ARM®-based 32-bit Cortex®-M0+ MCU with 16 to 64 KB Flash, sLib, 10 timers, ADC, 10 communication interfaces (1 CAN)

- **Core: ARM® 32-bit Cortex®-M0+ CPU**
  - 80 MHz maximum frequency, with a Memory Protection Unit (MPU), single-cycle multiplication
- **Memories**
  - 16 to 64 Kbytes of Flash memory
  - 4 Kbytes of boot memory used as a Bootloader or as a general instruction/data memory (one-time-configured)
  - sLib: configurable part of main Flash set as a library area with code executable but secured, non-readable
  - 9 Kbytes of SRAM (8 Kbytes with parity check)
- **Power control (PWC)**
  - 1.71 ~ 3.6 V power supply
  - Power-on reset (POR), low voltage reset (LVR), power voltage monitor (PVM)
  - Low power modes: Sleep, Deepsleep and Standby modes; 5 WKUP pins for wakeup from Standby mode
  - 5 x 32-bit battery powered registers (BPR)
- **Clock, reset and power management**
  - 4 to 25 MHz crystal oscillator (HEXT)
  - Internal 48 MHz factory-trimmed HICK
  - 32 kHz crystal (LEXT)
  - Low-speed internal clock (LICK)
- **Analog**
  - 1 x 12-bit 2 MSPS A/D converter, up to 15 external input channels, 12-bit/10-bit/8-bit/6-bit configurable resolution, hardware oversampling up to equivalent 16-bit resolution
  - Internal reference voltage (V_{INTRV})
- **Up to 39 fast GPIOs**
  - Core dedicated single-cycle GPIO bus
  - All mappable on external interrupts
  - Almost all 5 V-tolerant
- **DMA**
  - 1 x 5-channel DMA controller
- **Up to 10 timers**
  - 1 x 16-bit 7-channel advanced timer, including 3 pairs of complementary channels for PWM output, with dead-time generator and emergency brake
  - Up to 5 x 16-bit general-purpose timers, each with 4 IC/OC/PWM or pulse counter and quadrature (incremental) encoder input
  - 1 x 16-bit basic timer
  - 2 x watchdog timers (general WDT and windowed WWDT)
  - SysTick timer: a 24-bit downcounter
- **ERTC: enhanced RTC, with alarm, subsecond precision, hardware calendar, and calibration**
- **Up to 10 communication interfaces**
  - 2 x I^2C interfaces (SMBus/PMBus) supporting fast mode plus (1 MHz), with wakeup from Deepsleep mode
  - 4 x USARTs, supporting synchronous SPI and modem control; with ISO7816 interface, LIN, IrDA and RS485 driver enable, TX/RX swap
  - 2 x SPIs (36 MHz), all with multiplexed half-duplex I^2S;
  - CAN interface (2.0B Active), with 256 bytes of dedicated buffer
  - Infrared transmitter (IRTMR)
- **96-bit ID (UID)**
- **Debug mode**
  - Serial wire debug (SWD) and serial wire output (SWO)
- **Temperature range: -40 to +105 °C**
- **Packaging**
  - LQFP48 7 x 7 mm
  - LQFP32 7 x 7 mm
  - QFN32 5 x 5 mm
  - QFN32 4 x 4 mm
  - QFN28 4 x 4 mm
  - QFN20 3 x 3 mm
  - TSSOP20 6.5 x 4.4 mm
- **List of Models**

Internal Flash	Model
64 KB	AT32L021C8T7, AT32L021K8T7, AT32L021K8U7, AT32L021K8U7-4 AT32L021F8P7, AT32L021F8U7, AT32L021G8U7
32 KB	AT32L021C6T7, AT32L021K6T7, AT32L021K6U7, AT32L021K6U7-4 AT32L021F6P7, AT32L021F6U7, AT32L021G6U7
16 KB	AT32L021C4T7, AT32L021K4T7, AT32L021K4U7, AT32L021K4U7-4 AT32L021F4P7, AT32L021F4U7, AT32L021G4U7

Contents

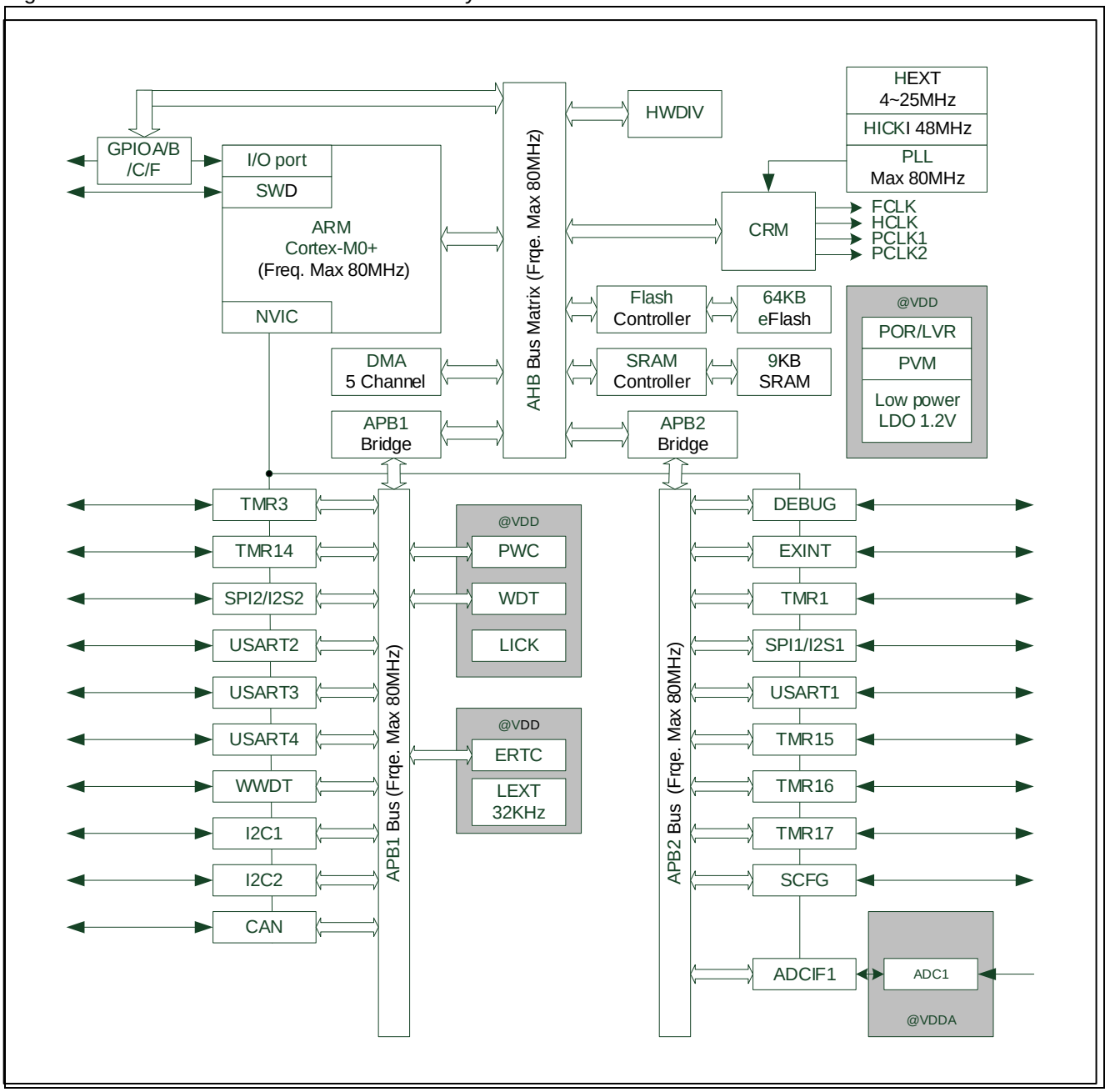
List of figures

List of tables

1 System architecture

AT32L021 series microcontrollers incorporate a 32-bit ARM®Cortex®-M0+ processor core, multiple 16-bit timers, infrared transmitter (IRTMR), DMA controller, ERTC, communication interfaces such as SPI, I2C, USART/UART, CAN bus controller, 12-bit ADC, programmable voltage monitor (PVM) and other peripherals. Cortex®-M0+ processor core supports single-cycle 16-bit/32-bit multiply accumulator (MAC). The system architecture is shown in Figure 1-1.

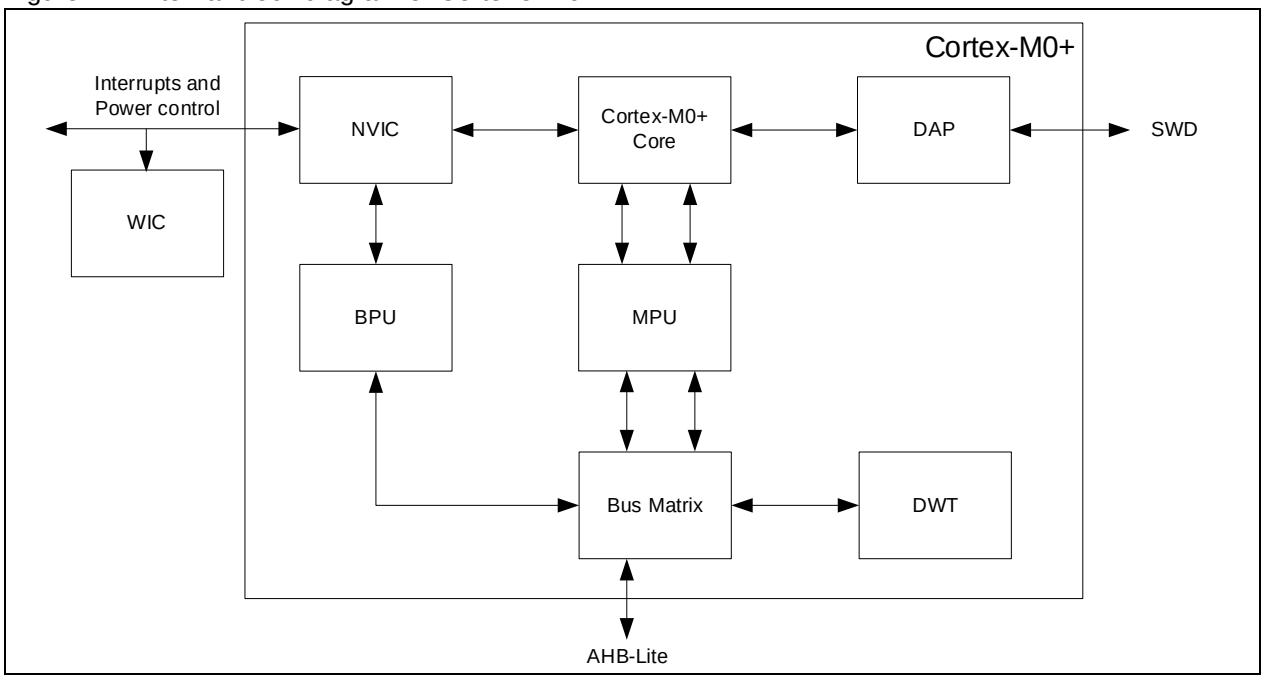
Figure 1-1 AT32L021 series microcontrollers system architecture



1.1 System overview

1.1.1 ARM® Cortex®-M0+ processor

Figure 1-2 Internal block diagram of Cortex®-M0+



1.1.2 Interrupt and exception vectors

The AT32L021 series vectors are listed in the table below.

Table 1-1 AT32L021 series vectors

Pos.	Priority	Priority type	Name	Description	Address
	-	-	-	Reserved	0x0000_0000
	-3	Fixed	Reset	Reset	0x0000_0004
	-2	Fixed	NMI	Non-maskable interrupt CRM clock fail detector (CFD) is linked to NMI vector	0x0000_0008
	-1	Fixed	HardFault	All class of default	0x0000_000C
	-	-	-	Reserved	0x0000_0010
	-	-	-	Reserved	0x0000_0014
	-	-	-	Reserved	0x0000_0018
	-	-	-	Reserved	0x0000_001C ~0x0000_002B
	3	Configurable	SVCall	System service call via SWI instruction	0x0000_002C
	-	-	-	Reserved	0x0000_0030
	-	-	-	Reserved	0x0000_0034
	5	Configurable	PendSV	Pendable request for system service	0x0000_0038
	6	Configurable	SysTick	System tick timer	0x0000_003C
	7	Configurable	WWDT	Window watchdog timer interrupt	0x0000_0040
	8	Configurable	PVM	PVM interrupt linked to EXINT 16	0x0000_0044
	9	Configurable	ERTC	ERTC interrupt linked to EXINT 17, 19, 20	0x0000_0048
	10	Configurable	FLASH	Flash global interrupt	0x0000_004C
	11	Configurable	CRM	Clock and Reset manage (CRM) interrupt	0x0000_0050
	12	Configurable	EXINT1_0	EXINT line 1_0 interrupt	0x0000_0054
	13	Configurable	EXINT3_2	EXINT line 3_2 interrupt	0x0000_0058

	14	Configurable	EXINT15_4	EXINT line 15_4 interrupt	0x0000_005C
	15	Configurable	HWDIV	HWDIV interrupt	0x0000_0060
	16	Configurable	DMA channel 1	DMA channel 1 global interrupt	0x0000_0064
10	17	Configurable	DMA channel 3_2	DMA channel 3_2 global interrupt	0x0000_0068
11	18	Configurable	DMA channel 5_4	DMA channel 5_4 global interrupt	0x0000_006C
12	19	Configurable	ADC	ADC global interrupt	0x0000_0070
13	20	Configurable	TMR1_BRK TMR1_UP TMR1_TRG TMR1_HALL	TMR1 interrupt	0x0000_0074
14	21	Configurable	TMR1_CH	TMR1 capture compare interrupt	0x0000_0078
15	22	Configurable	-	Reserved	0x0000_007C
16	23	Configurable	TMR3	TMR3 global interrupt	0x0000_0080
17	24	Configurable	TMR6	TMR6 global interrupt	0x0000_0084
18	25	Configurable	-	Reserved	0x0000_0088
19	26	Configurable	TMR14	TMR14 global interrupt	0x0000_008C
20	27	Configurable	TMR15	TMR15 global interrupt	0x0000_0090
21	28	Configurable	TMR16	TMR16 global interrupt	0x0000_0094
22	29	Configurable	TMR17	TMR17 global interrupt	0x0000_0098
23	30	Configurable	I2C1_EVT	I2C1 event interrupt	0x0000_009C
24	31	Configurable	I2C2_EVT	I2C2 event interrupt	0x0000_00A0
25	32	Configurable	SPI1	SPI1 global interrupt	0x0000_00A4
26	33	Configurable	SPI2	SPI2 global interrupt	0x0000_00A8
27	34	Configurable	USART1	USART1 global interrupt	0x0000_00AC
28	35	Configurable	USART2	USART2 global interrupt	0x0000_00B0
29	36	Configurable	USART3_4	USART3_4 global interrupt	0x0000_00B4
30	37	Configurable	CAN	CAN global interrupt	0x0000_00B8
31	38	Configurable	-	Reserved	0x0000_00BC

1.1.3 System Tick (SysTick)

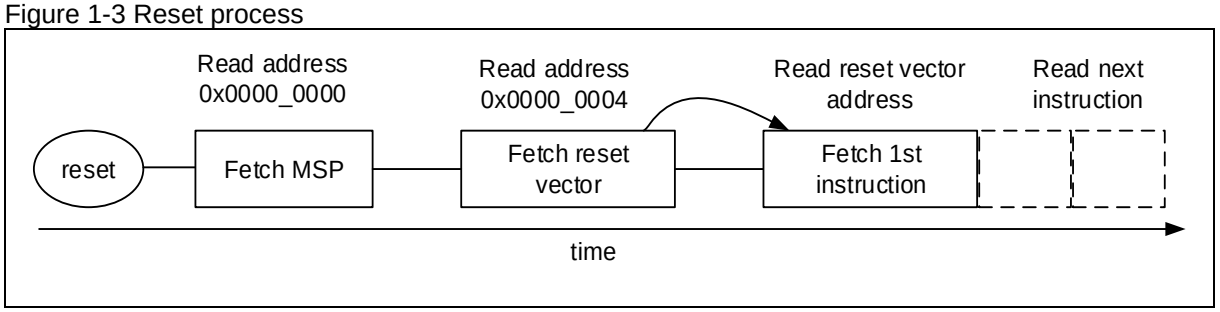
The System Tick is a 24-bit downcounter. It will be reloaded with the initial value automatically when it is decremented to zero. It can generate periodic interrupts, so it is often used as multi-task scheduling counter for embedded operating system, and also to call the periodic tasks for non-embedded system. The System Tick calibration value is fixed to 9000, which gives a reference time base of 1 ms when the System Tick clock is set to 9 MHz.

1.1.4 Reset

The processor reads the first two words from CODE memory after a system reset and before program execution.

- Get the initial value of the main stack pointer (MSP) from address 0x0000_0000.
- Get the initial value of the program counter (PC) from address 0x0000_0004. This value is a reset vector, and LSB must be 1. Then take the instructions from the address corresponding to this value.

Figure 1-3 Reset process

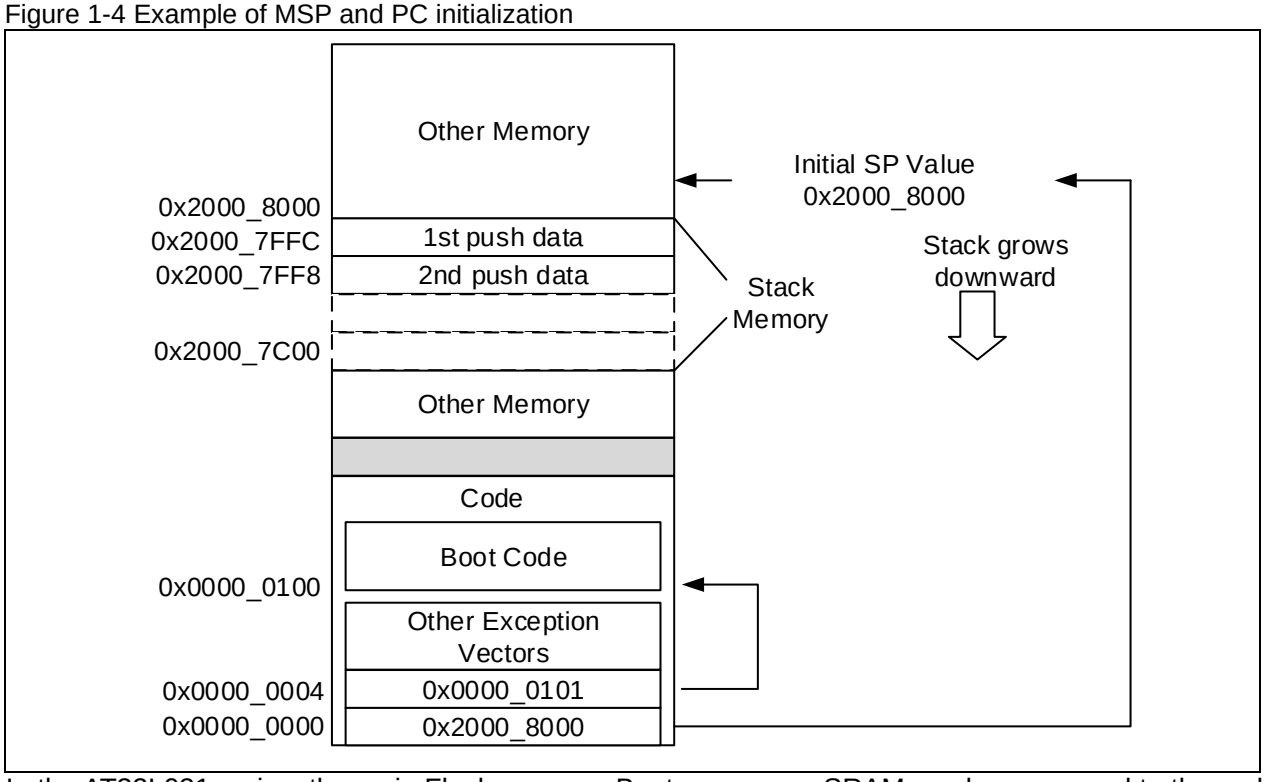


Cortex®-M0+ uses a full stack that increases downward, so the initial value of the main stack pointer (MSP) must be the end address of the stack memory plus 1. For example, if the stack area is set between 0x2000_7C00~0x2000_7FFF, then the initial value of MSP must be defined as 0x2000_8000.

The vector table follows the initial value of MSP. Cortex®-M0+ operates in Thumb state, and thus each value in the vector table must set the LSB to 1. In Figure1-4, 0x0000_0101 is used to represent 0x0000_0100. After the instruction at 0x0000_0100 is executed, the program starts running formally.

Before that, it is a must for initializing MSP, because the first instruction may be interrupted by NMI or other faults before being executed. After the completion of MSP initialization, it is ready to prepare stack room for its service routines.

Figure 1-4 Example of MSP and PC initialization



In the AT32L021 series, the main Flash memory, Boot memory or SRAM can be remapped to the code area between 0x0000_0000~0x07FF_FFFF. nBOOT1 corresponds to the value of the bit nBOOT1 in the SSB of the User System Dara (USD). nBOOT1 and BOOT0 are used to set the specific memory from which CODE starts.

{nBOOT1, BOOT0}=00/10, CODE starts from the main Flash memory

{nBOOT1, BOOT0}=11, CODE starts from Boot memory

{nBOOT1, BOOT0}=01, CODE starts from SRAM

After a system reset or when exiting Standby mode, the pin values of both nBOOT1 and BOOT0 will be relatched.

When the CODE starts from SRAM, the status of BOOT is latched, and it is impossible to load a new boot mode through a system reset. At this point, the power-on reset must be performed to reload a new boot mode.

Boot memory contains an embedded Bootloader that provides not only Flash programming function through USART1 or USART2, but also provides extra firmware including communication protocol stacks that can be called for use by software developers through API.

1.2 List of abbreviations for registers

Table 1-2 List of abbreviations for registers

Register type	Description
rw	Software can read and write to this bit.
ro	Software can only read this bit.
wo	Software can only write to this bit. Reading it returns its reset value.
rrc	Software can read this bit. Reading this bit automatically clears it.
rw0c	Software can read this bit and clear it by writing 0. Writing 1 has no effect on this bit.
rw1c	Software can read this bit and clear it by writing 1. Writing 0 has no effect on this bit.

rw1s	Software can read this bit and set it by writing 1. Writing 0 has no effect on this bit.
tog	Software can read this bit and toggle it by writing 1. Writing 0 has no effect on this bit.
rwt	Software can read this bit. Writing any value will trigger an event.
resd	Reserved.

1.3 Device characteristics information

Table 1-3 List of register addresses and reset values

	Register abbr.	Base address	Reset value
	F_SIZE	0x1FFF F7E0	0xXXXX
	UID[31:0]	0x1FFF F7E8	0xXXXX XXXX
	UID[63:32]	0x1FFF F7EC	0xXXXX XXXX
	UID[95:64]	0x1FFF F7F0	0xXXXX XXXX

1.3.1 Flash memory size register

The register contains the information about Flash memory size.

Bit	Abbr.	Reset value	Type	Description
Bit15:0	F_SIZE	0xXXXX	ro	Flash size, in terms of Kbyte For example: 0x0080 = 128 Kbytes

1.3.2 Device electronic signature

The device electronic signature contains the memory size and the unique device ID (96 bits). It is stored in the information block of the Flash memory. The 96-bit ID is unique for any device, and cannot be altered by users. It can be used for the following:

- Serial number
- Part of security keys

Bit	Abbr.	Reset value	Type	Description
Bit 31:0	UID[31:0]	0xXXXX XXXX	ro	UID for bit 31to bit 0

Bit	Abbr.	Reset value	Type	Description
Bit 31:0	UID[63:32]	0xXXXX XXXX	ro	UID for bit 63 to bit 32

Bit	Abbr.	Reset value	Type	Description
Bit 31:0	UID[95:64]	0xXXXX XXXX	ro	UID for bit 95 to bit 64

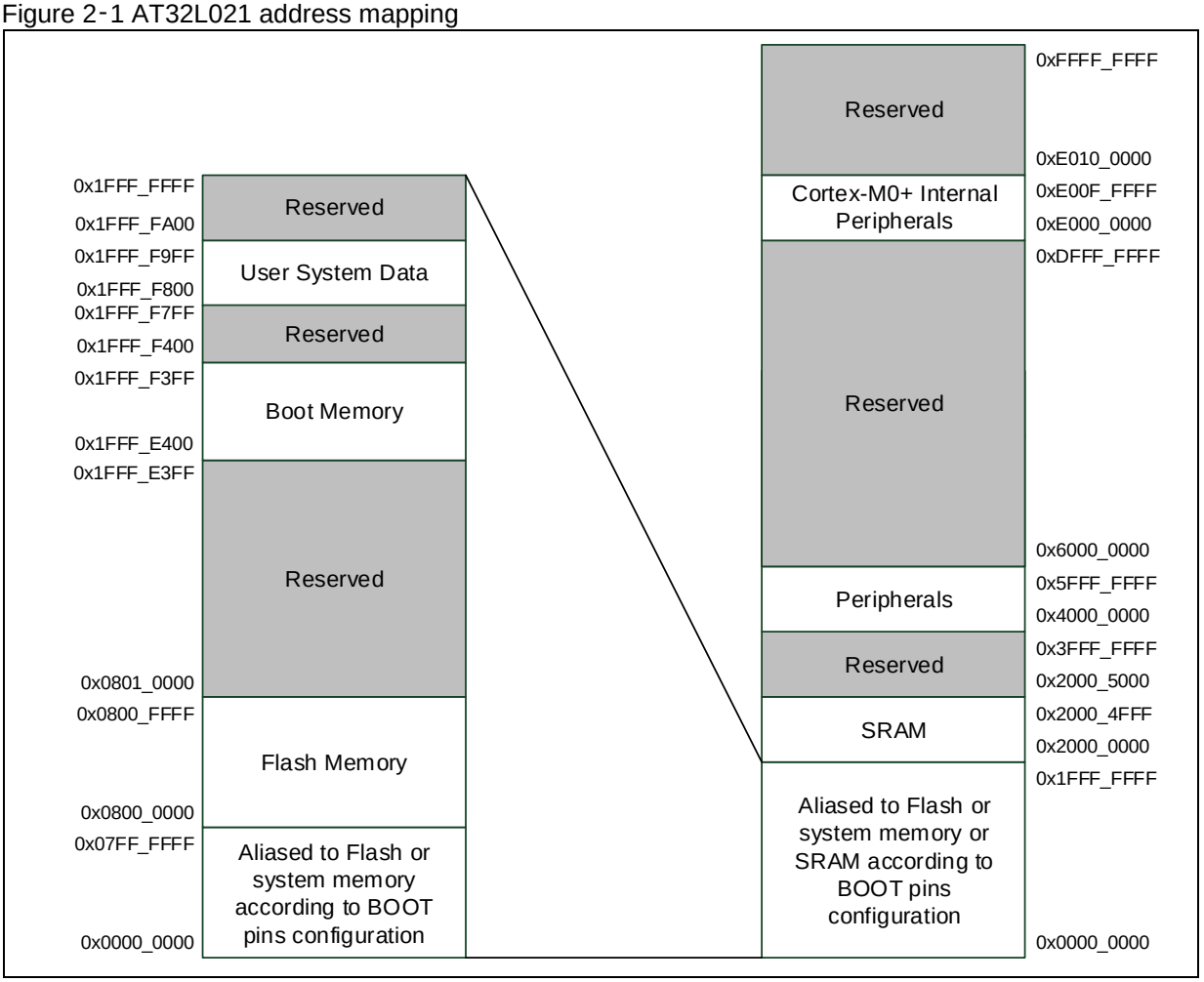
Note: UID[95:88] is Series ID, which is 0x0F for AT32L021.

2 Memory resources

2.1 Internal memory address map

Internal memory contains program memory (Flash), data memory (SRAM), peripheral registers and core registers. Their respective address mapping is shown in Figure 2-1.

Figure 2-1 AT32L021 address mapping



2.2 Flash memory

AT32L021 series provide up to 64 KB on-chip Flash memory, supporting a single-cycle 32-bit read operation.

Flash memory is operated by Flash memory controller. Refer to Chapter 5 for more details about Flash memory controller and register configuration.

Flash memory organization (64 KB)

The main memory contains bank 1 (64 Kbytes), including 64 pages, 1 Kbyte per page.

Table 2-1 Flash memory organization (64 KB)

Bank		Name	Address range
Main memory	Bank 1 (64 KB)	Page 0	0x0800 0000 – 0x0800 03FF
		Page 1	0x0800 0400 – 0x0800 07FF
		Page 2	0x0800 0800 – 0x0800 0BFF
		…	…
		Page 63	0x0800 FC00 – 0x0800 FFFF
Information block		4 KB bootloader	0x1FFF E400 – 0x1FFF F3FF
		512 B user system data	0x1FFF F800 – 0x1FFF F9FF

Flash memory organization (32 KB)

The main memory contains bank 1 (32 Kbytes), including 32 pages, 1 Kbyte per page.

Table 2-2 Flash memory organization (32 KB)

Bank		Name	Address range
Main memory	Bank 1 (32 KB)	Page 0	0x0800 0000 – 0x0800 03FF
		Page 1	0x0800 0400 – 0x0800 07FF
		Page 2	0x0800 0800 – 0x0800 0BFF
		…	…
		Page 31	0x0800 7C00 – 0x0800 7FFF
Information block		4 KB bootloader	0x1FFF E400 – 0x1FFF F3FF
		512 B user system data	0x1FFF F800 – 0x1FFF F9FF

Flash memory organization (16 KB)

The main memory contains bank 1 (16 Kbytes), including 16 pages, 1 Kbyte per page.

Table 2-3 Flash memory organization (16 KB)

Bank		Name	Address range
Main memory	Bank 1 (16 KB)	Page 0	0x0800 0000 – 0x0800 03FF
		Page 1	0x0800 0400 – 0x0800 07FF
		Page 2	0x0800 0800 – 0x0800 0BFF
		…	…
		Page 15	0x0800 3C00 – 0x0800 3FFF
Information block		4 KB bootloader	0x1FFF E400 – 0x1FFF F3FF
		512 B user system data	0x1FFF F800 – 0x1FFF F9FF

2.3 SRAM memory

AT32L021 series contains a 9-KB on-chip SRAM that starts at the address 0x2000_0000. It can be accessed by bytes, half-word (16 bits) or words (32 bits).

Users can configure the nRAM_PRT_CHK in the Flash user system data area, and can enable or disable odd parity check for SRAM. Once odd parity check is enabled for SRAM, only 8 Kbytes of SRAM size will be available, and another 1 Kbyte will be used to store odd parity check results.

When writing data into SRAM, the hardware automatically computes odd parity in 1-byte unit and stores the generated 1-bit odd parity bit in the SRAM. When reading, the hardware automatically checks the odd parity result. If the odd parity result is fault, NMI will be set, and the error status will be reflected to bit 8 of SCFG_CFGR2. Enabling the SRAM_OPERR_LK bit can connect the odd parity error status to the brake output of TMR1/15/16/17.

Note: Before enabling odd parity check for SRAM, the entire SRAM needs to be initialized to prevent false odd parity error.

2.4 Peripheral address map

Table 2-4 Peripheral boundary address

Bus	Boundary address	Peripherals
AHB	0xA000 1000 - 0xFFFF FFFF	Reserved
	0x6000 0000 - 0xA000 0FFF	Reserved
	0x5004 0000 - 0x5FFF FFFF	Reserved
	0x5000 0000 – 0x5003 FFFF	Reserved
	0x4800 1800 – 0x4FFF FFFF	Reserved
	0x4800 1400 - 0x4800 17FF	GPIOF
	0x4800 1000 - 0x4800 13FF	Reserved
	0x4800 0C00 - 0x4800 0FFF	Reserved
	0x4800 0800 - 0x4800 0BFF	GPIOC
	0x4800 0400 - 0x4800 07FF	GPIOB
	0x4800 0000 - 0x4800 03FF	GPIOA
	0x4003 0400 - 0x47FF FFFF	Reserved

Bus	Boundary address	Peripherals
	0x4003 0000 - 0x4003 03FF	HWDIV
	0x4002 3000 - 0x4002 33FF	CRC
	0x4002 2000 - 0x4002 23FF	Flash memory interface (FLASH)
	0x4002 1400 - 0x4002 1FFF	Reserved
	0x4002 1000 - 0x4002 13FF	Clock and reset manage (CRM)
	0x4002 0800 - 0x4002 0FFF	Reserved
	0x4002 0400 - 0x4002 07FF	Reserved
	0x4002 0000 - 0x4002 03FF	DMA
	0x4001 8400 - 0x4001 7FFF	Reserved
	0x4001 8000 - 0x4001 83FF	Reserved
APB2	0x4001 7C00 - 0x4001 7FFF	Reserved
	0x4001 7800 - 0x4001 7BFF	Reserved
	0x4001 7400 - 0x4001 77FF	Reserved
	0x4001 7000 - 0x4001 73FF	Reserved
	0x4001 6C00 - 0x4001 6FFF	Reserved
	0x4001 6800 - 0x4001 6BFF	Reserved
	0x4001 6400 - 0x4001 67FF	Reserved
	0x4001 6000 - 0x4001 63FF	Reserved
	0x4001 5C00 - 0x4001 5FFF	Reserved
	0x4001 5800 - 0x4001 5BFF	DEBUG
	0x4001 5400 - 0x4001 57FF	Reserved
	0x4001 5000 - 0x4001 53FF	Reserved
	0x4001 4C00 - 0x4001 4FFF	Reserved
	0x4001 4800 - 0x4001 4BFF	TMR17 timer
	0x4001 4400 - 0x4001 47FF	TMR16 timer
	0x4001 4000 - 0x4001 43FF	TMR15 timer
	0x4001 3C00 - 0x4001 3FFF	Reserved
	0x4001 3800 - 0x4001 3BFF	USART1
	0x4001 3400 - 0x4001 37FF	Reserved
	0x4001 3000 - 0x4001 33FF	SPI1/I2S1
	0x4001 2C00 - 0x4001 2FFF	TMR1 timer
	0x4001 2800 - 0x4001 2BFF	Reserved
	0x4001 2400 - 0x4001 27FF	ADC
	0x4001 2000 - 0x4001 23FF	Reserved
	0x4001 1C00 - 0x4001 1FFF	Reserved
	0x4001 1800 - 0x4001 1BFF	Reserved
	0x4001 1400 - 0x4001 17FF	Reserved
	0x4001 1000 - 0x4001 13FF	Reserved
	0X4001 0C00 - 0x4001 0FFF	Reserved
	0x4001 0800 - 0x4001 0BFF	Reserved
	0x4001 0400 - 0x4001 07FF	EXINT
	0x4001 0000 - 0x4001 03FF	SCFG
APB1	0x4000 8000 - 0x4000 FFFF	Reserved
	0x4000 7C00 - 0x4000 7FFF	Reserved
	0x4000 7800 - 0x4000 7BFF	Reserved
	0x4000 7400 - 0x4000 77FF	Reserved
	0x4000 7000 - 0x4000 73FF	Power control (PWC)
	0x4000 6C00 - 0x4000 6FFF	CAN
	0x4000 6800 - 0x4000 6BFF	Reserved
	0x4000 6400 - 0x4000 67FF	CAN
	0x4000 6000 - 0x4000 63FF	CAN (RAM)
	0x4000 5C00 - 0x4000 5FFF	Reserved
	0x4000 5800 - 0x4000 5BFF	I^2C2
	0x4000 5400 - 0x4000 57FF	I^2C1
	0x4000 5000 - 0x4000 53FF	Reserved
	0x4000 4C00 - 0x4000 4FFF	USART4
	0x4000 4800 - 0x4000 4BFF	USART3
	0x4000 4400 - 0x4000 47FF	USART2
	0x4000 4000 - 0x4000 43FF	Reserved
	0x4000 3C00 - 0x4000 3FFF	Reserved
	0x4000 3800 - 0x4000 3BFF	SPI2/I^2S2

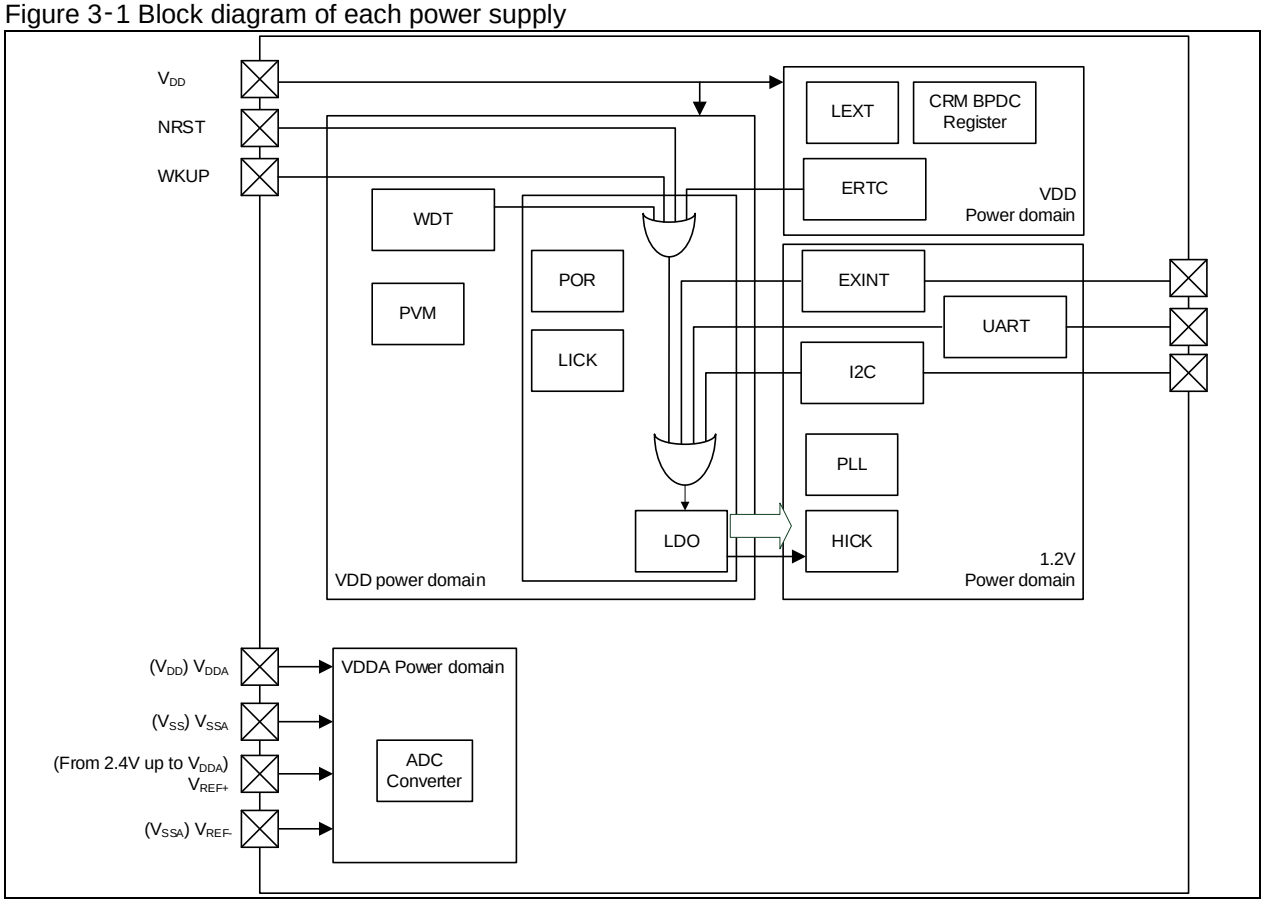
Bus	Boundary address	Peripherals
	0x4000 3400 - 0x4000 37FF	Reserved
	0x4000 3000 - 0x4000 33FF	Watchdog timer (WDT)
	0x4000 2C00 - 0x4000 2FFF	Window watchdog timer (WWDT)
	0x4000 2800 - 0x4000 2BFF	ERTC
	0x4000 2400 - 0x4000 27FF	Reserved
	0x4000 2000 - 0x4000 23FF	TMR14 timer
	0x4000 1C00 - 0x4000 1FFF	Reserved
	0x4000 1800 - 0x4000 1BFF	Reserved
	0x4000 1400 - 0x4000 17FF	Reserved
	0x4000 1000 - 0x4000 13FF	TMR6 timer
	0x4000 0C00 - 0x4000 0FFF	Reserved
	0x4000 0800 - 0x4000 0BFF	Reserved
	0x4000 0400 - 0x4000 07FF	TMR3 timer
	0x4000 0000 - 0x4000 03FF	Reserved

3 Power control (PWC)

3.1 Introduction

Power consumption is one of the most important problems to be considered for the AT32L021 series. For AT32L021 series, its operating voltage supply is 1.71 V ~ 3.6 V, with an operating temperature range of -40~+105 ºC. To reduce power consumption, this series provides three types of power-saving modes, including Sleep, Deepsleep and Standby modes, so as to achieve the best tradeoff among the conflicting demands of CPU operating time, speed and power consumption. The AT32L021 series has two power domains, i.e., VDD/VDDA domain and 1.2 V domain. The VDD/VDDA domain is supplied directly by external power. VDDA and VSSA provide separate analog module power supplies to reduce noise interference. The 1.2 V domain is powered by an embedded LDO in the VDD/VDDA domain.

Figure 3-1 Block diagram of each power supply



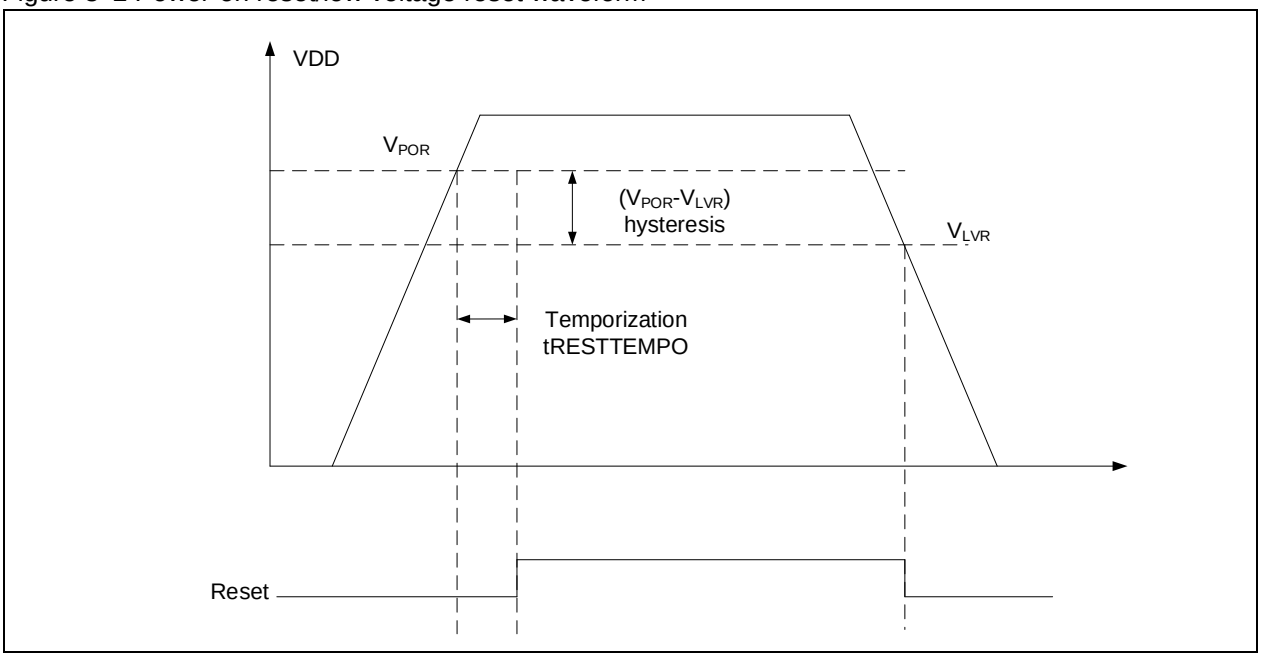
3.2 Main features

Two power domains: VDD/VDDA domain and 1.2 V domain.
Three types of power-saving modes: Sleep mode, Deepsleep mode and Standby mode.
Internal voltage regulator supplies 1.2 V voltage source for the core domain.
Power voltage detector is provided to issue an interrupt or event when the supply voltage is lower or higher than a programmed threshold.

3.3 POR/LVR

A POR analog module embedded in the VDD/VDDA domain is used to generate a power reset. The power reset signal is released at V_{POR} when the VDD is increased from 0 V to the operating voltage, or it is triggered at V_{LVR} when the VDD drops from the operating voltage to 0 V. During the power-on reset period, the reset signal has certain amount of time delay compared to VDD boost process. At the same time, hysteresis occurs in power-on reset (POR) and low voltage reset (LVR).

Figure 3-2 Power-on reset/low voltage reset waveform

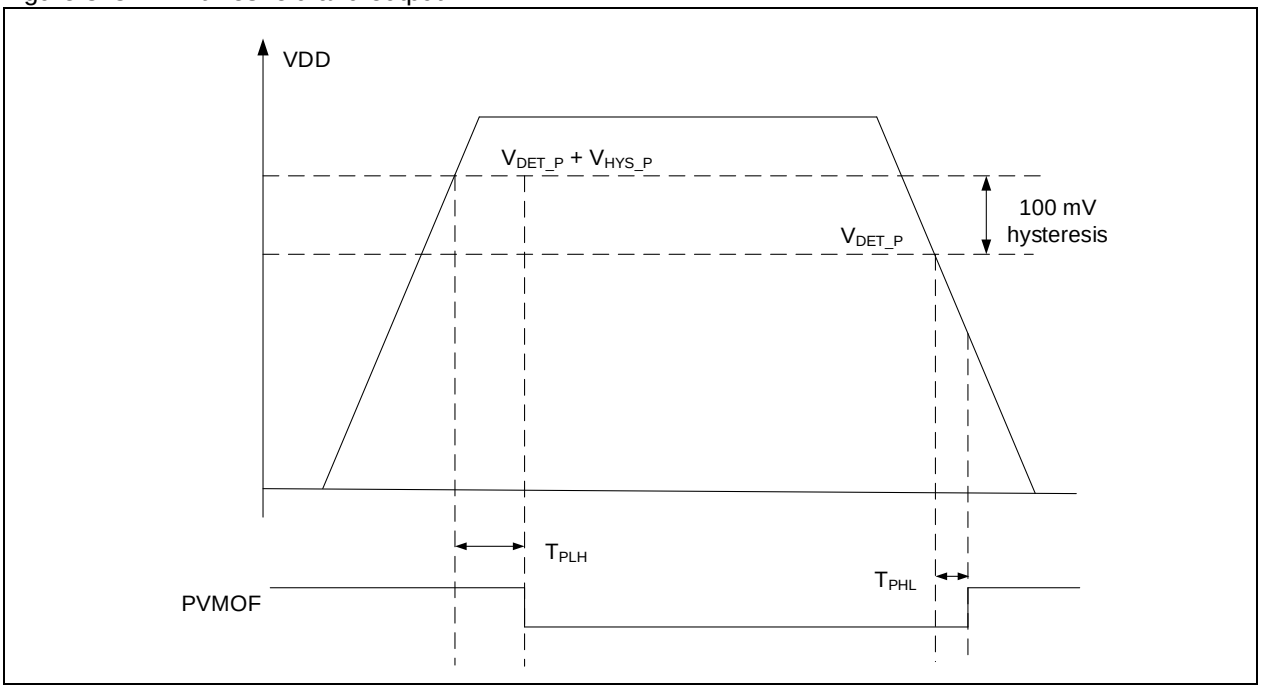


3.4 Power voltage monitor (PVM)

The PVM is used to monitor the power supply variations. It is enabled by setting the PVMEN bit in the power control register (PWC_CTRL), and the threshold value for voltage monitor is selected with the PVMSEL[2:0].

After PVM is enabled, the comparison result between VDD and the programmed threshold is indicated by the PVMOF bit in the PWC_CTRLSTS register, with the hysteresis voltage VHYS_P being 100 mv. The PVM interrupt will be generated (due to PWMOF level change) through the EXTI line 16 when VDD rises above the PVM threshold.

Figure 3-3 PVM threshold and output



3.5 Power domain

1.2 V domain

The 1.2 V core domain includes a CPU core, SRAM, embedded digital peripherals and Phase Locked Loop (PLL). Such power domain is supplied by LDO (voltage regulator).

VDD/VDDA domain

The VDD/VDDA domain includes VDD domain and VDDA domain. The VDD domain contains I/O circuit, power-saving mode wakeup circuit, watchdog timer (WDT), power-on reset/low voltage reset (POR/LVR), LDO, ERTC circuit, LEXT oscillator and all PAD circuits. The VDDA domain contains an ADC (AD converter), and so on.

Typically, to ensure a better accuracy of ADC at a low voltage, the digital circuit is supplied by VDD while the analog circuit is powered by VDDA. The external reference voltage VREF+ and VREF- are connected to VDDA pin and VSSA pin, respectively.

