Writing 1 to the RSTSTA bit in the control register (USART_CR) to perform a reset.

Once the sync field is fully received, if the SYNCDIS bit in the LIN mode register (USART_LINMR) is not disabled, the clock divider (LINCD) and the fractional part (LINFP) in the LIN baud rate register (USART_LINBRR) will be updated.

After receiving the sync field:

- If the calculated baud rate deviation compared to the initial baud rate is greater than the maximum tolerance FTol_Unsynch (±15%), the clock divider (LINCD) and fractional part (LINFP) will not be updated, and the error flag LINSTE in the channel status register (USART_CSR) will be set.
- If the sampled synchronization character is not equal to 0x55, the clock divider (LINCD) and fractional part (LINFP) will not be updated, and the error flag LINISFE in the channel status register (USART_CSR) will be set.

To reset the LINSTE and LINISFE flags, write 1 to the RSTSTA bit in the control register (USART_CR).

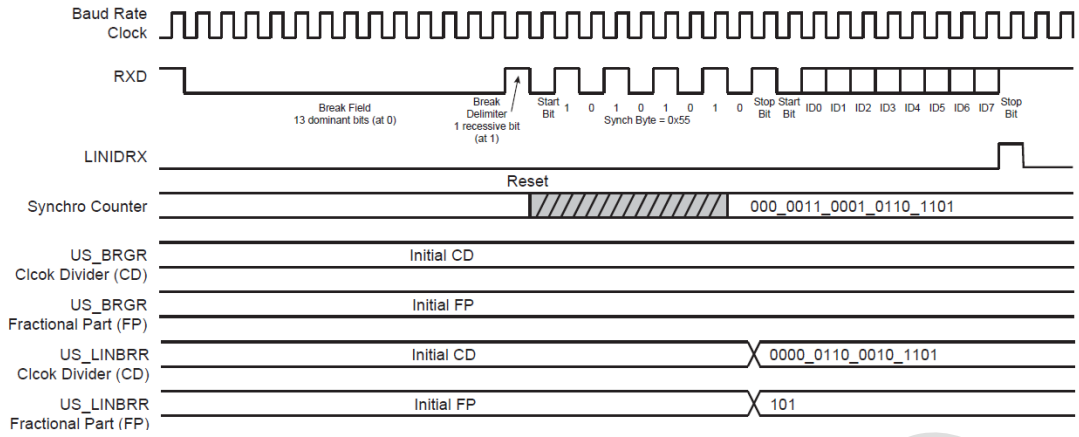


Figure 31-29: Timing Diagram of Slave Node Synchronization

The accuracy of synchronization depends on several parameters:

- Nominal clock frequency (f_{Nom}) (theoretical slave node clock frequency)
- Baud rate
- Oversampling (Over = 0 => 16X or Over = 1 => 8X)

The following formulas are used to calculate the deviation of the slave bit rate relative to the master bit rate (f_{SLAVE} is the actual slave node clock frequency):

$$Baudrate\_deviation = \left(100 \times \frac{[\alpha \times 8 \times (2 - Over) + \beta] \times Baudrate}{8 \times f_{SLAVE}}\right)\%$$

$$Baudrate\_deviation = \left(100 \times \frac{[\alpha \times 8 \times (2 - Over) + \beta] \times Baudrate}{8 \times \left(\frac{f_{TOL\_UNSYNCH}}{100}\right) \times f_{Nom}}\right)\%$$

$f_{TOL\_UNSYNCH}$ is the deviation of the actual slave node clock from the nominal clock frequency. The LIN standard specifies that this must not exceed ±15%. It also stipulates that the bit rate difference between two nodes in communication must not exceed ±2%, indicating that the baud rate deviation must not exceed ±1%. This leads to the minimum value for the nominal clock frequency:

$$f_{Nom}(min) = \left(100 \times \frac{[0{,}5 \times 8 \times (2 - Over) + 1] \times Baudrate}{8 \times \left(\frac{-15}{100} + 1\right) \times 1\%}\right)Hz$$

Examples:

- Baud rate = 20 kb/s, OVER = 0 (Oversampling 16X) => $f_{Nom(min)}$ = 2.64 MHz
- Baud rate = 20 kb/s, OVER = 1 (Oversampling 8X) => $f_{Nom(min)}$ = 1.47 MHz
- Baud rate = 1 kb/s, OVER = 0 (Oversampling 16X) => $f_{Nom(min)}$ = 132 kHz
- Baud rate = 1 kb/s, OVER = 1 (Oversampling 8X) => $f_{Nom(min)}$ = 74 kHz

31.4.6.9 Identifier Parity Check

The protected identifier consists of two subfields: the identifier and the identifier parity. Bits 5:0 are allocated to the identifier, and bits 7:6 are allocated to the parity.

The USART interface can generate and check these parity bits but can also disable these functions. The user can choose between the two modes using the PARDIS bit in the LIN mode register (USART_LINMR):

- PARDIS = 0:
  - During message header transmission, the parity bit is calculated and sent along with the lower 6 bits of the IDCHR field of the LIN identifier register (USART_LINIR), with bits 6 and 7 discarded.
  - During message header reception, the parity bit of the identifier is checked. If the parity bit is incorrect, an identifier parity error occurs. Only the lower 6 bits of the IDCHR field are updated with the received identifier, while bits 7:6 are fixed at 0.
- PARDIS = 1:
  - During message header transmission, all bits of the IDCHR field in the LIN identifier register (USART_LINIR) are sent on the bus.
  - During message header reception, all bits of the IDCHR field are updated with the received identifier.

31.4.6.10 Node Operation

The relevance of the node in the identifier function is related to LIN responses. Therefore, the USART must be configured after sending or receiving an identifier. There are three configurations:

- PUBLISH: The node sends a response.
- SUBSCRIBE: The node receives a response.
- IGNORE: The node is irrelevant to the response; it neither sends nor receives a response.

This is configured by the node operation (NACT) field in the USART_LINMR register. For example, a LIN group contains one master and two slaves:

- Data transmitted from the master to slaves 1 and 2:
  - NACT (master) = PUBLISH
  - NACT (slave 1) = SUBSCRIBE
  - NACT (slave 2) = SUBSCRIBE
- Data transmitted from the master to slave 1:
  - NACT (master) = PUBLISH
  - NACT (slave 1) = SUBSCRIBE
  - NACT (slave 2) = IGNORE
- Data transmitted from slave 1 to the master:
  - NACT (master) = SUBSCRIBE
  - NACT (slave 1) = PUBLISH
  - NACT (slave 2) = IGNORE
- Data transmitted from slave 1 to slave 2:
  - NACT (master) = IGNORE

- NACT (slave 1) = PUBLISH
- NACT (slave 2) = SUBSCRIBE

- Data transmitted from slave 2 to the master and slave 1:
  - NACT (master) = SUBSCRIBE
  - NACT (slave 1) = SUBSCRIBE
  - NACT (slave 2) = PUBLISH

31.4.6.11 Response Data Length

The LIN response data length is the number of data field bytes in the response, excluding the checksum.

The response data length can be configured by the user or automatically defined by bits 5:4 of the identifier (compatible with LIN specification 1.1). Users can select between these two modes using the DLM bit in the LIN mode register (USART_LINMR):

- DLM = 0: The response data length is configured by the user through the DLC field of USART_LINMR. The response data length equals (DLC + 1) bytes. The DLC setting range is from 0 to 255, so the response can contain from 1 data byte to 256 data bytes.
- DLM = 1: The response data length is defined by the identifier (IDCHR in USART_LINIR) according to the following table. The DLC field of USART_LINMR is discarded. The response can contain 2, 4 or 8 data bytes.

Table 31-8: Response Data Length with DLM = 1

IDCHR[5]	IDCHR[4]	Response Data Length (Bytes)
0	0	2
0	1	2
1	0	4
1	1	8

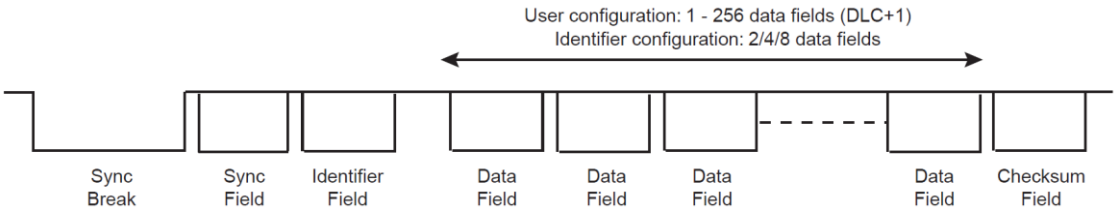


Figure 31-30: Response Data Length Diagram

31.4.6.12 Checksum

The last field of the frame is the checksum. The checksum consists of the carry-inclusive inverted 8-bit sum of either all data bytes or all data bytes along with the protected identifier. The checksum calculated only for the data bytes is referred to as a classic checksum, used for communication with LIN 1.3 slaves. The checksum that includes both data bytes and the protected identifier byte is referred to as an enhanced checksum, used for communication with LIN 2.0 slaves.

The USART can be configured to:

- Automatically send and check the enhanced checksum (CHKDIS = 0 & CHKTYP = 0)
- Automatically send and check the classic checksum (CHKDIS = 0 & CHKTYP = 0)
- Not send and check the checksum (CHKDIS = 1)

This configuration is controlled by the checksum type (CHKTYP) and checksum disable (CHKDIS) fields in the LIN mode register (USART_LINMR).

If the checksum feature is disabled, users can manually send it by treating the checksum as a normal data byte and adding 1 to the response data length.

31.4.6.13 Frame Slot Mode

This mode is only useful for the master node. It adheres to the following rules: each frame slot length must be greater than or equal to $t_{Frame\_Maximum}$.

If the frame slot mode is enabled (FSDIS = 0) and the frame transmission is completed, the TXRDY flag will only be reset after $t_{Frame\_Maximum}$. Therefore, if the duration of the previous frame slot is less than $t_{Frame\_Maximum}$, the master node cannot send a new header.

If the frame slot mode is disabled (FSDIS = 1), the TXRDY flag will be immediately reset upon completion of frame transmission.

The $t_{Frame\_Maximum}$ is calculated as follows:

- If the checksum is sent (CHKDIS = 0):
  - $t_{Header\_Nominal} = 34 \times Tbit$
  - $t_{Response\_Nominal} = 10 \times (NData + 1) \times t_{bit}$
  - $t_{Frame\_Maximum} = 1.4 \times (t_{Header\_Nominal} + t_{Response\_Nominal} + 1)$*
  - $t_{Frame\_Maximum} = 1.4 \times (34 + 10 \times (DLC + 1 + 1) + 1) \times t_{bit}$
  - $t_{Frame\_Maximum} = (77 + 14 \times DLC) \times t_{bit}$
- If the checksum is not sent (CHKDIS = 1):
  - $t_{Header\_Nominal} = 34 \times t_{bit}$
  - $t_{Response\_Nominal} = 10 \times NData \times t_{bit}$
  - $t_{Frame\_Maximum} = 1.4 \times (t_{Header\_Nominal} + t_{Response\_Nominal} + 1)$*
  - $t_{Frame\_Maximum} = 1.4 \times (34 + 10 \times (DLC + 1) + 1) \times t_{bit}$
  - $t_{Frame\_Maximum} = (63 + 14 \times DLC) \times t_{bit}$

Note*: The “+1” ensures that the result of $t_{Frame\_Maximum}$ is an integer (LIN specification 1.3).

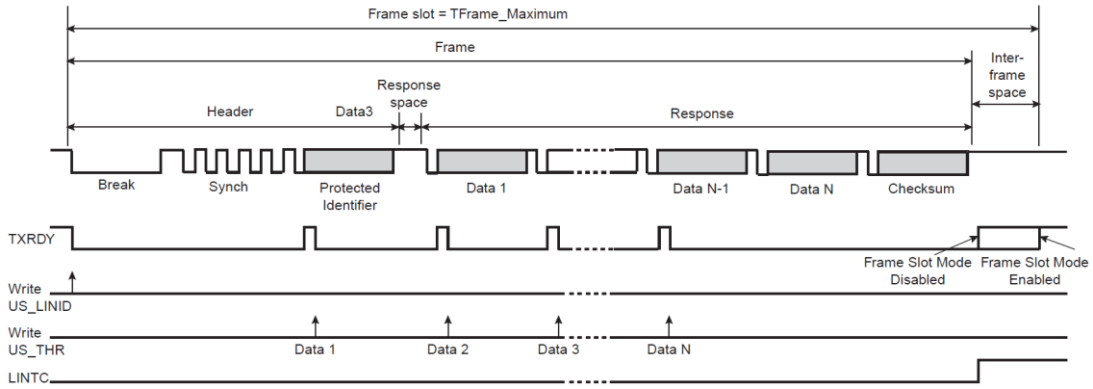


Figure 31-31: USART Frame Slot Mode

31.4.7 LIN Error

31.4.7.1 Bit Error

When the USART is transmitting, if the value transmitted on the TX line differs from the value sampled on the RX line, a bit error will be generated in the slave node configuration. If a bit error is detected, the transmission is aborted at the next byte boundary.

This error is indicated by the LINBE flag in the channel status register (USART_CSR).

31.4.7.2 Sync Field Inconsistency Error

If the received sync field character is not 0x55, this error will be generated in the slave node configuration.

This error is indicated by the LINISFE flag in the channel status register (USART_CSR).

31.4.7.3 Identifier Parity Error

If there is a parity error in the identifier, this error will be generated in the slave node configuration. This error can only be generated if the parity function is enabled (PARDIS = 0).

This error is indicated by the LINIPE flag in the channel status register (USART_CSR).

31.4.7.4 Checksum Error

If the received checksum is incorrect, this error will be generated in the master configuration of the slave node. This flag will only be set if the checksum function is enabled (CHKDIS = 0).

This error is indicated by the LINCE flag in the channel status register (USART_CSR).

31.4.7.5 Slave Not Responding Error

When the USART expects a response from other nodes (NACT = SUBSCRIBE), but no valid message appears on the bus within the maximum length $t_{Frame\_Maximum}$ of the message frame, this error will be generated in the master configuration of the slave node. This error is disabled if the USART does not expect any messages (NACT = PUBLISH or NACT = IGNORE).

This error is indicated by the LINSNRE flag in the channel status register (USART_CSR).

31.4.7.6 Synchronization Tolerance Error

If, after the clock synchronization procedure, the calculated baud rate deviation is greater than the maximum tolerance FTol_Unsynch (±15%) compared to the initial baud rate, this error will be generated in the slave node configuration.

This error is indicated by the LINSTE flag in the channel status register (USART_CSR).

31.4.7.7 Message Header Timeout Error

If the message header is not fully received within the maximum length $t_{Header\_Maximum}$, this error will be generated in the slave node configuration.

This error is indicated by the LINHTE flag in the channel status register (USART_CSR).

31.4.8 LIN Frame Processing

31.4.8.1 Master Node Configuration

- Write TXEN and RXEN of USART_CR to enable transmission and reception.
- Write USART_MODE of USART_MR to select LIN mode and master node.
- Write CD and FP of USART_BRGR to configure the baud rate.
- Configure the frame transmission by writing NACT, PARDIS, CHKDIS, CHKTYPE, DLCM, FSDIS and DLC of USART_LINMR.
- Check if TXRDY in USART_CSR is set.
- Write IDCHR of USART_LINIR to transmit the message header.

The subsequent operations depend on the NACT configuration:

- Case 1: NACT = PUBLISH, USART sends a response.
  1. Wait for TXRDY in USART_CSR to go high.
  2. Write TCHR of USART_THR to send one byte.
  3. If not all data has been written, repeat the above two steps.
  4. Wait for LINTC of USART_CSR to go high.
  5. Check for LIN errors.
- Case 2: NACT = SUBSCRIBE, USART receives a response.
  1. Wait for RXRDY of USART_CSR to go high.
  2. Read RCHR of USART_RHR.
  3. If not all data has been read, repeat the above two steps.
  4. Wait for LINTC in USART_CSR to go high.
  5. Check for LIN errors.
- Case 3: NACT = IGNORE, USART does not pay attention to the response.
  1. Wait for LINTC of USART_CSR to go high.
  2. Check for LIN errors.

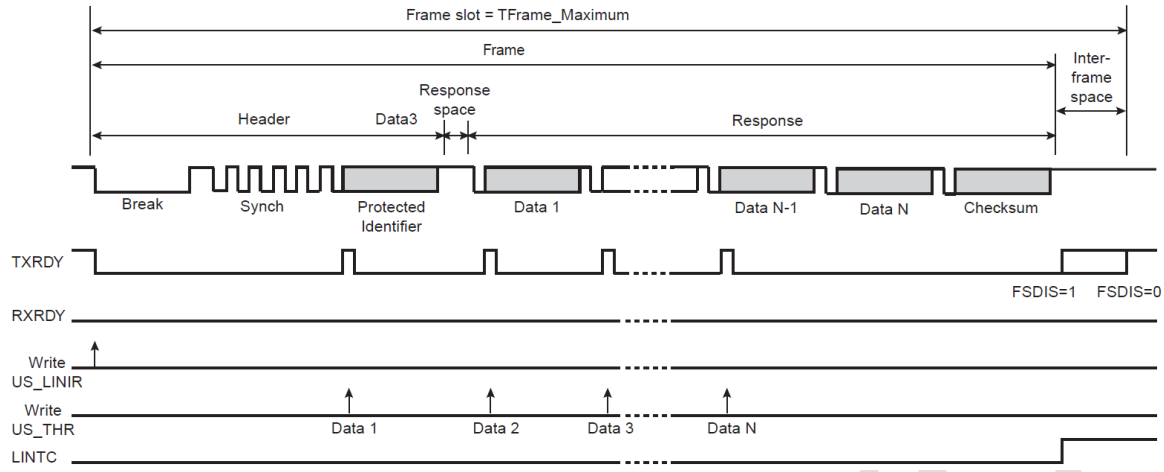


Figure 31-32: Master Node Configuration, NACT = PUBLISH

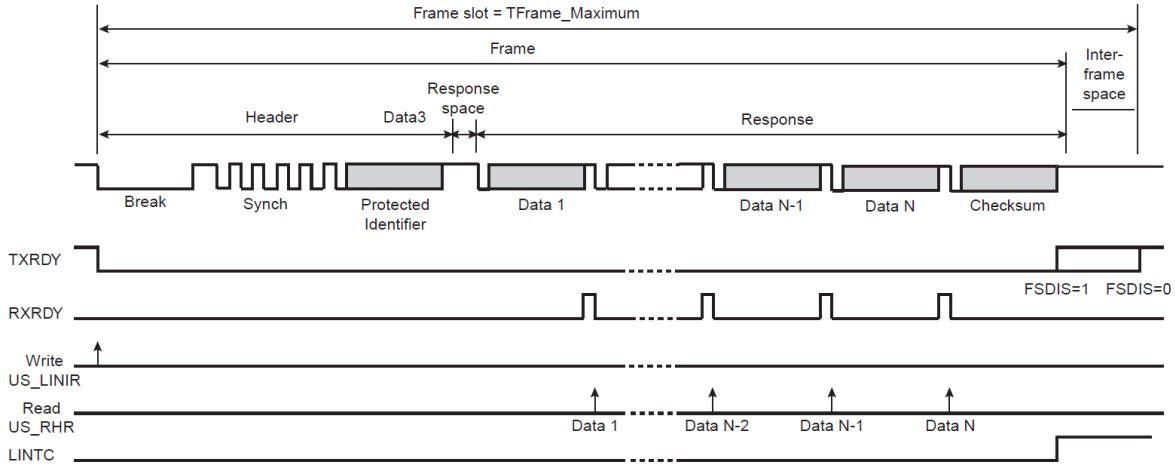


Figure 31-33: Master Node Configuration, NACT = SUBSCRIBE

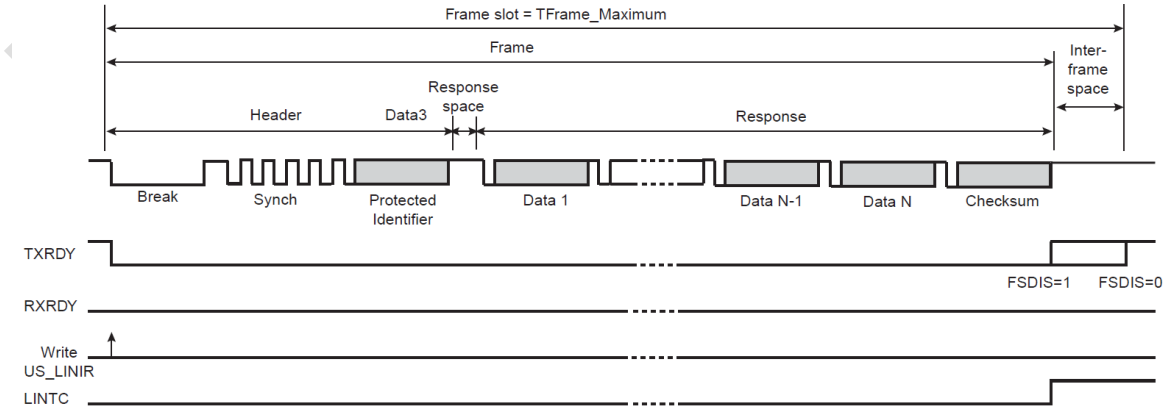


Figure 31-34: Master Node Configuration, NACT = IGNORE

31.4.8.2 Slave Node Configuration

- Write TXEN and RXEN of USART_CR to enable transmission and reception.
- Write USART_MODE of USART_MR to select LIN mode and slave node.
- Write CD and FP of USART_BRGR to configure the baud rate.
- Wait for LINID of USART_CSR to go high.
- Check for LINISFE and LINPE errors.
- Read IDCHR of USART_RHR.
- Configure the frame transmission by writing NACT, PARDIS, CHKDIS, CHKTYPE, DLCM and DLC of USART_LINMR.

Note: If NACT for this frame is configured as PUBLISH, the USART_LINMR register must be written with NACT = PUBLISH even if this field is properly configured, so as to set the TXREADY flag and the corresponding write transmission request.

The subsequent operations depend on the NACT configuration:

- Case 1: NACT = PUBLISH, USART sends a response.
  1. Wait for TXRDY of USART_CSR to go high.
  2. Write TCHR of USART_THR to send one byte.
  3. If not all data has been written, repeat the above two steps.
  4. Wait for LINTC of USART_CSR to go high.
  5. Check for LIN errors.
- Case 2: NACT = SUBSCRIBE, USART receives a response.
  1. Wait for RXRDY of USART_CSR to go high.
  2. Read RCHR of USART_RHR.

3. If not all data has been read, repeat the above two steps.
4. Wait for LINTC of USART_CSR to go high.
5. Check for LIN errors.

- Case 3: NACT = IGNORE, USART does not pay attention to the response.

1. Wait for LINTC of USART_CSR to go high.
2. Check for LIN errors.

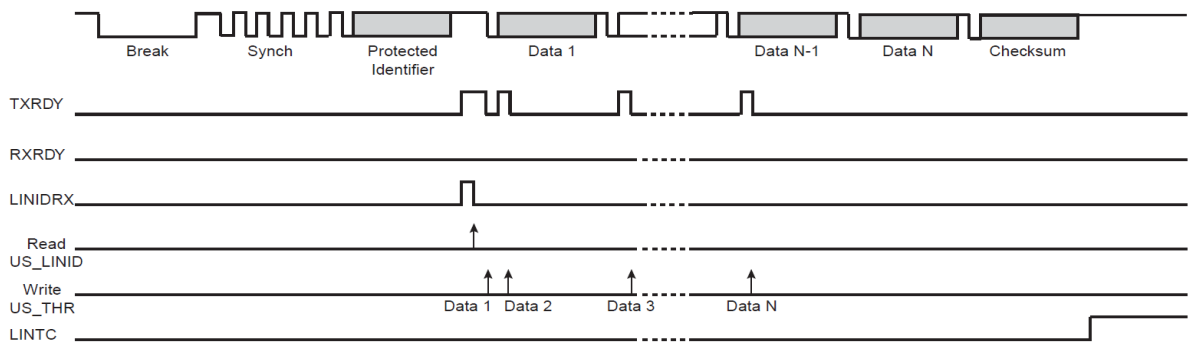


Figure 31-35: Slave Node Configuration, NACT = PUBLISH

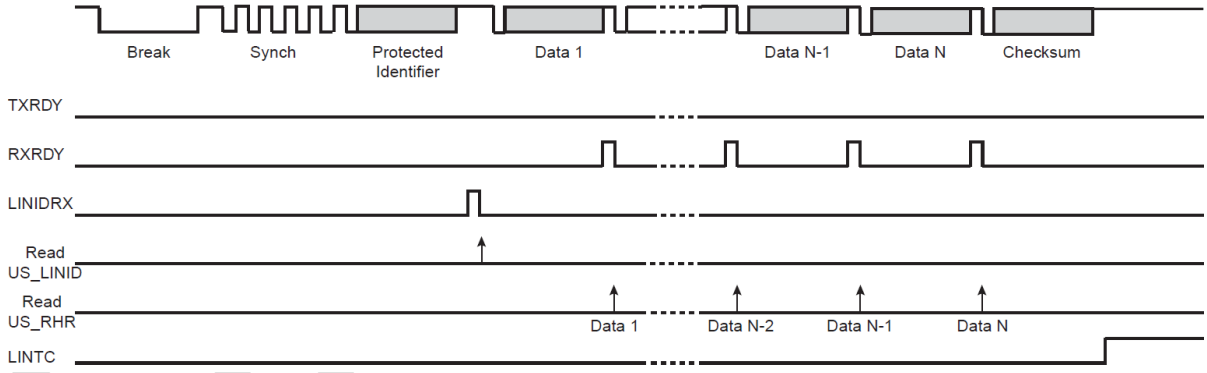


Figure 31-36: Slave Node Configuration, NACT = SUBSCRIBE

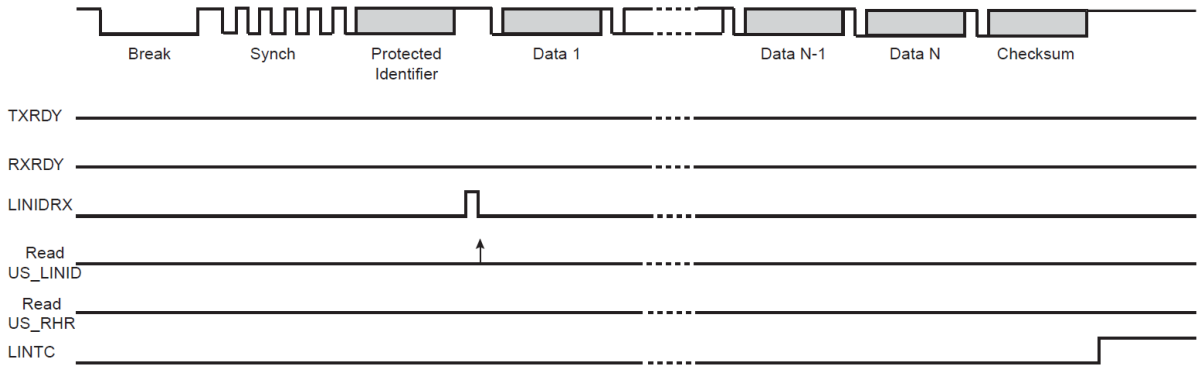


Figure 31-37: Slave Node Configuration, NACT = IGNORE

31.4.9 Using DMA to Process LIN Frames

USART can be used in conjunction with DMA to transfer data directly to and from on-chip and off-chip memory without any processor intervention.

31.4.9.1 Master Node Configuration

Users can select between two DMA modes via the LIN mode register (USART_LINMR) using the PDCM bit:

- PDCM = 1: The LIN configuration is stored in the write buffer, and is written by DMA to the transmit holding register (USART_THR) rather than the LIN mode register (USART_LINMR). Since the DMA transfer size is limited to one byte, the transfer is divided into two accesses. During the first access, the bits NACT, PARDIS, CHKDIS, CHKTYP, DLM and FSDIS are written. During the second access, the 8-bit DLC field is written.
- PDCM = 0: The LIN configuration is not stored in the write buffer and must be written by the user to the LIN mode register (USART_LINMR).

If the USART sends a response (NACT = PUBLISH), the write buffer also contains the identifier and data.

If the USART receives a response (NACT = SUBSCRIBE), the read buffer contains the data.

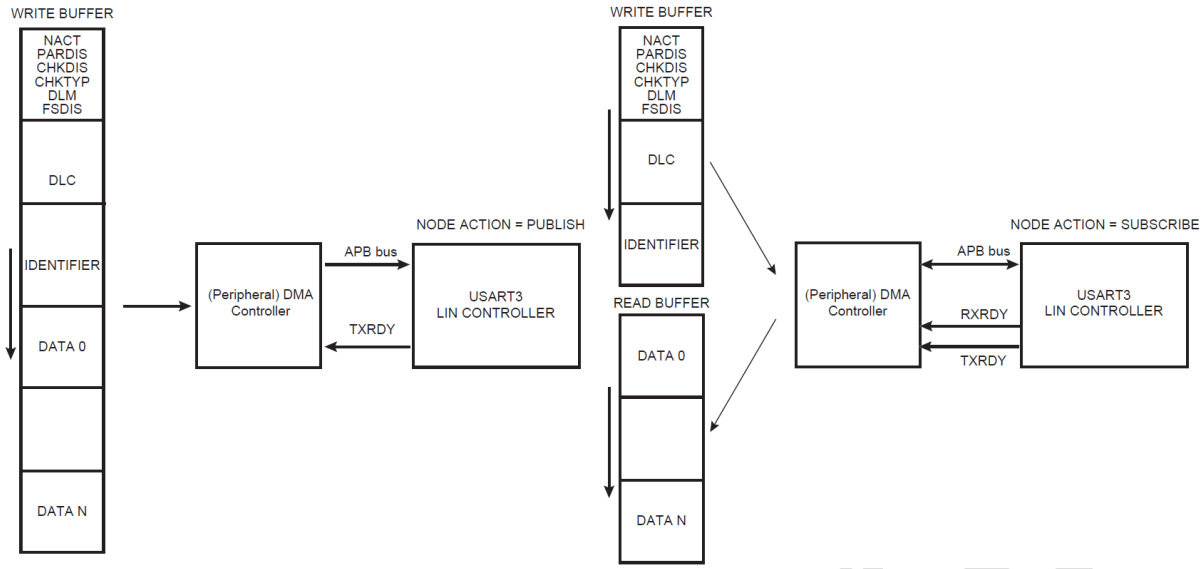


Figure 31-38: Master Node with DMA (PDCM = 1)

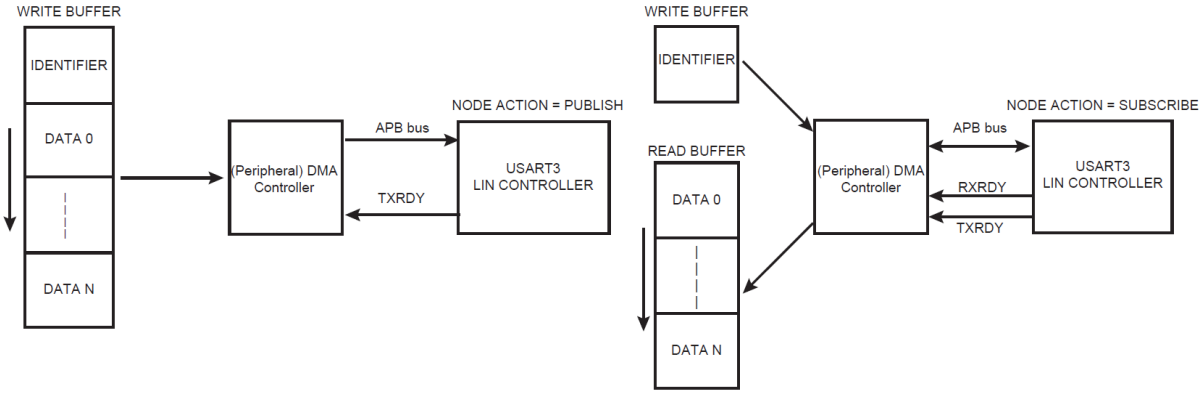


Figure 31-39: Master Node with DMA (PDCM = 0)

31.4.9.2 Slave Node Configuration

In this configuration, DMA only transfers data. The identifier must be read by the user from the LIN identifier register (USART_LINIR). The LIN mode must be written by the user to the LIN mode register (USART_LINMR).

If the USART sends a response (NACT = PUBLISH), the write buffer contains the data.

If the USART receives a response (NACT = SUBSCRIBE), the read buffer contains the data.

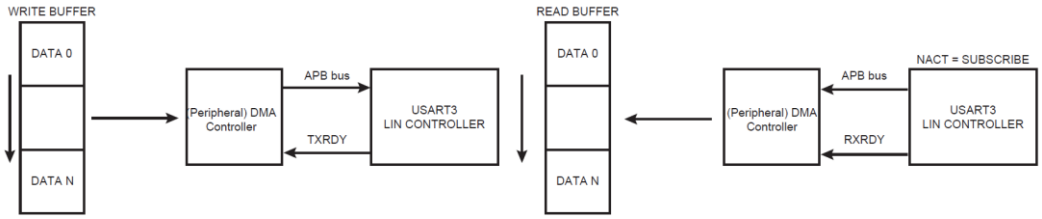


Figure 31-40: USART Slave Node Configuration

31.4.9.3 Wake-up Request

Any node in the LIN group can request to wake up from low power mode.

In the LIN 2.0 specification, a wake-up request is sent by forcing the bus into a dominant state for 250 µs to 5 ms. To achieve this, the character 0xF0 must be sent to impose five consecutive dominant bits. This character expects the specified timing regardless of the baud rate:

- Baud rate min = 1 kb/s -> t_{bit} = 1 ms -> 5 t_{bit}s = 5 ms
- Baud rate min = 20 kb/s -> t_{bit} = 50 µs -> 5 t_{bit}s = 250 µs

In the LIN 1.3 specification, to impose eight consecutive dominant bits, the character 0x80 is used to generate a wake-up request.

Users can choose to send a LIN 2.0 wake-up request (WKUPTYP = 0) or a LIN 1.3 wake-up request (WKUPTYP = 1) via the WKUPTYP bit in the LIN mode register (USART_LINMR).

To transmit the wake-up request, the LINWKUP bit in the control register (USART_CR) must be set to 1. After the transmission is complete, the LINTC flag in the status register (USART_SR) is set. It can be cleared by writing 1 to the RSTSTA bit in USART_CR.

31.4.9.4 Bus Idle Timeout

If the LIN bus remains inactive for a certain period, slave nodes should automatically enter the low power mode. In the LIN 2.0 specification, the timeout is fixed at 4 seconds. In the LIN 1.3 specification, it is fixed at 25000 t_{bit}s.

In the slave node configuration, the receiver detects the idle state on the RXD line for timeout. When a timeout is detected, the TIMEOUT bit in the channel status register (USART_CSR) is set high, generating an interrupt to indicate that the driver should enter the sleep mode.

The timeout delay period (during which the receiver waits to receive a new character) is configured in the TO field of the receiver timeout register (USART_RTOR). If TO is written to 0, the receiver timeout is disabled and no timeout is detected. The TIMEOUT bit in USART_SR remains at 0. Otherwise, the receiver loads a 17-bit counter with the TO set value. This counter decrements with each bit period and is reloaded each time a new character is received. If the counter reaches 0, the TIMEOUT bit in USART_CSR is set high.

If STTTO is executed, the counter clock stops until the first character is received.

If RETTO is executed, the counter immediately starts decrementing from the TO value.

31.4.10 Test Mode

The USART can be configured into three different test modes. The internal loop back enables on-board diagnostics. In loop back mode, it is configured for internal or external loop back depending on whether the USART interface pins are connected or not.

31.4.10.1 Normal Mode

In normal mode, the RXD pin is connected to the receiver input, while the TXD pin is connected to the transmitter output.

Normal mode configuration:

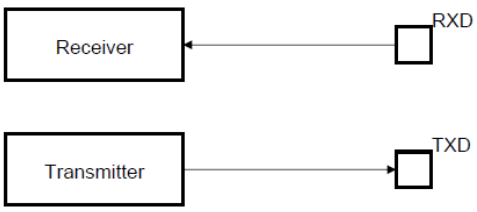


Figure 31-41: USART Normal Mode

31.4.10.2 Auto-response Mode

The auto-response mode allows one-bit retransmission. After receiving a bit at the RXD pin, send it to the TXD pin, as shown in the figure below. Programming the transmitter does not affect the TXD pin; the RXD pin remains connected to the receiver input, keeping the receiver active.

Auto-response mode configuration:

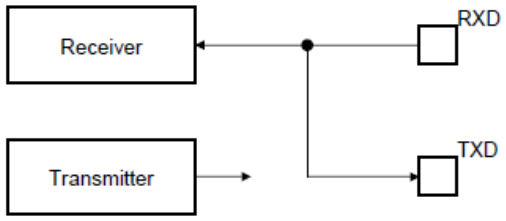


Figure 31-42: USART Auto-response Mode

31.4.10.3 Local Loop-back Mode

In local loop-back mode, the transmitter output is directly connected to the receiver input, as shown in the figure below. The TXD and RXD pins are unused. The RXD pin is inactive for the receiver, while the TXD pin is always high as in the idle state.

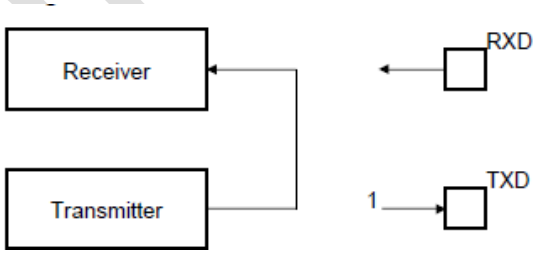


Figure 31-43: USART Local Loop-back Mode

31.4.10.4 Remote Loop-back Mode

In remote loop-back mode, the RXD pin is directly connected to the TXD pin, as shown in the figure below. Disabling the transmitter and receiver has no effect. This mode allows one-bit retransmission.

Remote loop-back mode configuration:

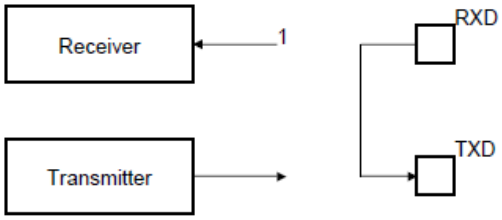


Figure 31-44: USART Remote Loop-back Mode

31.5 Register Description

USART6 register base address: 0x40B0_4000

USART7 register base address: 0x4700_C000

The registers are listed below:

Table 31-9: List of USART6 & USART7 Registers

Offset Address	Name	Description
0x00	USART_CR	Control register
0x04	USART_MR	Mode register
0x08	USART_IER	Interrupt enable register
0x0C	USART_IDR	Interrupt disable register
0x10	USART_IMR	Interrupt mask register
0x14	USART_CSR	Channel status register
0x18	USART_RHR	Receiver holding register
0x1C	USART_THR	Transmitter holding register
0x20	USART_BRGR	Baud rate generator register
0x24	USART_RTOR	Receiver timeout register
0x28	USART_TTGR	Transmitter time protection register
0x2C–0x3C	–	Reserved
0x40	USART_FIDI	FI/DI ratio register
0x44–0x48	-	Reserved
0x4C	USART_IF	IrDA filter register
0x50	USART_MAN	Manchester configuration register
0x54	USART_LINMR	LIN mode register
0x58	USART_LINIR	LIN identifier register
0x5C	USART_LINBRR	LIN baud rate register

Offset Address	Name	Description
0x60-0xE0	–	Reserved
0xE4	USART_WPMR	Write protection mode register
0xE8	USART_WPSR	Write protection status register

31.5.1 USART Control Register (USART_CR)

Offset address: 0x00

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:22	RSV	-	-	Reserved
21	LINWKUP	W	0x0	Send LIN wakeup signal: 0: invalid 1: send a wakeup signal to the LIN bus
20	LINABT	W	0x0	Abort LIN transmission: 0: invalid 1: abort the current LIN transmission
19	RTSDIS	W	0x0	Request transmission disable: 0: invalid 1: drive the RTS pin to 1
18	RTSEN	W	0x0	Request transmission enable: 0: invalid 1: drive the RTS pin to 0
17:16	RSV	-	-	Reserved
15	RETTO	W	0x0	Reload and start timeout: 0: invalid 1: restart timeout
14	RSTNACK	W	0x0	No acknowledgment reset: 0: invalid 1: reset NACK in the USART_CSR register
13	RSTIT	W	0x0	Iteration reset: 0: invalid 1: reset iteration in the USART_CSR register
12	SENDA	W	0x0	Send address: 0: invalid 1: only applicable to multipoint mode, send the

Bit	Name	Attribute	Reset Value	Description
				address character written to USART_THR
11	STTTO	W	0x0	Start timeout: 0: invalid 1: wait for one character before the timeout counter starts counting, reset the TIMEOUT status bit in USART_CSR
10	STPBRK	W	0x0	Stop break: 0: invalid 1: Stop sending breaks after at least one character time and a high level of 12-bit cycle. If a break has been sent, this option is invalid.
9	STTBRK	W	0x0	Start break: 0: invalid 1: Start sending a break after there is a character in USART_THR and the character has been sent from the shift register. If a break has been sent, this option is invalid.
8	RSTSTA	W	0x0	Status bit reset: 0: invalid 1: reset the status bits PARE, FRAME, OVRE, MANERR and XBRK in the USART_CSR register
7	TXDIS	W	0x0	Transmitter disable: 0: invalid 1: transmitter disabled
6	TXEN	W	0x0	Transmitter enable: 0: invalid 1: transmitter enabled when TXDIS is 0
5	RXDIS	W	0x0	Receiver disable: 0: invalid 1: receiver disabled
4	RXEN	W	0x0	Receiver enable: 0: invalid 1: receiver enabled when RXDIS is 0
3	RSTTX	W	0x0	Transmitter reset: 0: invalid 1: transmitter reset

Bit	Name	Attribute	Reset Value	Description
2	RSTRX	W	0x0	Receiver reset: 0: invalid 1: receiver reset
1:0	RSV	-	-	Reserved

31.5.2 USART Control Register (USART_CR, SPI Mode)

Offset address: 0x00

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:20	RSV	-	-	Reserved
19	RCS	W	0x0	Cancel SPI chip select: 0: invalid 1: release the slave select line NSS (RTS pin)
18	FCS	W	0x0	Force SPI chip select: 0: invalid 1: Force the slave select line NSS (RTS pin) to 0, even if USART is not sending data, to enable the SPI slave device to support CSAAT (chip select active after transfer) mode.
17:9	RSV	-	-	Reserved
8	RSTSTA	W	0x0	Status bit reset: 0: invalid 1: reset the status bits PARE, FRAME, OVRE, MANERR and XBRK in the USART_CSR register
7	TXDIS	W	0x0	Transmitter disable: 0: invalid 1: transmitter disabled
6	TXEN	W	0x0	Transmitter enable: 0: invalid 1: transmitter enabled when TXDIS is 0
5	RXDIS	W	0x0	Receiver disable: 0: invalid 1: receiver disabled

Bit	Name	Attribute	Reset Value	Description
4	RXEN	W	0x0	Receiver enable: 0: invalid 1: receiver enabled when RXDIS is 0
3	RSTTX	W	0x0	Transmitter reset: 0: invalid 1: transmitter reset
2	RSTRX	W	0x0	Receiver reset: 0: invalid 1: receiver reset
1:0	RSV	-	-	Reserved

31.5.3 USART Mode Register (SPI USART_MR)

Offset address: 0x04

Reset value: 0xC000 0000

Bit	Name	Attribute	Reset Value	Description
31	ONEBIT	R/W	0x1	Frame start delimiter selector: 0: Frame start delimiter is COMMAND or DATASYNC. 1: Frame start delimiter is a single bit.
30	MODSYNC	R/W	0x1	Manchester synchronization mode: 0: Manchester start bit is a level transition from 0 to 1. 1: Manchester start bit is a level transition from 1 to 0.
29	MAN	R/W	0x0	Manchester encoder/decoder enable: 0: Manchester encoder/decoder disabled 1: Manchester encoder/decoder enabled
28	FILTER	R/W	0x0	Infrared receiver line filter: 0: USART does not filter the receive line. 1: USART uses a 3-point sampling filter (1/16-bit clock) (2/3 more) to filter the receive line.
27:24	RSV	-	-	Reserved
23	INVDATA	R/W	0x0	Data inversion: 0: The data transmitted on the TXD line

Bit	Name	Attribute	Reset Value	Description
				matches the data written to the USART_THR register, or the content read from the USART_RHR register matches the data received on the RXD line. This is the normal mode operation. 1: The data transmitted on the TXD line is inverted (only the voltage polarity) compared to the value written to the USART_THR register; or the value read from the USART_RHR register is inverted compared to the data received on the RXD line. This is the inversion mode operation, which is particularly useful in non-contact smart card applications and can be configured through the MSBF bit.
22	VAR_SYNC	R/W	0x0	Multiple synchronization methods of command/data synchronization start frame separator: 0: User-defined command configuration or data sync field depends on the value of MODSYNC. 1: The sync field is updated when a character is written to the USART_THR register.
21	DSNACK	R/W	0x0	Disable continuous NACK: 0: Once a received character has a parity error, NACK is sent on the ISO line (unless INACK is set). 1: When the number of consecutive parity errors has not reached the value given by the MAX_ITERATION field, these parity errors generate NACK on the ISO line. Once that value is reached, no additional NACK is sent on the ISO line, and the ITERATION flag is set.
20	INACK	R/W	0x0	Suppress no acknowledgment: 0: Generate NACK. 1: Do not generate NACK. Note: In SPI master mode, if INACK is 0 and a character is written to

Bit	Name	Attribute	Reset Value	Description
				the USART_THR register (assuming TXRDY is set), the character start bit is sent immediately. When INACK is 1, another condition must also be met, which is that when a character is written to the USART_THR register, the character is sent immediately only if the RXRDY flag is cleared to 0 (achieved by reading the receiver holding register).
19	OVER	R/W	0x0	Oversampling mode: 0: 16 times oversampling 1: 8 times oversampling
18	CLKO	R/W	0x0	Clock output selection: 0: USART does not drive the SCK pin. 1: If USCLKS has not selected an external clock SCK, USART drives the SCK pin.
17	MODE9	R/W	0x0	Character length of 9 bits: 0: CHRL defines the character length. 1: The character length is 9 bits.
16	MSBF	R/W	0x0	Bit order: 0: LSB first 1: MSB first
15:14	CHMODE	R/W	0x0	Channel mode: 00: normal mode 01: auto-response mode, with receiver input connected to TXD pin 10: local loop-back mode, with transmitter output connected to receiver input 11: remote loop-back mode, with RXD pin connected to TXD pin
13:12	NBSTOP	R/W	0x0	Number of stop bit: Asynchronous (SYNC = 0) 00: 1 01: 1.5 10: 2 11: reserved Synchronous (SYNC = 1)

Bit	Name	Attribute	Reset Value	Description
				00: 1 01: reserved 10: 2 11: reserved
11:9	PAR	R/W	0x0	Parity type: 000: even parity 001: odd parity 010: parity forced to 0 (space) 011: parity forced to 1 (mark) 10x: no parity 11x: multi-point mode
8	SYNC	R/W	0x0	Synchronous mode selection: 0: USART operates in asynchronous mode. 1: USART operates in synchronous mode.
7:6	CHRL	R/W	0x0	Character length: 00: 5 bits 01: 6 bits 10: 7 bits 11: 8 bits
5:4	USCLKS	R/W	0x0	Clock selection: 00: MCK 01: MCK/DIV 10: reserved 11: SCK DIV setting: USART6: CFGR1[27:24] USART7: CFGR1[31:28]
3:0	USART_MODE	R/W	0x0	Operation mode: 0000: normal mode 0010: reserved 1000: IrDA 1010: LIN master 1011: LIN slave 1110: SPI master 1111: SPI slave Others: reserved

31.5.4 USART Mode Register (USART_MR, SPI Mode)

Offset address: 0x04

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:21	RSV	-	-	Reserved
20	WRDBT	R/W	0x0	Waiting for data to be read before transmission: 0: Character transmission begins immediately after the character is written to the USART_THR register (assuming TXRDY is set). 1: Character transmission starts only after the RXRDY flag is cleared (USART_RHR has been read).
19	RSV	-	-	Reserved
18	CLKO	R/W	0x0	Clock output selection: 0: USART does not drive the SCK pin. 1: If USCLKS has not selected an external clock SCK, USART drives the SCK pin.
17	RSV	-	-	Reserved
16	CPOL	R/W	0x0	SPI clock polarity: 0: The inactive state of SPCK is logic low (0). 1: The inactive state of SPCK is logic high (1).
15:9	RSV	-	-	Reserved
8	CPHA	R/W	0x0	SPI clock phase: 0: USART operates in asynchronous mode. 1: USART operates in synchronous mode.
7:6	CHRL	R/W	0x0	Character length: 11: 8 bits Others: reserved
5:4	USCLKS	R/W	0x0	Clock selection: 00: MCK 01: MCK/DIV 10: reserved 11: SCK DIV setting:

Bit	Name	Attribute	Reset Value	Description
				USART6: CFGR1[27:24] USART7: CFGR1[31:28]
3:0	USART_MODE	R/W	0x0	Operation mode: 1110: SPI master 1111: SPI slave Others: reserved

31.5.5 USART Interrupt Enable Register (UART_IER)

Offset address: 0x08

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:21	RSV	-	-	Reserved
20	MANE	W	0x0	Manchester error interrupt enable
19	CTSIC	W	0x0	Transmission input change clear interrupt enable
18:14	RSV	-	-	Reserved
13	NACK	W	0x0	NACK interrupt enable
12:11	RSV	-	-	Reserved
10	ITER	W	0x0	Reaching maximum repetition count interrupt enable
9	TXEMPTY	W	0x0	TXEMPTY interrupt enable
8	TIMEOUT	W	0x0	Timeout interrupt enable
7	PARE	W	0x0	Parity error interrupt enable
6	FRAME	W	0x0	Frame error interrupt enable
5	OVRE	W	0x0	Overflow error interrupt enable
4:3	RSV	-	-	Reserved
2	RXBRK	W	0x0	Receiver break interrupt enable
1	TXRDY	W	0x0	TXRDY interrupt enable
0	RXRDY	W	0x0	RXRDY interrupt enable

31.5.6 USART Interrupt Enable Register (USART_IER, SPI Mode)

Offset address: 0x08

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:11	RSV	-	-	Reserved
10	UNRE	W	0x0	SPI underrun error interrupt enable
9	TXEMPTY	W	0x0	TXEMPTY interrupt enable
8:6	RSV	-	-	Reserved
5	OVRE	W	0x0	Overflow error interrupt enable
4:2	RSV	-	-	Reserved
1	TXRDY	W	0x0	TXRDY interrupt enable
0	RXRDY	W	0x0	RXRDY interrupt enable

31.5.7 USART Interrupt Enable Register (USART_IER, LIN Mode)

Offset address: 0x08

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31	LINHTE	W	0x0	LIN message header timeout error interrupt enable
30	LINSTE	W	0x0	LIN synchronization tolerance error interrupt enable
29	LINSNRE	W	0x0	LIN slave NACK error interrupt enable
28	LINCE	W	0x0	LIN checksum error interrupt enable
27	LINIPE	W	0x0	LIN identifier parity interrupt enable
26	LINISFE	W	0x0	LIN sync field inconsistency error interrupt enable
25	LINBE	W	0x0	LIN bus error interrupt enable
24:16	RSV	-	-	Reserved
15	LINTC	W	0x0	LIN transmission complete interrupt enable
14	LINID	W	0x0	Transmit or receive LIN identifier interrupt enable
13	LINBK	W	0x0	Transmit or receive LIN break interrupt enable
12:11	RSV	-	-	Reserved
10	RSV	-	-	Reserved
9	TXEMPTY	W	0x0	TXEMPTY interrupt enable
8	TIMEOUT	W	0x0	Timeout interrupt enable
7	PARE	W	0x0	Parity error interrupt enable
6	FRAME	W	0x0	Frame error interrupt enable
5	OVRE	W	0x0	Overflow error interrupt enable

Bit	Name	Attribute	Reset Value	Description
4:2	RSV	-	-	Reserved
1	TXRDY	W	0x0	TXRDY interrupt enable
0	RXRDY	W	0x0	RXRDY interrupt enable

31.5.8 USART Interrupt Disable Register (USART_IDR)

Offset address: 0x0C

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:21	RSV	-	-	Reserved
20	MANE	W	0x0	Manchester error interrupt disable
19	CTSIC	W	0x0	Transmission input change clear interrupt disable
18:14	RSV	-	-	Reserved
13	NACK	W	0x0	NACK interrupt disable
12:11	RSV	-	-	Reserved
10	ITER	W	0x0	Reaching maximum repetition count interrupt disable
9	TXEMPTY	W	0x0	TXEMPTY interrupt disable
8	TIMEOUT	W	0x0	Timeout interrupt disable
7	PARE	W	0x0	Parity error interrupt disable
6	FRAME	W	0x0	Frame error interrupt disable
5	OVRE	W	0x0	Overflow error interrupt disable
4:3	RSV	-	-	Reserved
2	RXBRK	W	0x0	Receiver break interrupt disable
1	TXRDY	W	0x0	TXRDY interrupt disable
0	RXRDY	W	0x0	RXRDY interrupt disable

31.5.9 USART Interrupt Disable Register (USART_IDR, SPI Mode)

Offset address: 0x0C

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:11	RSV	-	-	Reserved
10	UNRE	W	0x0	SPI underrun error interrupt disable

Bit	Name	Attribute	Reset Value	Description
9	TXEMPTY	W	0x0	TXEMPTY interrupt disable
8:6	RSV	-	-	Reserved
5	OVRE	W	0x0	Overflow error interrupt disable
4:2	RSV	-	-	Reserved
1	TXRDY	W	0x0	TXRDY interrupt disable
0	RXRDY	W	0x0	RXRDY interrupt disable

31.5.10 USART Interrupt Disable Register (USART_IDR, LIN Mode)

Offset address: 0x0C

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31	LINHTE	W	0x0	LIN message header timeout error interrupt disable
30	LINSTE	W	0x0	LIN synchronization tolerance error interrupt disable
29	LINSNRE	W	0x0	LIN slave NACK error interrupt disable
28	LINCE	W	0x0	LIN checksum error interrupt disable
27	LINIPE	W	0x0	LIN identifier parity interrupt disable
26	LINISFE	W	0x0	LIN sync field inconsistency error interrupt disable
25	LINBE	W	0x0	LIN bus error interrupt disable
24:16	RSV	-	-	Reserved
15	LINTC	W	0x0	LIN transmission complete interrupt disable
14	LINID	W	0x0	Transmit or receive LIN identifier interrupt disable
13	LINBK	W	0x0	Transmit or receive LIN break interrupt disable
12:10	RSV	-	-	Reserved
9	TXEMPTY	W	0x0	TXEMPTY interrupt disable
8	TIMEOUT	W	0x0	Timeout interrupt disable
7	PARE	W	0x0	Parity error interrupt disable
6	FRAME	W	0x0	Frame error interrupt disable
5	OVRE	W	0x0	Overflow error interrupt disable
4:2	RSV	-	-	Reserved

Bit	Name	Attribute	Reset Value	Description
1	TXRDY	W	0x0	TXRDY interrupt disable
0	RXRDY	W	0x0	RXRDY interrupt disable

31.5.11 USART Interrupt Mask Register (USART_IMR)

Offset address: 0x10

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:25	RSV	-	-	Reserved
24:21	RSV	-	-	Reserved
20	MANE	R	0x0	Manchester error interrupt mask
19	CTSIC	R	0x0	Transmission input change clear interrupt mask
18:14	RSV	-	-	Reserved
13	NACK	R	0x0	NACK interrupt mask
12:11	RSV	-	-	Reserved
10	ITER	R	0x0	Reaching maximum repetition count interrupt mask
9	TXEMPTY	R	0x0	TXEMPTY interrupt mask
8	TIMEOUT	R	0x0	Timeout interrupt mask
7	PARE	W	0x0	Parity error interrupt mask
6	FRAME	R	0x0	Frame error interrupt mask
5	OVRE	R	0x0	Overflow error interrupt mask
4:3	RSV	-	-	Reserved
2	RXBRK	R	0x0	Receiver break interrupt mask
1	TXRDY	R	0x0	TXRDY interrupt mask
0	RXRDY	R	0x0	RXRDY interrupt mask

31.5.12 USART Interrupt Mask Register (USART_IMR, SPI Mode)

Offset address: 0x10

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:11	RSV	-	-	Reserved
10	UNRE	R	0x0	SPI underrun error interrupt mask

Bit	Name	Attribute	Reset Value	Description
9	TXEMPTY	R	0x0	TXEMPTY interrupt mask
8:6	RSV	-	-	Reserved
5	OVRE	R	0x0	Overflow error interrupt mask
4:2	RSV	-	-	Reserved
1	TXRDY	R	0x0	TXRDY interrupt mask
0	RXRDY	R	0x0	RXRDY interrupt mask

31.5.13 USART Interrupt Mask Register (USART_IMR, LIN Mode)

Offset address: 0x10

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31	LINHTE	R	0x0	LIN message header timeout error interrupt mask
30	LINSTE	R	0x0	LIN synchronization tolerance error interrupt mask
29	LINSNRE	R	0x0	LIN slave NACK error interrupt mask
28	LINCE	R	0x0	LIN checksum error interrupt mask
27	LINIPE	R	0x0	LIN identifier parity interrupt mask
26	LINISFE	R	0x0	LIN sync field inconsistency error mask
25	LINBE	R	0x0	LIN bus error interrupt mask
24:16	RSV	-	-	Reserved
15	LINTC	R	0x0	LIN transmission complete interrupt mask
14	LINID	R	0x0	Transmit or receive LIN identifier interrupt mask
13	LINBK	R	0x0	Transmit or receive LIN break interrupt mask
12:10	RSV	-	-	Reserved
9	TXEMPTY	R	0x0	TXEMPTY interrupt mask
8	TIMEOUT	R	0x0	Timeout interrupt mask
7	PARE	R	0x0	Parity error interrupt mask
6	FRAME	R	0x0	Frame error interrupt mask
5	OVRE	R	0x0	Overflow error interrupt mask
4:2	RSV	-	-	Reserved
1	TXRDY	R	0x0	TXRDY interrupt mask
0	RXRDY	R	0x0	RXRDY interrupt mask

31.5.14 USART Channel Status Register (USART_CSR)

Offset address: 0x14

Reset value: 0x0078 0000

Bit	Name	Attribute	Reset Value	Description
31:25	RSV	-	-	Reserved
24	MANERR	R	0x0	Manchester error: 0: no Manchester error detected since the last RSTSTA 1: at least one Manchester error detected since the last RSTSTA
23	CTS	R	0x0	CTS input mirror: 0: CTS is 0. 1: CTS is 1.
22:20	RSV	-	-	Reserved
19	CTSIC	R	0x1	Transmission input change clear flag: 0: no input change detected on the CTS pin since the last read of USART_CSR 1: at least one input change detected on the CTS pin since the last read of USART_CSR
18:14	RSV	-	-	Reserved
13	NACK	R	0x0	No acknowledgment: 0: no acknowledgment detected since the last RSTNACK 1: at least one no-acknowledgment detected since the last RSTNACK
12:11	RSV	-	-	Reserved
10	ITER	R	0x0	Reaching maximum repetition count: 0: The maximum repetition count has not been reached since the last RSTSTA. 1: The maximum repetition count has been reached since the last RSTSTA.
9	TXEMPTY	R	0x0	Transmitter empty: 0: There is at least one character in either USART_THR or the transmitter shift register, or the transmitter is disabled. 1: There are no characters in either

Bit	Name	Attribute	Reset Value	Description
				USART_THR or the transmitter shift register.
8	TIMEOUT	R	0x0	Receiver timeout: 0: No timeout has occurred since the last timeout command STTTO was initiated, or the timeout register is 0. 1: A timeout has occurred since the last timeout command STTTO was initiated.
7	PARE	R	0x0	Parity error: 0: no parity error detected since the last RSTSTA 1: at least one parity error detected since the last RSTSTA
6	FRAME	R	0x0	Frame error: 0: no stop bit detected since the last RSTSTA 1: at least one stop bit detected since the last RSTSTA
5	OVRE	R	0x0	Overflow error: 0: no overflow error occurred since the last RSTSTA 1: at least one overflow error occurred since the last RSTSTA
4:3	RSV	-	-	Reserved
2	RXBRK	R	0x0	Break receive/complete: 0: no break receive or break complete detected since the last RSTSTA 1: break receive or break complete detected since the last RSTSTA
1	TXRDY	R	0x0	Transmitter ready: 0: There are characters waiting to be sent from USART_THR to the transmitter shift register, or the STTBRK command is requested, or the transmitter is disabled. Once the transmitter is enabled, TXRDY becomes 1. 1: There are no characters in USART_THR.
0	RXRDY	R	0x0	Receiver ready: 0: No complete character has been received

Bit	Name	Attribute	Reset Value	Description
				since the last read of USART_RHR, or the receiver is disabled. If the receiver is disabled while receiving characters, RXRDY becomes 1 when the receiver is enabled. 1: At least one complete character has been received since the last read of USART_RHR, and USART_RHR has not been read.

31.5.15 USART Channel Status Register (USART_CSR, SPI Mode)

Offset address: 0x14

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:11	RSV	-	-	Reserved
10	UNRE	R	0x0	SPI underrun error: 0: no SPI underrun error occurred since the last RSTSTA 1: at least one SPI underrun error occurred since the last RSTSTA
9	TXEMPTY	R	0x0	Transmitter empty: 0: There is at least one character in either USART_THR or the transmitter shift register, or the transmitter is disabled. 1: There are no characters in either USART_THR or the transmitter shift register.
8:6	RSV	-	-	Reserved
5	OVRE	R	0x0	Overflow error: 0: no overflow error occurred since the last RSTSTA 1: at least one overflow error occurred since the last RSTSTA
4:2	RSV	-	-	Reserved
1	TXRDY	R	0x0	Transmitter ready: 0: There are characters waiting to be sent from USART_THR to the transmitter shift register, or the STTBRK command is

Bit	Name	Attribute	Reset Value	Description
				requested, or the transmitter is disabled. Once the transmitter is enabled, TXRDY becomes 1. 1: There are no characters in USART_THR.
0	RXRDY	R	0x0	Receiver ready: 0: No complete character has been received since the last read of USART_RHR, or the receiver is disabled. If the receiver is disabled while receiving characters, RXRDY becomes 1 when the receiver is enabled. 1: At least one complete character has been received since the last read of USART_RHR, and USART_RHR has not been read.

31.5.16 USART Channel Status Register (USART_CSR, LIN Mode)

Offset address: 0x14

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31	LINHTE	R	0x0	LIN message header timeout error: 0: LIN message header timeout error not detected since the last RSTSTA 1: LIN message header timeout error detected since the last RSTSTA
30	LINSTE	R	0x0	LIN synchronization tolerance error: 0: LIN synchronization tolerance error not detected since the last RSTSTA 1: LIN synchronization tolerance error detected since the last RSTSTA
29	LINSNRE	R	0x0	LIN slave NACK error: 0: LIN slave NACK error not detected since the last RSTSTA 1: LIN slave NACK error detected since the last RSTSTA
28	LINCE	R	0x0	LIN checksum error:

Bit	Name	Attribute	Reset Value	Description
				0: checksum error not detected since the last RSTSTA 1: checksum error detected since the last RSTSTA
27	LINIPE	R	0x0	LIN identifier parity error: 0: LIN identifier parity error not detected since the last RSTSTA 1: LIN identifier parity error detected since the last RSTSTA
26	LINISFE	R	0x0	LIN sync field inconsistency error: 0: LIN sync field inconsistency error not detected since the last RSTSTA 1: USART configured as slave node and sync field inconsistency error detected since the last RSTSTA
25	LINBE	R	0x0	LIN bit error: 0: bit error not detected since the last RSTSTA 1: bit error detected since the last RSTSTA
24	RSV	-	-	Reserved
23	LINBLS	R	0x0	LIN bus status: 0: LIN bus is 0. 1: LIN bus is 1.
22:16	RSV	-	-	Reserved
15	LINTC	R	0x0	LIN transmission complete: 0: USART being idle or LIN transmission being in progress 1: LIN transmission completed since the last RSTSTA
14	LINID	R	0x0	Transmit or receive LIN identifier: **LIN master** 0: no LIN identifier transmitted since the last RSTSTA 1: at least one LIN identifier transmitted since the last RSTSTA **LIN slave** 0: no LIN identifier received since the last RSTSTA

Bit	Name	Attribute	Reset Value	Description
				1: at least one LIN identifier received since the last RSTSTA
13	LINBK	R	0x0	Transmit or receive LIN break: **LIN master** 0: no LIN break transmitted since the last RSTSTA 1: at least one LIN break transmitted since the last RSTSTA **LIN slave** 0: no LIN break received since the last RSTSTA 1: at least one LIN break received since the last RSTSTA
12:10	RSV	-	-	Reserved
9	TXEMPTY	R	0x0	Transmitter empty: 0: There are characters in either USART_THR or the shift register, or the transmitter is disabled. 0: There are no characters in either USART_THR or the shift register.
8	TIMEOUT	R	0x0	Timeout: 0: No timeout has occurred since the last timeout command (STTTO in USART_CR) was initiated, or the timeout register is 0. 1: A timeout has occurred since the last timeout command (STTTO in USART_CR) was initiated.
7	PARE	R	0x0	Parity error: 0: no parity error occurred since the last RSTSTA 1: at least one parity error occurred since the last RSTSTA
6	FRAME	R	0x0	Frame error: 0: no frame error occurred since the last RSTSTA 1: at least one frame error occurred since the last RSTSTA
5	OVRE	R	0x0	Overflow error: 0: no overflow error occurred since the last RSTSTA 1: at least one overflow error occurred since the

Bit	Name	Attribute	Reset Value	Description
				last RSTSTA
4:2	RSV	-	-	Reserved
1	TXRDY	R	0x0	Transmitter ready: 0: There are characters waiting to be sent from USART_THR to the transmitter shift register, or the STTBRK command is requested, or the transmitter is disabled. Once the transmitter is enabled, TXRDY becomes 1. 1: There are no characters in USART_THR.
0	RXRDY	R	0x0	Receiver ready: 0: No complete character has been received since the last read of USART_RHR, or the receiver is disabled. If the receiver is disabled while receiving characters, RXRDY becomes 1 when the receiver is enabled. 1: At least one complete character has been received since the last read of USART_RHR, and USART_RHR has not been read.

31.5.17 USART Receive Holding Register (USART_RHR)

Offset address: 0x18

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:16	RSV	-	-	Reserved
15	RXSYNH	R	0x0	Received sync field: 0: The last received character is data. 1: The last received character is a command.
14:9	RSV	-	-	Reserved
8:0	RXCHR	R	0x0	Received character: If RXRDY is set, it is the last character received.

31.5.18 USART Transmit Holding Register (USART_THR)

Offset address: 0x1C

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:16	RSV	-	-	Reserved
15	TXSYNH	W	0x0	Transmitted sync field: 0: The next character to be transmitted is encoded as data and the frame start delimiter is DATA SYNC. 1: The next character to be transmitted is encoded as a command, and the frame start delimiter is COMMAND SYNC.
14:9	RSV	-	-	Reserved
8:0	TXCHR	W	0x0	Character to be transmitted: If TXRDY is not set, the next character is transmitted after the current character.

31.5.19 USART Baud Rate Generator Register (USART_BRGR)

Offset address: 0x20

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:19	RSV	-	-	Reserved
18:16	FP	R/W	0x0	Fractional part: 0: fractional divider disabled 1–7: baud rate resolution, defined as FP x 1/8
15:0	CD	R/W	0x0	Clock frequency division Refer to Table 31-10

Table 31-10 : Reference Table of USART Clock Frequency Division

CD	USART_MODE ≠ ISO7816			USART_MODE = ISO7816
	SYNC = 0		SYNC = 1 or USART_MODE = SPI (Master or Slave)	
	OVER = 0	OVER = 1		
0	Baud rate clock disabled			
1–65535	Baud rate = selected clock / 16 / CD	Baud rate = selected clock / 8 / CD	Baud rate = selected clock / CD	Baud rate = selected clock / CD / FI_DI_RATIO

31.5.20 USART Receive Timeout Register (USART_RTOR)

Offset address: 0x24

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:17	RSV	-	-	Reserved
16:0	TO	R/W	0x0	Timeout value: 0: receiver timeout disabled 1–131071: receiver timeout enabled and timeout delay of TO bit cycles

31.5.21 USART Transmit Time Protection Register (USART_TTGR)

Offset address: 0x28

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:8	RSV	-	-	Reserved
7:0	TG	R/W	0x0	Time protection value: 0: transmitter time protection disabled 1–255: transmitter time protection enabled and time protection delay of TG bit cycles

31.5.22 USART FI/DI Ratio Register (USART_FIDI)

Offset address: 0x40

Reset value: 0x0000 0174

Bit	Name	Attribute	Reset Value	Description
31:11	RSV	-	-	Reserved
15:0	FI_DI_RATIO	R/W	0x174	Ratio of FI to DI

31.5.23 USART IrDA Filter Register (USART_IF)

Offset address: 0x4C

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:8	RSV	-	-	Reserved
7:0	IrDA_FILTER	R/W	0x0	IrDA filter: The value of IrDA filter shall meet the following criteria: t_{MCK} * (IRDA_FILTER + 3) < 1.41 μs

31.5.24 USART Manchester Configuration Register (USART_MAN)

Offset address: 0x50

Reset value: 0xB001 1004

Bit	Name	Attribute	Reset Value	Description
31	RSV	-	-	Reserved
30	DRIFT	R/W	0x0	Drift compensation: 0: This USART cannot recover from a severe clock drift. 1: This USART can recover from clock drift. It must be realized in 16 times clock mode.
29	ONE	R/W	0x1	Must be set to 1: When writing to the USART_MAN register, this bit must always be set to 1.
28	RX_MPOL	R/W	0x1	Receiver Manchester polarity: 0: Logic 0 is encoded as a transition from 0 to 1, and logic 1 is encoded as a transition from 1 to 0. 1: Logic 0 is encoded as a transition from 1 to 0, and logic 1 is encoded as a transition from 0 to 1.
27:26	RSV	-	-	Reserved

Bit	Name	Attribute	Reset Value	Description
25:24	RX_PP	R/W	0x0	Receiver preamble pattern: 00: ALL_ONE 01: ALL_ZERO 10: ZERO_ONE 11: ONE_ZERO
23:20	RSV	-	-	Reserved
19:16	RX_PL	R/W	0x1	Receiver preamble length: 0: The generation of receiver preamble is disabled. 1-15: The length of the detected preamble is RX_PL bit periods.
15:13	RSV	-	-	Reserved
12	TX_MPOL	R/W	0x1	Transmitter Manchester polarity: 0: Logic 0 is encoded as a transition from 0 to 1, and logic 1 is encoded as a transition from 1 to 0. 1: Logic 0 is encoded as a transition from 1 to 0, and logic 1 is encoded as a transition from 0 to 1.
11:10	RSV	-	-	Reserved
9:8	TX_PP	R/W	0x0	Transmitter preamble pattern: 00: ALL_ONE 01: ALL_ZERO 10: ZERO_ONE 11: ONE_ZERO
7:4	RSV	-	-	Reserved
3:0	TX_PL	R/W	0x4	Transmitter preamble length: 0: The generation of transmitter preamble is disabled. 1-15: The preamble length is TX_PL bit periods.

31.5.25 USART LIN Mode Register (USART_LINMR)

Offset address: 0x54

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:18	RSV	-	-	Reserved
17	SYNCDIS	R/W	0x0	Synchronization disable: 0: synchronization performed in the LIN slave node configuration 1: synchronization not performed in the LIN slave node configuration
16	PDCM	R/W	0x0	DMA mode: 0: DMA cannot write to the LIN mode register USART_LINMR. 1: DMA writes to the LIN mode register USART_LINMR (except for flags).
15:8	DLC	R/W	0x0	Data length: When DLM = 0, it defines the response data length, which is equal to DLC + 1 bytes.
7	WKUPTYP	R/W	0x0	Wake-up signal type: 0: Set the LINWKUP bit in the control register to send a LIN 2.0 wake-up signal. 1: Set the LINWKUP bit in the control register to send a LIN 1.3 wake-up signal.
6	FSDIS	R/W	0x0	Frame slot mode disable: 0: frame slot mode enabled 1: frame slot mode disabled
5	DLM	R/W	0x0	Data length mode: 0: The response data length is defined by the DLC of this register. 1: The response data length is defined by bit 5 and bit 6 of the identifier (IDCHR of USART_LINIR).
4	CHKTYP	R/W	0x0	Checksum type: 0: LIN 2.0 “enhanced” checksum 1: LIN 1.3 “classic” checksum
3	CHKDIS	R/W	0x0	Checksum disable: 0: In the master node configuration, the checksum is automatically calculated and sent. In the slave node configuration, the checksum is automatically checked. 1: Regardless of how the node is configured, the

Bit	Name	Attribute	Reset Value	Description
				checksum will not be calculated/sent, nor will it be checked.
2	PARDIS	R/W	0x0	Parity disable: 0: In the master node configuration, the identifier parity is automatically calculated and sent. In both master node and slave node configurations, the parity is automatically checked. 1: Regardless of how the node is configured, the identifier parity will not be calculated/sent, nor will it be checked.
1:0	NACT	R/W	0x0	LIN node operation: 00: PUBLISH (send a response) 01: SUBSCRIBE (receive a response) 10: IGNORE (neither send nor receive a response) 11: reserved

31.5.26 USART LIN Identifier Register (USART_LINIR)

Offset address: 0x58

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:8	RSV	-	-	Reserved
7:0	IDCHR	R/W	0x0	Identifier character: When USART_MODE = 0xA (master node), IDCHR is read/write, and its value is the identifier character to be transmitted. When USART_MODE = 0xB (slave node), IDCHR is read-only, and its value is the identifier character to be received.

31.5.27 USART LIN Baud Rate Register (USART_LINBRR)

Offset address: 0x5C

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:19	RSV	-	-	Reserved
18:16	LINFP	R	0x0	Fractional part after synchronization
15:0	LINCD	R	0x0	Clock frequency division after synchronization

31.5.28 USART Write Protection Mode Register (USART_WPMR)

Offset address: 0xE4

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:8	WPKEY	W	0x0	Writing any value other than 0x555341 in this field will abort the write operation for the WPEN bit.
7:1	RSV	-	-	Reserved
0	WPEN	R/W	0x0	0: Write protection is disabled when WPKEY is 0x555341. 1: Write protection is enabled when WPKEY is 0x555341.

Note: The following registers shall be protected:

- USART mode register (USART_MR)
- USART baud rate generator register (USART_BRGR)
- USART receive timeout register (USART_RTOT)
- USART transmit time protection register (USART_TTGR)
- USART FI/DI ratio register (USART_FIDI)
- USART IrDA filter register (USART_IF)
- USART Manchester configuration register (USART_MAN)

31.5.29 USART Write Protection Status Register (USART_WPSR)

Offset address: 0xE8

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:24	RSV	-	-	Reserved
23:8	WPVSRC	R	0x0	Write protection conflict status: 0: No write protection conflict has occurred since the last read of USART_WPSR. 1: A write protection conflict has occurred since the last read of USART_WPSR.
7:1	RSV	-	-	Reserved
0	WPVS	R	0x0	Write protection conflict source: When it is valid, an attempt is made to write the write protection register.

31.6 Operation Procedure

31.6.1 UART Mode

31.6.1.1 Initialization

1. Enable the clock for the corresponding GPIO pins and configure the pins for USART_TX and USART_RX alternate functions.
2. Configure the system configuration register for the USART module clock.
3. Configure the USART_BRGR register to set the baud rate for serial communication.
4. Configure the USART_MR register to set the data frame length, parity type and stop bit length for serial communication.
5. Configure the USART_CR register to enable transmission and reception.
6. Configure the USART_IER register to enable the required interrupts.

31.6.1.2 Transmission Process

1. Before transmitting data, the software can configure the baud rate parameter, parity type and data frame format.
2. Write the first data byte to the USART_THR register.
3. Query the transmission complete flag USART_CSR[9], if USART_CSR[9] = 1, the current data transmission is completed.
4. Write the next data byte to USART_THR.

31.6.1.3 Reception Process

1. Before receiving data, the software can configure the baud rate parameters, parity type and data frame format.
2. For data reception, query the USART_CSR[0] flag or wait for an interrupt. If USART_CSR[0] = 1, it means the receiver is not empty, then read the data from the USART_RHR, after which the corresponding flag will be automatically cleared.
3. If there is an error in data reception, wait for the interrupt or query the USART_CSR register flag bit to determine the error type and execute corresponding error handling, after which the software clears the error flag bit.
4. Continue to receive data.

31.6.2 SPI Mode

31.6.2.1 Master Mode

1. Initialization
   A. Enable the clocks for the corresponding GPIO pins and configure the pins for USART_RTS, USART_TX, USART_RX and USART_SCK alternate functions.

B. Configure the system configuration register for the USART module clock.

C. Set USART_MR[3:0] to SPI master mode.

D. Configure USART_MR[7:6] to set the character length to 8 bits.

E. Configure USART_MR[18] to set the SCK pin to output clock.

F. Configure the USART_BRGR register to set the clock frequency division (the internal clock is divided by more than 6).

G. Configure USART_MR[16] and USART_MR[8] to set the SPI clock polarity and clock phase.

H. Configure the USART_CR register to enable the SPI transceiver.

2. Full-duplex data transmission

A. Set USART_CR[18] to 1 to enable the chip select (CS) to go low.

B. Wait until the transmitter is ready, then write data to the USART_THR register.

C. Query the transmission complete flag USART_CSR[9], if USART_CSR[9] = 1, the current data transmission is completed.

D. Query the USART_CSR flag or wait for an interrupt, If USART_CSR[0] = 1, read the data from the USART_RHR, after which the corresponding flag will be automatically cleared.

E. To continue transmitting and receiving data, repeat steps B–D.

F. After completing data transmission, set USART_CR[19] to 1 to disable the chip select (CS) to go high.

31.6.2.2 Slave Mode

1. Initialization

   A. Enable the clocks for the corresponding GPIO pins and configure the pins for USART_CTS, USART_TX, USART_RX and USART_SCK alternate functions.

   B. Configure the system configuration register for the USART module clock.

   C. Set USART_MR[3:0] to SPI slave mode.

   D. Configure USART_MR[7:6] to set the character length to 8 bits.

   E. Set USART_MR[18] to 3 to select SCK as the clock.

   F. Configure USART_MR[16] and USART_MR[8] to set the SPI clock polarity and clock phase.

   G. Configure the USART_CR register to enable the SPI transceiver.

2. Full-duplex data transmission

   A. Wait until the transmitter is ready, then write data to the USART_THR register.

   B. Query the transmission complete flag USART_CSR[9], if USART_CSR[9] = 1, the current data transmission is completed.

   C. Query the USART_CSR flag or wait for an interrupt, If USART_CSR[0] = 1, read the data from the USART_RHR, after which the corresponding flag will be automatically cleared.

   D. To continue transmitting and receiving data, repeat steps A–C.

31.6.3 LIN Mode

31.6.3.1 Master Node Mode

1. Initialization

   A. Enable the clock for the corresponding GPIO pins and configure the pins for USART_TX and USART_RX alternate functions.

   B. Configure the system configuration register for the USART module clock.

   C. Set USART_MR[3:0] to LIN master node mode.

   D. Configure the USART_LINMR register to set the operating mode of the master node.

   E. Configure the USART_BRGR register to set the baud rate for LIN communication.

2. Transmitting data

   A. Configure USART_LINMR[1:0] to set the master node operation mode as PUBLISH to send an acknowledgment.

   B. Configure USART_LINMR[15:8] to set the data length of the LIN frame.

   C. Wait for USART_CSR[1] to be set, indicating the transmitter is ready, then write the ID identifier into USART_LINIR[7:0].

   D. Wait for USART_CSR[1] to be set, indicating the ID identifier has been sent, then write the data to be transmitted into USART_THR, and wait for USART_CSR[1] to be set again indicating the data transmission is complete before writing the next data, until all data is transmitted.

   E. To transmit data again, repeat steps C and D.

3. Receiving data

   A. Configure USART_LINMR[1:0] to set the master node operation mode as SUBSCRIBE

to receive an acknowledgment.

B. Configure USART_LINMR[15:8] to set the data length of the LIN frame.

C. Wait for USART_CSR[1] to be set, indicating the transmitter is ready, then write the ID identifier into USART_LINIR[7:0].

D. Wait for USART_CSR[0] to be set, then read the data from USART_RHR.

E. Repeat step D until all data has been read.

F. To receive data again, repeat steps C and D.

31.6.3.2 Slave Node Mode

1. Initialization

   A. Enable the clock for the corresponding GPIO pins and configure the pins for USART_TX and USART_RX alternate functions.

   B. Configure the system configuration register for the USART module clock.

   C. Set USART_MR[3:0] to LIN slave node mode.

   D. Configure the USART_LINMR register to set the operating mode of the slave node.

   E. Configure the USART_BRGR register to set the baud rate for LIN communication.

2. Transmitting data

   A. Wait for USART_CSR[14] to be set, indicating that the received identifier is correct.

   B. Configure USART_LINMR[15:8] to set the data length of the LIN frame.

   C. Configure USART_LINMR[1:0] to set the slave node operation mode as PUBLISH to send an acknowledgment, wait for USART_CSR[1] to be set, write the data to be transmitted into USART_THR, and wait for USART_CSR[1] to be set again, indicating the data transmission is complete before writing the next data, until all data is transmitted.

3. Receiving data

   A. Configure USART_LINMR[1:0] to set the slave node operation mode as SUBSCRIBE to receive an acknowledgment.

   B. Wait for USART_CSR[14] to be set, indicating that the received identifier is correct.

   C. Configure USART_LINMR[15:8] to set the data length of the LIN frame.

   D. Wait for USART_CSR[0] to be set, then read the data from USART_RHR.

   E. Repeat step D until all data has been read.

   F. Wait for USART_CSR[15] to be set, indicating the LIN bus transmission is complete.

   G. Set USART_CR[8] to 1 to reset the status bit.

32 Serial Peripheral Interface (SPI0 & SPI1)

32.1 Overview

SPI is widely used to provide economical board-level interface between different devices such as EEPROM, FLASH, micro controller, DAC, ADC, etc.