3.6 Power-saving modes

When the CPU does not need to be kept running, there are three types of low-power modes available (Sleep mode, Deepsleep mode and Standby mode) to save power. Users can select the mode that gives the best compromise according to the low-power consumption, short startup time and available wakeup sources. In addition, the power consumption in Run mode can be reduced by slowing down the system clocks or gating the clocks to the APB and AHB peripherals when they are not used.

Sleep Mode

The Sleep mode is enabled by executing WFI or WFE instruction. There are two options to select the Sleep mode entry mechanism through the SLEEPONEXIT bit in the Cortex®-M0+ system control register.

SLEEP-NOW mode:

When SLEEPDEEP=0 and SLEEPONEXIT=0, the MCU enters Sleep mode as soon as WFI or WFE instruction is executed.

SLEEP-ON-EXIT mode:

When SLEEPDEEP=0 and SLEEPONEXIT=1, the MCU enters Sleep mode as soon as the system exits the lowest-priority interrupt service routine by executing the WFI instruction.

In Sleep mode, all clocks and LDO work normally except CPU clock (stopped), and all I/O pins keep the same state as in Run mode. The LDO provides a 1.2 V power (for CPU core, memory and embedded peripherals) as it is in normal power consumption mode.

1) If WFI is executed to enter Sleep mode, any peripheral interrupt can wake up the device from Sleep mode.
2) If WFE is executed to enter Sleep mode, the MCU exits Sleep mode as soon as an event occurs. The wakeup event can be generated by the following:

- Enabling a peripheral interrupt (it is not enabled in the NVIC) and enabling the SEVONPEND bit.

When the MCU resumes, the peripheral interrupt pending bit and NVIC channel pending bit must be cleared.

- Configuring an internal EXINT line as an event mode to generate a wakeup event.

The wakeup time required by a WFE instruction is the shortest, since no time is wasted on interrupt entry/exit.

Deepsleep Mode

Deepsleep mode is entered by setting the SLEEPDEEP bit in the Cortex®-M0+ system control register and clearing the LPSEL bit in the power control register (PWC_CTRL) before executing WFI or WFE instruction.

The power consumption in Deepsleep mode can be further reduced by setting the VRLPEN bit in the PWC_CTRL register.

In Deepsleep mode, all clocks in 1.2 V domain are stopped, and both HICK and HEXT oscillators are disabled. The LDO supplies power to the 1.2 V domain in normal mode or low-power mode. All I/O pins keep the same state as in Run mode. SRAM and register contents are preserved.

1) When the Deepsleep mode is entered by executing a WFI instruction, the interrupt generated on any external interrupt line in Interrupt mode can wake up the system from Deepsleep mode.
2) When the Deepsleep mode is entered by executing a WFE instruction, the interrupt generated on any external interrupt line in Event mode can wake up the system from Deepsleep mode.

When the MCU exits the Deepsleep mode, the HICK RC oscillator is enabled and selected as a system clock after stabilization. When the LDO operates in low-power mode, an additional wakeup delay is incurred for the reason that the LDO must be stabilized before the system is waken from the Deepsleep mode.
Note: Do not disable HICK before entering Deepsleep mode.

Standby Mode
Standby mode can achieve the lowest power consumption for the device. In this mode, the LDO is disabled. The whole 1.2 V domain, PLL, HICK and HEXT oscillators are also powered off except VDD/VDDA domain. SRAM and register contents are lost.
The Standby mode is entered by the following procedures:
- Set the SLEEPDEEP bit in the Cortex®-M0+ system control register
- Set the LPSEL bit in the power control register (PWC_CTRL)
- Clear the SWEF bit in the power control/status register (PWC_CTRLSTS)
- Execute a WFI/WFE instruction

In Standby mode, all I/O pins remain in a high-impedance state except the reset pins, TAMPER pins that are set as anti-tamper or calibration output, and the wakeup pins enabled.
The MCU exits the Standby mode when an external reset (NRST pin), a WDT reset, a rising edge on the WKUPx pin or the rising edge of an ERTC alarm/tamper/timestamp/periodic wakeup event occurs.

Debug mode
By default, the debug connection is lost if the MCU enters Deepsleep mode or Standby mode while debugging. The reason is that the Cortex®-M0+ is no longer clocked. However, the software can be debugged even in the low-power mode by setting some configuration bits in the DEBUG control register (DEBUG_CTRL).

3.7 PWC registers

The peripheral registers must be accessed by words (32 bits).

Table 3-1 PWC register map and reset values

Register abbr.	Offset	Reset value
PWC_CTRL	0x00	0x0000 C000
PWC_CTRLSTS	0x04	0x0000 0000

3.7.1 Power control register (PWC_CTRL)

Note: Write operation to this register is allowed only when the HICK is enabled.

Bit	Name	Reset value	Type	Description
Bit 31:16	Reserved	0x000000	resd	Kept at its default value.
Bit 15	PWOPTL	0x1	rw	Power optimization level 0: Low-power consumption; refer to Datasheet for details about the maximum frequency of system clock. 1: Normal (default value)
Bit 14:13	Reserved	0x2	resd	Kept at its default value.
Bit 12	VRLPEN	0x0	rw	Voltage regulator low power mode enable This bit works with the LPSEL bit of the PWC_CTRL register, and it is valid only when the MCU in Deepsleep mode. 0: Voltage regulator low power mode disabled 1: Voltage regulator low power mode enabled
Bit 11:9	Reserved	0x0	resd	Kept at its default value.
Bit 8	BPWEN	0x0	rw	Battery power domain write enable 0: Disabled 1: Enabled Note: After reset, ERTC is write-protected. To

				write, this bit must be set.
Bit 7:5	PVMSEL	0x0	rw	Power voltage monitoring boundary select 000: 2.15 V 001: 2.30 V 010: 2.45 V 011: 2.60 V 100: 2.75 V 101: 2.90 V 110: 3.00 V 111: Unused, not configurable
Bit 4	PVMEN	0x0	rw	Power voltage monitoring enable 0: Disabled 1: Enabled
Bit 3	CLSEF	0x0	wo	Clear SEF flag 0: No effect 1: Clear the SEF flag Note: This bit is cleared by hardware after clearing the SEF flag. Reading this bit at any time will return zero.
Bit 2	CLSWEF	0x0	wo	Clear SWEF flag 0: No effect 1: Clear the SWEF flag Note: Clear the SWEF flag after two system clock cycles. This bit is cleared by hardware after clearing the SWEF flag. Reading this bit at any time will return zero.
Bit 1	LPSEL	0x0	rw	Low power mode select when Cortex®-M0+ sleepdeep 0: Enter DEEPSLEEP mode 1: Enter Standby mode
Bit 0	Reserved	0x0	rw	Kept at its default value.

3.7.2 Power control/status register (PWC_CTRLSTS)

Note: Write operation to this register is allowed only when the HICK is enabled.

Bit	Name	Reset value	Type	Description
Bit 31:15	Reserved	0x0000	resd	Kept at its default value.
Bit 14	SWPEN7	0x0	rw	Standby wake-up pin 7 enable 0: Disabled (this pin is used for general-purpose I/O) 1: Enabled (this pin is forced in input pull-down mode, and no longer used for general-purpose I/O) Note: This bit is cleared by hardware after system reset.
Bit 13	SWPEN6	0x0	rw	Standby wake-up pin 6 enable 0: Disabled (this pin is used for general-purpose I/O) 1: Enabled (this pin is forced in input pull-down mode, and no longer used for general-purpose I/O) Note: This bit is cleared by hardware after system reset.
Bit 12	Reserved	0x0000	resd	Kept at its default value.
Bit 11	SWPEN4	0x0	rw	Standby wake-up pin 4 enable 0: Disabled (this pin is used for general-purpose I/O) 1: Enabled (this pin is forced in input pull-down mode, and no longer used for general-purpose I/O) Note: This bit is cleared by hardware after system reset.
Bit 10	Reserved	0x0	resd	Kept at its default value.
Bit 9	SWPEN2	0x0	rw	Standby wake-up pin 2 enable 0: Disabled (this pin is used for general-purpose

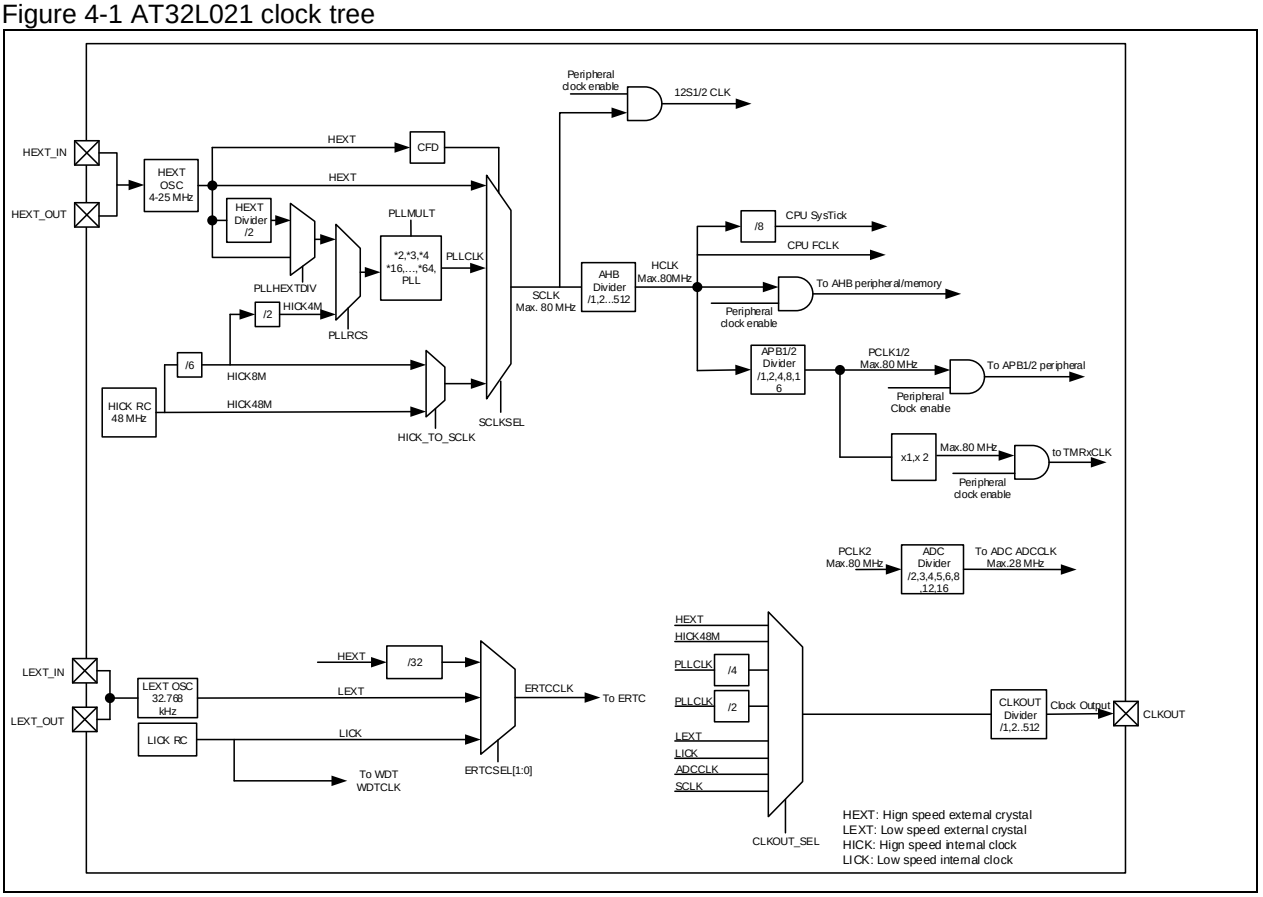
				I/O) 1: Enabled (this pin is forced in input pull-down mode, and no longer used for general-purpose I/O) Note: This bit is cleared by hardware after system reset.
Bit 8	SWPEN1	0x0	rw	Standby wake-up pin 1 enable 0: Disabled (this pin is used for general-purpose I/O) 1: Enabled (this pin is forced in input pull-down mode, and no longer used for general-purpose I/O) Note: This bit is cleared by hardware after system reset.
Bit 7:3	Reserved	0x00	resd	Kept at its default value.
Bit 2	PVMOF	0x0	ro	Power voltage monitoring output flag 0: Power voltage is higher than the threshold 1: Power voltage is lower than the threshold Note: The power voltage monitor is stopped in Standby mode.
Bit 1	SEF	0x0	ro	Standby mode entry flag 0: Device is not in Standby mode 1: Device is in Standby mode Note: This bit is set by hardware (enter Standby mode) and cleared by POR/LVR or by setting the CLSEF bit.
Bit 0	SWEF	0x0	ro	Standby wake-up event flag 0: No wakeup event occurred 1: A wakeup event occurred This bit is set by hardware (on a wakeup event) and cleared by POR/LVR or setting the CLSWEF bit. A wakeup event is generated by one of the following: When the rising edge on the Standby wakeup pin occurs; When the ERTC alarm event occurs; If the Standby wakeup pin is enabled when the Standby wakeup pin level is high.

4 Clock and reset manage (CRM)

4.1 Clock

AT32L021 series provide different clock sources: HEXT oscillator clock, HICK oscillator clock, PLL clock, LEXT oscillator clock and LICK oscillator clock.

Figure 4-1 AT32L021 clock tree



AHB, APB1 and APB2 all support multiple frequency divisions, with a maximum of 80 MHz.

4.1.1 Clock sources

- High speed external oscillator (HEXT)

The HEXT includes two clock sources: crystal/ceramic resonator and HEXT bypass clock

The HEXT crystal/ceramic resonator is connected externally to a 4~25 MHz HEXT crystal that produces a highly accurate clock for the system. The HEXT clock signal is not released until it becomes stable.

An external clock source can be provided by HEXT bypass. Its frequency can be up to 25 MHz. The external clock signal should be connected to the HEXT_IN pin while the HEXT_OUT pin should be left floating.

- High speed internal clock (HICK)

The HICK oscillator is clocked by a high-speed RC in the microcontroller. The internal frequency of the HICK clock is 48 MHz. Although it is less accurate, its startup time is shorter than the HEXT crystal oscillator. The HICK clock frequency of each device is calibrated by ARTERY to ±1% accuracy (T_A=25 °C) in factory. The factory calibration value is loaded in the HICKCAL[5:0] bit of the lock control register. The RC oscillator speed may be affected by voltage or temperature variations. Thus the HICK frequency can be trimmed using the HICKTRIM[2:0] bit in the clock control register.

The HICK clock signal is not released until it becomes stable.

- PLL clock

The HICK or HEXT clock can be used as an input clock source of the PLL. The PLL input clock, after being divided by a pre-divider internally, is sent to the VCO for frequency multiplication, and the VCO output frequency is output after being divided by a post-divider. At the same time, the clock after pre-dividing must remain between 2 MHz and 16 MHz, and the VCO operating frequency must be kept

between 500 MHz and 1000 MHz. The PLL must be configured before being enabled. The reason is that the configuration parameters cannot be changed once PLL is enabled. The PLL clock signal is not released until it becomes stable.
PLL formula:
PLL output clock = PLL input clock x PLL frequency multiplication factor / (PLL pre-divider factor x PLL post-divider factor)
500 MHz <= PLL input clock x PLL frequency multiplication factor / PLL pre-divider factor <= 1000 MHz
2 MHz <= PLL input clock / PLL pre-divider factor <= 16 MHz
For example, when PLL input clock is 16 MHz, the PLL output frequency equals = 16 x 80 / (2 x 8) = 80 MHz.

- Low speed external oscillator (LEXT)

The LEXT oscillator provides two clock sources: LEXT crystal/ceramic resonator and LEXT bypass.
LEXT crystal/ceramic resonator:
The LEXT crystal/ceramic resonator provides a 32.768 KHz low-speed clock source. The LEXT clock signal is not released until it becomes stable.

- LEXT bypass

In this mode, an external clock source with a frequency of 32.768 kHz can be provided. The external clock signal should be connected to the LEXT_IN pin, while the LEXT_OUT pin should be left floating.

- Low speed internal RC oscillator (LICK)

The LICK oscillator is clocked by an internal low-speed RC oscillator. The clock frequency is between 30 kHz and 60 kHz. It acts as a low-power clock source that can be kept running in Deepsleep mode and Standby mode for independent watchdog and auto-wakeup unit.
The LICK clock signal is not released until it becomes stable.

4.1.2 System clock

After a system reset, the HICK oscillator is selected as the system clock. The system clock can make flexible switch among HICK oscillator, HEXT oscillator and PLL clock. However, a switch from one clock source to another occurs only if the target clock source becomes stable. When the HICK oscillator is used directly or indirectly through the PLL as the system clock, it cannot be stopped.

4.1.3 Peripheral clock

Most peripherals use HCLK, PCLK1 or PCLK2. The individual peripherals have their dedicated clocks.
System Tick timer (SysTick) is clocked by HCLK or HCLK/8.
ADC is clocked by APB2 divided by 2, 3, 4, 5, 6, 8, 12, 16.
The timers are clocked by APB1/2. In particularly, if the APB prescaler is 1, the timer clock frequency is equal to that of APB1/2; otherwise, the timer clock frequency doubles that of the APB1/2 frequency.
ERTC clock sources: HEXT/32 oscillator, LEXT oscillator and LICK oscillator. Once the ERTC clock source is selected, it cannot be altered unless resetting the battery powered domain. If the LEXT is selected as the ERTC clock source, the ERTC is not affected when VDD is powered off. If the HEXT or LICK is selected as the ERTC clock source, the ERTC state is not guaranteed when both HEXT and LICK are powered off.
Independent watchdog is clocked by LICK oscillator. If the watchdog is enabled by either hardware option or software access, the LICK oscillator is forced ON. The clock is provided to the watchdog only after the LICK oscillator temporization.

4.1.4 Clock fail detector

The clock fail detector (CFD) is designed to respond to HEXT clock failure when the HEXT is used as a system clock. If a failure is detected on the HEXT clock, a clock failure event is sent to the brake input of TMR1 and an interrupt is generated. This interrupt is directly linked to CPU NMI so that the software can perform rescue operations. The NMI interrupt keeps executing until the CFD interrupt pending bit is cleared. This is why the CFD interrupt has to be cleared in the NMI service routine. The HEXT clock failure will result in a switch of the system clock to the HICK clock, the CFD to be disabled, HEXT clock to be stopped, and even PLL to be disabled if the HEXT clock is selected as the system clock through PLL.

4.1.5 Clock output

The microcontroller allows the internal clock signal to be output to external CLKOUT pins. That is, ADC CLK, SCLK, LICK, LEXT, HICK48, HEXT, PLLCLK/2 and PLLCLK/4 can be used as CLKOUT clocks. When being used as the CLKOUT clock output pin, the corresponding GPIO port registers must be configured accordingly.

4.1.6 Interrupts

The microcontroller specifies a stable flag for each clock source. As a result, when a clock source is enabled, it is possible to determine if the clock is stable by checking the flag pertaining to the clock source. An interrupt request is generated when the interrupt corresponding to the clock source is enabled. If a failure is detected on the HEXT clock, the CFD interrupt is generated. Such interrupt is directly linked to CPU NMI.

4.2 Reset

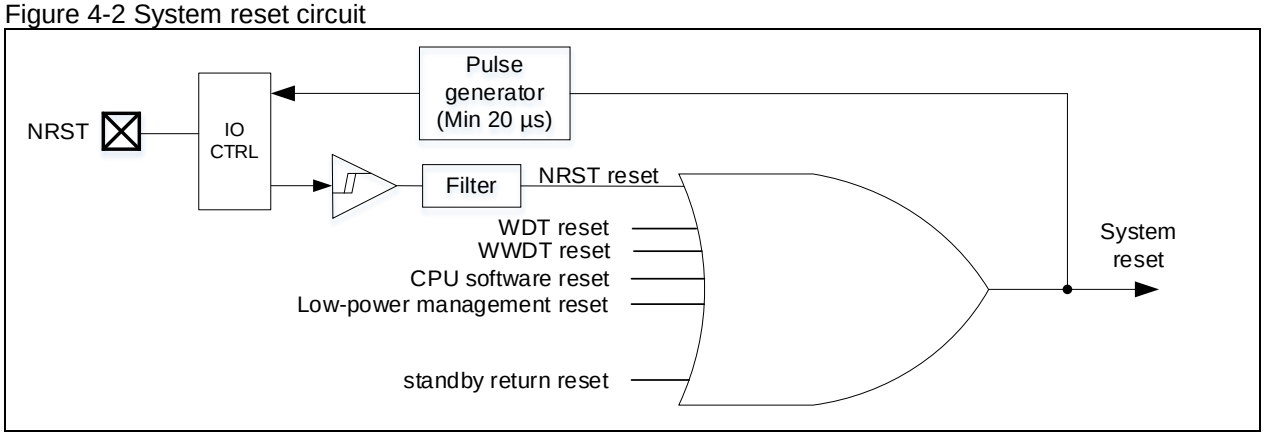
4.2.1 System reset

AT32L021 series provide the following system reset sources:

- NRST reset: on the external NRST pin
- WDT reset: watchdog overflow reset
- WWDT reset: window watchdog overflow reset
- CPU software reset: Cortex®-M0+ software reset
- Low-power management reset: This type of reset is enabled when entering Standby mode (by clearing the nSTDBY_RST bit in the user system data area); this type of reset is also enabled when entering Deepsleep mode (by clearing the nDEPSLP_RST bit in the user system data area).
- When exiting Standby mode

NRST reset, WDT reset, WWDT reset, software reset and low-power management reset sets all registers to their reset values except the clock control/status register (CRM_CTRLSTS) and the battery powered domain registers; the power-on reset, low-voltage reset or reset generated when exiting Standby mode sets all registers to their reset values except the battery powered domain registers.

Figure 4-2 System reset circuit



4.2.2 Battery powered domain reset

Battery powered domain has two specific reset sources:

- Software reset: triggered by setting the BPDRST bit in the battery powered domain control register (CRM_BPDC)
- VDD power on, if VDD has been powered off

Software reset affects only the battery powered domain.

4.3 CRM registers

CRM register map and reset values are shown in the table below.
These peripheral registers have to be accessed by bytes (8 bits), half-words (16 bits) or words (32 bits).

Table 4-1 CRM register map and reset values

Register	Offset	Reset value
CRM_CTRL	0x000	0x0000 XX83
CRM_CFG	0x004	0x0000 0000
CRM_CLKINT	0x008	0x0000 0000
CRM_APB2RST	0x00C	0x0000 0000
CRM_APB1RST	0x010	0x0000 0000
CRM_AHBEN	0x014	0x0000 0014
CRM_APB2EN	0x018	0x0000 0000
CRM_APB1EN	0x01C	0x0000 0000
CRM_BPDC	0x020	0x0000 0000
CRM_CTRLSTS	0x024	0x0C00 0000
CRM_AHBRST	0x028	0x0000 0000
CRM_PLL	0x02C	0x0000 1F10
CRM_MISC1	0x030	0x0000 0000
CRM_HSEDRV	0x034	0x0000 0001
CRM_PICLKS	0x048	0x0000 0000
CRM_MISC2	0x054	0x0000 000D

4.3.1 Clock control register (CRM_CTRL)

Bit	Name	Reset value	Type	Description
Bit 30:26	Reserved	0x00	resd	Kept at its default value.
Bit 25	PLLSTBL	0	ro	PLL clock stable This bit is set by hardware after PLL is ready. 0: PLL clock is not ready; 1: PLL clock is ready.
Bit 24	PLLEN	0	rw	PLL enable This bit is set or cleared by software. It can also be cleared by hardware when entering Standby or Deepsleep mode. When the PLL clock is used as the system clock, this bit cannot be cleared. 0: PLL is OFF; 1: PLL is ON.
Bit 23:20	Reserved	0	resd	Kept at its default value.
Bit 19	CFDEN	0	rw	Clock failure detector enable 0: OFF 1: ON
Bit 18	HEXTBYPS	0	rw	High speed external crystal bypass This bit can be written by software only if the HEXT is disabled. 0: OFF 1: ON
Bit 17	HEXTSTBL	0	ro	High speed external crystal stable This bit is set by hardware after HEXT becomes stable. 0: HEXT is not ready; 1: HEXT is ready.
Bit 16	HEXTEN	0	rw	High speed external crystal enable This bit is set or cleared by software. It can also be cleared by hardware when entering Standby or Deepsleep mode. When the HEXT is used as the

				system clock, this bit cannot be cleared. 0: OFF 1: ON
Bit 13:8	HICKCAL	0xXX	rw	High speed internal clock calibration The default value of this field is the initial factory calibration value. When the HICK output frequency is 48 MHz, it needs adjust 480 kHz (design value) based on this frequency for each HICKCAL value change. Note: This bit can be written only if the HICKCAL_KEY[7:0] is set as 0x5A.
Bit 7:5	Reserved	0x0	resd	Kept at its default value.
Bit 4:2	HICKTRIM	0x04	rw	High speed internal clock trimming This bit works with the HICKCAL[5:0] to determine the HICK oscillator frequency. The default value is 4, which can trim the HICK to be ±0.5%.
Bit 1	HICKSTBL	1	ro	High speed internal clock stable This bit is set by hardware after the HICK is ready. 0: Not ready; 1: Ready.
Bit 0	HICKEN	1	rw	High speed internal clock enable This bit is set or cleared by software. It can also be set by hardware when exiting Standby or Deepsleep mode. When a HEXT clock failure occurs, this bit can also be set by hardware. When the HICK is used as the system clock, this bit cannot be cleared. 0: Disabled; 1: Enabled.

4.3.2 Clock configuration register (CRM_CFG)

Access: 0 to 2 wait states, accessible by bytes, half-words and words.

One or two wait states are inserted only when the access occurs during a clock source switch.

Bit	Name	Reset value	Type	Description
Bit 31	Reserved	0	resd	Kept at its default value.
Bit 26:24	CLKOUT_SEL	0x0	rw	Clock output selection CLKOUT_SEL[3] is the bit 16 of the CRM_MISC1 register. 0000: None 0001: Reserved 0010: LICK 0011: LEXT 0100: SCLK 0101: HICK48 0110: HEXT 0111: PLL/2 1100: PLL/4 1101: Reserved 1110: ADC
Bit 27 Bit 23:22	Reserved	0	resd	Kept at its default value.
Bit 30:29 Bit 21:18	PLLMULT	0x00	rw	PLL multiplication factor (bit 30:29, bit 21:18) 000000: PLL x 2 000001: PLL x 3 000010: PLL x 4 000011: PLL x 5 …… 001100: PLL x 14 001101: PLL x 15 001110: PLL x 16 001111: PLL x 16 010000: PLL x 17 010001: PLL x 18 010010: PLL x 19 010011: PLL x 20 …… 111110: PLL x 63

				111111: PLL x 64
Bit 17	PLLHEXTDIV	0	rw	HEXT division selection for PLL entry clock 0: Not divided 1: HEXT/2
Bit 16	PLLRCS	0	rw	PLL reference clock select 0: HICK-divided clock (4 MHz) 1: HEXT clock
Bit 28 Bit 15:14	ADCDIV	0x0	rw	ADC division The HCLK that is divided by the following factors serves the ADC. 000: Divided by 2 001: Divided by 4 010: Divided by 6 011: Divided by 8 100: Divided by 3 101: Divided by 12 110: Divided by 5 111: Divided by 16
Bit 13:11	APB2DIV	0x0	rw	APB2 division The divided HCLK is used as APB2 clock. 0xx: Not divided 100: Divided by 2 101: Divided by 4 110: Divided by 8 111: Divided by 16 Note: The software must set these bits correctly to ensure that the APB2 clock frequency does not exceed 80 MHz.
Bit 10:8	APB1DIV	0x0	rw	APB1 division The divided HCLK is used as APB1 clock. 0xx: Not divided 100: Divided by 2 101: Divided by 4 110: Divided by 8 111: Divided by 16 Note: The software must set these bits correctly to ensure that the APB1 clock frequency does not exceed 80 MHz.
Bit 7:4	AHBDIV	0x0	rw	AHB division The divided SCLK is used as AHB clock. 0xxx: Not divided 1000: Divided by 2 1001: Divided by 4 1010: Divided by 8 1011: Divided by 16 1100: Divided by 64 1101: Divided by 128 1110: Divided by 256 1111: Divided by 512
Bit 3:2	SCLKSTS	0x0	ro	System clock select status 00: HICK 01: HEXT 10: PLL 11: Reserved. Kept at its default value.
Bit 1:0	SCLKSEL	0x0	rw	System clock select 00: HICK 01: HEXT 10: PLL 11: Reserved. Kept at its default value.

4.3.3 Clock interrupt register (CRM_CLKINT)

Access: 0 wait state, accessible by words, half-words and bytes.

Bit	Name	Reset value	Type	Description
Bit 31:24	Reserved	0x00	resd	Kept at its default value.
Bit 23	CFDFC	0	wo	Clock failure detection flag clear Writing “1” by software to clear CFDF. 0: No effect 1: Clear
Bit 22:21	Reserved	0x0	resd	Kept at its default value.
Bit 20	PLLSTBLFC	0	wo	PLL stable flag clear Writing “1” by software to clear PLLSTBLF. 0: No effect 1: Clear
Bit 19	HEXTSTBLFC	0	wo	HEXT stable flag clear Writing “1” by software to clear HEXTSTBLF. 0: No effect 1: Clear
Bit 18	HICKSTBLFC	0	wo	HICK stable flag clear Writing “1” by software to clear HICKSTBLF. 0: No effect 1: Clear
Bit 17	LEXTSTBLFC	0	wo	LEXT stable flag clear Writing “1” by software to clear LEXTSTBLF. 0: No effect 1: Clear
Bit 16	LICKSTBLFC	0	wo	LICK stable flag clear Writing “1” by software to clear LICKSTBLF. 0: No effect 1: Clear
Bit 15:13	Reserved	0x0	resd	Kept at its default value.
Bit 12	PLLSTBLIEN	0	rw	PLL stable interrupt enable 0: Disabled 1: Enabled
Bit 11	HEXTSTBLIEN	0	rw	HEXT stable interrupt enable 0: Disabled 1: Enabled
Bit 10	HICKSTBLIEN	0	rw	HICK stable interrupt enable 0: Disabled 1: Enabled
Bit 9	LEXTSTBLIEN	0	rw	LEXT stable interrupt enable 0: Disabled 1: Enabled
Bit 8	LICKSTBLIEN	0	rw	LICK stable interrupt enable 0: Disabled 1: Enabled
Bit 7	CFDF	0	ro	Clock failure detection flag This bit is set by hardware when the HEXT clock failure occurs. 0: No clock failure 1: Clock failure
Bit 6:5	Reserved	0x0	resd	Kept at its default value.
Bit 4	PLLSTBLF	0	ro	PLL stable flag Set by hardware. 0: PLL is not ready. 1: PLL is ready.
Bit 3	HEXTSTBLF	0	ro	HEXT stable flag Set by hardware. 0: HEXT is not ready. 1: HEXT is ready.
Bit 2	HICKSTBLF	0	ro	HICK stable flag Set by hardware. 0: HICK is not ready. 1: HICK is ready.
Bit 1	LEXTSTBLF	0	ro	LEXT stable flag Set by hardware. 0: LEXT is not ready.

				1: LEXT is ready.
Bit 0	LICKSTBLF	0	ro	LICK stable flag Set by hardware. 0: LICK is not ready. 1: LICK is ready.

4.3.4 APB2 peripheral reset register (CRM_APB2RST)

Access: 0 wait state, accessible by words, half-words and bytes.

Bit	Name	Reset value	Type	Description
Bit 31:23	Reserved	0x00	resd	Kept at its default value.
Bit 22	DBGRST	0	rw	DBG reset 0: Does not reset DBG 1: Reset DBG
Bit 21:19	Reserved	0x00	resd	Kept at its default value.
Bit 18	TMR17RST	0	rw	TMR17 reset 0: Does not reset TMR17 1: Reset TMR17
Bit 17	TMR16RST	0	rw	TMR16 reset 0: Does not reset TMR16 1: Reset TMR16
Bit 16	TMR15RST	0	rw	TMR15 reset 0: Does not reset TMR15 1: Reset TMR15
Bit 15	Reserved	0x0	resd	Kept at its default value.
Bit 14	USART1RST	0	rw	USART1 reset 0: Does not reset USART1 1: Reset USART1
Bit 13	Reserved	0x0	resd	Kept at its default value.
Bit 12	SPI1RST	0	rw	SPI1 reset 0: Does not reset SPI1 1: Reset SPI1
Bit 11	TMR1RST	0	rw	TMR1 reset 0: Does not reset TMR1 1: Reset TMR1
Bit 10	Reserved	0x0	resd	Kept at its default value.
Bit 9	ADC1RST	0	rw	ADC1 reset 0: Does not reset ADC1 1: Reset ADC1
Bit 8:2	Reserved	0x00	resd	Kept at its default value.
Bit 1	EXINTRST	0	rw	EXINT reset 0: Does not reset EXINT 1: Reset EXINT Note: This bit is always 0 when reading by software.
Bit 0	SCFGRST	0	rw	SCFG reset 0: Does not reset SCFG 1: Reset SCFG

4.3.5 APB1 peripheral reset register (CRM_APB1RST)

Access: 0 wait state, accessible by words, half-words and bytes.

Bit	Name	Reset value	Type	Description
Bit 31:29	Reserved	0x0	resd	Kept at its default value.
Bit 28	PWCRST	0	rw	PWC reset 0: Does not reset PWC 1: Reset PWC
Bit 27:26	Reserved	0x0	resd	Kept at its default value.
Bit 25	CAN1RST	0	rw	CAN1 reset 0: Does not reset CAN1 1: Reset CAN1
Bit 24:23	Reserved	0x0	resd	Kept at its default value.
Bit 22	I2C2RST	0	rw	I2C2 reset 0: Does not reset I2C2 1: Reset I2C2
Bit 21	I2C1RST	0	rw	I2C1 reset 0: Does not reset I2C1

				1: Reset I2C1
Bit 20	Reserved	0x0	resd	Kept at its default value.
Bit 19	USART4RST	0	rw	USART4 reset 0: Does not reset USART4 1: Reset USART4
Bit 18	USART3RST	0	rw	USART3 reset 0: Does not reset USART3 1: Reset USART3
Bit 17	USART2RST	0	rw	USART2 reset 0: Does not reset USART2 1: Reset USART2
Bit 16:15	Reserved	0x0	resd	Kept at its default value.
Bit 14	SPI2RST	0	rw	SPI2 reset 0: Does not reset SPI2 1: Reset SPI2
Bit 13:12	Reserved	0x0	resd	Kept at its default value.
Bit 11	WWDTRST	0	rw	WWDT reset 0: Does not reset WWDT 1: Reset WWDT
Bit 10:9	Reserved	0x0	resd	Kept at its default value.
Bit 8	TMR14RST	0	rw	TMR14 reset 0: Does not reset TMR14 1: Reset TMR14
Bit 7:5	Reserved	0x0	resd	Kept at its default value.
Bit 4	TMR6RST	0	rw	TMR6 reset 0: Does not reset TMR6 1: Reset TMR6
Bit 3:2	Reserved	0x0	resd	Kept at its default value.
Bit 1	TMR3RST	0	rw	TMR3 reset 0: Does not reset TMR3 1: Reset TMR3
Bit 0	Reserved	0x0	resd	Kept at its default value.

4.3.6 AHB peripheral clock enable register (CRM_AHBEN)

Access: 0 wait state, accessible by words, half-words and bytes.

Bit	Name	Reset value	Type	Description
Bit 31:27	Reserved	0x0	resd	Kept at its default value.
Bit 26	HWDIVEN	0	rw	HWDIV clock enable 0: Disabled 1: Enabled
Bit 25:23	Reserved	0x0	resd	Kept at its default value.
Bit 22	GPIOFEN	0	rw	GPIOF clock enable 0: Disabled 1: Enabled
Bit 21:20	Reserved	0x0	resd	Kept at its default value.
Bit 19	GPIOCEN	0	rw	GPIOC clock enable 0: Disabled 1: Enabled
Bit 18	GPIOBEN	0	rw	GPIOB clock enable 0: Disabled 1: Enabled
Bit 17	GPIOAEN	0	rw	GPIOA clock enable 0: Disabled 1: Enabled
Bit 16:7	Reserved	0x0	resd	Kept at its default value.
Bit 6	CRCEN	0	rw	CRC clock enable 0: Disabled 1: Enabled
Bit 5	Reserved	0	resd	Kept at its default value.
Bit 4	FLASHEN	1	rw	Flash clock enable This bit is used to enable Flash clock in Sleep mode. 0: Disabled 1: Enabled *Note: Deepsleep mode must be enabled.*
Bit 3	Reserved	0	resd	Kept at its default value.
Bit 2	SRAMEN	1	rw	SRAM clock enable

				This bit is used to enable SRAM clock in Sleep or Deepsleep mode. 0: Disabled 1: Enabled
Bit 1	Reserved	0	resd	Kept at its default value.
Bit 0	DMA1EN	0	rw	DMA1 clock enable 0: Disabled 1: Enabled

4.3.7 APB2 peripheral clock enable register (CRM_APB2EN)

Access: by words, half-words and bytes.

When accessing to peripherals on APB2 bus, wait states are inserted until the completion of the peripheral access on APB2.

Bit	Name	Reset value	Type	Description
Bit 31:23	Reserved	0x00	resd	Kept at its default value.
Bit 22	DBGEN	0	rw	Debug clock enable 0: Disabled 1: Enabled
Bit 21:19	Reserved	0x00	rw	Kept at its default value.
Bit 18	TMR17EN	0	rw	TMR17 clock enable 0: Disabled 1: Enabled
Bit 17	TMR16EN	0	rw	TMR16 clock enable 0: Disabled 1: Enabled
Bit 16	TMR15EN	0	rw	TMR15 clock enable 0: Disabled 1: Enabled
Bit 15	Reserved	0x0	resd	Kept at its default value.
Bit 14	USART1EN	0	rw	USART1 clock enable 0: Disabled 1: Enabled
Bit 13	Reserved	0x0	resd	Kept at its default value.
Bit 12	SPI1EN	0	rw	SPI1 clock enable 0: Disabled 1: Enabled
Bit 11	TMR1EN	0	rw	TMR1 clock enable 0: Disabled 1: Enabled
Bit 10	Reserved	0x0	resd	Kept at its default value.
Bit 9	ADC1EN	0	rw	ADC1 clock enable 0: Disabled 1: Enabled
Bit 8:1	Reserved	0x00	rw	Kept at its default value.
Bit 0	SCFGEN	0	rw	SCFG clock enable 0: Disabled 1: Enabled

4.3.8 APB1 peripheral clock enable register (CRM_APB1EN)

Access: by words, half-words and bytes.

No-wait state in most cases. When accessing to peripherals on APB1 bus, wait-states are inserted until the completion of peripheral access on the APB1 bus.

Bit	Name	Reset value	Type	Description
Bit 31:29	Reserved	0x0	resd	Kept at its default value.
Bit 28	PWCEN	0	rw	Power control clock enable 0: Disabled 1: Enabled
Bit 27:26	Reserved	0x0	resd	Kept at its default value.
Bit 25	CAN1EN	0	rw	CAN1 clock enable 0: Disabled 1: Enabled
Bit 24:23	Reserved	0x0	resd	Kept at its default value.

Bit 22	I2C2EN	0	rw	I2C2 clock enable 0: Disabled 1: Enabled
Bit 21	I2C1EN	0	rw	I2C1 clock enable 0: Disabled 1: Enabled
Bit 20	Reserved	0x0	resd	Kept at its default value.
Bit 19	USART4EN	0	rw	USART4 clock enable 0: Disabled 1: Enabled
Bit 18	USART3EN	0	rw	USART3 clock enable 0: Disabled 1: Enabled
Bit 17	USART2EN	0	rw	USART2 clock enable 0: Disabled 1: Enabled
Bit 16:15	Reserved	0x0	resd	Kept at its default value.
Bit 14	SPI2EN	0	rw	SPI2 clock enable 0: Disabled 1: Enabled
Bit 13:12	Reserved	0x0	resd	Kept at its default value.
Bit 11	WWDTEN	0	rw	WWDT clock enable 0: Disabled 1: Enabled
Bit 10:9	Reserved	0x0	resd	Kept at its default value.
Bit 8	TMR14EN	0	rw	TMR14 clock enable 0: Disabled 1: Enabled
Bit 7:5	Reserved	0x0	resd	Kept at its default value.
Bit 4	TMR6EN	0	rw	TMR6 clock enable 0: Disabled 1: Enabled
Bit 3:2	Reserved	0x0	resd	Kept at its default value.
Bit 1	TMR3EN	0	rw	TMR3 clock enable 0: Disabled 1: Enabled
Bit 0	Reserved	0x0	resd	Kept at its default value.

4.3.9 Battery powered domain control register (CRM_BPDC)

Access: 0 to 3 wait states, accessible by words, half-words and bytes. Wait states are inserted in the case of consecutive accesses to this register.

Note: LEXTEN, LEXTBYPS, ERTCSEL and ERTCEN bits of the battery powered domain control register (CRM_BPDC) are in the battery powered domain. As a result, these bits are write-protected after reset, and only can be modified by setting the BPWEN bit in the power control register (PWC_CTRL). These bits could be reset only by battery powered domain software reset. Any internal or external reset does not affect these bits.

Bit	Name	Reset value	Type	Description
Bit 31:17	Reserved	0x0000	resd	Kept at its default value.
Bit 16	BPDRST	0	rw	Battery powered domain software reset 0: Does not reset battery powered domain software 1: Reset battery powered domain software
Bit 15	ERTCEN	0	rw	ERTC clock enable This bit is set and cleared by software. 0: Disabled 1: Enabled
Bit 14:10	Reserved	0x00	resd	Kept at its default value.
Bit 9:8	ERTCSEL	0x0	rw	ERTC clock selection Once the ERTC clock source is selected, it cannot be changed until the BPDRST bit is reset. 00: No clock 01: LEXT 10: LICK 11: HEXT/32
Bit 7:5	Reserved	0x0	resd	Kept at its default value.
Bit 4:3	LEXTDRV	0x3	rw	Low speed external crystal driving strength

				00: LOW 01: MEDIUM LOW 10: MEDIUM HIGH 11: HIGH
Bit 2	LEXTBYPS	0	rw	Low-speed external crystal bypass 0: Disabled 1: Enabled
Bit 1	LEXTSTBL	0	ro	External low-speed oscillator stable Set by hardware after the LEXT is ready. 0: LEXT is not ready. 1: LEXT is ready.
Bit 0	LEXTEN	0	rw	External low-speed oscillator enable 0: Disabled 1: Enabled

4.3.10 Control/status register (CRM_CTRLSTS)

Reset flag can only be cleared by power reset or by setting the RSTFC bit, while others are cleared by system set.

Access: 0 to 3 wait states, accessible by words, half-words and bytes. Wait states are inserted in the case of consecutive accesses to this register.

Bit	Name	Reset value	Type	Description
Bit 31	LPRSTF	0	ro	Low-power reset flag Set by hardware. Cleared by writing to the RSTFC bit. 0: No low-power reset occurs 1: Low-power reset occurs
Bit 30	WWDTRSTF	0	ro	WWDT reset flag Set by hardware. Cleared by writing to the RSTFC bit. 0: No WWDT reset occurs 1: WWDT reset occurs
Bit 29	WDTRSTF	0	ro	WDT reset flag Set by hardware. Cleared by writing to the RSTFC bit. 0: No WDT reset occurs 1: WDT reset occurs
Bit 28	SWRSTF	0	ro	Software reset flag Set by hardware. Cleared by writing to the RSTFC bit. 0: No software reset occurs 1: Software reset occurs
Bit 27	PORRSTF	1	ro	POR/LVR reset flag Set by hardware. Cleared by writing to the RSTFC bit. 0: No POR/LVR reset occurs 1: POR/LVR reset occurs
Bit 26	NRSTF	1	ro	NRST reset flag Set by hardware. Cleared by writing to the RSTFC bit. 0: No NRST reset occurs 1: NRST reset occurs
Bit 25	Reserved	0	resd	Kept at its default value.
Bit 24	RSTFC	0	rw	Reset flag clear Cleared by writing 1 through software. 0: No effect 1: Clear the reset flag
Bit 23:2	Reserved	0x000000	resd	Kept at its default value.
Bit 1	LICKSTBL	0	ro	LICK stable 0: LICK is not ready 1: LICK is ready
Bit 0	LICKEN	0	rw	LICK enable 0: Disabled 1: Enabled

4.3.11 AHB peripheral reset register (CRM_AHBRST)

Access: 0 wait state, accessible by words, half-words and bytes.

Bit	Name	Reset value	Type	Description
Bit 31:27	Reserved	0x0000	resd	Kept at its default value.
Bit 26	HWDIVRST	0	rw	HWDIV reset 0: No effect 1: Reset HWDIV
Bit 25:23	Reserved	0x0	resd	Kept at its default value.
Bit 22	GPIOFRST	0	rw	GPIOF reset 0: Does not reset GPIOF 1: Reset GPIOF
Bit 21:20	Reserved	0x0	resd	Kept at its default value.
Bit 19	GPIOCRST	0	rw	GPIOC reset 0: Does not reset GPIOC 1: Reset GPIOC
Bit 18	GPIOBRST	0	rw	GPIOB reset 0: Does not reset GPIOB 1: Reset GPIOB
Bit 17	GPIOARST	0	rw	GPIOA reset 0: Does not reset GPIOA 1: Reset GPIOA
Bit 16:13	Reserved	0x0	resd	Kept at its default value.
Bit 12:0	Reserved	0x0000	resd	Kept at its default value.

4.3.12 PLL configuration register (CRM_PLL)

Access: 0 wait state, accessible by words, half-words and bytes.

Bit	Name	Reset value	Type	Description
Bit 31	PLLCFGEN	0x0	rw	PLL configure enable 0: Common integer multiplication mode, which is done by PLL_FREF and PLLMULT bits. 1: Flexible configuration mode, which is done by PLL_MS/PLL_NS/PLL_FR bits.
Bit 30:27	Reserved	0x0	resd	Kept at its default value.
Bit 26:24	PLL_FREF	0x0	rw	PLL input clock selection This field is valid only if PLLCFGEN = 0. 000: 3.9 ~ 5 MHz 001: 5.2 ~ 6.25 MHz 010: 7.8125 ~ 8.33 MHz 011: 8.33 ~ 12.5 MHz 100: 15.625 ~ 20.83 MHz 101: 20.83 ~ 31.25 MHz 110: Reserved 111: Reserved
Bit 23:17	Reserved	0x00	resd	Kept at its default value.
Bit 16:8	PLL_NS	0x1F	rw	PLL multiplication factor PLL_NS range (31 ~ 500)
Bit 7:4	PLL_MS	0x1	rw	PLL pre-division factor PLL_MS range (1 ~ 15)
Bit 3	Reserved	0x0	resd	Kept at its default value.
Bit 2:0	PLL_FR	0x0	rw	PLL post-division factor PLL_FR range (0~5) 000: PLL post-division=1, divided by 1 001: PLL post-division=2, divided by 2 010: PLL post-division=4, divided by 4 011: PLL post-division=8, divided by 8 100: PLL post-division=16, divided by 16 101: PLL post-division=32, divided by 32 Others: Reserved It should be noted the relationship between the PLL_FR values and post-division factors.

Note: PLL clock formulas:
PLL output clock = PLL input clock x PLL frequency multiplication factor / (PLL pre-division factor x PLL post-division factor)
500 MHz <= PLL input clock x PLL frequency multiplication factor / PLL pre-division factor <= 1000 MHz

2 MHz <= PLL input clock / PLL pre-division factor <= 16 MHz

4.3.13 Additional register 1 (CRM_MISC1)

Access: 0 wait state, accessible by words, half-words and bytes.

Bit	Name	Reset value	Type	Description
Bit 31:28	CLKOUTDIV	0x0	rw	Clock output division 0xxx: Clock output 1000: Clock output divided by 2 1001: Clock output divided by 4 1010: Clock output divided by 8 1011: Clock output divided by 16 1100: Clock output divided by 64 1101: Clock output divided by 128 1110: Clock output divided by 256 1111: Clock output divided by 512
Bit 27:17	Reserved	0x0	resd	Kept at its default value.
Bit 16	CLKOUT_SEL[3]	0	rw	Clock output selection This bit works with the bit [26:24] of the CRM_CFG register.
Bit 15:8	Reserved	0x00	resd	Kept at its default value.
Bit 7:0	HICKCAL_KEY	0x00	rw	HICK calibration key The HICKCAL [5:0] can be written only when this field is set to 0x5A.

4.3.14 HSE driving strength control register (CRM_HSEDRV)

Access: 0 wait state, accessible by words, half-words and bytes.

Bit	Name	Reset value	Type	Description
Bit 31:2	Reserved	0x0	resd	Kept at its default value.
Bit 1:0	HSEDRV	0x01	rw	HSE driving strength control bits 00: Low 01: Medium low 10: Medium high 11: High

4.3.15 Peripheral independent clock select (CRM_PICLKS)

Access: 0 wait state, accessible by words, half-words and bytes.