32.2 Main Features

- Full-duplex 4 wires or half-duplex 3 wires serial synchronous transmission and reception
- Master / slave mode
- Programmable clock polarity and phase
- Programmable bit rate
- Up to f_{sys} / 2 frequency in slave mode
- Transmission complete interrupt flag
- Write conflict error flag
- Error detection, protection and interrupt flags in master mode
- Build-in 8-byte FIFO
- DMA single word transmission
- Supporting 2 slave devices

32.3 Pin Description

Table 32-1: SPI0 & SPI1 Pin Description

Function Pin	Alternate Function Pin	Direction	Functional Description
SPI0_SCK	PA5, PA10, PB1, PB3	Input/output	Serial clock signal
SPI0_MISO	PA0, PA6, PB0, PB4	Input/output	Normal data signal
SPI0_MISO2	PA2, PB2	Input/output	Second data signal (special mode: supporting two external slaves simultaneously)
SPI0_MOSI	PA7, PB1, PC1, PB5	Input/output	Data signal
SPI0_SSN	PA1, PA4, PA8, PA15, PB6	Input/output	Normal chip select signal
SPI0_SSN2	PC0, PA3	Output	Second slave chip select signal
SPI1_SCK	PA10, PB1, PB3, PB6, PB10, PB13, PC7	Input/output	Serial clock signal
SPI1_MISO	PA9, PA11, PA13, PB4, PB7, PB14, PC2, PC8	Input/output	Normal data signal
SPI1_MISO2	PC4	Input/output	Second data signal (special mode: supporting two external slaves simultaneously)
SPI1_MOSI	PA12, PA14, PB1, PB5, PC1, PB15, PC3, PC9	Input/output	Data signal
SPI1_SSN	PA4, PA13, PA15, PB9, PB12, PC6	Input/output	Normal chip select signal
SPI1_SSN2	PC5	Output	Second slave chip select signal

32.4 Functional Description

To be compatible with different SPI peripherals, the timing of SPI serial clock can be produced in four clock/data relationships by setting the clock phase selection bit (SPICSx.CPHA) and the clock polarity selection bit (SPICSx.CPOL). To ensure the correct data transmission, the timing configuration of the master and slave devices must be consistent.

There is no serial clock output at the SCK pin of SPI when it is in slave mode or when the SPI system enable bit (SPICR.SPIEN) is 0.

32.4.1 Clock Phase CPHA = 0

When CPHA = 0, the SPI module samples data at the first transition edge of the serial clock, that is:

If CPOL = 1, the data is sampled on the falling edge of the serial clock.

If CPOL = 0, the data is sampled on the rising edge of the serial clock.

As shown in the figure below:

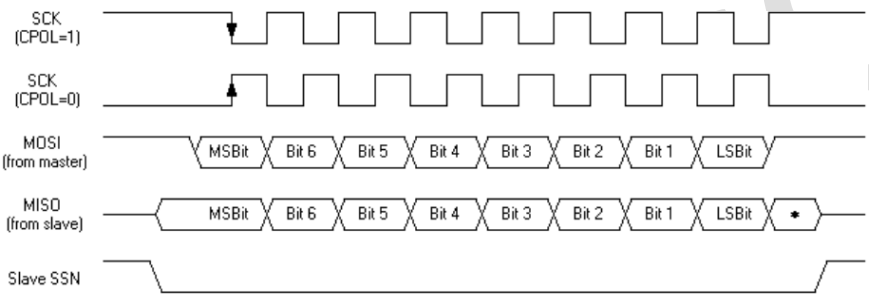


Figure 32-1: SPI Data / Clock Timing Diagram (CPHA = 0)

32.4.2 Clock Phase CPHA = 1

When CPHA = 1, the SPI module samples data at the second transition edge of the serial clock, that is:

If CPOL = 1, the data is sampled on the rising edge of the serial clock.

If CPOL = 0, the data is sampled on the falling edge of the serial clock.

As shown in the figure below:

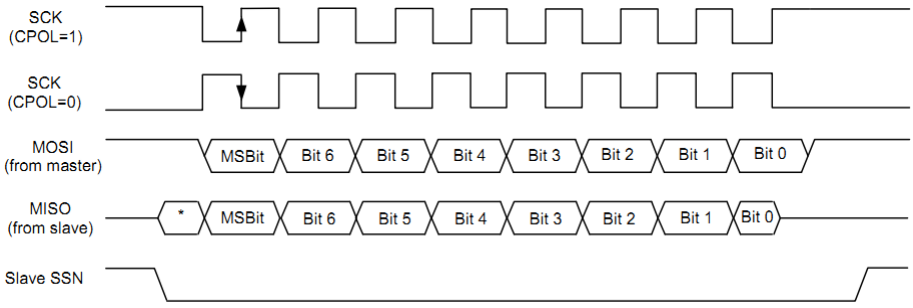


Figure 32-2: SPI Data / Clock Timing Diagram (CPHA = 1)

32.4.3 Slave SSN

If SPI is a slave device, then when CPHA = 0, the SSN pin must be pulled high after each byte of data is transferred so that it can be pulled low to initiate transmission of the next byte and write conflict errors can be avoided.

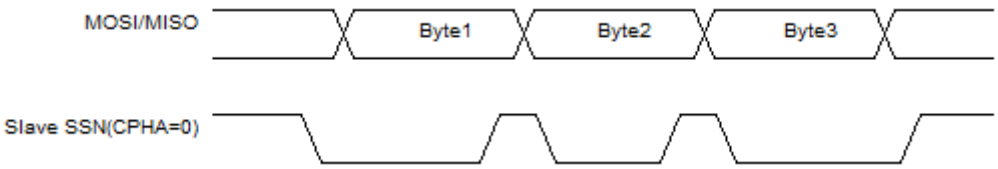


Figure 32-3: SPI SSN Timing Diagram (CPHA = 0)

When CPHA = 1, the SSN pin of the slave device remains low during continuous data transfers.

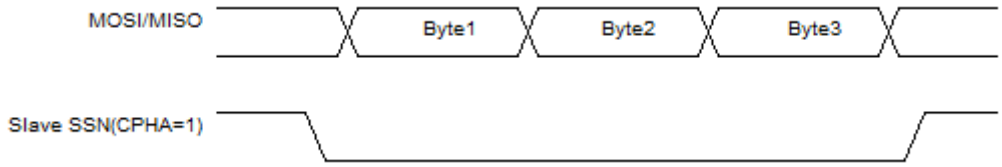


Figure 32-4: SPI SSN Timing Diagram (CPHA = 1)

32.4.4 Communications between One Master and One Slave

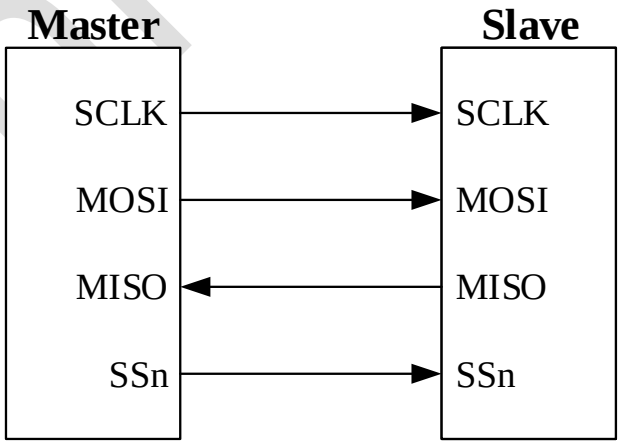


Figure 32-5: Full-duplex Communication

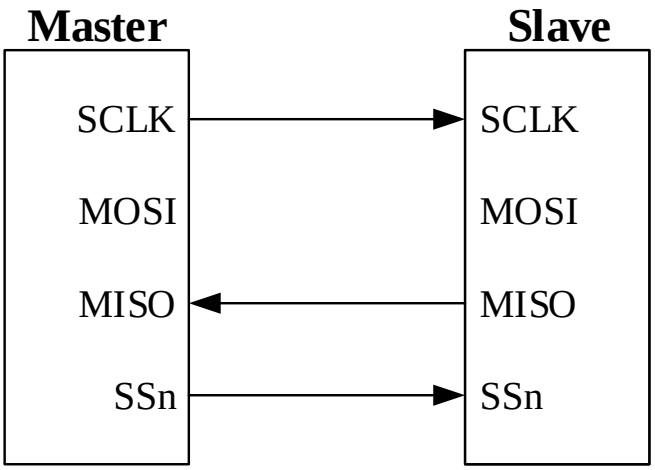


Figure 32-6: Simplex Communication (Master Receiving, Slave Transmitting)

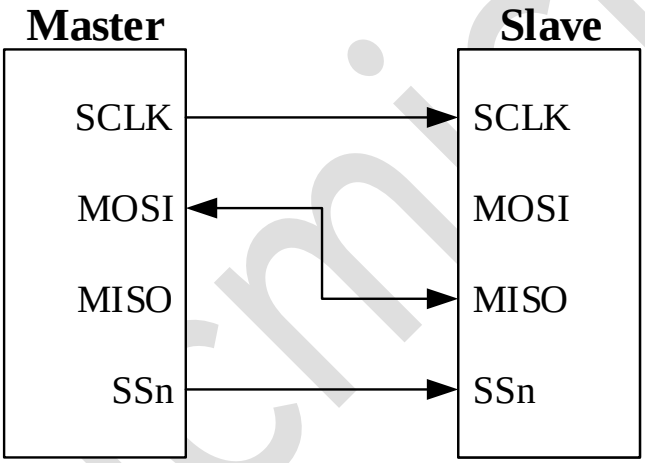


Figure 32-7: Half-duplex Communication

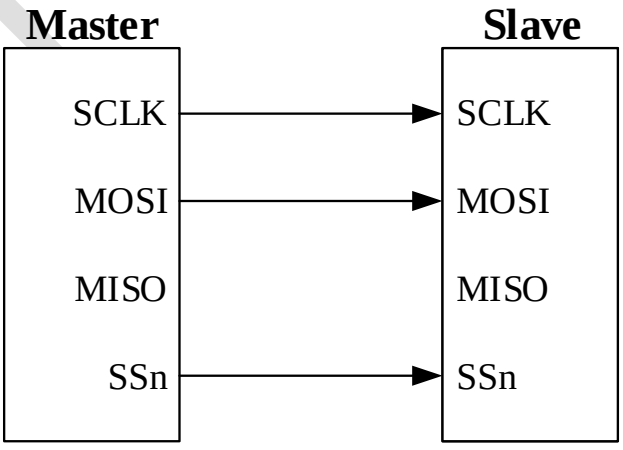


Figure 32-8: Simplex Communication (Master Transmitting, Slave Receiving)

Notes:

1. In half-duplex mode, the TXO bit of the SPI configuration register determines the data transmission direction. However, in half-duplex receive mode, it is necessary to write dummy data (any data) to the transmit buffer register for the master to generate the clock; the slave does not require this.
2. In simplex mode, this SPI module only supports “transmit-only mode” and does not support “receive-only mode.” In simplex receive mode, it is necessary to write dummy data to the transmit buffer register for the master to generate the clock; the slave does not require this.

32.5 Register Description

SPI0 register base address: 0x4700_0000

SPI1 register base address: 0x4700_1000

The registers are listed below:

Table 32-2: List of SPI0 & SP1 Registers

Offset Address	Name	Description
0x00	SPI_CR	SPI configuration register
0x04	SPI_CS0	SPI master mode control register 0
0x08	SPI_CS1	SPI master mode control register 1
0x14	SPI_OPCR	SPI process control register
0x18	SPI_IE	SPI interrupt enable register
0x1C	SPI_IF	SPI interrupt flag register
0x20	SPI_TXBUF	SPI transmit buffer register
0x24	SPI_RXBUF	SPI receive buffer register
0x28	SPI_DMARXLEV	SPI DMA receive setting register
0x2C	SPI_DMATXLEV	SPI DMA transmit setting register

32.5.1 SPI Configuration Register (SPI_CR)

Offset address: 0x00

Reset value: 0x0000_0A20

Bit	Name	Attribute	Reset Value	Description
31:15	RSV	-	-	Reserved
14	SSN_PD	R/W	0x0	Enable the software to pull down the SSN signal in slave mode: 0: Use the signal from the external SSN pin. 1: Software pulls down the SSN signal, ignoring the external SSN pin. Note: This register must be written in two steps: first switch to slave mode, then configure this bit, otherwise it may cause communication timing errors.
13	DMA_TX_EN	R/W	0x0	DMA TX enable: 0: DMA TX request disabled 1: DMA TX request enabled
12	DMA_RX_EN	R/W	0x0	DMA RX enable: 0: DMA RX request disabled 1: DMA RX request enabled
11	FLTEN	R/W	0x1	Slave input pin (SSN/SCK/MOSI) filter enable: 0: 4 ns filter enabled 1: no filter
10	SSN_M	R/W	0x0	SSN control mode selection in master mode: 0: The master pulls SSN low after each 8-bit transmission, and the low-level duration is controlled by the WAIT register. 1: The master pulls SSN high after each 8-bit transmission, and the high-level duration is controlled by the WAIT register.
9	TXO_AC	R/W	0x1	TXONLY hardware auto-clear enable: 0: TXONLY hardware auto-clear disabled 1: TXONLY hardware auto-clear is enabled. After TXO is enabled by software, it will be cleared by hardware upon completion of the

Bit	Name	Attribute	Reset Value	Description
				transmission. Note: If half-duplex mode is enabled, this bit shall be set to 0.
8	TXO	R/W	0x0	TXONLY control bit: 0: Disable transmit-only mode; indicating receive-only in half-duplex mode 1: Enable transmit-only mode; received data will not be stored in FIFO.
7	MSPA	R/W	0x0	Master adjusts the sampling position of MISO signal to compensate PCB routing delay in high-speed communication: 0: adjustment not required 1: sampling point delayed by half an SCK cycle
6	SSPA	R/W	0x0	Slave adjusts the transmit position of MISO signal: 0: adjustment not required 1: transmitted half an SCK cycle in advance
5	MM	R/W	0x1	Master/slave mode selection: 0: slave mode 1: master mode
4:3	WAIT	R/W	0x0	In master mode, after transmitting each byte (8 bits), add (WAIT + 1) SCK cycles of wait time before transmitting the next byte of data.
2	TRI_EN	R/W	0x0	SPI half-duplex mode (three-wire mode) enable: 0: half-duplex mode disabled 1: half-duplex mode enabled The data transmission direction is determined by the 8th bit TXO of this register.
1	SSN_EN	R/W	0x0	SSN software control enable in master mode: 0: SSN output is automatically controlled by hardware in master mode. 1: SSN output is controlled by software in master mode.
0	SPI_EN	R/W	0x0	SPI enable:

Bit	Name	Attribute	Reset Value	Description
				0: disable SPI to empty the transmit/receive buffer 1: enable SPI (Disable by turning off the clock.)

32.5.2 SPI Master Mode Control Register 0 (SPI_CS0)

Offset address: 0x04

Reset value: 0x0000 0008

Bit	Name	Attribute	Reset Value	Description
31:7	RSV	-	-	Reserved
6	SSN0	R/W	0x0	In SPI master mode, CS0 is used to control the selection of a slave device. If SSN_EN is set to 1, the software can control the SSN output level to select the slave device: 0: SSN outputs a high level, indicating that the slave device is not selected. 1: SSN outputs a low level, indicating that the slave device is selected. Note: Correspond to SPI0_NSS or SPI1_NSS.
5:3	BAUD0	R/W	0x1	In SPI master mode, CS0 corresponds to the baud rate configuration: 000: f_{PCLK} / 2 001: f_{PCLK} / 4 010: f_{PCLK} / 8 011: f_{PCLK} / 16 100: f_{PCLK} / 32 101: f_{PCLK} / 64 110: f_{PCLK} / 128 111: f_{PCLK} / 256
2	LSBF0	R/W	0x0	In SPI master mode, CS0 corresponds to the frame format: 0: MSB first 1: LSB first
1	CPOL0	R/W	0x0	In SPI master mode, CS0 corresponds to the clock polarity selection:

Bit	Name	Attribute	Reset Value	Description
				0: The serial clock remains low when idle. 1: The serial clock remains high when idle. Note: The value of this bit shall not be changed when SSN is low.
0	CPHA0	R/W	0x0	In SPI master mode, CS0 also corresponds to the clock phase selection: 0: Data is first sampled on the first clock edge. 1: Data is first sampled on the second clock edge.

Notes:

1. The [5:0] bits cannot be modified while communication is ongoing.
2. In the slave mode, it is not necessary to configure SSN0 and SSN1, and this register can also be used to configure the clock polarity and phase, and select the bit transmission order of characters.

32.5.3 SPI Master Mode Control Register 1 (SPI_CS1)

Offset address: 0x08

Reset value: 0x0000 0008

Bit	Name	Attribute	Reset Value	Functional Description
31:7	RSV	-	-	Reserved
6	SSN1	R/W	0x0	In SPI master mode, CS1 is used to control the selection of a slave device. If SSN_EN is set to 1, the software can control the SSN output level to select the slave device: 0: SSN outputs a high level, indicating that the slave device is not selected. 1: SSN outputs a low level, indicating that the slave device is selected. Note: Correspond to SPI0_NSS2 or SPI1_NSS2.
5:3	BAUD1	R/W	0x1	In SPI master mode, CS1 corresponds to the baud rate configuration: 000: f_{PCLK} / 2

Bit	Name	Attribute	Reset Value	Functional Description
				001: f_{PCLK} / 4 010: f_{PCLK} / 8 011: f_{PCLK} / 16 100: f_{PCLK} / 32 101: f_{PCLK} / 64 110: f_{PCLK} / 128 111: f_{PCLK} / 256
2	LSBF1	R/W	0x0	In SPI master mode, CS1 corresponds to the frame format: 0: MSB first 1: LSB first
1	CPOL1	R/W	0x0	In SPI master mode, CS1 corresponds to the clock polarity selection: 0: The serial clock remains low when idle. 1: The serial clock remains high when idle. Note: The value of this bit shall not be changed when SSN is low.
0	CPHA1	R/W	0x0	In SPI master mode, CS1 also corresponds to the clock phase selection: 0: The first clock transition is the first data capture edge. 1: The second clock transition is the first data capture edge.

Note: The [5:0] bits cannot be modified while communication is ongoing.

32.5.4 SPI Process Control Register (SPI_OPCR)

Offset address: 0x14

Reset value: 0x0000 0010

Bit	Name	Attribute	Reset Value	Functional Description
31:5	RSV	-	-	Reserved
4	SSN_STAT	R	0x1	Filtered SSN pin level status
3	TXBFC	R/W1C	0x0	Transmit buffer clear: Writing 1 by software clears the transmit buffer; writing 0 is invalid, and reading yields 0.

Bit	Name	Attribute	Reset Value	Functional Description
2	RXBFC	R/W1C	0x0	Receive buffer clear: Writing 1 by software clears the receive buffer; writing 0 is invalid, and reading yields 0.
1	MERRC	R/W1C	0x0	Master error clear: Writing 1 by software clears the SPIIF.MERR register; writing 0 is invalid, and reading yields 0.
0	SERRC	R/W1C	0x0	Slave error clear: Writing 1 by software clears the SPIIF.SERR register; writing 0 is invalid, and reading yields 0.

32.5.5 SPI Interrupt Control Register (SPI_IE)

Offset address: 0x18

Reset value: 0x0000 7000

Bit	Name	Attribute	Reset Value	Functional Description
31:16	RSV	-	-	Reserved
15	RX_BIT_IE	R/W	0x0	Receive bit interrupt enable: After receiving the RX_BIT_NUM bits and before writing the data into RX FIFO, the RX_BIT_IF interrupt flag bit will be set to 1. 0: receive bit interrupt disabled 1: receive bit interrupt enabled
14:12	RX_BIT_NUM	R/W	0x7	Number of receive bits that trigger the interrupt: If the receive bit interrupt is enabled, after receiving the RX_BIT_NUM bits and before writing the data into RX FIFO, the RX_BIT_IF interrupt flag bit will be set to 1. Note: This number cannot be set to 0. The default setting is 7, which means that the RX_BIT_IF interrupt will be set to 1 when the last bit of the 8-bit data is received.
11	RSV	-	-	Reserved

Bit	Name	Attribute	Reset Value	Functional Description
10	SSN_NEG_IE	R/W	0x0	Falling edge interrupt enable for filtered SSN in slave mode: 0: disabled 1: enabled
9	SSN_POS_IE	R/W	0x0	Rising edge interrupt enable for filtered SSN in slave mode: 0: disabled 1: enabled
8	RXF_IE	R/W	0x0	RX FIFO full interrupt enable: 0: disabled 1: enabled
7	TXNF_IE	R/W	0x0	TX FIFO not-full interrupt enable: 0: disabled 1: enabled
6	MERR_IE	R/W	0x0	Master error interrupt enable: 0: disabled 1: enabled
5	SERR_IE	R/W	0x0	Slave error interrupt enable: 0: disabled 1: enabled
4	RXCOL_IE	R/W	0x0	Receive buffer overflow interrupt enable: 0: disabled 1: enabled
3	TXCOL_IE	R/W	0x0	Transmit buffer overflow interrupt enable: 0: disabled 1: enabled
2	IDLE_IE	R/W	0x0	Idle interrupt enable: 0: disabled 1: enabled
1	TXBE_IE	R/W	0x0	Transmit buffer empty interrupt enable: 0: disabled 1: enabled
0	RXBF_IE	R/W	0x0	Receive buffer non-empty interrupt enable: 0: disabled 1: enabled

32.5.6 SPI Interrupt Flag Register (SPI_IF)

Offset address: 0x1C

Reset value: 0x0000 0086

Bit	Name	Attribute	Reset Value	Functional Description
31:24	RSV	-	-	Reserved
23:20	RXFIFO_LEVEL	R	0x0	Number of data stored in RX FIFO
19:16	TXFIFO_LEVEL	R	0x0	Number of data stored in TX FIFO
15	RX_BIT_IF	R	0x0	Receive bit interrupt flag: After receiving the RX_BIT_NUM bits and before writing the data into RX FIFO, the interrupt flag bit will be set to 1. 0: RX_BIT_NUM bits not received 1: RX_BIT_NUM bits received Note: Reading this register bit will clear it.
14:11	RSV	-	-	Reserved
10	SSN_NEG_IF	R/W1C	0x0	Filtered SSN falling edge interrupt flag in slave mode: 0: no 1→0 transition occurred in filtered SSN in slave mode 0: 1→0 transition occurred in filtered SSN in slave mode
9	SSN_POS_IF	R/W1C	0x0	Filtered SSN rising edge interrupt flag in slave mode: 0: no 0→1 transition occurred in filtered SSN in slave mode 1: 0→1 transition occurred in filtered SSN in slave mode
8	RXF	R	0x0	RX FIFO full: 0: RX FIFO not full 1: RX FIFO full
7	TXNF	R	0x1	TX FIFO not full: 0: TX FIFO full 1: TX FIFO not full
6	MERR	R	0x0	Transmission error flag in master mode:

Bit	Name	Attribute	Reset Value	Functional Description
				0: normal transmission 1: transmission error occurred
5	SERR	R	0x0	Transmission error flag in slave mode: 0: normal transmission 1: transmission error occurred
4	RXCOL	R/W1C	0x0	Receive buffer overflow: 0: no overflow 1: overflow
3	TXCOL	R/W1C	0x0	Transmit buffer overflow: 0: no overflow 1: overflow
2	IDLE	R	0x1	SPI idle state level flag: For the master, idle means no data needs to be transmitted. For the slave, idle means SSN is in a high state. 0: transmission in progress 1: idle
1	TXBE	R	0x1	Transmit buffer empty flag bit: 0: transmit buffer not empty 1: transmit buffer empty, software writes TXBUF to clear this bit.
0	RXBF	R	0x0	Receive buffer non-empty flag bit: 0: receive buffer empty 1: receive buffer not empty

The transmission error flag will be set when the chip select (SSN) is pulled high and the serial clock (SCK) toggles on the rising edge, or when serial data (MOSI or MISO) is being transmitted and the chip select (SSN) toggles on the rising edge. The transmission error flag will be cleared when the SPI is reset or disabled, will be cleared by the corresponding transmission error flag clear bit in the process control register, and will also be cleared when switching between master and slave modes.

32.5.7 SPI Transmit Buffer Register (SPI_TXBUF)

Offset address: 0x20

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Functional Description
31:8	RSV	-	-	Reserved
7:0	TXBUF	W	-	SPI transmit buffer, transmit FIFO entry: A transmit FIFO contains 8 bytes in total. Write this register to write the data to be transmitted into the FIFO.

32.5.8 SPI Receive Buffer Register (SPI_RXBUF)

Offset address: 0x24

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Functional Description
31:8	RSV	-	-	Reserved
7:0	RXBUF	R	0x0	SPI receive buffer, receive FIFO entry: A receive FIFO contains 8 bytes in total. Read this register to read the received data from the FIFO.

32.5.9 SPI DMA Receive Setting Register (SPI_DMARXLEV)

Offset address: 0x28

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Functional Description
31:3	RSV	-	-	Reserved
2:0	DMA_RX_LEV	R/W	0x0	Setting of SPI RX FIFO DMA request: When the number of data in RX FIFO is greater than the set value of this register, a DMA RX request is generated.

32.5.10 SPI DMA Transmit Setting Register (SPI_DMATXLEV)

Offset address: 0x2C

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Functional Description
31:4	RSV	-	-	Reserved
3:0	DMA_TX_LEV	R/W	0x0	Setting of SPI TX FIFO DMA request: When the number of data in TX FIFO is smaller than the set value of this register, a DMA TX request is generated. Note: The maximum setting of this register is 8.

32.6 Operation Procedure

32.6.1 Master Mode

32.6.2 Initialization Process

1. Configure to enable the SPI module clock and the reset.
2. Configure the corresponding GPIO pins to be alternated as SPI_SCK, SPI_MISO, SPI_MOSI and SPI_CLK.
3. Configure SPI_CR[5] to set the master/slave mode.
4. Configure SPI_CR[10] to set the SSN control mode.
5. Configure SPI_CR[1] to set whether SSN output is controlled by software or hardware.
6. Configure SPI_CSx[6] to set whether SSN outputs high level or low level.
7. Configure SPI_CSx[2] to set whether MSB first or LSB first.
8. Configure SPI_CSx[0] to set whether sampling at the first clock transition or the second clock transition.

9. Configure SPI_CSx[1] to set whether the serial clock stops at high level or low level.
10. Configure SPI_CSx.BAUDx[2:0] to set the serial clock baud rate (no need to set if it is in slave mode where the serial clock rate is determined by the master device). Configure SPI_IE to enable the corresponding interrupt if required.
11. Configure SPI_CR[0] to enable SPI.

32.6.3 Transmission Process

- Master transmission process:

  Configure SPI_CSx[6] to pull the SSN pin low to start the transmission, configure SPI_CR[8] to be high, write the data into SPI_TXBUF register, wait for SPI_IF[2] to be set and sent, configure SPI_CR[8] to be low, and pull the SSN high after the transmission is completed.

- Slave transmission process:

  Configure SPI_CR[8] to be high, write the data into SPI_TXBUF register, wait for SPI_IF[2] to be set and sent, and configure SPI_CR[8] to be low.

32.6.4 Reception Process

- Master reception process:

  Configure SPI_CSx[6] to pull the SSN pin low to start transmission, write the data into SPI_TXBUF register, wait for the SPI_IF[0] to be set, read the data in SPI_RXBUF register to complete data reception, and pull the SSN high after the transmission is completed.

- Slave reception process:

  Wait for SPI_IF[0] to be set, and read the data in SPI_RXBUF register to complete data reception.

33 Enhanced Serial Peripheral Interface (SPI2)

33.1 Overview

SPI is widely used to provide economical board-level interface between different devices such as EEPROM, FLASH, micro controller, DAC, ADC, etc.

33.2 Main Features

- Configurable master or slave mode
- Supporting serial full duplex, half duplex, simplex transmitting and simplex receiving in both master and slave modes
- Configurable 8-bit SPI clock frequency control register: SCK frequency up to $f_{PCLK}/2$ in master mode while up to $f_{PCLK}/4$ in slave mode
- Programmable SCK polarity and phase
- Supporting SPI Motorola mode or TI mode
- Programmable data order with MSB-first or LSB-first shifting
- Configurable character length from 4 bits to 32 bits, with the default being 8 bits
- Built-in TX FIFO and RX FIFO with 8 x 32-bit depth
- Programmable software- or hardware-controlled chip select
- DMA operation

33.3 System Block Diagram

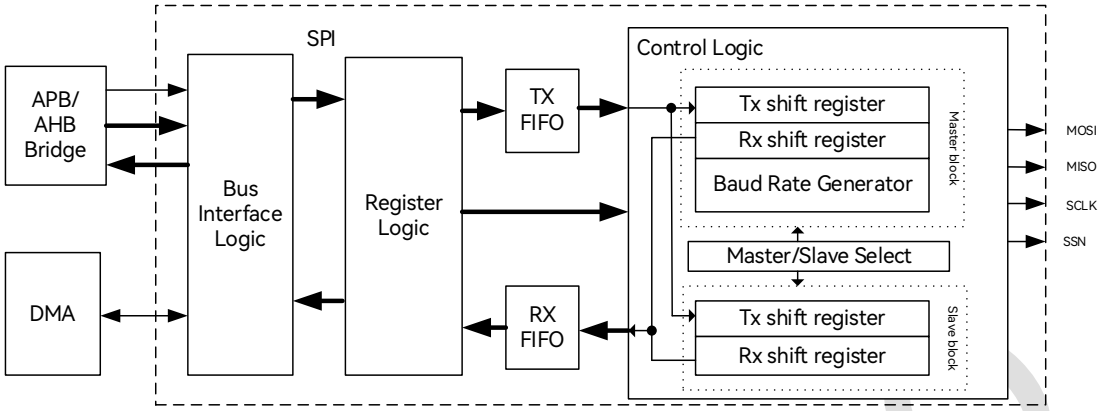


Figure 33-1: SPI Module Block Diagram

As shown in Figure 33-1, the SPI module consists of five blocks: APB bus interface logic, configuration register logic, TX FIFO, RX FIFO, and SPI control logic. The SPI control logic is divided into slave and master modules, where the master module includes transmit shift logic, receive shift logic, and SPI clock frequency generator logic.

33.4 Pin Description

Table 33-1: SPI2 Pin Description

Function Pin	Alternate Function Pin	Direction	Functional Description
SPI2_SCK	PA9, PB3, PB7, PB13, PC10, PD8, PD9, PD10	Input/output	Serial clock signal (SCLK)
SPI2_MISO	PA12, PA13, PB4, PB8, PB14, PC11, PD8, PD9, PD10	Input/output	Data signal
SPI2_MOSI	PA1, PA10, PA14, PB2, PB5, PB9, PB15, PC1, PC12, PD8, PD9, PD10	Input/output	Data signal
SPI2_SSN	PA4, PA11, PA15, PB6, PB12, PD8, PD9, PD10	Input/output	Chip select signal

33.5 Functional Description

33.5.1 SPI Clock Frequency Control Register

The dedicated 16-bit "SPI clock frequency control register (SPBRG)" is applicable only in SPI master mode, and it controls the frequency of data transmission and reception.

The SPI clock (SCLK) is generated by dividing the APBx clock (PCLK), and the value of SPBRG is the division factor.

$$f_{SCLK} = f_{PCLK} / SPBRG$$

Where the value in the SPBRG register defaults to 2 and shall range from 2 to 255. In master mode, the maximum SCLK frequency is up to $f_{PCLK}/2$ in master mode while up to $f_{PCLK}/4$ in slave mode.

33.5.2 TX FIFO

The TX FIFO has a depth of 8 words (one word is 32 bits). When the CPU or DMA writes data to the "transmit data register (TXREG)," the data is written into the TX FIFO, and the transmission begins. During the transmission, the TX FIFO shifts data to the "transmit shift register (TSR)."

When the "transmit shift register (TSR)" is empty, the TX FIFO outputs data to the TSR. The TSR then shifts the data to the transmit serial port (TX).

When the TX FIFO is not full and data transmission is required, the "transmit burst request (TXBREQ)" signal is asserted. This signal is sent to the DMA to request it to transfer data to the TX FIFO.

During transmission, when the TX FIFO is empty, the "transmitter empty interrupt flag (TXEPT_INTF)" is generated. When the TX FIFO receives enough data (depending on "TXTLF"), the "transmitter data available interrupt flag (TX_INTF)" is generated. When the SPI slave

attempts to send a new character while idle, the “slave transmitter underrun interrupt flag (UNDERRUN_INTF)” is generated.

33.5.3 RX FIFO

The RX FIFO has a depth of 8 words (one word is 32 bits). The CPU or DMA receives data from the RX FIFO. The “receive shift register (RSR)” is used to receive data from the receive serial port (RX). Incoming data is placed into the RSR. Once all bits have been shifted in, the data is transferred to the RX FIFO. When the data is transferred from the “shift register” to the “data register,” the “receive register available bit (RXAVL)” in the “current status register (CSTAT)” is set. The “RX FIFO data available interrupt flag (RX_INTF)” can also be set in the “interrupt status register (INTSTAT)”. The CPU then receives data from the “receive data register (RXREG)”. If a “receive burst request (RXBREQ)” is received, the DMA will read data from RXREG.

When the master or slave attempts to write data to a full RX FIFO, the most recent data will not be written, and the “receiver overflow error interrupt flag (RXOERR_INTF)” will be generated.

33.5.4 SPI Control Logic

33.5.4.1 Overview

SPI allows synchronous transmission and reception of 4 to 32 bits of data. It can be configured as either a slave or a master in a single-master environment. The clock polarity (CKPL) and clock phase (CKPH) settings can utilize all four SPI timing modes. The bit order can also be set to LSB first or MSB first.

The transmitter and receiver use the same clock. Data is output only during the rising or falling edge of the generated serial clock (SCLK) and is latched on the opposite edge of SCLK. The

SCLK polarity and phase of the SPI master device and slave device shall be the same.

SPI is used for data exchange, with the data to be transmitted temporarily stored in the TX FIFO and the received data stored in the RX FIFO. If the SPI module is disabled, all data in these FIFOs will be lost; therefore, it is necessary to read the data from the RX FIFO through the "receive data register (RXREG)" before disabling the SPI module at the end of transmission.

The standard SPI protocol includes two data lines: one clock line and one chip select line, as described below:

- Master out slave in (MOSI or TX or SDA): This data line provides output data transferred from the master to the slave input.
- Master in slave out (MISO or RX): This data line provides the output data from the slave to the master input. During any specific transmission, only one slave device can transmit data.
- Serial clock (SCLK or SCL): This clock line is driven by the master and regulates the data flow. The master can transmit data at various SPI clock frequencies. The SCLK line cycles once for each transmitted bit.
- Chip select (CS or SSN): This is the slave select input signal from the master, which is active low.

33.5.4.2 Master Mode

The SPI module only supports single-master environment. The sole SPI master device can initiate transmission and reception.

The core of the transmitter is the "transmit shift register". The software loads the data to be transmitted into the "transmit buffer register (SPI_TXREG)" via the APB bus or DMA bus, and

TXREG immediately transmits the data to the TX FIFO. Once the transmission starts, TSR retrieves data from the TX FIFO; if TSR is empty, the data byte from the TX FIFO is immediately transferred to TSR. TSR is not mapped to data memory, so it cannot be accessed by the user.

Once there are enough vacancies in TX FIFO, the “TX FIFO vacancy available interrupt flag (TX_INTF)” is set to 1. When all the data in TX FIFO and the “transmit shift register” has been sent, the “transmitter empty interrupt flag (txept_intf)” is set to 1. It should be noted that whether these interrupt flags are set to 1 is independent of the state of the “interrupt enable register (SPI_TXIEN)”. To read the enabled interrupt status, the “masked interrupt status register (SPI_MINTSTAT)” can be read, and writing 1 to the “interrupt clear register (SPI_INTCLR)” will clear both masked and unmasked interrupt flags.

The master, controlling the SPI clock (SCLK), can initiate data transmission at any time. In master mode, once data is written to the TXREG register, the transmission and reception of data begin. If only receiving data is intended, the transmit function can be disabled, and the receive shift register (RSR) will still shift the signal on the MISO pin at the SCLK frequency. After receiving each SPI character, RSR will save the SPI character into RX FIFO, and the “interrupt status register (SPI_INTSTAT)” and “current status register (SPI_CSTAT)” will change accordingly.

The SCLK clock is only valid during data transmission.

33.5.4.3 Slave Mode

The SPI slave receives signals from the SPI master. Data is transmitted and received when an external clock pulse appears on the SCLK pin. This external clock must meet the minimum hold time requirements for high and low levels as specified in the electrical specifications.

To prepare for data transmission, the SPI slave must write the data to be transmitted into the

TXREG register before the transmission starts. The TXREG register writes the data into TX FIFO, which is then transmitted to the “transmit shift register (TSR)”. Once the master issues SCLK, the slave shifts the data to be transmitted out from the MISO pin while shifting the received data into the MOSI pin.

33.5.5 Data Concatenation

By setting the register SPI_CCTL[7:6], the data concatenation feature can be used to concatenate short characters into a word in the TX FIFO and RX FIFO. The concatenation format is as follows:

1. If SPI_CCTL[12:8] ≤ 7, then every four SPI characters will be concatenated into one word (32 bits).

   Example 1: SPI_CCTL[12:8] = 5, each SPI character is 6 bits long, the concatenation format is as follows:

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

Figure 33-1: Data Concatenation Example 1 (SPI_CCTL[12:8] = 5)

2. If 8 ≤ SPI_CCTL[12:8] ≤ 15, then every two SPI characters will be concatenated into one word.

   Example 2: SPI_CCTL[12:8] = 13, each SPI character is 14 bits long, the concatenation format is as follows:

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

Figure 33-2: Data Concatenation Example 2 (SPI_CCTL[12:8] = 13)

3. If 16 ≤ SPI_CCTL[12:8] ≤ 31, then every one SPI character will be concatenated into one word.

   Example 3: SPI_CCTL[12:8] = 28, each SPI character is 29 bits long, the concatenation

format is as follows:

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

Figure 33-3: Data Concatenation Example 3 (SPI_CCTL[12:8] = 28)

33.5.6 Interface Timing

33.5.6.1 Motorola Timing Mode

The Motorola timing mode refers to the conventional timing mode.

Table 33-2: Motorola Timing Mode

Mode	Clock Polarity (CKPL)	Clock Phase (CKPH)
0	0: Clock is low when idle.	0: Data sampled on the first edge (rising edge) of the clock
1	0: Clock is low when idle.	1: Data sampled on the second edge (falling edge) of the clock
2	1: Clock is high when idle.	0: Data sampled on the first edge (falling edge) of the clock
3	1: Clock is high when idle.	1: Data sampled on the second edge (rising edge) of the clock

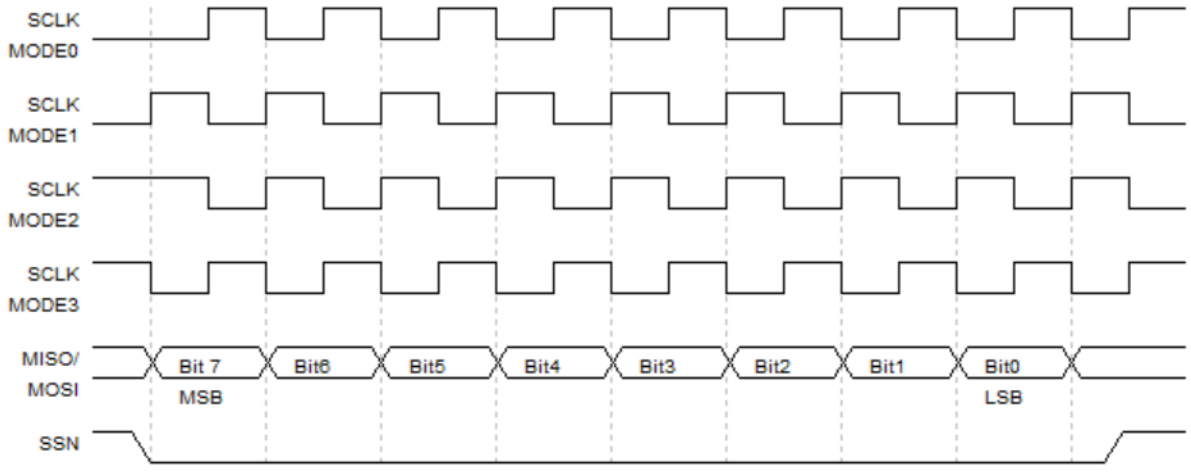


Figure 33-4: Transmission Timing Diagram of SPI in Different Modes in Motorola Mode

33.5.6.2 TI Timing Mode

In TI timing mode, CKPL is the same as in the Motorola mode, but CKPH is the opposite. Furthermore, as shown in Figure 33-5, before transmitting each piece of data, the chip select (SSN, as shown in Figure 33-5) signal shall be pulled high for one serial clock cycle. This module only supports single transfer format and does not support continuous transfer format.

Table 33-2: TI Timing Mode

Mode	Clock Polarity (CKPL)	Clock Phase (CKPH)
0	0: clock is low when idle	1: data sampled on the second edge (falling edge) of the clock
1	0: clock is low when idle	0: data sampled on the first edge (rising edge) of the clock
2	1: clock is high when idle	1: data sampled on the second edge (rising edge) of the clock
3	1: clock is high when idle	0: data sampled on the first edge (falling edge) of the clock

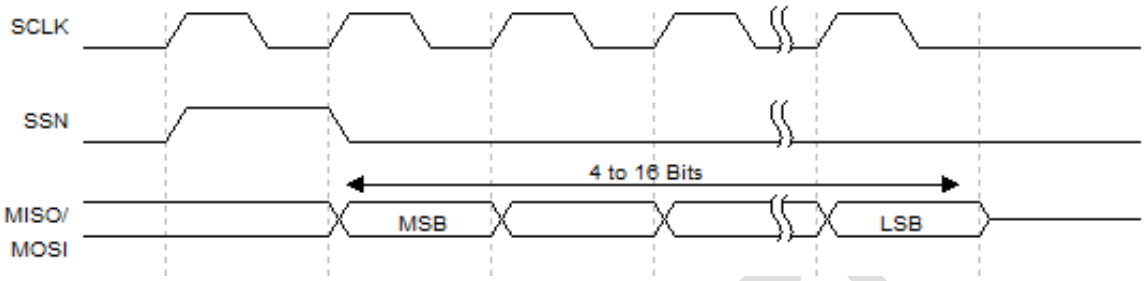


Figure 33-5: Transmission Timing Diagram of SPI in MODE2 in TI Mode

Note: SPI_CLK refers to the serial clock (SCLK). SPI_CS_N refers to the chip select (SSN). SPIDI/SPIDO refer to the serial data MISO and MOSI. This SPI module supports SPI character lengths of 1 to 32 bits.

Application note: The master and slave must maintain consistent timing modes, clock polarities and clock phases.

33.6 Register Description

SPI2 register base address: 0x4700_2000

The registers are listed below:

Table 33-3: List of SPI2 Registers

Offset Address	Name	Description
0x00	SPI2_TXREG	Transmit data register
0x04	SPI2_RXREG	Receive data register
0x08	SPI2_CSTAT	Current status register
0x0C	SPI2_INTSTAT	Interrupt status register
0x10	SPI2_MINTSTAT	Masked interrupt status register
0x14	SPI2_INTEN	Interrupt enable register
0x18	SPI2_INTCLR	Interrupt clear register

Offset Address	Name	Description
0x1C	SPI2_GCTL	Global control register
0x20	SPI2_CCTL	General control register
0x24	SPI2_SPBRG	SPI clock frequency control register
0x28	SPI2_RXDNR	Receive data number register
0x2C	SPI2_TXDNR	Transmit data number register
0x30	SPI2_SCSR	Slave chip select register

33.6.1 Transmit Data Register (SPI2_TXREG)

Offset address: 0x00

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:0	TXREG	R/W	0x0	This register stores the latest data written into the TX FIFO. The number of valid data bits depends on the value of SPI2_CCTL[12:8] and data concatenation.

33.6.2 Receive Data Register (SPI2_RXREG)

Offset address: 0x04

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:0	RXREG	R	0x0	This register stores the last data shifted out of the RX FIFO. The number of valid data bits depends on the value of SPI2_CCTL[12:8] and data concatenation. This register is read-only.

33.6.3 Current Status Register (SPI2_CSTAT)

Offset address: 0x08

Reset value: 0x0000 0011

Bit	Name	Attribute	Reset Value	Description
31:5	RSV	-	-	Reserved
4	CSSTAT	R	0x1	Chip select status register (useful as SPI slave):

Bit	Name	Attribute	Reset Value	Description
	US			1: idle state (CS signal is high) 0: selected state (CS signal is low)
3	RXAVL_4BYTE	R	0x0	RX FIFO has a flag with 4 words of available data. This bit is set to 1 when the RX FIFO has received at least 4 words of available data. 0: less than 4 words of data in RX FIFO 1: 4 words or more data in RX FIFO
2	TXFULL	R	0x0	TX FIFO full status flag: 0: TX FIFO not full 1: TX FIFO full
1	RXAVL	R	0x0	Available data received flag: This bit is set to 1 when the RX FIFO has received a complete word of data. 0: RX FIFO empty 1: RX FIFO has received a complete word of data.
0	TXEPT	R	0x1	Transmitter empty status flag: 0: transmitter not empty 1: Both TX FIFO and transmit shift register are empty.

33.6.4 Interrupt Status Register (SPI2_INTSTAT)

Offset address: 0x0C

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:10	RSV	-	-	Reserved
9	CSPOS_INTF	R	0x0	Chip select rising edge interrupt (SPI as slave) 0: no rising edge transition detected 1: rising edge transition detected on CS signal
8	CSNEG_INTF	R	0x0	Chip select falling edge interrupt (SPI as slave) 0: no falling edge transition detected 1: falling edge transition detected on CS signal
7	TXMATCH_IN	R	0x0	Transmission complete interrupt flag:

Bit	Name	Attribute	Reset Value	Description
	TF			When the amount of the actually-transmitted data matches the amount of data to be transmitted in the SPI2_TXDNR register, the transmitting process is completed and an interrupt is generated. This interrupt can occur in master single transmission mode, master duplex mode, or slave single transmission mode. 0: no transmission complete interrupt generated 1: transmission complete interrupt generated
6	TXEPT_INTF	R	0x0	Transmitter empty interrupt flag: An interrupt is generated when the data in the TX FIFO and the transmit shift register is completely sent out, or when the transmitter function is disabled, causing the TX FIFO and TSR data to be cleared. 0: no transmitter empty interrupt generated 1: transmitter empty interrupt generated
5	RXFIFO_FULL_INTF	R	0x0	RX FIFO full interrupt flag: An interrupt is generated when the RX FIFO is full. 0: no RX FIFO full interrupt generated 1: RX FIFO full Interrupt generated
4	RXMATCH_INTF	R	0x0	Reception complete interrupt flag: When the amount of the actually-received data matches the amount of data to be received in the SPI2_RXDNR register, the receiving process is completed and an interrupt is generated. This interrupt can occur in master single reception mode, slave single reception mode, or slave duplex mode. 0: no reception complete interrupt generated 1: reception complete interrupt generated
3	RXOERR_INTF	R	0x0	Receiver overflow error interrupt flag: An overflow error occurs when the master or

Bit	Name	Attribute	Reset Value	Description
				slave attempts to write data into a full RX FIFO, causing the latest data to be lost. 0: no receiver overflow error interrupt generated 1: receiver overflow error interrupt generated
2	UNDERRUN_INTF	R	0x0	Slave transmitter underrun interrupt flag: An underrun error occurs when the SPI slave attempts to send a new character while in an idle state. 0: no slave transmitter underrun error interrupt generated 1: slave transmitter underrun error interrupt generated
1	RX_INTF	R	0x0	RX FIFO data available interrupt flag: This bit is set when reception is enabled and the RX FIFO has enough data (depending on “rxtlf”). 0: no RX FIFO data available interrupt generated 1: RX FIFO data available interrupt generated
0	TX_INTF	R	0x0	TX FIFO vacancy available interrupt flag: This bit is set when transmission is enabled and the TX FIFO has enough vacancies (depending on “txtlf”). 0: no TX FIFO vacancy available interrupt generated 1: TX FIFO vacancy available interrupt generated

Note: Setting a bit in the interrupt enable register (SPI2_INTEN) to 0 can prevent the corresponding interrupt signal from being generated, but it cannot prevent the corresponding flag in the register from being set to 1 when the condition is met. Writing 1 to a bit in the interrupt clear register (SPI2_INTCLR) will clear the corresponding bit in the register.

33.6.5 Masked Interrupt Status Register (SPI2_MINTSTAT)

Offset address: 0x10

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:10	RSV	-	-	Reserved
9	CSPOS_MINTF	R	0x0	Chip select rising edge interrupt (SPI as slave): 0: no rising edge transition detected 1: rising edge transition detected on CS signal
8	CSNEG_MINTF	R	0x0	Chip select falling edge interrupt (SPI as slave): 0: no falling edge transition detected 1: falling edge transition detected on CS signal
7	TXMATCH_MINTF	R	0x0	Transmission complete interrupt flag: When the amount of data actually transmitted matches the amount of data to be transmitted in the SPI2_TXDNR register, the transmitting process is completed and an interrupt is generated. This interrupt can occur in master single transmission mode, master duplex mode, or slave single transmission mode. 0: no transmission complete interrupt generated 1: transmission complete interrupt generated
6	TXEPT_MINTF	R	0x0	Transmitter empty interrupt flag: An interrupt is generated when both TX FIFO and the transmit shift register are empty. 0: no transmitter empty interrupt generated 1: transmitter empty interrupt generated
5	RXFIFO_FULL_	R	0x0	RX FIFO full interrupt flag:

Bit	Name	Attribute	Reset Value	Description
	MINTF			An interrupt is generated when the RX FIFO is full. 1: RX FIFO full Interrupt generated 0: no RX FIFO full interrupt generated
4	RXMATCH_MINTF	R	0x0	Reception complete interrupt flag: When the amount of the actually-received data matches the amount of data to be received in the RXDNR register, the receiving process is completed and an interrupt is generated. This interrupt can occur in master single reception mode, slave single reception mode, or slave duplex mode. 0: no reception complete interrupt generated 1: reception complete interrupt generated
3	RXOERR_MINTF	R	0x0	Receiver overflow error interrupt flag: An overflow error occurs when the master or slave attempts to write data into a full RX FIFO, causing the latest data to be lost. 0: no receiver overflow error interrupt generated 1: receiver overflow error interrupt generated
2	UNDERRUN_MINTF	R	0x0	Slave transmitter underrun interrupt flag: An underrun error occurs when the SPI slave attempts to send a new character while in an idle state. 0: no slave transmitter underrun error interrupt generated 1: slave transmitter underrun error interrupt generated
1	RX_MINTF	R	0x0	Receiver data available interrupt flag: This bit is set to 1 when the RX FIFO has received sufficient data (depending on SPI2_GCTL[19:16]).

Bit	Name	Attribute	Reset Value	Description
				0: no receiver data available interrupt generated 1: receiver data available interrupt generated
0	TX_MINTF	R	0x0	TX FIFO vacancy available interrupt flag: This bit is set to 1 when there is sufficient vacancies in the TX FIFO (depending on SPI2_GCTL[23:20]). 0: no TX FIFO vacancy available interrupt generated 1: TX FIFO vacancy available interrupt generated

Note: This register is a shadow register, which maps the result of the logical AND interrupt enable register (SPI2_INTEN) of the interrupt status register (SPI2_INTSTAT). If an interrupt is not enabled, the corresponding interrupt flag in this register will not be set.

33.6.6 Interrupt Enable Register (SPI2_INTEN)

Offset address: 0x14

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:10	RSV	-	-	Reserved
9	CSPOS_IEN	R/W	0x0	Chip select rising edge interrupt (SPI as slave) enable: 0: disabled 1: enabled
8	CSNEG_IEN	R/W	0x0	Chip select falling edge interrupt (SPI as slave) enable: 0: disabled 1: enabled
7	TXMATCHEN	R/W	0x0	Transmission complete interrupt enable: 0: disabled

Bit	Name	Attribute	Reset Value	Description
				1: enabled
6	TXEPT_IEN	R/W	0x0	Transmitter empty interrupt enable: 0: disabled 1: enabled
5	RXFIFO_FULL_IEN	R/W	0x0	RX FIFO full interrupt enable: 0: disabled 1: enabled
4	RXMATCHEN	R/W	0x0	Reception complete interrupt enable: 0: disabled 1: enabled
3	RXOERREN	R/W	0x0	Receiver overflow error interrupt enable: 0: disabled 1: enabled
2	UNDERRUNEN	R/W	0x0	Slave transmitter underrun interrupt enable: 0: disabled 1: enabled
1	RXIEN	R/W	0x0	RX FIFO data available interrupt enable: 0: disabled 1: enabled
0	TXIEN	R/W	0x0	TX FIFO vacancy available interrupt enable: 0: disabled 1: enabled

Note: Setting a bit in this register to 0 can prevent the corresponding interrupt signal from being generated, but it cannot prevent the corresponding flag in the interrupt status register (SPI2_INTSTAT) from being set to 1 when the condition is met.

33.6.7 Interrupt Clear Register (SPI2_INTCLR)

Offset address: 0x18

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:10	RSV	-	-	Reserved
9	CSPOS_ICLR	R/W	0x0	Chip select rising edge interrupt (SPI as slave) clear: 0: not cleared 1: cleared
8	CSNEG_ICLR	R/W	0x0	Chip select falling edge interrupt (SPI as slave) clear: 0: not cleared 1: cleared
7	TXMATCHCLR	W	0x0	Transmission complete interrupt clear: 0: not cleared 1: cleared
6	TXEPT_ICLR	W	0x0	Transmitter empty interrupt clear: 0: not cleared 1: cleared
5	RXFIFO_FULL_ICLR	W	0x0	RX FIFO full interrupt clear: 0: not cleared 1: cleared
4	RXMATCHCLR	W	0x0	Reception complete interrupt clear: 0: not cleared 1: cleared
3	RXOERRCLR	W	0x0	Receiver overflow error interrupt clear: 0: not cleared 1: cleared
2	UNDERRUNCLR	W	0x0	Slave transmitter underrun interrupt clear: 0: not cleared 1: cleared
1	RXICLR	W	0x0	RX FIFO data available interrupt clear: This bit is valid only when DMAMODE is 0. 0: not cleared 1: cleared
0	TXICLR	W	0x0	TX FIFO vacancy available interrupt clear: This bit is valid only when DMAMODE is 0. 0: not cleared 1: cleared

Writing 1 to a bit in this register will clear the corresponding bit in the interrupt status register (SPI2_INTSTAT).

33.6.8 Global Control Register (SPI2_GCTL)

Offset address: 0x1C

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:24	RSV	-	-	Reserved
23:20	TXTLF	R/W	0x0	Selection of the number of empty bits that can trigger the TX FIFO to receive data: 0000: TX FIFO with at least 1 empty bit (default) 0001: TX FIFO with at least 2 empty bits 0010: TX FIFO with at least 3 empty bits ….: TX TXFIFO with at least “TXTLF + 1” empty bits 0111: TX FIFO with at least 8 empty bits
19:16	RXTLF	R/W	0x0	Selection of the number of data that can trigger the RX FIFO to transmit data: 0000: RX FIFO with at least 1 data (default) 0001: RX FIFO with at least 2 data 0010: RX FIFO with at least 3 data ….: RX FIFO with at least “RXTLF +1” data (default) 0111: RX FIFO with at least 8 data
15:13	RSV	-	-	Reserved
12	HALF_DIR	R/W	0x0	Half-duplex mode data output enable: 0: data input 1: data output
11	BIDIRMODE	R/W	0x0	Half-duplex mode enable: In the master half-duplex mode, MOSI acts as a bidirectional data line; in the slave half-duplex mode, MISO acts as a bidirectional data line. The direction of data transmission in half-duplex mode depends on “txen & !rxen”, i.e., when txen == 1

Bit	Name	Attribute	Reset Value	Description
				and rxen == 0, the pin outputs data; otherwise, the pin inputs data. For full-duplex mode, simply configure the unused pin for GPIO function. 0: full-duplex mode or simplex mode 1: half-duplex mode
10	CSN_SEL	R/W	0x0	Chip select signal controller selection: In the Motorola timing mode, the chip select signal can be controlled by software or hardware. Under hardware control, the chip select signal will still be consistent with the SPI2_SCSR register during data transmission; however, after transmitting or receiving each SPI character, the chip select signal will be pulled high for one PCLK cycle and then pulled low again to transmit the next SPI character. When all data has been transmitted, the chip select signal will be automatically pulled high. In the TI timing mode, the chip select signal is controlled jointly by software and hardware, and this bit is invalid. The chip select signal will be consistent with the SPI2_SCSR register during idle or data transmission; however, before transmitting or receiving each SPI character, the chip select signal will be pulled high for one SCLK cycle and then pulled low again to transmit that SPI character. 0: Chip select signal is configured by the SPI2_SCSR register. 1: Chip select signal is controlled by hardware.
9	DMAMODE	R/W	0x0	DMA access mode enable: 0: use CPU access mode (CPU reads and writes TX FIFO and RX FIFO) 1: use DMA access mode (DMA reads and writes TX FIFO and RX FIFO)
8:5	RSV	-	-	Reserved
4	RXEN	R/W	0x0	Receive enable:

Bit	Name	Attribute	Reset Value	Description
				0: data reception disabled and RX FIFO cleared 0: data reception enabled
3	TXEN	R/W	0x0	Transmit enable: 0: data transmission disabled and TX FIFO cleared 1: data transmission enabled
2	MM	R/W	0x0	Master/slave mode selection: 0: slave mode (serial clock generated by external module) 1: master mode (serial clock generated by this module)
1	INT_EN	R/W	0x0	SPI interrupt enable: 0: all SPI interrupts disabled 1: SPI interrupts enabled (SPI2_INTEN register also needs to be configured)
0	SPI_EN	R/W	0x0	SPI enable: 0: SPI disabled (most circuitry remains in reset state.) 1: SPI enabled

33.6.9 General Control Register (SPI2_CCTL)

Offset address: 0x20

Reset value: 0x0000 0700

Bit	Name	Attribute	Reset Value	Description
31:13	RSV	-	-	Reserved
12:8	SPILEN	R/W	0x7	SPI character length control: define the length of each SPI character as (SPILEN + 1) bits; support character lengths from 4 to 32 bits, with a default of 8 bits.
7	RX_STITCH	R/W	0x0	Receive data concatenation enable: 0: disabled 1: enabled
6	TX_STITCH	R/W	0x0	Transmit data concatenation enable: 0: disabled

Bit	Name	Attribute	Reset Value	Description
				1: enabled
5	LSBFE	R/W	0x0	Transmission bit order selection: 0: transmit or receive data with MSB first 1: transmit or receive data with LSB first
4	RXEDGE	R/W	0x0	Data receive phase adjustment in master mode: 0: data sampled on the receive edge of SCLK 1: data sampled on the next transmit edge of SCLK (delayed)
3	TXEDGE	R/W	0x0	Data transmit phase adjustment in slave mode: Note: When 2 $f_{SCLK} \le f_{PCLK} < 3\ f_{SCLK}$, this bit can only be set to 1; when $f_{PCLK} \ge 3\ f_{SCLK}$, this bit is recommended to be set to 0. 0: transmit data changed on the transmit edge of SCLK 1: transmit data changed on the rising edge of PCLK (in advance)
2	TI_MOD	R/W	0x0	Timing mode selection: In TI timing mode, an extra idle SCLK cycle is required to pull SSN high before transmitting each character, and its CKPH is opposite to that in Motorola timing mode. 0: Motorola timing mode 1: TI timing mode
1	CKPL	R/W	0x0	SCLK polarity: The clock polarity determines whether the clock is high or low when idle. 0: SCLK is low when idle. 1: SCLK is high when idle.
0	CKPH	R/W	0x0	SCLK phase: 1 (in Motorola mode) or 0 (in TI mode): Data changes on the first SPI clock edge and is sampled on the second clock edge. 0 (in Motorola mode) or 1 (in TI mode): Data is sampled on the first SPI clock edge and changes on the second clock edge.

33.6.10 SPI Clock Frequency Control Register (SPI2_SPBRG)

Offset address: 0x24

Reset value: 0x0000 0002

Bit	Name	Attribute	Reset Value	Description
31:8	RSV	-	-	Reserved
7:0	SPBRG	R/W	0x2	This register is used to control the SPI clock frequency. The SPI clock frequency formula is: $f_{SCLK} = f_{PCLK} / SPBRG$ Where f_{PCLK} is the APB clock frequency. Note: Do not write 0 or 1 to this register; writing 0 or 1 is invalid.

33.6.11 Receive Data Number Register (SPI2_RXDNR)

Offset address: 0x28

Reset value: 0x0000 0001

Bit	Name	Attribute	Reset Value	Description
31:16	RSV	-	-	Reserved
15:0	RXDNR	R/W	0x1	This register is used to store the number of characters (not concatenated) to be received in the next reception. SPI master mode: This register is valid only in simplex receive mode and will affect the number of cycles of the generated SPI clock, so it must be properly configured. SPI slave mode (valid only for character concatenation): This register is valid in both simplex receive mode and duplex (receive and transmit) mode. In particular, in duplex mode, it indicates both the number of characters to be received and to be transmitted. The default value of this register is 1, and it

Bit	Name	Attribute	Reset Value	Description
				will not change until the CPU writes a new value to it. Note: Do not write 0 to this register.

33.6.12 Transmit Data Number Register (SPI2_TXDNR)

Offset address: 0x2C

Reset value: 0x0000 0001

Bit	Name	Attribute	Reset Value	Description
31:16	RSV	-	-	Reserved
15:0	TXDNR	R/W	0x1	This register is used to store **the number of characters (not concatenated) to be transmitted in the next transmission**, it is only valid for character concatenation. **SPI master mode:** This register is valid in both simplex transmit mode and duplex (receive and transmit) mode. In particular, in duplex mode, it indicates both the number of characters to be transmitted and to be received. **SPI slave mode:** This register is valid only in simplex transmit mode. The default value of this register is 1, and it will not change until the CPU writes a new value to it. Note: Do not write 0 to this register.

33.6.13 Slave Chip Select Register (SPI2_SCSR)

Offset address: 0x30

Reset value: 0x0000 00FF

Bit	Name	Attribute	Reset Value	Description
31:8	RSV	-	-	Reserved
7:0	CS0	R/W	0x1	The chip select output signal in master mode is active low. This register is not available in SPI slave mode.

Bit	Name	Attribute	Reset Value	Description
				0: the corresponding slave selected 1: the corresponding slave not selected Note: Only bit [0] is useful, and bits [7:1] are not required.

33.7 Operation Procedure

33.7.1 Master Mode

33.7.1.1 Master Duplex Transmit/Receive or Simplex Transmit

1. Register configuration

   A. Configure the system registers related to peripheral clock, reset, and pin alternate function, to enable the clock and release the reset.

   B. Set SPI2_GCTL[0:3] to 1, and configure other bits as required.

   C. Configure the SPI2_CCTL register to set the timing mode and data format.

   D. Configure the SPI2_SPBRG register to set the SCLK division factor.

   E. Write 0xFF to the SPI2_INTCLR register to clear the interrupt status of the SPI2_INTSTAT register.

   F. Configure the SPI2_INTEN register to optionally enable interrupts.

   G. When the character concatenation function is enabled, configure the SPI2_TXDNR register to set the number of characters to be transmitted.

   H. Configure the SPI2_SCSR register to pull down the chip select signal corresponding to the selected slave.

2. Data transmission

   A. If data needs to be transmitted, write the data to be transmitted into the SPI2_TXREG register through CPU or DMA.

B. When SPI2_CSTAT[1] is 1, it indicates that data has been received. If data needs to be received, read the received data from the SPI2_RXREG register through CPU or DMA.

3. Interrupt handling

If an interrupt occurs, read the SPI2_MINTSTAT register to determine the cause of the interrupt. After processing, write 1 to the corresponding bit of the SPI2_INTCLR register to clear the interrupt status in the SPI2_INTSTAT and SPI2_MINTSTAT registers.

4. End of transmission

A. Wait for SPI2_INTSTAT[6] to become 1, or wait for SPI2_CSTAT[0] to become 1. The rate of data transmission may be slower than the rate of writing to the TX FIFO, so writing the data to be transmitted to the SPI2_TXREG register does not mean that all the data has been transmitted. If the transmitter is not empty, wait for all data in the transmitter to be transmitted before concluding that the transmission is complete.

B. Configure the SPI2_SCSR register to pull up the chip select signal. If hardware control of the chip select signal is chosen, this register may not need to be configured.

C. Set SPI2_GCTL[0] to 0 to disable the SPI module.

D. Configure the system registers related to peripheral clock and reset to disable the clock and restore the reset.

33.7.1.2 Master Half-duplex Receive

1. Register configuration

A. Configure the system registers related to peripheral clock, reset, and pin alternate function (the master using MOSI for communication), to enable the clock and release the reset.

B. Configure the SPI2_GCTL register, noting that SPI2_GCTL[4] and SPI2_GCTL[3] shall not be set temporarily; set SPI2_GCTL[11], SPI2_GCTL[0] and SPI2_GCTL[2] to 1, and configure other bits as required.

C. Configure the SPI2_CCTL register to set the timing mode and data format.

D. Configure the SPI2_SPBRG register to set the SCLK division factor.

E. Write 0xFF to the SPI2_INTCLR register to clear the interrupt status in the SPI2_INTSTAT and SPI2_MINTSTAT registers.

F. Configure the SPI2_INTEN register to optionally enable interrupts.

G. Configure the SPI2_SCSR register to pull down the chip select signal corresponding to the selected slave.

2. Data transmission

A. Set SPI2_GCTL[12] and SPI2_GCTL[3] to 0, and set SPI2_GCTL[4] and SPI2_GCTL[0] to 1.

B. When SPI2_CSTAT[1] is 1, it indicates that data has been received. If data needs to be received, read the received data from the RXREG register through CPU or DMA.

C. Set SPI2_GCTL [4] to 0.

3. Interrupt handling

If an interrupt occurs, read the SPI2_MINTSTAT register to determine the cause of the interrupt. After processing, write 1 to the corresponding bit of the SPI2_INTCLR register to clear the interrupt status in the SPI2_INTSTAT and SPI2_MINTSTAT registers.

4. End of transmission

A. Configure the SPI2_SCSR register to pull up the chip select signal. If hardware control of the chip select signal is chosen, this register may not need to be configured.

B. Set SPI2_GCTL[0] to 0 to disable the SPI module.

C. Configure the system registers related to peripheral clock and reset to disable the clock and restore the reset.

33.7.1.3 Master Half-duplex Transmit

1. Configuration register

   A. Configure the system registers related to peripheral clock, reset, and pin alternate function (the master using MOSI for communication), to enable the clock and release the reset.

   B. Configure the SPI2_GCTL register, noting that SPI2_GCTL[4] and SPI2_GCTL[3] shall not be set temporarily; set SPI2_GCTL[11], SPI2_GCTL[0] and SPI2_GCTL[2] to 1, and configure other bits as required.

   C. Configure the SPI2_CCTL register to set the timing mode and data format.

   D. Configure the SPI2_SPBRG register to set the SCLK division factor.

   E. Write 0xFF to the SPI2_INTCLR register to clear the interrupt status in the SPI2_INTSTAT and SPI2_MINTSTAT registers.

   F. Configure the SPI2_INTEN register to optionally enable interrupts.

   G. Configure the SPI2_SCSR register to pull down the chip select signal corresponding to the selected slave.

2. Data transmission

   A. Set SPI2_GCTL[0] to 0, SPI2_GCTL[12] to 1, SPI2_GCTL[3] to 1, SPI2_GCTL[4] to 0, and then set SPI2_GCTL[0] to 1.

   B. When SPI2_CSTAT[2] is 0, it indicates that the TX FIFO is not full, and data can be written into the SPI2_TXREG register.

3. Interrupt handling

If an interrupt occurs, read the SPI2_MINTSTAT register to determine the cause of the interrupt. After processing, write 1 to the corresponding bit of the SPI2_INTCLR register to clear the interrupt status in the SPI2_INTSTAT and SPI2_MINTSTAT registers.

4. End of transmission

   A. When SPI2_CSTAT[0] is 1, it indicates that data has been transmitted from the SPI2_TXREG register.

   B. Configure the SPI2_SCSR register to pull up the chip select signal. If hardware control of the chip select signal is chosen, this register may not need to be configured.

   C. Set SPI2_GCTL[0] to 0 to disable the SPI module.

   D. Configure the system registers related to peripheral clock and reset to disable the clock and restore the reset.

33.7.1.4 Master Simplex Receive

1. Register configuration

   A. Configure the system registers related to peripheral clock, reset, and pin alternate function, to enable the clock and release the reset.

   B. Configure the SPI2_GCTL register, noting that SPI2_GCTL[4] shall not be set to 1 temporarily; set SPI2_GCTL[0] and SPI2_GCTL[2] to 1, and configure other bits as required.

   C. Configure the SPI2_CCTL register to set the timing mode and data format.

   D. Configure the SPI2_SPBRG register to set the SCLK division factor.

   E. Write 0xFF to the SPI2_INTCLR register to clear the interrupt status in the SPI2_INTSTAT and SPI2_MINTSTAT registers.

   F. Configure the SPI2_INTEN register to enable interrupts.

G. Configure the SPI2_RXDNR register to set the number of characters to be received.

H. Configure the SPI2_SCSR register to pull down the chip select signal corresponding to the selected slave.

I. Configure the SPI2_GCTL register and set SPI2_GCTL[4] to 1.

2. Data transmission

When SPI2_CSTAT[1] is 1, it indicates that data has been received. If data needs to be received, read the received data from the SPI2_RXREG register through CPU or DMA.

3. Interrupt handling

If an interrupt occurs, read the SPI2_MINTSTAT register to determine the cause of the interrupt. After processing, write 1 to the corresponding bit of the SPI2_INTCLR register to clear the interrupt status in the SPI2_INTSTAT and SPI2_MINTSTAT registers.

4. End of transmission

A. Since the master controls the clock and chip select signal, it also controls when the transmission ends. When the character concatenation function is enabled, wait for SPI2_INTSTAT[4] to become 1, indicating that the specified number of characters in the SPI2_RXDNR register has been received, i.e., the reception is complete. At the same time, set SPI2_GCTL[4] to 0.

B. Configure the SPI2_SCSR register to pull up the chip select signal. If hardware control of the chip select signal is chosen, this register may not need to be configured.

C. Set SPI2_GCTL[0] to 0 to disable the SPI module.

D. Configure the system registers related to peripheral clock and reset to disable the clock and restore the reset.

33.7.2 Slave Mode

33.7.2.1 Slave Duplex Transmit/Receive or Simplex Receive

1. Register configuration

   A. Configure the system registers related to peripheral clock, reset, and pin alternate function, to enable the clock and release the reset.

   B. Set SPI2_GCTL[0] and SPI2_GCTL[4] to 1, set SPI2_GCTL[2] to 0 and configure other bits as required.

   C. Configure the SPI2_CCTL register to set the timing mode and data format.

   D. Write 0xFF to the SPI2_INTCLR register to clear the interrupt status in the SPI2_INTSTAT and SPI2_MINTSTAT registers.

   E. Configure the SPI2_INTEN register to enable interrupts.

   F. When the character concatenation function is enabled, configure the SPI2_RXDNR register to set the number of characters to be received.

2. Data transmission

   A. If data needs to be transmitted, write the data to be transmitted into the SPI2_TXREG register through CPU or DMA. Note: The slave must be prepared with the data to be transmitted before the master pulls the chip select signal low.

   B. When SPI2_CSTAT[1] is 1, it indicates that data has been received. If data needs to be received, read the received data from the SPI2_RXREG register through CPU or DMA.

3. Interrupt handling

   If an interrupt occurs, read the SPI2_MINTSTAT register to determine the cause of the interrupt. After processing, write 1 to the corresponding bit of the SPI2_INTCLR register to clear the interrupt status in the SPI2_INTSTAT and SPI2_MINTSTAT registers.

4. End of transmission

   A. Since the master controls the clock and chip select signal, when the end of transmission is also controlled by the master. When character concatenation is enabled, the slave may wait for SPI2_INTSTAT[4] to become 1, indicating that the specified number of SPI characters set in the SPI2_RXDNR register has been received, i.e., the reception is complete. At the same time, set SPI2_GCTL[4] to 0.

   B. After the transmission ends, configure SPI2_GCTL[0] to 0 to disable the SPI module.

   C. Configure the system registers related to peripheral clock and reset to disable the clock and restore the reset.

33.7.2.2 Slave Half-duplex Receive

1. Register configuration

   A. Configure the system registers related to peripheral clock, reset, and pin alternate function (the slave using MISO for communication), to enable the clock and release the reset.

   B. Configure the SPI2_GCTL register, noting that SPI2_GCTL[4] and SPI2_GCTL[3] shall not be set temporarily; set SPI2_GCTL[11] and SPI2_GCTL[0] to 1, set SPI2_GCTL[2] to 0, and configure other bits as required.

   C. Configure the SPI2_CCTL register to set the timing mode and data format.

   D. Write 0xFF to the SPI2_INTCLR register to clear the interrupt status in the SPI2_INTSTAT and SPI2_MINTSTAT registers.

   E. Configure the SPI2_INTEN register to optionally enable interrupts.

2. Data transmission

A. Set SPI2_GCTL[12] and SPI2_GCTL[3] to 0, and set SPI2_GCTL[4] and Bit SPI2_GCTL[0] to 1.

B. When SPI2_CSTAT[1] is 1, it indicates that data has been received. If data needs to be received, read the received data from the SPI2_RXREG register through CPU or DMA.

3. Interrupt handling

If an interrupt occurs, read the SPI2_MINTSTAT register to determine the cause of the interrupt. After processing, write 1 to the corresponding bit of the SPI2_INTCLR register to clear the interrupt status in the SPI2_INTSTAT and SPI2_MINTSTAT registers.

4. End of transmission

A. Since the master controls the clock and chip select signal, when the end of transmission is also controlled by the master.

B. After the transmission ends, set SPI2_GCTL[0] to 0 to disable the SPI module.

C. Configure the system registers related to peripheral clock and reset to disable the clock and restore the reset.

33.7.2.3 Slave Half-duplex Transmit

1. Register configuration

A. Configure the system registers related to peripheral clock, reset, and pin alternate function (the slave using MISO for communication), to enable the clock and release the reset.

B. Configure the SPI2_GCTL register, noting that SPI2_GCTL[4] and SPI2_GCTL[3] shall not be set temporarily; set SPI2_GCTL[11] and SPI2_GCTL[0] to 1, set SPI2_GCTL[2] to 0, and configure other bits as required.

C. Configure the SPI2_CCTL register to set the timing mode and data format.

D. Write 0xFF to the SPI2_INTCLR register to clear the interrupt status in the SPI2_INTSTAT and SPI2_MINTSTAT registers.

E. Configure the SPI2_INTEN register to optionally enable interrupts.

2. Data transmission

A. If data needs to be transmitted, write the data to be transmitted into the SPI2_TXREG register through CPU or DMA. Note: The slave must be prepared with the data to be transmitted before the master pulls the chip select signal low.

B. Set SPI2_GCTL[12] and SPI2_GCTL[3] to 1, set SPI2_GCTL[4] to 0, and then set bit SPI2_GCTL[0] to 1.

C. Write data into the SPI2_TXREG register.

D. When SPI2_CSTAT[0] is 1, it indicates that the data has been transmitted.

E. Set SPI2_GCTL [3] to 0.

3. Interrupt handling

If an interrupt occurs, read the SPI2_MINTSTAT register to determine the cause of the interrupt. After processing, write 1 to the corresponding bit of the SPI2_INTCLR register to clear the interrupt status in the SPI2_INTSTAT and SPI2_MINTSTAT registers.

4. End of transmission

A. Since the master controls the clock and chip select signal, when the end of transmission is also controlled by the master. It is recommended for the slave to wait for SPI2_INTSTAT[6] to become 1, or wait for SPI2_CSTAT[0] to become 1, indicating that all data in the transmitter has been transmitted.

B. After the transmission ends, set SPI2_GCTL[0] to 0 to disable the SPI module.

C. Configure the system registers related to peripheral clock and reset to disable the clock and restore the reset.

33.7.2.4 Slave Simplex Transmit

1. Register configuration

   A. Configure the system registers related to peripheral clock, reset, and pin alternate function, to enable the clock and release the reset.

   B. Set SPI2_GCTL[0] and SPI2_GCTL[3] to 1, set SPI2_GCTL[2] to 0 and configure other bits as required.

   C. Configure the SPI2_CCTL register to set the timing mode and data format.

   D. Write 0xFF to the SPI2_INTCLR register to clear the interrupt status in the SPI2_INTSTAT and SPI2_MINTSTAT registers.

   E. Configure the SPI2_INTEN register to enable interrupts.

   F. When the character concatenation function is enabled, configure the SPI2_TXDNR register to set the number of characters to be transmitted.

2. Data transmission

   A. If data needs to be transmitted, write the data to be transmitted into the SPI2_TXREG register through CPU or DMA. Note: The slave must be prepared with the data to be transmitted before the master pulls the chip select signal low.

   B. When SPI2_CSTAT[1] is 1, it indicates that data has been received. If data needs to be received, read the received data from the SPI2_RXREG register through CPU or DMA.

3. Interrupt handling

If an interrupt occurs, read the SPI2_MINTSTAT register to determine the cause of the interrupt. After processing, write 1 to the corresponding bit of the SPI2_INTCLR register to clear the interrupt status in the SPI2_INTSTAT and SPI2_MINTSTAT registers.

4. End of transmission

   A. Since the master controls the clock and chip select signal, it also controls when the transmission ends. It is recommended for the slave to wait for SPI2_INTSTAT[6] to become 1, or wait for SPI2_CSTAT[0] to become 1, indicating that all data in the transmitter has been transmitted.

   B. After the transmission ends, configure SPI2_GCTL[0] to 0 to disable the SPI module.

   C. Configure the system registers related to peripheral clock and reset to disable the clock and restore the reset.

34 Universal Serial Bus Full-speed Device Interface (USB)

34.1 Overview

The USB device controller is an interface compatible with the USB 2.0 full-speed protocol, which works in conjunction with USB PHY to realize communication between the chip and the USB HOST.

34.2 Main Features

- USB1.1 and USB2.0 full-speed protocols compliant
- 4 universal bidirectional transmission endpoints (EP1, EP2, EP3, EP4)
- Endpoints support a packet length of up to 64 bytes, and support Memory and FIFO access modes
- FIFO mode supports 32-bit access
- Memory mode supports 8/16/32-bit accesses
- Suspend, wake-up and remote wake-up functions
- Toggle hardware comparison and software control functions
- Interrupt enabled for data transmission on each endpoint
- Interrupt enabled for bus reset, suspend and resume operations
- Optional NAK handshake packet for CRC error
- Automatic NAK response for data packets exceeding the maximum packet length (64 bytes)
- NAK response from USB device for subsequent IN operations if the host does not return ACK
- Detection of lost token packets and data packets EOP, with optional automatic NAK response for lost EOPs

34.3 Functional Description

34.3.1 Interrupt Status and Control Registers

For each interrupt of the USB device, there are interrupt status register, interrupt enable register, and interrupt clear register, as detailed below:

- USB_INTSTATRAW: Raw interrupt status register that displays the interrupt occurrence status regardless of whether the interrupt is enabled.
- USB_INTEN: Interrupt enable register that enables a specific interrupt.
- USB_INTCLR: Interrupt clear register that is used to clear the interrupt occurrence flag. Just write 1 to clear the interrupt, and the register will be cleared automatically.

34.3.2 Address Setting

The Set Address command in control transfers can be entirely handled by hardware. Software can use the interrupt bit SETADDR to know that the Set Address command has occurred.

34.3.3 Remote Wakeup

The remote wakeup function is implemented through USB_WORKINGMODE[20]. Writing 1 to USB_WORKINGMODE[20] will send a K state on the USB port, with the duration controlled by software. Writing 0 to USB_WORKINGMODE[20] will stop the USB port from sending K states.

Before initiating remote wakeup, it is necessary to confirm whether the current USB PHY is enabled and whether the CLK48M clock generated by PHY is activated. Without the CLK48M clock, the controller will not be able to initiate remote wakeup.

34.3.4 Selective NAK Response for CRC Errors in Token and Data Packets

When CRC errors occur in token packets and data packets, the controller can selectively respond with NAK, controlled by USB_WORKINGMODE[10]. When USB_WORKINGMODE[10] is set to 1, the USB device will respond with an NAK handshake packet during the handshake phase in the event of a CRC error; otherwise, no handshake packet will be returned.

34.3.5 Packet Length Exceeding 64 Bytes

When the endpoint of the USB controller receives a data packet with a length exceeding 64 bytes, the status register USB_INTSTATRAW[24] will be set to 1. If USB_INTEN[24] is set to 1, an interrupt will be generated.

34.3.6 Packet Loss EOP

When both USB_WORKINGMODE[23] and USB_WORKINGMODE[24] are set to 1, if a token packet or data packet does not receive the EOP (end of packet) within the specified packet length, the status register USB_INTSTATRAW[30] will be set to 1. If USB_INTEN[30] is set to 1, an interrupt will be generated. The controller detects the token packet based on a length of 32 full-speed bits. If the token packet exceeds 32 full-speed bits without receiving an EOP (end of packet), this error will be triggered, and the device will reset its internal state machine, ensuring that the reception of the next packet is not affected. The data packet is detected based on the valid data length of 64 bytes + 8 bytes. If the actual length of the valid data in the packet exceeds this value, this error will also be triggered.

When such errors occur in the token packet and data packet, it is possible to choose whether to respond to the USB host with an NAK handshake packet. This can be configured by setting USB_WORKINGMODE[21] and USB_WORKINGMODE[22] respectively. When

USB_WORKINGMODE[21] is set to 1, upon the occurrence of such an error in the token packet, the USB controller will respond with an NAK handshake packet after the (32 + 11)th full-speed bit of the token packet. When USB_WORKINGMODE[22] is set to 1, upon the occurrence of such an error in the data packet, the USB controller will respond with an NAK handshake packet after the (64 bytes + 8 bytes + 11 bits)th full-speed bit of the data packet.

34.3.7 IN Operation ACK Timeout, Responding NAK for Next IN Operation

When the host initiates an IN operation, if the ACK handshake packet that the host needs to respond with times out during the handshake phase, the EPx_IN_HANDSHAKE_ERR_RAW in the status register USB_INTSTATRAW will be set to 1 for different endpoints. When EPx_IN_HANDSHAKE_ERR_RAW is set to 1 and the host initiates another IN operation for that endpoint, the USB controller will respond with an NAK handshake packet until the software clears the EPx_IN_HANDSHAKE_ERR_RAW bit.

34.3.8 FIFO Access and Memory Access

This USB controller supports both FIFO access and memory access. FIFO access only supports 32-bit access, while memory access supports 8-, 16-, and 32-bit access.