Bit	Name	Reset value	Type	Description
Bit 31:14	Reserved	0x0000	resd	Kept at its default value.
Bit 13:12	I2C1CLK_SEL	0	rw	I2C1 clock select 00: PCLK 01: SCLK 10: HICK48 11: reserved
Bit 11:4	Reserved	0x0000	resd	Kept at its default value.
Bit 3:2	USART2CLK_SEL	0x0	resd	USART2 clock select 00: PCLK 01: SCLK 10: HICK48 11: LEXT
Bit 1:0	USART1CLK_SEL	0x0	resd	USART1 clock select 00: PCLK 01: SCLK 10: HICK48 11: LEXT

4.3.16 Additional register 2 (CRM_MISC2)

Access: 0 wait state, accessible by words, half-words and bytes.

Bit	Name	Reset value	Type	Description
Bit 31:10	Reserved	0x0000	resd	Kept at its default value.
Bit 9	HICK_TO_SCLK	0	rw	HICK as system clock frequency select When HICK is used as the clock source of SCLKSEL, the SCLK frequency is: 0: Fixed 8 MHz 1: Fixed 48 MHz
Bit 8:0	Reserved	0xd	resd	Fixed 0xd. Do not change.

5 Flash memory controller (FLASH)

5.1 FLASH introduction

Flash memory is divided into three parts: main Flash memory, information block and Flash memory registers.

- Main Flash memory is up to 64 Kbytes.
- Information block consists of 4 Kbytes bootloader and user system data area. The bootloader uses USART1 or USART2 for ISP programming.

Main Flash memory contains bank 1 (64 Kbytes), including 64 pages, 1 Kbyte per page.

Table 5-1 Flash memory organization (64 KB)

Bank		Name	Address range
Main memory	Bank 1 (64 KB)	Page 0	0x0800 0000 – 0x0800 03FF
		Page 1	0x0800 0400 – 0x0800 07FF
		Page 2	0x0800 0800 – 0x0800 0BFF
		…	…
		Page 63	0x0800 FC00 – 0x0800 FFFF
Information block		4 KB bootloader	0x1FFF E400 – 0x1FFF F3FF
		512 B user system data	0x1FFF F800 – 0x1FFF F9FF

Main Flash memory contains bank 1 (32 Kbytes), including 32 pages, 1 Kbyte per page.

Table 5-2 Flash memory organization (32 KB)

Bank		Name	Address range
Main memory	Bank1 (32 KB)	Page 0	0x0800 0000 – 0x0800 03FF
		Page 1	0x0800 0400 – 0x0800 07FF
		Page 2	0x0800 0800 – 0x0800 0BFF
		…	…
		Page 31	0x0800 7C00 – 0x0800 7FFF
Information block		4 KB bootloader	0x1FFF E400 – 0x1FFF F3FF
		512 B user system data	0x1FFF F800 – 0x1FFF F9FF

Main Flash memory contains bank 1 (16 Kbytes), including 16 pages, 1 Kbyte per page.

Table 5-3 Flash memory organization (16 KB)

Bank		Name	Address range
Main memory	Bank 1 (16 KB)	Page 0	0x0800 0000 – 0x0800 03FF
		Page 1	0x0800 0400 – 0x0800 07FF
		Page 2	0x0800 0800 – 0x0800 0BFF
		…	…
		Page 15	0x0800 3C00 – 0x0800 3FFF
Information block		4 KB bootloader	0x1FFF E400 – 0x1FFF F3FF
		512 B user system data	0x1FFF F800 – 0x1FFF F9FF

User system data area

The system data will be read from the information block of Flash memory whenever a system reset occurs, and it is saved in the user system data register (FLASH_USD) and erase programming protection status register (FLASH_EPPS).

Each system data occupies two bytes, where the low byte corresponds to the content in the system data area, and the high byte represent the inverse code that is used to verify the correctness of the selected bit. When the high byte is not equal to the inverse code of the low byte (except when both high byte and low byte are all 0xFF), the system data loader will issue a system data error flag (USDERR), and the corresponding system data and their inverse codes will be forced to 0xFF.

Note: The update of the content in the user system data area becomes effective only after a system reset.

Table 5-4 User system data area

Address	Bit	Description	
0x1FFF_F800	[7:0]	**FAP[7:0]**: Flash memory access protection (Access protection enable/disable results are stored in the FLASH_USD register bit [1] and bit [26]) 0xA5: Flash access protection disabled 0xCC: High-level Flash access protection enabled Others: Low-level Flash access protection enabled	
	[15:8]	**nFAP[7:0]**: Inverse code of FAP[7:0]	
	[23:16]	SSB[7:0] : System configuration byte (it is stored in the FLASH_USD[9:2] register)	
		Bit 7 (nRAM_PRT_CHK)	0: Odd parity for RAM is enabled. 1: Odd parity for RAM is disabled
		Bit 6 (nSTDBY_WDT)	0: WDT stops counting while entering Standby mode 1: WDT does not stop counting while entering Standby mode
		Bit 5 (nDEPSLP_WDT)	0: WDT stops counting while entering Deepsleep mode 1: WDT does not stop counting while entering Deepsleep mode
		Bit 4 (nBOOT1)	nBOOT1: It defines boot mode together with BOOT0 pin. When BOOT0 = 1, 0: Boot from SRAM 1: Boot from boot memory
		Bit 3	Reserved
		Bit 2 (nSTDBY_RST)	0: Reset occurs when entering Standby mode 1: No reset occurs when entering Standby mode
		Bit 1 (nDEPSLP_RST)	0: Reset occurs when entering Deepsleep mode 1: No reset occurs when entering Deepsleep mode
		Bit 0 (nWDT_ATO_EN)	0: Watchdog is enabled 1: Watchdog is disabled
	[31:24]	**nSSB[7:0]**: Inverse code of SSB[7:0]	
0x1FFF_F804	[7:0]	**Data0[7:0]**: User data 0 (stored in the FLASH_USD[17:10] register)	
	[15:8]	**nData0[7:0]**: Inverse code of Data0[7:0]	
	[23:16]	**Data1[7:0]**: User data 1 (stored in the FLASH_USD[25:18] register)	
	[31:24]	**nData1[7:0]**: Inverse code of Data1[7:0]	
0x1FFF_F808	[7:0]	**EPP0[7:0]**: Flash erase/write protection byte 0 (stored in the FLASH_EPPS[7:0] register) This field is used to protect page0~page31 of the main Flash memory. Each bit takes care of 4 KB pages (1 KB per page). 0: Erase/write protection is enabled 1: Erase/write protection is disabled	
	[15:8]	**nEPP0[7:0]**: Inverse code of EPP0[7:0]	
	[23:16]	**EPP1[7:0]**: Flash erase/write protection byte 1 (stored in the FLASH_EPPS[15:8] register) This field is used to protect page32~page63 of the main Flash memory. Each bit takes care of 4 KB pages (1 KB per page). 0: Erase/write protection is enabled 1: Erase/write protection is disabled	
	[31:24]	**nEPP1[7:0]**: Inverse code of EPP1[7:0]	
0x1FFF_F80C	[7:0]	**EPP2[7:0]**: Flash erase/write protection byte 2 (stored in the FLASH_EPPS[23:16] register) Reserved	
	[15:8]	**nEPP2[7:0]**: Inverse code of EPP2[7:0]	
	[23:16]	EPP3[7:0]: Flash erase/write protection byte 3 (stored in the FLASH_EPPS[31:24] register) Bit [6:0] is reserved. Bit [7] is used to protect the main Flash memory extension area. 0: Erase/write protection is enabled 1: Erase/write protection is disabled	
	[31:24]	**nEPP3[7:0]**: Inverse code of EPP3[7:0]	
0x1FFF_F810	[7:0]	**Data2[7:0]**: User data 2	

	[15:8]	**nData2[7:0]**: Inverse code of Data2[7:0]
	[23:16]	**Data3[7:0]**: User data 3
	[31:24]	**nData3[7:0]**: Inverse code of Data3[7:0]
0x1FFF_F814	[7:0]	**Data4[7:0]**: User data 4
	[15:8]	**nData4[7:0]**: Inverse code of Data4[7:0]
	[23:16]	**Data5[7:0]**: User data 5
	[31:24]	**nData5[7:0]**: Inverse code of Data5[7:0]
. .	. .	. .
0x1FFF_F9FC	[7:0]	**Data248[7:0]**: User data 248
	[15:8]	**nData248[7:0]**: Inverse code of Data248[7:0]
	[23:16]	**Data249[7:0]**: User data 249
	[31:24]	**nData249[7:0]**: Inverse code of Data249[7:0]

5.2 Flash memory operation

5.2.1 Unlock/lock

After reset, Flash memory is protected, by default. FLASH_CTRL cannot be written. Write and erase operation can be performed only when the Flash memory is unlocked.

Unlock procedure:

Flash memory block can be unlocked by writing KEY1 (0x45670123) and KEY2 (0xCDEF89AB) to the FLASH_UNLOCK register.

Note: Writing an incorrect key sequence leads to a bus error and the Flash memory is also locked until the next reset.

Lock procedure:

Flash memory block can be locked by setting the OPLK bit in the FLASH_CTRL register.

5.2.2 Erase operation

Erase operation must be done before programming. Flash memory erase includes page erase and mass erase.

Page erase

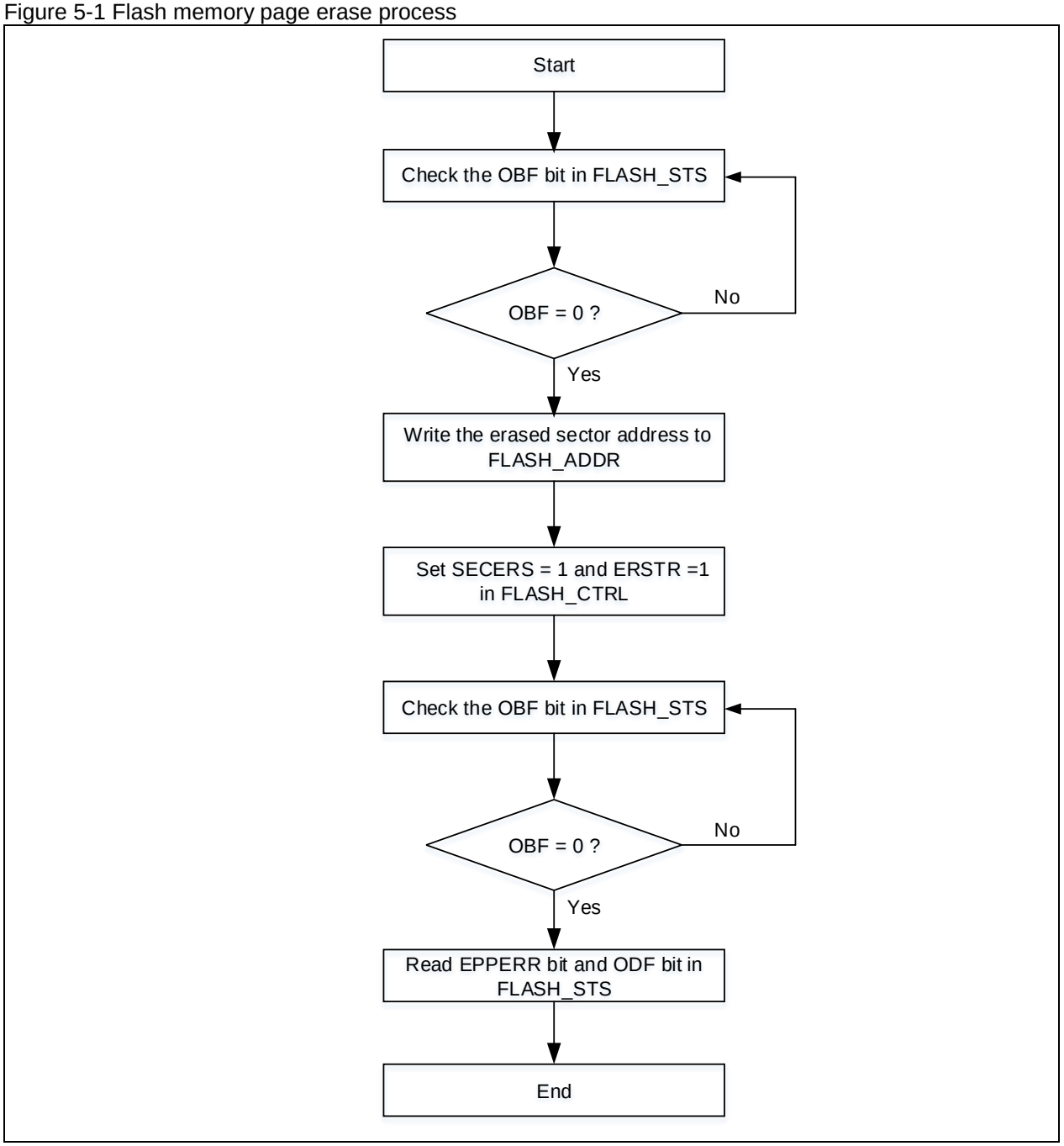
Any page in the Flash memory can be erased with page erase function independently.

The following process is recommended:

- Check the OBF bit in the FLASH_STS register to confirm that there is no other programming operation in progress;
- Write the page to be erased in the FLASH_ADDR register;
- Set SECERS=1 and ERSTR=1 in the FLASH_CTRL register to enable page erase;
- Wait until the OBF bit in the FLASH_STS register becomes “0”. Read the EPPERR and ODF bits in the FLASH_STS register to verify the erase result.

Note: When the boot memory is configured as the Flash memory extension area, performing page-erase operation erases the entire Flash memory extension area.

Figure 5-1 Flash memory page erase process



Mass erase

Mass erase function can be used to erase the entire Flash memory.

The following process is recommended:

- Check the OBF bit in the FLASH_STS register to confirm that there is no other programming operation in progress;
- Set BANKERS=1 and ERSTR=1 in the FLASH_CTRL register to enable the mass erase;
- Wait until the OBF bit in the FLASH_STS register becomes “0”. Read the EPPERR and ODF bits in the FLASH_STS register to verify the erase result.

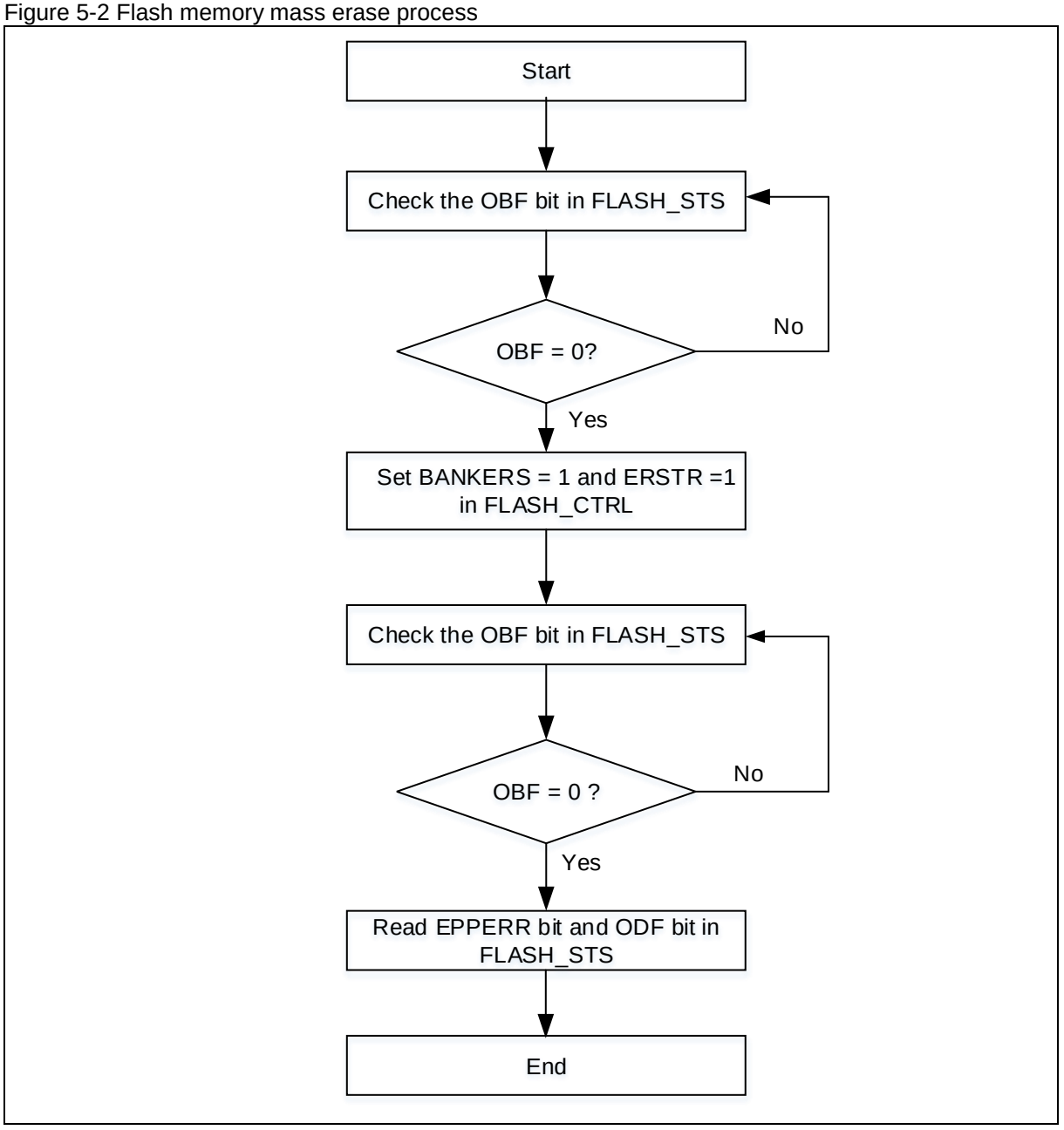
Note:

1) When the boot memory is configured as the Flash memory extension area, performing mass-erase operation erases automatically the entire Flash memory and its extension area.

2) Read access during erase operation halts the CPU and waits until the completion of erase.

3) Internal HICK must be enabled prior to erase operation.

Figure 5-2 Flash memory mass erase process



5.2.3 Programming operation

The Flash memory can be programmed with 32 bits, 16 bits or 8 bits at a time.
The following process is recommended:

- Check the OBF bit in the FLASH_STS register to confirm that there is no other programming operation in process;
- Set FPRGM=1 in the FLASH_CTRL register, so that the Flash memory programming instructions can be received;
- Write the data (word/half-word/byte) to be programmed to the designated address;
- Wait until the OBF bit in the FLASH_STS register becomes “0”. Read the EPPERR, PRGMERR and ODF bits in the FLASH_STS register to verify the programming result.

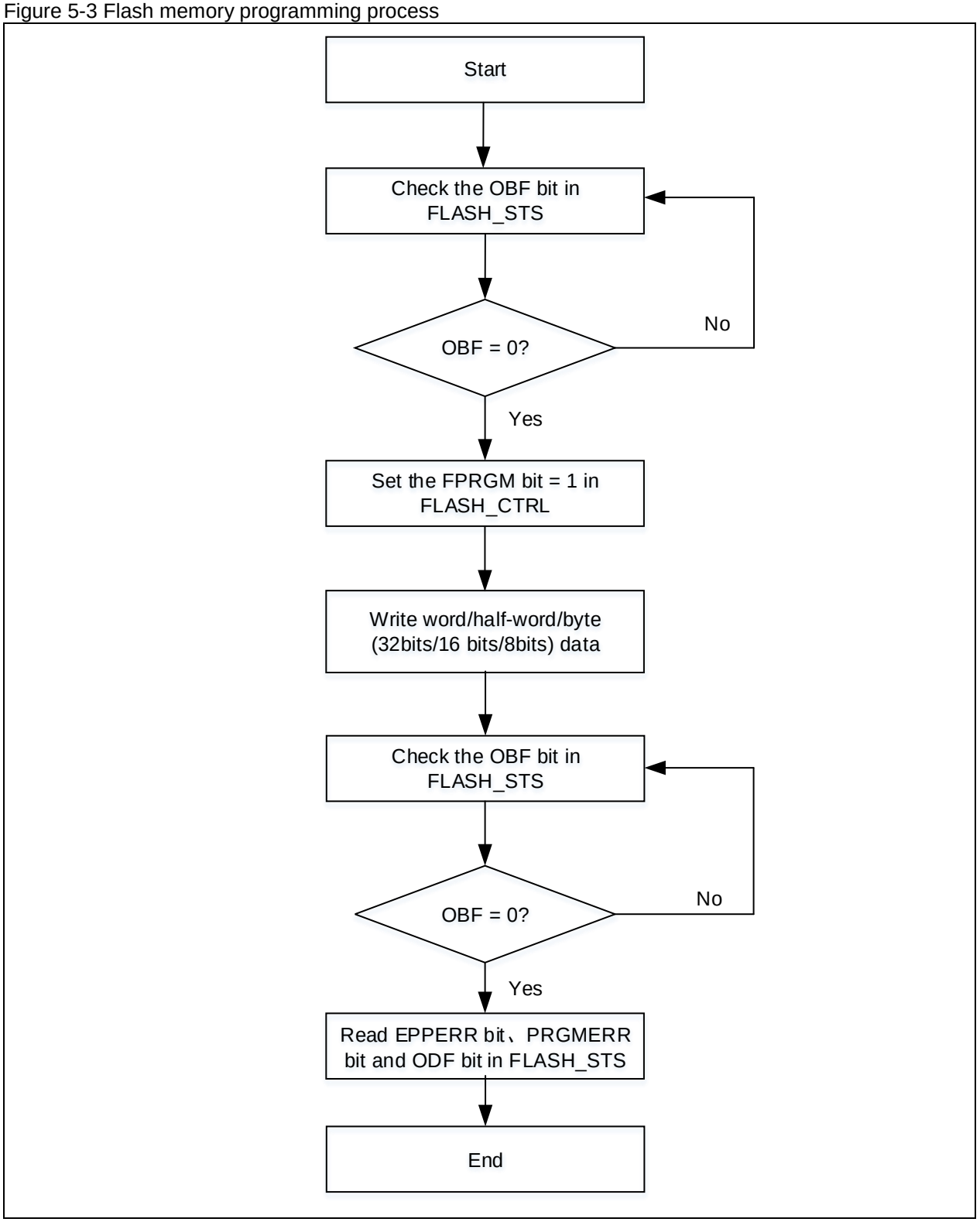
Note:

1) When the address to be written is not erased in advance, the programming operation is not executed unless the data to be written is all 0. In this case, a programming error is reported by the PRGMERR bit in the FLASH_STS register.

2) Read access to the Flash memory during programming halts the CPU and waits until the completion of programming.

3) Internal HICK must be enabled prior to programming.

Figure 5-3 Flash memory programming process



5.2.4 Read operation

Flash memory can be accessed through AHB bus of the CPU.

5.3 Main Flash memory extension area

The boot memory can also be programmed as the extension area of the main Flash memory to store user-application code. When used as the main Flash memory extension area, it behaves like the main Flash memory, including read, unlock, erase and programming operations.

5.4 User system data area operation

5.4.1 Unlock/lock

After reset, the user system data area is protected, by default. Write and erase operations can be performed only after the Flash memory is unlocked before the user system data area.

Unlock procedure:

Flash memory block can be locked by writing KEY1 (0x45670123) and KEY2 (0xCDEF89AB) to the FLASH_UNLOCK register;

When KEY1 (0x45670123) and KEY2 (0xCDEF89AB) are written to the FLASH_USD_UNLOCK register, the USDULKS bit in the FLASH_CTRL register will be automatically set by hardware, indicating that it supports write/erase operation to the user system data area.

Note: Writing an incorrect key sequence leads to bus error and the Flash memory is also locked until the next reset.

Lock procedure:

User system data area is locked by clearing the USDULKS bit in the FLASH_CTRL register by software.

5.4.2 Erase operation

Erase operation must be done before programming. User system data area can perform erase operation independently.

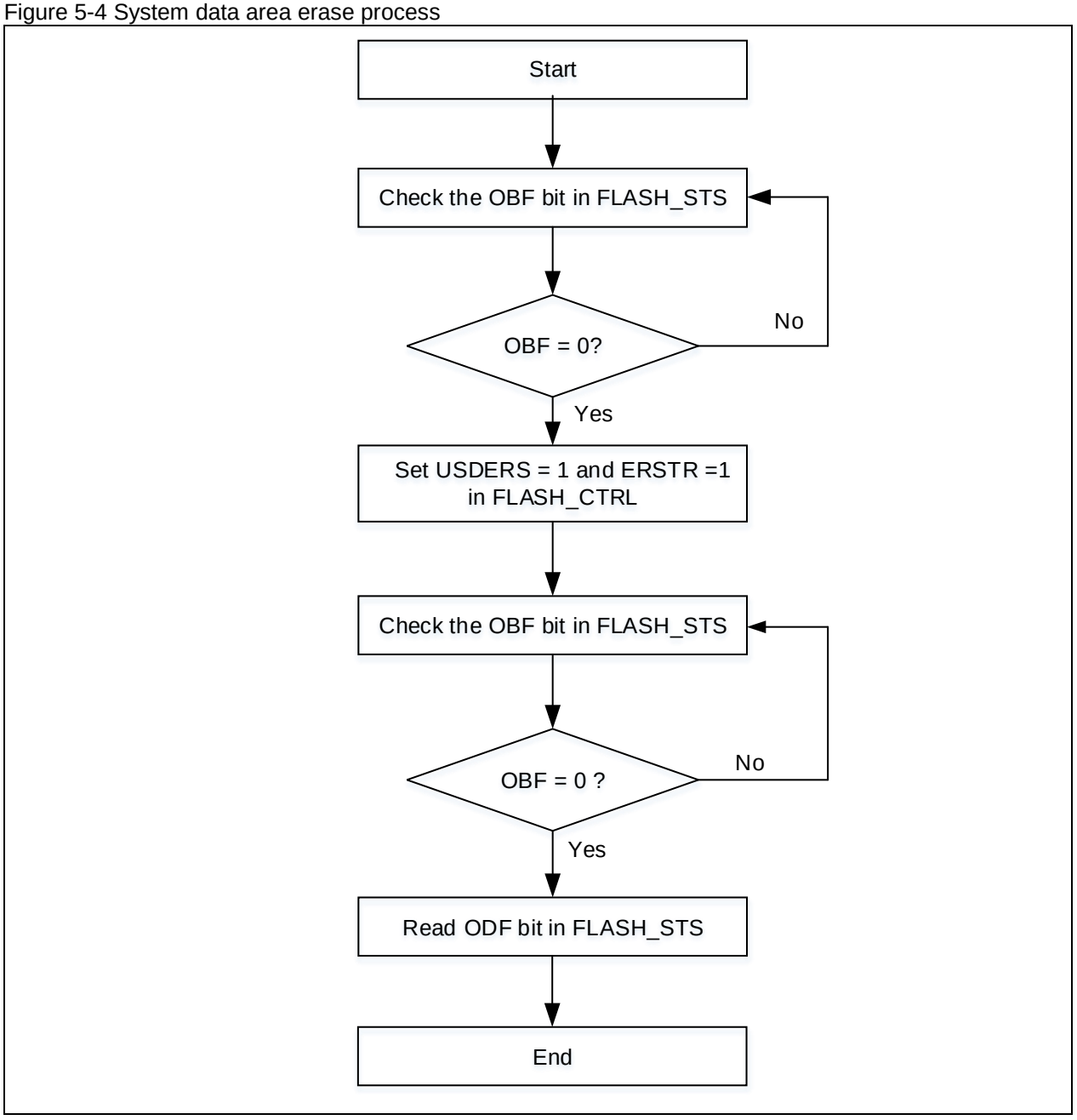
The following process is recommended:

- Check the OBF bit in the FLASH_STS register to confirm that there is no other programming operation in progress;
- Set USDERS=1 and ERSTR=1 in the FLASH_CTRL register to enable erase operation;
- Wait until the OBF bit in the FLASH_STS register becomes “0”. Read the ODF bit in the FLASH_STS register to verify the erase result.

Note:

Read access to the Flash memory during programming halts the CPU and waits until the completion of erase. The internal HICK must be enabled prior to erase operation.

Figure 5-4 System data area erase process



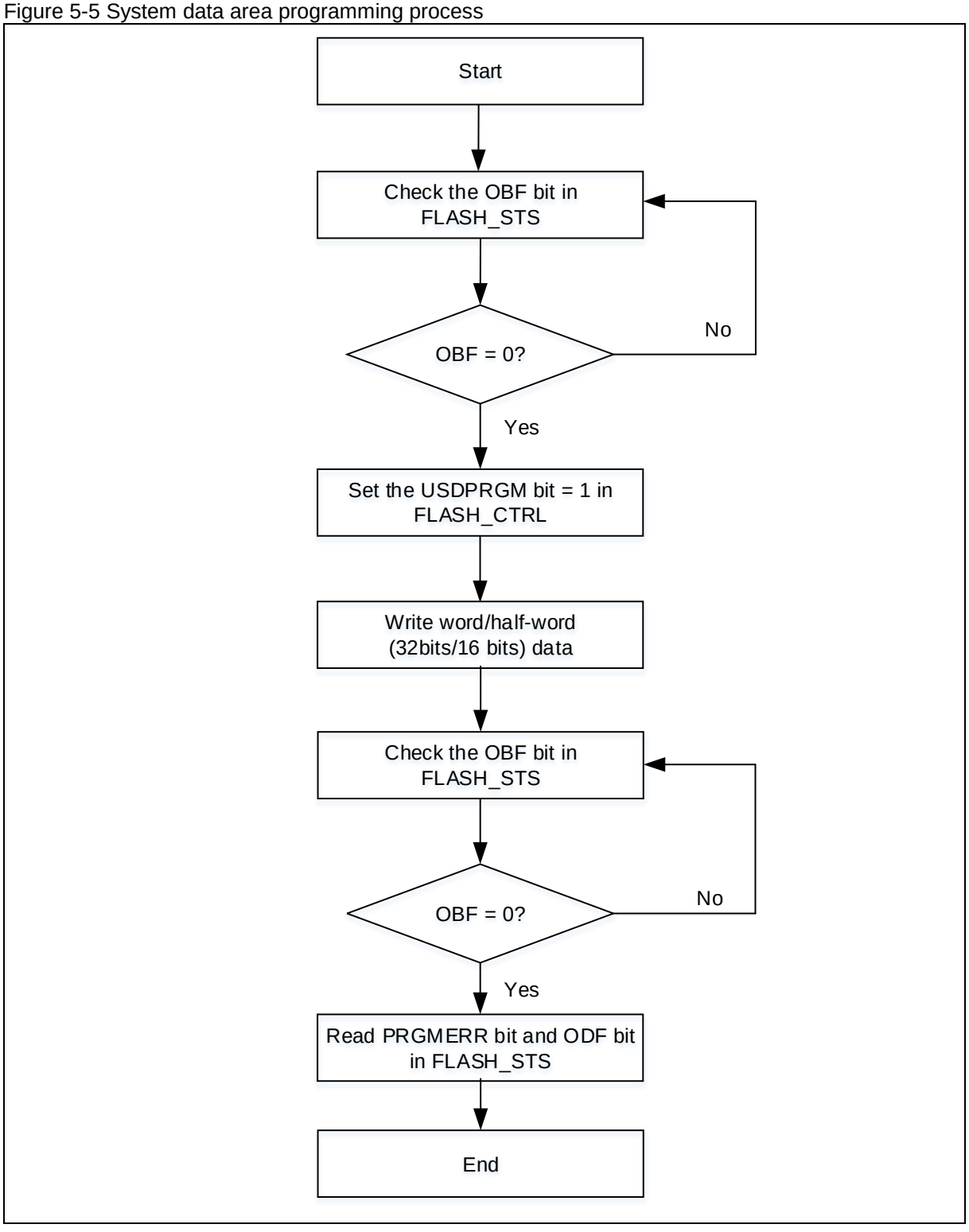
5.4.3 Programming operation

The user system data area can be programmed with 16 bits or 32 bits at a time.
The following process is recommended:

- Check the OBF bit in the FLASH_STS register to confirm that there is no other programming operation in progress;
- Set USDPRGM=1 in the FLASH_CTRL, so that the programming instructions for the user system data area can be received;
- Write the data (word/half-word) to be programmed to the designated address;
- Wait until the OBF bit in the FLASH_STS register becomes “0”. Read the PRGMERR bit and ODF bit in the FLASH_STS register to verify the programming result.

Note:
Read access to the Flash memory during programming halts CPU and waits until the completion of programming. The internal HICK must be enabled prior to programming operation.

Figure 5-5 System data area programming process



5.4.4 Read operation

User system data area can be accessed through AHB bus of the CPU.

5.5 Flash memory protection

Flash memory includes access and erase/program protection.

5.5.1 Access protection

Flash memory access protection is divided into two parts: high-level and low-level.
Once enabled, only the Flash program is allowed to read Flash memory data. This read operation is not permitted in debug mode or by booting from non-Flash memory.

Low-level access protection

When the contents in the nFAP and FAP bytes are different from 0x5A and 0xA5, and 0x33 and 0xCC, the low-level Flash memory access protection is enabled after a system reset.

When the Flash memory is protected, the user can re-erase the system data area, unlock Flash access protection (switching from protected to unprotected state will trigger mass erase on the Flash memory automatically) by writing 0xA5 to FAP byte, and then perform a system reset. Subsequently, the system data loader will be reloaded with system data and updated with Flash memory access protection disable state (FAP byte).

High-level access protection

When the content in the nFAP is equal to 0x33 and the content in the FAP byte is equal to 0xCC, the high-level Flash memory access protection is enabled after a system reset.

Once enabled, it cannot be unlocked, and it is not permissible for users to re-erase and write the system data area.

Note:

1) The main Flash memory extension area can also be protected.

2) If the access protection bit is set in debug mode, then the debug mode has to be cleared by POR instead of system reset in order to resume access to the Flash memory data.

3) The SWD is disabled as soon as the high-level access protection is enabled.

Table 5-5 Flash memory access limit

<table>
<tr><th colspan="2">Block</th><th colspan="6">Protection level</th></tr>
<tr><td colspan="2" rowspan="2"></td><td colspan="3" rowspan="2">In debug mode or boot from SRAM and boot memory</td><td colspan="3">Boot from main Flash memory</td></tr>
<tr><td>Read</td><td>Write</td><td>Erase</td><td>Read</td><td>Write</td><td>Erase</td></tr>
<tr><td rowspan="2">Main Flash memory</td><td>Low-level protection</td><td colspan="2">Not allowed</td><td>Not allowed (1)(2)</td><td colspan="3">Accessible</td></tr>
<tr><td>High-level protection</td><td colspan="3">None (3)</td><td colspan="3">Accessible</td></tr>
<tr><td rowspan="2">User system data area</td><td>Low-level protection</td><td>Not allowed</td><td colspan="2">Accessible</td><td colspan="3">Accessible</td></tr>
<tr><td>High-level protection</td><td colspan="3">None (3)</td><td>Accessible</td><td colspan="2">Not allowed</td></tr>
</table>

(1) The main Flash memory is cleared automatically by hardware when the access protection is disabled;

(2) Only page erase is forbidden, and mass erase is not affected;

(3) When the high-level access protection is enabled, the system automatically boots from the main Flash memory.

5.5.2 Erase/program protection

For 64 KB and less Flash memory, erase/program protection is performed on the basis of 4 pages. This is used to protect the contents in the Flash memory against inadvertent operation when the program crash occurs.

Erase/program operation is not permitted in one of the following cases, and the EPPERR bit is set accordingly when

- Erasing/programming the pages (in Flash memory and its extension area) where erase/program protection is enabled
- Performing mass erase on the pages where erase/program protection is enabled
- When the Flash access protection is enabled, the page0~page3 in the main Flash memory will be protected against erase/program automatically
- When the Flash access protection is enabled, the main Flash memory is protected against erase/program when it is in debug mode when it boots from non-main Flash memory.

5.6 Read performance

Before increasing the system clock frequency, configure the delay time to be inserted when reading the Flash memory according to the description of WTCYC bit in the Flash performance select register (FLASH_PSR).

The Flash memory read times can be reduced through enabling the PFT_EN, PFT_EN2 and PFT_LAT_DIS bits in the Flash performance select register (FLASH_PSR).

5.7 Special functions

5.7.1 Security library settings

Security library is a defined area protected by a code in the main memory. This area is only executable but cannot be written or deleted, unless a correct code is keyed in. Security library includes the instruction security library that cannot be read and the data security library that can be read.

Advantages of security library:

Security library is protected by codes so that solution providers can program core algorithm into this area;
Security library cannot be read or deleted (including ISP/IAP/SWD) but only executed unless code defined by the solution provider is keyed in;
The rest of the area can be used for secondary development by solution providers;
Solution providers can sell core algorithm with security library function and do not have to develop full solutions for every customer;
Security library helps prevent from deliberate damage or changing terminal application codes.

Note:
Security library code must be programmed based on page level, and the start address must be aligned with Flash memory address;
Only CPU is permitted to read instruction security library;
In an attempt of writing or erasing security library code, a warning will be given by EPPERR=1 in the FLASH_STS register.
Executing mass erase in the main memory will not erase the security library.

By default, security library setting register is unreadable and write-protected. To enable write access to this register, security library must be unlocked first by writing 0xA35F6D24 to the SLIB_UNLOCK register, and checking the SLIB_ULKF bit in the SLIB_MISC_STS register to verify if it is unlocked successfully and then writing the programmed value into the security library setting register.

Follow the steps below to enable the security library:

- Check the OBF bit in the FLASH_STS register to confirm that there is no other programming operation in progress;
- Write 0xA35F6D24 to the SLIB_UNLOCK register to unlock security library;
- Check the SLIB_ULKF bit in the SLIB_MISC_STS register to verify that it is unlocked successfully;
- If the security library is located in the main Flash memory, it is necessary to set the pages to be protected (including the addresses of instruction and data areas) in the SLIB_SET_RANGE register; if the security library is in the main Flash memory extension area, it is required to set the EM_SLIB_SET register;
- Wait until the OBF bit becomes “0”;
- Set a password in the SLIB_SET_PWD register;
- Wait until the OBF bit becomes “0”;
- Program the code to be saved in security library;
- Perform system reset, and then reload security library setting word;
- Read the SLIB_STS0/STS1 register to verify the security library settings.

Note:
It is not permitted to configure the main Flash memory and its extension area as the security library simultaneously.
Security library should be enabled when the Flash access protection is disabled.
Follow the steps below to disable the security library:

- Write the programmed password to the SLIB_PWD_CLR register;
- Wait until the OBF bit becomes “0”;
- Perform system reset, and the reload security library setting word;
- Read the SLIB_STS0 register to verify the security library disable state.

Note: Disabling the security library will automatically perform mass erase for the main Flash memory and its extension area and erase the security library setting block.

5.7.2 Boot memory used as Flash memory extension

There is only one chance for users to program the boot memory as the main Flash memory extension area, which will have the same features as those of Flash memory after successful configuration as follows:

- Read the bit 0 in the SLIB_STS0 register to obtain the current mode of the boot memory;
- Write 0xA35F6D24 to the SLIB_UNLOCK register to unlock the current mode of the boot memory;
- Write non-0xFF to the bit [7:0] in the BTM_MODE_SET register;
- Wait until the OBF bit becomes “0”;
- Perform a system reset and reload setting word;
- Read the SLIB_STS0 register to verify the settings.

Note: The main Flash memory extension area must be set before the Flash access protection is enabled.

5.7.3 CRC verify

The optional CRC check for sLib code or user code is performed on a page level.
CRC check procedures are as follows:

- Check the OBF bit in the FLASH_STS register to confirm that there is no other programming operation in progress;
- Program the start address of the code for CRC check in the FLASH_CRC_ADDR register;
- Perform the code count (in terms of pages) to be verified through bit [15:0] in the FLASH_CRC_CTRL register;
- Enable CRC verify by setting the bit 16 in the FLASH_CRC_CTRL register;
- Wait until the OBF bit becomes “0”;
- Read the FLASH_CRC_CHKR register to verify the CRC check result.

Note:
The values of the FLASH_CRC_ADDR register must be aligned with the start address of the page;
CRC verify must not cross the main Flash memory and its extension area.

5.8 Flash memory registers

These peripheral registers must be accessed by words (32 bits).

Table 5-6 Flash memory interface – Register map and reset value

Register	Offset	Reset value
FLASH_PSR	0x00	0x0000 01F0
FLASH_UNLOCK	0x04	0xXXXX XXXX
FLASH_USD_UNLOCK	0x08	0xXXXX XXXX
FLASH_STS	0x0C	0x0000 0000
FLASH_CTRL	0x10	0x0000 0080
FLASH_ADDR	0x14	0x0000 0000
FLASH_USD	0x1C	0x03FF FFFC
FLASH_EPPS	0x20	0xFFFF FFFF
SLIB_STS0	0x74	0x00FF 0000
SLIB_STS1	0x78	0xFFFF FFFF
SLIB_PWD_CLR	0x7C	0xFFFF FFFF
SLIB_MISC_STS	0x80	0x0000 0000

FLASH_CRC_ADDR	0x84	0x0000 0000
FLASH_CRC_CTRL	0x88	0x0000 0000
FLASH_CRC_CHKR	0x8C	0x0000 0000
SLIB_SET_PWD	0x160	0x0000 0000
SLIB_SET_RANGE	0x164	0x0000 0000
EM_SLIB_SET	0x168	0x0000 0000
BTM_MODE_SET	0x16C	0x0000 0000
SLIB_UNLOCK	0x170	0x0000 0000

5.8.1 Flash performance select register (FLASH_PSR)

Bit	Name	Reset value	Type	Description
Bit 31:9	Reserved	0x000000	resd	Kept at its default value.
Bit 8	PFT_LAT_DIS	1	rw	Prefetch latency disable 0: Prefetch latency of Flash is enabled, meaning that accessing buffer requires one system clock cycle; 1: Prefetch latency of Flash is disabled, meaning that accessing buffer requires no wait state. It is recommended to set this bit to 1 and do not change.
Bit 7	PFT_ENF2	1	ro	Prefetch enabled flag 2 When this bit is set, it indicates that the Flash prefetch buffer block 2 is enabled.
Bit 6	PFT_EN2	1	rw	Prefetch enable 2 0: Prefetch buffer block 2 is disabled; 1: Prefetch buffer block 2 is enabled. It is recommended to set this bit to 1 and do not change.
Bit 5	PFT_ENF	1	ro	Prefetch enabled flag When this bit is set, it indicates that the Flash Prefetch is enabled.
Bit 4	PFT_EN	1	rw	Prefetch enable 0: Prefetch is disabled; 1: Prefetch is enabled.
Bit 3	Reserved	0	resd	Kept at its default value.
Bit 2:0	WTCYC	0x0	rw	Wait cycle The wait states depend on the size of the system clock, and they are in terms of system clocks. 000: Zero wait state, for 0 MHz<system clock≤32 MHz 001: One wait state, for 32 MHz<system clock≤64 MHz 010: Two wait states, for 64 MHz<system clock≤80 MHz

5.8.2 Flash unlock register (FLASH_UNLOCK)

Bit	Name	Reset value	Type	Description
Bit 31:0	UKVAL	0xXXXX XXXX	wo	Unlock key value This is used to unlock Flash memory bank and its extension area.

Note: All these bits are write-only, and return 0 when being read.

5.8.3 Flash user system data unlock register (FLASH_USD_UNLOCK)

Bit	Name	Reset value	Type	Description
Bit 31:0	USD_UKVAL	0xXXXX XXXX	wo	User system data unlock key value

Note: All these bits are write-only, and return 0 when being read.

5.8.4 Flash status register (FLASH_STS)

Bit	Name	Reset value	Type	Description
Bit 31:6	Reserved	0x0000000	resd	Kept at its default value.
Bit 5	ODF	0	rwc1	Operation done flag This bit is set by software when Flash memory operations (program/erase) are completed. It is cleared by writing “1”.
Bit 4	EPPERR	0	rwc1	Erase/Program protection error This bit is set by hardware when programming the erase/program-protected Flash memory address. It is cleared by writing “1”.
Bit 3	Reserved	0	resd	Kept at its default value.
Bit 2	PRGMERR	0	rwc1	Program error This bit is set by hardware when the programming address is in “non-erase” state. It is cleared by writing “1”.
Bit 1	Reserved	0	resd	Kept at its default value.
Bit 0	OBF	0	ro	Operation busy flag When this bit is set, it indicates that Flash memory operation is in progress. It is cleared when operation is completed.

5.8.5 Flash control register (FLASH_CTRL)

Bit	Name	Reset value	Type	Description
Bit 31:13	Reserved	0x0000	resd	Kept at its default value.
Bit 12	ODFIE	0	rw	Operation done flag interrupt enable 0: Disabled 1: Enabled
Bit 11,8,3	Reserved	0	resd	Kept at its default value.
Bit 10	ERRIE	0	rw	Error interrupt enable This bit enables EPPERR or PRGMERR interrupt. 0: Disabled 1: Enabled
Bit 9	USDULKS	0	rw	User system data unlock success This bit is set by hardware when the user system data area is unlocked properly, indicating that erase/program operation to the user system data is allowed. It is cleared by writing “0” by software, which will re-lock the user system data area.
Bit 7	OPLK	1	rw	Operation lock This bit is set by default, indicating that Flash memory is protected against operations. This bit is cleared by hardware after unlock, indicating that erase/program operation to Flash memory is allowed. Writing “1” by software can re-lock Flash memory operations.
Bit 6	ERSTR	0	rw	Erasing start An erase operation is triggered when this bit is set. This bit is cleared automatically by hardware after the completion of the erase operation.
Bit 5	USDERS	0	rw	User system data erase It indicates the user system data erase.
Bit 4	USDPRGM	0	rw	User system data program It indicates the user system data program.
Bit 2	BANKERS	0	rw	Bank erase It indicates the bank erase operation.
Bit 1	SECERS	0	rw	Sector erase It indicates the page erase operation.
Bit 0	FPRGM	0	rw	Flash program It indicates Flash program operation.

5.8.6 Flash address register (FLASH_ADDR)

Bit	Name	Reset value	Type	Description
Bit 31:0	FA	0x0000 0000	wo	Flash address Select the address of the banks/pages to be erased.

5.8.7 User system data register (FLASH_USD)

Bit	Name	Reset value	Type	Description
Bit 31:27	Reserved	0x00	resd	Kept at its default value.
Bit 26	FAP_HL	0	ro	Flash access protection high level The status of Flash access protection is determined by bit 26 and bit 1. 00: Flash access protection disabled, and FAP =0xA5 01: Low-level Flash access protection enabled, and FAP=non-0xCC and non-0xA5 10: Reserved 11: High-level Flash access protection enabled, and FAP=0xCC
Bit 25:18	USER_D1	0xFF	ro	User data 1
Bit 17:10	USER_D0	0xFF	ro	User data 0
Bit 9:2	SSB	0xFF	ro	System setting byte Include the system setting bytes in the loaded user system data area. Bit 9: nRAM_PRT_CHK Bit 8: nSTDBY_WDT Bit 7: nDEPSLP_WDT Bit 6: nBOOT1 Bit 5: Unused Bit 4: nSTDBY_RST Bit 3: nDEPSLP_RST Bit 2: nWDT_ATO_EN
Bit 1	FAP	0	ro	Flash access protection
Bit 0	USDERR	0	ro	User system data error When this bit is set, it indicates that certain byte does not match its inverse code in the user system data area. At this point, this byte and its inverse code will be forced to 0xFF by hardware when being read.

5.8.8 Erase/program protection status register (FLASH_EPPS)

Bit	Name	Reset value	Type	Description
Bit 31:0	EPPS	0xFFFF FFFF	ro	Erase/Program protection status This register reflects the erase/program protection byte status in the loaded user system data.

5.8.9 Flash security library status register 0 (SLIB_STS0)

For Flash security library only.

Bit	Name	Reset value	Type	Description
Bit 31:24	Reserved	0x00	resd	Kept at its default value.
Bit 23:16	EM_SLIB_INST_SS	0xFF	ro	Extension memory sLib instruction start page 00000000: Page 0 00000001: Page 1 00000010: Page 2 00000011: Page 3 11111111: None Others: Invalid
Bit 15:4	Reserved	0x000	resd	Kept at its default value.
Bit 3	SLIB_ENF	0	ro	sLib enabled flag When this bit is set, it indicates that the main Flash memory is partially or completely (depending on the setting of SLIB_STS1) used as security library code.
Bit 2	EM_SLIB_ENF	0	ro	Extension memory sLib enabled flag When this bit is set, it indicates that the boot memory is used as the main Flash extension area (BTM_AP_ENF is

				set) and stores security library code.
Bit 1	Reserved	0	resd	Kept at its default value.
Bit 0	BTM_AP_ENF	0	ro	Boot memory store application code enabled flag When this bit is set, it indicates that the boot memory can be used as the main Flash extension area to store user application code; otherwise, it is only used for system boot code.

5.8.10 Flash security library status register 1 (SLIB_STS1)

For Flash security library only.

Bit	Name	Reset value	Type	Description
Bit 31:22	SLIB_ES	0x3FF	ro	sLib end page 0000000000: Page 0 0000000001: Page 1 0000000010: Page 2 … 0000001111: Page 15 (the last page of 16 KB main Flash memory) … 0000011111: Page 31 (the last page of 32 KB main Flash memory) … 0000111111: Page 63 (the last page of 64 KB main Flash memory)
Bit 21:11	SLIB_INST_SS	0x7FF	ro	sLib instruction start page 00000000000: Page 0 00000000001: Page 1 00000000010: Page 2 … 00000001111: Page 15 (the last page of 16 KB main Flash memory) … 00000011111: Page 31 (the last page of 32 KB main Flash memory) … 00000111111: Page 63 (the last page of 64 KB main Flash memory) 11111111111: None
Bit 10:0	SLIB_SS	0x7FF	ro	sLib start page 00000000000: Page 0 00000000001: Page 1 00000000010: Page 2 … 00000001111: Page 15 (the last page of 16 KB main Flash memory) … 00000011111: Page 31 (the last page of 32 KB main Flash memory) … 00000111111: Page 63 (the last page of 64 KB main Flash memory)

5.8.11 Flash security library password clear register (SLIB_PWD_CLR)

For Flash security library only.

Bit	Name	Reset value	Type	Description
Bit 31:0	SLIB_PCLR_VAL	0x0000 0000	wo	sLib password clear value This register is used to key in a correct sLib password in order to unlock sLib function. The write status of this register is indicated by bit 0 and bit 1 of the SLIB_MISC_STS register.

5.8.12 Security library additional status register (SLIB_MISC_STS)

For Flash security library only.

Bit	Name	Reset value	Type	Description
Bit 31:3	Reserved	0x0000000	resd	Kept at its default value.
Bit 2	SLIB_ULKF	0	ro	sLib unlock flag When this bit is set, it indicates that sLib-related setting registers can be configured.
Bit 1	SLIB_PWD_OK	0	ro	sLib password ok This bit is set by hardware when the password is correct.
Bit 0	SLIB_PWD_ERR	0	ro	sLib password error This bit is set by hardware when the password is incorrect and the setting value of the password clear register is different from 0xFFFF FFFF. Note: When this bit is set, the hardware will no longer agree to re-program the password clear register until the next reset.

5.8.13 Flash CRC address register (FLASH_CRC_ADDR)

For main Flash memory and its extension area.

Bit	Name	Reset value	Type	Description
Bit 31:0	CRC_ADDR	0x0000 0000	wo	CRC address This register is used to select a start address of a page to be CRC checked. Note: It must be aligned with the page start address.

Note: All these bits are write-only, and return no response when being read.

5.8.14 Flash CRC control register (FLASH_CRC_CTRL)

For main Flash memory and its extension area.

Bit	Name	Reset value	Type	Description
Bit 31:17	Reserved	0x0000	resd	Kept at its default value.
Bit 16	CRC_STRT	0	wo	CRC start This bit is used to enable CRC check for user code or sLib code. It is automatically cleared after enabling CRC by hardware. Note: CRC data ranges from CRC_ADDR to CRC_ADDR+CRC_SN*1.
Bit 15:0	CRC_SN	0x0000	wo	CRC sector number This bit defines the pages to be CRC checked.

5.8.15 Flash CRC check result register (FLASH_CRC_CHKR)

For main Flash memory and its extension area.

Bit	Name	Reset value	Type	Description
Bit 31:0	CRC_CHKR	0x0000 0000	ro	CRC check result

Note: All these bits are read-only, and return no response when being written.

5.8.16 Security library password setting register (SLIB_SET_PWD)

For Flash security library password setting only.

Bit	Name	Reset value	Type	Description
Bit 31:0	SLIB_PSET_VAL	0x0000 0000	wo	sLib password setting value Note: This register can be written after sLib is unlocked. It is used to set a password for sLib. Writing 0xFFFF_FFFF or 0x0000_0000 has no effect.

Note: All these bits are write-only, and return 0 when being read.

5.8.17 Security library address setting register (SLIB_SET_RANGE)

For Flash security library address setting only.

Bit	Name	Reset value	Type	Description
Bit 31:22	SLIB_ES_SET	0x000	wo	sLib end page setting These bits are used to set the security library end page. 0000000000: Page 0 0000000001: Page 1 0000000010: Page 2 … 0000001111: Page 15 (the last page of 16 KB main Flash memory) … 0000011111: Page 31 (the last page of 32 KB main Flash memory) … 0000111111: Page 63 (the last page of 64 KB main Flash memory)
Bit 21:11	SLIB_ISS_SET	0x000	wo	sLib instruction start page setting These bits are used to set the security library instruction start page. 00000000000: Page 0 00000000001: Page 1 00000000010: Page 2 … 00000001111: Page 15 (the last page of 16 KB main Flash memory) … 00000011111: Page 31 (the last page of 32 KB main Flash memory) … 00000111111: Page 63 (the last page of 64 KB main Flash memory) 11111111111: No security library instruction area
Bit 10:0	SLIB_SS_SET	0x000	wo	sLib start page setting These bits are used to set the security library start page. 0000000000: Page 0 0000000001: Page 1 0000000010: Page 2 … 0000001111: Page 15 (the last page of 16 KB main Flash memory) … 0000011111: Page 31 (the last page of 32 KB main Flash memory) … 0000111111: Page 63 (the last page of 64 KB main Flash memory)

Note:
All these bits are write-only, and return 0 when being read.
The register can be written only after unlocking the security library.
Being out of the Flash address range is an invalid setting.

5.8.18 Flash extension memory security library setting register (EM_SLIB_SET)

For Flash extension area only.

Bit	Name	Reset value	Type	Description
Bit 31:24	Reserved	0x00	resd	Kept at its default value.
Bit 23:16	EM_SLIB_ISS_SET	0x000	wo	Extension memory sLib instruction start page setting These bits are used to set the security library instruction start page. 00000000: Page 0 00000001: Page 1 00000010: Page 2 00000011: Page 3

				11111111: No sLib instruction area Others: Invalid Note: When it is set to 0xFF, it indicates that the Flash memory extension area from page 0 to page 3 is the security library, and the entire security library is read-only.
Bit 15:0	EM_SLIB_SET	0x000	wo	Extension memory sLib setting Extension memory is configured as security library by writing 0x5AA5.

Note: All these bits are write-only, and return no response when being read.

5.8.19 Boot memory mode setting register (BTM_MODE_SET)

For boot memory only.

Bit	Name	Reset value	Type	Description
Bit 31:8	Reserved	0x000000	resd	Kept at its default value.
Bit 7:0	BTM_MODE_SET	0x00	wo	Boot memory mode setting 0xFF: Boot memory is used as the system area that stores system boot code. Others: Boot memory serves a Flash extension area that store application code. Note: The register can be set only when Flash access protection is disabled.

Note: All these bits are write-only, and return no response when being read.

5.8.20 Security library unlock register (SLIB_UNLOCK)

For security library register unlock only.

Bit	Name	Reset value	Type	Description
Bit 31:0	SLIB_UKVAL	0x0000 0000	wo	sLib unlock key value Fixed key value is 0xA35F_6D24, used for security library setting register unlock.

Note: All these bits are write-only, and return 0 when being read.

6 GPIOs and IOMUX

6.1 Introduction

AT32L021 series supports up to 39 bidirectional I/O pins that are divided into four sets, namely PA0-PA15, PB0-PB15, PC13-PC15, PF0-PF1 and PF6-PF7. Each of these pins features communication, control and data collection. In addition, their main features also include:

- Support general-purpose I/O (GPIO) or multiplexed function I/O (IOMUX).
- Each pin can be configured by software as floating input, pull-up/pull-down input, analog input/output, push-pull/open-drain output, multiplexed push-pull/open-drain output.
- Each pin has individual weak pull-up/pull-down capability.
- Each pin's output drive capability is configurable by software.
- Each pin can be configured as external interrupt input.
- Each pin can be locked.

6.2 Function overview

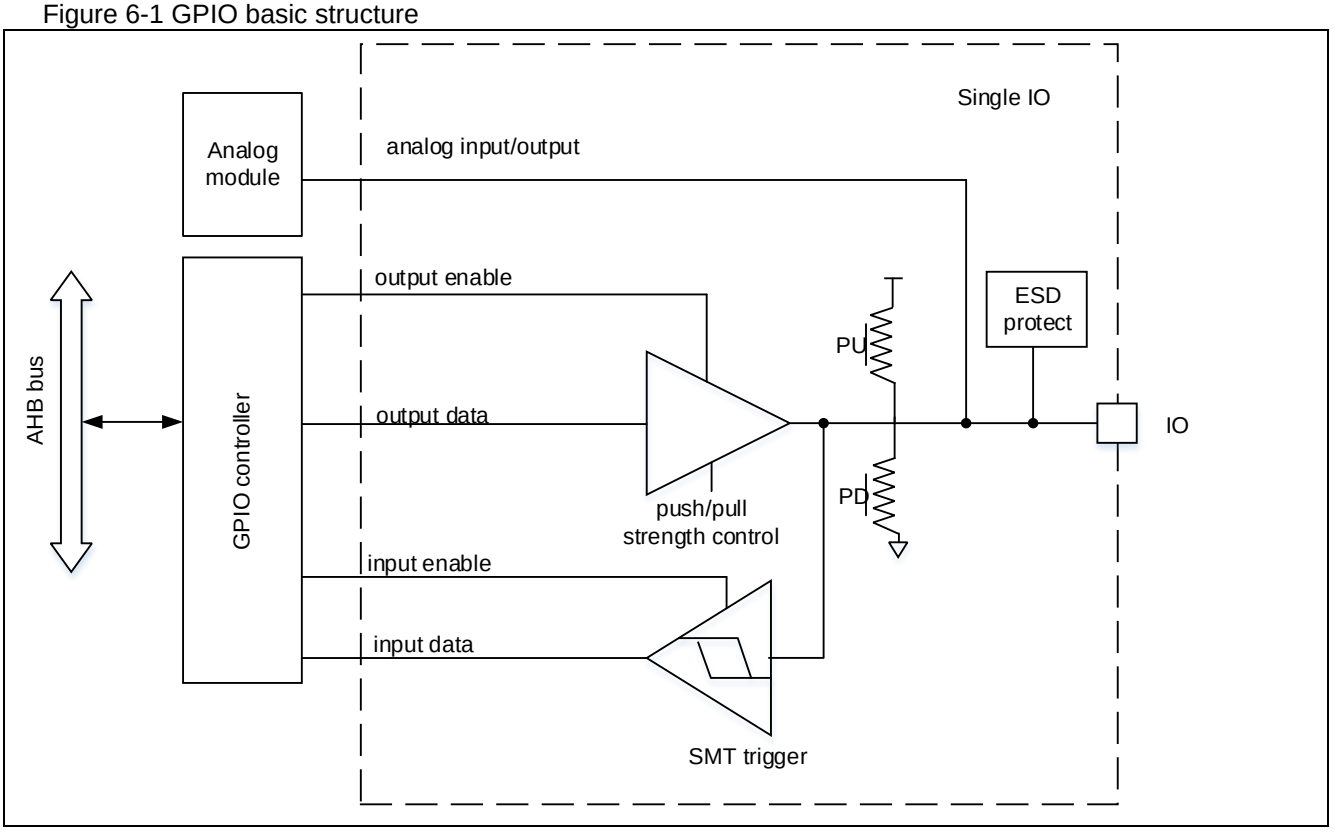
6.2.1 GPIO structure

Each of the GPIO pins can be configured by software as four input modes (floating, pull-up/pull-down and analog input) and four output modes (open-drain, push-pull, multiplexed push-pull/open-drain output).

I/O port registers can be accessed by byte, half-word and word. Each I/O port bit can be programmed freely.

The figure below show the basic structure of an I/O port bit.

Figure 6-1 GPIO basic structure



6.2.2 GPIO reset status

After power-on or system reset, all pins are configured as analog mode except SWD-related pins. SWD-related pins are configured as follows:

PA13/SWDIO in multiplexed function pull-up mode;

PA14/SWCLK in multiplexed function pull-down mode.

6.2.3 General-purpose input configuration

Mode	IOMC	PUPD
Floating input	00	00
Pull-down input		10
Pull-up input		01

When I/O port is configured as input:

- Get I/O states by reading the input data register.
- Support floating input, pull-up/pull-down input configuration.
- Schmitt-trigger input is activated.
- Output is disabled.

Note: In floating input mode, it is recommended to set the unused pins as analog input mode in order to avoid leakage caused by interference from unused pins in a complex environment.

6.2.4 Analog input/output configuration

Mode	IOMC	PUPD
Analog input/output	11	Unused

When GPIO port is configured as analog input:

- Schmitt-trigger input is disabled.
- Digital input/output is disabled.
- Without any pull-up/pull-down resistor.

6.2.5 General-purpose output configuration

Mode	IOMC	OM	HDRV	ODRV[1:0]	PUPD
Push-pull without pull-up/pull-down	01	0	000: output mode, normal sourcing/sinking strength 001: output mode, large sourcing/sinking strength		00 or 11
Push-pull with pull-up	01	0	010: output mode, normal sourcing/sinking strength		01
Push-pull with pull-down	01	0	011: output mode, normal sourcing/sinking strength 1xx: output mode, maximum sourcing/sinking strength		10
Open-drain without pull-up/pull-down	01	1	000: output mode, normal sourcing/sinking strength 001: output mode, maximum sourcing/sinking strength		00 or 11
Open-drain with pull-up	01	1	010: output mode, normal sourcing/sinking strength		01
Open-drain with pull-down	01	1	011: output mode, normal sourcing/sinking strength 1xx: output mode, maximum sourcing/sinking strength		10

When GPIO port is configured as output:

- Schmitt-trigger input is enabled.
- Output through output register.
- In open-drain mode, forced output 0, and use pull-up resistor to output 1.
- In push-pull mode, output 0/1 using output register.
- GPIO set/clear register is used to set or clear the corresponding GPIO data output register.

Note: When writing "1" to the IOCB/IOSB bits of the GPIO set/clear register, IOSB has priority over IOCB.

6.2.6 GPIO port protection

Locking mechanism can freeze the I/O configuration for the purpose of protection. When LOCK is applied to a port bit, its configuration cannot be modified until the next reset or power on.

6.2.7 IOMUX structure

Most of the pins support output function mapping for multiple peripherals. It is possible to select the peripheral input/output functions for each pin by using the IOMUX input/output checklist described in the section of IOMUX input/output. The multiplexed function of pins is configured using the corresponding GPIO multiplexed register low (GPIOx_MUXL) (for pin 0 to pin 7) or the GPIO multiplexed register high GPIOx_MUXH (for pin 8 to pin 15). A single pin has up to 8 different IOMUX mapping configurations for flexible selection.

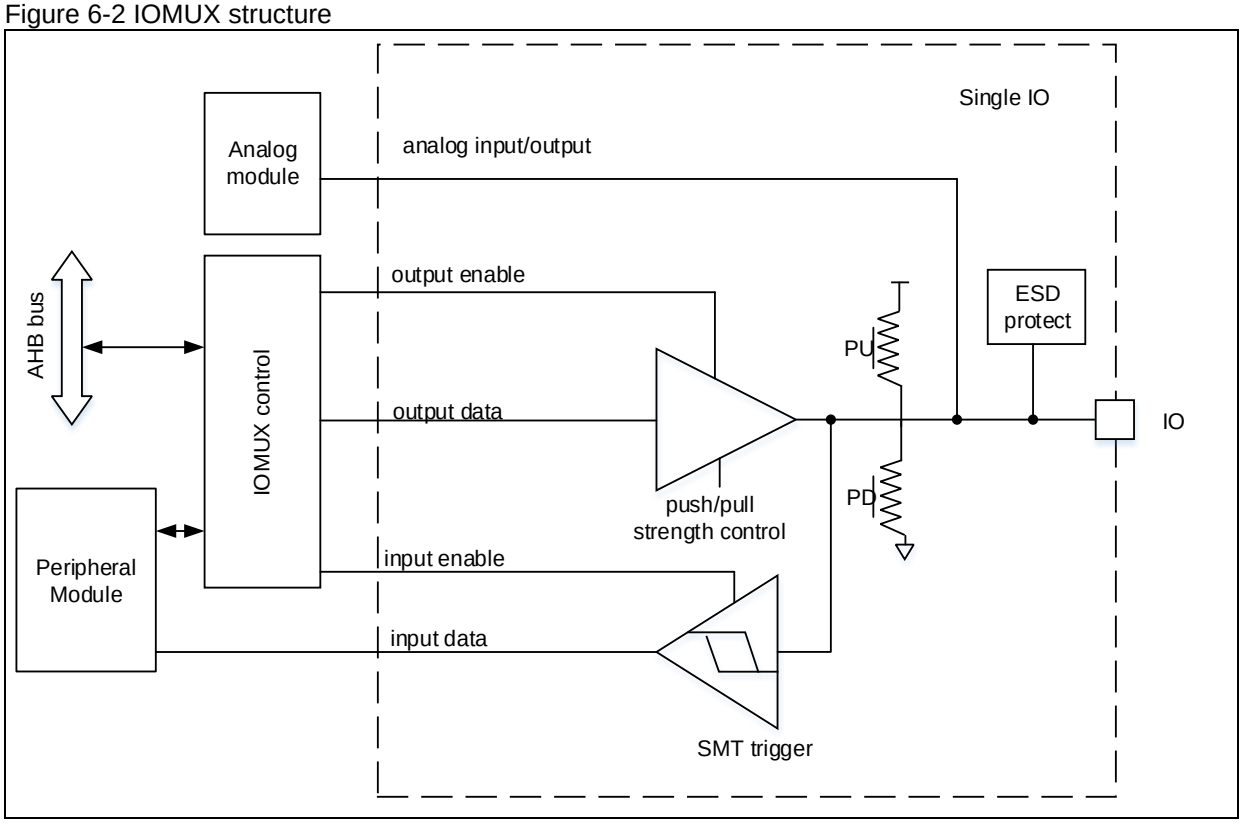
It is only allowed to select one of the peripheral functions for each pin by setting the GPIOx_MUXL or GPIOx_MUXH register. Hence, it is not possible that a single pin is occupied by several peripherals at a time.

When used as multiplexed function input, the port should be configured as input mode (floating, pull-up and pull-down).

To enable multiplexed function output, the port must be configured as multiplexed function output mode and push-pull or open-drain mode by setting GPIOx_CFGR or GPIOx_OMODE register. In this case, the pin is disconnected from GPIO controller, and is controlled by IOMUX controller, instead.

To achieve bidirectional multiplexed function, the port needs to be configured as multiplexed function output mode (push-pull or open-drain), controlled by IOMUX controller.

Figure 6-2 IOMUX structure



6.2.8 Multiplexed function pull-up/pull-down configuration

Mode	IOMC	PUPD
Multiplexed function floating	10	00
Multiplexed function pull-down		10
Multiplexed function pull-up		01

When I/O ports are configured as multiplexed function input:

- Get I/O pin state by reading input data registers;
- Support floating input, pull-up or pull-down input configuration;
- Schmitt-trigger input is valid;
- GPIO pin output is disabled.

6.2.9 IOMUX input/output

The selection of the valid multiplexed functions for each port is done by the GPIOx_MUXL (for pin 0 to pin 7) or GPIOx_MUXH (for pin 8 to pin 15) registers.

Table 6-1 Port A multiplexed function configuration with GPIOA_MUX* register

Pin	MUX0	MUX1	MUX2	MUX3	MUX4	MUX5	MUX6	MUX7
PA0	USART2_RX	USART2_CTS		I2C2_SCL	USART4_TX	TMR1_ETR		
PA1		USART2_RTS		I2C2_SDA	USART4_RX	TMR15_CH1C	I2C1_SMBA	EVENTOUT
PA2	TMR15_CH1	USART2_TX			CAN_RX			
PA3	TMR15_CH2	USART2_RX			CAN_TX	I2S2_MCK		
PA4	SPI1_NSS / I2S1_WS	USART2_CK			TMR14_CH1	I2C1_SCL	SPI2_NSS / I2S2_WS	
PA5	SPI1_SCK / I2S1_CK			USART3_CK	USART3_RX			
PA6	SPI1_MISO / I2S1_MCK	TMR3_CH1	TMR1_BKIN	USART3_RX	USART3_CTS	TMR16_CH1	I2S2_MCK	EVENTOUT
PA7	SPI1_MOSI / I2S1_SD	TMR3_CH2	TMR1_CH1C	USART3_TX	TMR14_CH1	TMR17_CH1	EVENTOUT	
PA8	CLKOUT	USART1_CK	TMR1_CH1	EVENTOUT	USART2_TX			I2C2_SCL
PA9	TMR15_BKIN	USART1_TX	TMR1_CH2		I2C1_SCL	CLKOUT		I2C2_SMBA
PA10	TMR17_BKIN	USART1_RX	TMR1_CH3		I2C1_SDA	RTC_REFIN		I2C2_SMBA
PA11		USART1_CTS	TMR1_CH4		CAN_RX	I2C2_SCL	I2C1_SMBA	EVENTOUT
PA12		USART1_RTS_DE	TMR1_ETR		CAN_TX	I2C2_SDA		EVENTOUT
PA13	SWDIO	IR_OUT				I2C1_SDA	SPI2_MISO / I2S2_MCK	
PA14	SWCLK	USART2_TX				I2C1_SMBA	SPI2_MOSI / I2S2_SD	
PA15	SPI1_NSS / I2S1_WS	USART2_RX		EVENTOUT	USART4_RTS_DE		SPI2_NSS / I2S2_WS	

Table 6-2 Port B multiplexed function configuration with GPIOB_MUX* register

Pin	MUX0	MUX1	MUX2	MUX3	MUX4	MUX5	MUX6	MUX7
PB0		TMR3_CH3	TMR1_CH2C	USART2_RX	USART3_CK	EVENTOUT	I2S1_MCK	SPI1_MISO /I2S1_MCK
PB1	TMR14_CH1	TMR3_CH4	TMR1_CH3C	USART2_CK	USART3_RTS_DE		SPI2_SCK / I2S2_CK	SPI1_MOSI / I2S1_SD
PB2			TMR3_ETR					I2C1_SMBA
PB3	SPI1_SCK / I2S1_CK	EVENTOUT			USART1_RTS_DE	USART2_CTS	SPI2_SCK / I2S2_CK	
PB4	SPI1_MISO / I2S1_MCK	TMR3_CH1	EVENTOUT		USART1_CTS	TMR17_BKIN	SPI2_MISO / I2S2_MCK	I2C2_SDA
PB5	SPI1_MOSI / I2S1_SD	TMR3_CH2	TMR16_BKIN	I2C1_SMBA	USART1_CK	USART2_RTS_DE	SPI2_MOSI / I2S2_SD	
PB6	USART1_TX	I2C1_SCL	TMR16_CH1C		USART4_CK		I2S1_MCK	
PB7	USART1_RX	I2C1_SDA	TMR17_CH1C		USART4_CTS			
PB8	USART1_TX	I2C1_SCL	TMR16_CH1	EVENTOUT	CAN_RX			
PB9	IR_OUT	I2C1_SDA	TMR17_CH1	EVENTOUT	CAN_TX		I2S1_MCK	SPI2_NSS / I2S2_WS
PB10		I2C2_SCL			USART3_TX			SPI2_SCK / I2S2_CK
PB11		I2C2_SDA		EVENTOUT	USART3_RX			
PB12	SPI2_NSS / I2S2_WS	EVENTOUT	TMR1_BKIN		USART3_CK	TMR15_BKIN		I2C2_SMBA
PB13	SPI2_SCK / I2S2_CK	TMR15_CH1C	TMR1_CH1C	CLKOUT	USART3_CTS	I2C2_SCL		
PB14	SPI2_MISO / I2S2_MCK	TMR15_CH1	TMR1_CH2C		USART3_RTS_DE	I2C2_SDA		
PB15	SPI2_MOSI / I2S2_SD	TMR15_CH2	TMR1_CH3C	TMR15_CH1C		RTC_REFIN		

Table 6-3 Port C multiplexed function configuration with GPIOC_MUX* register

Pin	MUX0	MUX1	MUX2	MUX3	MUX4	MUX5	MUX6	MUX7
PC13								
PC14								
PC15								

Table 6-4 Port F multiplexed function configuration with GPIOF_MUX* register

Pin	MUX0	MUX1	MUX2	MUX3	MUX4	MUX5	MUX6	MUX7
PF0		I2C1_SDA	TMR1_CH1					
PF1		I2C1_SCL	TMR1_CH2C				SPI2_NSS / I2S2_WS	
PF6	I2C2_SCL				USART4_RX			
PF7	I2C2_SDA				USART4_TX			

Note: EVENTOUT is the Cortex-MT TXEV signal.

6.2.10 Peripheral multiplexed function configuration

IOMUX function configuration is as follows:

- To use a peripheral pin in MUX output, it is configured as multiplexed push-pull/open-drain output.
- To use a peripheral pin in MUX input, it is configured as multiplexed floating/pull-up/pull-down input.
- For ADC peripherals, the pins of analog channels should be configured as analog input/output mode.
- For I^2C peripherals that intend to use pins as bidirectional functions, open-drain mode is required.

6.2.11 IOMUX mapping priority

The unique peripheral multiplexed function can be configured through the GPIOx_MUXL/GPIOx_MUXH register, except individual pins that may be directly owned by hardware.
Some pins have been directly owned by specific hardware feature, whatever GPIO configuration.

Table 6-5 Pins owned by hardware

Pin	Enable bit	Description
PA0	PWC_CTRLTS[8] =1	Once enabled, PA0 pin acts as WKUP1 of PWC.
PC13	PWC_CTRLTS[9] =1	Once enabled, PC13 pin acts as WKUP2 of PWC.
PA2	PWC_CTRLTS[11] =1	Once enabled, PA2 pin acts as WKUP4 of PWC.
PB5	PWC_CTRLTS[13] =1	Once enabled, PB5 pin acts as WKUP6 of PWC.
PB15	PWC_CTRLTS[14] =1	Once enabled, PB15 pin acts as WKUP7 of PWC.
PC13	(ERTC_CTRL[23]=1)| (ERTC_CTRL[22:21]!=00)| (ERTC_CTRL[11]=1& ERTC_TAMP[17]=0)| (ERTC_TAMP[0]=1& ERTC_TAMP[16]=0)	Once enabled, PC13 is used as RTC channel.
PC14	CRM_BPDC[0]=1	Once enabled, PC14 is used as LEXT channel.
PC15	CRM_BPDC[0]=1 & CRM_BPDC[2]=0	Once enabled, PC15 is used as LEXT channel.
PF0	CRM_CTRL[16]=1	Once enabled, PF0 is used as HEXT channel.
PF1	CRM_CTRL[16]=1& CRM_CTRL[18]=0	Once enabled, PF1 is used as HEXT channel.

Note: PC13 cannot enable TAMPER_BPR and WKUP of PWC simultaneously.

6.2.12 External interrupt/wake-up lines

Each pin can be used as an external interrupt input, and thus the corresponding pin must be configured as input mode.

6.3 GPIO registers

The table below lists GPIO register map and their reset values.
These peripheral registers can be accessed by byte (8 bits), half-word (16 bits) or word (32 bits).

Table 6-6 GPIO register map and reset values

Register	Offset	Reset value
GPIOA_CFGR	0x00	0xEBFF FFFF
GPIOx_CFGR(x =B,C,F)	0x00	0xFFFF FFFF
GPIOx_OMODER	0x04	0x0000 0000
GPIOA_ODRVR	0x08	0x0C00 0000
GPIOx_ODRVR(x =B,C,F)	0x08	0x0000 0000
GPIOA_PULL	0x0C	0x2400 0000
GPIOx_PULL(x = B,C,F)	0x0C	0x0000 0000
GPIOx_IDT	0x10	0x0000 XXXX
GPIOx_ODT	0x14	0x0000 0000
GPIOx_SCR	0x18	0x0000 0000
GPIOx_WPR	0x1C	0x0000 0000

GPIOx_MUXL	0x20	0x0000 0000
GPIOx_MUXH	0x24	0x0000 0000
GPIOx_CLR	0x28	0x0000 0000
GPIOx_TOGR	0x2C	0x0000 0000
GPIOx_HDRV	0x3C	0x0000 0000

6.3.1 GPIO configuration register (GPIOx_CFGR) (x=A..C,F)

Address offset: 0x00
Reset value: 0xEBFFFFFF for port A
0xFFFFFFFF for other ports

Bit	Name	Reset value	Type	Description
Bit 2y+1:2y	IOMCy	0xEBFF FFFF	rw	GPIOx mode configuration (y=0~15) These bits are used to configure the operating modes of GPIOx: 00: Input mode 01: General-purpose output 10: Multiplexed function 11: Analog

6.3.2 GPIO output mode register (GPIOx_OMODE) (x=A..C,F)

Bit	Name	Reset value	Type	Description
Bit 31:16	Reserved	0x0000	resd	Always 0.
Bit 15:0	OM	0x0000	rw	GPIOx output mode configuration (y=0..15) When GPIOx is used as output, there are two output modes for selection: 0: Push-pull (reset state) 1: Open-drain

6.3.3 GPIO drive capability register (GPIOx_ODRVR) (x=A..C,F)

Address offset: 0x08
Reset value: 0x0C00 0000 for port A
0x00000000 for other ports

Bit	Name	Reset value	Type	Description
Bit 2y+1:2y	ODRVy	0x0C00 0000	rw	GPIOx drive capability (y=0…15) This field is used to configure the I/O port drive capability. x0: Normal sourcing/sinking strength 01: Large sourcing/sinking strength 11: Normal sourcing/sinking strength

6.3.4 GPIO pull-up/pull-down register (GPIOx_PULL) (x=A..C,F)

Address offset: 0x0C
Reset value: 0x2400 0000 for port A
0x00000000 for other ports

Bit	Name	Reset value	Type	Description
Bit 2y+1:2y	PULLy	0x2400 0000	rw	GPIOx pull configuration (y=0…15) This field is used to configure the pull-up/pull-down of the I/O port. 00: No pull-up/pull-down 01: Pull-up 10: Pull-down

6.3.5 GPIO input data register (GPIOx_IDT) (x=A..C,F)

Bit	Name	Reset value	Type	Description
Bit 31:16	Reserved	0x0000	resd	Always 0.
Bit 15:0	IDT	0xXXXX	ro	GPIOx input data Indicate the input status of I/O port. Each bit corresponds to an I/O.

6.3.6 GPIO output data register (GPIOx_ODT) (x=A..C,F)

Bit	Name	Reset value	Type	Description
Bit 31:16	Reserved	0x0000	resd	Always 0.
Bit 15:0	ODT	0x0000	rw	GPIOx output data Each bit corresponds to an I/O. It indicates the output level status of I/O port. 0: Low 1: High

6.3.7 GPIO set/clear register (GPIOx_SCR) (x=A..C,F)

Bit	Name	Reset value	Type	Description
Bit 31:16	IOCB	0x0000	wo	GPIOx clear bit The corresponding ODT register bit is cleared by writing “1” to these bits. Otherwise, the corresponding ODT register bit remains unchanged, which acts as ODT register bit operations. 0: No action to the corresponding ODT bits 1: Clear the corresponding ODT bits
Bit 15:0	IOSB	0x0000	wo	GPIOx set bit The corresponding ODT register bit is set by writing “1” to these bits. Otherwise, the corresponding ODT register bit remains unchanged, which acts as ODT register bit operations. If write “1” to the same bit of IOCB and IOSB, the IOSB with higher priority will take effect. 0: No action to the corresponding ODT bits 1: Set the corresponding ODT bits

6.3.8 GPIO write protection register (GPIOx_WPR) (x=A..C,F)

Bit	Name	Reset value	Type	Description
Bit 31:17	Reserved	0x0000	resd	Kept at its default value.
Bit 16	WPSEQ	0x0	rw	Write protect sequence Write protect enable sequence bit and WPEN bit must be enabled at the same time to achieve write protection for some I/O bits. Write protect enable bit is executed four times in the order below: write “1” -> write “0” -> write “1” -> read. Note that the value of WPEN bit cannot be modified during this period.
Bit 15:0	WPEN	0x0000	rw	Write protect enable Each bit corresponds to an I/O port. 0: No effect 1: Write protect

6.3.9 GPIO multiplexed function low register (GPIOx_MUXL) (x=A..C,F)

Bit	Name	Reset value	Type	Description
Bit 4y+3:4y	MUXLy	0x0	rw	Multiplexed function select for GPIOx pin y (y=0…7) This field is used to configure multiplexed function I/Os. 0000: MUX0 0001: MUX1 0010: MUX2 0011: MUX3 0100: MUX4 0101: MUX5 0110: MUX6 0111: MUX7 1xxx: Reserved

6.3.10 GPIO multiplexed function high register (GPIOx_MUXH) (x=A..C,F)

Bit	Name	Reset value	Type	Description
Bit 4y+3:4y	MUXHy	0x0	rw	Multiplexed function select for GPIOx pin y (y=8…15) This field is used to configure multiplexed function I/Os. 0000: MUX0 0001: MUX1 0010: MUX2 0011: MUX3 0100: MUX4 0101: MUX5 0110: MUX6 0111: MUX7 1xxx: Reserved

6.3.11 GPIO port bit clear register (GPIOx_CLR) (x=A..C,F)

Bit	Name	Reset value	Type	Description
Bit 31:16	Reserved	0x0000	resd	Kept at its default value.
Bit 15:0	IOCB	0x0000	wo	GPIOx clear bit The corresponding ODT register bit is cleared by writing “1” to these bits. Otherwise, the corresponding ODT register bit remains unchanged, which acts as ODT register bit operations. 0: No action to the corresponding ODT bits 1: Clear the corresponding ODT bits

6.3.12 GPIO port bit toggle register (GPIOx_TOGR) (x=A..C,F)

Bit	Name	Reset value	Type	Description
Bit 31:16	Reserved	0x0000	resd	Kept at its default value.
Bit 15:0	IOTB	0x0000	wo	GPIOx toggle bit The corresponding ODT register bit is toggled by writing “1” to these bits. Otherwise, the corresponding ODT register bit remains unchanged, which acts as ODT register bit operations. 0: No action to the corresponding ODT bits 1: Toggle the corresponding ODT bits

6.3.13 GPIO huge current control register (GPIOx_HDRV) (x=A..C,F)

Bit	Name	Reset value	Type	Description
Bit 31:16	Reserved	0x0000	resd	Kept at its default value.
Bit 15:0	HDRV	0x0000	rw	Huge sourcing/sinking strength control 0: Not active 1: GPIO is configured as huge sourcing/sinking strength

7 System configuration controller (SCFG)

7.1 Introduction

This device contains a set of system configuration register. The system configuration controller is mainly used to:

- Manage the external interrupts connected to GPIOs
- Control the memory mapping mode
- Manage IRTMR GPIO configurations

7.2 SCFG registers

The table below shows SCFG register map and their reset values.
These peripheral registers can be accessed by words (32 bits).

Table 7-1 SCFG register map and reset values

Register	Offset	Reset value
SCFG_CFG1	0x00	0x0000 000X
SCFG_EXINTC1	0x08	0x0000 0000
SCFG_EXINTC2	0x0C	0x0000 0000
SCFG_EXINTC3	0x10	0x0000 0000
SCFG_EXINTC4	0x14	0x0000 0000
SCFG_CFG2	0x18	0x0000 0000

7.2.1 SCFG configuration register 1 (SCFG_CFGR1)

Bit	Name	Reset value	Type	Description
Bit 31:18	Reserved	0x000	resd	Kept at its reset value.
Bit 17	PB9_UH	0x0	rw	PB9 ultra high sourcing/sinking strength This bit is written by software to control the PB9 PAD sourcing/sinking strength. 0: Not active 1: The corresponding GPIO is switched to ultra-high sourcing/sinking strength. When this bit is set, the control bit of GPIOx_OTYPER&GPIOx_HDRV becomes invalid.
Bit 16	PB8_UH	0x0	rw	PB8 Ultra high sourcing/sinking strength This bit is written by software to control the PB8 PAD sourcing/sinking strength. 0: Not active 1: The corresponding GPIO is switched to ultra-high sourcing/sinking strength. When this bit is set, the control bit of GPIOx_OTYPER&GPIOx_HDRV becomes invalid.
Bit 15:8	Reserved	0x0	resd	Kept at its reset value.
Bit 7:6	IR_SRC_SEL	0x0	rw	Infrared modulation envelope signal source selection This field is used to select the infrared modulation envelope signal source. 00: TMR16 01: Reserved 10: Reserved 11: Reserved
Bit 5	IR_POL	0	rw	Infrared output polarity selection 0: Infrared output (IR_OUT) is not inversed. 1: Infrared output (IR_OUT) is inversed.
Bit 4	PA11_12_RMP	0x0	rw	PA11 and PA12 remap This bit is set and cleared by software. This bit controls PA9/PA10 and PA11/PA12 remap on small packages (20PIN, 28PIN).

				0: No remap (PA9/PA10 corresponds to PA9/PA10) 1: Remap (PA11/PA12 corresponds to PA9/PA10)
Bit 3:2	Reserved	0x0	resd	Kept at its reset value.
Bit 1:0	MEM_MAP_SEL	0xX	r	Memory address mapping selection This field is read-only, indicating the boot mode after reset. X0: Boot from main Flash memory 01: Boot from system memory 11: Boot from internal SRAM

7.2.2 SCFG external interrupt configuration register 1 (SCFG_EXTINC1)

Bit	Name	Reset value	Type	Description
Bit 31:16	Reserved	0x0000	resd	Kept at its default value.
Bit 15:12	EXINT3	0x0	rw	EXINT3 input source configuration These bits are used to select the input source for the EXINT3 external interrupt. 0000: GPIOA pin 3 0001: GPIOB pin 3 Others: Reserved
Bit 11:8	EXINT2	0x0	rw	EXINT2 input source configuration These bits are used to select the input source for the EXINT2 external interrupt. 0000: GPIOA pin 2 0001: GPIOB pin 2 Others: Reserved
Bit 7:4	EXINT1	0x0	rw	EXINT1 input source configuration These bits are used to select the input source for the EXINT1 external interrupt. 0000: GPIOA pin 1 0001: GPIOB pin 1 0101: GPIOF pin 1 Others: Reserved
Bit 3:0	EXINT0	0x0	rw	EXINT0 input source configuration These bits are used to select the input source for the EXINT0 external interrupt. 0000: GPIOA pin 0 0001: GPIOB pin 0 0101: GPIOF pin 0 Others: Reserved

7.2.3 SCFG external interrupt configuration register 2 (SCFG_EXINTC2)

Bit	Name	Reset value	Type	Description
Bit 31:16	Reserved	0x0000	resd	Kept at its default value.
Bit 15:12	EXINT7	0x0	rw	EXINT7 input source configuration These bits are used to select the input source for the EXINT7 external interrupt. 0000: GPIOA pin 7 0001: GPIOB pin 7 0101: GPIOF pin 7 Others: Reserved
Bit 11:8	EXINT6	0x0	rw	EXINT6 input source configuration These bits are used to select the input source for the EXINT6 external interrupt. 0000: GPIOA pin 6 0001: GPIOB pin 6 0101: GPIOF pin 6 Others: Reserved
Bit 7:4	EXINT5	0x0	rw	EXINT5 input source configuration These bits are used to select the input source for the EXINT5 external interrupt. 0000: GPIOA pin 5 0001: GPIOB pin 5 Others: Reserved
Bit 3:0	EXINT4	0x0	rw	EXINT4 input source configuration These bits are used to select the input source for the EXINT4 external interrupt.

				0000: GPIOA pin 4 0001: GPIOB pin 4 Others: Reserved

7.2.4 SCFG external interrupt configuration register 3 (SCFG_EXINTC3)

Bit	Name	Reset value	Type	Description
Bit 31:16	Reserved	0x0000	resd	Kept at its default value.
Bit 15:12	EXINT11	0x0	rw	EXINT11 input source configuration These bits are used to select the input source for the EXINT11 external interrupt. 0000: GPIOA pin 11 0001: GPIOB pin 11 Others: Reserved
Bit 11:8	EXINT10	0x0	rw	EXINT10 input source configuration These bits are used to select the input source for the EXINT10 external interrupt. 0000: GPIOA pin 10 0001: GPIOB pin 10 Others: Reserved
Bit 7:4	EXINT9	0x0	rw	EXINT9 input source configuration These bits are used to select the input source for the EXINT9 external interrupt. 0000: GPIOA pin 9 0001: GPIOB pin 9 Others: Reserved
Bit 3:0	EXINT8	0x0	rw	EXINT8 input source configuration These bits are used to select the input source for the EXINT8 external interrupt. 0000: GPIOA pin 8 0001: GPIOB pin 8 Others: Reserved

7.2.5 SCFG external interrupt configuration register 4 (SCFG_EXINTC4)

Bit	Name	Reset value	Type	Description
Bit 31:16	Reserved	0x0000	resd	Kept at its default value.
Bit 15:12	EXINT15	0x0	rw	EXINT15 input source configuration These bits are used to select the input source for the EXINT15 external interrupt. 0000: GPIOA pin 15 0001: GPIOB pin 15 0002: GPIOC pin 15 Others: Reserved
Bit 11:8	EXINT14	0x0	rw	EXINT14 input source configuration These bits are used to select the input source for the EXINT14 external interrupt. 0000: GPIOA pin 14 0001: GPIOB pin 14 0002: GPIOC pin 14 Others: Reserved
Bit 7:4	EXINT13	0x0	rw	EXINT13 input source configuration These bits are used to select the input source for the EXINT13 external interrupt. 0000: GPIOA pin 13 0001: GPIOB pin 13 0002: GPIOC pin 13 Others: Reserved
Bit 3:0	EXINT12	0x0	rw	EXINT12 input source configuration These bits are used to select the input source for the EXINT12 external interrupt. 0000: GPIOA pin 12 0001: GPIOB pin 12 Others: Reserved

7.2.6 SCFG configuration register 2 (SCFG_CFGR2)

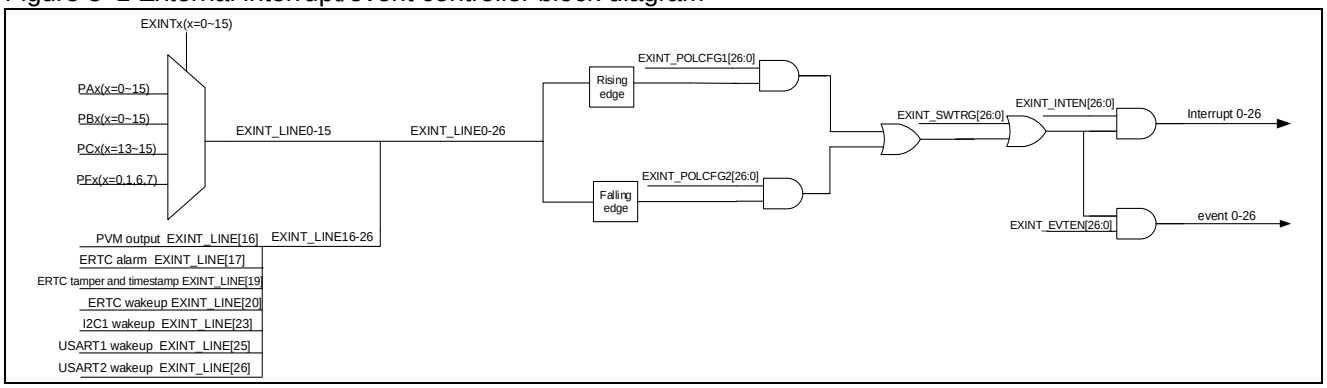
Bit	Name	Reset value	Type	Description
Bit 31:30	Reserved	0x0000 00	resd	Kept at its reset value.
Bit 29:9	Reserved	0x0000 00	resd	Kept at its reset value.
Bit 8	SRAM_OPE_STS	0	rc_w1	SRAM odd parity error status This bit is cleared by writing “1”. 0: No SRAM odd parity error occurs 1: SRAM odd parity error occurs
Bit 7:3	Reserved	0x0	resd	Kept at its reset value.
Bit 2	PVM_LK	0	rw	PVM lock enable 0: Disconnect PVM interrupt from TMR1/TMR15/16/17 brake input. PVMSEL and PVMEN bits can be changed by software. 1: Connect PVM interrupt and TMR1/TMR15/16/17 brake input. PVMSEL and PVMEN bits are read-only, unchangeable.
Bit 1	SRAM_OPERR_LK	0	rw	SRAM odd parity error lock enable 0: Disconnect SRAM odd parity error from TMR1/TMR15/16/17 brake input. 1: Connect SRAM odd parity error and TMR1/TMR15/16/17 brake input.
Bit 0	CPU_LK	0	rw	CPU lock enable 0: Disconnect CPU lock from TMR1/TMR15/16/17 brake input. 1: Connect CPU lock and TMR1/TMR15/16/17 brake input.

8 External interrupt/event controller (EXINT)

8.1 Introduction

EXINT consists of 23 interrupt lines EXINT_LINE[26:0] (bit 18, bit 21, bit 22 and bit 24 are reserved), each of which can generate an interrupt or event by edge detection trigger or software trigger. EXINT can enable or disable an interrupt or event independently through software configuration, and utilizes different edge detection modes (rising edge, falling edge or both edges) and trigger modes (edge detection, software trigger or both triggers) to respond to the trigger source in order to generate an interrupt or event.

Figure 8-1 External interrupt/event controller block diagram



Main features:

- EXINT 0~15 mapping IO can be configured independently
- Independent trigger selection on each interrupt line
- Independent enable bit on each interrupt
- Independent enable bit on each event
- Up to 23 software triggers that can be generated and cleared independently
- Independent status bit on each interrupt
- Each interrupt can be cleared independently

8.2 Function overview and configuration procedure

With up to 23 interrupt lines EXINT_LINE[26:0] (bit 18, bit 21, bit 22 and bit 24 are reserved), EXINT can detect not only GPIO external interrupt sources but also seven internal interrupt sources including PVM output, ERTC alarm, ERTC wakeup, ERTC tamper and time stamp events, USART1 wakeup event, USART2 wakeup event and I2C1 wakeup event by the means of edge detection. The GPIO interrupt sources can be selected with SCFG_EXINTCx register. It should be noted that these input sources are mutually exclusive. For example, EXINT_LINE0 is allowed to select only one of PA0/PB0/PF0 pins, instead of taking both PA0 and PB0 as the input sources at the same time.

EXINT supports multiple edge detection modes, including rising edge, falling edge or both edges, selected by EXINT_POLCFG1 and EXINT_POLCFG2 registers. Active edge detection on an interrupt line can generate an event or interrupt.

In addition, EXINT supports independent software trigger for the generation of an interrupt or event. This is achieved by setting the corresponding bits in the EXINT_SWTRG register.

EXINT has independent interrupt and event enable bits. EXINT can enable or disable an interrupt or event independently through software configuration such as EXINT_INTEN and EXINT_EVTEN registers, indicating that the corresponding interrupt or event control bit must be enabled in advance.

EXINT also features an independent interrupt status bit. Reading access to EXINT_INTSTS register can obtain the corresponding interrupt status. The status flag is cleared by writing “1” to this register.

Interrupt initialization procedure

1. Select an interrupt source by setting SCFG_EXINTCx register (This is required if GPIO is used as an interrupt source);
2. Select a trigger mode by setting EXINT_POLCFG1 and EXINT_POLCFG2 registers;
3. Enable interrupt or event by setting EXINT_INTEN and EXINT_EVTEN registers;
4. Generate software trigger by setting EXINT_SWTRG register (This is applied to only software trigger interrupt).

Note: To modify the interrupt source configuration, disable the EXINT_INTEN and EXINT_EVTEN registers before re-start the interrupt initialization procedure.

Interrupt clear procedure

- Writing “1” to the EXINT_INTSTS register to clear the generated interrupts and the corresponding bits in the EXINT_SWTRG register.

8.3 EXINT registers

These peripheral registers must be accessed by words (32 bits).
The table below shows the EXINT register map and their reset values.

Table 8-1 External interrupt/event controller register map and reset value

Register	Offset	Reset value
EXINT_INTEN	0x00	0x0000 0000
EXINT_EVTEN	0x04	0x0000 0000
EXINT_ POLCFG1	0x08	0x0000 0000
EXINT_ POLCFG2	0x0C	0x0000 0000
EXINT_ SWTRG	0x10	0x0000 0000
EXINT_ INTSTS	0x14	0x0000 0000

8.3.1 Interrupt enable register (EXINT_INTEN)

Bit	Name	Reset value	Type	Description
Bit 31:29	Reserved	0x000	resd	Forced to be 0 by hardware.
Bit 28:0	INTENx	0x00000	rw	Interrupt enable or disable on line x 0: Interrupt request is disabled 1: Interrupt request is enabled Note: Bit 18, bit 21, bit 22, bit 24, bit 27 and bit 28 are reserved.

8.3.2 Event enable register (EXINT_EVTEN)

Bit	Name	Reset value	Type	Description
Bit 31:29	Reserved	0x000	resd	Forced to be 0 by hardware.
Bit 28:0	EVTENx	0x00000	rw	Event enable or disable on line x 0: Event request is disabled 1: Event request is enabled Note: Bit 18, bit 21, bit 22, bit 24, bit 27 and bit 28 are reserved.