34.4 Register Description

USB register base address: 0x4090_0000

The registers are listed below:

Table 34-1: List of USB Registers

Offset	Name	Description
0X00	USB_WORKINGMODE	Working mode register
0X04	USB_EP0CSR	EP0 transfer control register
0X08	USB_EP1CSR	EP1 transfer control register
0X0C	USB_EP2CSR	EP2 transfer control register

Offset	Name	Description
0X10	USB_EP3CSR	EP3 transfer control register
0X14	USB_EP4CSR	EP4 transfer control register
0X18	USB_ADDR	USB address register
0X1C	USB_SETUP03DATA	SETUP data packet register 0
0X20	USB_SETUP47DATA	SETUP data packet register 1
0X24	USB_EPADDR	Endpoint address configuration register
0X28	USB_CURRENTPID	Current USB bus packet PID register
0X2C	USB_CURRENTFRAMENUMBER	Frame number register
0X30	USB_CRCERRORCNT	CRC error counter register
0X34	USB_STATUSDETECTCNT	Suspend/resume/reset detection time register
0X40	USB_EP0SENDBN	EP0 transmit data number register
0X44	USB_EP1SENDBN	EP1 transmit data number register
0X48	USB_EP2SENDBN	EP2 transmit data number register
0X4C	USB_EP3SENDBN	EP3 transmit data number register
0X50	USB_EP4SENDBN	EP4 transmit data number register
0X100	USB_EP0FIFO	EP0 FIFO access entry
0X104	USB_EP1FIFO	EP1 FIFO access entry
0X108	USB_EP2FIFO	EP2 FIFO access entry
0X10C	USB_EP3FIFO	EP3 FIFO access entry
0X110	USB_EP4FIFO	EP4 FIFO access entry
0x200–23C	USB_EP0MEM	EP0 memory access entry
0x240–27C	USB_EP1MEM	EP1 memory access entry
0x280–2BC	USB_EP2MEM	EP2 memory access entry
0x2C0–2FC	USB_EP3MEM	EP3 memory access entry
0x300–33C	USB_EP4MEM	EP4 memory access entry
0XFFE4	USB_INTSTATRAW	Status register
0XFFE8	USB_INTEN	Interrupt enable register
0XFFF0	USB_INTCLR	Interrupt clear register

34.4.1 Working Mode Register (USB_WORKINGMODE)

Offset address: 0x00

Reset value: 0x0000 0009

Bit	Name	Attribute	Reset Value	Description
31:26	RSV	-	-	Reserved
25	USB_EP0_ZOD_INTR_EN	R/W	0x0	Interrupt generation enable for 0-length data packet in USB OUT operation: 1: enabled 0: disabled
24	USB_DATA_NOEOP_EN	R/W	0x0	When the USB controller receives a data packet longer than 64 + 8 bytes, it will consider this data packet as losing the EOP, and the controller will reset the state machine internally and generate an error state. 1: this function enabled 0: this function disabled
23	USB_TOKEN_NOEOP_EN	R/W	0x0	When the USB controller receives a token packet that exceeds the specified length, it will consider this token packet as losing the EOP, and the controller will reset the state machine internally and generate an error state. 1: this function enabled 0: this function disabled
22	USB_DATA_NOEOP_NAK_EN	R/W	0x0	When the USB controller receives a data packet longer than 64 + 8 bytes, it will consider this data packet as losing the EOP. When this bit is set to 1, the controller will reply with an NAK handshake packet to the host. 1: reply with NAK 0: do not reply with NAK This function is valid only when

Bit	Name	Attribute	Reset Value	Description
				USB_DATA_NOEOP_EN is set to 1.
21	USB_TOKEN_NOEOP_NAK_EN	R/W	0x0	When the USB controller receives a token packet that exceeds the specified length, it will consider this token packet as losing the EOP. When this bit is set to 1, the controller will reply with an NAK handshake packet to the host. 1: reply with NAK 0: do not reply with NAK This function is valid only when USB_TOKEN_NOEOP_EN is set to 1.
20	USB_REMOTE_WAKEUP	R/W	0x0	USB remote wakeup control bit: 1: The controller sends K state. 0: The controller does not send K state. Upon completion of the wakeup, this bit shall be cleared by software writing 0.
19:13	RSV	-	-	Reserved
12	USB_MORETHAN64_NAK_EN	R/W	0x0	Enable reply with NAK handshake packet for receiving a data packet longer than 64 bytes: 1: this function enabled 0: this function disabled
11	USB_IN_TIMEOUT_NAK_EN	R/W	0x0	After the IN operation data packet is sent successfully, if the controller does not receive the ACK handshake packet from the host, the device will reply with NAK for the next IN operation on that endpoint, until the software clears the error status bit. 1: this function enabled 0: this function disabled
10	USB_CRCERR_NAK_EN	R/W	0x0	Enable NAK response for CRC errors in token and data packets: 1: this function enabled 0: this function disabled
9:8	USB_LINE_STATE	R	0x0	USB DP/DM signal status bit: Bit 8: DM

Bit	Name	Attribute	Reset Value	Description
				Bit 9: DP
7:6	RSV	-	-	Reserved
5	USB_DMPU	R/W	0x0	This bit can only be written as 0.
4	USB_DPPU	R/W	0x0	USB controller DP signal 2.1k pull-up resistor control bit: 1: pull-up enabled 0: pull-up disabled
3	USB_BUS_AUTO_RST_EN	R/W	0x1	Enable USB bus reset of controller address, transmit/receive state machine and TX/RX FIFO: 1: enabled 0: disabled
2	USB_FORCE_RST	R/W	0x0	Reset controller address, transmit/receive state machine and TX/RX FIFO: 1: reset 0: not reset After each reset operation, the software shall write 0 to clear this bit.
1	USB_SUSPEND	R/W	0x0	1: Set USB PHY to be in suspended state. 0: Clear the suspended state of USB PHY.
0	SPEED_MODE	R/W	0x1	USB operation mode selection bit: 1: full-speed mode 0: low-speed mode This chip only supports full-speed mode. This bit can only be written as 1.

34.4.2 EP0 Transfer Control Register (USB_EP0CSR)

Offset address: 0x04

Reset value: 0x0000 0100

Bit	Name	Attribute	Reset Value	Description
31:25	RSV	-	-	Reserved
24	EP0_RECEIVED	R	0x0	Flag of EP0 receiving the ACK handshake

Bit	Name	Attribute	Reset Value	Description
	_ACK			packet from the host: 1: handshake packet received 0: handshake packet not received This bit is read-only and can be cleared by USB reset or writing 1 to EP0_DATA_START.
23:20	RSV	-	-	Reserved
19	EP0_OUT_TOGGLE_STATE	R	0x0	EP0 out/setup status error flag: 1: toggle received not as expected 0: toggle is normal
18	EP0_OUT_TOGGLE_CTRL_En	W	0x0	Enable the value in EP0_OUT_TOGGLE_WANT to take effect: 1: The value written to EP0_OUT_TOGGLE_WANT is effective.
17	EP0_OUT_TOGGLE_WANT	R/W	0x0	Comparison value of toggle for EP0 out/setup packets: 1: Data 1 0: Data 0
16	EP0_OUT_TOGGLE_VALUE	R	0x0	Toggle value of packets received by EP0: 1: Data 1 0: Data 0
15	EP0_IN_TOGGLE_CTRL_EN	W	0x0	Enable the value in EP0_IN_TOGGLE_VALUE to take effect: 1: The value written to EP0_IN_TOGGLE_VALUE is effective.
14	EP0_IN_TOGGLE_VALUE	R/W	0x0	IN operation toggle control bit: 1: Data 1 0: Data 0 Writing this bit will change the comparison register value, reading this bit returns the current IN toggle value.
13	EP0_SEND_STALL_DONE	R/W	0x0	EP0 STALL transmission complete flag: 1: STALL transmission completed 0: STALL transmission not completed Writing 1 clears this bit.
12	EP0_SEND_STALL	W	0x0	STALL transmission control bit: 1: STALL transmitted

Bit	Name	Attribute	Reset Value	Description
				0: STALL not transmitted A single write operation transmits STALL only once.
11	EP0_RECEIVED_DONE	W	0x0	Reception complete control bit: 1: reception completed 0: reception not completed Writing 1 to this bit will set the FIFO to ready state, and it is necessary to write a 1 to this bit each time the data is received and read, otherwise, the device will respond with NAK during the next OUT/SETUP operation.
10	EP0_DATA_START	W	0x0	Transmit START: 1: transmit the data in FIFO 0: no operation
9	EP0_FIFOCLR	W	0x0	EP0 FIFO pointer reset control bit: 1: reset FIFO pointer 0: do not reset FIFO pointer
8	EP0_EN	R/W	0x1	EP0 enable: 1: enabled 0: disabled
7:0	EP0_RECEIVED_BYTE	R	0x0	Number of data bytes received by EP0: This value must be less than the maximum packet length.

34.4.3 EP1 Transfer Control Register (USB_EP1CSR)

Offset address: 0x08

Reset value: 0x0000 0100

Bit	Name	Attribute	Reset Value	Description
31:25	RSV	-	-	Reserved
24	EP1_RECEIVED_ACK	R	0x0	Flag of EP1 receiving the ACK handshake packet from the host: 1: handshake packet received 0: handshake packet not received

Bit	Name	Attribute	Reset Value	Description
				This bit is read-only and can be cleared by USB reset or writing 1 to EP1_DATA_START.
23:20	RSV	-	-	Reserved
19	EP1_OUT_TOGGLE_STATE	R	0x0	EP1 out/setup status error flag: 1: toggle received not as expected 0: toggle is normal
18	EP1_OUT_TOGGLE_CTRL_En	W	0x0	Enable the value in EP1_OUT_TOGGLE_WANT to take effect: 1: The value written to EP1_OUT_TOGGLE_WANT is effective.
17	EP1_OUT_TOGGLE_WANT	R/W	0x0	Comparison value of toggle for EP1 out/setup packets: 1: Data 1 0: Data 0
16	EP1_OUT_TOGGLE_VALUE	R	0x0	Toggle value of packets received by EP1: 1: Data 1 0: Data 0
15	EP1_IN_TOGGLE_CTRL_EN	W	0x0	Enable the value in EP1_IN_TOGGLE_VALUE to take effect: 1: The value written to EP1_IN_TOGGLE_VALUE is effective.
14	EP1_IN_TOGGLE_VALUE	R/W	0x0	IN operation toggle control bit: 1: Data 1 0: Data 0 Writing this bit will change the comparison register value, reading this bit returns the current IN toggle value.
13	EP1_SEND_STALL_DONE	R	0x0	EP1 STALL transmission complete flag: 1: STALL transmission completed 0: STALL transmission not completed Writing 1 clears this bit.
12	EP1_SEND_STALL	R/W	0x0	STALL transmission control bit: 1: STALL transmitted 0: STALL not transmitted A single write operation transmits STALL multiple times until this bit is cleared.

Bit	Name	Attribute	Reset Value	Description
11	EP1_RECEIVED_DONE	W	0x0	Reception complete control bit: 1: reception completed 0: reception not completed Writing 1 to this bit will set the FIFO to ready state, and it is necessary to write a 1 to this bit each time the data is received and read, otherwise, the device will respond with NAK during the next OUT/SETUP operation.
10	EP1_DATA_START	W	0x0	Transmit START: 1: transmit the data in FIFO 0: no operation
9	EP1_FIFOCLR	W	0x0	EP1 FIFO pointer reset control bit: 1: reset FIFO pointer 0: do not reset FIFO pointer
8	EP1_EN	R/W	0x1	EP1 enable: 1: enabled 0: disabled
7:0	EP1_RECEIVED_BYTE	R	0x0	Number of data bytes received by EP1: This value must be less than the maximum packet length.

34.4.4 EP2 Transfer Control Register (USB_EP2CSR)

Offset address: 0x0C

Reset value: 0x0000 0100

Bit	Name	Attribute	Reset Value	Description
31:25	RSV	-	-	Reserved
24	EP2_RECEIVED_ACK	R	0x0	Flag of EP2 receiving the ACK handshake packet from the host: 1: handshake packet received 0: handshake packet not received This bit is read-only and can be cleared by USB reset or writing 1 to EP2_DATA_START.
23:20	-	-	-	-

Bit	Name	Attribute	Reset Value	Description
19	EP2_OUT_TOGGLE_STATE	R	0x0	EP2 out/setup status error flag: 1: toggle received not as expected 0: toggle is normal
18	EP2_OUT_TOGGLE_CTRL_En	W	0x0	Enable the value in EP2_OUT_TOGGLE_WANT to take effect: 1: The value written to EP2_OUT_TOGGLE_WANT is effective.
17	EP2_OUT_TOGGLE_WANT	R/W	0x0	Comparison value of toggle for EP2 out/setup packets: 1: Data 1 0: Data 0
16	EP2_OUT_TOGGLE_VALUE	R	0x0	Toggle value of packets received by EP2: 1: Data 1 0: Data 0
15	EP2_IN_TOGGLE_CTRL_EN	W	0x0	Enable the value in EP2_IN_TOGGLE_VALUE to take effect: 1: The value written to EP2_IN_TOGGLE_VALUE is effective.
14	EP2_IN_TOGGLE_VALUE	R/W	0x0	IN operation toggle control bit: 1: Data 1 0: Data 0 Writing this bit will change the comparison register value, reading this bit returns the current IN toggle value.
13	EP2_SEND_STALL_DONE	R	0x0	EP2 STALL transmission complete flag: 1: STALL transmission completed 0: STALL transmission not completed Writing 1 clears this bit.
12	EP2_SEND_STALL	R/W	0x0	STALL transmission control bit: 1: STALL transmitted 0: STALL not transmitted A single write operation transmits STALL multiple times until this bit is cleared.
11	EP2_RECEIVED_DONE	W	0x0	Reception complete control bit: 1: reception completed 0: reception not completed

Bit	Name	Attribute	Reset Value	Description
				Writing 1 to this bit will set the FIFO to ready state, and it is necessary to write a 1 to this bit each time the data is received and read, otherwise, the device will respond with NAK during the next OUT/SETUP operation.
10	EP2_DATA_START	W	0x0	Transmit START: 1: transmit the data in FIFO 0: no operation
9	EP2_FIFOCLR	W	0x0	EP2 FIFO pointer reset control bit: 1: reset FIFO pointer 0: do not reset FIFO pointer
8	EP2_EN	R/W	0x1	EP2 enable: 1: enabled 0: disabled
7:0	EP2_RECEIVED_BYTE	R	0x0	Number of data bytes received by EP2: This value must be less than the maximum packet length.

34.4.5 EP3 Transfer Control Register (USB_EP3CSR)

Offset address: 0x10

Reset value: 0x0000 0100

Bit	Name	Attribute	Reset Value	Description
31:25	RSV	-	-	Reserved
24	EP3_RECEIVED_ACK	R	0x0	Flag of EP3 receiving the ACK handshake packet from the host: 1: handshake packet received 0: handshake packet not received This bit is read-only and can be cleared by USB reset or writing 1 to EP3_DATA_START.
23:20	-	-	-	-
19	EP3_OUT_TOGGLE_STATE	R	0x0	EP3 out/setup status error flag: 1: toggle received not as expected 0: toggle is normal

Bit	Name	Attribute	Reset Value	Description
18	EP3_OUT_TOGGLE_CTRL_En	W	0x0	Enable the value in EP3_OUT_TOGGLE_WANT to take effect: 1: The value written to EP3_OUT_TOGGLE_WANT is effective.
17	EP3_OUT_TOGGLE_WANT	R/W	0x0	Comparison value of toggle for EP3 out/setup packets: 1: Data 1 0: Data 0
16	EP3_OUT_TOGGLE_VALUE	R	0x0	Toggle value of packets received by EP3: 1: Data 1 0: Data 0
15	EP3_IN_TOGGLE_CTRL_EN	W	0x0	Enable the value in EP3_IN_TOGGLE_VALUE to take effect: 1: The value written to EP3_IN_TOGGLE_VALUE is effective.
14	EP3_IN_TOGGLE_VALUE	R/W	0x0	IN operation toggle control bit: 1: Data 1 0: Data 0 Writing this bit will change the comparison register value, reading this bit returns the current IN toggle value.
13	EP3_SEND_STALL_DONE	R	0x0	EP3 STALL transmission complete flag: 1: STALL transmission completed 0: STALL transmission not completed Writing 1 clears this bit.
12	EP3_SEND_STALL	R/W	0x0	STALL transmission control bit: 1: STALL transmitted 0: STALL not transmitted A single write operation transmits STALL multiple times until this bit is cleared.
11	EP3_RECEIVED_DONE	W	0x0	Reception complete control bit: 1: reception completed 0: reception not completed Writing 1 to this bit will set the FIFO to ready state, and it is necessary to write a 1 to this bit each time the data is received and read,

Bit	Name	Attribute	Reset Value	Description
				otherwise, the device will respond with NAK during the next OUT/SETUP operation.
10	EP3_DATA_START	W	0x0	Transmit START: 1: transmit the data in FIFO 0: no operation
9	EP3_FIFOCLR	W	0x0	EP3 FIFO pointer reset control bit: 1: reset FIFO pointer 0: do not reset FIFO pointer
8	EP3_EN	R/W	0x1	EP3 enable: 1: enabled 0: disabled
7:0	EP3_RECEIVED_BYTE	R	0x0	Number of data bytes received by EP3: This value must be less than the maximum packet length.

34.4.6 EP4 Transfer Control Register (USB_EP4CSR)

Offset address: 0x14

Reset value: 0x0000 0100

Bit	Name	Attribute	Reset Value	Description
31:25	RSV	-	-	Reserved
24	EP4_RECEIVED_ACK	R	0x0	Flag of EP4 receiving the ACK handshake packet from the host: 1: handshake packet received 0: handshake packet not received This bit is read-only and can be cleared by USB reset or writing 1 to EP4_DATA_START.
23:20	-	-	-	-
19	EP4_OUT_TOGGLE_STATE	R	0x0	EP4 out/setup status error flag: 1: toggle received not as expected 0: toggle is normal
18	EP4_OUT_TOGGLE_CTRL_En	W	0x0	Enable the value in EP4_OUT_TOGGLE_WANT to take effect: 1: The value written to

Bit	Name	Attribute	Reset Value	Description
				EP4_OUT_TOGGLE_WANT is effective.
17	EP4_OUT_TOGGLE_WANT	R/W	0x0	Comparison value of toggle for EP4 out/setup packets: 1: Data 1 0: Data 0
16	EP4_OUT_TOGGLE_VALUE	R	0x0	Toggle value of packets received by EP4: 1: Data 1 0: Data 0
15	EP4_IN_TOGGLE_CTRL_EN	W	0x0	Enable the value in EP4_IN_TOGGLE_VALUE to take effect: 1: The value written to EP4_IN_TOGGLE_VALUE is effective.
14	EP4_IN_TOGGLE_VALUE	R/W	0x0	IN operation toggle control bit: 1: Data 1 0: Data 0 Writing this bit will change the comparison register value, reading this bit returns the current IN toggle value.
13	EP4_SEND_STALL_DONE	R	0x0	EP4 STALL transmission complete flag: 1: STALL transmission completed 0: STALL transmission not completed Writing 1 clears this bit.
12	EP4_SEND_STALL	R/W	0x0	STALL transmission control bit: 1: STALL transmitted 0: STALL not transmitted A single write operation transmits STALL multiple times until this bit is cleared.
11	EP4_RECEIVED_DONE	W	0x0	Reception complete control bit: 1: reception completed 0: reception not completed Writing 1 to this bit will set the FIFO to ready state, and it is necessary to write a 1 to this bit each time the data is received and read, otherwise, the device will respond with NAK during the next OUT/SETUP operation.
10	EP4_DATA_STA	W	0x0	Transmit START:

Bit	Name	Attribute	Reset Value	Description
	RT			1: transmit the data in FIFO 0: no operation
9	EP4_FIFOCLR	W	0x0	EP4 FIFO pointer reset control bit: 1: reset FIFO pointer 0: do not reset FIFO pointer
8	EP4_EN	R/W	0x1	EP4 enable: 1: enabled 0: disabled
7:0	EP4_RECEIVED_BYTE	R	0x0	Number of data bytes received by EP4: This value must be less than the maximum packet length.

34.4.7 USB Address Register (USB_ADDR)

Offset address: 0x18

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:7	RSV	-	-	Reserved
6:0	USB_ADDR	R	0x0	USB address register (read-only)

34.4.8 SETUP Data Packet Register (USB_SETUP03DATA)

Offset address: 0x1C

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:0	SETUP_0_3_DATA	R	0x0	SETUP data packet byte 0−byte 3 register

34.4.9 SETUP Data Packet Register (USB_SETUP47DATA)

Offset address: 0x20

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:0	SETUP_4_7_DATA	R	0x0	SETUP data packet byte 4~byte 7 register

34.4.10 Endpoint Address Configuration Register (USB_EPADDR)

Offset address: 0x24

Reset value: 0x0000 4321

Bit	Name	Attribute	Reset Value	Description
31:16	RSV	-	-	Reserved
15:12	EP4_ADDR	R/W	0x4	EP4 address configuration
11:8	EP3_ADDR	R/W	0x3	EP3 address configuration
7:4	EP2_ADDR	R/W	0x2	EP2 address configuration
3:0	EP1_ADDR	R/W	0x1	EP1 address configuration

34.4.11 Bus Packet PID Register (USB_CURRENTPID)

Offset address: 0x28

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:4	RSV	-	-	Reserved
3:0	CURRENT_PID	R	0x0	Current PID value of the received USB packet

34.4.12 Frame Number Register (USB_CURRENTFRAMENUMBER)

Offset address: 0x2C

Reset value: 0x0000 003F

Bit	Name	Attribute	Reset Value	Description
31:11	RSV	-	-	Reserved
10:0	CURRENT_FRAME_NUMBER	R	0x3f	Current frame number

34.4.13 CRC Error Counter Register (USB_CRCERRORCNT)

Offset address: 0x30

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:8	RSV	-	-	Reserved
7:0	CRC_ERROR_CNT	R	0x0	Number of CRC error packets, reset during USB reset

34.4.14 Detection Time Register (USB_STATUSDETECTCNT)

Offset address: 0x34

Reset value: 0x0000 00FF

Bit	Name	Attribute	Reset Value	Description
31:8	RSV	-	-	Reserved
7:0	USB_STATUS_DETECT_CNT	R/W	0xff	Reset/resume/suspend detection threshold setting: each unit is 2.5 μs (120 48M clock cycles) 0x0: The threshold is 2.5 μs. 0x1: The threshold is 5.0 μs. …..

34.4.15 EP0 Transmit Data Number Register (USB_EP0SENDBN)

Offset address: 0x40

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:8	RSV	-	-	Reserved
7:0	EP0_SEND_BYTE	R/W	0x0	EP0 transmit data byte number register

34.4.16 EP1 Transmit Data Number Register (USB_EP1SENDBN)

Offset address: 0x44

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:8	RSV	-	-	Reserved
7:0	EP1_SEND_BYTE	R/W	0x0	EP1 transmit data byte number register

34.4.17 EP2 Transmit Data Number Register (USB_EP2SENDBN)

Offset address: 0x48

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:8	RSV	-	-	Reserved
7:0	EP2_SEND_BYTE	R/W	0x0	EP2 transmit data byte number register

34.4.18 EP3 Transmit Data Number Register (USB_EP3SENDBN)

Offset address: 0x4C

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:8	RSV	-	-	Reserved
7:0	EP3_SEND_BYTE	R/W	0x0	EP3 transmit data byte number register

34.4.19 EP4 Transmit Data Number Register (USB_EP4SENDBN)

Offset address: 0x50

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:8	RSV	-	-	Reserved
7:0	EP4_SEND_BYTE	R/W	0x0	EP4 transmit data byte number register

34.4.20 EP0 FIFO Access Entry (USB_EP0FIFO)

Offset address: 0x100

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:0	EP0FIFO	R/W	0x0	EP0 FIFO entry address, only 32-bit access is supported.

34.4.21 EP1 FIFO Access Entry (USB_EP1FIFO)

Offset address: 0x104

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:0	EP1FIFO	R/W	0x0	EP1 FIFO entry address, only 32-bit access is supported.

34.4.22 EP2 FIFO Access Entry (USB_EP2FIFO)

Offset address: 0x108

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:0	EP2FIFO	R/W	0x0	EP2 FIFO entry address, only 32-bit access is supported.

34.4.23 EP3 FIFO Access Entry (USB_EP3FIFO)

Offset address: 0x10C

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:0	EP3FIFO	R/W	0x0	EP3 FIFO entry address, only 32-bit access is supported.

34.4.24 EP4 FIFO Access Entry (USB_EP4FIFO)

Offset address: 0x110

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:0	EP4FIFO	R/W	0x0	EP4 FIFO entry address, only 32-bit access is supported.

34.4.25 EP0 Memory Access Entry (USB_EP0MEM)

Offset address: 0x200 + 4 * N, N = 0–15

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:24	EP0_DATA3	R/W	0x0	EP0 memory supports 8-bit, 16-bit and 32-bit accesses
23:16	EP0_DATA2	R/W	0x0	
15:8	EP0_DATA1	R/W	0x0	
7:0	EP0_DATA0	R/W	0x0	

34.4.26 EP1 Memory Access Entry (USB_EP1MEM)

Offset address: 0x240 + 4 * N, N = 0–15

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:24	EP1_DATA3	R/W	0x0	EP1 memory supports 8-bit, 16-bit and 32-bit accesses
23:16	EP1_DATA2	R/W	0x0	
15:8	EP1_DATA1	R/W	0x0	
7:0	EP1_DATA0	R/W	0x0	

34.4.27 EP2 Memory Access Entry (USB_EP2MEM)

Offset address: 0x280 + 4 * N, N = 0–15

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:24	EP2_DATA3	R/W	0x0	EP2 memory supports 8-bit, 16-bit and 32-bit accesses
23:16	EP2_DATA2	R/W	0x0	
15:8	EP2_DATA1	R/W	0x0	
7:0	EP2_DATA0	R/W	0x0	

34.4.28 EP3 Memory Access Entry (USB_EP3MEM)

Offset address: 0x2C0 + 4 * N, N = 0–15

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:24	EP3_DATA3	R/W	0x0	EP3 memory supports 8-bit, 16-bit and 32-bit accesses
23:16	EP3_DATA2	R/W	0x0	
15:8	EP3_DATA1	R/W	0x0	
7:0	EP3_DATA0	R/W	0x0	

34.4.29 EP4 Memory Access Entry (USB_EP4MEM)

Offset address: 0x300 + 4 * N, N = 0–15

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:24	EP4_DATA3	R/W	0x0	EP4 memory supports 8-bit, 16-bit and 32-bit accesses
23:16	EP4_DATA2	R/W	0x0	
15:8	EP4_DATA1	R/W	0x0	
7:0	EP4_DATA0	R/W	0x0	

34.4.30 Status Register (USB_INTSTATRAW)

Offset address: 0xFFE4

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31	TOGGLE_STATE_ERR_RAW	R	0x0	When a toggle error occurs during IN/OUT/setup operations, this status bit is set to 1.
30	NOEOP_ERR_RAW	R	0x0	If the length of the token packet received by the device exceeds the specified value in the protocol, or if the data packet exceeds (64 + 8) bytes, this status bit will be set to 1.
29	EP4_IN_HANDSHAKE_ERR_RAW	R	0x0	For EP4 IN operations, if the host does not successfully return an ACK signal, this bit will be set to 1.
28	EP3_IN_HANDSHAKE_ERR_RAW	R	0x0	For EP3 IN operations, if the host does not successfully return an ACK signal, this bit will be set to 1.
27	EP2_IN_HANDSHAKE_ERR_RAW	R	0x0	For EP2 IN operations, if the host does not successfully return an ACK signal, this bit will be set to 1.
26	EP1_IN_HANDSHAKE_ERR_RAW	R	0x0	For EP1 IN operations, if the host does not successfully return an ACK signal, this bit will be set to 1.
25	EP0_IN_HANDSHAKE_ERR_RAW	R	0x0	For EP0 IN operations, if the host does not successfully return an ACK signal, this bit will be set to 1.
24	DATA_BYTE_MORETHAN_64_RAW	R	0x0	If the length of the received data packet exceeds 64 bytes, this bit will be set to 1.
23	CRC_ERR_RAW	R	0x0	If there is a CRC error in the received token packet or data packet, this bit will be set to 1.
22	SETADDR_RAW	R	0x0	When the host successfully sets the USB device address, this bit will be set to 1.

Bit	Name	Attribute	Reset Value	Description
21	TURNAROUND_ERROR_RAW	R	0x0	A timeout interrupt occurs when the host replies with an ACK packet.
20	EP4_ACK_RAW	R	0x0	EP4 ACK status: When transmitting or receiving an ACK packet, this bit will be set to 1.
19	EP4_OUT_RAW	R	0x0	EP4 OUT interrupt: When the valid data enters the FIFO, this bit will be set to 1.
18	EP4_IN_RAW	R	0x0	When EP4 receives an IN token packet, this bit will be set to 1.
17	EP3_ACK_RAW	R	0x0	EP3 ACK status: When transmitting or receiving an ACK packet, this bit will be set to 1.
16	EP3_OUT_RAW	R	0x0	EP3 OUT interrupt: When the valid data enters the FIFO, this bit will be set to 1.
15	EP3_IN_RAW	R	0x0	When EP3 receives an IN token packet, this bit will be set to 1.
14	EP2_ACK_RAW	R	0x0	EP2 ACK status: When transmitting or receiving an ACK packet, this bit will be set to 1.
13	EP2_OUT_RAW	R	0x0	EP2 OUT interrupt: When the valid data enters the FIFO, this bit will be set to 1.
12	EP2_IN_RAW	R	0x0	When EP2 receives an IN token packet, this bit will be set to 1.
11	EP1_ACK_RAW	R	0x0	EP1 ACK status: When transmitting or receiving an ACK packet, this bit will be set to 1.
10	EP1_OUT_RAW	R	0x0	EP1 OUT interrupt: When the valid data enters the FIFO, this bit will be set to 1.
9	EP1_IN_RAW	R	0x0	When EP1 receives an IN token packet, this bit will be set to 1.
8	EP0_ACK_RAW	R	0x0	EP0 ACK status: When transmitting or receiving an ACK packet, this bit

Bit	Name	Attribute	Reset Value	Description
				will be set to 1.
7	EP0_OUT_RAW	R	0x0	EP0 OUT interrupt: When the valid data enters the FIFO, this bit will be set to 1.
6	EP0_IN_RAW	R	0x0	When EP0 receives an IN token packet, this bit will be set to 1.
5	SUDAV_RAW	R	0x0	When a setup data packet is received, this bit will be set to 1.
4	SETUPTOK_RAW	R	0x0	When a setup token packet is received, this bit will be set to 1.
3	SOF_RAW	R	0x0	When a SOF packet is received, this bit will be set to 1.
2	RESUME_RAW	R	0x0	When the host resumes, this bit will be set to 1.
1	SUSPEND_RAW	R	0x0	When the host suspends, this bit will be set to 1.
0	BUS_RESET_RAW	R	0x0	When the host resets, this bit will be set to 1.

34.4.31 Interrupt Enable Register (USB_INTEN)

Offset address: 0xffe8

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31	TOGGLE_STATE_ERR_EN	R/W	0x0	TOGGLE_STATE_ERR interrupt enable
30	NOEOP_ERR_EN	R/W	0x0	NOEOP_ERR interrupt enable
29	EP4_IN_HANDSHAKE_ERR_EN	R/W	0x0	EP4_IN_HANDSHAKE_ERR interrupt enable
28	EP3_IN_HANDSHAKE_ERR_EN	R/W	0x0	EP3_IN_HANDSHAKE_ERR interrupt enable
27	EP2_IN_HANDSHAKE_ERR_EN	R/W	0x0	EP2_IN_HANDSHAKE_ERR interrupt enable
26	EP1_IN_HANDSHAKE_ERR_EN	R/W	0x0	EP1_IN_HANDSHAKE_ERR interrupt enable

Bit	Name	Attribute	Reset Value	Description
25	EP0_IN_HANDSHAKE_ERR_EN	R/W	0x0	EP0_IN_HANDSHAKE _ERR interrupt enable
24	DATA_BYTE_MORETHAN_64_EN	R/W	0x0	Enable interrupt for DATA packet length exceeding 64 bytes
23	CRC_ERR_EN	R/W	0x0	CRC error interrupt enable
22	SETADDR_EN	R/W	0x0	Enable interrupt when the host sets the USB device address successfully
21	TURNAROUND_ERROR_EN	R/W	0x0	Enable interrupt for host ACK packet timeout
20	EP4_ACK_EN	R/W	0x0	Enable interrupt for EP4 ACK status: transmitting or receiving ACK packets
19	EP4_OUT_EN	R/W	0x0	EP4 OUT interrupt is triggered when the valid data enters the FIFO.
18	EP4_IN_EN	R/W	0x0	Enable interrupt for EP4 receiving an IN token packet
17	EP3_ACK_EN	R/W	0x0	Enable interrupt for EP3 ACK status: transmitting or receiving ACK packets
16	EP3_OUT_EN	R/W	0x0	EP3 OUT interrupt is triggered when the valid data enters the FIFO.
15	EP3_IN_EN	R/W	0x0	Enable interrupt for EP3 receiving an IN token packet
14	EP2_ACK_EN	R/W	0x0	Enable interrupt for EP2 ACK status: transmitting or receiving ACK packets
13	EP2_OUT_EN	R/W	0x0	EP2 OUT interrupt is triggered when the valid data enters the FIFO.
12	EP2_IN_EN	R/W	0x0	Enable interrupt for EP2 receiving an IN token packet
11	EP1_ACK_EN	R/W	0x0	Enable interrupt for EP1 ACK status: transmitting or receiving ACK packets

Bit	Name	Attribute	Reset Value	Description
10	EP1_OUT_EN	R/W	0x0	EP1 OUT interrupt is triggered when the valid data enters the FIFO.
9	EP1_IN_EN	R/W	0x0	Enable interrupt for EP1 receiving an IN token packet
8	EP0_ACK_EN	R/W	0x0	Enable interrupt for EP0 ACK status: transmitting or receiving ACK packets
7	EP0_OUT_EN	R/W	0x0	EP0 OUT interrupt is triggered when the valid data enters the FIFO.
6	EP0_IN_EN	R/W	0x0	Enable interrupt for EP0 receiving an IN token packet
5	SUDAV_EN	R/W	0x0	Enable interrupt for receiving a setup data packet
4	SETUPTOK_EN	R/W	0x0	Enable interrupt for receiving a setup token packet
3	SOF_EN	R/W	0x0	Enable interrupt for receiving a SOF packet
2	RESUME_EN	R/W	0x0	Host resume interrupt enable
1	SUSPEND_EN	R/W	0x0	Host suspend interrupt enable
0	BUS_RESET_EN	R/W	0x0	Host reset interrupt enable

34.4.32 Interrupt Clear Register (USB_INTCLR)

Offset address: 0xFFF0

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31	TOGGLE_STATE_ERR_CLR	R/W	0x0	TOGGLE_STATE_ERR _RAW interrupt register clear bit: 1: clear TOGGLE_STATE_ERR_RAW to 0 0: keep TOGGLE_STATE_ERR_RAW unchanged
30	NOEOP_ERR_CLR	R/W	0x0	NOEOP_ERR_RAW interrupt register clear bit: 1: clear NOEOP_ERR_RAW to 0

Bit	Name	Attribute	Reset Value	Description
				0: keep NOEOP_ERR_RAW unchanged
29	EP4_IN_HANDSHAKE_ERR_CLR	R/W	0x0	EP4_IN_HANDSHAKE_ERR_RAW interrupt register clear bit: 1: clear EP4_IN_HANDSHAKE_ERR_RAW to 0 0: keep EP4_IN_HANDSHAKE_ERR_RAW unchanged
28	EP3_IN_HANDSHAKE_ERR_CLR	R/W	0x0	EP3_IN_HANDSHAKE_ERR_RAW interrupt register clear bit: 1: clear EP3_IN_HANDSHAKE_ERR_RAW to 0 0: keep EP3_IN_HANDSHAKE_ERR_RAW unchanged
27	EP2_IN_HANDSHAKE_ERR_CLR	R/W	0x0	EP2_IN_HANDSHAKE_ERR_RAW interrupt register clear bit: 1: clear EP2_IN_HANDSHAKE_ERR_RAW to 0 0: keep EP2_IN_HANDSHAKE_ERR_RAW unchanged
26	EP1_IN_HANDSHAKE_ERR_CLR	R/W	0x0	EP1_IN_HANDSHAKE_ERR_RAW interrupt register clear bit: 1: clear EP1_IN_HANDSHAKE_ERR_RAW to 0 0: keep EP1_IN_HANDSHAKE_ERR_RAW unchanged
25	EP0_IN_HANDSHAKE_ERR_CLR	R/W	0x0	EP0_IN_HANDSHAKE_ERR_RAW interrupt register clear bit: 1: clear EP0_IN_HANDSHAKE_ERR_RAW to 0 0: keep EP0_IN_HANDSHAKE_ERR_RAW unchanged
24	DATA_BYTE_MORETHAN_64_CLR	R/W	0x0	DATA_BYTE_MORETHAN_64_RAW interrupt register clear bit: 1: clear DATA_BYTE_MORETHAN_64_RAW to 0 0: keep DATA_BYTE_MORETHAN_64_RAW

Bit	Name	Attribute	Reset Value	Description
				unchanged
23	CRC_ERR_CLR	R/W	0x0	CRC_ERR_RAW interrupt register clear bit: 1: clear CRC_ERR_RAW to 0 0: keep CRC_ERR_RAW unchanged
22	SETADDR_CLR	R/W	0x0	SETADDR_RAW interrupt register clear bit: 1: clear SETADDR_RAW to 0 0: keep SETADDR_RAW unchanged
21	TURNAROUND_ERROR_CLR	R/W	0x0	TURNAROUND_ERROR_RAW interrupt register clear bit: 1: clear TURNAROUND_ERROR_RAW to 0 0: keep TURNAROUND_ERROR_RAW unchanged
20	EP4_ACK_CLR	R/W	0x0	EP4_ACK_RAW interrupt register clear bit: 1: clear EP4_ACK_RAW to 0 0: keep EP4_ACK_RAW unchanged
19	EP4_OUT_CLR	R/W	0x0	EP4_OUT_RAW interrupt register clear bit: 1: clear EP4_OUT_RAW to 0 0: keep EP4_OUT_RAW unchanged
18	EP4_IN_CLR	R/W	0x0	EP4_IN_RAW interrupt register clear bit: 1: clear EP4_IN_RAW to 0 0: keep EP4_IN_RAW unchanged
17	EP3_ACK_CLR	R/W	0x0	EP3_ACK_RAW interrupt register clear bit: 1: clear EP3_ACK_RAW to 0 0: keep EP3_ACK_RAW unchanged
16	EP3_OUT_CLR	R/W	0x0	EP3_OUT_RAW interrupt register clear bit: 1: clear EP3_OUT_RAW to 0 0: keep EP3_OUT_RAW unchanged
15	EP3_IN_CLR	R/W	0x0	EP3_IN_RAW interrupt register clear bit: 1: clear EP3_IN_RAW to 0 0: keep EP3_IN_RAW unchanged
14	EP2_ACK_CLR	R/W	0x0	EP2_ACK_RAW interrupt register clear bit: 1: clear EP2_ACK_RAW to 0 0: keep EP2_ACK_RAW unchanged

Bit	Name	Attribute	Reset Value	Description
13	EP2_OUT_CLR	R/W	0x0	EP2_OUT_RAW interrupt register clear bit: 1: clear EP2_OUT_RAW to 0 0: keep EP2_OUT_RAW unchanged
12	EP2_IN_CLR	R/W	0x0	EP2_IN_RAW interrupt register clear bit: 1: clear EP2_IN_RAW to 0 0: keep EP2_IN_RAW unchanged
11	EP1_ACK_CLR	R/W	0x0	EP1_ACK_RAW interrupt register clear bit: 1: clear EP1_ACK_RAW to 0 0: keep EP1_ACK_RAW unchanged
10	EP1_OUT_CLR	R/W	0x0	EP1_OUT_RAW interrupt register clear bit: 1: clear EP1_OUT_RAW to 0 0: keep EP1_OUT_RAW unchanged
9	EP1_IN_CLR	R/W	0x0	EP1_IN_RAW interrupt register clear bit: 1: clear EP1_IN_RAW to 0 0: keep EP1_IN_RAW unchanged
8	EP0_ACK_CLR	R/W	0x0	EP0_ACK_RAW interrupt register clear bit: 1: clear EP0_ACK_RAW to 0 0: keep EP0_ACK_RAW unchanged
7	EP0_OUT_CLR	R/W	0x0	EP0_OUT_RAW interrupt register clear bit: 1: clear EP0_OUT_RAW to 0 0: keep EP0_OUT_RAW unchanged
6	EP0_IN_CLR	R/W	0x0	EP0_IN_RAW interrupt register clear bit: 1: clear EP0_IN_RAW to 0 0: keep EP0_IN_RAW unchanged
5	SUDAV_CLR	R/W	0x0	Receiving setup data packet interrupt register clear bit: 1: clear SUDAV_RAW to 0 0: keep SUDAV_RAW unchanged
4	SETUPTOK_CLR	R/W	0x0	Receiving setup token packet interrupt register clear bit: 1: clear SETUPTOK_RAW to 0 0: keep SETUPTOK_RAW unchanged
3	SOF_CLR	R/W	0x0	Receiving SOF packet interrupt register clear bit:

Bit	Name	Attribute	Reset Value	Description
				1: clear SOF_RAW to 0 0: keep SOF_RAW unchanged
2	RESUME_CLR	R/W	0x0	Host resume interrupt register clear bit: 1: clear RESUME_RAW to 0 0: keep RESUME_RAW unchanged
1	SUSPEND_CLR	R/W	0x0	Host suspend interrupt register clear bit: 1: clear SUSPEND_RAW to 0 0: keep SUSPEND_RAW unchanged
0	BUS_RESET_CLR	R/W	0x0	Host reset interrupt register clear bit: 1: clear BUS_RESET_RAW to 0 0: keep BUS_RESET_RAW unchanged

34.5 Operation Procedure

34.5.1 USB Connection

By default, USB is not connected. After completing the initialization, the USB connection can be established by setting USB_WORKINGMODE[4] and USB_WORKINGMODE[6]. Conversely, the USB connection can be disconnected by clearing these bits.

34.5.2 SETUP Data and EP0 Control Transfer Data

The set address command in control transfers is fully handled by the hardware, and the software can use the last interrupt bit SETADDR if it needs to know that the set address command has occurred.

Each control transfer over USB goes through the SETUP phase, DATA phase (optional), and STATUS phase.

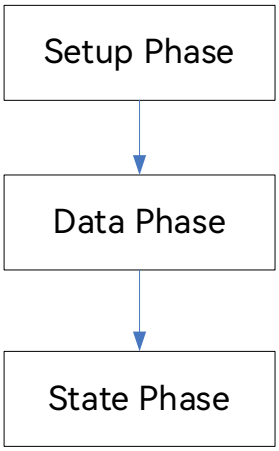


Figure 34-1: USB Transfer Diagram

The STATUS phase is completed with a packet of length 0.

- Setup phase: When the control transfer setup data has been received, a SUDAV_INT interrupt will be generated by the USB, and the received setup data will be stored in the registers USB_SETUP03DATA and USB_SETUP47DATA.
- Data phase: If the direction of the data phase is OUT, then the EP0OUT_INT interrupt will occur when the data transfer is complete, and the results will be stored in the EP0 FIFO, while the data transfer status will be preserved in EP0CSR. If the direction of the data phase is IN, when the host sends an IN token for EP0, the EP0IN_INT interrupt will be triggered, and the data in the EP0 FIFO will be sent back to the host according to the indications in EP0CSR. If the data is not ready, USB will automatically respond with an NAK packet.
- Status phase: The status of the status phase may be sent from the host to the device or from the device back to the host, depending on the direction of the data phase. This requires the software to maintain this status. If it is required to send from the device to the host, the user needs to prepare a 0-length data using the data phase method. USB will send the data back to the host when the EN0IN_INT interrupt occurs, completing the status phase of the entire control transfer.

34.5.3 Endpoint In Transfer

When the host sends the EPx IN token, the EPxIN_INT interrupt will be triggered, and the data in the EPx FIFO will be sent back to the host according to the indications in EPxCSR, including the length of the data transmitted. The flowchart for the endpoint in transfer is shown in the following figure:

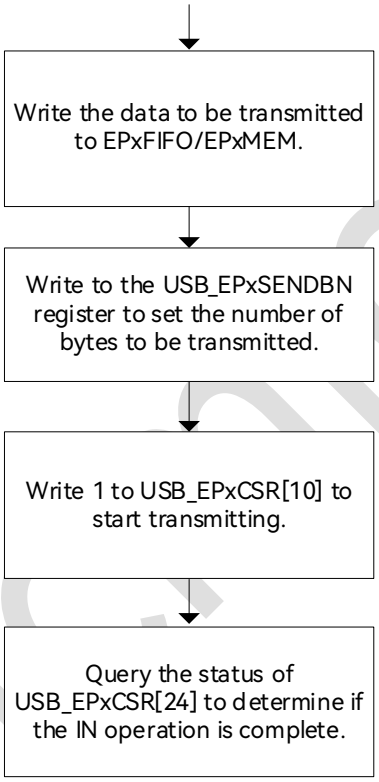


Figure 34-2: USB Endpoint In Transfer Flowchart

34.5.4 Endpoint Out Transfer

When the EPxOUT_INT occurs, the USB_EPxCSR[7:0] field records the length of the received data, according to which the user can read data from EPxFIFO or EPxMEM. After the data from EPxOUT has been read, the user needs to write USB_EPxCSR[11] to indicate that the next data is allowed to be received. The flowchart for the endpoint out transfer is shown in the following figure:

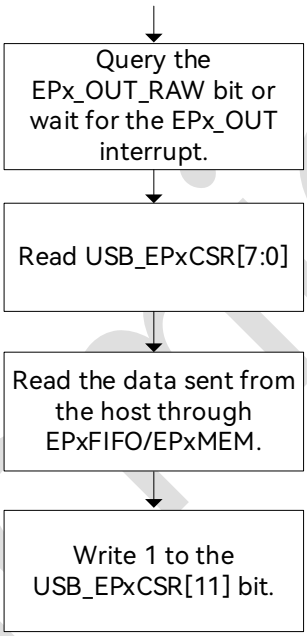


Figure 34-3: USB Endpoint Out Transfer Flowchart

35 Analog-to-digital Converter (ADC)

35.1 Overview

The analog-to-digital converter (ADC) samples the channel voltage, receives the analog-to-digital conversion results, and transmits the simply processed results to the CPU or other modules in the system.

This ADC controller manages a 12-bit successive approximation ADC with 16 external channels, where two adjacent external channels can form a pair of differential input signals. The ADC controller supports regular sequences of up to 16 positions and injection sequences of up to 4 positions, supports dual-ADC cooperative scanning, and features an analog watchdog function. It has a 32-level deep RX FIFO, and each channel has its own independent data register.

35.2 Main Features

- Input voltage range: VREFN ≤ VIN ≤ VREFP, where VREFN is connected to the ground (VSSA) of the analog circuit, and VREFP can be the power supply (VDDA) of the analog circuit or an externally input reference voltage (VREF).
- The two ADCs can be used with the internal operational amplifier (OPA) to amplify the signal before sampling or as a buffer.
- Configurable ADCCLK clock
- Resolution: 12 bits
- Power-on stabilization time: configurable from 32 to 63 ADCCLK cycles

- Input channels: 16 single-ended external input channels are provided, where adjacent pairs of channels can serve as differential external input channels. The regular sequence provides up to 8 positions for channel conversion, each of which can select any channel.
- Trigger mode: software configuring register, or internal timer events (rising edge, falling edge or both edges), or GPIOA interrupts
- Conversion modes: single mode, continuous mode, discontinuous mode, dual-ADC cooperative mode
- Regular and injection sequences: The regular sequence contains up to 20 positions for channel conversion, while the injection sequence contains up to 4 positions for channel conversion. Both groups consist of a sequence of conversions that can be done on any channel and in any order. The injection sequence has higher priority over the regular sequence.
- It is possible to perform continuous multiple samplings (2, 4, 8, 16, 32, 64 or 128 times) on a specific channel or several channels and calculate the average value.
- It is provided with an analog watchdog for monitoring the conversion result.
- Each channel has an independent data register, which can be set uniformly to clear data automatically after reading.
- Each of the two ADC controllers is provided with an RX FIFO that is 32-level deep and 16-bit wide, for storing the conversion results of the regular and injection sequences.
- DMA mode supported

35.3 System Block Diagram

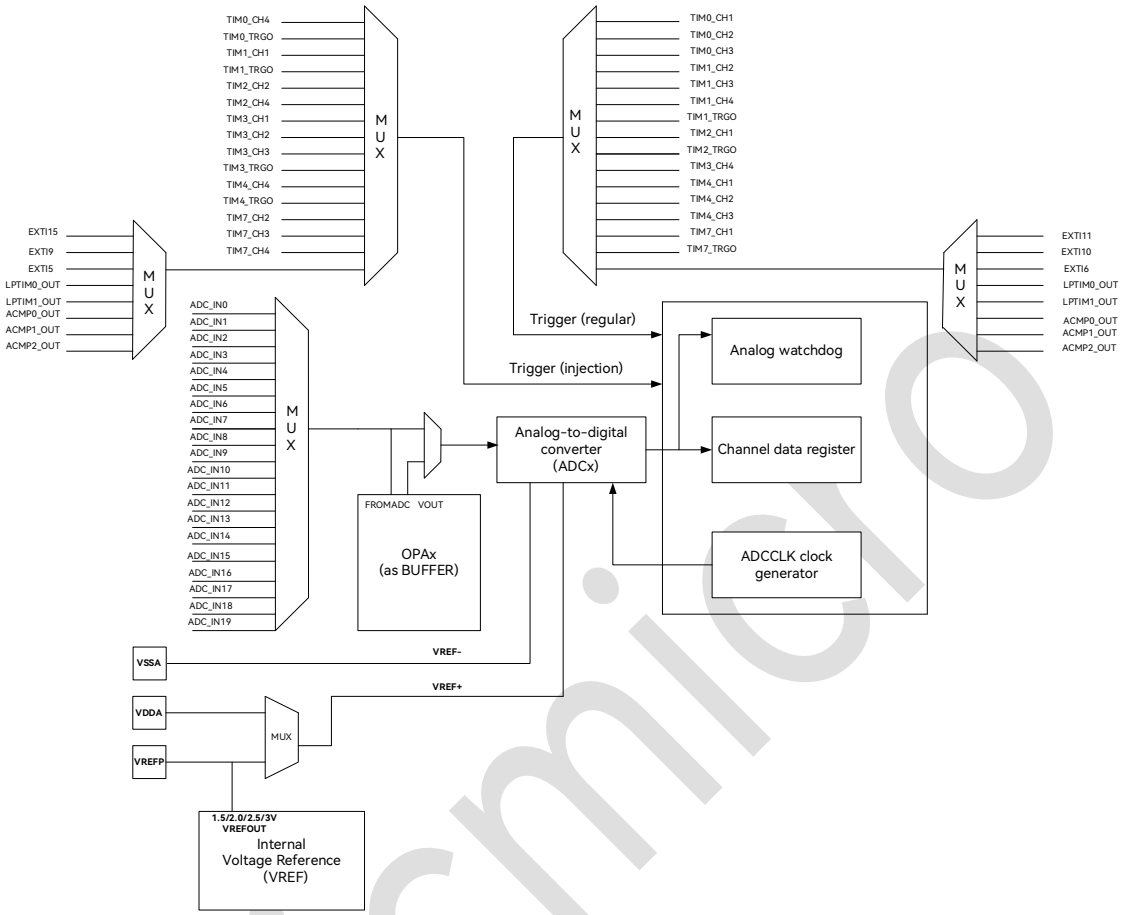


Figure 35-1: ADC System Block Diagram

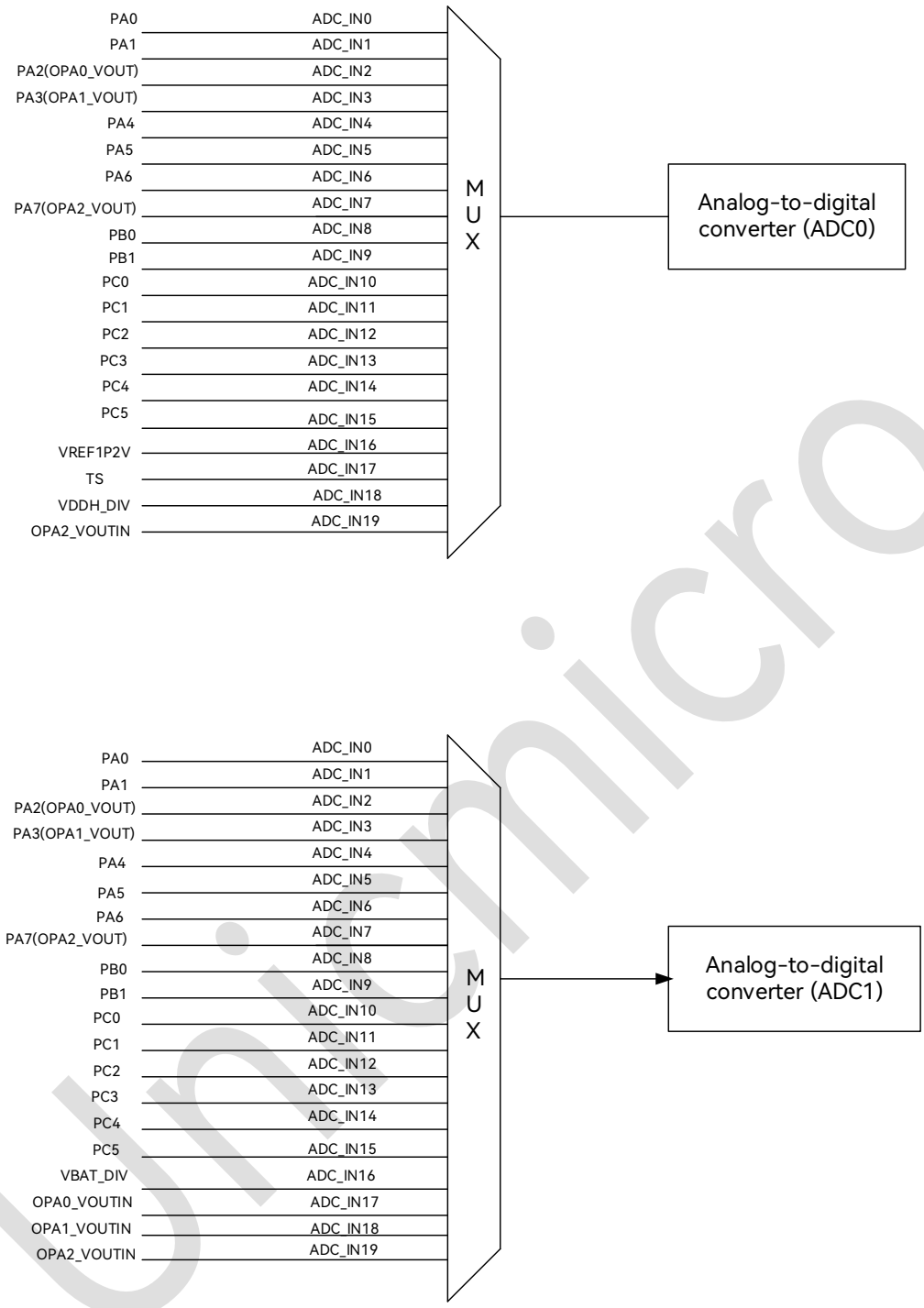


Figure 35-2: ADC Channel Connection Diagram

Notes:

- OPA0 is used as the buffer for ADC0.
- OPA1 is used as the buffer for ADC1.

35.4 Pin Description

Table 35-1: ADC Pin Description

Function Pin	Alternate Function Pin	Direction	Functional Description
ADC_IN0	PA0	AI	ADC analog input channel
ADC_IN1	PA1	AI	ADC analog input channel
ADC_IN2	PA2	AI	ADC analog input channel
ADC_IN3	PA3	AI	ADC analog input channel
ADC_IN4	PA4	AI	ADC analog input channel
ADC_IN5	PA5	AI	ADC analog input channel
ADC_IN6	PA6	AI	ADC analog input channel
ADC_IN7	PA7	AI	ADC analog input channel
ADC_IN8	PB0	AI	ADC analog input channel
ADC_IN9	PB1	AI	ADC analog input channel
ADC_IN10	PC0	AI	ADC analog input channel
ADC_IN11	PC1	AI	ADC analog input channel
ADC_IN12	PC2	AI	ADC analog input channel
ADC_IN13	PC3	AI	ADC analog input channel
ADC_IN14	PC4	AI	ADC analog input channel
ADC_IN15	PC5	AI	ADC analog input channel
VDDA	-	AP	Analog power 3.3 V
VSSA	-	AG	Analog ground
VREFP	-	AP	Analog positive reference voltage

35.5 Functional Description

35.5.1 Analog Watchdog

The analog watchdog can compare the channel data against the upper and lower threshold values. If the channel data exceeds the range defined by these thresholds, the watchdog will generate an interrupt flag. The channels to be compared can be independently enabled, both the upper and lower thresholds can be freely configured, and the comparison conditions can also be selected. There are three interrupt flags: channel data watchdog alarm interrupt flag, regular sequence watchdog alarm interrupt flag, and injection sequence watchdog alarm interrupt flag. The channel data watchdog alarm interrupt flag does not distinguish which

sequence the data belongs to. All three interrupt flags shall be cleared by software writing 1, and will not be cleared by hardware automatically.

Note: In addition to setting up the analog watchdog before starting the conversion, the settings of the analog watchdog can also be modified during the conversion. Whenever the analog watchdog finishes comparing the latest channel data, the 27th bit of the current status register (STAT), the watchdog comparison channel data completion flag (Wdg_cmplt), will be set to 1. At this point, the settings of the analog watchdog can be changed, and the changes will take effect immediately. The Wdg_cmplt flag will be cleared by hardware when software writes to the watchdog comparison condition register, or it can also be cleared by software writing 1 directly to this flag. However, due to the high frequency of channel switching and the relatively long time taken for software to read the register, analyze data, and then write back to the register, there is a risk that the comparison conditions may not be updated in a timely manner.

35.5.2 ADC Clock Generator

The ADC clock generator generates a clock (ADCCLK) specifically for the analog ADC by dividing the APB clock (PCLK) according to the division factor PCLK_DIV set in the CLKCTRL register. The clock used by the ADC controller is the system clock (PCLK).

In dual-ADC cooperative operation mode, whether for regular sequence cooperation or injection sequence cooperation, ADC1 always uses the same clock as ADC0. This means that the clock for ADC1 is selected and generated by the CLKCTRL register of ADC controller 0, while the clock selection and clock division settings in the CLKCTRL register of ADC controller 1 have no effect. When ADC1 operates independently, its clock is still generated by ADC controller 1.

ADCCLK comes from two sources: an ADC clock generated by the internal clock generator of the ADC controller, or an external input clock.

35.5.2.1 Internal Clock Generator

The formula for the internally generated ADC clock frequency is as follows:

$$f_{ADCCLK} = f_{PCLK}/(Pclk\_div + 1)$$

Wherein, f_{ADCCLK} is the internal ADC clock frequency, f_{PCLK} is the APB clock frequency, and Pclk_div is the frequency division factor stored in the CLKCTRL register, with a valid range of 0 ≤ Pclk_div ≤ 65535.

In terms of the timing logic for the ADC controller itself, the fastest division frequency allowed is 1 for the ADC single mode or the regular sequence delayed continuous scanning mode. For the dual-ADC collaborative mode, the fastest division frequency allowed is 2. When ADCC is in single scanning or delayed continuous scanning mode, the ADCCLK is allowed to operate at a maximum of 1 division frequency. In parallel scanning, delayed single scanning, or delayed intermittent scanning modes, the ADCCLK is allowed to operate at a maximum of 2 division frequency.

Note: The analog ADC supports a maximum sampling rate of 5.25 MSPS.

35.5.2.2 External Clock Synchronization Circuit

The external input ADC clock is first synchronized by the PCLK clock domain (sampled by PCLK) before being input to the analog ADC. Therefore, the frequency of the external input ADC clock cannot exceed half of the PCLK frequency, i.e.:

$$f_{ADCCLK} \leq f_{PCLK} / 2$$

35.5.3 Data Receiver

The data receiver is responsible for receiving the conversion results from the ADC, calculating the sum and average, and passing the average value to the register and FIFO.

35.5.4 RX FIFO

The RX FIFO has a depth of 32 levels (capable of storing 32 data entries), with each entry consisting of a 4-bit channel number and a 12-bit conversion result, as shown in the table below. The data in the RX FIFO is read from the register logic module and transmitted to the CPU or DMA.

Table 35-2: Data Format in RX FIFO

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Low 4 bits of channel number				Conversion result											

When the RX FIFO has received enough data (as specified by the watermark in the DGCTRL register), the RX FIFO data available interrupt flag (FIFO_AVL) will be set to 1; if DMA transfer is enabled, a DMA request will also be issued.

If the RX FIFO is full, new data cannot be written.

If an overflow occurs in the RX FIFO, the RX FIFO overflow interrupt flag (FIFO_OVF) will be set to 1.

35.6 Conversion Mode Description

35.6.1 Overview of Conversion Sequence

The ADC controller has regular sequence and injection sequence. The regular sequence contains up to 20 positions for channel conversion, while the injection sequence contains up to 4 positions for channel conversion. Both groups consist of a sequence of conversions that can be done on any channel and in any order.

The regular trigger signal can be received when the ADC is in the stabilization state or idle state; the injection trigger signal can be received when the ADC is in the stabilization state, idle state or regular state. If the ADC receives a regular or injection trigger signal while in a

stabilization state, it will begin the conversion only after stabilization. If the ADC receives a regular or injection trigger signal while in an idle state, it will start the conversion immediately.

The injection sequence always takes priority over the regular sequence. However, new incoming injection trigger signals cannot interrupt a conversion that is currently ongoing for an injection sequence. If an injection sequence conversion is triggered while a regular sequence is being converted, it must be checked whether the number of ADC_SOC signal occurrences for the interrupted regular sequence channel equals the number of conversions set for that channel (i.e., whether the interrupted channel is on its last conversion). If the set number of conversions has been reached, the interrupted channel can proceed with the conversion and write data to the data register or FIFO. If the set number of conversions has not been reached, the ADC will wait for the most recent conversion to complete, discard the data, and then start the injection sequence conversion. If an injection sequence is currently converting and a regular trigger signal or a new injection trigger signal arrives, the new trigger signal will be ignored, and the ongoing injection sequence conversion will continue. If both a regular sequence conversion and an injection sequence conversion are triggered within the same PCLK clock cycle while in an idle state, the regular trigger signal will be ignored and the injection sequence will start converting. If a regular sequence conversion is interrupted by an injection sequence, after the injection sequence conversion ends, the ADC will require 16 ADCCLK cycles (during which the ADC_SOC signal will not be sent) to resume the regular sequence conversion according to the selection of the RGL_RCV_SEL bit in the DGCTRL register. It will either continue from the interrupted position or restart from the first position.

For independently operating ADC, there are three conversion modes for the regular sequence: single scan mode, continuous scan mode and intermittent scan mode, and only one conversion mode for the injection sequence: single-trigger scan.

For cooperatively operating ADC, there are three cooperative modes for the regular sequence:

independent scan mode, synchronous scan mode, and delayed scan mode; and three cooperative modes for the injection sequence: independent scan mode, parallel scan mode, and alternating trigger scan mode. The options for cooperative operation mode shall be used in combination with the options for independent operation mode, for example, adopting the mode of “synchronous continuous scan for regular sequence + alternating trigger scan for injection sequence”.

Specifically, if the cooperative mode of the regular sequence is set to synchronous scan mode, the ADC with “short” sequence will wait for the ADC with “long” sequence to ensure that each round of regular sequence conversions starts synchronously. Since the number of positions in the regular sequences of the two ADCs may differ, and the sampling times of each channel (multiple samples averaged) may also differ, it is possible for the regular sequence conversion times of the two ADCs to vary. Similarly, the conversion times of the injection sequences may also differ.

It is worth noting that the parallel conversion mode of the injection sequence does not guarantee a synchronous start, but it ensures synchronization upon completion through mutual waiting. If the regular sequence operates in independent scan mode, the clock frequency, phase and the time point for starting the regular sequence scanning of the two ADCs can be different. When a start signal of injection sequence conversion arrives, the two ADCs will immediately start the injection sequence conversion after completing the current ongoing regular sequence conversion lasting 16 ADCCLK cycles, without waiting. Therefore, the start time points of the injection sequence conversion are likely to be different. However, when both ADCs complete the injection sequence conversion, they need to wait for each other to finish the injection sequence conversion before the state machine can return to the state it was in prior to entering the injection state (which could be the regular state or idle state).

In summary, if the cooperative mode of the regular sequence is set to synchronous scan mode, there are two scenarios of "waiting": one where an ADC that has completed the regular sequence conversion waits for another ADC that is still converting the regular sequence; and another where an ADC that has completed the injection sequence conversion waits for another ADC that is still converting the injection sequence, and then they synchronously resume the regular sequence conversion. Note that the ADC in the waiting state is not idle, i.e., the idle state flag remains 0.

35.6.2 Basic Conversion Mode of Regular Sequence

The basic conversion modes for regular sequences include single scan mode, continuous scan mode and intermittent scan mode, which can be used in combination with the dual-ADC cooperative method.

35.6.2.1 Single Scan Mode for Regular Sequence

In single scan mode, when the regular sequence conversion is triggered, the regular sequence is only converted once and then returns to the idle state.

35.6.2.2 Continuous Scan Mode for Regular Sequence

In continuous scan mode, when the regular sequence conversion is triggered, the regular sequence is converted continuously. If it is necessary to stop the conversion of the regular sequence, then the sampling controller and data receiver of the ADC controller can be briefly reset by software writing a 1 to bit 2 Adc_stop of the analog circuit control register (ANCTRL), allowing the state machine to return to the stabilization state.

35.6.2.3 Intermittent Scan Mode for Regular Sequence

In intermittent scan mode, each time the regular sequence conversion is triggered, it will convert the channels corresponding to (Short_leng + 1) positions in the regular sequence, and

then return to the idle state. If there are less than (Short_leng + 1) unconverted positions remaining in the regular sequence, all the remaining unconverted positions will be converted before returning to the idle state. For example, if there are 8 positions in the regular sequence and 3 positions in the regular short sequence, then the first trigger will convert the channels corresponding to positions 0, 1 and 2, the second trigger will convert the channels corresponding to positions 3, 4, and 5, and the third trigger will convert the channels corresponding to positions 6 and 7.

35.6.3 Basic Conversion Mode of Injection Sequence

There is only one basic conversion mode for injection sequences, the single scan mode. In single scan mode, when the injection sequence conversion is triggered, the injection sequence is only converted once and then returns to the idle state.

35.6.4 Dual-ADC Cooperative Mode for Regular Sequence

The dual-ADC cooperative mode for regular sequences includes synchronous scan mode (also known as simultaneous mode or parallel mode) and delayed scan mode (also known as alternating mode or following mode). By default, the two ADCs operate independently.

When one of the dual-ADC cooperative modes is selected for the regular sequence, both ADC controllers will use the regular trigger signal from ADC controller 0, and both ADCs will utilize the clock generated by ADC controller 0.

35.6.4.1 Synchronous Scan Mode for Dual-ADC Regular Sequence

In synchronous scan mode for regular sequences, if both ADCs are in the idle state, both ADCs will start synchronous conversion when ADC controller 0 receives the regular trigger signal.

If the regular sequences or (in intermittent scan mode) the regular short sequences of the two ADC controllers do not finish at the same time, the two ADC controllers will wait for each other

to finish the current sequence to ensure that the next conversion of the regular sequence or regular short sequence still starts synchronously.

Note: If both ADCs are not idle at the arrival of the regular trigger signal from ADC controller 0, the two ADCs will start the first round of conversions sequentially, and the synchronous start of conversion can only be realized after a round of waiting process.

35.6.4.2 Delayed Scan Mode for Dual-ADC Regular Sequence

In delayed scan mode for regular sequences, when ADC controller 0 receives the regular trigger signal, ADC0 will start conversion first. After a delay of 6−21 ADCCLK cycles, ADC1 will then start conversion. If the delay time does not exceed 10 ADCCLK cycles, then the conversion of ADC1 appears to be delayed relative to the conversion of ADC0.

If the delay time exceeds 10 ADCCLK cycles, then, apart from the first conversion of ADC0, subsequent start of conversion for its own ADC will have a delay relative to the start of conversion of the other ADC.

In addition, the delayed continuous scan mode for regular sequences is the only cooperative mode that does not involve a waiting process at the end of the regular sequence conversion, whereas the delayed single or intermittent scan modes for regular sequences still have an end waiting process.

35.6.5 Dual-ADC Cooperative Mode for Injection Sequence

The dual-ADC cooperative mode for injection sequences includes parallel scan mode and alternating trigger scan mode. By default, the two ADCs operate independently.

When one of the dual-ADC cooperative modes is selected for the injection sequence, both ADC controllers will use the injection trigger signal from ADC controller 0, and both ADCs will utilize the clock generated by ADC controller 0.

35.6.5.1 Parallel Scan Mode for Dual-ADC Injection Sequence

In the injection sequence parallel scan mode, when ADC controller 0 receives the injection trigger signal, if both ADCs are performing regular conversions, they will each start the injection parallel conversion after completing the most recent regular conversion; if both ADCs are idle, they will directly enter the injection state and start the injection parallel conversion.

Suppose that in the partial process of independent continuous conversion of regular sequence and parallel conversion of injection sequence, because the regular sequence conversions of the two ADCs are independent, their ADCCLK counters may step differently, resulting in different starting times for the parallel conversion of injection sequence. However, they will still have a waiting process at the end of the injection sequence conversion. The situation is similar for delayed continuous conversion of regular sequence and parallel conversion of injection sequence.

However, if the regular sequence is synchronously converted and the injection sequence is converted in parallel, the ADCCLK counters of the two ADCs step in unison, leading to the parallel conversion of injection sequence starting at the same time.

35.6.5.2 Alternating Trigger Scan Mode for Dual-ADC Injection Sequence

In the alternating trigger scan mode for injection sequence, when ADC controller 0 receives the injection trigger signal, only one of the two ADCs will start the injection sequence conversion, while the other will enter the injection state and wait, and then both ADCs will simultaneously resume their previous states. The 1st, 3rd, 5th, … (odd-numbered) injection trigger signals will cause ADC0 to convert the injection sequence; the 2nd, 4th, 6th, … (even-numbered) injection trigger signals will cause ADC1 to convert the injection sequence.

Note: In alternating trigger mode, the injection conversion of the next ADC can only be triggered after the previous ADC has completed its injection conversion. If a new trigger signal

arrives while the previous ADC injection conversion is not yet completed, the incoming trigger signal will be ignored.

35.7 Temperature Measurement

The temperature inside the chip is measured by the built-in temperature sensor. The accurate temperature coefficient of the chip is determined by the measured values at 25℃ and 85℃.

In general, the temperature value can be obtained using the following calculation formula:

$T = (V_{REF} * (A2-A1) / 4095) / K + 25℃$

Notes:

T: temperature to be measured

V_{REF}: reference voltage (typical 3.3 V)

K: temperature coefficient (-1.9 mV/℃)

A2: ADC value corresponding to temperature

A1: corresponding ADC value at 25℃ (corresponding ADC value at 3.3 V: 830)

35.8 Register Description

ADCC0 register base address: 0x4700_6000

ADCC1 register base address: 0x4700_6800

The registers are listed below:

Table 35-3: List of ADC Memory-mapped Registers

Offset Address	Name	Description
0x0	ADC_ANCTRL	Analog circuit control register
0x4	ADC_DGCTRL	Digital circuit control register
0x8	ADC_CLKCTRL	ADC clock control register
0xC	ADC_CHAVGCFG0	Multiple average setting register 0
0x10	ADC_CHAVGCFG1	Multiple average setting register 1

Offset Address	Name	Description
0x14	ADC_RGLCHCFG0	Regular sequence channel setting register 0
0x18	ADC_RGLCHCFG1	Regular sequence channel setting register 1
0x1C	ADC_RGLCHCFG2	Regular sequence channel setting register 2
0x20	ADC_RGLCHCFG3	Regular sequence channel setting register 3
0x24	ADC_INJCHCFG	Injection sequence channel setting register
0x28	ADC_CHDAT0	Channel data register 0
0x2C	ADC_CHDAT1	Channel data register 1
0x30	ADC_CHDAT2	Channel data register 2
0x34	ADC_CHDAT3	Channel data register 3
0x38	ADC_CHDAT4	Channel data register 4
0x3C	ADC_CHDAT5	Channel data register 5
0x40	ADC_CHDAT6	Channel data register 6
0x44	ADC_CHDAT7	Channel data register 7
0x48	ADC_CHDAT8	Channel data register 8
0x4C	ADC_CHDAT9	Channel data register 9
0x50	ADC_CHDAT10	Channel data register 10
0x54	ADC_CHDAT11	Channel data register 11
0x58	ADC_CHDAT12	Channel data register 12
0x5C	ADC_CHDAT13	Channel data register 13
0x60	ADC_CHDAT14	Channel data register 14
0x64	ADC_CHDAT15	Channel data register 15
0x68	ADC_CHDAT16	Channel data register 16
0x6C	ADC_CHDAT17	Channel data register 17
0x70	ADC_CHDAT18	Channel data register 18
0x74	ADC_CHDAT19	Channel data register 19
0x78	ADC_DUALDAT	Dual data shadow register
0x7C	ADC_FIFO_OUT	RX FIFO data register
0x80	ADC_WDGEN	Watchdog enable register
0x84	ADC_WDGCOND	Watchdog comparison condition register
0x88	ADC_STAT	Current status register
0x8C	ADC_INTSTAT	Interrupt status/clear register
0x90	ADC_INTEN	Interrupt enable register
0x94	ADC_MINTSTAT	Enabled interrupt status shadow register

Registers are detailed in the following sections.

35.8.1 Analog Circuit Control Register (ADC_ANCTRL)

Offset address: 0x00

Reset value: 0x000 0000

Bit	Name	Attribute	Reset Value	Description
32:19	RSV	-	-	Reserved
18:17	VDDH_DIV_SEL	R/W	0x0	VDDH voltage divider selection: 00: 1/4 VDDH 01: 1/3 VDDH 10: 1/2 VDDH 11: 2/3 VDDH Note: Only ADCC0 can control this register.
16	VDDH_DIV_EN	R/W	0x0	VDDH voltage divider function enable: 0: disabled 1: enabled Note: Only ADCC0 can control this register.
15	TS_EN	R/W	0x0	TEMPSENSOR enable control bit: 0: TS disabled 1: TS enabled Note: Only ADCC0 can control TS.
14	OPAMP_EN	R/W	0x0	ADC and OPA connection enable: Note: This register of ADC controller 0 corresponds to OPA0, and ADC controller 1 corresponds to OPA1. 0: disconnected 1: connected
13:7	RSV	-	-	Reserved
6	DIFF_EN	R/W	0x0	Channel differential input enable: 0: single-ended input 1: differential input
5:4	VREFP_SEL	R/W	0x0	ADC positive reference voltage selection: 0 or 1: power supply V_{DDA} of analog circuit 2: external input reference voltage V_{REFP} or provided by the internal VREF module (to use the internal VREF module, the corresponding function must be called to enable this feature)

Bit	Name	Attribute	Reset Value	Description
				3: reserved
3	POWER_ON	R/W	0x0	ADC power-up control: The analog ADC requires at least 32 ADCCLK cycles for stabilization after powering up. 0: power down 1: power up
2	ADC_STOP	W1C	0x0	Conversion stop control: Writing 1 in any state will reset the sampler and receiver of the ADC controller to their initial values and place them into the idle state. The sampler will stop sending adc_soc signals to the analog ADC, and the receiver will no longer receive new incoming data unless a new sequence conversion is initiated by software or hardware. This bit does not affect the generation of ADCCLK, nor does it affect the register or RX FIFO. 0: write invalid 1: stop conversion
1	INJ_START	W1C	0x0	Injection sequence conversion start control: In the stable state, idle state or normal state, writing 1 will start the injection sequence conversion after the current conversion ends; writing 1 during the injection state is invalid. 0: write invalid 1: software starts the injection sequence conversion
0	RGL_START	W1C	0x0	Regular sequence conversion start control: In the stable state or idle state, writing 1 will immediately start the regular sequence conversion; writing 1 during the regular or injection state is invalid. 0: write invalid 1: software starts the regular sequence conversion

35.8.2 Digital Circuit Control Register (ADC_DGCTRL)

Offset address: 0x04

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:27	WATERMARK	R/W	0x0	RX FIFO data available trigger count selection: When a certain amount of data is available in RX FIFO, RXAVAIL is set to 1, triggering a DMA transfer. 00000: at least 1 data in RX FIFO 00001: at least 2 data in RX FIFO 00010: at least 3 data in RX FIFO 01110: at least 15 data in RX FIFO … 11111: reserved
26	COMPLEMNT_EN	R/W	0x0	Enable the ADC conversion data to be stored in the register and RX FIFO in two's complement format: The watchdog compares the sign-magnitude representation with the threshold and is not affected by this bit. 0: save data in sign-magnitude format 1: save data in two's complement format
25:24	DMA_MOD	R/W	0x0	DMA receive mode selection: 0 or 3: DMA receive disabled 1: DMA receive mode 1, where a DMA request is issued when the number of data stored in FIFO (mixed regular sequence and injection sequence) reaches the number set by the watermark. 2: DMA receive mode 2, where a DMA request is issued when the ADC controller itself receives data, allowing reading the data of channel data registers 0−16 (CHDAT0−15) or dual data shadow

Bit	Name	Attribute	Reset Value	Description
				registers (DUALDAT).
23	FIFO_EN	R/W	0x0	RX FIFO enable: 0: RX FIFO disabled, data cleared 1: RX FIFO enabled
22	DATA_R_CLR	R/W	0x0	Enable automatic clearing of channel data after reading: 0: Data will not be cleared after reading the channel data register. 1: Data will be automatically cleared after reading the channel data register.
21:20	INJ_TRIG_EDG	R/W	0x0	Trigger signal edge selection for injection sequence conversion: 0: signal trigger disabled 1: rising- edge trigger 2: falling-edge trigger 3: double-edge trigger
19:16	INJ_TRIG_SEL	R/W	0x0	Trigger signal source selection for injection sequence conversion: 0x0: CC4 event of timer 0 0x1: TRGO event of timer 0 0x2: CC1 event of timer 1 0x3: TRGO event of timer 1 0x4: CC2 event of timer 2 0x5: CC4 event of timer 2 0x6: CC1 event of timer 3 0x7: CC2 event of timer 3 0x8: CC3 event of timer 3 0x9: TRGO event of timer 3 0xA: CC4 event of timer 4 0xB: TRGO event of timer 4 0xC: CC2 event of timer 7 0xD: CC3 event of timer 7 0xE: CC4 event of timer 7 0xF: external input port 15
15:14	RGL_TRIG_EDG	R/W	0x0	Trigger signal edge selection for regular sequence conversion:

Bit	Name	Attribute	Reset Value	Description
				0: signal trigger disabled 1: rising edge trigger 2: falling edge trigger 3: double-edge trigger
13:10	RGL_TRIG_SEL	R/W	0x0	Trigger signal source selection for regular sequence conversion: 0x0: CC1 event of timer 0 0x1: CC2 event of timer 0 0x2: CC3 event of timer 0 0x3: CC2 event of timer 1 0x4: CC3 event of timer 1 0x5: CC4 event of timer 1 0x6: TRGO event of timer 1 0x7: CC1 event of timer 2 0x8: TRGO event of timer 2 0x9: CC4 event of timer 3 0xA: CC1 event of timer 4 0xB: CC2 event of timer 4 0xC: CC3 event of timer 4 0xD: CC1 event of timer 7 0xE: TRGO event of timer 7 0xF: external input port 11
9	RGL_RCV_SEL	R/W	0x0	Position selection for resuming scanning after a regular sequence is interrupted by an injection sequence: 0: resume scanning from the interrupted position 1: restart scanning from the beginning
8:7	DUAL_INJ_MOD	R/W	0x0	Cooperation mode selection for injection sequence conversion in dual-ADC cooperative mode: 0 or 3: independent scan mode for injection sequence 1: parallel scan mode for injection sequence 2: alternating trigger scan mode for injection sequence

Bit	Name	Attribute	Reset Value	Description
				Note: If it is configured to the parallel scan mode or alternating trigger scan mode for the injection sequence, ADC1 will always use the same clock as ADC0. The clock for ADC1 is selected and generated by the CLKCTRL register of ADC controller 0, while the clock selection and clock division settings in the CLKCTRL register of ADC controller 1 will not take effect.
6:5	DUAL_RGL_MOD	R/W	0x0	Cooperation mode selection for regular sequence conversion in dual-ADC cooperative mode: 0 or 3: independent scan mode for regular sequence 1: synchronous scan mode for regular sequence 2: delayed scan mode for regular sequence Note: If it is configured to the synchronous scan mode or delayed scan mode for the regular sequence, ADC1 will always use the same clock as ADC0. The clock for ADC1 is selected and generated by the CLKCTRL register of ADC controller 0, while the clock selection and clock division settings in the CLKCTRL register of ADC controller 1 will not take effect.
4	INJ_MOD	R/W	0x0	Injection sequence conversion mode selection: 0: no conversion 1: single trigger scan mode
3:2	RGL_MOD	R/W	0x0	Regular sequence conversion mode selection: 0: no conversion 1: single scan mode 2: continuous scan mode 3: intermittent scan mode

Bit	Name	Attribute	Reset Value	Description
1	RD_EDGE	R/W	0x0	Trigger edge selection for ADC controller to read conversion results: 0: ADC_EOC rising edge 1: ADC_EOC falling edge
0	ADCC_EN	R/W	0x0	ADC controller enable: This bit controls the clock generator, sampling controller and data receiver. Note: The RX FIFO data is cleared through the FIFO_EN bit (bit 23) of this register. 0: ADC controller disabled 1: ADC controller enabled

35.8.3 ADCx Clock Control Register (ADC_CLKCTRL)

Offset address: 0x08

Reset value: 0x0A40 0003

Bit	Name	Attribute	Reset Value	Description
31:28	SYNC_DELAY	R/W	0x0	In the regular sequence delay mode of the dual-ADC cooperative mode, the delay time setting for the current ADC to start conversion relative to the previous ADC is applicable only in this mode and does not take effect in other operation modes of the regular sequence. For example, this bit in ADC controller 0 indicates the delay time of the ADC_SOC signal set of ADC0 relative to ADC1; while this bit in ADC controller 1 indicates the delay time of ADC_SOC signal set of ADC1 relative to ADC0. To stagger the sampling time by 4 ADCCLK cycles, the delay time is set to (Sync_delay + 6) ADCCLK cycles, with a range of 6 to 21 ADCCLK cycles.
27:24	CH_SWITCH	R/W	0xA	Channel switching timing selection: This bit decides when to switch channels based

Bit	Name	Attribute	Reset Value	Description
				on ADCCLK counter value. Note: Do not set it to 0-3, as the ADC is sampling when the ADCCLK counter reaches 0–3, and channel switching is prohibited at this time.
23:22	SOC_WIDTH	R/W	0x1	ADC_SOC signal width: The duration of the ADC_SOC signal being high is from the falling edge of the ADCCLK to the rising edge of the (Soc_width + 2)th ADCCLK. 0: 1.5 ADCCLK clock cycles 1: 2.5 ADCCLK clock cycles 2: 3.5 ADCCLK clock cycles 3: 4.5 ADCCLK clock cycles
21:17	STABL_TIME	R/W	0x0	ADC stabilization time configuration: ADC stabilization time = (Stabl_time + 32) ADCCLK cycles, with a range of 32 to 63.
16	ADCCLK_SEL	R/W	0x0	ADC clock source selection: 0: PCLK divided clock 1: synchronized external clock (XTH clock only)
15:0	PCLK_DIV	R/W	0x3	PCLK division factor setting: The effective division factor is (PCLK_DIV + 1), with a range of 1 to 65535. Note: 1. If f_{PCLK} = 168 MHz, the actual division factor shall be greater than or equal to 2, that is, PCLK_DIV ≥ 1 should be set. 2. For different PCLK frequencies, the ADC independent scan mode and the delayed continuous scan mode for the regular sequence of dual ADCs without injection sequence support a minimum division of 1; other dual-ADC cooperative modes support a minimum division of 2, that is, PCLK_DIV ≥ 1 should be set.

Notes: ADC0 always uses the ADCCLK clock generated by ADC controller 0. ADC1 uses the ADCCLK clock generated by ADC controller 1 in independent operation mode, but uses the ADCCLK clock generated by ADC controller 0 in dual-ADC cooperative mode.

Furthermore, in dual-ADC cooperative mode, ADC controller 1 still needs to properly configure the bits [31:17] of this register.