8.3.3 Polarity configuration register 1 (EXINT_POLCFG1)

Bit	Name	Reset value	Type	Description
Bit 31:29	Reserved	0x000	resd	Forced to be 0 by hardware.
Bit 28:0	RPx	0x00000	rw	Rising polarity configuration bit of line x These bits are used to select a rising edge to trigger an interrupt or event on line x. 0: Rising trigger on line x is disabled 1: Rising trigger on line x is enabled Note: Bit 18, bit 21, bit 22, bit 24, bit 27 and bit 28 are reserved.

8.3.4 Polarity configuration register 2 (EXINT_POLCFG2)

Bit	Name	Reset value	Type	Description
Bit 31:29	Reserved	0x000	resd	Forced to be 0 by hardware.
Bit 28:0	FPx	0x00000	rw	Falling polarity event configuration bit of line x These bits are used to select a falling edge to trigger an interrupt or event on line x. 0: Falling trigger on line x is disabled 1: Falling trigger on line x is enabled Note: Bit 18, bit 21, bit 22, bit 24, bit 27 and bit 28 are reserved.

8.3.5 Software trigger register (EXINT_SWTRG)

Bit	Name	Reset value	Type	Description
Bit 31:29	Reserved	0x000	resd	Forced to be 0 by hardware.
Bit 28:0	SWTx	0x00000	rw	Software trigger on line x When the corresponding bit in EXINT_INTEN register is 1, if the software writes to this bit, the hardware sets the corresponding bit in the EXINT_INTSTS register automatically to generate an interrupt. When the corresponding bit in EXINT_EVTEN register is 1, if the software writes to this bit, the hardware generates an event on the corresponding interrupt line automatically. 0: Default value 1: Software trigger generated Note: This bit is cleared by writing “1” to the corresponding bit in the EXINT_INTSTS register. Note: Bit 18, bit 21, bit 22, bit 24, bit 27 and bit 28 are reserved.

8.3.6 Interrupt status register (EXINT_INTSTS)

Bit	Name	Reset value	Type	Description
Bit 31:29	Reserved	0x000	resd	Forced to be 0 by hardware.
Bit 28:0	LINEx	0x00000	rw1c	Line x state bit 0: No interrupt occurred 1: Interrupt occurred Note: This bit is cleared by writing “1” to the corresponding bit in this register. Note: Bit 18, bit 21, bit 22, bit 24, bit 27 and bit 28 are reserved.

9 DMA controller (DMA)

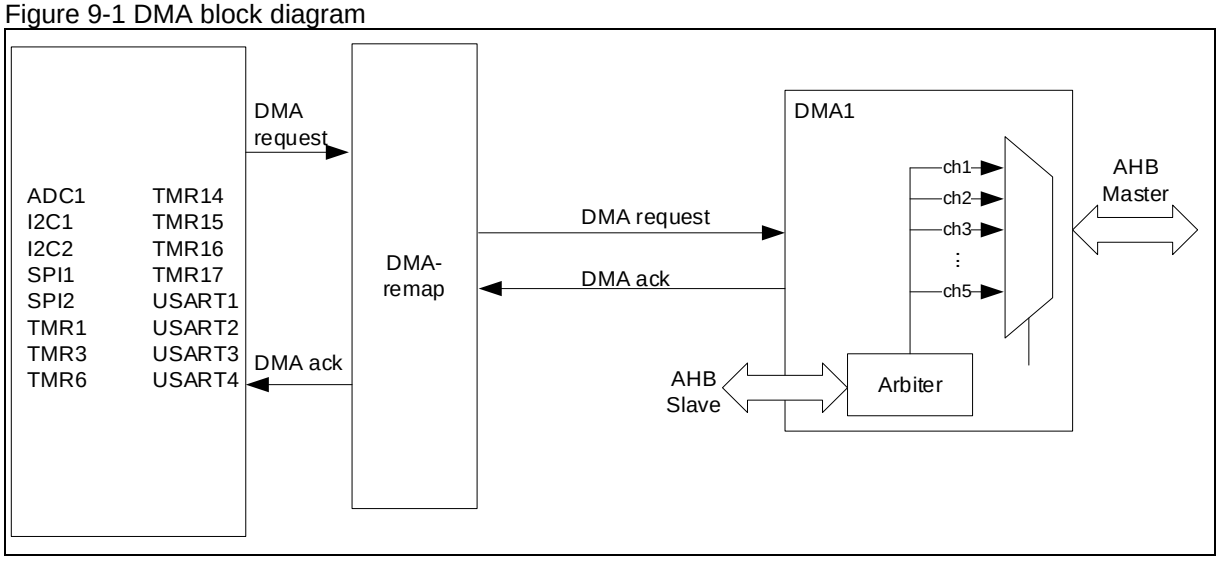
9.1 Introduction

Direct memory access (DMA) controller is designed for 32-bit MCU applications with the aim of enhancing system performance and reducing the generation of interrupts.
One DMA controller is available in the microcontroller. Each controller contains 5 DMA channels. Each channel manages memory access requests from one or more peripherals. An arbiter is available for coordinating the priority of each DMA request.

9.2 Main features

- AMBA compliant (Rev. 2.0)
- Only support AHB OKAY and ERROR
- HBUSREQ and HGRANT of AHB master interface are not supported
- Support 5 channels
- Peripheral-to-memory, memory-to-peripheral, and memory-to-memory transfers
- Support hardware handshake
- Support 8-bit, 16-bit and 32-bit data transfers
- Programmable amount of data to be transferred: up to 65535
- Support flexible mapping

Figure 9-1 DMA block diagram



Note: The number of DMA peripherals in the figure may decrease depending on different models.

9.3 Function overview

9.3.1 DMA configuration

1. **Set the peripheral address in the DMA_CxPADDR register**
   The initial peripheral address for data transfer remains unchanged during transmission.
2. **Set the memory address in the DMA_CxMADDR register**
   The initial memory address for data transfer remains unchanged during transmission.
3. **Configure the amount of data to be transferred in the DMA_CxDTCNT register**
   programmable data transfer size is up to 65535. The value is decremented after each data transfer.
4. **Configure the channel setting in the DMA_CxCTRL register**
   Including channel priority, data transfer direction/width, address incremented mode, circular mode and interrupt mode.

- **Channel priority (CHPL)**

There are four levels, including the very high priority, high priority, medium priority and low priority.
If the two channels have the same priority level, the then channel with lower number will get priority over the one with higher number. For example, channel 1 has priority over channel 2.

- **Data transfer direction (DTD)**

Memory-to-peripheral (M2P), peripheral-to-memory (P2M)

- **Address incremented mode (PINCM/MINCM)**

In incremented mode (PWIDTH/MWIDTH), the subsequent transfer address is the previous address plus transfer width.

- **Circular mode (LM)**

In circular mode, the contents in the DMA_CxDTCNT register is automatically reloaded with the initially programmed value after the completion of the last transfer.

- **Memory-to-memory mode (M2M)**

This mode indicates that DMA channels perform data transfer without requests from peripherals. Circular mode and memory-to-memory mode cannot be used at the same time.

5. **In non-M2M mode, configure flexible mapping mode through the DMA_SRC_SEL0/1 register**
   Write DMA request ID into the corresponding bit of relevant channel.
6. **Enable DMA transfer by setting CHEN bit in the DMA_CxCTRL register**

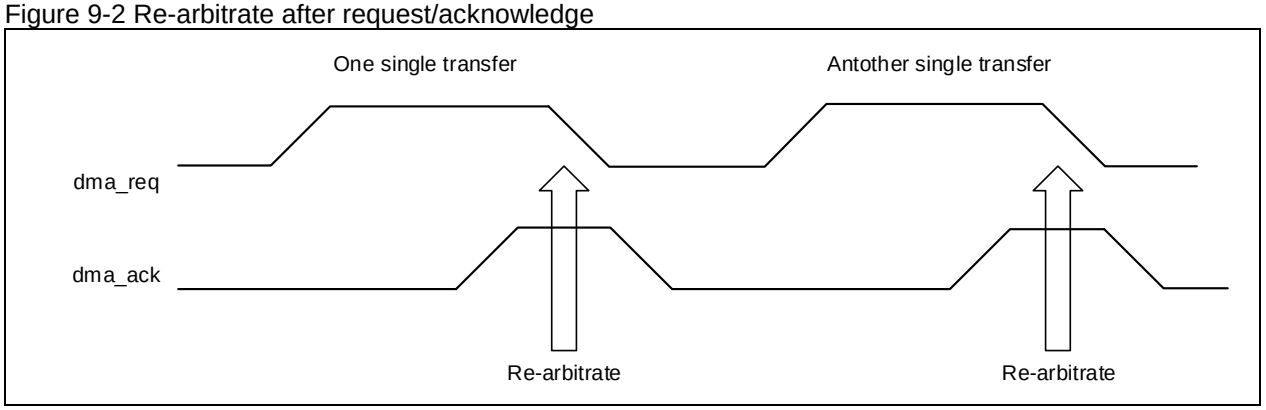
9.3.2 Handshake mechanism

In P2M and M2P modes, the peripherals need to send a request signal to the DMA controller. The DMA channel will send the peripheral transfer request (single) until the signal is acknowledged. After the completion of peripheral transmission, the DMA controller sends the acknowledge signal to the peripheral. The peripheral then releases its request as soon as it receives the acknowledge signal. If the peripheral cancels the request, the DMA controller will release the acknowledge signal.

9.3.3 Arbiter

When several channels are enabled simultaneously, the arbiter will restart arbitration after full data transfer by the master controller. The channel with very high priority waits until the channel of the master controller has completed data transfers before taking control of it. The master controller will re-arbitrate to serve other channels as long as the channel completes a single transfer based on the master controller priority.

Figure 9-2 Re-arbitrate after request/acknowledge



9.3.4 Programmable data transfer width

Transfer width of the source data and destination data is programmable through the PWIDTH and MWIDTH bits in the DMA_CxCTRL register. When PWIDTH is not equal to MWIDTH, it can be aligned according to the settings of PWIDTH/ MWIDTH.

Figure 9-3 PWIDTH: byte, MWIDTH: half-word

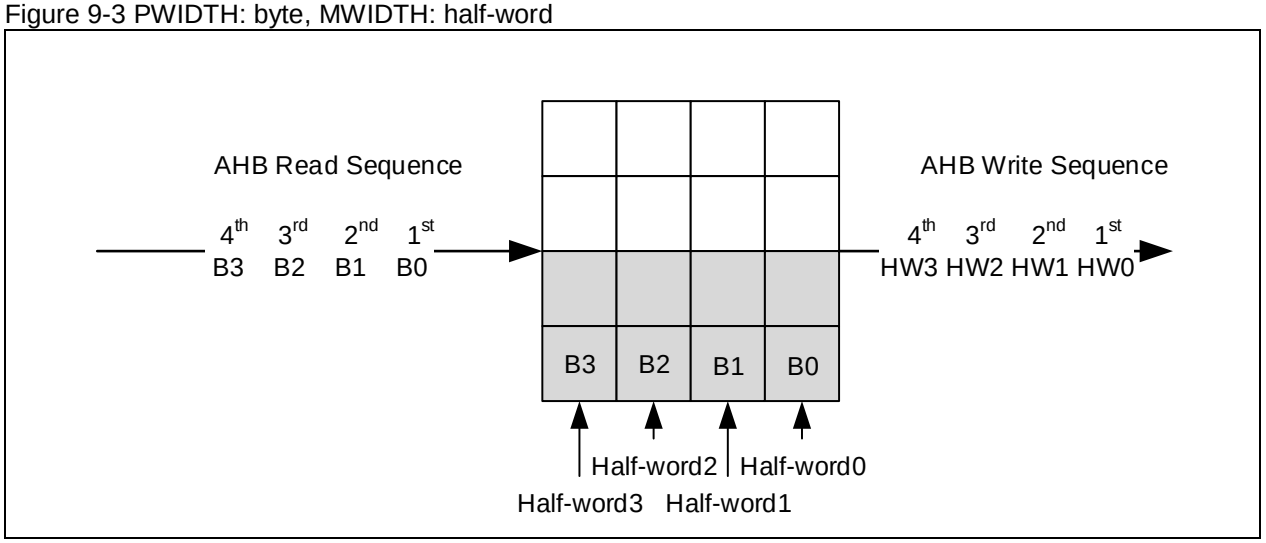


Figure 9-4 PWIDTH: half-word, MWIDTH: word

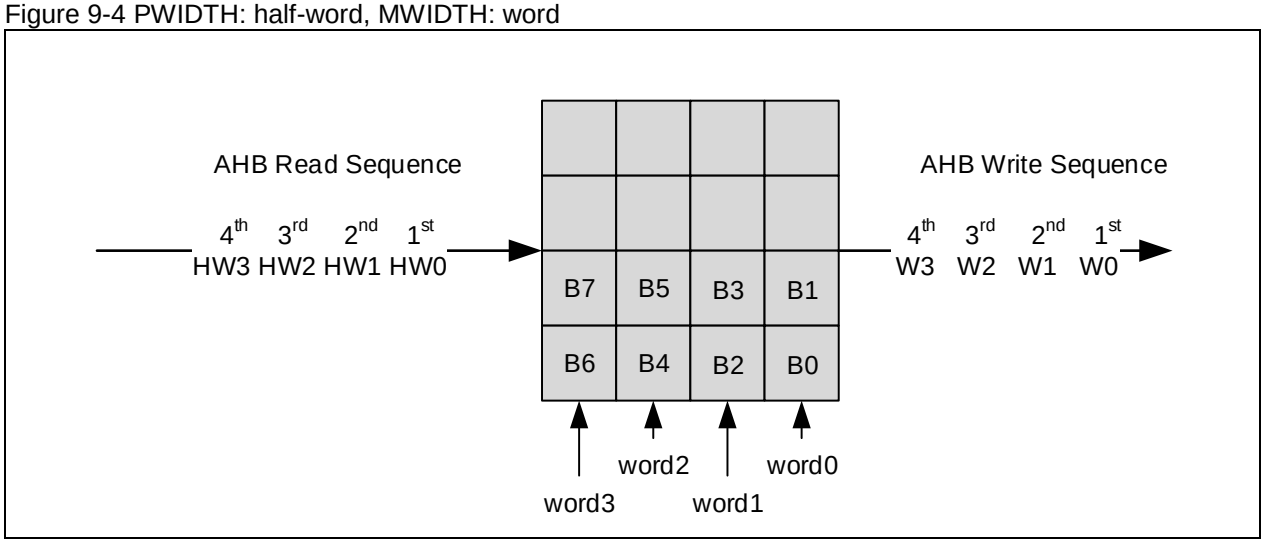
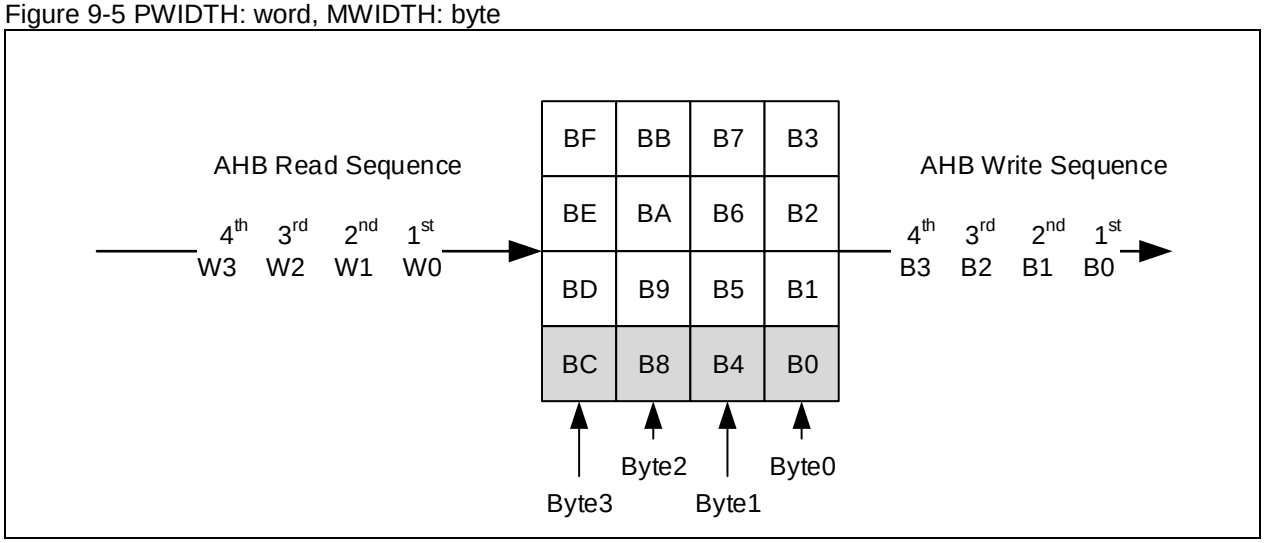


Figure 9-5 PWIDTH: word, MWIDTH: byte



9.3.5 Errors

Table 9-1 DMA error event

Error event	
Transfer error	AHB response error occurred during DMA read/write access

9.3.6 Interrupts

An interrupt can be generated on a DMA half-transfer, transfer complete and transfer error. Each channel has its specific interrupt flag, clear and enable bits, as shown in the table below.

Table 9-2 DMA interrupts

Interrupt event	Event flag bit	Clear control bit	Enable control bit
Half transfer	HDTF	HDTFC	HDTIEN
Transfer complete	FDTF	FDTFC	FDTIEN
Transfer error	DTERRF	DTERRFC	DTERRIEN

9.3.7 Flexible DMA request mapping

In flexible request mode (DMA_FLEX_EN = 1), the request source for each channel is set through the CHx_SRC register [x=1~5].

For example, to configure the DMA channel 1 to I2C1_TX and channel 3 to I2C1_RX, others unused, then DMA_FLEX_EN=1, CH1_SRC=11, CH3_SRC=10, CH[2/4/5]_SRC=0 must be asserted.

The table below lists the DMA flexible request sources.

Table 9-3 DMA flexible request sources

CHx_SRC	Request source	CHx_SRC	Request source	CHx_SRC	Request source	CHx_SRC	Request source
0	Unselected	16	SPI1/I2S1_RX	32	TMR3_CH1	48	-
1	-	17	SPI1/I2S1_TX	33	TMR3_CH2	49	TMR17_ OVERFLOW
2	-	18	SPI2/I2S2_RX	34	TMR3_CH3	50	USART1_RX
3	-	19	SPI2/I2S2_TX	35	TMR3_CH4	51	USART1_TX
4	-	20	TMR1_CH1	36	TMR3_TRIG	52	USART2_RX
5	ADC1	21	TMR1_CH2	37	TMR3_ OVERFLOW	53	USART2_TX
6	-	22	TMR1_CH3	38	TMR6_ OVERFLOW	54	USART3_RX
7	-	23	TMR1_CH4	39	-	55	USART3_TX
8	-	24	TMR1_TRIG/ TMR1_HALL	40	TMR15_CH1	56	USART4_RX
9	-	25	TMR1_ OVERFLOW	41	TMR15_CH2	57	USART4_TX
10	I2C1_RX	26	-	42	TMR15_TRIG/ TMR15_HALL	58	-
11	I2C1_TX	27	-	43	TMR15_ OVERFLOW	59	-
12	I2C2_RX	28	-	44	TMR16_CH1	60	-
13	I2C2_TX	29	-	45	-	61	-
14	-	30	-	46	TMR16_ OVERFLOW		-
15	-	31	-	47	TMR17_CH1		-

9.4 DMA registers

The table below shows DMA register map and their reset values.
These peripheral registers can be accessed by byte (8 bits), half-word (16 bits) or word (32 bits).

Table 9-4 DMA register map and reset values

Register	Offset	Reset value
DMA_STS	0x00	0x0000 0000
DMA_CLR	0x04	0x0000 0000
DMA_C1CTRL	0x08	0x0000 0000
DMA_C1DTCNT	0x0C	0x0000 0000
DMA_C1PADDR	0x10	0x0000 0000
DMA_C1MADDR	0x14	0x0000 0000
DMA_C2CTRL	0x1C	0x0000 0000
DMA_C2DTCNT	0x20	0x0000 0000
DMA_C2PADDR	0x24	0x0000 0000
DMA_C2MADDR	0x28	0x0000 0000
DMA_C3CTRL	0x30	0x0000 0000
DMA_C3DTCNT	0x34	0x0000 0000
DMA_C3PADDR	0x38	0x0000 0000
DMA_C3MADDR	0x3C	0x0000 0000
DMA_C4CTRL	0x44	0x0000 0000
DMA_C4DTCNT	0x48	0x0000 0000
DMA_C4PADDR	0x4C	0x0000 0000
DMA_C4MADDR	0x50	0x0000 0000
DMA_C5CTRL	0x58	0x0000 0000
DMA_C5DTCNT	0x5C	0x0000 0000
DMA_C5PADDR	0x60	0x0000 0000
DMA_C5MADDR	0x64	0x0000 0000
DMA_SRC_SEL0	0xA0	0x0000 0000
DMA_SRC_SEL1	0xA4	0x0000 0000

9.4.1 DMA interrupt status register (DMA_STS)

Access: 0 wait state, accessible by bytes, half-words or words.

Bit	Name	Reset value	Type	Description
Bit 31:20	Reserved	0x0	resd	Kept at its default value.
Bit 19	DTERRF5	0x0	ro	Channel 5 data transfer error event flag 0: No transfer error occurred 1: Transfer error occurred
Bit 18	HDTF5	0x0	ro	Channel 5 half transfer event flag 0: No half-transfer event occurred 1: Half-transfer event occurred
Bit 17	FDTF5	0x0	ro	Channel 5 transfer complete event flag 0: No transfer complete event occurred 1: Transfer complete event occurred
Bit 16	GF5	0x0	ro	Channel 5 global event flag 0: No transfer error, half transfer or transfer complete event occurred 1: Transfer error, half transfer or transfer complete event occurred

Bit 15	DTERRF4	0x0	ro	Channel 4 data transfer error event flag 0: No transfer error occurred 1: Transfer error occurred
Bit 14	HDTF4	0x0	ro	Channel 4 half transfer event flag 0: No half-transfer event occurred 1: Half-transfer event occurred
Bit 13	FDTF4	0x0	ro	Channel 4 transfer complete event flag 0: No transfer complete event occurred 1: Transfer complete event occurred
Bit 12	GF4	0x0	ro	Channel 4 global event flag 0: No transfer error, half transfer or transfer complete event occurred 1: Transfer error, half transfer or transfer complete event occurred
Bit 11	DTERRF3	0x0	ro	Channel 3 data transfer error event flag 0: No transfer error occurred 1: Transfer error occurred
Bit 10	HDTF3	0x0	ro	Channel 3 half transfer event flag 0: No half-transfer event occurred 1: Half-transfer event occurred
Bit 9	FDTF3	0x0	ro	Channel 3 transfer complete event flag 0: No transfer complete event occurred 1: Transfer complete event occurred
Bit 8	GF3	0x0	ro	Channel 3 global event flag 0: No transfer error, half transfer or transfer complete event occurred 1: Transfer error, half transfer or transfer complete event occurred
Bit 7	DTERRF2	0x0	ro	Channel 2 data transfer error event flag 0: No transfer error occurred 1: Transfer error occurred
Bit 6	HDTF2	0x0	ro	Channel 2 half transfer event flag 0: No half-transfer event occurred 1: Half-transfer event occurred
Bit 5	FDTF2	0x0	ro	Channel 2 transfer complete event flag 0: No transfer complete event occurred 1: Transfer complete event occurred
Bit 4	GF2	0x0	ro	Channel 2 global event flag 0: No transfer error, half transfer or transfer complete event occurred 1: Transfer error, half transfer or transfer complete event occurred
Bit 3	DTERRF1	0x0	ro	Channel 1 data transfer error event flag 0: No transfer error occurred 1: Transfer error occurred
Bit 2	HDTF1	0x0	ro	Channel 1 half transfer event flag 0: No half-transfer event occurred 1: Half-transfer event occurred
Bit 1	FDTF1	0x0	ro	Channel 1 transfer complete event flag 0: No transfer complete event occurred 1: Transfer complete event occurred
Bit 0	GF1	0x0	ro	Channel 1 global event flag 0: No transfer error, half transfer or transfer complete event occurred 1: Transfer error, half transfer or transfer complete event occurred

9.4.2 DMA interrupt flag clear register (DMA_CLR)

Access: 0 wait state, accessible by bytes, half-words or words.

Bit	Name	Reset value	Type	Description
Bit 31:20	Reserved	0x0	resd	Kept at its default value.
Bit 19	DTERRFC5	0x0	rw1c	Channel 5 data transfer error flag clear 0: No effect 1 Clear the DTERRF5 flag in the DMA_STS register
Bit 18	HDTFC5	0x0	rw1c	Channel 5 half data transfer flag clear 0: No effect

				1: Clear the HDTF5 flag in the DMA_STS register
Bit 17	FDTFC5	0x0	rw1c	Channel 5 transfer complete flag clear 0: No effect 1: Clear the FDTF5 flag in the DMA_STS register
Bit 16	GFC5	0x0	rw1c	Channel 5 global interrupt flag clear 0: No effect 1: Clear DTERRF5, HDTF5, FDTF5 and GF5 in the DMA_STS register
Bit 15	DTERRFC4	0x0	rw1c	Channel 4 data transfer error flag clear 0: No effect 1: Clear the DTERRF4 flag in the DMA_STS register
Bit 14	HDTFC4	0x0	rw1c	Channel 4 half data transfer flag clear 0: No effect 1: Clear the HDTF4 flag in the DMA_STS register
Bit 13	FDTFC4	0x0	rw1c	Channel 4 transfer complete flag clear 0: No effect 1: Clear the FDTF4 flag in the DMA_STS register
Bit 12	GFC4	0x0	rw1c	Channel 4 global interrupt flag clear 0: No effect 1: Clear DTERRF4, HDTF4, FDTF4 and GF4 in the DMA_STS register
Bit 11	DTERRFC3	0x0	rw1c	Channel 3 data transfer error flag clear 0: No effect 1: Clear the DTERRF7 flag in the DMA_STS register
Bit 10	HDTFC3	0x0	rw1c	Channel 3 half data transfer flag clear 0: No effect 1: Clear the HDTF7 flag in the DMA_STS register
Bit 9	FDTFC3	0x0	rw1c	Channel 3 transfer complete flag clear 0: No effect 1: Clear the FDTF3 flag in the DMA_STS register
Bit 8	GFC3	0x0	rw1c	Channel 3 global interrupt flag clear 0: No effect 1: Clear DTERRF3, HDTF3, FDTF3 and GF3 in the DMA_STS register
Bit 7	DTERRFC2	0x0	rw1c	Channel 2 data transfer error flag clear 0: No effect 1: Clear the DTERRF2 flag in the DMA_STS register
Bit 6	HDTFC2	0x0	rw1c	Channel 2 half data transfer flag clear 0: No effect 1: Clear the HDTF2 flag in the DMA_STS register
Bit 5	FDTFC2	0x0	rw1c	Channel 2 transfer complete flag clear 0: No effect 1: Clear the FDTF2 flag in the DMA_STS register
Bit 4	GFC2	0x0	rw1c	Channel 2 global interrupt flag clear 0: No effect 1: Clear DTERRF2, HDTF2, FDTF2 and GF2 in the DMA_STS register
Bit 3	DTERRFC1	0x0	rw1c	Channel 1 data transfer error flag clear 0: No effect 1: Clear the DTERRF1 flag in the DMA_STS register
Bit 2	HDTFC1	0x0	rw1c	Channel 1 half data transfer flag clear 0: No effect 1: Clear the HDTF1 DMA_STS register
Bit 1	FDTFC1	0x0	rw1c	Channel 1 transfer complete flag clear 0: No effect 1: Clear the FDTF1 flag in the DMA_STS register
Bit 0	GFC1	0x0	rw1c	Channel 1 global interrupt flag clear 0: No effect 1: Clear DTERRF1, HDTF1, FDTF1 and GF1DMA_STS in the register

9.4.3 DMA channel-x configuration register (DMA_CxCTRL) (x = 1…5)

Access: 0 wait state, accessible by bytes, half-words or words.

Bit	Name	Reset value	Type	Description
Bit 31:15	Reserved	0x00000	resd	Kept at its default value.
Bit 14	M2M	0x0	rw	Memory to memory mode 0: Disabled 1: Enabled
Bit 13:12	CHPL	0x0	rw	Channel priority level 00: Low 01: Medium 10: High 11: Very high
Bit 11:10	MWIDTH	0x0	rw	Memory data bit width 00: 8 bits 01: 16 bits 10: 32 bits 11: Reserved
Bit 9:8	PWIDTH	0x0	rw	Peripheral data bit width 00: 8 bits 01: 16 bits 10: 32 bits 11: Reserved
Bit 7	MINCM	0x0	rw	Memory address increment mode 0: Disabled 1: Enabled
Bit 6	PINCM	0x0	rw	Peripheral address increment mode 0: Disabled 1: Enabled
Bit 5	LM	0x0	rw	Loop mode 0: Disabled 1: Enabled
Bit 4	DTD	0x0	rw	Data transfer direction 0: Read from peripherals 1: Read from memory
Bit 3	DTERRIEN	0x0	rw	Data transfer error interrupt enable 0: Disabled 1: Enabled
Bit 2	HDTIEN	0x0	rw	Half-transfer interrupt enable 0: Disabled 1: Enabled
Bit 1	FDTIEN	0x0	rw	Transfer complete interrupt enable 0: Disabled 1: Enabled
Bit 0	CHEN	0x0	rw	Channel enable 0: Disabled 1: Enabled

9.4.4 DMA channel-x number of data register (DMA_CxDTCNT) (x= 1…5)

Access: 0 wait state, accessible by bytes, half-words or words.

Bit	Name	Reset value	Type	Description
Bit 31:16	Reserved	0x0000	resd	Kept at its default value.
Bit 15:0	CNT	0x0000	rw	Number of data to transfer The number of data to transfer is from 0x0 to 0xFFFF. This register can only be written when the CHEN bit in the corresponding channel is set to 0. The value is decremented by 1 after each DMA transfer. Note: This register holds the number of data to transfer, instead of transfer size. The transfer size is calculated by data width.

9.4.5 DMA channel-x peripheral address register (DMA_CxPADDR) (x = 1…5)

Access: 0 wait state, accessible by bytes, half-words or words.

Bit	Name	Reset value	Type	Description
Bit 31:0	PADDR	0x0000 0000	rw	Peripheral base address Base address of peripheral data register is the source or destination of data transfer. Note: The register can only be written when the CHEN bit in the corresponding channel is set to 0.

9.4.6 DMA channel-x memory address register (DMA_CxMADDR) (x = 1…5)

Access: 0 wait state, accessible by bytes, half-words or words.

Bit	Name	Reset value	Type	Description
Bit 31:0	MADDR	0x0000 0000	rw	Memory base address Memory address is the source or destination of data transfer. Note: The register can only be written when the CHEN bit is set to 0.

9.4.7 DMA channel source register 0 (DMA_SRC_SEL0)

Access: 0 wait state, accessible by bytes, half-words or words.

Bit	Name	Reset value	Type	Description
Bit 31:24	CH4_SRC	0x00	rw	CH4 source select When DMA_FLEX_EN=1, channel 4 is selected by the CH4_SRC. Refer to 9.3.7 for more information.
Bit 23:16	CH3_SRC	0x00	rw	CH3 source select When DMA_FLEX_EN=1, channel 3 is selected by the CH3_SRC. Refer to 9.3.7 for more information.
Bit 15:8	CH2_SRC	0x00	rw	CH2 source select When DMA_FLEX_EN=1, channel 2 is selected by the CH2_SRC. Refer to 9.3.7 for more information.
Bit 7:0	CH1_SRC	0x00	rw	CH1 source select When DMA_FLEX_EN=1, channel 1 is selected by the CH1_SRC. Refer to 9.3.7 for more information.

9.4.8 DMA channel source register 1 (DMA_SRC_SEL1)

Access: 0 wait state, accessible by bytes, half-words or words.

Bit	Name	Reset value	Type	Description
Bit 31:25	Reserved	0x00	resd	Kept at its default value.
Bit 24	DMA_FLEX_EN	0x0	rw	DMA flexible mapping select enable 0: Fixed mapping mode 1: Flexible mapping mode
Bit 23:8	Reserved	0x00	resd	Kept at its default value.
Bit 7:0	CH5_SRC	0x00	rw	CH5 source select When DMA_FLEX_EN=1, channel 5 is selected by the CH5_SRC. Refer to 9.3.7 for more information.

10 CRC calculation unit (CRC)

10.1 Introduction

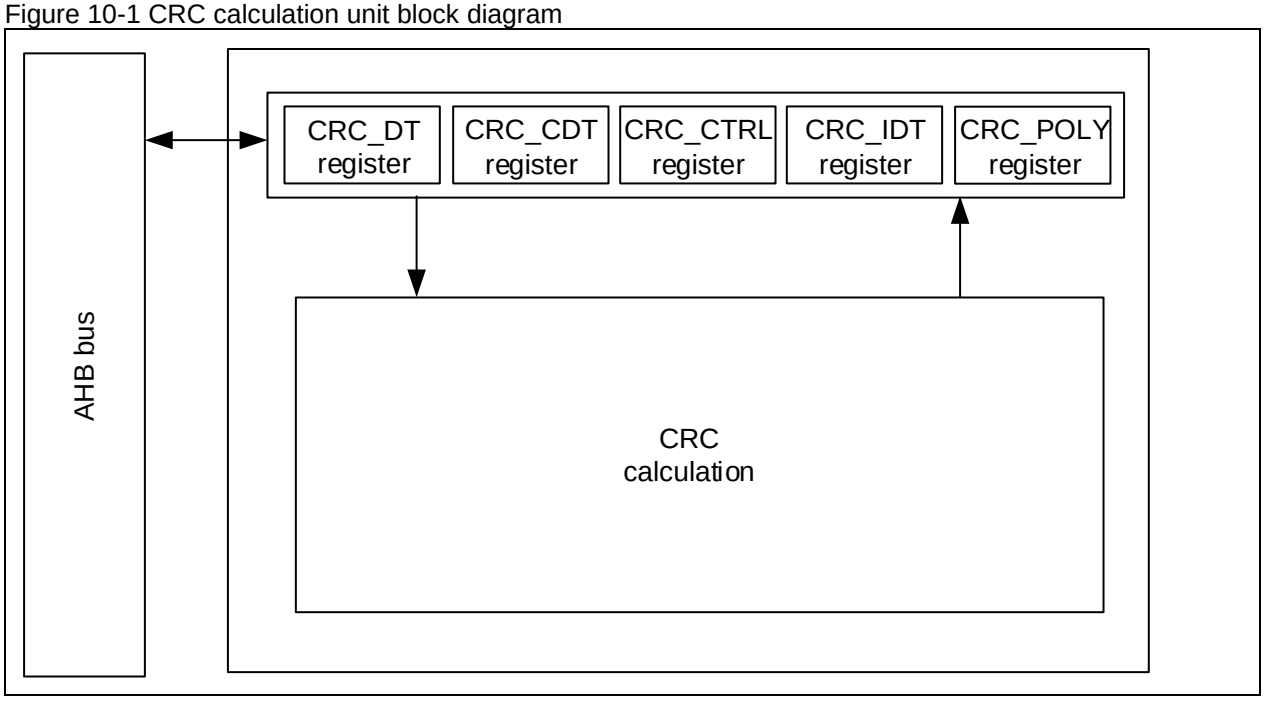
The Cyclic Redundancy Check (CRC) is an independent peripheral CRC check feature. It follows CRC32/MPEG-2 standard.

The CRC_CTRL register is used to select output data toggle (word, REVOD=1) or input data toggle (byte: REVID=01; half-word: REVID=10; word: REVID=11). CRC calculation unit is also designed with initialization function. After each reset, the value in the CRC_IDT register is written into the data register (CRC_DT) by CRC.

The CRC_POLY register is used to set different polynomial coefficients. The polynomial size can be set as 7 bits, 8 bits, 16 bits or 32 bits through the POLY_SIZE bit in the CRC_CTRL register.

Users can write the data to go through CRC check and read the calculated result through CRC_DT register. Note that the calculation result is the combination of the previous result and the current value to be calculated.

Figure 10-1 CRC calculation unit block diagram



Main features:

- Use CRC-32 code
- Support generation of polynomial
- 4 HCLK cycles for each CRC calculation
- Support input/output data format toggle
- Perform write/read operation through CRC_DT register
- Set an initialization value with the CRC_IDT register. The value is loaded into CRC_DT register after each CRC reset.

10.2 CRC functional description

According to CRC calculation principle: the input data is taken as dividend, and the generator polynomial as a division. Using mod 2 division logic, the input data divided by the generator polynomial gets a remainder, that is, the CRC value.

CRC calculation procedure

- Input data reverse. After data input, reverse input data depending on the REVID value in the CRC_CTRL register;
- Initialization. The first data input needs to be XOR-ed with the initial value defined in the CRC_IDT register. If it is not the first data input, the initial value is the previously calculated result.

- CRC calculation. Dividing the input data by the generator polynomial (0x4C11DB7) using mod 2 division method produces a remainder, that is, CRC value.
- Output data toggle. Select whether to perform word toggle before output CRC value through the REVOD bit in the CRC_CTRL register.
- XOR calculation. The XOR-ed result is fixed at 0x0000 0000.

CRC-32/MPEG-2 parameters

- Generator polynomial: 0x4C11DB7,

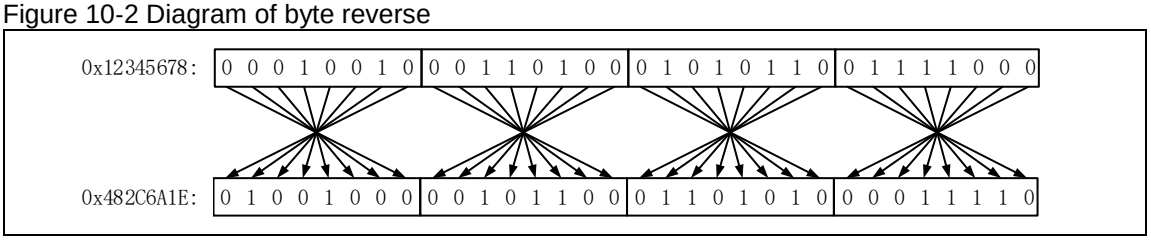
that is, $X^{32}+X^{26}+X^{23}+X^{22}+X^{16}+X^{12}+X^{11}+X^{10}+X^8+X^7+X^5+X^4+X^2+X+1$

- Initial value: 0xFFFF FFFF, in order to avoid that 1-byte 0x00 data to be calculated has the same result as that of multiple-byte 0x00.
- XOR-ed value: 0x0000 0000, indicating that the CRC result will not be XOR-ed.

Toggle function

- Byte reverse, 8 bits in a group, and sequence is reversed within a group. As shown in the figure below, if the original data is 0x12345678, it is reversed as 0x482C6A1E.
- Half-word reverse, 16 bits in a group, and sequence is reversed within a group.
- Word reverse, 32 bits in a group, and sequence is reversed within a group.

Figure 10-2 Diagram of byte reverse



10.3 CRC registers

CRC_DT register can be accessed by bytes (8 bits), half-words (16 bits) or words (32 bits). Other registers have to be accessed by words (32 bits).

Table 10-1 CRC register map and reset values

Register	Offset	Reset value
CRC_DT	0x00	0xFFFF FFFF
CRC_CDT	0x04	0x0000 0000
CRC_CTRL	0x08	0x0000 0000
CRC_IDT	0x10	0xFFFF FFFF
CRC_POLY	0x14	0x04C1 1DB7

10.3.1 Data register (CRC_DT)

Bit	Name	Reset value	Type	Description
Bit 31:0	DT	0xFFFF FFFF	rw	Data value It is used as input register when writing new data into the CRC calculator. It returns CRC calculation results when it is read.

10.3.2 Common data register (CRC_CDT)

Bit	Name	Reset value	Type	Description
Bit 31:8	Reserved	0x000000	resd	Kept at its default value.
Bit 7:0	CDT	0x00	rw	Common 8-bit data value This field is used to store 1-byte data temporarily. This register is not affected by the CRC reset generated by the RST bit in the CRC_CTRL register.

10.3.3 Control register (CRC_CTRL)

Bit	Name	Reset value	Type	Description
Bit 31:8	Reserved	0x000000	resd	Kept at its default value.
Bit 7	REVOD	0	resd	Reverse output data This bit is set or cleared by software. It is used to control whether or not to reverse output data. 0: No effect 1: Word reverse
Bit 6:5	REVID	0x0	rw	Reverse input data This bit is set or cleared by software. It is used to control how to reverse input data. 00: No effect 01: Byte reverse 10: Half-word reverse 11: Word reverse
Bit 4:3	POLY_SIZE	0x0	rw	Polynomial size This field is used to set the size of polynomial. It is used in conjunction with the CRC_POLY register. 00: 32 bits 01: 16 bits 10: 8 bits 11: 7 bits
Bit 2:1	Reserved	0x0	resd	Kept at its default value.
Bit 0	RST	0	rw	Reset CRC calculation unit This bit is set by software and cleared by hardware automatically. To reset CRC calculation unit, the data register is set to 0xFFFF FFFF. 0: No effect 1: Reset

10.3.4 Initialization register (CRC_IDT)

Bit	Name	Reset value	Type	Description
Bit 31:0	IDT	0xFFFF FFFF	rw	Initialization data register When CRC reset is triggered by the RST bit in the CRC_CTRL register, the value in the initialization register is written into the CRC_DT register as an initial value.

10.3.5 Polynomial register (CRC_POLY)

Bit	Name	Reset value	Type	Description
Bit 31:0	POLY	0x04C1 1DB7	rw	Polynomial coefficient The generated polynomial is a divisor in CRC calculation. Using CRC32 mode, this polynomial coefficient is 0x4C11DB7. Users can also set the polynomial coefficient according to their needs.

11 I²C interface

11.1 I²C instruction

I²C (inter-integrated circuit) bus interface manages the communication between the microcontroller and serial I²C bus. It supports master and slave modes, with up to 1 Mbit/s of communication speed (enhanced edition).

11.2 I²C main features

- I²C bus
  - Master and slave modes
  - Multimaster capability
  - Standard mode (100 kHz), fast mode (400 kHz) and enhanced fast mode (1 MHz)
  - 7-bit and 10-bit address modes
  - Two 7-bit slave address (2 addresses, one of them can be masked)
  - Broadcast call mode
  - Programmable data setup and hold time
  - Clock stretching capability
- Support DMA transfer
- Programmable digital noise filter
- Wake up from Deepsleep mode through address match event
- Support SMBus 2.0 protocol
  - PEC generation and verification
  - Acknowledgement control for command and data
  - ARP(address resolution protocol)
  - Master capability
  - Device capability
  - SMBus reminder capability
  - Timeout detection
  - Idle detection
- PMBus

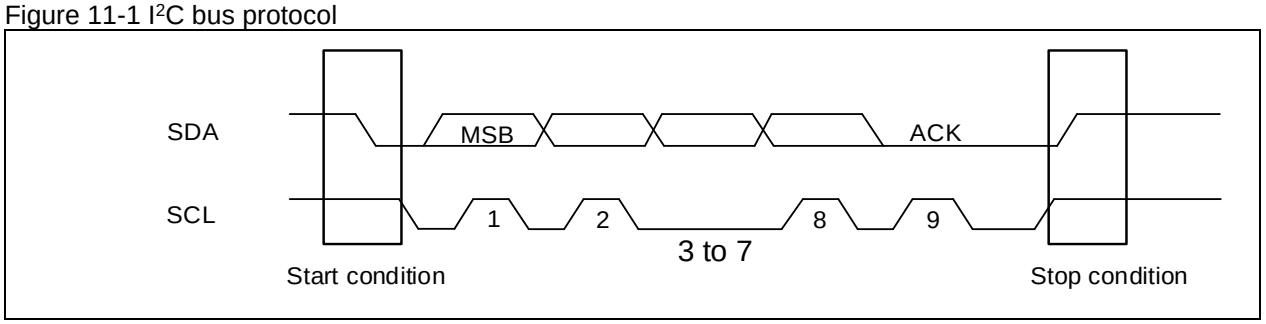
11.3 I²C function overview

I²C bus consists of a data line (SDA) and clock line (SCL). It can achieve a maximum of 100 kHz speed in standard mode, up to 400 kHz in fast mode, and 1 MHz in fast mode plus. A frame of data transmission begins with a Start condition and ends with a Stop condition. The bus is kept in busy state after receiving the Start condition, and becomes idle as long as it receives the Stop condition.

Start condition: SDA switches from high to low when SCL is set high.

Stop condition: SDA switches from low to high when SCL is set high.

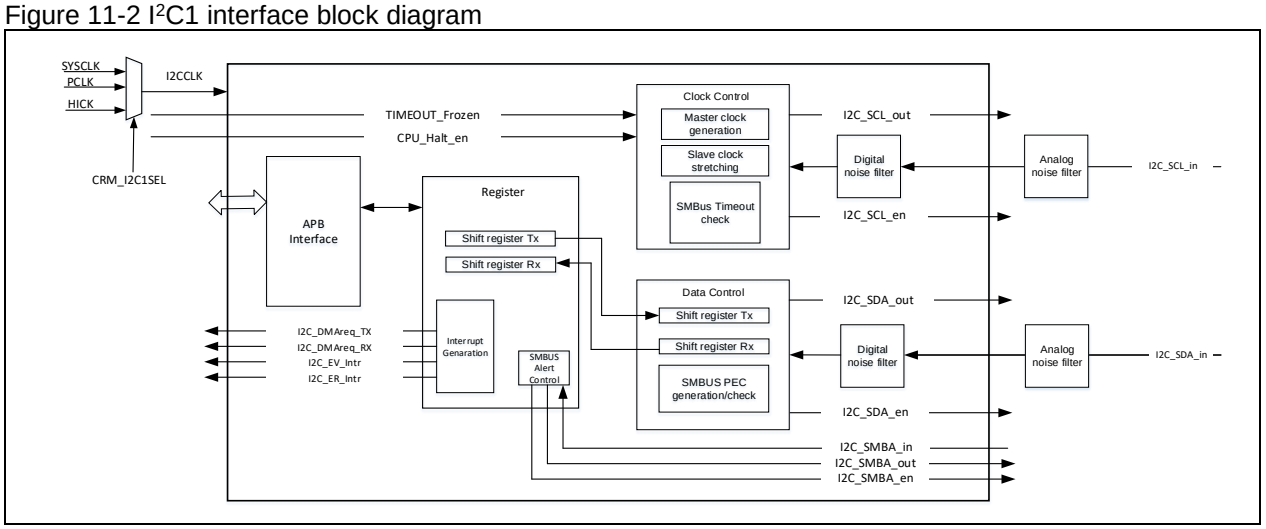
Figure 11-1 I²C bus protocol



11.4 I²C interface

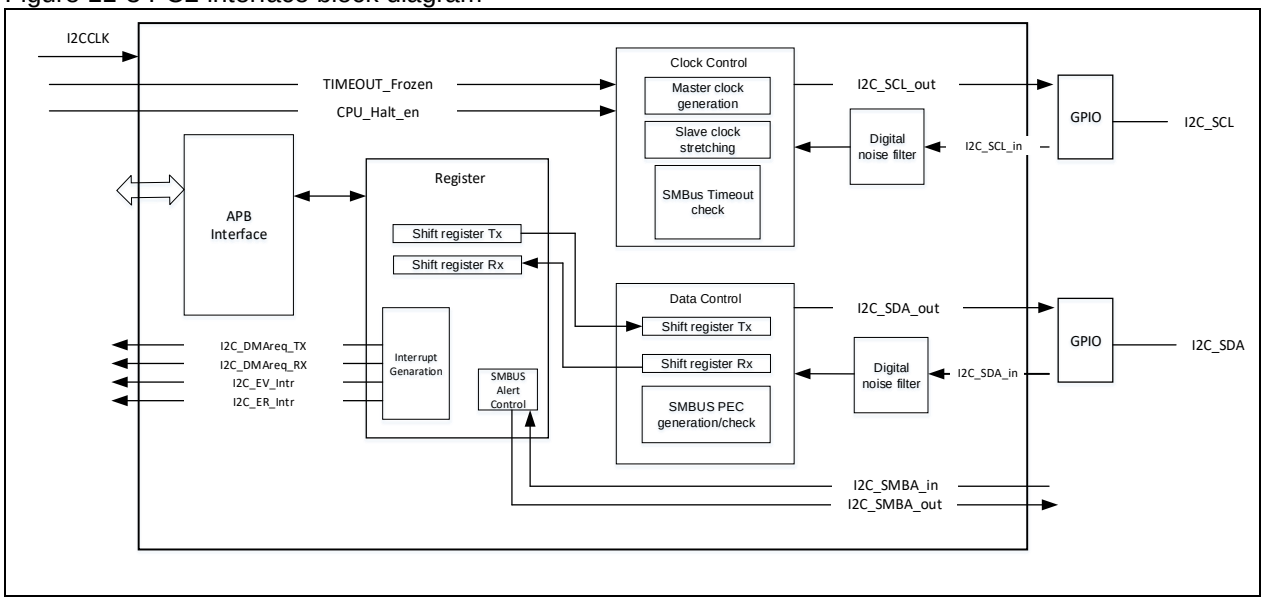
The figure below shows the block diagram of I²C interface.

Figure 11-2 I²C1 interface block diagram



For I²C1, it is possible to select SYSCLK, PCLK or HICK as its clock source through the I2C1SEL bit in the CRM_MISC2 register. Waking up from Deepsleep mode is supported. The I²C1 has an analog filter able to shield noise within 50ns.

Figure 11-3 I²C2 interface block diagram



The PCLK is used as the clock source of I2C2. The I2C2 does not support to wake up from Deepsleep mode, and it has no analog filter.

1. Operating mode

I²C bus interface can operate both in master mode and slave mode. Switching from master mode to slave mode, vice versa, is supported as well. By default, the interface operates in slave mode. When the Start condition is activated, the I²C bus interface switches from slave mode to master mode, and returns to slave mode automatically at the end of data transfer (Stop condition is triggered).

2. Communication process

- Master mode communication:
  1. Start condition generation
  2. Address transmission
  3. Data Tx or Rx
  4. Stop condition generation
  5. End of communication

- Slave mode communication:
  1. Wait until the address is matched
  2. Data Tx or Rx
  3. Wait for the generation of Stop condition
  4. End of communication

3. Digital filter capability

The digital filter is available on both SCL and SDA lines. It is enabled by setting the DFLT[3: 0] bit (0~15) in the I2C_CTRL1 register to reduce noise on bus on a large scale. The filter time is DLFT x t_{I2C_CLK}. The digital filter is not allowed to be altered when the I^2C is enabled.

4. Address control

Both master and slave support 7-bit and 10-bit addressing modes.

Slave address mode:

- In 7-bit mode (ADDR1MODE=0)
  - ADDR1EN=1, ADDR2EN=0 stands for a single address mode: only matches OADDR1
  - ADDR1EN=1, ADDR2EN=1 stands for dual address mode: matches OADDR1 and OADDR2
- In 10-bit mode (ADDR1MODE=1)
  - Only supports a single address mode (ADDR1EN=1, ADDR2EN=0), matches OADDR1

Slave address masking capability

The Slave address 2 (OADDR2) is maskable, which is done by setting the ADDR2MASK[2: 0].

- 0: Address bit [7:1]
- 1: Address bit [7:2]
- 2: Address bit [7:3]
- 3: Address bit [7:4]
- 4: Address bit [7:5]
- 5: Address bit [7:6]
- 6: Address bit [7]
- 7: All addresses, excluding those reserved by I^2C

Support special slave address:

- Broadcast call address (0b0000000x): This address is enabled when GCAEN=1.
- SMBus device default address (0b1100001x): This address is enabled for SMBus address resolution protocol in SMBus device mode (DEVADDREN = 1).
- SMBus master default address (0b0001000x): This address is enabled for SMBus master notification protocol in SMBus master mode (HADDREN = 1).
- SMBus alert address (0b0001100x): This address is enabled for SMBus alert response address protocol in SMBus master mode when SMBALERT = 1.

Refer to SMBus2.0 protocol for more information.

Slave address matching procedure:

- Receive a Start condition
- Address matching
- The slave sends an ACK if address is matched
- ADDRF=1, with SDIR indicating the transmission direction
  - When SDIR=0, slave enters receiver mode, starting receiving data.
  - When SDIR=1, slave enters transmitter mode, starting transmitting data.

5. Clock stretching capability

Clock stretching is enabled by default (STRETCH=0 in the I2C_CTRL1 register). The slave can hold the SCL line low for software operation. If the clock stretching capability is not supported by master, then the STRETCH must be set to 1 in the I2C_CTRL register. It should be noted that the clock stretching capability of I^2C slave must be configured before the I^2C peripherals are enabled.

Clock stretching capability enabled

I^2C slave stretches the SCL clock in one of the following conditions:

- Address reception: When the address received from slave matches the local address enabled (ADDRF=1 in the I2C_STS), the SCL line is pulled down until the ADDRF is cleared by setting the ADDRC in the I2C_CLR.

- Data reception: the shift register receives a new data before the data in the I2C_RXDT register has been read. In this case, the SCL line is pulled low until the data of the I2C_RXDT register is read.
- Data transmission: If no data is written when the ADDRF is cleared, TDBE= 1 in the I2C_STS, then the SCL line will be pulled down until the data is written to the I2C_TXDT.
- Data transmission: If no data is written to the I2C_TXDT after the completion of the previous data transfer, the SCL line will be pulled low until data is written to the I2C_TXDT
- Data reception: When the shift register has received another new byte before the data in the I2C_RXDT register is read, the I2C will hold the SCL bus low to wait for the software to read I2C_RXDT register
- When slave byte control mode is selected (SCTRL=1 in the I2C_CTRL1) and RLDEN=1 in the I2C_CTRL2 register, if TCRLD = 1, indicating the completion of the last data transfer, then the TCRLD will be cleared by hardware so as to release the SCL line after a non-zero value is written to the CNT bit in the I2C_CTRL2 register

Clock stretching capability disabled

The SCL clock is disabled when STRETCH=1 in the I2C_CTRL1 register, with the following conditions worth our notice:

- Address reception: The SCL clock is not stretched when the address received by slave matches its own address enabled.
- Data reception: If there are data pending to be read in the I2C_RXDT register before the next ACK signal, an overflow will occur, and the OUF bit will be set to 1 in the I2C_STS register.
- Data transmission: If no new data is written to the I2C_TXDT register after the completion of the previous data transfer, an underflow will occur, and the OUF will be set to 1 in I2C_STS register

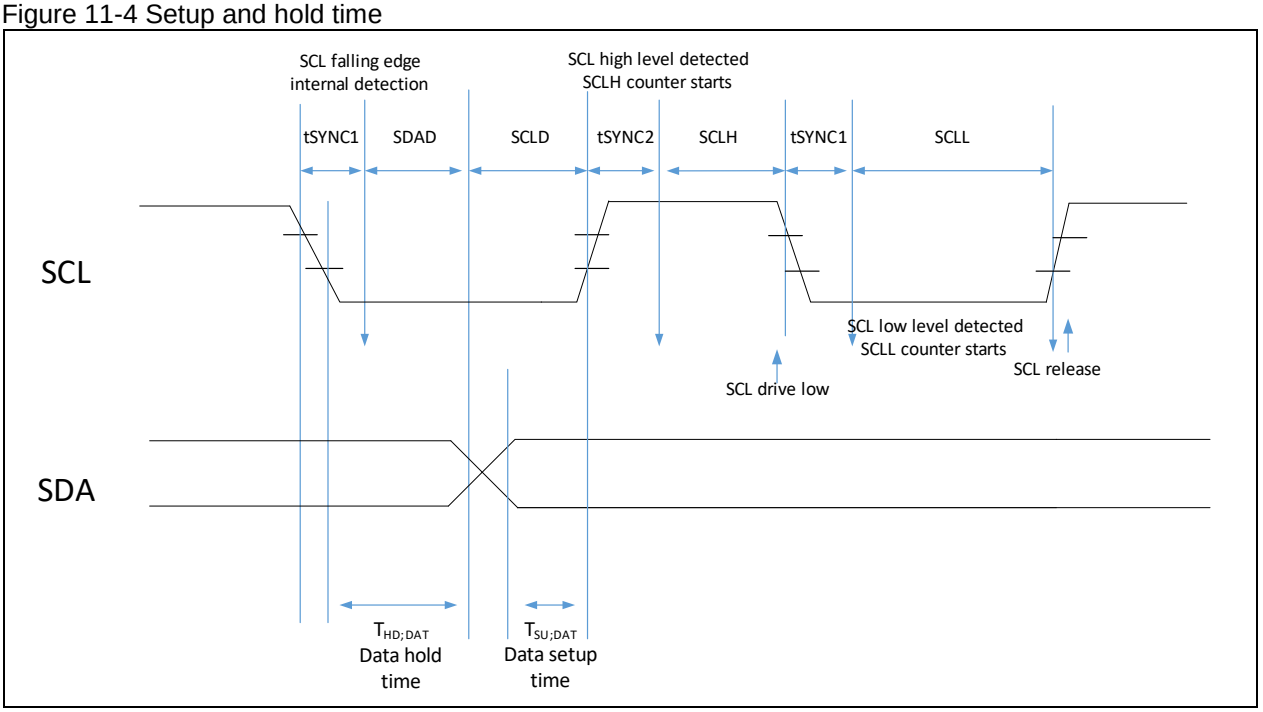
11.4.1 I²C timing control

I²C core is clocked by I2C_CLK whereas the I2C_CLK is clocked by PCLK1. The PCLK1 should be set to be less than 4/3 SCL cycles. The corresponding bits in the I2C_CLKCTR register are used for timing configuration.

— DIV[7:0]: I²C clock divider
— SDAD[3:0]: Data hold time ($t_{HD;DAT}$)
— SCLD[3:0]: Data setup time ($t_{SU;DAT}$)
— SCLH[7:0]: SCL high
— SCLL[7:0]: SCL low

Note: Timing configuration cannot be modified once the I²C is enabled.

Figure 11-4 Setup and hold time



It is possible to configure data hold time ($t_{HD;DAT}$) and data setup time ($t_{SU;DAT}$) by setting the DIV[7:0], SDAD[3:0] and SCLD[3:0] in the I2C_CLKCTRL register.

- Data hold time ($t_{HD;DAT}$): refers to the duration from SCL falling edge to SDA output

  $t_{HD;DAT} = t_{SDAD} + t_{SYNC}$

  $t_{SDAD} = SDAD \times (DIV + 1) \times t_{I2C\_CLK}$

  $t_{SYNC} = (DLFT + 3) \times t_{I2C\_CLK} - t_f$

  t_{SYNC} consists of three parts:
  - SCL falling edge time (t_f)
  - Digital filter input latency (DLFT x t_{I2C_CLK})
  - Synchronization delay between SCL and I2C_CLK (2~3 I2C_CLK cycles)

- Data setup time ($t_{SU;DAT}$): refers to the duration from SDA output to SCL rising edge

  $t_{SU;DAT} = SCLD \times (DIV+1) \times t_{I2C\_CLK} - t_r$

In master mode, the width of SCL signals (high and low) can be configured freely by setting the DIV[7: 0], SCLH[7: 0] and SCLL[7: 0] in the I2C_CLKCTRL register.

SCL low: When the SCL low signal is detected, the internal SCLL counter starts counting until it reaches the SCLL value. At this point, the SCL line is released and become high.

SCL high: When the SCL high signal is detected, the internal SCLH counter starts counting. When the counter value reaches the SCLH value, the SCL line is pulled low. In the process of SCL remaining high, if it is pulled low by external bus, the internal SCLH counter will stop counting and start counting in SCL low mode, laying the foundation for clock synchronization

- SCL high signal width:

  $t_{HIGH} = (SCLH + 1) \times (DIV + 1) \times t_{I2C\_CLK}$

- SCL low signal width:

  $T_{Low} = (SCLL + 1) \times (DIV + 1) \times t_{I2C\_CLK}$

Table 11-1 I²C timing specifications

Parameter		Standard mode		Fast mode		Fast mode plus		SMBus	
		Min.	Max.	Min.	Max.	Min.	Max.	Min.	Max.
f_{SCL} (kHz)	SCL clock frequency		100		400		1000		100
t_{LOW} (us)	SCL clock low	4.7		1.3		0.5		4.7	
t_{HIGH} (us)	SCL clock high	4.0		0.6		0.26		4.0	50
tHD;DAT (us)	Data hold time	0		0	0.9	0	0.45	300	
tSU;DAT (ns)	Data setup time	250		100		50		250	
tr (ns)	SCL/SDA rising edge		1000		300		120		1000
tf (ns)	SCL/SDA falling edge		300		300		120		300

11.4.2 Data transfer management

Data transfer counter is available in the I²C interface to control communication flow. It is mainly used for:

- NACK transmission: master reception mode
- STOP transmission: master reception/transmission modes
- RESTART generation: master reception/transmission modes
- ACK control: slave mode (SMBus)
- PEC transmission/reception: master/slave modes

Generally, the data transfer management counter (by setting the CNT[7:0] in the I2C_CTRL2) is applicable to master mode. It is disabled in slave mode. This counter is used only in SMBus mode for the ACK control and PEC reception of each byte by the slave. In SMBus mode, the slave enables data counter with the SCTRL bit in the I2C_CTRL2 register.

Byte control through master

The CNT[7:0] bit in the I2C_CTRL2 register is used to configure the number of bytes to be transferred, ranging from 1 to 255. If the number of data to be transferred is greater than 255, then the RLDEN bit has to be set to 1 in the I2C_CTRL2 register to enable reload mode. The following configuration processes are described in two aspects:

- ≤255 bytes, for example, the number of data to be transferred is 100 bytes
  - Step 1: Disable reload mode by setting RLDEN=0
  - Step 2: Set CNT[7:0]=100
- >255 bytes, for example, the number of data to be transferred is 600 bytes
  - Step 1: Enable reload mode by setting RLDEN=1
  - Step 2: Set CNT[7:0]=255, the remaining bytes are 600-255=345 bytes
  - Step 3: After the completion of 255-byte data transfer, the TCRLD is set in the I2C_STS register, and then set CNT[7:0]=255 for continuous transfer, and the remaining bytes are 345-255=90
  - Step 4: After the completion of the second 255-byte data transfer, the TCRLD is set in the I2C_STS register, and then set RLDEN=0 to disable reload mode before setting CNT[7:0]=90 for continuous transfer.

There are two ways to stop the last data transfer (RLDEN=0, reload mode is disabled)

- Stop data transfer automatically (ASTOPEN=1 in the I2C_CTRL2)
  - When the number of data programmed in the CNT[7:0] bit has been fully transferred, the master will automatically send a STOP condition
- Stop data transfer by software (ASTOPEN=0 in the I2C_CTRL2)
  - When the number of data programmed in the CNT[7:0] has been fully transferred, the TDC will be set in the I2C_STS register, and the SCL, at this point, will be pulled low, an interrupt generated if TDCIEN is enabled. In this case, it is possible to send a STOP condition by setting GENSTOP=1 in the I2C_CTRL2 register, or send a RESTART condition by setting GENSTART=1 in the I2C_CTRL2 register, before clearing TDC flag by software.

Byte control through slave

This feature is enabled by setting the SCTRL bit in the I2C_CTRL2 register so that the slave is able to control ACK/NACK signals of each byte independently.

- Proceed as below:
  - Set SCTRL=1 to enable Byte Control Through Slave
  - After the slave address is matched (ADDRF=1), enable reload mode by setting RLDEN=1, and then set CNT[7:0]=1
  - When a byte is received, the TCRLD is set in the I2C_STS register, and the slave will pull the SCL bus low between the 8th and 9th clock edges. At this point, the user can read the RXDT register and generate an ACK or NACK signal through the NACKEN bit in the I2C_CTRL2 register
  - When an NACK signal is generated, it indicates the end of communication
  - When an ACK signal is generated, the communication flow keeps going on. At this point, set CNT[7:0]=1, the TCRLD flag is cleared automatically by hardware, and the SCL bus is released for the reception of the next byte

As we know, the value in the CNT[7:0] bit is not limited to 1. If you want to receive 8 data, for example, but just want to control the ACK/NACK signals of the 8th data. Proceed as below: set CNT[7:0]=8, the slave will receive 7 consecutive data, with ACK signals sent. Once the 8th data reception is completed,

the SCL bus is pulled low, and then proceed as above to select whether to send an ACK or NACK.

It should be noted that the clock stretching capability must be enabled (STRETCH=0 in the I2C_CTRL1 register) before selecting Byte Control Mode Through Slave.

Table 11-2 I²C configuration table

Description	RLDEN bit	ASTOPEN bit	SCTRL bit
Master transmit/receive RESTART	0	0	×
Master transmit/receive STOP	0	1	×
Slave receive (control ACK/NACK of each byte)	1	×	1
Slave transmit/receive (ACK response to all bytes)	×	×	0

11.4.3 I²C master communication flow

1. **I²C clock initialization (by setting the I2C_CLKCTRL register)**
   - I²C clock divider: DIV[7:0]
   - Data hold time ($t_{HD;DAT}$): SDAD[3:0]
   - Date setup time ($t_{SU;DAT}$): SCLD[3:0]
   - SCL high duration: SCLH[7:0]
   - SCL low duration: SCLL[7:0]

   The register can be configured by means of Artery_I2C_Timing_Configuration tool.

2. **Set the number of bytes to be transferred**
   - ≤255 bytes

     Disable reload mode by setting RLDEN=0 in the I2C_CTRL2 register

     Set CNT[7:0]=N in the I2C_CTRL2 register
   - >255 bytes

     Enable reload mode by setting RLDEN=1 in the I2C_CTRL2 register

     Set CNT[7:0]=255 in the I2C_CTRL2 register

     Remaining bytes N=N-255

3. **End of data transfer**
   - ASTOPEN=0: stop data transfer by software. After the completion of data transfer, the TDC is set in the I2C_STS register, and GENSTOP=1 or GENSTART=1 is written by software to send a STOP or START condition
   - ASTOPEN=1: data transfer is stopped automatically. A STOP condition is sent at the end of data transfer

4. **Set slave address**
   - Set slave address value (by setting the SADDR bit in the I2C_CTRL2 register)
   - Set slave address mode (by setting the ADDR10 bit in the I2C_CTRL2 register)

     ADDR10=0: 7-bit address mode

     ADDR10=1: 10-bit address mode

5. **Set transfer direction (by setting the DIR bit in the I2C_CTRL2 register)**
   - DIR=0: Master reception
   - DIR=1: Master transmission

6. **Start data transfer**

   When GENSTART=1 in the I2C_CTRL2 register, the master starts sending a START condition and slave address. After receiving the ACK from the slave, ADDRF=1 is asserted in the I2C_STS register. The ADDRF flag can be cleared by setting ADDRC=1 in the I2C_CLR register, and then data transfer starts.

7. **Master transmit**
   1. I2C_TXDT data register is empty, the shift register is empty, TDIS=1 in the I2C_STS register
   2. Writing 1 to the TXDT register, and data is immediately moved to the shift register
   3. TXDT register becomes empty, TDIS=1 again

4. Writing 2 to the TXDT register, TDIS is cleared
5. Repeat step 2 and 3 until the data in the CNT[7:0] is sent
6. If TCRLD=1 (reload mode) in the I2C_STS register, the following two circumstances should be noted:

   Remaining bytes N>255: write 255 to the CNT bit, N=N-255, TCRLD is cleared, and data transfer continues

   Remaining bytes N≤255: Disable reload mode (RLDEN=0), write N to the CTN bit, TCRLD is cleared, and data transfer continues

8. Master receive

1. After the slave address is matched, ADDRF=1 in the I2C_STS register, clear ADDRF flag by setting ADDRC=1 in the I2C_CLR register, and then it starts sending data
2. After the reception of data, RDBF=1, read the RXDT register will clear the RDBF automatically
3. Repeat step 2 until the reception of data programmed in the CNT[7:0] bit
4. If TCRLD=1 (reload mode) in the I2C_STS, the following two circumstances should be noted:

   Remaining bytes N>255: write 255 to the CNT bit, N=N-255, TCRLD is cleared, and data transfer continues

   Remaining bytes N≤255: Disable reload mode (RLDEN=0), write N to the CTN bit, TCRLD is cleared, and data transfer continues
5. After the reception of the last data, an NACK signal will be sent by master

9. Stop condition

- STOP condition generation:

  ASTOPEN=0: TDC=1 in the I2C_STS register, set GENSTOP=1 to generate a STOP condition

  ASTOPEN=1: A STOP condition is generated automatically
- Wait for the generation of a STOP condition, when a STOP condition is generated, STOPF=1 is asserted in the I2C_STS register. The STOPF flag can be cleared by setting STOPC=1 in the I2C_CLR register, and then transfer stops.

When the host receives an NACK signal during transmission, then ACKFAIL is set in the I2C_STS register, and a STOP condition is sent to stop communication, whatever mode (either ASTOPEN=0 or ASTOPEN=1).

Master transmitter

Figure 11-5 I²C master transmission flow

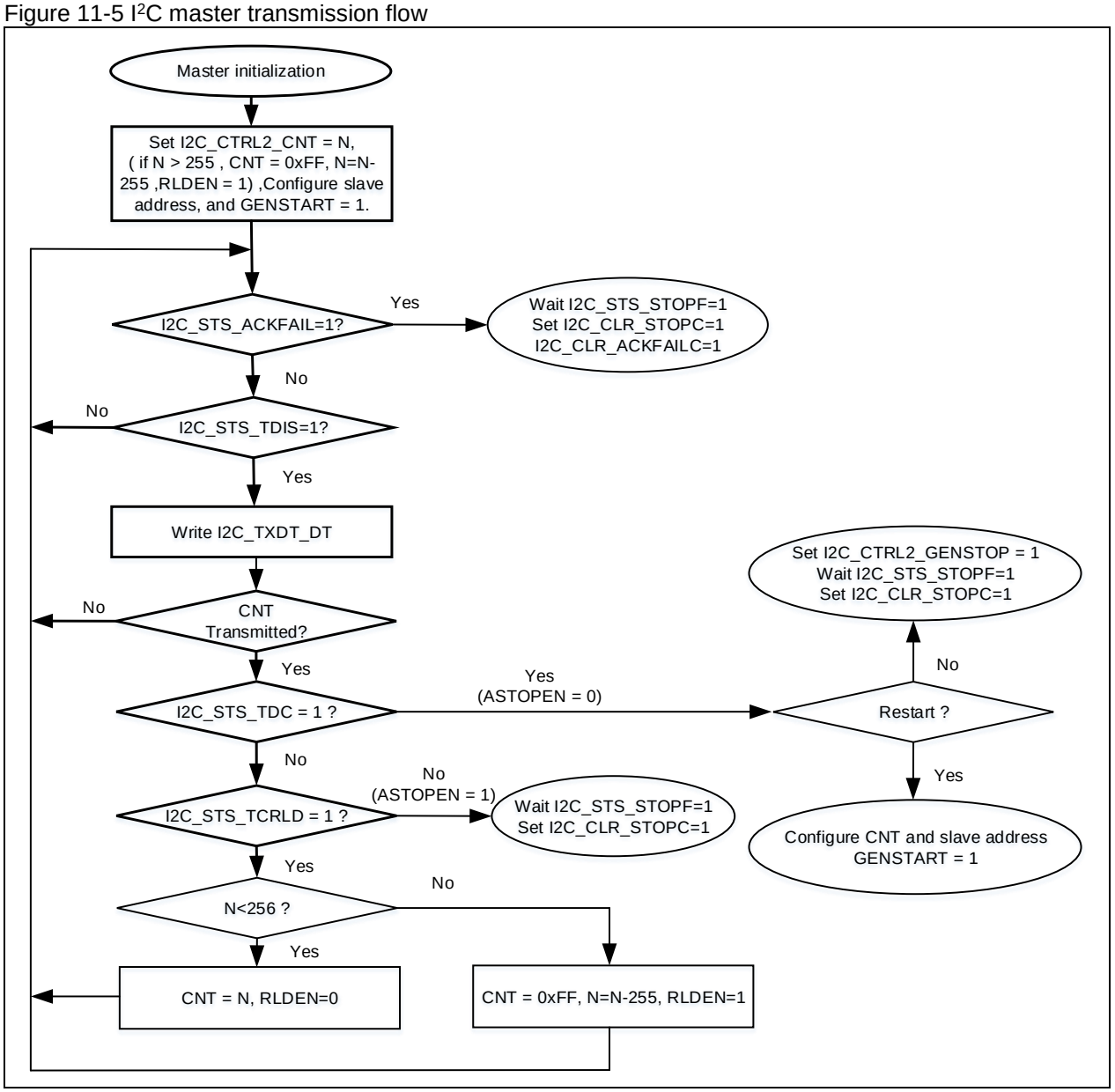
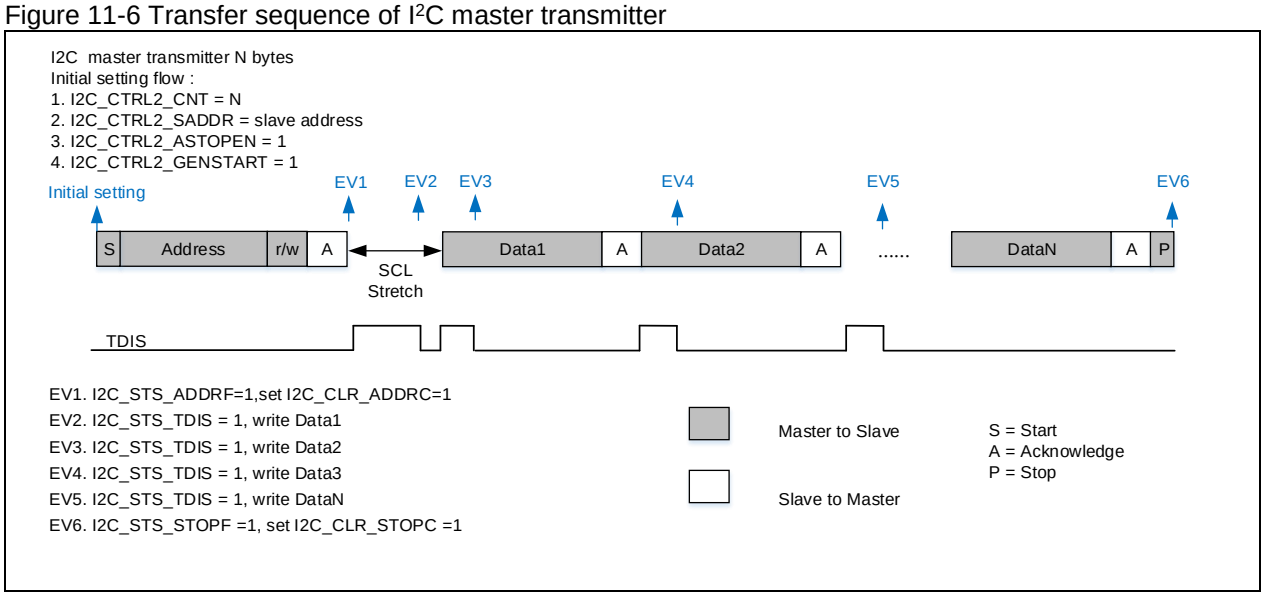


Figure 11-6 Transfer sequence of I²C master transmitter



Master receiver

Figure 11-7 I²C master receive flow

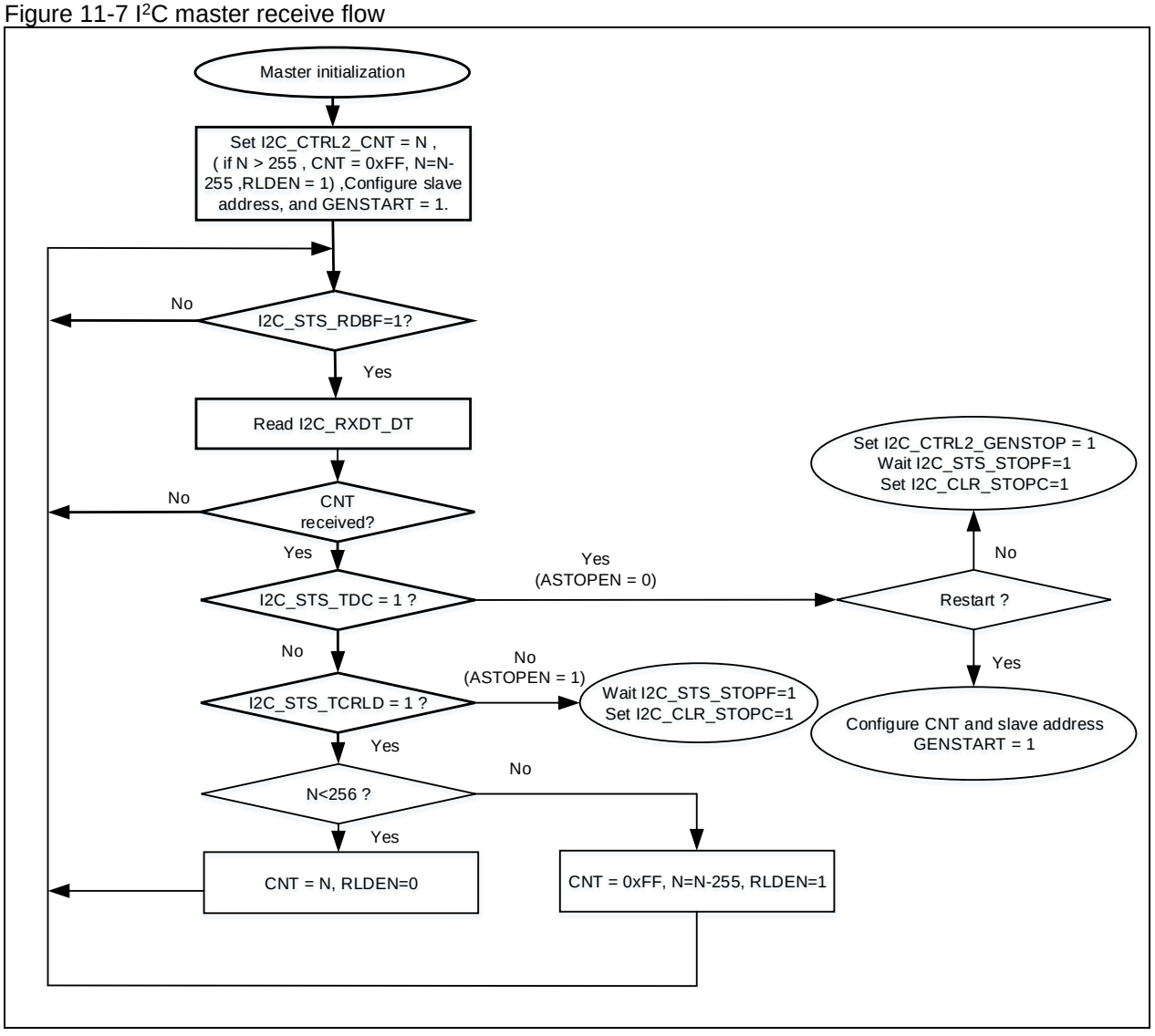
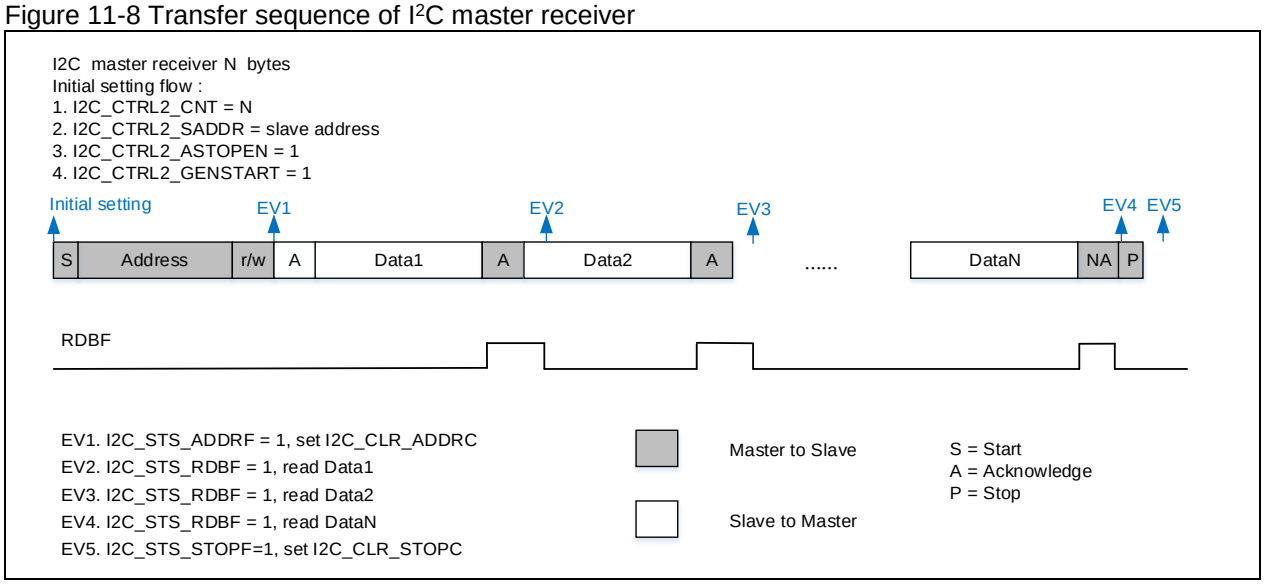


Figure 11-8 Transfer sequence of I²C master receiver



Master special transfer sequence

In 10-bit addressing mode, the READH10 bit of the I2C_CTRL2 register is used to generate a special timing. When READH10=1, the master sends data to the slave before read access to the slave, as shown in the figure below:

Operating method:

When ASTOPEN=0, data is transferred from the master to the slave. At the end of data transfer, READH10=0 is asserted, and then the master starts receiving data from the slave.

Figure 11-9 10-bit address read access when READH10=1

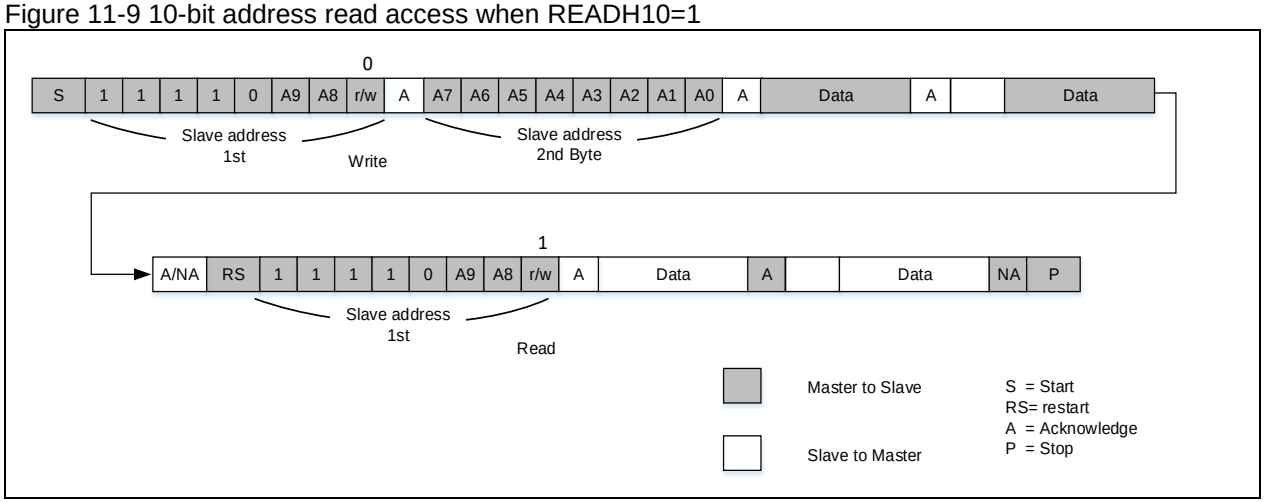
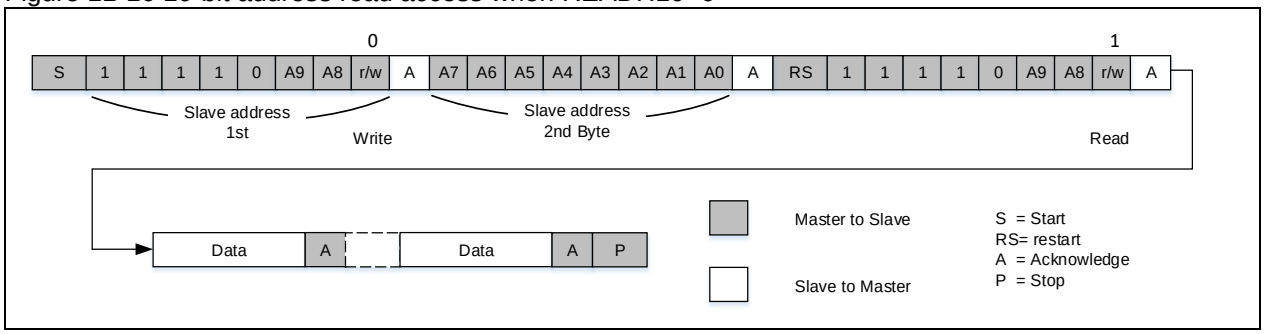


Figure 11-10 10-bit address read access when READH10=0



11.4.4 I²C slave communication flow

1. I²C clock initialization (by setting the I2C_CLKCTRL register)

- I²C clock divider: DIV[7:0]
- Data hold time ($t_{HD;DAT}$): SDAD[3:0]
- Data setup time ($t_{SU;DAT}$): SCLD[3:0]

The register can be configured by means of Artery_I2C_Timing_Configuration tool.

2. Set local address 1

- Set address mode:

  7-bit address: by setting ADDR1MODE=0 in the I2C_OADDR1 register

  10-bit address: by setting ADDR1MODE=1 in the I2C_OADDR1 register
- Set address 1: by setting the ADDR1 bit in the I2C_OADDR1 register
- Enable address 1: by setting ADDR1EN=1 in the I2C_OADDR1 register

3. Set local address 2

- Set address 2: by setting the ADDR2 bit in the I2C_OADDR2 register
- Set address 2 mask bit: by setting the ADDR2MASK bit in the I2C_OADDR2 register
- Enable address 2: by setting ADDR2EN=1 in the I2C_OADDR2 register

Note: Only 7-bit address mode is available in the address 2 mode. The ADDR2MASK bit is used to mask some address bits so that the slave can respond to some specific addresses. Refer to Section 14.2 for more information about the ADDR2MASK bit.

In the case of using only one address, only address 1 needs to be configured, without the need of address 2 mode.

4. Wait for address matching

When the local address is received, the ADDRF bit is set to 1 in the I2C_STS register. The data transfer direction can be obtained by read access to the SDIR bit in the I2C_STS register. When SDIR=0, it indicates that the slave is receiving data, whereas SDIR=1 indicates that the slave is sending data. The ADDR[6:0] bit of the I2C_STS register indicates what kind of address has been received, which is particularly helpful in the case when the dual address mode is used and the address 2 mode mask bit is set.

Data transfer starts when the ADDRF is cleared by setting ADDRC=1 of the I2C_CLR register

5. Data transfer (slave transmission, clock stretching enabled, STRETCH=0)

After address matching:

1. I2C_TXDT data register becomes empty, the shift register becomes empty, and TDIS=1 in the I2C_STS register
2. Data is then transferred to the shift register after writing 1 to the TXDT register
3. The TXDT register then becomes empty, and the TDIS is set again
4. TDIS is cleared by writing 2 to the TXDT register
5. Repeat step 3 and 4 until the completion of data transfer
6. Wait for the generation of an NACK signal. Once received, the ACKFAILF is set in the I2C_STS register. The ACKFAILF flag is cleared by writing 1 to the ACKFAILC
7. Wait for the generation of a STOP condition. Once received, the STOPF is set in the I2C_STS register. At the end of data transfer, the STOPF is cleared by writing 1 to the STOPC, transmission ends.

In the case of the clock stretching being disabled (STRETCH=1), if data has not yet been written to the TXDT register before the transmission of the first bit of the to-be-transferred data (that is, before the generation of SDA edge), an underrun error may occur, and the OUF bit is set in the I2C_STS register, sending 0xFF to the bus.

In order to write data in time, data must be written to the DT register first before communication, in two different ways:

- Write operation through software: Clear the TXDT register by setting TDBE=1 through software, and then write the first data to the TXDT register, the TDBE is cleared
- Write operation through interrupts or DMA: Clear the TXDT register by setting TDBE=1 through software, then set the TDIS bit to generate a TDIS event, which generates an interrupt or DMA request. At this point, data is written to the TXDT register using DMA or interrupts

6. Data transfer (slave receive, clock stretching enabled, STRETCH=0)

After address matching:

1. I2C_RXDT register becomes empty, the shift register becomes empty, and RDBF=0 in the I2C_STS register
2. Upon the receipt of data, RDBF=1; The RDBF is cleared by read operation to the RXDT register
3. Repeat step 2 until the completion of all data transfer
4. Wait for the generation of a STOP condition. Once received, the STOPF is set in the I2C_STS register. The STOPF can be cleared by writing 1 to the STOPC bit in the I2C_CLR register, transfer ends.

In slave receive mode, the slave byte control mode (SCTRL=1) can be used for data reception. This mode allows to control ACK/NACK signals of each byte received. This mode is typically available in SMBus protocol. Refer to the data transmission management section for more information about this mode.

Note that the slave must read the received data in the case of the clock stretching being disabled (STRETCH=1). If one-byte data has been received and data is not read yet before the end of the next data reception, an overrun error occurs, setting the OUF bit in the I2C_STS register, and sending NCAK.

An interrupt will be generated if the corresponding interrupt enable bit is enabled. For more information about interrupt generation, refer to the interrupt chapter.

Slave transmission

Figure 11-11 I²C slave transmission flow

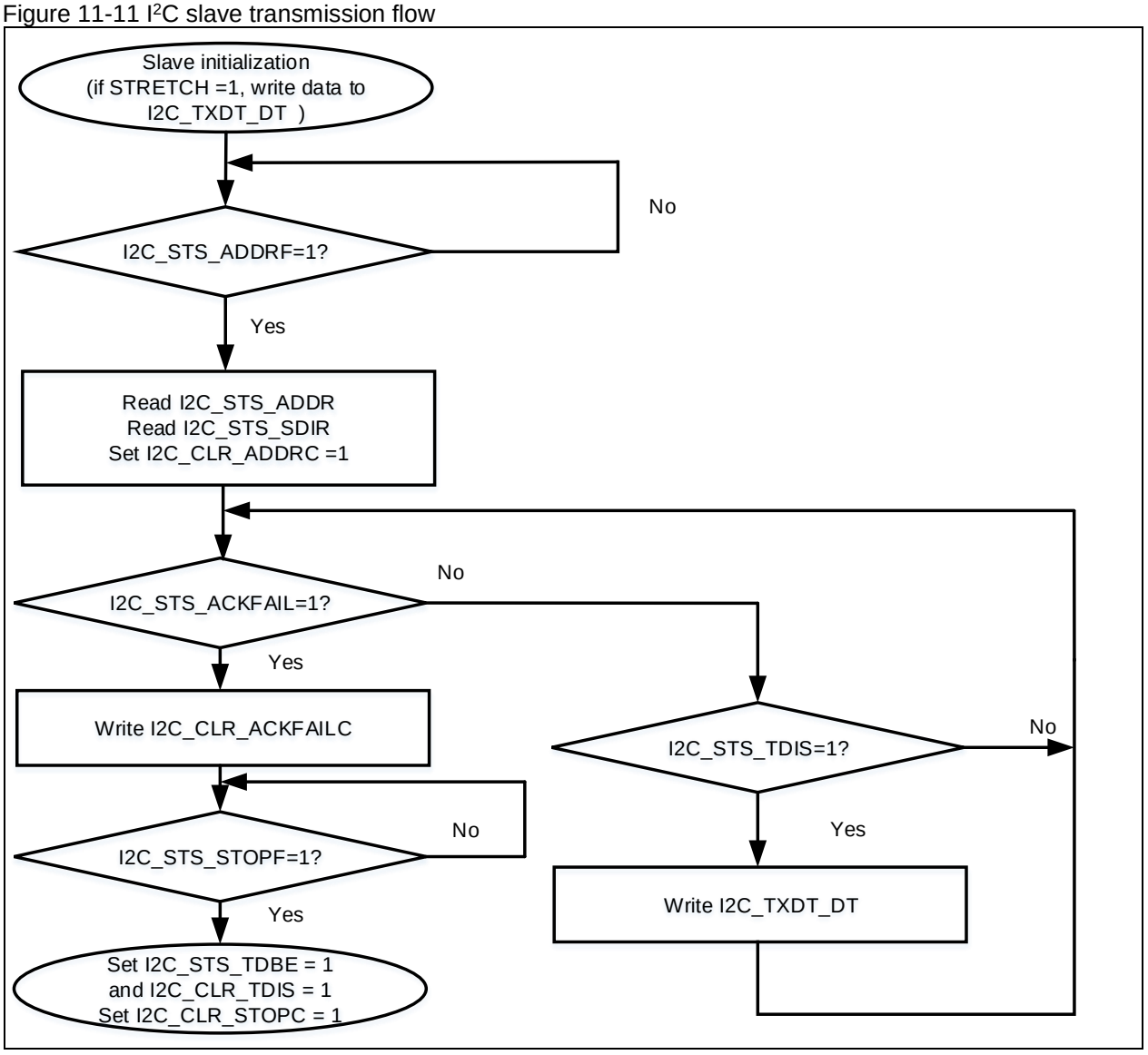
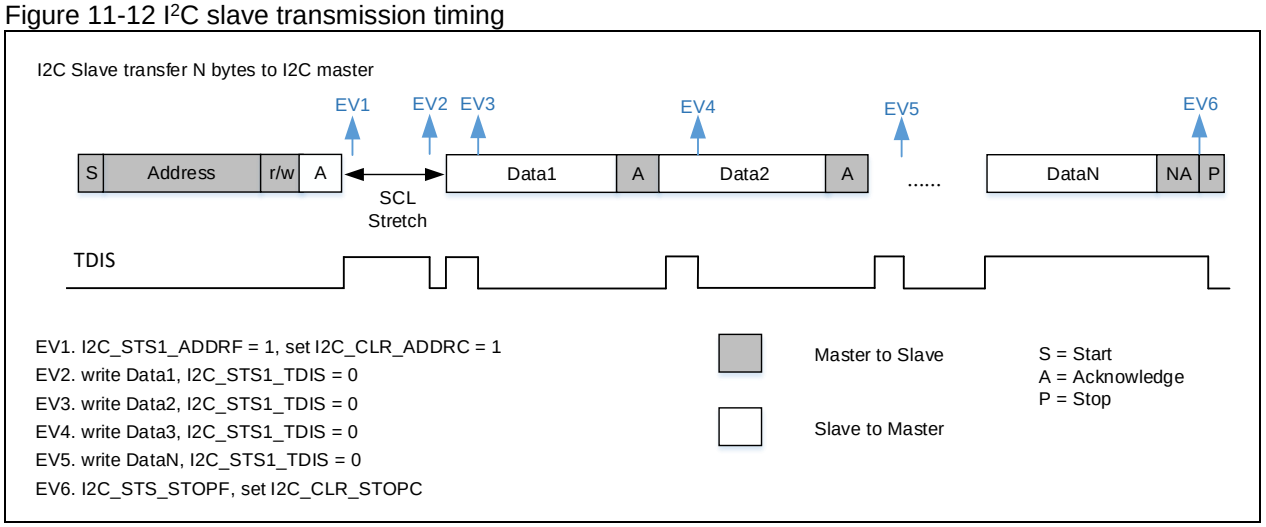


Figure 11-12 I²C slave transmission timing



Slave receive

Figure 11-13 I²C slave receive flow

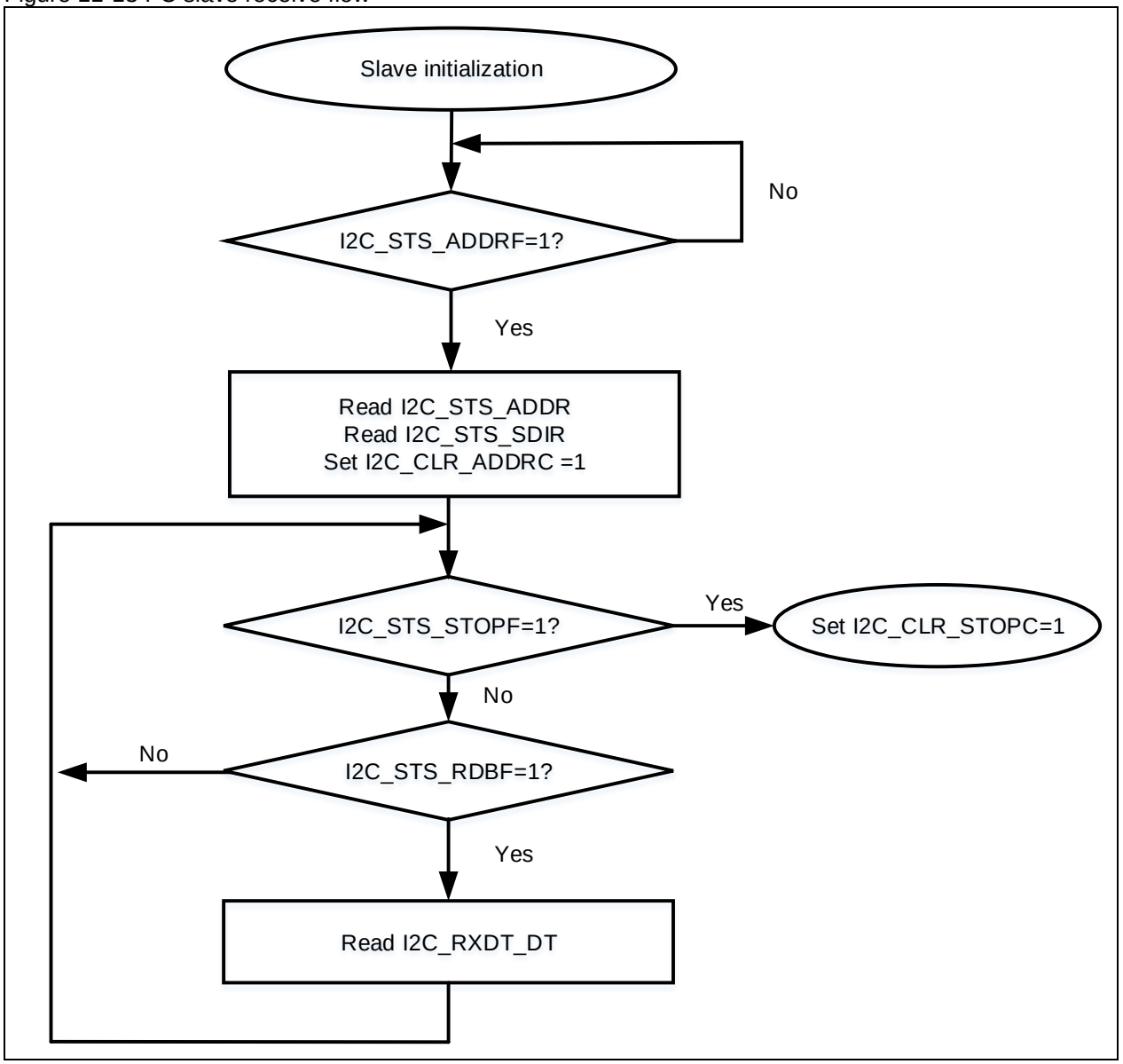
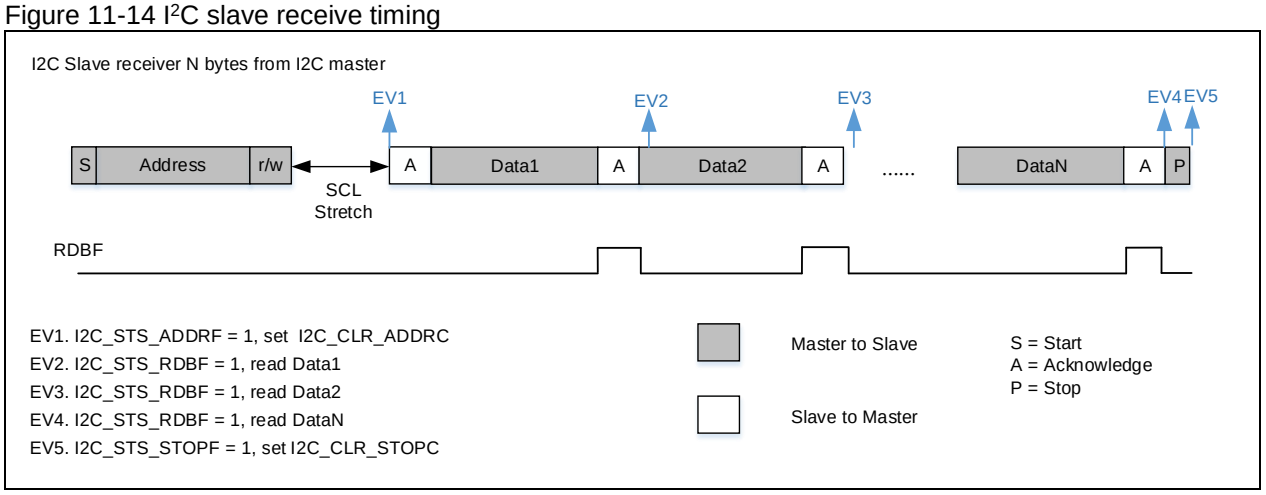


Figure 11-14 I²C slave receive timing



11.4.5 SMBus

The System Management Bus (SMBus) is a two-wire interface through which various devices can communicate with each other. It is based on I²C. With SMBus, the device can provide manufacturer information, tell the system its model/part number, report different types of errors and accept control parameters and so on. For more information, refer to SMBus 2.0 protocol.