35.8.4 Multiple Average Setting Register 0 (ADC_CHAVGCFG0)

Offset address: 0x0C

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:30	RSV	-	-	Reserved
29:27	CH9_AVG_SEL	R/W	0x0	Channel 9 conversion number setting (Refer to channel 0 conversion number setting)
26:24	CH8_AVG_SEL	R/W	0x0	Channel 8 conversion number setting (Refer to channel 0 conversion number setting)
23:21	CH7_AVG_SEL	R/W	0x0	Channel 7 conversion number setting (Refer to channel 0 conversion number setting)
20:18	CH6_AVG_SEL	R/W	0x0	Channel 6 conversion number setting (Refer to channel 0 conversion number setting)
17:15	CH5_AVG_SEL	R/W	0x0	Channel 5 conversion number setting (Refer to channel 0 conversion number setting)
14:12	CH4_AVG_SEL	R/W	0x0	Channel 4 conversion number setting (Refer to channel 0 conversion number setting)
11:9	CH3_AVG_SEL	R/W	0x0	Channel 3 conversion number setting (Refer to channel 0 conversion number setting)
8:6	CH2_AVG_SEL	R/W	0x0	Channel 2 conversion number setting (Refer to channel 0 conversion number setting)
5:3	CH1_AVG_SEL	R/W	0x0	Channel 1 conversion number setting (Refer to channel 0 conversion number setting)
2:0	CH0_AVG_SEL	R/W	0x0	Channel 0 conversion number setting: 0: 1 conversion 1: 2 conversions with results averaged

Bit	Name	Attribute	Reset Value	Description
				2: 4 conversions with results averaged 3: 8 conversions with results averaged 4: 16 conversions with results averaged 5: 32 conversions with results averaged 6: 64 conversions with results averaged 7: 128 conversions with results averaged

35.8.5 Multiple Average Setting Register 1 (ADC_CHAVGCFG1)

Offset address: 0x10

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:30	RSV	-	-	Reserved
29:27	CH19_AVG_SEL	R/W	0x0	Channel 19 conversion number setting (Refer to channel 0 conversion number setting)
26:24	CH18_AVG_SEL	R/W	0x0	Channel 18 conversion number setting (Refer to channel 0 conversion number setting)
23:21	CH17_AVG_SEL	R/W	0x0	Channel 17 conversion number setting (Refer to channel 0 conversion number setting)
20:18	CH16_AVG_SEL	R/W	0x0	Channel 16 conversion number setting (Refer to channel 0 conversion number setting)
17:15	CH15_AVG_SEL	R/W	0x0	Channel 15 conversion number setting: (Refer to channel 0 conversion number setting)
14:12	CH14_AVG_SEL	R/W	0x0	Channel 14 conversion number setting (Refer to channel 0 conversion number setting)
11:9	CH13_AVG_SEL	R/W	0x0	Channel 13 conversion number setting (Refer to channel 0 conversion number setting)

Bit	Name	Attribute	Reset Value	Description
8:6	CH12_AVG_SEL	R/W	0x0	Channel 12 conversion number setting (Refer to channel 0 conversion number setting)
5:3	CH11_AVG_SEL	R/W	0x0	Channel 11 conversion number setting (Refer to channel 0 conversion number setting)
2:0	CH10_AVG_SEL	R/W	0x0	Channel 10 conversion number setting (Refer to channel 0 conversion number setting)

35.8.6 Regular Sequence Channel Setting Register 0 (ADC_RGLCHCFG0)

Offset address: 0x14

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:30	RSV	-	-	Reserved
29:25	RGL_CH6_SEL	R/W	0x0	Selection of 6th conversion channel for regular sequence: (Refer to the 1st conversion channel selection)
24:20	RGL_CH5_SEL	R/W	0x0	Selection of 5th conversion channel for regular sequence: (Refer to the 1st conversion channel selection)
19:15	RGL_CH4_SEL	R/W	0x0	Selection of 4th conversion channel for regular sequence: (Refer to the 1st conversion channel selection)
14:10	RGL_CH3_SEL	R/W	0x0	Selection of 3rd conversion channel for regular sequence: (Refer to the 1st conversion channel selection)
9:5	RGL_CH2_SEL	R/W	0x0	Selection of 2nd conversion channel for regular sequence: (Refer to the 1st conversion channel

Bit	Name	Attribute	Reset Value	Description
				selection)
4:0	RGL_CH1_SEL	R/W	0x0	Selection of 1st conversion channel for regular sequence:: 0x0: external channel 0 0x1: external channel 1 0x2: external channel 2 0x3: external channel 3 0x4: external channel 4 0x5: external channel 5 0x6: external channel 6 0x7: external channel 7 0x8: external channel 8 0x9: external channel 9 0xA: external channel 10 0xB: external channel 11 0xC: external channel 12 0xD: external channel 13 0xE: external channel 14 0xF: external channel 15 0x10: internal channel 0 (ADC0: VREF1P2V, ADC1: VBAT_DIV) 0x11: external channel 1 (ADC0: TS, ADC1: OPA0) 0x12: external channel 2 (ADC0: VDDH_DIV, ADC1: OPA1) 0x13: internal channel 3 (ADC0: OPA2, ADC1: OPA2) 0x14–0x1F: reserved

35.8.7 Regular Sequence Channel Setting Register 1 (ADC_RGLCHCFG1)

Offset address: 0x18

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:30	RSV	-	-	Reserved

Bit	Name	Attribute	Reset Value	Description
29:25	RGL_CH12_SEL	R/W	0x0	Selection of 12th conversion channel for regular sequence: (Refer to the 1st conversion channel selection)
24:20	RGL_CH11_SEL	R/W	0x0	Selection of 11th conversion channel for regular sequence: (Refer to the 1st conversion channel selection)
19:15	RGL_CH10_SEL	R/W	0x0	Selection of 10th conversion channel for regular sequence: (Refer to the 1st conversion channel selection)
14:10	RGL_CH9_SEL	R/W	0x0	Selection of 9th conversion channel for regular sequence: (Refer to the 1st conversion channel selection)
9:5	RGL_CH8_SEL	R/W	0x0	Selection of 8th conversion channel for regular sequence: (Refer to the 1st conversion channel selection)
4:0	RGL_CH7_SEL	R/W	0x0	Selection of 7th conversion channel for regular sequence: (Refer to the 1st conversion channel selection)

35.8.8 Regular Sequence Channel Setting Register 2 (ADC_RGLCHCFG2)

Offset address: 0x1C

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:30	RSV	-	-	Reserved
29:25	RGL_CH18_SEL	R/W	0x0	Selection of 18th conversion channel for regular sequence: (Refer to the 1st conversion channel selection)

Bit	Name	Attribute	Reset Value	Description
24:20	RGL_CH17_SEL	R/W	0x0	Selection of 17th conversion channel for regular sequence: (Refer to the 1st conversion channel selection)
19:15	RGL_CH16_SEL	R/W	0x0	Selection of 16th conversion channel for regular sequence: (Refer to the 1st conversion channel selection)
14:10	RGL_CH15_SEL	R/W	0x0	Selection of 15th conversion channel for regular sequence: (Refer to the 1st conversion channel selection)
9:5	RGL_CH14_SEL	R/W	0x0	Selection of 14th conversion channel for regular sequence: (Refer to the 1st conversion channel selection)
4:0	RGL_CH13_SEL	R/W	0x0	Selection of 13th conversion channel for regular sequence: (Refer to the 1st conversion channel selection)

35.8.9 Regular Sequence Channel Setting Register 3 (ADC_RGLCHCFG3)

Offset address: 0x20

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:18	RSV	-	-	Reserved
17:15	SHORT_LENG	R/W	0x0	Channel conversion position number setting in regular short sequence in intermittent scan mode: The number of channel positions in a regular short sequence is set to (Short_leng + 1). The regular short sequence can contain a maximum of 8 channel conversion

Bit	Name	Attribute	Reset Value	Description
				positions, i.e., the range of Short_leng is 0–7.
14:10	RGL_LENG	R/W	0x0	Channel conversion position number setting in regular sequence: The number of channel positions in a regular sequence is set to (Rgl_leng + 1). The regular sequence can contain a maximum of 20 channel conversion positions, i.e., the valid range of Rgl_leng is 0–19. Note: The software shall avoid configuring Rgl_leng > 19, as any positions beyond this will be assigned to channel 0 and will only be converted once.
9:5	RGL_CH20_SEL	R/W	0x0	Selection of 20th conversion channel for regular sequence: (Refer to the 1st conversion channel selection)
4:0	RGL_CH19_SEL	R/W	0x0	Selection of 19th conversion channel for regular sequence: (Refer to the 1st conversion channel selection)

35.8.10 Injection Sequence Channel Setting Register (ADC_INJCHCFG)

Offset address: 0x24

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:22	RSV	-	-	Reserved
21:20	INJ_LENG	R/W	0x0	Channel conversion position number setting in injection sequence: The number of channel positions in an injection sequence is set to (Inj_leng + 1). The injection sequence can contain a

Bit	Name	Attribute	Reset Value	Description
				maximum of 4 channel conversion positions.
19:15	INJ_CH4_SEL	R/W	0x0	Selection of 4th conversion channel for injection sequence: (Refer to the 1st conversion channel selection)
14:10	INJ_CH3_SEL	R/W	0x0	Selection of 3rd conversion channel for injection sequence: (Refer to the 1st conversion channel selection)
9:5	INJ_CH2_SEL	R/W	0x0	Selection of 2nd conversion channel for injection sequence: (Refer to the 1st conversion channel selection)
4:0	INJ_CH1_SEL	R/W	0x0	Selection of 1st conversion channel for injection sequence: 0x0: external channel 0 0x1: external channel 1 0x2: external channel 2 0x3: external channel 3 0x4: external channel 4 0x5: external channel 5 0x6: external channel 6 0x7: external channel 7 0x8: external channel 8 0x9: external channel 9 0xA: external channel 10 0xB: external channel 11 0xC: external channel 12 0xD: external channel 13 0xE: external channel 14 0xF: external channel 15 0x10–0x1F: reserved

35.8.11 Channel Data Register 0–19 (ADC_CHDAT0–19)

Offset address: 0x28–0x74

Reset value: 0000 0000

Bit	Name	Attribute	Reset Value	Description
31:17	RSV	-	-	Reserved
16	DATA_VALID	R	0x0	Data valid signal: This signal is activated upon obtaining valid data. It is cleared by hardware when Adc_en = 0 or after the software reads this register. 0: data being invalid 1: data being valid
15:12	RSV	-	-	Reserved
11:0	CH_DATA	R	0x0	Channel data: If Dat_r_clr = 1, it will be cleared by hardware after the software reads this register.

Note: Registers 0 to 15 correspond to the data from 16 external input channels. For dual-port differential inputs, the differential data from ports 0 and 1 will be stored in register 0, the differential data from ports 2 and 3 will be stored in register 2, and so on.

35.8.12 Dual-data Shadow Register (ADC_DUALDAT)

Offset address: 0x78

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:28	ADC1_LAST_CH_NUM	R	0x0	ADC1 last converted channel number
27:16	ADC1_DATA	R	0x0	ADC1 last conversion result
15:12	ADC0_LAST_CH_NUM	R	0x0	ADC0 last converted channel number
11:0	ADC0_DATA	R	0x0	ADC0 last conversion result

Notes:

- The channel numbers and results of the most recent conversions from both ADCs are

mapped to this register address simultaneously, which facilitates data reading in dual-ADC cooperative mode.

- In DMA mode 2, a DMA request is issued as soon as the ADC itself has valid data ready. However, if the other ADC does not have data ready at that moment, the channel number of the other ADC indicates the channel currently being prepared, and the data is the previous data from that channel (invalid). Therefore, to ensure that both ADCs are ready for data simultaneously, the conversion counts for each channel of both ADCs shall be set to the same value, and conversions shall be triggered only after the IDLE_STATE status flag is set, allowing both ADCs to start conversions at the same time. Of course, since each ADC controller issues its own DMA request, it is also feasible to read only the valid data from each ADC during each request.

35.8.13 RX FIFO Data Register (ADC_FIFO_OUT)

Offset address: 0x7C

Reset value: 0000 0000

Bit	Name	Attribute	Reset Value	Description
31:16	RSV	-	-	Reserved
15:12	LAST_CH_NUM	R	0x0	Channel number
11:0	FIFO_OUT	R	0x0	RX FIFO data

35.8.14 Watchdog Enable Register (ADC_WDGEN)

Offset address: 0x80

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:16	RSV	-	-	Reserved
15	WDG_EN_15	R/W	0x0	Channel 15 watchdog enable: 0: watchdog function disabled 1: watchdog function enabled

Bit	Name	Attribute	Reset Value	Description
14	WDG_EN_14	R/W	0x0	Channel 14 watchdog enable: 0: watchdog function disabled 1: watchdog function enabled
13	WDG_EN_13	R/W	0x0	Channel 13 watchdog enable: 0: watchdog function disabled 1: watchdog function enabled
12	WDG_EN_12	R/W	0x0	Channel 12 watchdog enable: 0: watchdog function disabled 1: watchdog function enabled
11	WDG_EN_11	R/W	0x0	Channel 11 watchdog enable: 0: watchdog function disabled 1: watchdog function enabled
10	WDG_EN_10	R/W	0x0	Channel 10 watchdog enable: 0: watchdog function disabled 1: watchdog function enabled
9	WDG_EN_9	R/W	0x0	Channel 9 watchdog enable: 0: watchdog function disabled 1: watchdog function enabled
8	WDG_EN_8	R/W	0x0	Channel 8 watchdog enable: 0: watchdog function disabled 1: watchdog function enabled
7	WDG_EN_7	R/W	0x0	Channel 7 watchdog enable: 0: watchdog function disabled 1: watchdog function enabled
6	WDG_EN_6	R/W	0x0	Channel 6 watchdog enable: 0: watchdog function disabled 1: watchdog function enabled
5	WDG_EN_5	R/W	0x0	Channel 5 watchdog enable: 0: watchdog function disabled 1: watchdog function enabled
4	WDG_EN_4	R/W	0x0	Channel 4 watchdog enable: 0: watchdog function disabled 1: watchdog function enabled
3	WDG_EN_3	R/W	0x0	Channel 3 watchdog enable: 0: watchdog function disabled 1: watchdog function enabled

Bit	Name	Attribute	Reset Value	Description
2	WDG_EN_2	R/W	0x0	Channel 2 watchdog enable: 0: watchdog function disabled 1: watchdog function enabled
1	WDG_EN_1	R/W	0x0	Channel 1 watchdog enable: 0: watchdog function disabled 1: watchdog function enabled
0	WDG_EN_0	R/W	0x0	Channel 0 watchdog enable: 0: watchdog function disabled 1: watchdog function enabled

35.8.15 Watchdog Comparison Condition Register (ADC_WDGCOND)

Offset address: 0x84

Reset value: 0x0FFF 0000

Bit	Name	Attribute	Reset Value	Description
31	CONDITION	R/W	0x0	Selection of trigger condition for watchdog event: 0: The watchdog will trigger an alarm when the sign-magnitude of converted data exceeds the threshold range, i.e., the sign-magnitude of converted data > H_threshold, or the sign-magnitude of converted data < L_threshold. 1: The watchdog will trigger an alarm when the sign-magnitude of converted data is within the threshold range, i.e., L_threshold ≤ the sign-magnitude of converted data ≤ H_threshold.
30:28	RSV	R	0x0	Reserved
27:16	H_THRESHOLD	R/W	0xFFF	Watchdog threshold upper limit
15:12	RSV	R	0x0	Reserved
11:0	L_THRESHOLD	R/W	0x0	Watchdog threshold lower limit

Notes:

1. If multiple-sampling mode is used, the watchdog only compares the average value of the converted data with the thresholds.
2. In addition to setting the watchdog comparison conditions before ADC startup, the comparison conditions can also be changed during ADC conversion whenever the WDG_CMPLT flag is set to 1, so that the comparison conditions for each channel data can vary. However, this approach carries the risk of delayed updates to the comparison conditions.
3. If it is set to the watchdog reset value, the watchdog will not trigger an alarm.

35.8.16 Current Status Register (ADC_STAT)

Offset address: 0x88

Reset value: ADC0: 0x3000000

ADC1: 0x2000000

Bit	Name	Attribute	Reset Value	Description
31:28	RSV	-	-	Reserved
27	WDG_CMPLT	R/W1C	0x0	Watchdog channel data comparison completion flag, can be cleared by writing 1 When the watchdog finishes comparing the data of a channel, this flag will be set to 1. When the flag is 1, the watchdog comparison conditions can be changed, and the new comparison conditions is used for the comparison of the next channel data (regardless of regular sequence or injection sequence). After writing to the watchdog comparison condition register (WDGCOND), this flag will also be cleared. During ADC operation, when this flag is 0, do not change the watchdog comparison conditions to avoid erroneous comparison results.

Bit	Name	Attribute	Reset Value	Description
				0: watchdog channel data comparison completed 1: watchdog channel data comparison not completed
26	FIFO_FULL	R	0x0	RX FIFO full flag: 0: less than 32 data entries in RX FIFO 1: 32 data entries in RX FIFO
25	FIFO_EMPTY	R	0x1	RX FIFO empty flag: 0: There is data in RX FIFO. 1: There is no data in RX FIFO.
24	INJ_IN_TURN	R	ADC0: 0x1 ADC1: 0x0	Alternating trigger injection cycle: This register indicates which ADC is currently executing the injection conversion in the alternating trigger scan mode, or which ADC will perform the conversion the next time the injection conversion is triggered. 0: Current ADC is not in turn. 1: Current ADC is in turn.
23:21	INJ_NUM	R	0x0	Position number of the channel currently being converted in the injection sequence: 1–4 indicates the position number of the channel being converted in the injection sequence, while 0 indicates that no channel in the injection sequence is currently being converted.
20:16	RGL_NUM	R	0x0	Position number of the channel currently being converted in the regular sequence: 1–20 indicates the position number of the channel being converted in the regular sequence, while 0 indicates that no channel in the regular sequence is currently being converted (including when the injection sequence is being converted).
15:11	CH_NUM	R	0x0	Number of channel being converted: Valid range: 0–15; when idle, it displays the channel number of the first position in the regular sequence (set by Rgl_ch1_sel).

Bit	Name	Attribute	Reset Value	Description
10:4	CONV_CNT	R	0x0	Number of completed conversions at the current conversion position: Since each channel can independently set the number of conversions used to calculate the average, the valid range of this register value is no greater than (the selected value of Chn_avg_sel - 1).
3:1	RSV	-	-	Reserved
0	IDLE_STATE	R	0x0	Idle status flag: When this flag is 1, the ADC can respond to the trigger signals for the regular sequence. After ADC reset, it first enters the stabilization state, and then enters the idle state after 32 ADC clock cycles. In dual-ADC cooperative mode, if the conversion time length of the two ADCs is different, this flag will remain 0 when the ADC that has completed the conversion is waiting for the other ADC that is still converting.

35.8.17 Interrupt Status / Clear Register (ADC_INTSTAT)

Offset address: 0x8C

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:7	RSV	-	-	Reserved
6	WDG_INJ	R/W1C	0x0	Injection sequence watchdog alarm interrupt flag, can be cleared by writing 1 During the conversion of the injection sequence, if the watchdog function is enabled for the most recently converted channel and its conversion result meets the watchdog alarm condition, an interrupt will be triggered.
5	WDG_RGL	R/W1C	0x0	Regular sequence watchdog alarm interrupt flag, can be cleared by writing 1

Bit	Name	Attribute	Reset Value	Description
				During the conversion of the regular sequence, if the watchdog function is enabled for the most recently converted channel and its conversion result meets the watchdog alarm condition, an interrupt will be triggered.
4	WDG_CH	R/W1C	0x0	Channel data watchdog alarm interrupt flag, can be cleared by writing 1 If the watchdog function is enabled for the most recently converted channel and its conversion result meets the watchdog alarm condition, an interrupt will be triggered.
3	FIFO_OVF	R/W1C	0x0	RX FIFO overflow interrupt flag, can be cleared by writing 1
2	FIFO_AVL	R/W1C	0x0	RX FIFO data available interrupt flag, can be cleared by writing 1 When the amount of data in the RX FIFO reaches the value set by WATERMARK in the DGCTRL register, this flag will be set to 1.
1	EOIC	R/W1C	0x0	Injection sequence conversion complete interrupt flag, can be cleared by writing 1
0	EORC	R/W1C	0x0	Regular sequence conversion complete interrupt flag, can be cleared by writing 1

35.8.18 Interrupt Enable Register (ADC_INTEN)

Offset address: 0x90

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:7	RSV	-	-	Reserved
6	WDG_INJ_EN	R/W	0x0	Injection sequence watchdog alarm interrupt enable
5	WDG_RGL_EN	R/W	0x0	Regular sequence watchdog alarm interrupt enable
4	WDG_CH_EN	R/W	0x0	Channel data watchdog alarm interrupt enable

Bit	Name	Attribute	Reset Value	Description
3	FIFO_OVF_EN	R/W	0x0	RX FIFO overflow interrupt enable
2	FIFO_AVL_EN	R/W	0x0	RX FIFO data available interrupt enable
1	EOIC_EN	R/W	0x0	Injection sequence conversion complete interrupt enable
0	EORC_EN	R/W	0x0	Regular sequence conversion complete interrupt enable

35.8.19 Enabled Interrupt Status Shadow Register (ADC_MINTSTAT)

Offset address: 0x94

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:7	RSV	-	-	Reserved
6	WDG_INJ_M	R	0x0	Enabled injection sequence watchdog alarm interrupt flag
5	WDG_RGL_M	R	0x0	Enabled regular sequence watchdog alarm interrupt flag
4	WDG_CH_M	R	0x0	Enabled channel data watchdog alarm interrupt flag
3	FIFO_OVF_M	R	0x0	Enabled RX FIFO overflow interrupt flag
2	FIFO_AVL_M	R	0x0	Enabled RX FIFO data available interrupt flag
1	EOIC_M	R	0x0	Enabled injection sequence conversion complete interrupt flag
0	EORC_M	R	0x0	Enabled regular sequence conversion complete interrupt flag

Note: This register maps the result of the logical “AND” operation between the interrupt status/clear register (INTSTAT) and the interrupt enable register (INTEN). To clear an interrupt flag, write “1” to the (original) interrupt flag in the interrupt status/clear register (INTSTAT).

35.9 Operation Procedure

35.9.1 Multi-channel A/D Conversion in Single Scan Mode of Regular Sequence

In single scan mode, ADC performs only one conversion after starting the conversion.

1. Configure the channel to be converted as an analog interface (GPIOx_MODE) based on the GPIO pin corresponding to the ADC input channel.
2. Configure the APB2 peripheral clock enable register (APB2CKENR) to enable the ADC clock.
3. Set ADC_ANCTRL[3] to 1 to power on the ADC.
4. Configure ADC_ANCTRL[5:4] to set the ADC positive reference voltage as the analog circuit power supply V_{DDA}.
5. Set ADC_DGCTRL[3:2] to 1 to set the regular sequence conversion mode to single scan mode.
6. Set ADC_DGCTRL[22] to 1 to automatically clear the channel data after reading.
7. Configure ADC_CLKCTRL[27:24] to set the timing at which the channel selects to switch.
8. Configure ADC_CLKCTRL[15:0] to set the ADC sampling rate.
9. Configure the multiple average setting register to set the number of conversions for each channel and average the results.
10. Configure the regular sequence channel setting register to set the channels for conversion at each position and the number of positions for channel conversions in the regular sequence.
11. Set ADC_DGCTRL[0] to 1 to enable the ADC controller.
12. Set ADC_ANCTRL[0] to 1 to start the regular sequence conversion via software.

13. Wait for the ADC regular sequence conversion to complete and read the data from the data register of the corresponding channel.
14. If multiple conversions are required, repeat steps 12 and 13.

35.9.2 Multi-channel A/D Conversion in Continuous Scan Mode of Regular Sequence

1. Configure the channel to be converted as an analog interface (GPIOx_MODE) based on the GPIO pin corresponding to the ADC input channel.
2. Configure the APB2 peripheral clock enable register (APB2CKENR) to enable the ADC clock.
3. Set ADC_ANCTRL[3] to 1 to power on the ADC.
4. Configure ADC_ANCTRL[5:4] to set the ADC positive reference voltage as the analog circuit power supply V_{DDA}.
5. Set ADC_DGCTRL[3:2] to 2 to set the regular sequence conversion mode to continuous scan mode.
6. Set ADC_DGCTRL[22] to 1 to automatically clear the channel data after reading.
7. Configure ADC_CLKCTRL[27:24] to set the timing at which the channel selects to switch.
8. Configure ADC_CLKCTRL[15:0] to set the ADC sampling rate.
9. Configure the multiple average setting register to set the number of conversions for each channel and average the results.
10. Configure the regular sequence channel setting register to set the channels for conversion at each position and the number of positions for channel conversions in the regular sequence.
11. Set ADC_DGCTRL[0] to 1 to enable the ADC controller.

12. Set ADC_ANCTRL[0] to 1 to start the regular sequence conversion via software.
13. Wait for the ADC regular sequence conversion to complete and read the data from the data register of the corresponding channel.

35.9.3 Multi-channel A/D Conversion in Intermittent Scan Mode of Regular Short Sequence

1. Configure the channel to be converted as an analog interface (GPIOx_MODE) based on the GPIO pin corresponding to the ADC input channel.
2. Configure the APB2 peripheral clock enable register (APB2CKENR) to enable the ADC clock.
3. Set ADC_ANCTRL[3] to 1 to power on the ADC.
4. Configure ADC_ANCTRL[5:4] to set the ADC positive reference voltage as the analog circuit power supply V_{DDA}.
5. Set ADC_DGCTRL[3:2] to 3 to set the regular sequence conversion mode to intermittent scan mode.
6. Set ADC_DGCTRL[22] to 1 to automatically clear the channel data after reading.
7. Configure ADC_CLKCTRL[27:24] to set the timing at which the channel selects to switch.
8. Configure ADC_CLKCTRL[15:0] to set the ADC sampling rate.
9. Configure the multiple average setting register to set the number of conversions for each channel and average the results.
10. Configure the regular sequence channel setting register to set the channels for conversion at each position and the number of positions for channel conversions in the regular sequence.
11. Configure ADC_RGLCHCFG3[17:15] to set the number of positions for channel conversions in the regular short sequence.

12. Set ADC_DGCTRL[0] to 1 to enable the ADC controller.
13. Set ADC_ANCTRL[0] to 1 to start the regular sequence conversion via software for one regular short sequence at a time.
14. Wait for the ADC regular sequence conversion to complete and read the data from the data register of the corresponding channel.
15. If multiple conversions are required, repeat steps 13 and 14. Once all channel positions have been converted, start conversion again from channel position 0.

35.9.4 Multi-channel A/D Conversion in Single Scan Mode of Injection Sequence

1. Configure the channel to be converted as an analog interface (GPIOx_MODE) based on the GPIO pin corresponding to the ADC input channel.
2. Configure the APB2 peripheral clock enable register (APB2CKENR) to enable the ADC clock.
3. Set ADC_ANCTRL[3] to 1 to power on the ADC.
4. Configure ADC_ANCTRL[5:4] to set the ADC positive reference voltage as the analog circuit power supply V_{DDA}.
5. Set ADC_DGCTRL[4] to 1 to set the injection sequence conversion mode to single trigger scan mode.
6. Set ADC_DGCTRL[22] to 1 to automatically clear the channel data after reading.
7. Configure ADC_CLKCTRL[27:24] to set the timing at which the channel selects to switch.
8. Configure ADC_CLKCTRL[15:0] to set the ADC sampling rate.
9. Configure the multiple average setting register to set the number of conversions for each channel and average the results.

10. Configure the injection sequence channel setting register to set the channels for conversion at each position and the number of positions for channel conversions in the injection sequence.
11. Set ADC_DGCTRL[0] to 1 to enable the ADC controller.
12. Set ADC_ANCTRL[1] to 1 to start the injection sequence conversion via software.
13. Wait for the ADC injection sequence conversion to complete and read the data from the data register of the corresponding channel.
14. If multiple conversions are required, repeat steps 12 and 13.

35.9.5 Analog Watchdog

1. Configure the channel to be converted as an analog interface (GPIOx_MODE) based on the GPIO pin corresponding to the ADC input channel.
2. Configure the APB2 peripheral clock enable register (APB2CKENR) to enable the ADC clock.
3. Set ADC_ANCTRL[3] to 1 to power on the ADC.
4. Configure ADC_ANCTRL[5:4] to set the ADC positive reference voltage as the analog circuit power supply V_{DDA}.
5. Set ADC_DGCTRL[3:2] to 1 to set the regular sequence conversion mode to single scan mode.
6. Set ADC_DGCTRL[22] to 1 to automatically clear the channel data after reading.
7. Configure ADC_CLKCTRL[27:24] to set the timing at which the channel selects to switch.
8. Configure ADC_CLKCTRL[15:0] to set the ADC sampling rate.
9. Configure the multiple average setting register to set the number of conversions for each channel and average the results.

10. Configure the regular sequence channel setting register to set the channels for conversion at each position and the number of positions for channel conversions in the regular sequence.
11. Configure the watchdog comparison condition register to set the upper and lower limits for the watchdog threshold.
12. Set ADC_DGCTRL[0] to 1 to enable the ADC controller.
13. Set ADC_INTEN[5] to 1 to enable the regular sequence watchdog alarm interrupt.
14. Set ADC_ANCTRL[0] to 1 to start the regular sequence conversion via software.
15. Wait for the ADC regular sequence conversion to complete; a watchdog interrupt will be generated if the conversion result exceeds the watchdog limits.
16. Write 1 to clear the ADC_INTSTAT[5] regular sequence watchdog alarm interrupt flag and continue the next conversion.

35.9.6 Differential Channel Input

1. Configure the channel to be converted as an analog interface (GPIOx_MODE) based on the GPIO pin corresponding to the ADC input channel.
2. Configure the APB2 peripheral clock enable register (APB2CKENR) to enable the ADC clock.
3. Set ADC_ANCTRL[3] to 1 to power on the ADC.
4. Configure ADC_ANCTRL[5:4] to set the ADC positive reference voltage as the analog circuit power supply V_{DDA}.
5. Set ADC_DGCTRL[3:2] to 1 to set the regular sequence conversion mode to single scan mode.
6. Set ADC_DGCTRL[22] to 1 to automatically clear the channel data after reading.

7. Configure ADC_CLKCTRL[27:24] to set the timing at which the channel selects to switch.
8. Configure ADC_CLKCTRL[15:0] to set the ADC sampling rate.
9. Configure the ADC_ANCTRL register to enable channel differential input.
10. Configure the multiple average setting register to set the number of conversions for each channel and average the results.
11. Configure the regular sequence channel setting register to set the channels for conversion at each position and the number of positions for channel conversions in the regular sequence.
12. Set ADC_DGCTRL[0] to 1 to enable the ADC controller.
13. Set ADC_ANCTRL[0] to 1 to start the regular sequence conversion via software.
14. Wait for the ADC regular sequence conversion to complete and read the differential data of the corresponding channels; for example, the differential data of channels 0 and 1 will be stored in the channel data register 0, and so on.
15. If multiple conversions are required, repeat steps 12 and 13.

35.9.7 Dual-ADC Cooperative Mode for Regular Sequence

1. Configure the channel to be converted as an analog interface (GPIOx_MODE) based on the GPIO pins corresponding to the ADC0/1 input channels.
2. Configure the APB2 peripheral clock enable register (APB2CKENR) to enable the ADC0/1 clock.
3. Set ADC_ANCTRL[3] to 1 to power on the ADC0/1.
4. Configure ADC_ANCTRL[5:4] to set the ADC0/1 positive reference voltage as the analog circuit power supply V_{DDA}.

5. Configure ADC_DGCTRL[3:2] to set the regular sequence conversion mode.
6. Set ADC_DGCTRL[22] to 1 to automatically clear the channel data after reading.
7. Configure ADC_CLKCTRL[27:24] to set the timing at which the channel selects to switch.
8. Configure ADC_CLKCTRL[15:0] to set the ADC0 sampling rate.
9. Configure the multiple average setting register to set the number of conversions for each channel and average the results.
10. Configure the regular sequence channel setting register to set the channels for conversion at each position and the number of positions for channel conversions in the regular sequence.
11. Set ADC_DGCTRL[0] to 1 to enable the ADC0/1 controller.
12. Configure ADC_DGCTRL[6:5] to set the cooperative mode for regular sequence conversion when ADC0/1 work cooperatively.
13. Set ADC_ANCTRL[0] to 1 to start the regular sequence conversion for ADC0 via software.
14. Wait for the ADC0/1 regular sequence conversion to complete and read the data from the data register of the corresponding ADC0/1 channels.
15. If multiple conversions are required, repeat steps 13 and 14.

35.9.8 Dual-ADC Cooperative Mode for Injection Sequence

1. Configure the channel to be converted as an analog interface (GPIOx_MODE) based on the GPIO pins corresponding to the ADC0/1 input channels.
2. Configure the APB2 peripheral clock enable register (APB2CKENR) to enable the ADC0/1 clock.
3. Set ADC_ANCTRL[3] to 1 to power on the ADC0/1.

4. Configure ADC_ANCTRL[5:4] to set the ADC0/1 positive reference voltage as the analog circuit power supply V_{DDA}.
5. Set ADC_DGCTRL[4] to 1 to set the injection sequence conversion mode to single trigger scan mode.
6. Set ADC_DGCTRL[22] to 1 to automatically clear the channel data after reading.
7. Configure ADC_CLKCTRL[27:24] to set the timing at which the channel selects to switch.
8. Configure ADC_CLKCTRL[15:0] to set the ADC0 sampling rate.
9. Configure the multiple average setting register to set the number of conversions for each channel and average the results.
10. Configure the injection sequence channel setting register to set the channels for conversion at each position and the number of positions for channel conversions in the injection sequence.
11. Set ADC_DGCTRL[0] to 1 to enable the ADC0/1 controller.
12. Configure ADC_DGCTRL[8:7] to set the cooperative mode for regular sequence conversion when ADC0/1 work cooperatively.
13. Set ADC_ANCTRL[1] to 1 to start the injection sequence conversion for ADC0 via software.
14. Wait for the ADC injection sequence conversion to complete and read the data from the data register of the corresponding channel.
15. If multiple conversions are required, repeat steps 13 and 14.

35.9.9 ADC Sampling via OPA Buffer

1. Set the OPA to the unit buffer mode and select FROMADC as the OPA positive channel selection signal SELP.

2. Configure the channel to be converted as an analog interface (GPIOx_MODE) based on the GPIO pin corresponding to the ADC input channel.
3. Configure the APB2 peripheral clock enable register (APB2CKENR) to enable the ADC clock.
4. Set ADC_ANCTRL[3] to 1 to power on the ADC.
5. Configure ADC_ANCTRL[5:4] to set the ADC positive reference voltage as the analog circuit power supply V_{DDA}.
6. Set ADC_DGCTRL[3:2] to 1 to set the regular sequence conversion mode to single scan mode.
7. Set ADC_DGCTRL[22] to 1 to automatically clear the channel data after reading.
8. Configure ADC_CLKCTRL[27:24] to set the timing at which the channel selects to switch.
9. Configure ADC_CLKCTRL[15:0] to set the ADC sampling rate.
10. Configure the multiple average setting register to set the number of conversions for each channel and average the results.
11. Configure the regular sequence channel setting register to set the channels for conversion at each position and the number of positions for channel conversions in the regular sequence.
12. Set ADC_ANCTRL[14] to 1 to enable the connection between the ADC and OPA.
13. Set ADC_DGCTRL[0] to 1 to enable the ADC controller.
14. Set ADC_ANCTRL[0] to 1 to start the regular sequence conversion via software.
15. Wait for the ADC regular sequence conversion to complete and read the data from the data register of the corresponding channel.
16. If multiple conversions are required, repeat steps 14 and 15.

36 Digital-to-analog Converter (DAC)

36.1 Overview

The digital-to-analog converter (DAC) can convert 12-bit digital signals into analog voltage output.

36.2 Main Features

- 8-bit or 12-bit digital input
- Simultaneous update of conversion data for two converters
- Generation of triangle waves and noise waves
- DMA operation
- External trigger for updating conversion data

36.3 System Block Diagram

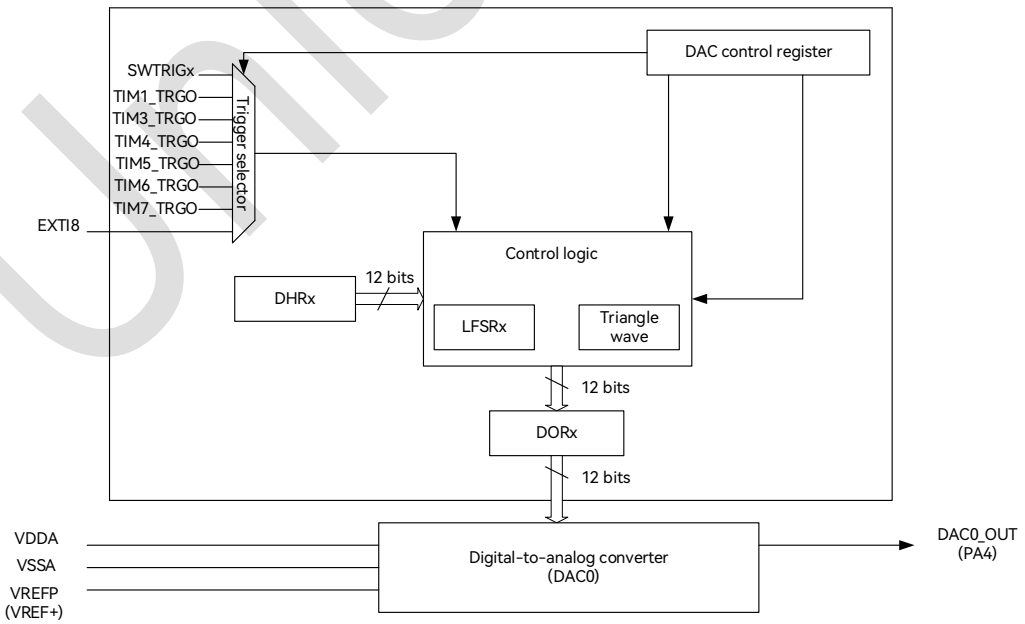


Figure 36-1: DAC System Block Diagram

36.4 Functional Description

36.4.1 DAC Conversion Control

A DAC channel can be independently enabled or disabled using the DACENx bit in the DAC_CTRL register. Once enabled, the DAC converts the data in DORx to an analog voltage output. The output voltage is given by VREF * (DOR / 4095).

The DAC output buffer can be enabled or disabled using the BUFx bit in the DAC_CTRL register, which can reduce the output impedance and directly drive certain external loads.

The DAC supports 8-bit or 12-bit data; when using 8-bit data, the lower 4 bits of DORx will automatically be loaded with 0.

36.4.2 Trigger Mode

The DAC supports 8 different trigger sources. When the trigger mode is enabled, the data in DHR will be loaded into DORx upon receiving a trigger event.

36.4.3 Noise Wave

The DAC can generate noise waves. When the noise wave mode is enabled, the LFSR will generate noise wave amplitudes in a specific manner. Each time a trigger event is detected, the LFSR value will be added to the value in DHR and then loaded into DORx. The high bits in the LFSR can be masked to control the noise amplitude.

36.4.4 Triangle Wave

The DAC can generate triangle waves. When the triangle wave mode is enabled, the triangle wave counter will generate triangle wave amplitudes within the range specified by MAMPx in the DAC_CTRL register. Each time a trigger event is detected, the triangle wave counter value will be added to the value in DHR and then loaded into DORx.

36.4.5 DMA

When the DMA mode is enabled, the DAC will generate a DMA request upon receiving a hardware trigger. The value written to the DHR register will be loaded into DORx after one cycle. If a hardware trigger is received while the previous DMA request has not been acknowledged, an underflow interrupt will be generated, and the DMA mode will be disabled.

36.5 Register Description

DAC register base address: 0x40B0_D000

The registers are listed below:

Table 36-1: List of DAC Registers

Offset Address	Name	Description
0x00	DAC_CTRL	Control register
0x04	DAC_SWTRG	Software trigger register
0x08	DAC_DHR12R0	DAC0 12-bit right-aligned data register
0x0C	DAC_DHR12L0	DAC0 12-bit left-aligned data register
0x10	DAC_DHR8R0	DAC0 8-bit right-aligned data register
0x2C	DAC_DOR0	DAC0 output data register
0x34	DAC_IS	Interrupt status register
0x38	DAC_CLK	DAC clock setting register
0x3C	DAC_PADEN	DAC pin enable register

36.5.1 Control Register (DAC_CTRL)

Offset address: 0x00

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:16	RSV	-	-	Reserved
15:14	DAC0_VREF_MODE	R/W	0	DAC0 reference voltage selection: 00/01: VREFP_AVDD (VDDA) 10: external input reference voltage V_{VREFP} or provided by the internal VREF module 11: reserved

Bit	Name	Attribute	Reset Value	Description
13	DMAIE0	R/W	0	DAC0 DMA underflow interrupt enable: 0: disabled 1: enabled
12	DMAEN0	R/W	0	DAC0 DMA mode enable: 0: enabled 1: enabled
11:8	MAMP0	R/W	0	Selection of DAC0 noise mode mask / triangle wave mode amplitude: 0000: no masking of LSFR bit [0] / triangle wave amplitude is 0 0001: no masking of LSFR bit [1:0] / triangle wave amplitude is 3 0010: no masking of LSFR bit [2:0] / triangle wave amplitude is 7 0011: no masking of LSFR bit [3:0] / triangle wave amplitude is 15 0100: no masking of LSFR bit [4:0] / triangle wave amplitude is 31 0101: no masking of LSFR bit [5:0] / triangle wave amplitude is 63 0110: no masking of LSFR bit [6:0] / triangle wave amplitude is 127 0111: no masking of LSFR bit [7:0] / triangle wave amplitude is 255 1000: no masking of LSFR bit [8:0] / triangle wave amplitude is 511 1001: no masking of LSFR bit [9:0] / triangle wave amplitude is 1023 1001: no masking of LSFR bit [10:0] / triangle wave amplitude is 2047 ≥ 1011: no masking of LSFR bit [11:0] / triangle wave amplitude is 4095
7:6	WAVE0	R/W	0	Select DAC0 generation of noise wave/triangle wave: 00: neither noise wave nor triangle wave generated 01: noise wave generated

Bit	Name	Attribute	Reset Value	Description
				10/11: triangle wave generated
5:3	TSEL0	R/W	0	DAC0 trigger source selection: 000: TIM5 TRGO 001: TIM7 TRGO 010: TIM2 TRGO 011: TIM4 TRGO 100: TIM1 TRGO 101: TIM3 TRGO 110: EXTI8 external interrupt pin 111: software trigger
2	TEN0	R/W	0	DAC0 trigger mode enable: 0: trigger disabled; data written to DACDHR starts conversion and output after one cycle. 1: trigger enabled; data written to DACDHR starts conversion and output after three cycles (for hardware trigger) / one cycle (for software trigger).
1	BUF0	R/W	0	DAC0 output buffer enable: 0: disabled 1: enabled
0	DACEN0	R/W	0	DAC0 enable: 0: disabled 1: enabled; a power-up delay is required when switching to the enabled state.