Difference between SMBus and I²C

1. SMBus requires a minimum speed of 10 kHz for the purpose of management and monitor. It is quite easy to know whether the bus is in Idle state or not as long as a parameter is input while running on a certain transmission speed, without the need of detecting the STOP signals one after another, or even keeping STOP and other parameter monitor. There is no limit for I²C.
2. SMBus transmission speed ranges from 10 kHz to 100 kHz. In contrast, I²C has no minimum requirement, and its maximum speed varies from one mode to another, namely, 100 kHz in standard mode and 400 kHz in fast mode.
3. After reset, SMBus needs timeout, but there is no limit for I²C in this regard.

SMBus address resolution protocol (ARP)

SMBus address conflicts can be resolved by dynamically assigning a new unique address to each device. Refer to SMBus 2.0 protocol for more information about ARP.

Setting the DEVADDREN bit in the I2C_CTRL1 register can enable the I²C interface to recognize the default device address (0b1100001x). However, unique device identifier (UDID) and the detailed protocol implementation should be handled by software.

SMBus host notify protocol

The slave device can send data to the master device through SMBus host notify protocol. For example, the slave can notify the host to implement ARP with this protocol. Refer to SMBus 2.0 protocol for details on SMBus host notify protocol.

In host mode (HADDREN =1), the I²C interface is enabled to recognize the 0b0001000x (default host address)

SMBus Alert

SMBALERT is an optional signal that connects the ALERT pin between the host and the salve. With this signal, the slave notifies the host to access the slave. SMBALERT is a wired-AND signal. For more information about SMBus Alert, refer to SMBus2.0 protocol.

The detailed sequences are as follows:

SMBus host

1. Enable SMBus Alert mode by setting SMBALERT=1
2. Enable ALERT interrupt if necessary
3. When an alert event occurs on the ALERT pin (ALERT pin changes from high to low)
4. The host will generate ALERT interrupt if enabled
5. The host then processes the interrupt and accesses to all devices through ARA (Alert Response Address 0001100x) so as to get the slave addresses. Only the devices with pulled-down SMBALERT can acknowledge ARA.

6. The host then continues to operate based on the slave addresses available.

SMBus slave

1. When an alert event occurs and the ALERT pin changes from high to low (SMBALERT=1), the slave responses to ARA (Alert Response Address) address (0001100x)
2. Wait until the host gets the slave addresses through ARA
3. Report its own address, but it continues to wait if the arbitration is lost
4. Address is reported properly, and the ALERT pin is released (SMBALERT=0).

Packet error checking (PEC)

Packet error checking (PEC) is used to guarantee the correctness and integrity of data transfer. This is done by using CRC-8 polynomial:

$C(x) = x^8 + x^2 + x + 1$

PEC calculation is enabled when PECEN=1 in the I2C_CTRL1 register to check address and data.

PEC transfer:

- Host: PEC transfer is enabled by setting PECTEN=1 in the I2C_CTRL2 register. The host sends a PEC as soon as the number of data transfer reaches N-1 (CNT=N)
- Slave: PEC transfer is enabled by setting PECTEN=1 in the I2C_CTRL2 register. When the number of data transfer reaches N-1 (CNT=N), the slave will consider the Nth data as a PEC and check it. A NACK will be sent if the PEC checking result is not correct, setting the PECERR flag in the I2C_STS register. In case of slave transmission mode, a NACK must follow the PEC whatever the checking result.

SMBus timeout

The SMBus protocol specifies three timeout detection modes:

- Low level timeout ($t_{TIMEOUT}$): The time duration when the SCL is kept low in a single mode (taking into account master/slave device, however actively or passively pulled low)
- Cumulative timeout for a slave device at low level ($t_{LOW:SEXT}$): The cumulative time duration when the SCL is pulled low by a slave device during the period from a START condition to a STOP condition
- Cumulative timeout for a master device at low level ($t_{LOW:MEXT}$): The cumulative time duration when the SCL is pulled low by a master device during the period from the ACK of the last byte to the 8th bit of the next byte (a single byte)

It should be noted that both $t_{LOW:SEXT}$ and $t_{LOW:SEXT}$ only deal with the time when they set themselves low level, excluding the time when they are pulled low by external sources. In contrast, both of these cases are considered in the calculation of $t_{TIMEOUT}$.

Table 11-3 SMBus timeout specification

Type of timeout	Min	Max	Unit
$t_{TIMEOUT}$	25	35	ms
$t_{LOW:SEXT}$	-	25	ms
$t_{LOW:MEXT}$	-	10	ms

The I²C peripherals embeds two counters for timeout detection, which can be configured through the I2C_TIMEOUT register. When a timeout event occurs, the TMOUT is set in the I2C_STS register. The TMOUT bit can be cleared by writing 1 to the TMOUTC bit in the I2C_CLR register

- EXTTIME: This is used to the cumulative timeout detection for master/slave devices at low level Timeout duration=(EXTTIME + 1) x 2048 x T_{I2C_CLK}
- TOTIME: This is used for clock level timeout detection, selected through the TOMODE bit.

  TOMODE=0: Low level timeout detection, timeout duration=(TOTIME + 1) x 2048 x T_{I2C_CLK}

  TOMODE=1: High level timeout detection, timeout duration=(TOTIME + 1) x 4 x T_{I2C_CLK}

Table 11-4 SMBus timeout detection configuration

Type of timeout	Other configuration	Enable bit	Timeout calculation
$t_{TIMEOUT}$	TOMODE=0	TOEN=1	(TOTIME + 1) x 2048 x T_{I2C_CLK}
$t_{LOW:SEXT}$	-	EXTEN=1	(EXTTIME + 1) x 2048 x T_{I2C_CLK}
$t_{LOW:MEXT}$	-	EXTEN=1	(EXTTIME + 1) x 2048 x T_{I2C_CLK}

Slave receive byte control

In slave receive mode, the slave receive byte control mode (SCTRL=1) can be used to control ACK/NACK signals of each received byte. Refer to the data transfer management section for more information.

Table 11-5 SMBus mode configuration

Transfer mode	PECTEN	RLDEN	ASTOPEN	SCTRL
Master transmit/receive +STOP	1	0	1	-
Master transmit/receive+RESTART	1	0	0	-
Slave receive	1	1	-	1
Slave transmit	1	0	-	-

How to use the interface in SMBus mode

1. Set SMBus default address acknowledgement:

   HADDREN=1: Master default address acknowledged (0b0001000x)

   DEVADDREN=1: Device default address acknowledged (0b1100001x)
2. Configure PEC
3. Slave receive byte control mode can be enabled (with SCTRL bit in the I2C_CTRL1) in slave mode, if necessary
4. Other configurations follow the I^2C

However, the detailed SMBus protocol implementation should be handled by software, since the I^2C interface is only enabled to recognize the addresses of SMBus protocols.

11.4.6 SMBus master communication flow

The SMBus is similar to the I^2C in terms of master communication flow.

1. **I^2C clock initialization (by setting the I2C_CLKCTRL register)**
   - I^2C clock divider: DIV[7:0]
   - Data hold time ($t_{HD;DAT}$): SDAD[3:0]
   - Data setup time ($t_{SU;DAT}$): SCLD[3:0]
   - SCL high duration: SCLH[7:0]
   - SCL low duration: SCLL[7:0]

   The register can be configured by means of Artery_I2C_Timing_Configuration tool.
2. **SMBus-related initialization**
   - Select SMBus host: host default address acknowledged (0b0001000x) by setting HADDREN=1
   - Enable PEC calculation: set PECEN=1 in the I2C_CTRL register
   - Enable PEC transfer: set PECTEN=1 in the I2C_CTRL2 register
3. **Set the number of bytes to be transferred**
   - Disable reload mode by setting RLDEN=0 in the I2C_CTRL2 register
   - Set CNT[7:0]=N in the I2C_CTRL2 register

   The number of bytes to be transferred is <255 in SMBus mode at one time.
4. **End of data transfer**

- ASTOPEN=0: stop data transfer by software. After the completion of data transfer, the TDC is set in the I2C_STS register, and GENSTOP=1 or GENSTART=1 is written by software to send a STOP or START condition
- ASTOPEN=1: data transfer is stopped automatically. A STOP condition is sent at the end of data transfer

5. Set slave address

- Set slave address value (by setting the SADDR bit in the I2C_CTRL2 register)
- Set 7-bit slave address mode (by setting the ADDR10=0 in the I2C_CTRL2 register)

6. Set transfer direction (by setting the DIR bit in the I2C_CTRL2 register)

- DIR=0: Master reception
- DIR=1: Master transmission

7. Start data transfer

In case of GENSTART=1 in the I2C_CTRL2 register, the master starts sending a START condition and slave address. After receiving the ACK from the slave, ADDRF=1 is asserted in the I2C_STS register. The ADDRF flag can be cleared by setting ADDRC=1 in the I2C_CLR register, and then data transfer starts.

8. Master transmit

1. I2C_TXDT data register is empty, the shift register is empty, TDIS=1 in the I2C_STS register
2. Writing 1 to the TXDT register, and data is immediately moved to the shift register
3. TXDT register becomes empty, TDIS=1 again
4. Writing 2 to the TXDT register, TDIS is cleared
5. Repeat step 2 and 3 until the specified data (N-1) is sent
6. The master will automatically transmit the Nth data, that is, PEC.

9. Master receive

1. After the reception of data, RDBF=1, read the RXDT register will clear the RDBF automatically
2. Repeat step 1 until the reception of the specified data (N). The Nth data is set as PEC. A NACK is automatically sent after the receipt of the Nth data (PEC) whatever the PEC result.

10. STOP condition

- STOP condition generation:

  ASTOPEN=0: TDC=1 in the I2C_STS register, set GENSTOP=1 to generate a STOP condition

  ASTOPEN=1: A STOP condition is generated automatically
- Wait for the generation of a STOP condition, when a STOP condition is generated, STOPF=1 is asserted in the I2C_STS register. The STOPF flag can be cleared by setting STOPC=1 in the I2C_CLR register, and then transfer stops

Figure 11-15 SMBus master transmission flow

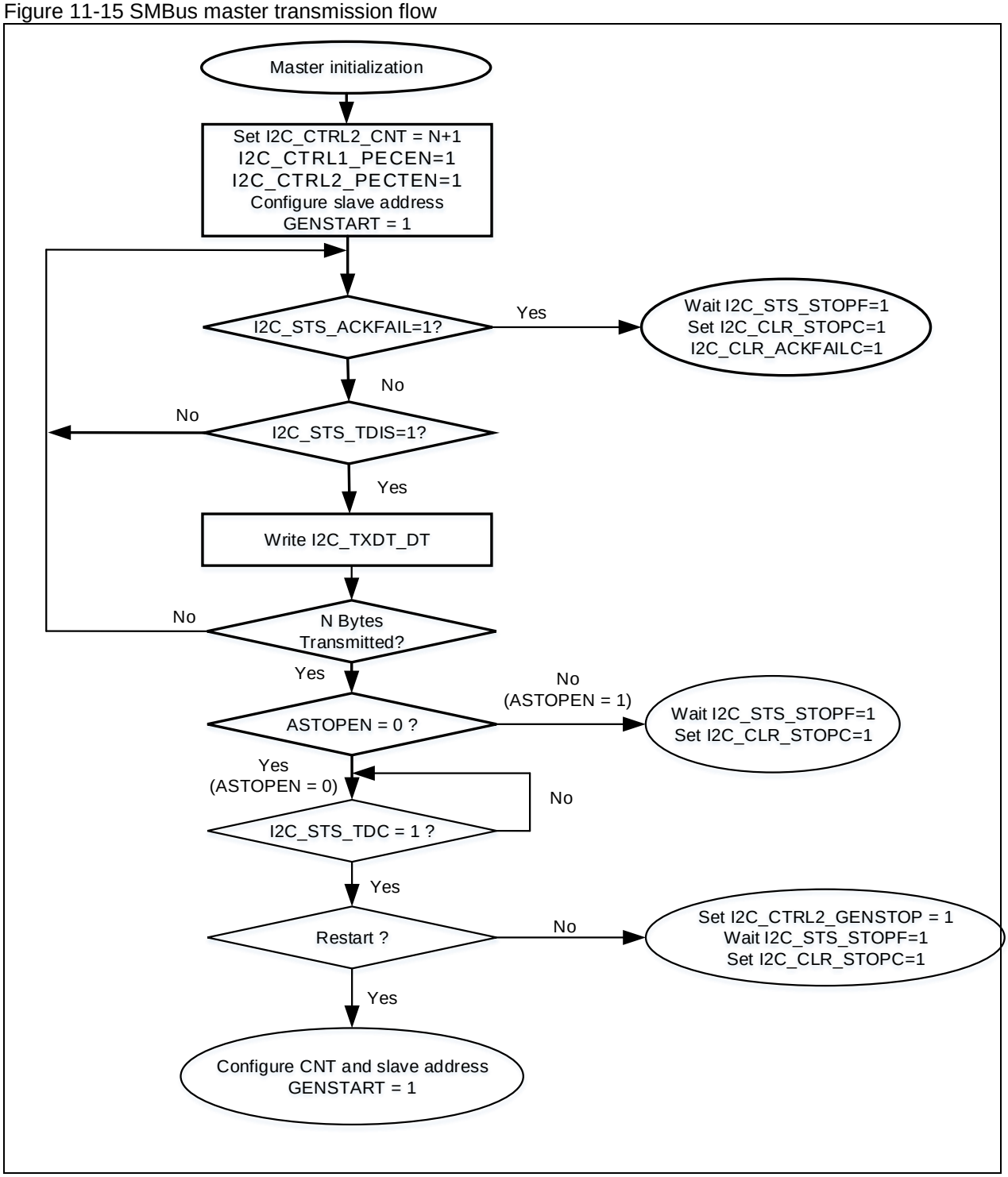
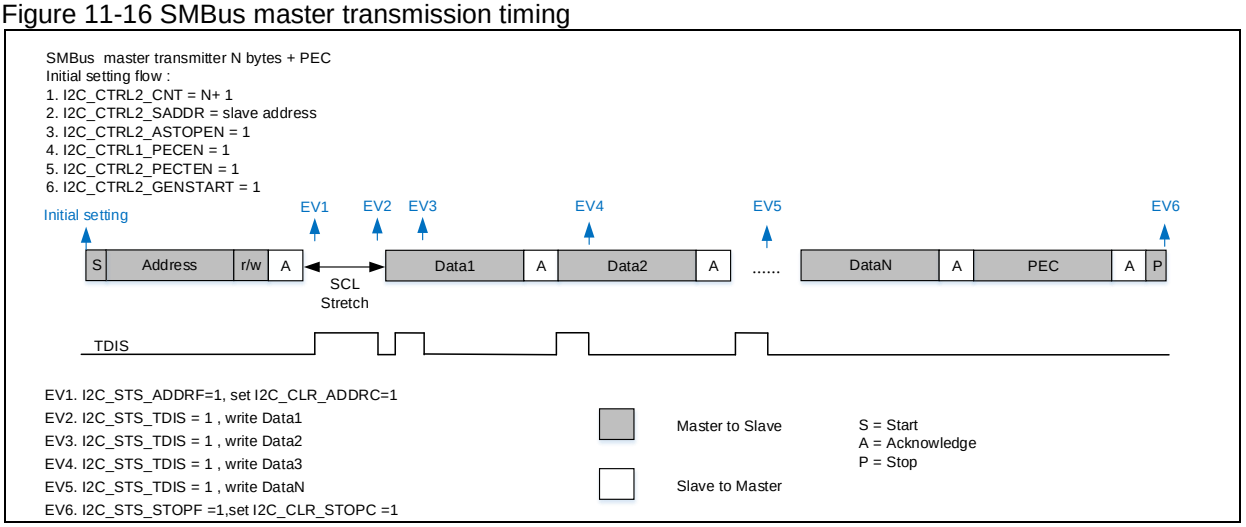


Figure 11-16 SMBus master transmission timing



SMBus master receive flow

Figure 11-17 SMBus master receive flow

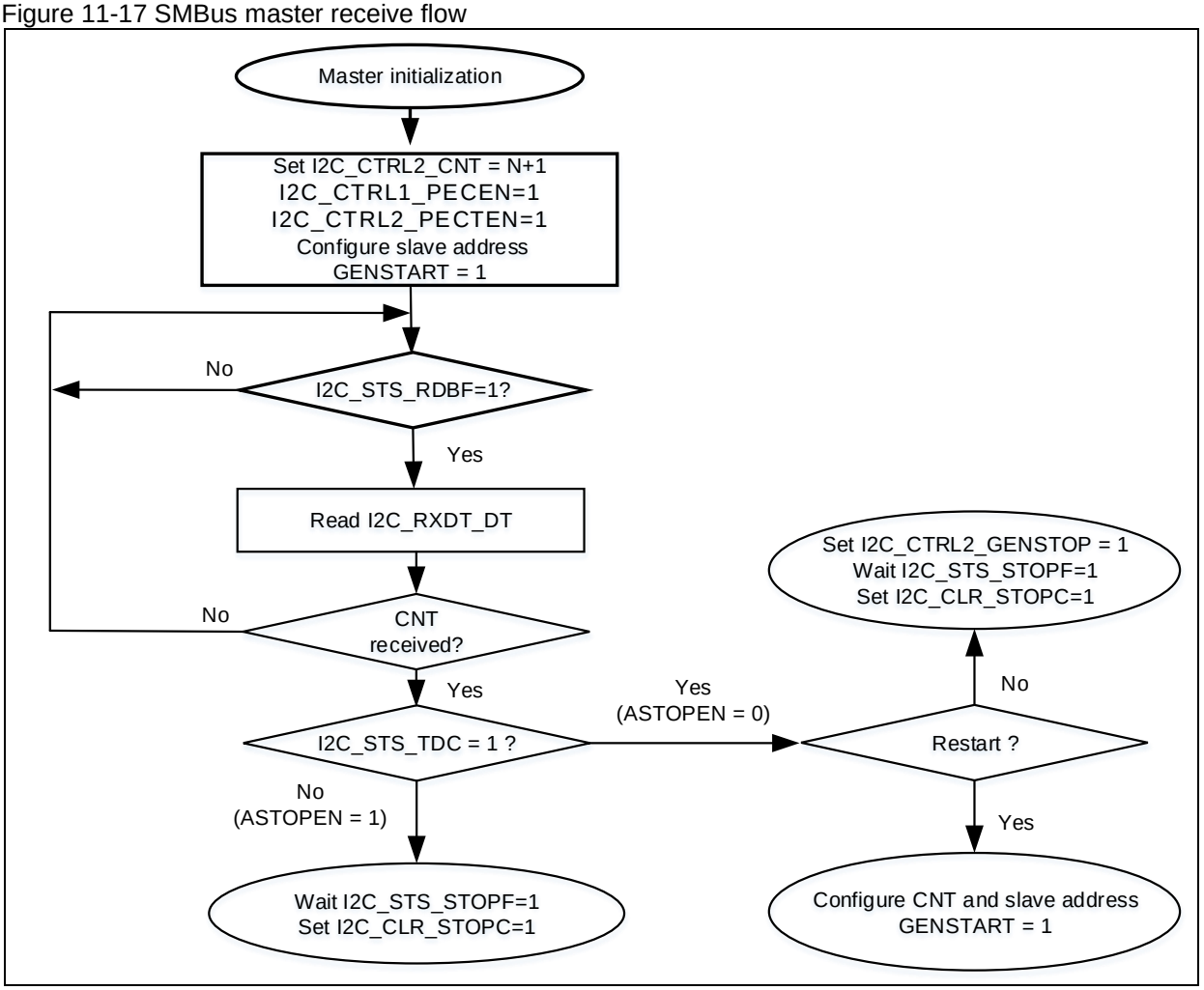
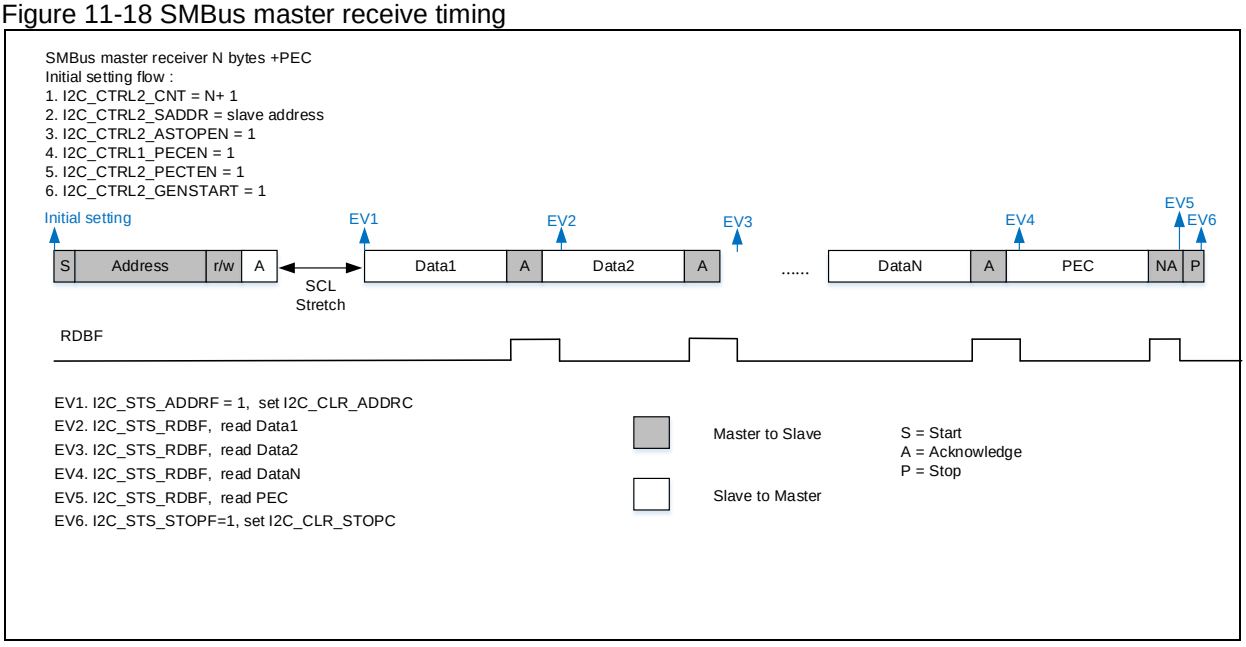


Figure 11-18 SMBus master receive timing



11.4.7 SMBus slave communication flow

The SMBus is similar to the I²C in terms of slave communication flow.

1. I²C clock initialization (by setting the I2C_CLKCTRL register)

- I²C clock divider: DIV[7:0]
- Data hold time ($t_{HD;DAT}$): SDAD[3:0]
- Data setup time ($t_{SU;DAT}$): SCLD[3:0]

The register can be configured by means of Artery_I2C_Timing_Configuration too

2. Set own address

- Set 7-bit address mode: by setting ADDR1MODE = 0 in the I2C_OADDR register
- Set address 1: by setting the ADDR1 bit in the I2C_OADDR1 register
- Enable address 1: by setting ADDR1EN=1 in the I2C_OADDR1 register

3. SMBus-related initialization

- Select SMBus host: device default address acknowledged (0b1100001x) by setting DEVADDREN=1
- Enable PEC calculation: Set PECEN=1 in the I2C_CTRL1 register
- Set slave byte control mode:

  Slave transmit: disable byte control mode by setting SCTRL=0 in the I2C_CTRL1 register

  Slave receive: enable byte control mode by setting SCTRL=1 in the I2C_CTRL1 register

4. Wait for address matching

When the local address is received, the ADDRF bit is set in the I2C_STS register. The data transfer direction can be obtained by read access to the SDIR bit in the I2C_STS register. When SDIR=0, it indicates that the slave is receiving data, whereas SDIR=1 indicates that the slave is sending data. The ADDR[6:0 bit of the I2C_STS register indicates what kind of address has been received, which is particularly helpful in the case when the dual address mode is used and the address 2 mode mask bit is set.

Enable PEC transfer: by setting PECTEN=1 in the I2C_CTRL2 register

Set the number of data to be transferred:

- Slave transmit: by setting CNT=N in the I2C_CTRL2 register
- Slave receive: by setting CNT=1 in the I2C_CTRL2 register

Set reload mode:

- Slave transmit: by setting RLDEN=0 in the I2C_CTRL2 register
- Slave receive: by setting RLDEN=1 in the I2C_CTRL2 register

The ADDRF flag can be cleared by setting ADDRC=1 in the I2C_CLR register, and then data

transfer starts.

5. Data transfer (slave transmission, clock stretching enabled, STRETCH=0)

After address matching:

1. I2C_TXDT data register becomes empty, the shift register becomes empty, and TDIS=1 in the I2C_STS register
2. Data is then transferred to the shift register after writing 1 to the TXDT register
3. The TXDT register then becomes empty, and the TDIS is set again
4. TDIS is cleared by writing 2 to the TXDT register
5. Repeat step 3 and 4 until data (N-1) is sent
6. The slave will automatically transmit the Nth data, that is, PEC
7. Wait for the generation of an NACK signal. Once received, the ACKFAILF is set in the I2C_STS register. The ACKFAILF flag is cleared by writing 1 to the ACKFAILC
8. Wait for the generation of a STOP condition. Once received, the STOPF is set in the I2C_STS register. At the end of data transfer, the STOPF is cleared by writing 1 to the STOPC, transmission ends.

6. Data transfer (slave receive, clock stretching enabled, STRETCH=0)

After address matching:

1. I2C_RXDT register becomes empty, the shift register becomes empty, and RDBF=0 in the I2C_STS register
2. Upon the receipt of one-byte data, RDBF=1 and TCRLD=1, then the SCL is pulled low by the slave
3. The RDBF is cleared by read operation to the RXDT register
4. NACKEN bit of the I2C_CTRL register can be configured to generate an ACK or NACK, if needed

   If a NACK is detected, it indicates the completion of communication

   If an ACK is detected, communication continues. Writing CNT=1 will automatically clear the TCRLD flag by hardware, and the SCL is released by the slave for the reception of the next data
5. Repeat step 2/3/4 until the completion of data reception (N-1)
6. Set RLDEN=0 of the I2C_CTRL2 register to disable reload mode. Set CNT=1 to repeat step 2/3 to receive a PEC. The PECERR bit will be set if a PEC error occurs
7. Wait for the generation of a STOP condition. Once received, the STOPF is set in the I2C_STS register. The STOPF can be cleared by writing 1 to the STOPC bit in the I2C_CLR register, transfer ends.

SMBus slave transmission

Figure 11-19 SMBus slave transmission flow

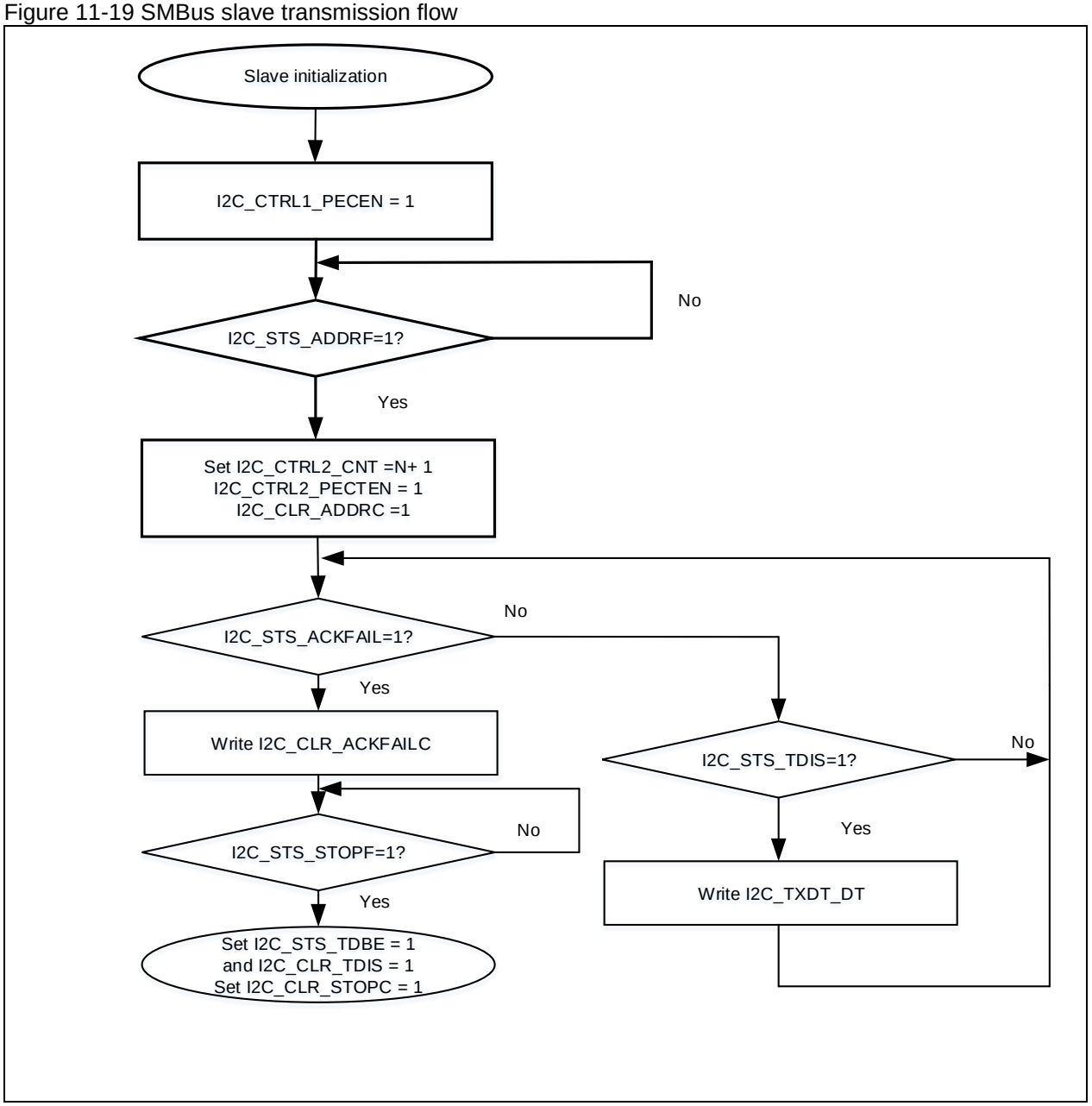
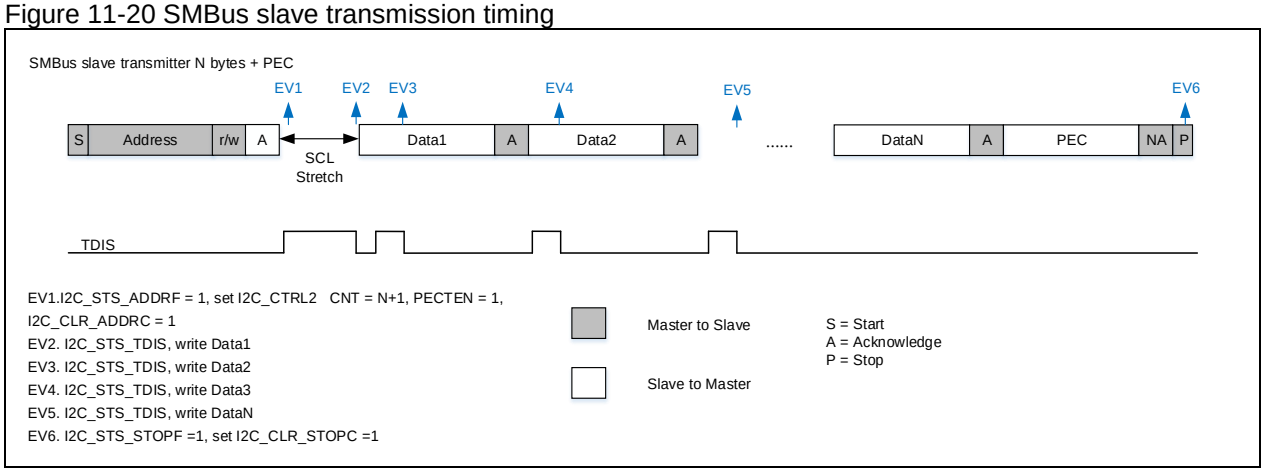


Figure 11-20 SMBus slave transmission timing



SMBus slave receive

Figure 11-21 SMBus slave receive flow

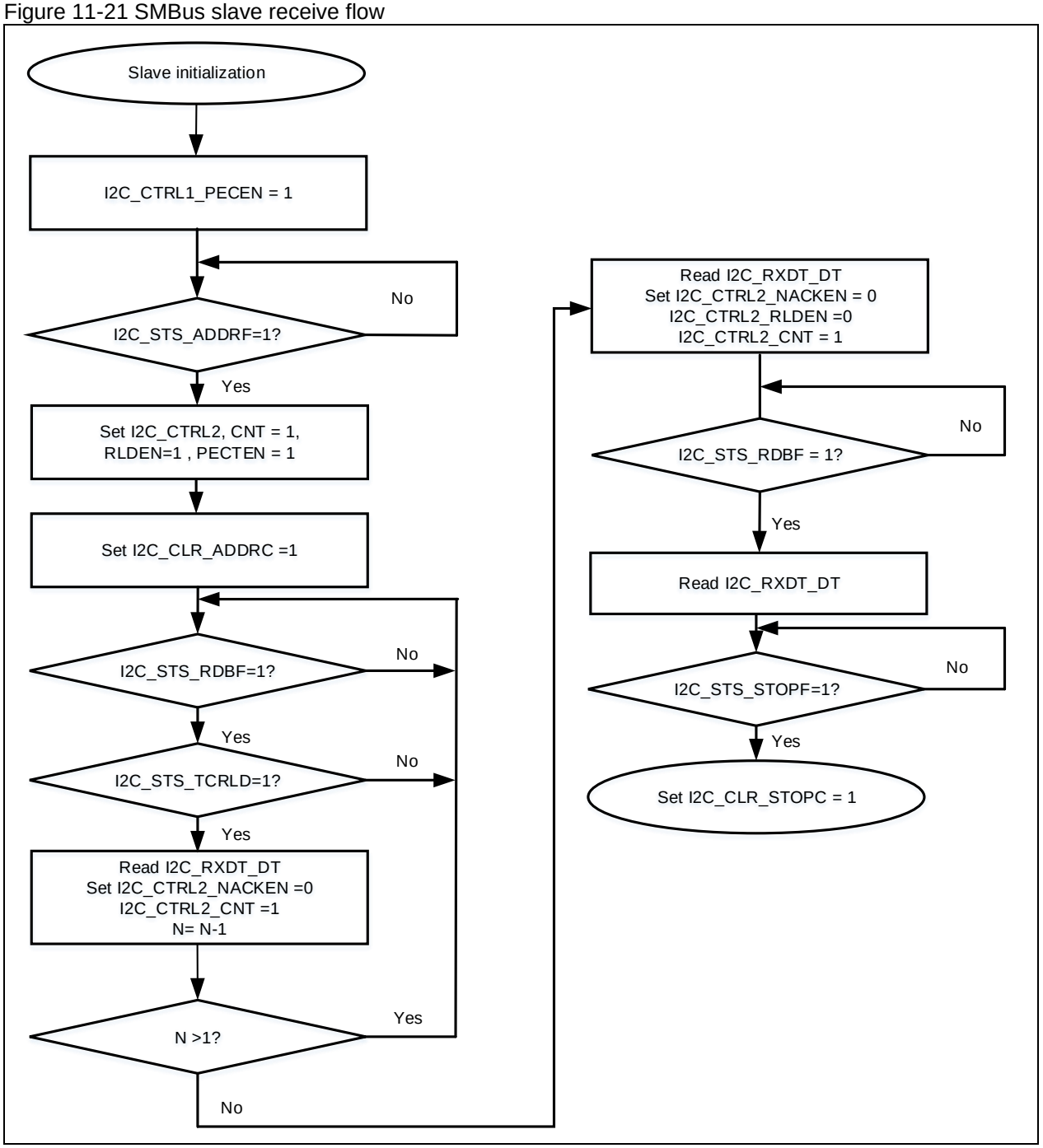
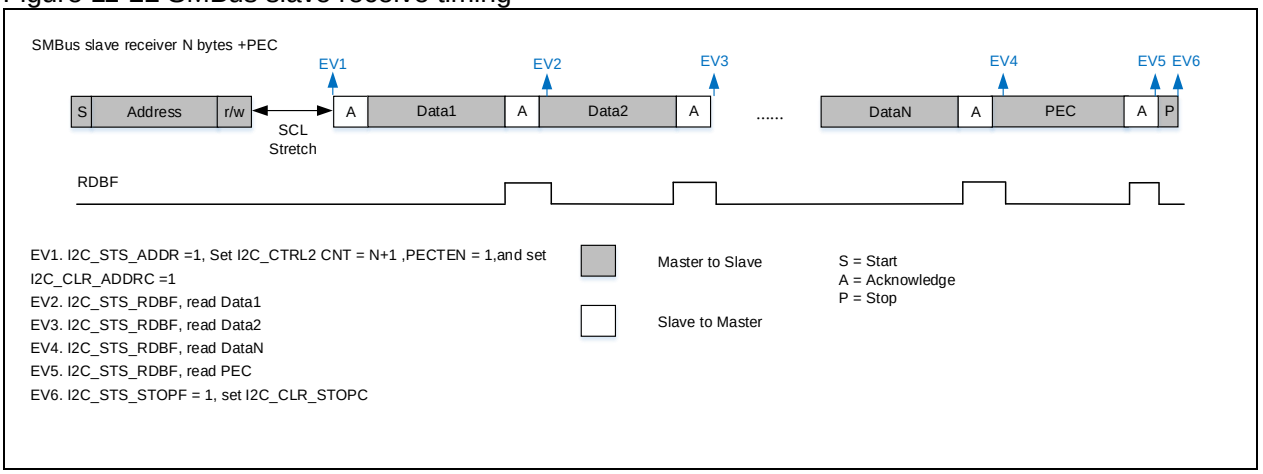


Figure 11-22 SMBus slave receive timing



11.4.8 Data transfer using DMA

I²C data transfer can be done using DMA controller so as to reduce the burden on the CPU. The TDIEN and RDIEN must be set 0 when using DMA for data transfer.

Transmission using DMA (DMATEN=1)

1. Set the peripheral address (DMA_CxPADDR= I2C_TXDT address)
2. Set the memory address (DMA_CxMADDR=data memory address)
3. The transmission direction is set from memory to peripheral (DTD=1 in the DMA_CHCTRL register)
4. Configure the total number of bytes to be transferred in the DMA_CxDTCNT register
5. Configure other parameters such as priority, memory data width, peripheral data width, interrupts, etc. in the DMA_CHCTRL register
6. Enable the DMA channel by setting CHEN=1 in the DMA_CxCTRL register
7. Enable I2C DMA request by setting DMAEN=1 in the I2C_CTRL2 register. Once the TDBE bit in the I2C_STS1 register is set, the data is loaded from the programmed memory to the I2C_DT register through DMA
8. When the number of data transfers, programmed in the DMA controller, is reached (DMA_CxDTCNT=0), the data transfer is complete (An interrupt is generated if enabled).
9. Master transmitter: Once the TDC flag is set, the STOP condition is generated, indicating that transfer is complete.

   Slave transmitter: Once the ACKFAIL flag is set, clear the ACKFAIL flag, transfer is complete.

Reception using DMA (DMAREN=1)

1. Set the peripheral address (DMA_CxPADDR = I2C_RXDT address)
2. Set the memory address (DMA_CxMADDR = memory address)
3. The transmission directions set from peripheral to memory (DTD=0 in the DMA_CHCTRL register)
4. Configure the total number of bytes to be transferred in the DMA_CxDTCNT register
5. Configure other parameters such as priority, memory data width, peripheral data width, interrupts, etc. in the DMA_CHCTRL register
6. Enable the DMA channel by setting CHEN=1 in the DMA_CxCTRL register
7. Enable I 2C DMA request by setting DMAEN=1 in the I2C_CTRL2 register. Once the RDBE bit in the I2C_STS1 register is set, the data is loaded from the I2C_DT register to the programmed memory through DMA
8. When the number of data transfers, programmed in the DMA controller, is reached (DMA_CxDTCNT=0), the data transfer is complete (An interrupt is generated if enabled).
9. Master receiver: refer to the I²C master communication flow section for STOP condition

   Slave receiver: refer to the I²C slave communication flow section for STOP condition

11.4.9 Error management

The error management feature included in the I²C provides a guarantee for the reliability of communication. The manageable error events are listed below:

Table 11-6 I²C error event

Error event	Event flag	Enable control bit	Clear bit
SMBus Alert	ALERTF	ERRIEN	ALERTC
Timeout error	TMOUT	ERRIEN	TMOUTC
PEC error	PECERR	ERRIEN	PECERRC
Overrun/underrun	OUF	ERRIEN	OUFC
Arbitration lost	ARLOST	ERRIEN	ARLOSTC
Bus error	BUSERR	ERRIEN	BUSERRC

Overrun/Underrun (OUF)

In slave mode, an underrun/overrun may appear if the clock stretching feature is disabled (STRETCH=1 in the I2C_CTRL1 register).

In slave transmit mode: if data has not yet been written to the TXDT register before the transmission of the first bit of the to-be-transferred data (that is, before the generation of SDA edge), an underrun error may occur, and the OUF bit is set in the I2C_STS register, sending 0xFF to the bus.

In slave receive mode: The slave must read the received data in the case of the clock stretching being disabled (STRETCH=1). If one-byte data has been received and data is not read yet before the end of the next data reception, an overrun error occurs, setting OUF=1 in the I2C_STS register, and sending NCAK.

Arbitration lost (ARLOST)

An arbitration lost may occur when the device controls the SDA line to output high level but the actual bus output is low.

- Master transmit: An arbitration may occur during an address transfer and a data transfer
- Master receive: An arbitration may occur during an address transfer and an ACK response
- Slave transmit: An arbitration may occur during a data transfer
- Slave receive: An arbitration may occur during an ACK response

Once an arbitration lost is detected, the ARLOST is set by hardware in the I2C_STS register. The SCL and SDA buses will be released and go automatically back to slave mode.

Bus error (BUSERR)

The SDA line, during a data transfer, must be kept in a stable state when the SCL is in high level. The SDA can be changed only when the SCL signal becomes low, otherwise, a bus error may appear.

When the SCL is high:

- SDA changes from 1 to 0: a misplaced START condition
- SDA changes from 0 to 1: a misplaced STOP condition

Both of these conditions above may trigger a bus error. Once it occurs, the BUSERR is set by hardware in the I2C_STS register.

Packet error checking (PECERR)

The PEC is available only in SMBus mode. In master receive and slave receive modes, a PEC error may appear if the received PEC is not equal to the internally calculated PEC. In this case, the PECERR bit is set by hardware in the I2C_STS register.

In slave receive mode, an NACK is sent when a PEC error is detected.

In master receive mode, an NACK is always sent, whatever the PEC check result.

SMBus alert (ALERTF)

The SMBus alert feature is present when HADDREN=1 (SMBus master mode) and SMBALERT=1 (SMBus alert mode).Once an alert event is detected on the ALERT pin (ALERT pin changes from high to low), the ALERTF bit is set by hardware in the I2C_STS register.

Timeout error (TMOUT)

SMBus defines a timeout mechanism for the improvement of the system stability, preventing the bus from being pulled down in the case of a master or slave failure. Once a timeout event (defined in SMBus chapter) is detected, the TMOUT is set by hardware in the I2C_STS register. If a timeout error occurs in slave mode, the SCL and SDA buses are immediately released; if a timeout error occurs in master mode, a STOP condition is automatically by host to abort the communication

11.4.10 Wakeup from Deepsleep mode at address matching event

I2C1 supports to wake up from Deepsleep mode when the address matches. To enable this function, configure internal registers as below before entering Deepsleep mode:

- Set WAKEUPEN=1 in the I2C1_CTRL1 register
- Set DFLT=0 in the I2C1_CTRL1 register
- Set STRETCH=0 in the I2C1_CTRL1 register
- Select HICK in the I2C1SEL bit in the CRM_PICLKS register

Deepsleep mode wakeup procedure:

1. After completion of the above mentioned configurations, the system enters Deepsleep mode, and at this point, HICK is disabled
2. When the I2C bus enabling condition is detected, I2C interface enable HICK and pull the SCL bus low
3. After the HICK is enabled, start receiving address
   - Address matches: wake up the system, and during the wakeup process, I2C interface keeps pulling SCL bus low until the address match interrupt is handled and ADDRF flag is cleared
   - Address does not match: disable HICK, and the system does not wake up
4. SCL bus is released and enters normal transmission status
   - The I2C is not allowed to enter Deepsleep mode after it is accessed as master transfer data or slave

11.5 I²C interrupt requests

The following table lists all the I²C interrupt requests.

Table 11-7 I²C interrupt requests

Interrupt event	Event flag	Enable control bit
Address matched	ADDRF	ADDRIEN
Acknowledge failure	ACKFAIL	ACKFAILIEN
Stop condition received	STOPF	STOPIEN
Transmit interrupt state	TDIS	TDIEN
Receive data buffer full	RDBF	RDIEN
Transfer complete, wait for loading data	TCRLD	TDCIEN
Data transfer complete	TDC	
SMBus alert	ALERTF	ERRIEN
Timeout error	TMOUT	
PEC error	PECERR	
Overrun/underrun	OUF	
Arbitration lost	ARLOST	
Bus error	BUSERR	

11.6 I²C debug mode

When the microcontroller enters debug mode (Cortex®-M0+ halted), the SMBUS timeout either continues to work or stops, depending on the I2Cx_SMBUS_TIMEOUT configuration bit in the DEBUG module.