36.5.2 Software Trigger Register (DAC_SWTRG)

Offset address: 0x04

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:1	RSV	-	-	Reserved
0	SWTRG0	W	0x0	Writing 1 to this bit can generate a software trigger for DAC0, and this bit is cleared automatically.

36.5.3 DAC0 12-bit Right-aligned Data Register (DAC_DHR12R0)

Offset address: 0x08

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:12	RSV	-	-	Reserved
11:0	DHR0	R/W	0x0	Specify 12 bits of data for DAC0

36.5.4 DAC0 12-bit Left-aligned Data Register (DAC_DHR12L0)

Offset address: 0x0C

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:16	RSV	-	-	Reserved
15:4	DHR0	R/W	0x0	Specify 12 bits of data for DAC0
3:0	RSV	-	-	Reserved

36.5.5 DAC0 8-bit Right-aligned Data Register (DAC_DHR8R0)

Offset address: 0x10

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:8	RSV	-	-	Reserved
7:0	DHR0	R/W	0x0	Specify 8 bits of data for DAC0

36.5.6 DAC0 Output Data Register (DAC_DOR0)

Offset address: 0x2C

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:12	RSV	-	-	Reserved
11:0	DOR0	R	0x0	The 12-bit data being output by DAC0 can be read.

36.5.7 Interrupt Status Register (DAC_IS)

Offset address: 0x34

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:14	RSV	-	-	Reserved
13	IS	R/W1C	0x0	DMA underflow interrupt status of DAC0, can be cleared by writing 1 0: no interrupt occurred 1: interrupt occurred
12:0	RSV	-	-	Reserved

36.5.8 DAC Clock Setting Register (DAC_CLK)

Offset address: 0x38

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:9	RSV	-	-	Reserved
8	EXT	R/W	0x0	An additional APBCLK cycle is required for a low level.
7:0	DIV	R/W	0x0	Division factor For example, filling in decimal 82 can divide the frequency from 168 down to 1.00. Note: It is not set by default.

36.5.9 DAC Pin Enable Register (DAC_PADEN)

Offset address: 0x3C

Reset value: 0x0000 0001

Bit	Name	Attribute	Reset Value	Description
31:1	RSV	-	-	Reserved
0	PADEN	R/W	0x1	Select whether the DAC output is connected to an external pin: 1: output connected to the external pin 0: output not connected to the external pin, used inside the chip Note: It is recommended to set this bit to 0 when the DAC is not in use.

36.6 Operation Procedure

36.6.1 Direct Output of New Values

1. Configure PA4 as an analog interface and enable the DAC clock in the RCM module.
2. Set DAC_CTRL[2] to 0 to disable triggering.
3. Set DAC_CTRL[0] to 1 to enable DAC0. After enabling the DAC, a power-up delay is required (add a few milliseconds of delay in software).
4. Write the output value to the corresponding DAC_DHR register, and the respective channel will directly output the the voltage corresponding to the value.

36.6.2 Output Value Update Using Trigger Control

1. Configure PA4 as an analog interface and enable the DAC clock in the RCM module.
2. Set DAC_CTRL[2] to 1 to enable triggering.
3. Configure DAC_CTRL[5:3] to select the trigger source. Here, using software trigger as an example, configure it to 0x7.
4. Set DAC_CTRL[0] to 1 to enable DAC0. After enabling the DAC, a power-up delay is required (add a few milliseconds of delay in software).
5. Write the output value to the corresponding DAC_DHR register. Whenever a trigger event occurs, the respective channel will output the voltage corresponding to the value. In this example, DAC_SWTRG[0] is set to 1 to generate a software trigger.

36.6.3 Output Value Update Using DMA Control

1. Configure PA4 as an analog interface and enable the DAC clock in the RCM module.
2. Use TIM1 as the trigger source and configure TIM1 to generate a TRGO signal upon the update event.

3. Set DAC_CTRL[12] to 1 to enable the DMA mode for DAC.
4. Set DAC_CTRL[2] to 1 to enable triggering.
5. Configure DAC_CTRL[5:3] to select the trigger source. In this example, TIM1_TRGO is used as the trigger source, and it is set as 0x4.
6. Set DAC_CTRL[0] to 1 to enable DAC0. After enabling the DAC, a power-up delay is required (add a few milliseconds of delay in software).
7. Configure DMA0. For details on DMA controller configuration, please refer to chapter “11 DMA Controller”.
8. Whenever a trigger event occurs, the DAC will generate a DMA request, and the DMA will write the output value into the corresponding DAC_DHR register, causing the respective channel to output the corresponding voltage value. Here, TIM1 generates a TRGO signal upon each update event to trigger DMA transfer.

36.6.4 Noise Wave Generation

1. Configure PA4 as an analog interface and enable the DAC clock in the RCM module.
2. Set DAC_CTRL[7:6] to 0x1 to generate a noise wave.
3. Configure DAC_CTRL[11:8] to select the output amplitude in noise mode.
4. Set DAC_CTRL[2] to 1 to enable triggering.
5. Configure DAC_CTRL[5:3] to select the trigger source.
6. Set the corresponding DAC_DHR register to specify the initial output value.
7. Set DAC_CTRL[0] to 1 to enable DAC0. After enabling the DAC, a power-up delay is required (add a few milliseconds of delay in software).

8. Whenever a trigger event occurs, the corresponding channel will output a noise wave with certain variations.

36.6.5 Triangle Wave Generation

1. Configure PA4 as an analog interface and enable the DAC clock in the RCM module.
2. Set DAC_CTRL[7:6] to 0x2 to generate a triangle wave.
3. Configure DAC_CTRL[11:8] to select the output amplitude of the triangle wave.
4. Set DAC_CTRL[2] to 1 to enable triggering.
5. Configure DAC_CTRL[5:3] to select the trigger source.
6. Set the corresponding DAC_DHR register to specify the initial output value.
7. Set DAC_CTRL[0] to 1 to enable DAC0. After enabling the DAC, a power-up delay is required (add a few milliseconds of delay in software).

Whenever a trigger event occurs, the corresponding channel will output a triangle wave with certain variations.

37 Analog Comparator (ACMP)

37.1 Overview

The device embeds three independent analog comparators (ACMP0, ACMP1, ACMP2).

The ACMP module is used to compare the magnitudes of two input analog voltages and output a high or low logic level to the internal chip based on the comparison result, which can generate an interrupt. When the voltage at the “INP” input pin is higher than that at the “INN” input pin, the comparator outputs a high level; when the voltage at the “INP” input pin is lower than that at the “INN” input pin, the comparator outputs a low level.

37.2 Main Features

- The analog comparator output can generate interrupts, and the interrupt generation method is programmable.
- Up to three independent comparators
- Configurable hysteresis voltage
- Input channels can be selected from I/O ports, DAC output channels, internal reference voltage VBG, VDDH, and VREF
- The system can be woken up from sleep mode via interrupt generation.

37.3 System Block Diagram

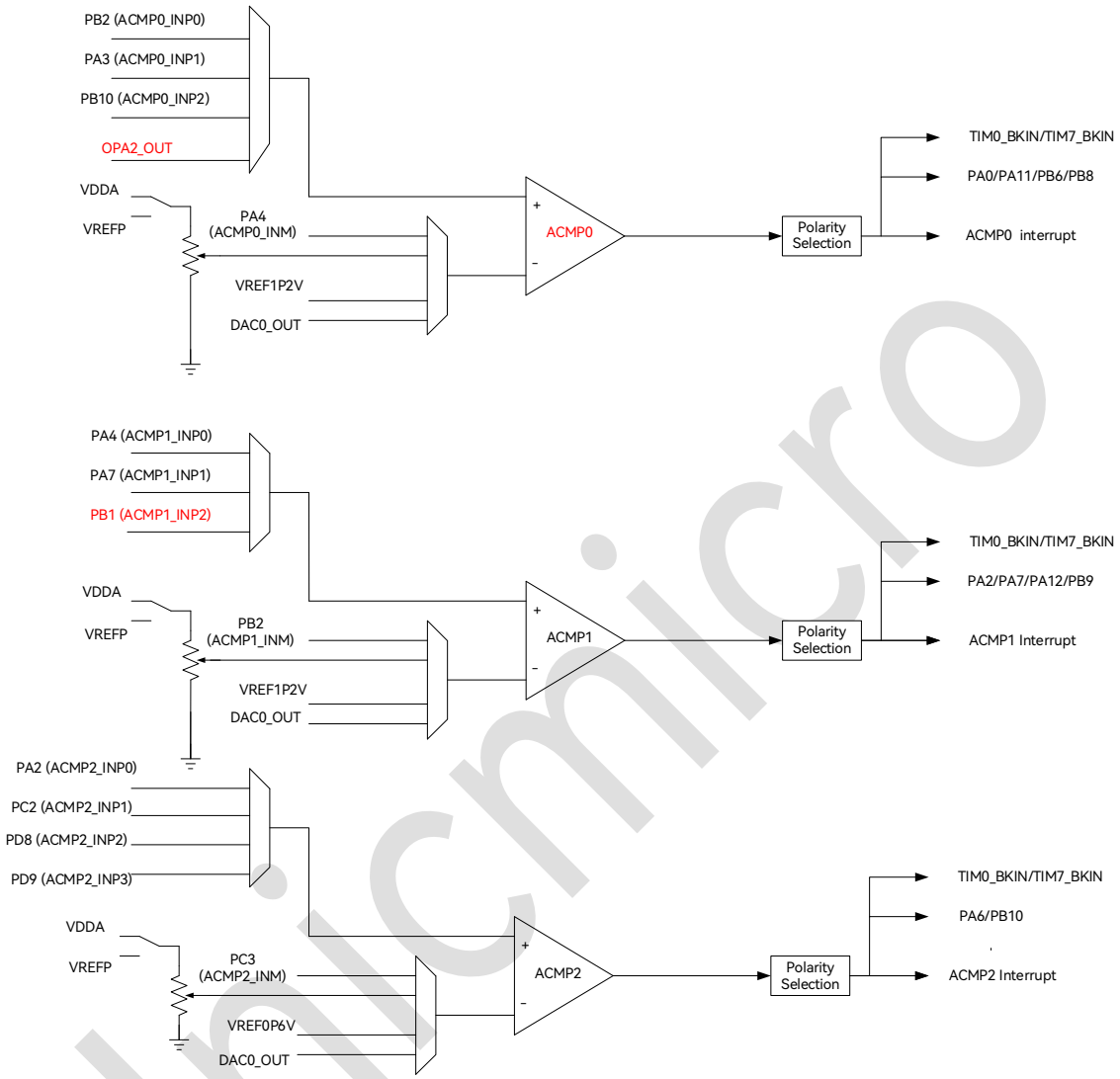


Figure 37-1: ACMP System Block Diagram

37.4 Functional Description

37.4.1 Voltage Comparator

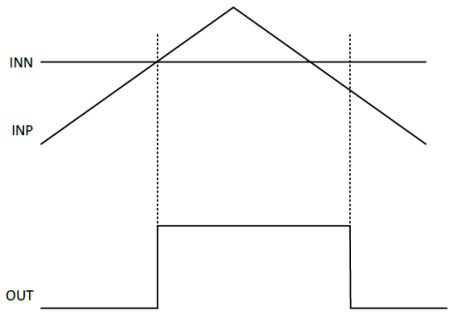


Figure 38-1: Basic Function Diagram of Comparator

When configuring the independent comparator function, if the positive input INP voltage is greater than the negative input INN voltage, the comparator outputs high; otherwise, it outputs low.

37.4.2 Comparator Hysteresis Function

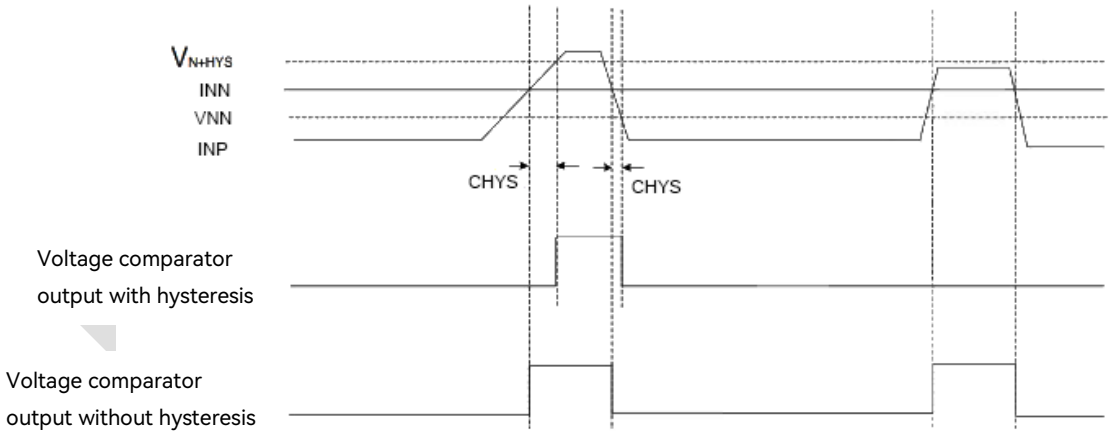


Figure 38-2: Voltage Comparison Function with Hysteresis

The hysteresis function of the comparator is configured by setting ACMP_REG.CHYS. When the hysteresis voltage is set, the comparator outputs high when the positive input INP voltage is greater than the negative input INN voltage plus the hysteresis voltage CHYS. Conversely,

the comparator outputs low when the positive input voltage is less than the negative input voltage minus the hysteresis voltage CHYS.

37.5 Register Description

Register base address: 0x4008_0000

The registers are listed below:

Table 37-1: List of ACMP Registers

Offset Address	Register Name	Description
0x00	ACMP_UNLOCK	Write unlock register
0x1C	ACMP_CFG0	Analog comparator 0 register
0x20	ACMP_CFG1	Analog comparator 1 register
0x24	ACMP_CFG2	Analog comparator 2 register

37.5.1 Write Unlock Register (ACMP_UNLOCK)

Offset address: 0x00

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:1	RSV	-	-	Reserved
0	UNLOCK	R/W	0	Write unlock register Writing 0xA5A55A5A sets this bit to 1 to unlock the writing to other registers. Writing any other value will relock the register to prevent misoperation.

37.5.2 Analog Comparator 0 Register (ACMP_CFG0)

Offset address: 0x1C

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31	RSV	-	-	Reserved
30	DB_EN	R/W	0	Filter enable: 0: filter disabled 1: filter enabled

Bit	Name	Attribute	Reset Value	Description
29:20	DB_LENGTH	R/W	0	Filter length
19	RSV	-	-	Reserved
18	INTS	R/W1C	0	Interrupt status, can be cleared by writing 1
17:16	INT_CFG	R/W	0	Control interrupt enable: 00: interrupt disabled 01: interrupt generated on any edge of OUTPUT 10: interrupt generated on falling edge of OUTPUT 11: interrupt generated on rising edge of OUTPUT
15:14	RSV	-	-	Reserved
13	OUTPUT	R	0	Output signal of comparator 0
12	CRVINCTRL	R/W	0	Negative input resistor divider input selection: 0: VDDA 1: VREF
11:8	CRVCTRL	R/W	0	Negative input resistor divider ratio: The voltage divider output is (1/6 + CRVCTRL/24) * (VDDA or VREF).
7:6	CNEGSEL	R/W	0	Negative input channel selection: 0: CIN (PA4) 1: VDDA / VREF 2: VBG (1.2 V) 3: DACOUT (DAC output)
5:4	CPOSSEL	R/W	0	Positive input channel selection: 0: CIP 0 (PB2) 1: CIP 1 (PA3) 2: CIP (PB10) 3: CIP 3 (OPA2_OUT)
3:2	CHYS	R/W	0	Hysteresis voltage selection: 0: 0 mV 1: 10 mV 2: 20 mV 3: 30 mV
1	CLPM	R/W	0	Low-power mode enable:

Bit	Name	Attribute	Reset Value	Description
				0: low-power mode disabled 1: low-power mode enabled Note: For general applications, it is recommended to enable the low-power mode.
0	EN	R/W	0	ACMP0 enable: 0: disabled 1: enabled

37.5.3 Analog Comparator 1 Register (ACMP_CFG1)

Offset address: 0x20

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31	RSV	-	-	Reserved
30	DB_EN	R/W	0	Filter enable: 0: filter disabled 1: filter enabled
29:20	DB_LENGTH	R/W	0	Filter length
19	RSV	-	-	Reserved
18	INTS	R/W1C	0	Interrupt status, can be cleared by writing 1
17:16	INT_CFG	R/W	0	Control interrupt enable: 00: interrupt disabled 01: interrupt generated on any edge of OUTPUT 10: interrupt generated on falling edge of OUTPUT 11: interrupt generated on rising edge of OUTPUT
15:14	RSV	-	-	Reserved
13	OUTPUT	R	0	Output signal of comparator 1
12	CRVINCTRL	R/W	0	Negative input resistor divider input selection: 0: VDDA 1: VREF
11:8	CRVCTRL	R/W	0	Negative input resistor divider ratio: The voltage divider output is (1/6 + CRVCTRL/24)

Bit	Name	Attribute	Reset Value	Description
				* (VDDA or VREF).
7:6	CNEGSEL	R/W	0	Negative input channel selection: 0: CIN (PB2) 1: VDDA / VREF 2: VBG (1.2 V) 3: DACOUT (DAC output)
5:4	CPOSSEL	R/W	0	Positive input channel selection: 0: CIP 0 (PA4) 1: CIP 1 (PA7) 2: CIP (PB1) 3: CIP 3 (not supported)
3:2	CHYS	R/W	0	Hysteresis voltage selection: 0: 0 mV 1: 10 mV 2: 20 mV 3: 30 mV
1	CLPM	R/W	0	Low-power mode enable: 0: low-power mode disabled 1: low-power mode enabled
0	EN	R/W	0	ACMP1 enable: 0: disabled 1: enabled

37.5.4 Analog Comparator 2 Register (ACMP_CFG2)

Offset address: 0x24

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31	RSV	-	-	Reserved
30	DB_EN	R/W	0	Filter enable: 0: filter disabled 1: filter enabled
29:20	DB_LENGTH	R/W	0	Filter length
19	RSV	-	-	Reserved

Bit	Name	Attribute	Reset Value	Description
18	INTS	R/W1C	0	Interrupt status, can be cleared by writing 1
17:16	INT_CFG	R/W	0	Control interrupt enable: 00: interrupt disabled 01: interrupt generated on any edge of OUTPUT 10: interrupt generated on falling edge of OUTPUT 11: interrupt generated on rising edge of OUTPUT
15:14	RSV	-	-	Reserved
13	OUTPUT	R	0	Output signal of comparator 2
12	CRVINCTRL	R/W	0	Negative input resistor divider input selection: 0: VDDA 1: VREF
11:8	CRVCTRL	R/W	0	Negative input resistor divider ratio: The voltage divider output is (1/6 + CRVCTRL/24) * (VDDA or VREF).
7:6	CNEGSEL	R/W	0	Negative input channel selection: 0: CIN (PC3) 1: VDDA / VREF 2: VBG (1.2 V) 3: DACOUT (DAC output)
5:4	CPOSSEL	R/W	0	Positive input channel selection: 0: CIP 0 (PA2) 1: CIP 1 (PC2) 2: CIP (PD8) 3: CIP 3 (PD9)
3:2	CHYS	R/W	0	Hysteresis voltage selection: 0: 0 mV 1: 10 mV 2: 20 mV 3: 30 mV
1	CLPM	R/W	0	Low-power mode enable: 0: low-power mode disabled 1: low-power mode enabled
0	EN	R/W	0	ACMP2 enable: 0: disabled 1: enabled

37.6 Operation Procedure

1. Configure the analog GPIO used by the ACMP.
2. Configure the unlock register ACMP_UNLOCK, and write 0xA5A55A5A once to unlock it.
3. Configure the ACMP input signal selection and set whether to enable the hysteresis voltage.
4. Decide whether to configure the ACMP interrupt signal, and clear the interrupt flag in the interrupt service routine.
5. Enable the ACMP module to operate normally.
6. When INP > INN, check whether the comparator output signal is 1.

38 Operational Amplifier (OPA)

38.1 Overview

This device features three independent operational amplifiers (OPA0, OPA1, OPA2) that can be configured as pure operational amplifiers, comparators, unity buffers, or programmable gain amplifiers (PGA). Operational amplifiers are widely used linear integrated circuits, with the most common applications in audio systems.

38.2 Main Features

- Configurable in multiple operating modes: operational amplifier / comparator / unity buffer / PGA
- Typical operating current: 3 mA
- Operating temperature: -40~105℃

38.3 System Block Diagram

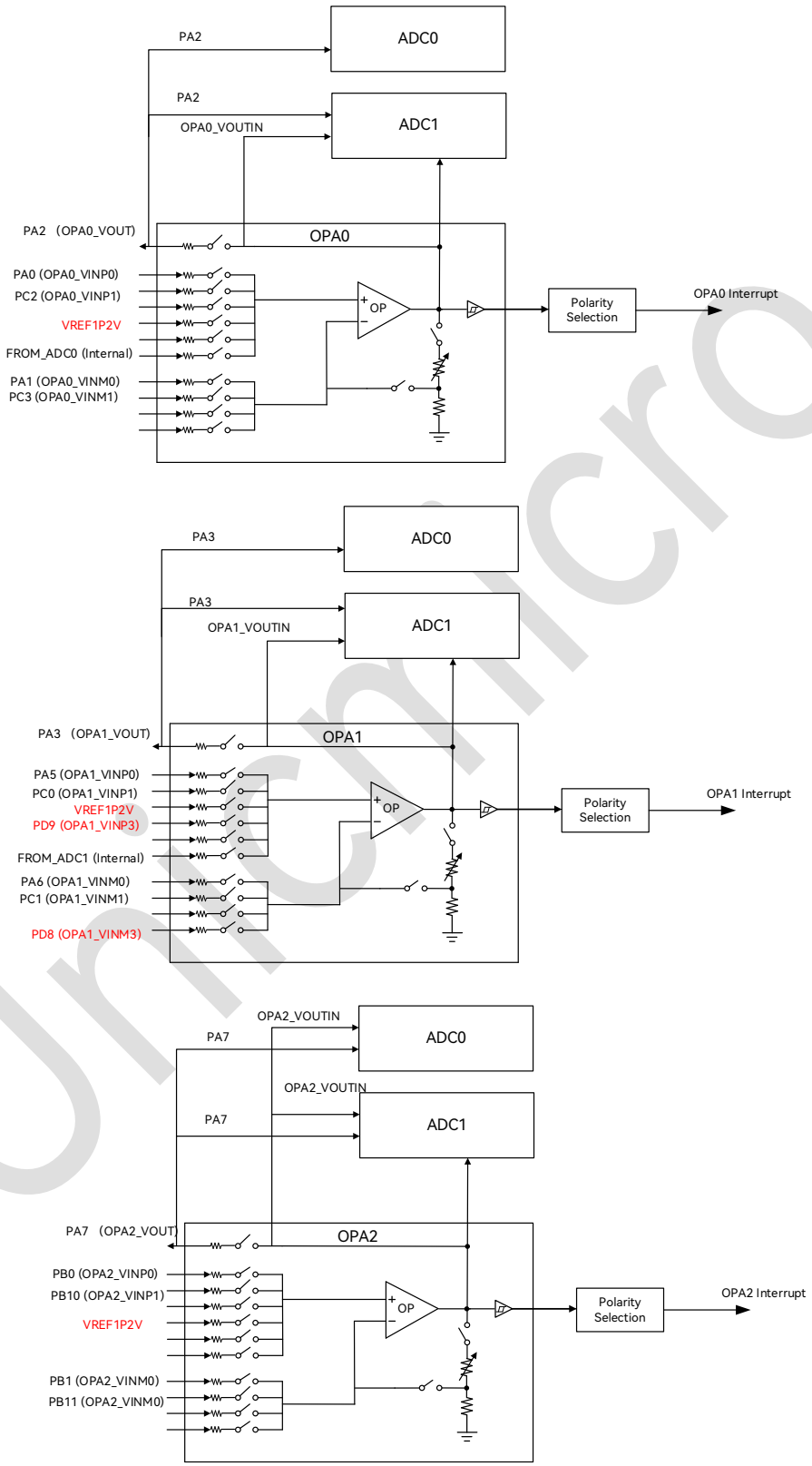


Figure 38-1: OPA System Block Diagram

38.4 Register Description

OPA register base address: 0x4008_0000

The registers are listed below:

Table 38-1: List of OPA Registers

Offset Address	Register Name	Description
0x00	OPA_UNLOCK	Write unlock register
0x10	OPA0_CFG	OPA0 setting register
0x14	OPA1_CFG	OPA1 setting register
0x18	OPA2_CFG	OPA2 setting register

38.4.1 Write Unlock Register (OPA_UNLOCK)

Offset address: 0x00

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:1	RSV	-	-	Reserved
0	UNLOCK	R/W	0	Write unlock register Writing 0xA5A55A5A sets this bit to 1 to unlock the writing to other registers. Writing any other value will relock the register to prevent misoperation.

38.4.2 OPA0 Setting Register (OPA0_CFG)

Offset address: 0x10

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:19	RSV	-	-	Reserved
18	INT	R/W1C	0	Interrupt status, can be cleared by writing 1
17:16	INT_CFG	R/W	0	Control interrupt enable: 00: disabled 01: interrupt generated on any edge of COMPOUT 10: interrupt generated on falling edge of

Bit	Name	Attribute	Reset Value	Description
				COMPOUT 11: interrupt generated on rising edge of COMPOUT
15	RSV	-	-	Reserved
14	COMPOUT	R	0	Output state of OPA as a comparator
13	COMPEN	R/W	0	Comparator function enable: 0: disabled 1: enabled
12:10	GAINSEL	R/W	0	Single-ended PGA gain selection signal: 000: 1X 001: 2X 010: 4X 011: 8X 100: 16X 101: 32X 110: 64X 111: reserved Note: If a gain of 16X/32X/64X is selected, an external feedback resistor is required.
9:7	SELP	R/W	0	OPA positive channel selection signal (SELP[2:0]): 000: input from pin PA0 001: input from pin PC2 010: reserved 011: 1.2V VREF voltage 100: connected from ADC MUX output Others: reserved
6:4	SELN	R/W	0	OPA negative channel selection signal (SELN[2:0]): 000: input from pin PA1 001: input from pin PC3 010: analog ground 011: analog ground Others: none of the above (nothing selected)
3	OTPEN	R/W	0	Enabling OPA output to IO pin: 0: OPA output not connected to IO pin

Bit	Name	Attribute	Reset Value	Description
				1: OPA output connected to IO pin PA2
2	CAPEN	R/W	0	PGA internal feedback capacitor enable signal: 0: internal feedback capacitor disabled 1: internal feedback capacitor enabled
1	FBRESEN	R/W	0	PGA internal feedback resistor enable signal: 0: internal feedback resistor disabled 1: internal feedback resistor enabled
0	EN	R/W	0	OPA module enable signal: 0: disabled 1: enabled

38.4.3 OPA1 Setting Register (OPA1_CFG)

Offset address: 0x14

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:19	RSV	-	-	Reserved
18	INT	R/W1C	0	Interrupt status, can be cleared by writing 1
17:16	INT_CFG	R/W	0	Control interrupt enable: 00: interrupt disabled 01: interrupt generated on any edge of COMPOUT 10: interrupt generated on falling edge of COMPOUT 11: interrupt generated on rising edge of COMPOUT
15	RSV	-	-	Reserved
14	COMPOUT	R	0	Output state of OPA as a comparator
13	COMPEN	R/W	0	Comparator function enable: 0: disabled 1: enabled
12:10	GAINSEL	R/W	0	PGA gain selection signal: 000: 1X 001: 2X 010: 4X

Bit	Name	Attribute	Reset Value	Description
				011: 8X 100: 16X 101: 32X 110: 64X 111: reserved Note: If a gain of 16X/32X/64X is selected, an external feedback resistor is required.
9:7	SELP	R/W	0	OPA positive channel selection signal (SELP[2:0]): 000: input from pin PA5 001: input from pin PC0 010: reserved 011: input from pin PD9 100: connected from ADC MUX output Others: reserved
6:4	SELN	R/W	0	OPA negative channel selection signal (SELN[2:0]): 000: input from pin PA6 001: input from pin PC1 010: analog ground 011: input from pin PD8 Others: none of the above (nothing selected)
3	OTPEN	R/W	0	Enabling OPA output to IO pin: 0: OPA output not connected to IO pin 1: OPA output connected to IO pin (PA3)
2	CAPEN	R/W	0	PGA internal feedback capacitor enable signal: 0: internal feedback capacitor disabled 1: internal feedback capacitor enabled
1	FBRESEN	R/W	0	PGA internal feedback resistor enable signal: 0: internal feedback resistor disabled 1: internal feedback resistor enabled
0	EN	R/W	0	OPA module enable signal: 0: disabled 1: enabled

38.4.4 OPA2 Setting Register (OPA2_CFG)

Offset address: 0x18

Reset value: 0x0000 0000

Bit	Name	Attribute	Reset Value	Description
31:19	RSV	-	-	Reserved
18	INT	R/W1C	0	Interrupt status, can be cleared by writing 1
17:16	INT_CFG	R/W	0	Control interrupt enable: 00: interrupt disabled 01: interrupt generated on any edge of COMPOUT 10: interrupt generated on falling edge of COMPOUT 11: interrupt generated on rising edge of COMPOUT
15	RSV	-	-	Reserved
14	COMPOUT	R	0	Output state of OPA as a comparator
13	COMPEN	R/W	0	Comparator function enable: 0: disabled 1: enabled
12:10	GAINSEL	R/W	0	PGA gain selection signal: 000: 1X 001: 2X 010: 4X 011: 8X 100: 16X 101: 32X 110: 64X 111: reserved Note: If a gain of 16X/32X/64X is selected, an external feedback resistor is required.
9:7	SELP	R/W	0	OPA positive channel selection signal (SELP[2:0]): 000: input from pin PB0 001: input from pin PB10 010: reserved 011: reserved

Bit	Name	Attribute	Reset Value	Description
				100: connected from ADC MUX output Others: reserved
6:4	SELN	R/W	0	OPA negative channel selection signal (SELN[2:0]): 000: input from pin PB1 001: input from pin PB11 010: analog ground 011: analog ground Others: none of the above (nothing selected)
3	OTPEN	R/W	0	Enabling OPA output to IO pin: 0: OPA output not connected to IO pin 1: OPA output connected to IO pin (PA7)
2	CAPEN	R/W	0	PGA internal feedback capacitor enable signal: 0: internal feedback capacitor disabled 1: internal feedback capacitor enabled
1	FBRESEN	R/W	0	PGA internal feedback resistor enable signal: 0: internal feedback resistor disabled 1: internal feedback resistor enabled
0	EN	R/W	0	OPA module enable signal: 0: disabled 1: enabled

38.5 Operation Procedure

38.5.1 UNITBUFF Mode

1. Multiplex the OPA-related GPIO for analog function.
2. Configure the unlock register OPA_UNLOCK, and write 0xA5A55A5A once to unlock it.
3. Set OPAx_CFG[1] to 1, OPAx_CFG[2] to 1, OPAx_CFG[3] to 1, OPAx_CFG[12:10] to 0, and OPAx_CFG[13] to 0.
4. Configure the OPAx_CFG register, and set SELN to 0 to select the SELP input signal.
5. Set OPAx_CFG[0] to 1 to enable normal operation of the OPA.

38.5.2 OPA Mode

1. Multiplex the OPA-related GPIO for analog function.
2. Configure the unlock register OPA_UNLOCK, and write 0xA5A55A5A once to unlock it.
3. Set OPAx_CFG[1] to 0, OPAx_CFG[2] to 0, OPAx_CFG[3] to 1, OPAx_CFG[12:10] to 7, and OPAx_CFG[13] to 0.
4. Configure the OPAx_CFG register, and set SELN to 0 to select the SELP input signal.
5. Set OPAx_CFG[0] to 1 to enable normal operation of the OPA.

38.5.3 External Feedback Circuit Setup in OPA Mode

1. Multiplex the OPA-related GPIO for analog function.
2. Configure the unlock register OPA_UNLOCK, and write 0xA5A55A5A once to unlock it.
3. Set OPAx_CFG[1] to 0, OPAx_CFG[2] to 0, OPAx_CFG[3] to 1, OPAx_CFG[12:10] to 7, and OPAx_CFG[13] to 0.
4. Configure the OPAx_CFG register, and set SELN to 0 to select the SELP input signal.
5. Set bit 0 to 1 to enable normal operation of the OPA.
6. Connect the external circuit according to the diagram.

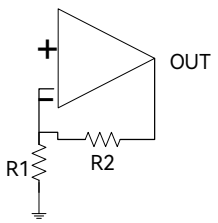


Figure 39-1: External Amplifier Circuit Connection Diagram

Note: Gain = (R1 + R2) / R1

7. Check whether the output voltage is amplified according to the desired gain.

38.5.4 PGA Mode

1. Multiplex the OPA-related GPIO for analog function.
2. Configure the unlock register OPA_UNLOCK, and write 0xA5A55A5A once to unlock it.
3. Set OPAx_CFG[1] to 1, OPAx_CFG[2] to 1, OPAx_CFG[3] to 1, and OPAx_CFG[13] to 0.
4. Select the gain factor by setting OPAx_CFG[12:10].
5. Configure the OPAx_CFG register, and set SELN to 7 to select the SELP input signal.
6. Set OPAx_CFG[0] to 1 to enable normal operation of the OPA.

38.5.5 CMP Mode

1. Multiplex the OPA-related GPIO for analog function.
2. Configure the unlock register OPA_UNLOCK, and write 0xA5A55A5A once to unlock it.
3. Set OPAx_CFG[1] to 0, OPAx_CFG[2] to 0, OPAx_CFG[3] to 0, OPAx_CFG[12:10] to 7, and OPAx_CFG[13] to 1.
4. Configure the OPAx_CFG register and set SELN to 0.
5. Configure the interrupt trigger method and decide whether to enable the interrupt [if interrupts are enabled, the interrupt signal must be cleared].
6. Set OPAx_CFG[0] to 1 to enable normal operation of the OPA.
7. Check if OPAx_CFG[14] is set under the specified conditions.

39 Internal Reference Voltage Source (VREF)

39.1 Overview

The built-in independent voltage reference source (VREF) can output voltages of 1.5 V, 2.0 V, 2.5 V or 3.0 V.

39.2 Register Description

VREF register base address: 0x4008_0000

The registers are listed below:

Table 39-1: List of VREF Registers

Offset Address	Register Name	Description
0x00	VREF_UNLOCK	Write unlock register
0x28	TS_VREFCFG	TSVREF setting register

39.2.1 Write Unlock Register (VREF_UNLOCK)

Offset address: 0x00

Reset value: 0x0000 000

Bit	Name	Attribute	Reset Value	Description
31:1	RSV	-	-	Reserved
0	UNLOCK	R/W	0	**Write unlock register** Writing 0xA5A55A5A sets this bit to 1 to unlock the writing to other registers. Writing any other value will relock the register to prevent misoperation.

39.2.2 TSVREF Setting Register (TS_VREFCFG)

Offset address: 0x28

Reset value: 0x0000 0EF1

Bit	Name	Attribute	Reset Value	Description
31:14	RSV	-	-	Reserved
13:4	CHOP_CLK_DIV	R/W	10'hEF	Chopper clock divider setting: Chopper clock frequency = System clock / CHOP_CLK_DIV
3	CHOP_CLK_EN	R/W	0	Chopper clock enable signal: 0: chopper clock disabled 1: chopper clock enabled
2:1	VREF_SEL	R/W	0	VREF voltage selection: 00: 1.5 V 01: 2.0 V 10: 2.5 V 11: 3.0 V
0	PD	R/W	1	VREF enable signal: 0: normal operating mode 1: power-down mode

39.3 Operation Procedure

1. Configure the unlock register VREF_UNLOCK, and write 0xA5A55A5A once to unlock it.
2. Set TS_VREFCFG[3] to enable the chopper clock.
3. Configure TS_VREFCFG[13:4] to set the chopper clock division value.
4. Configure TS_VREFCFG[2:1] to set the VREF output voltage.
5. Configure TS_VREFCFG[0] to enable VREF to operate normally.
6. Check the output voltage on the VREFP pin.

40 Unique Device Serial Number (UID)

40.1 Overview

The MCU series products are equipped with a 128-bit unique device serial number (UID), which is stored in the system configuration block of the flash memory. The information contained within the UID is programmed at the factory and is guaranteed to be unique for any MCU microcontroller under any circumstances. The user application or external devices can read it through the CPU or JTAG/SWD interface, and it cannot be modified.

40.2 UID Register

Starting address: 0x0400_1D48, with a length of 128 bits.

41 Debug Support (DBG)

The Cortex®-M4 core integrates a hardware debug module. It supports instruction breakpoints (stopping when an instruction is fetched) and data breakpoints (stopping when data is accessed). When the core is halted, users can view the internal state of the core and the external state of the system. Upon completion of the query operations, the core and peripherals can be restored to continue executing the corresponding program. The hardware debugging module of the chip core is available when connected to a debugger (unless disabled).

The following debug interfaces are supported:

- Serial interface (two-wire SWD)
- JTAG debug interface (4-wire or 5-wire JTAG)

42 Revision History

Version	Date	Modifications
V1.0	-	Initial release.
V1.1	-	1. Modified some descriptions in the “Main Features” section. 2. Modified some register descriptions. 3. Modified some descriptions in the “Timer Operating Mode” section. 4. Modified some descriptions in the “DMA Access” section.
V1.1.1	-	1. Updated the maximum clock frequency. 2. Updated the parameters in tables regarding LVD (VDT) settings, BOR settings, and PDR settings.
V1.2	-	1. Modified some descriptions regarding GPIO_PULL and ACMP_CFG0 registers. 2. Modified some descriptions regarding I2C_TXABRTSRC and DAC_PADEN registers. 3. Added the “24.4.14 Test Mode” section. 4. Updated the operating temperature range. 5. Updated the descriptions for entries 13 and 14 in the “EFC One-Time Programming (OTP) Area” table.
V1.3	Sep-20-2024	1. Updated the notes below the table in “6.3.2 PLL0 Configuration Register 0 (RCM_PLL0CFGR0)”. 2. Added the range of θ to the table “13-1: CORDIC Controller Function Mode 0 Input/Output”. 3. Modified the description of the CTL register in Chapter 11 DMA. 4. Modified some register descriptions, mainly in Chapters 16, 26, 29, and 31. 5. Added sections “16.6.12 Asynchronous PWM Mode” and “16.6.13 Combined PWM Mode.” 6. Removed PA10 from the alternated pins for I2C1_SDA in Table "26-1: I2C1 & I2C2 Pin Description.” 7. Added EP0-EP4 memory access entries in sections 34.4.25-34.4.29. 8. Updated some timer timing diagrams in Chapters 16-19. 9. Added the section “35.7 Temperature Measurement”. 10. Added a new Chapter “40 Unique Device Serial Number (UID)”. 11. Changed the filename from “UM32x42x User Manual” to

Version	Date	Modifications
		“UM32G421 User Manual.”
V1.3.1	Aug-14-2025	1. Added note for bit 1 in “5.3.4 PDR/BOR/LVD Configuration Register (PMU_VDCR)”. 2. Modified the status of bits 22 and 23 to “reserved” in “6.3.7 Clock Configuration Register 1”. 3. Deleted the GPIO_IPE register in “7.4.15 Configuration Lock Register”. 4. Corrected typographical errors in "12 DMA Request Multiplexer (DMAMUX)". 5. Added descriptions on CAN baud rate calculation, updated frame format diagrams and optimized some wordings in “24 CANFD Bus Controller (CANFD)”. 6. Modified some descriptions in “26 Enhanced Inter-integrated Circuit Interface (I2C1 & I2C2)”. 7. Deleted ENDTX, ENDRX, hardware handshaking and descriptions thereof, and optimized some wordings in “31 Universal Synchronous / Asynchronous Receiver Transmitter (USART6 & USART7)”. 8. Optimized some wordings in “32 Serial Peripheral Interface (SPI0 & SPI1)”. 9. Modified descriptions related to bit 6, bit 5 and bit 0 in “34.4.1 Working Mode Register (USB_WORKINGMODE)”. 10. Added the descriptions for bits [31:21] in “35.8.5 Multiple Average Setting Register 1”. 11. Modified the descriptions for bits [9:7] and bits [6:4] in “38.4.3 OPA1 Setting Register” to “Input from pin PD9” and “Input from pin PD8” respectively.