11.7 I²C registers

These peripheral registers must be accessed by words (32 bits).

Table 11-8 I²C register map and reset values

Register	Offset	Reset value
I2C_CTRL1	0x00	0x00000000
I2C_CTRL2	0x04	0x00000000
I2C_OADDR1	0x08	0x00000000
I2C_OADDR2	0x0C	0x00000000
I2C_CLKCTRL	0x10	0x00000000
I2C_TIMEOUT	0x14	0x00000000
I2C_STS	0x18	0x00000000
I2C_CLR	0x1C	0x00000000
I2C_PEC	0x20	0x00000000
I2C_RXDT	0x24	0x00000000
I2C_TXDT	0x28	0x00000000

11.7.1 Control register 1 (I2C_CTRL1)

Bit	Name	Reset value	Type	Description
Bit 31:24	Reserved	0x00	res	Kept at its default value.
Bit 23	PECEN	0x0	rw	PEC calculation enable 0: PEC calculation disabled 1: PEC calculation enabled
Bit 22	SMBALERT	0x0	rw	SMBus alert enable / pin set To enable SMBus master alert feature: 0: SMBus alert disabled 1: SMBus alert enabled To enable SMBus slave alert feature: 0: Pin high 1: Pin low, response address 0001100x
Bit 21	DEVADDREN	0x0	rw	SMBus device default address enable 0: SMBus device default address disabled 1: SMBus device default address enabled, response device default address 1100001x.
Bit 20	HADDREN	0x0	rw	SMBus host address enable 0: SMBus host address disabled 1: SMBus host address enabled, response host address 0001000x
Bit 19	GCAEN	0x0	rw	General call address enable 0: General call address disabled 1: General call address enabled, response address 0000000x
Bit 18	WAKEUPEN	0x0	res	Deepsleep mode wakeup enable 0: Disabled 1: Enabled
Bit 17	STRETCH	0x0	rw	Clock stretching mode 0: Clock stretching mode enabled 1: Clock stretching mode disabled
Bit 16	SCTRL	0x0	rw	Slave receiving data control 0: Slave receiving data control disabled 1: Slave receiving data control enabled

Bit 15	DMAREN	0x0	rw	DMA receive data request enable 0: DMA receive data request disabled 1: DMA receive data request enabled
Bit 14	DMATEN	0x0	rw	DMA transmit data request enable 0: DMA transmit data request disabled 1: DMA transmit data request enabled
Bit 13	Reserved	0x0	res	Kept at its default value.
Bit 12	ANGNFOFF	0x0	rw	Analog filter off 0: Enabled 1: Disabled
Bit 11:8	DFLT	0x0	rw	Digital filter value The glitches less than the filter time on the SCL bus will be filtered; filter time= DFLT x T_{I2C_CLK}.
Bit 7	ERRIEN	0x0	rw	Error interrupt enable 0: Error interrupt disabled 1: Error interrupt enabled
Bit 6	TDCIEN	0x0	rw	Transfer data complete interrupt enable 0: Transfer data complete interrupt disabled 1: Transfer data complete interrupt enabled
Bit 5	STOPIEN	0x0	rw	Stop generation complete interrupt enable 0: Stop generation complete interrupt disabled 1: Stop generation complete interrupt enabled
Bit 4	ACKFAILIEN	0x0	rw	Acknowledge fail interrupt enable 0: Acknowledge fail interrupt disabled 1: Acknowledge fail interrupt enabled
Bit 3	ADDRIEN	0x0	rw	Address match interrupt enable 0: Address match interrupt disabled 1: Address match interrupt enabled
Bit 2	RDIEN	0x0	rw	Receive data interrupt enable 0: Receive data interrupt disabled 1: Receive data interrupt enabled
Bit 1	TDIEN	0x0	rw	Transmit data interrupt enable 0: Transmit data interrupt disabled 1: Transmit data interrupt enabled
Bit 0	I2CEN	0x0	rw	I^2C peripheral enable 0: Disabled 1: Enabled

11.7.2 Control register 2 (I2C_CTRL2)

Bit	Name	Reset value	Type	Description
Bit 31:27	Reserved	0x00	res	Kept at its default value.
Bit 26	PECTEN	0x0	rw	Request PEC transmission enable 0: Transmission disabled 1: Transmission enabled
Bit 25	ASTOPEN	0x0	rw	Automatically send stop condition enable 0: Disabled (Software sends STOP condition) 1: Enabled (Automatically send STOP condition)
Bit 24	RLDEN	0x0	rw	Send data reload mode enable 0: Send data reload mode disabled 1: Send data reload mode enabled
Bit 23:16	CNT[7:0]	0x00	rw	Transmit data counter
Bit 15	NACKEN	0x0	rw	Not acknowledge enable 0: Acknowledge enabled 1: Acknowledge disabled
Bit 14	GENSTOP	0x0	rw	Generate stop condition 0: No stop generation 1: Stop generation
Bit 13	GENSTART	0x0	rw	Generate start condition 0: No start generation 1: Start generation
Bit 12	READH10	0x0	rw	10-bit address header read enable 0: 10-bit address header read disabled 1: 10-bit address header read enabled
Bit 11	ADDR10	0x0	rw	Host send 10-bit address mode enable 0: 7-bit address mode 1: 10-bit address mode
Bit 10	DIR	0x0	rw	Master data transmission direction

				0: Receive 1: Transmit
Bit 9:0	SADDR[9:0]	0x000	rw	The slave address sent by the master In 7-bit address mode, BIT0 and BIT[9:8] don't care

11.7.3 Own address register 1 (I2C_OADDR1)

Bit	Name	Reset value	Type	Description
Bit 31:16	Reserved	0x0000	res	Kept at its default value.
Bit 15	ADDR1EN	0x0	rw	Own Address 1 enable 0: Own Address 1 disabled 1: Own Address 1 enabled
Bit 14:11	Reserved	0x0	res	Kept at its default value.
Bit 10	ADDR1MODE	0x0	rw	Own Address mode 0: 7-bit address mode 1: 10-bit address mode
Bit 9:0	ADDR1[9:0]	0x000	rw	Own address 1 In 7-bit address mode, bit 0 and bit [9:8] don't care

11.7.4 Own address register 2 (I2C_OADDR2)

Bit	Name	Reset value	Type	Description
Bit 31:16	Reserved	0x0000	res	Kept at its default value.
Bit 15	ADDR2EN	0x0	rw	Own address 2 enable 0: Own address 2 disabled 1: Own address 2 enabled
Bit 14:11	Reserved	0x0	res	Kept at its default value.
Bit 10:8	ADDR2MASK[2:0]	0x0	rw	Own address 2-bit mask 000: Match address bit [7:1] 001: Match address bit [7:2] 010: Match address bit [7:3] 011: Match address bit [7:4] 100: Match address bit [7:5] 101: Match address bit [7:6] 110: Match address bit [7] 111: Response all addresses other than those reserved for I^2C
Bit 7:1	ADDR2[7:1]	0x00	rw	Own address 2 7-bit address mode
Bit 0	Reserved	0x0	res	Kept at its default value.

11.7.5 Timing register (I2C_CLKCTRL)

Bit	Name	Reset value	Type	Description
Bit 31:28	DIVL[3:0]	0x0	rw	Low 4 bits of clock divider value
Bit 27:24	DIVH[7:4]	0x0	rw	High 4 bits of clock divider value DIV = (DIVH << 4) + DIVL
Bit 23:20	SCLD[3:0]	0x0	rw	SCL output delay T_{SCLD} = (SCLD + 1) x (DIV + 1) x T_{I2C_CLK}
Bit 19:16	SDAD[3:0]	0x0	rw	SDA output delay T_{SDAD} = (SDAD + 1) x (DIV + 1) x T_{I2C_CLK}
Bit 15:8	SCLH[7:0]	0x00	rw	SCL high level T_{SCLH} = (SCLH + 1) x (DIV + 1) x T_{I2C_CLK}
Bit 7:0	SCLL[7:0]	0x00	rw	SCL low level T_{SCLL} = (SCLL + 1) x (DIV + 1) x T_{I2C_CLK}

11.7.6 Timeout register (I2C_TIMEOUT)

Bit	Name	Reset value	Type	Description
Bit 31	EXTEN	0x0	rw	Cumulative clock low extend timeout enable 0: Cumulative clock low extend timeout disabled 1: Cumulative clock low extend timeout enabled Corresponds to $T_{LOW:SEXT}$ / $T_{LOW:MEXT}$ in SMBus
Bit 30:28	Reserved	0x0	res	Kept at its default value.
Bit 27:16	EXTTIME[11:0]	0x000	rw	Cumulative clock low extend timeout value Timeout duration = (EXTTIME + 1) x 2048 x T_{I2C_CLK}
Bit 15	TOEN	0x0	rw	Detect clock low/high timeout enable 0: Clock low/high timeout detection disabled 1: clock low/high timeout detection enabled Corresponds to $T_{TIMEOUT}$ in SMBus
Bit 14:13	Reserved	0x0	res	Kept at its default value.
Bit 12	TOMODE	0x0	rw	Clock timeout detection mode 0: Clock low level detection 1: Clock high level detection
Bit 11:0	TOTIME[11:0]	0x000	rw	Clock timeout detection time For clock low level detection (TOMODE = 0): Timeout duration = (EXTTIME + 1) x 2048 x T_{I2C_CLK} For clock high level detection (TOMODE = 1): Timeout duration = (EXTTIME + 1) x 4 x T_{I2C_CLK}

11.7.7 Status register (I2C_STS)

Bit	Name	Reset value	Type	Description
Bit 31:24	Reserved	0x00	res	Kept at its default value.
Bit 23:17	ADDR[6:0]	0x00	r	Slave address matching value In 7-bit address mode: Slave address received In 10-bit address mode: 10-bit slave address header received
Bit 16	SDIR	0x0	r	Slave data transmit direction 0: Receive data 1 Transmit data
Bit 15	BUSYF	0x0	r	Bus busy flag transmission mode 0: Bus idle 1: Bus busy Once a START condition is detected, this bit is set; Once a STOP condition is detected, this bit is automatically cleared.
Bit 14	Reserved	0x00	res	Kept at its default value.
Bit 13	ALERTF	0x0	r	SMBus alert flag SMBus host: This bit indicates the reception of an alert signal (ALERT pin changes from high to low) 0: No alert signal received 1: Alert signal received SMBus slave: This bit indicates the device default address reception 0001100x 0: No alert signal received 1: Alert signal received
Bit 12	TMOUT	0x0	r	SMBus timeout flag 0: No timeout 1: Timeout
Bit 11	PECERR	0x0	r	PEC receive error flag 0: No PEC error 1: PEC error
Bit 10	OUF	0x0	r	Overflow or underflow flag In transmission mode: 0: No overrun or underrun 1: Underrun In reception mode: 0: No overrun or underrun 1: Overrun
Bit 9	ARLOST	0x0	r	Arbitration lost flag 0: No arbitration lost detected. 1: Arbitration lost detected.
Bit 8	BUSERR	0x0	r	Bus error flag

				0: No Bus error occurred 1: Bus error occurred
Bit 7	TCRLD	0x0	r	Transmission is complete, waiting to load data 0: Data transfer is not complete yet 1: Data transfer is complete This bit is set when data transfer is complete (CNT=1) and reload mode is enabled (RLDEN=1). It is automatically cleared when writing a CNT value. This bit is applicable in master mode or when SCTRL=1 in slave mode
Bit 6	TDC	0x0	r	Data transfer complete flag 0: Data transfer is not completed yet (the shift register still holds data) 1: Data transfer is completed (shift register become empty and all data has been sent to the bus) This bit is set when ASTOPEN 0, RLDEN=0 and CNT=0. It is automatically cleared after a START or a STOP condition is received.
Bit 5	STOPF	0x0	r	Stop condition generation complete flag 0: No Stop condition detected. 1: Stop condition detected.
Bit 4	ACKFAILF	0x0	r	Acknowledge failure flag 0: No acknowledge failure 1: Acknowledge failure
Bit 3	ADDRF	0x0	r	0~7 bit address match flag 0: 0~7 bit address mismatch 1: 0~7 bit address match
Bit 2	RDBF	0x0	r	Receive data buffer full flag 0: Data register has not received data yet 1: Data register has received data
Bit 1	TDIS	0x0	rw1s	Transmit data interrupt status 0: Data has been written to the I2C_TXDT 1: Data has been sent from the I2C_TXDT to the shift register. I2C_TXDT become empty, and thus the to-be transferred data must be written to the I2C_TXDT. When the clock stretching mode is disabled, a TDIS event is generated by writing 1 so that data is written to the I2C_TXDT register in advance.
Bit 0	TDBE	0x0	rw1s	Transmit data buffer empty flag 0: Data has been written to the I2C_TXDT 1: I2C_TXDT empty This bit is only used to indicate the current status of the I2C_TXDT register. The I2C_TXDT register can be cleared by writing 1 through software.

11.7.8 Status clear register (I2C_CLR)

Bit	Name	Reset value	Type	Description
Bit 31:14	Reserved	0x00000	res	Kept at its default value.
Bit 13	ALERTC	0x0	w	Clear SMBus alert flag SMBus alert flag is cleared by writing 1.
Bit 12	TMOUTC	0x0	w	Clear SMBus timeout flag SMBus timeout flag is cleared by writing 1.
Bit 11	PECERRC	0x0	w	Clear PEC receive error flag PEC receive error flag is cleared by writing 1.
Bit 10	OUFC	0x0	w	Clear overload / underload flag The overload / underload flag is cleared by writing 1.
Bit 9	ARLOSTC	0x0	w	Clear arbitration lost flag The arbitration lost flag is cleared by writing 1.
Bit 8	BUSERRC	0x0	w	Clear bus error flag The bus error flag is cleared by writing 1.
Bit 7:6	Reserved	0x0	res	Kept at its default value.
Bit 5	STOPC	0x0	w	Clear stop condition generation complete flag The stop condition generation complete flag is cleared by writing 1.
Bit 4	ACKFAILC	0x0	w	Clear acknowledge failure flag The acknowledge failure flag is cleared by writing 1.
Bit 3	ADDRC	0x0	w	Clear 0~7 bit address match flag The 0~7 bit address match flag is cleared by writing 1.
Bit 2:0	Reserved	0x0	res	Kept at its default value.

11.7.9 PEC register (I2C_PEC)

Bit	Name	Reset value	Type	Description
Bit 31:8	Reserved	0x000000	res	Kept at its default value.
Bit 7:0	PECVAL[7:0]	0x00	r	PEC value

11.7.10 Receive data register (I2C_RXDT)

Bit	Name	Reset value	Type	Description
Bit 31:8	Reserved	0x000000	res	Kept at its default value.
Bit 7:0	DT[7:0]	0x00	r	Receive data register

11.7.11 Transmit data register (I2C_TXDT)

Bit	Name	Reset value	Type	Description
Bit 31:8	Reserved	0x000000	res	Kept at its default value.
Bit 7:0	DT[7:0]	0x00	rw	Transmit data register

12 Universal synchronous/asynchronous receiver/transmitter (USART)

12.1 USART introduction

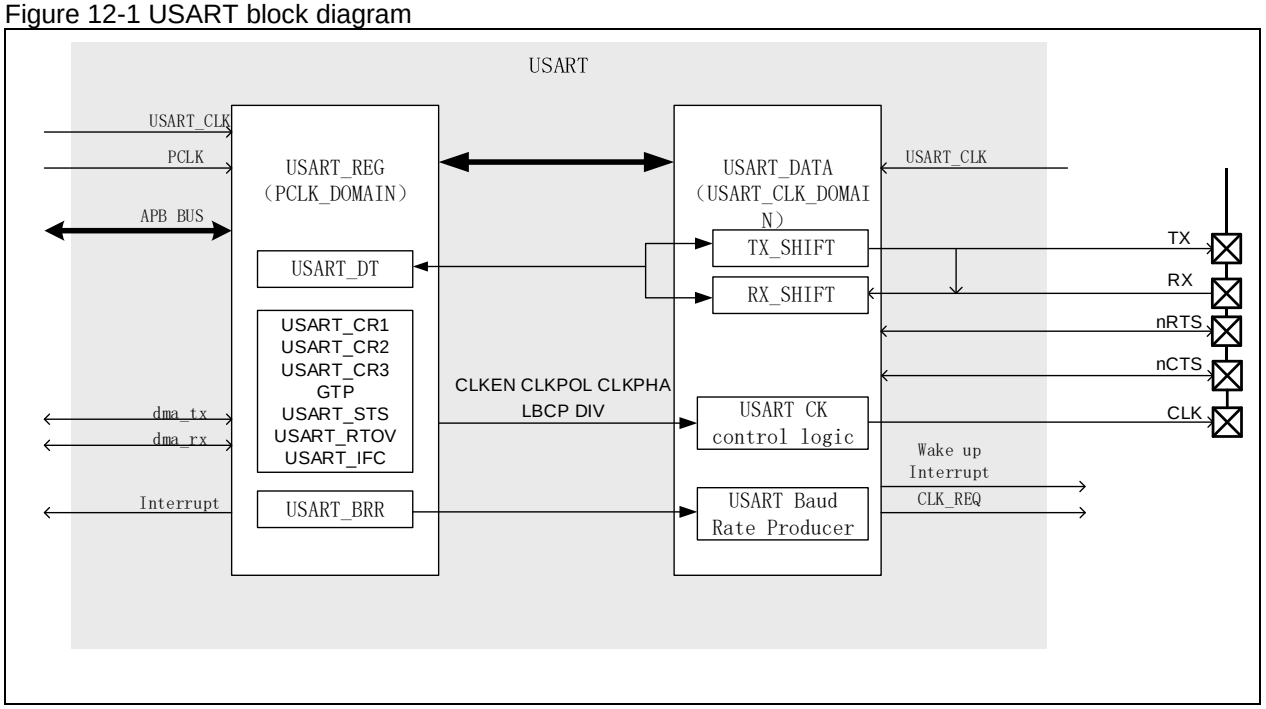
The universal synchronous/asynchronous receiver/transmitter (USART) functions as a general-purpose interface for communications between peripherals. It supports asynchronous full-duplex and half-duplex as well as synchronous transfer. With a programmable baud rate generator, USART offers up to 5 Mbits/s of baud rate by setting the system frequency and frequency divider, which is also convenient for users to configure the required communication frequency.

In addition to standard NRZ asynchronous and synchronous receiver/transmitter communication protocols, USART also supports widely-used serial communication protocols such as LIN (Local Interconnection Network), IrDA (Infrared Data Association) SIRENDEC specification, Asynchronous SmartCard protocol defined in ISO7816-3 standard, CTS/RTS (Clear To Send/Request To Send) hardware flow operation, RS485 and Modbus.

It also allows multi-processor communication, and supports silent mode waken up by idle frames or ID matching to build up a USART network. Meanwhile, high-speed communication is possible by DMA.

It supports dual clock domain. The PCLK is sourced by a divided system clock, and the USART_CLK is clocked by PCLK, HICK or LEXT, which allows USART to work in Deepsleep mode and support low-power mode wakeup.

Figure 12-1 USART block diagram



USART main features:

- Programmable full-duplex or half-duplex communication
  - Full-duplex, asynchronous communication
  - Half-duplex, single-wire communication
- Programmable communication modes
  - NRZ standard format (Mark/Space)
  - LIN (Local Interconnection Network)
  - IrDA SIR
  - Asynchronous SmartCard protocol defined in ISO7816-3 standard: Support 0.5 or 1.5 stop bits in Smartcard mode

  - RS-232 CTS/RTS (Clear To Send/Request To Send) hardware flow operation
  - RS-485
  - Support multi-processor communication using mute mode (waken up by address matching and bus idle)
  - Synchronous mode
- Programmable baud rate generator
  - Shared by transmission and reception, up to 5 Mbits/s
- Programmable frame format
  - Programmable data word length (7 bits, 8 bits or 9 bits)
  - Programmable stop bits-support 1 or 2 stop bits
  - Programmable parity control: transmitter with parity bit transmission capability, and receiver with received data parity check capability
  - Programmable data transmission order (MSB/LSB)
  - Programmable Tx/Rx pin polarity
  - Programmable DT polarity
- Programmable DMA multi-processor communication
- Programmable separate enable bits for transmitter and receiver
- Programmable output CLK phase, polarity and frequency
- Detection flags
  - Receive buffer full
  - Transmit buffer empty
  - Transfer complete flag
- Four error detection flags
  - Overrun error
  - Noise error
  - Framing error
  - Parity error
- Programmable 13 interrupt sources with flags
  - CTSF changes
  - LIN break detection
  - Transmit data register empty
  - Transmission complete
  - Receive data register full
  - Idle bus detected
  - Overrun error
  - Framing error
  - Noise error
  - Parity error
  - Receiver timeout detection
  - Byte match detection
  - Low-power mode wakeup

12.2 Full-duplex/half-duplex selector

The full-duplex and half-duplex selector enables USART to perform data exchanges with peripherals in full-duplex or half-duplex mode by setting the corresponding registers.

In two-wire unidirectional full-duplex mode (by default), TX pin is used for data output, while the RX pin is used for data input. Since the transmitter and receiver are independent of each other, USART is able to send/receive data at the same time so as to achieve full-duplex communication.

When the HALFSEL is set to 1, the single-wire bidirectional half-duplex mode is selected for communication. In this mode, the LINEN, CLKEN, SCMEN and IRDAEN bits must be set 0. RX pin is inactive, while TX and SW_RX are interconnected inside USART. For USART part, TX pins is used for data output, and SW_RX for data input. For peripheral part, bidirectional data transfer is executed through IO mapped by TX pin.

12.3 Mode selector

12.3.1 Introduction

USART mode selector allows USART to work in different operation modes through software configuration so as to enable data exchanges between USART and peripherals in different communication protocols.

USART supports NRZ standard format (Mark/Space) by default. It also supports LIN (Local Interconnection Network), IrDA SIR (Serial Infrared), Asynchronous Smartcard protocol in ISO7816-3 standard, RS-232 CTS/RTS (Clear To Send/Request To Send) hardware flow operation, mute mode and synchronous mode, depending on USART mode selection configuration.

12.3.2 Configuration procedures

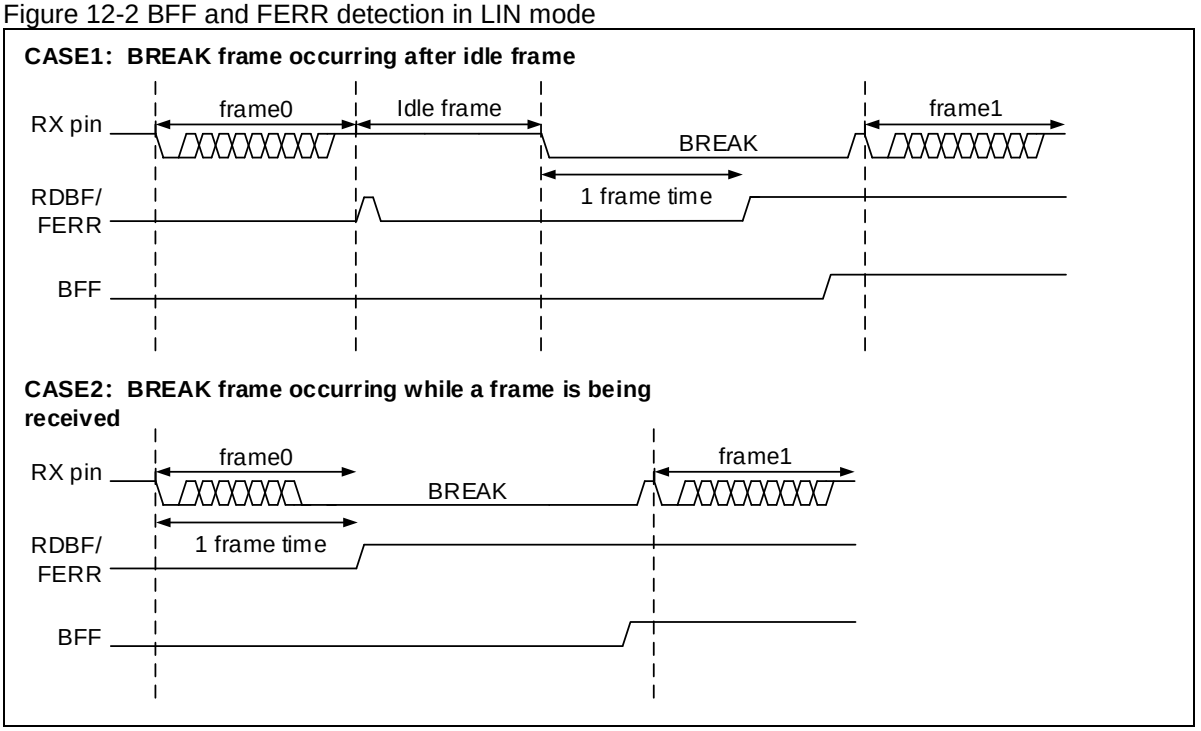
Selection of operation mode is done by following the configuration process below. In addition, such configuration method, along with those of receiver and transmitter described in the subsequent sections, are used to make USART initialization configuration

1. **LIN mode**

   Set LINEN=1, CLKEN=0, STOPBN[1:0]=0, SCMEN=0, SLBEN=0, IRDAEN=0, DBN[1:0]=00.

   LIN master has break generation capability, and can transmit 13-bit low-level LIN synchronous break frame by setting SBF=1. The LIN slave has the same break detection capability, and can select 11-bit or 10-bit break detection by setting BFBN=1 or BFBN=0.

   Figure 12-2 BFF and FERR detection in LIN mode



2. **Smartcard mode**

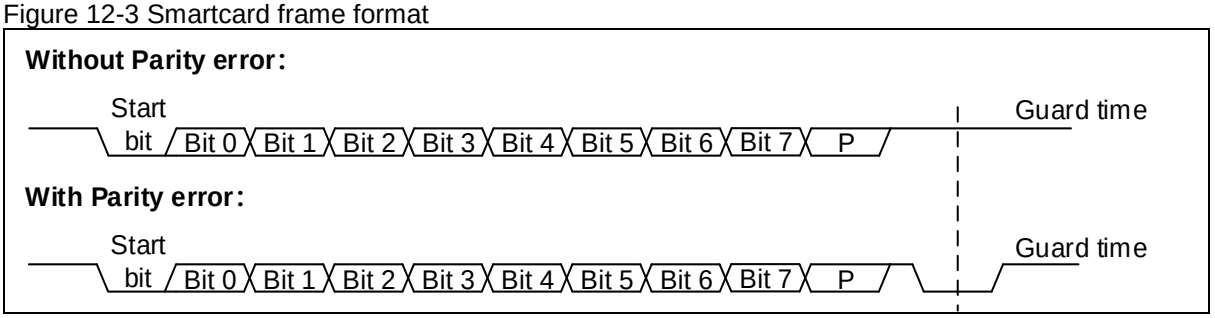
Set SCMEN=1, LINEN=0, SLBEN=0, IRDAEN=0, CLKEN=1, DBN[1:0]=01, PEN=1 and STOPBN[1:0]=11.

The polarity, phase and pulse number of the clock can be configured by setting the CLKPOL, CLKPHA and LBCP bits (Refer to Synchronous mode for details).

The assertion of the TDC flag can be delayed by setting the SCGT[7: 0] bit (guard time bit). The TDF bit can be asserted high after the guard time counter reaches the value programmed in the SCGT[7: 0] bit.

The Smartcard is a single-wire half-duplex communication protocol. The SCNACKEN bit is used to select whether to send NACK when a parity error occurs. This is to indicate to the Smartcard that the data has not been correctly received.

Figure 12-3 Smartcard frame format

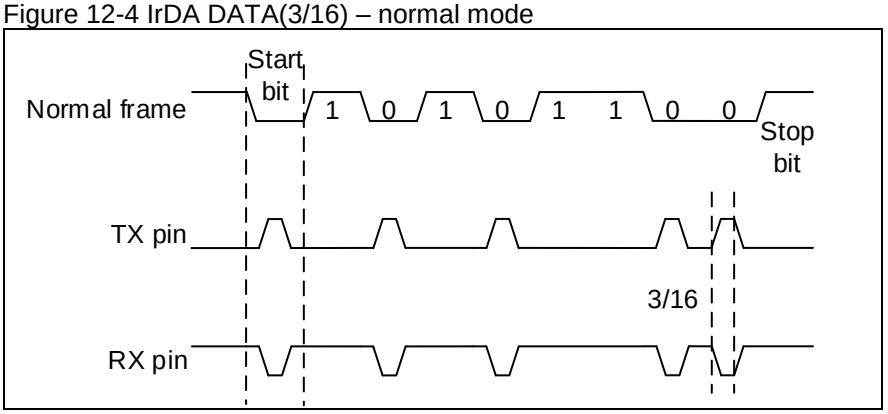


3. **Infrared mode**

Set IRDAEN=1, CLKEN=0, STOPBN[1:0]=0, SCMEN=0 and SLBEN=0.

The infrared low-power mode can be enabled by setting IRDALP=1. In normal mode, the transmitted pulse width is specified as 3/16 bit. In infrared low-power mode, the pulse width can be configurable, and the ISDIV[7:0] bit can be used to achieve the desired low-power frequency.

Figure 12-4 IrDA DATA(3/16) – normal mode



4. **Modbus**

USART only supports basic hardware required by Modbus/RTU and Modbus/ASCII implementation, which means that the control must be done by software (USART provides EOB (end of block) detection only).

In Modbus/RTU, the EOB detection is implemented by the programmable timeout recognizing the receive line idle time being larger than 2 bytes. Users can configure the RTOV register to set the required timeout value (unit: 1 bit width), and enable timeout detection by setting RTODEN=1. When the receive line idle time detected by USART receiver is equal to the programmed timeout value, the USART will set RTODF. An interrupt is generated when RTODIE=1, and the RTODF bit can be cleared by writing 1 to the RTODCF bit.

In Modbus/ASCII, the EOB detection is implemented by the byte match feature recognizing the special byte sequence (CR/LF). Write LF ASCII code to the ID[7:0], and set CMDIE=1 to enable byte match feature. When the data received by USART matches ID[7:0], the USART will set CMDF. An interrupt is generated when CMDIE=1, and the CMDCF bit can be cleared by writing 1 to the CMDF bit.

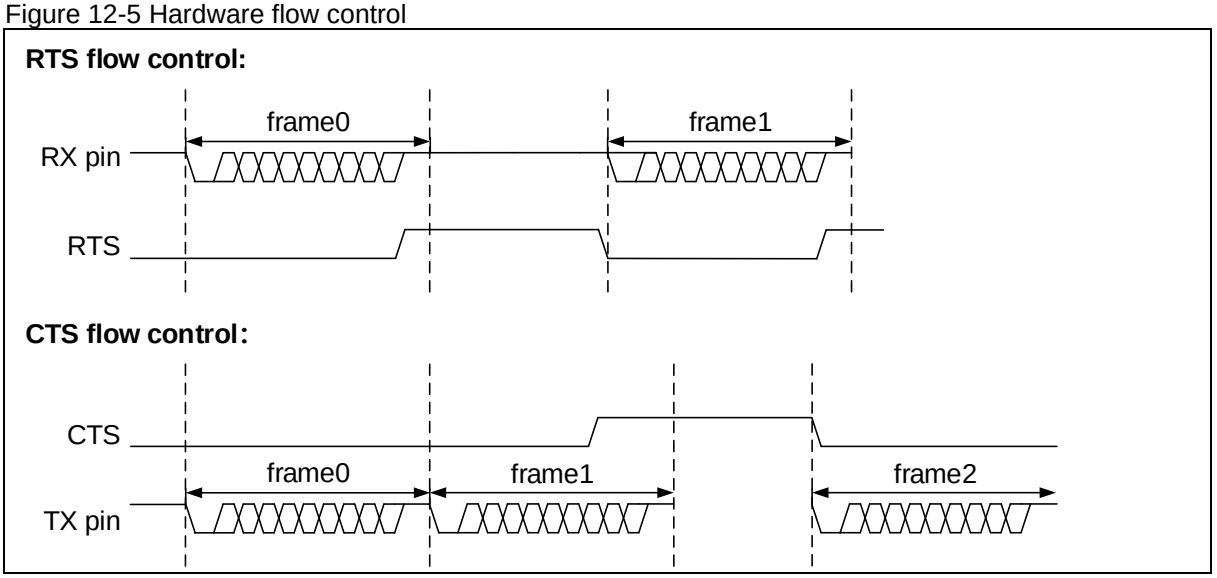
5. **Hardware flow control mode**

Setting the RTSEN and CTSEN bit will enable RTS and CTS flow control, respectively.

RTS flow control: When the USART receiver is ready to receive new data, the RTS becomes effective (pull down low). When the data is received in the receiver (at the beginning of each stop bit), the RTS bit is set, indicating the data transmission is to be stopped at the end of the current frame.

CTS flow control: USART transmitter checks CTS input before transmitting the next frame. If CTS is effective (that is, CTS is low), the next data is to be transmitted. If CTS becomes invalid (CTS is high) during transmission, the data transmission will stop after the completion of the current transmission.

Figure 12-5 Hardware flow control



6. **RS485 mode**

This mode is enabled by setting RS485EN=1. The enable signal is output on the RTS pin. The DEP bit is used to select the polarity of the DE signal. The TSDT[4: 0] bit is used to define the latency before the transmission of the start bit on the transmitter side, while the TCDT[4: 0] is used to define the latency before the TC flag is set following the stop bit at the end of the last data.

7. **Mute mode**

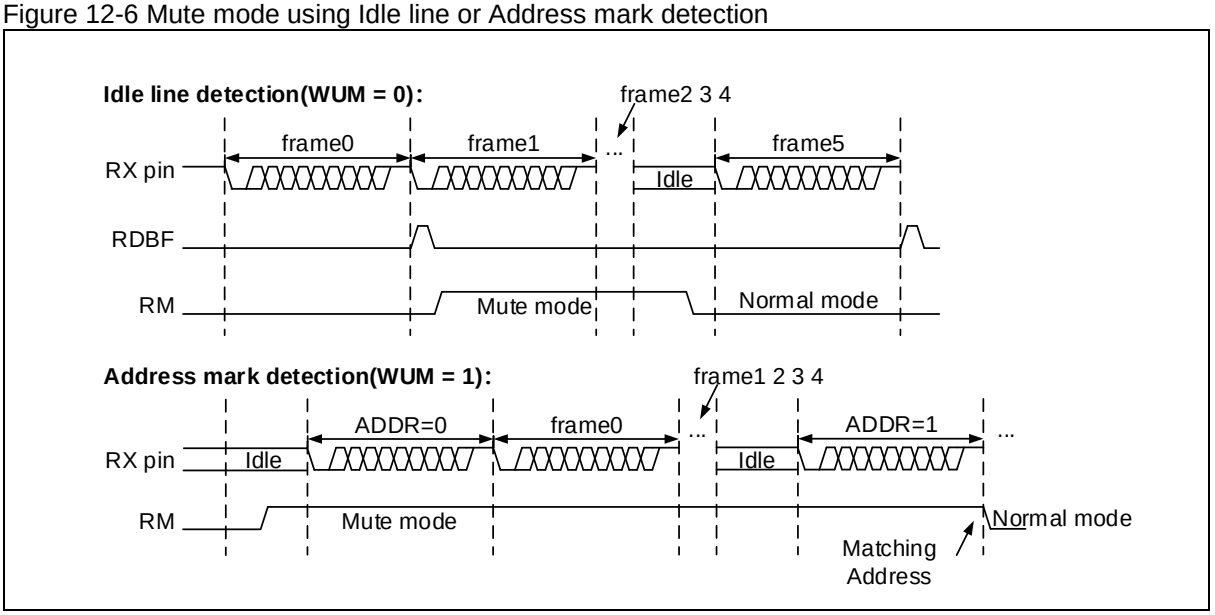
This mode is enabled by setting RM=1. When the WUM bit is set 1 or 0, it wakes up from silent mode through ID match and idle bus, respectively. The ID[7: 0] is configurable. Select ID[7: 0] or ID[3: 0] by setting the IDBN bit. When ID match is selected, if the MSB of data bit is set, it indicates that the current data stands for ID.

When parity check is disabled, if DBN[1:0]=10, the MSB refers to the USART_DT[6]; if DBN[1:0]=00, MSB refers to the USART_DT[7]; if DBN[1:0]=01, MSB refers to the USART_DT[8].

When parity check is enabled, if DBN[1:0]=10, the MSB refers to the USART_DT[5]; if DBN[1:0]=00, MSB refers to the USART_DT[6]; if DBN[1:0]=01, MSB refers to the USART_DT[7].

When the ID[3: 0] bit is selected, the four LSB bits indicate the ID value; When the ID[7: 0] bit is selected, all of the LSB bits indicates the ID value, except for the above parity check bits and MSB bits.

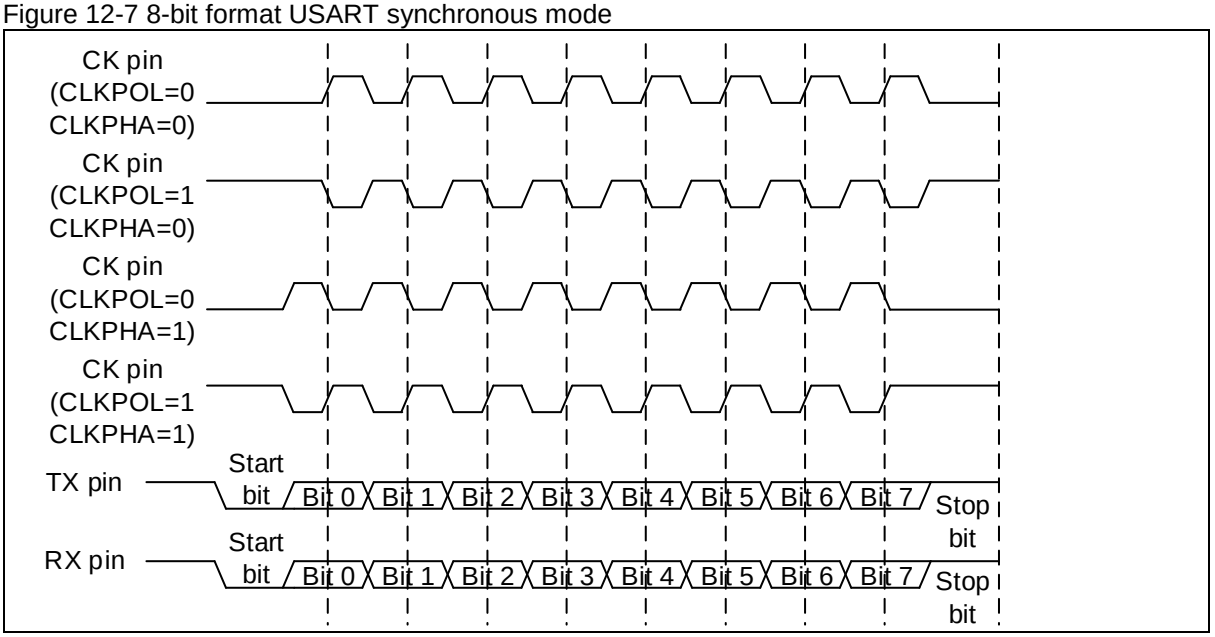
Figure 12-6 Mute mode using Idle line or Address mark detection



8. Synchronous mode

Setting the CLKEN bit enables synchronous mode and clock pin output. Select CK pin high or low in idle state by setting the CLKPOL bit (1 or 0). Whether to sample data on the second or first edge of the clock depends on the CLKPHA bit (1 or 0). The LBCP bit (1 or 0) is used to select whether to output clock on the last data bit, and the ISDIV[4: 0] is used to select the required clock output frequency

Figure 12-7 8-bit format USART synchronous mode



12.4 USART frame format and configuration

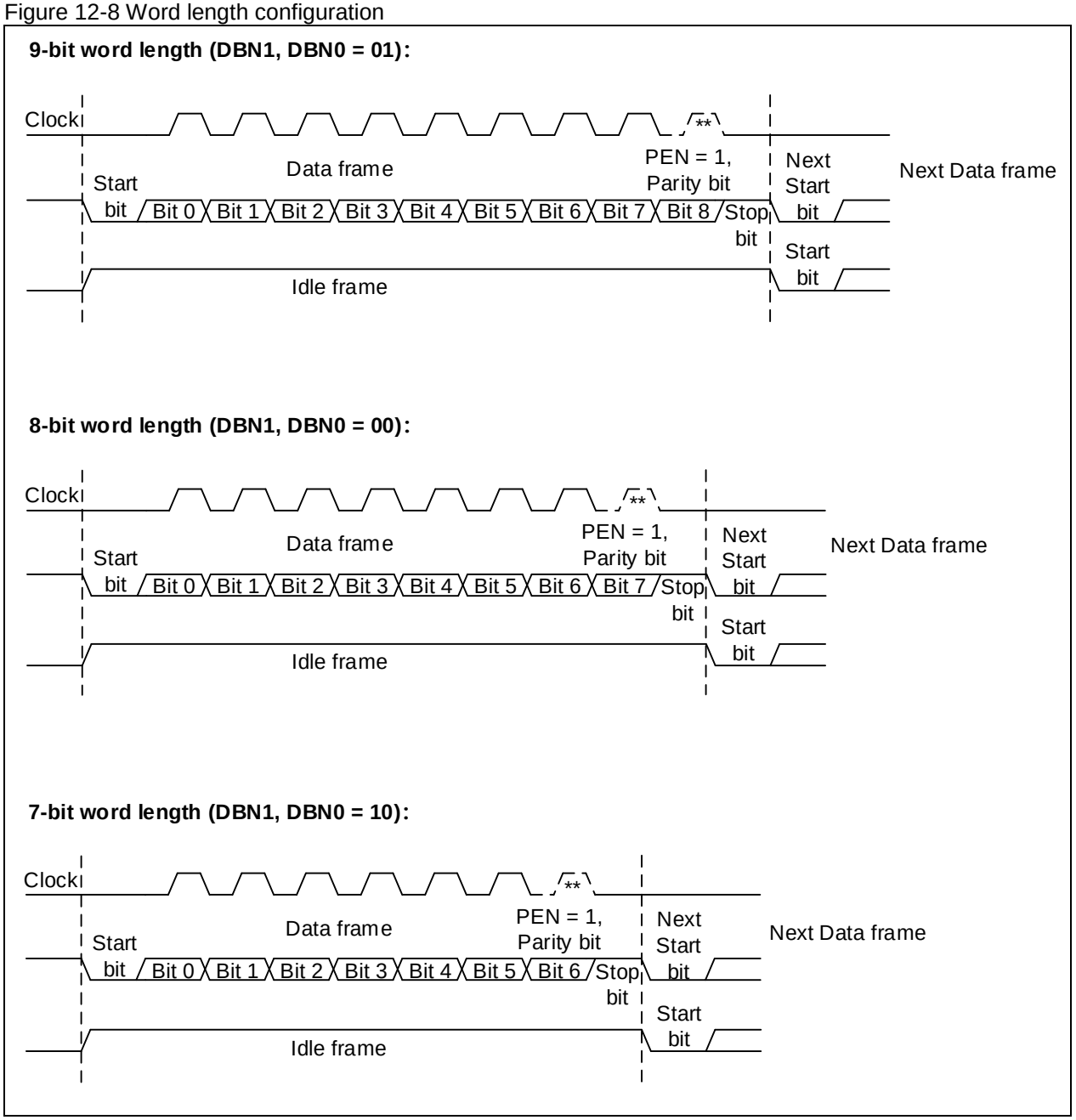
USART data frame consists of start bit, data bit and stop bit, with the last data bit being as a parity bit.

USART idle frame size is equal to that of the data frame under current configuration, but all bits are 1.

USART break frame size is the current data frame size plus its stop bit. All bits before the stop bit are 0.

In non-LIN mode, a break frame transmission and detection must be in line with this rule. For instance, if DBN[1:0]=00, the break frame size for transmission and detection should be 10-bit low level plus its stop bit. In LIN mode, refer to Mode selector and configuration process for more details. The DBN1 and DBN0 bits are used to program 7-bit (DBN[1:0]=10), 8-bit (DBN[1:0]=00) or 9-bit (DBN[1:0]=01) data bits.

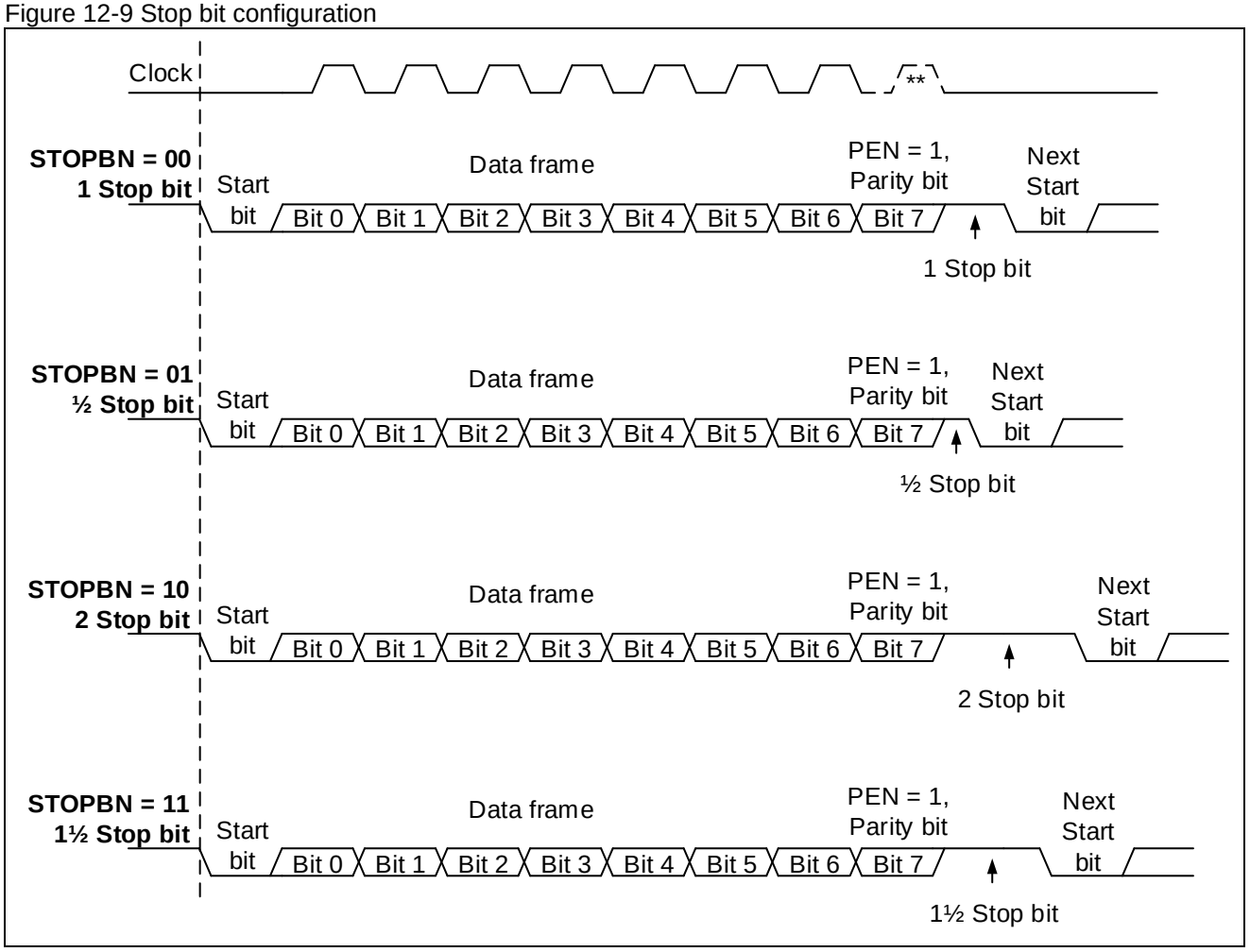
Figure 12-8 Word length configuration



The STOPBN bit is used to program one-bit (STOPBN=00), 0.5-bit (STOPBN=01), two-bit (STOPBN=10) and 1.5-bit (STOPBN=11) stop bits.

Set the PEN bit will enable parity control. PSEL=1 indicates Odd parity, while PSEL=0 for Even parity. Once the parity control is enabled, the MSB of the data bit will be replaced with parity bit, that is, the significant bits is reduced by one bit.

Figure 12-9 Stop bit configuration



Set the MTF bit to determine whether MSB (MTF=1) first or LSB (MTF=0) first.

Set USART_DT as 1=L,0=H (DTREV=1) or 0=L,1=H (DTREV=0) for transmission and reception by setting the DTREV bit.

Set the TXREV bit to select VDD=0/mark,Gnd=1/idle (TXREV=1) or VDD=1/idle,Gnd=0/mark (TXREV=0) for signal transmission on USART_TX pin.

Set the RXREV bit to select VDD=0/mark,Gnd=1/idle (RXREV=1) or VDD=1/idle,Gnd=0/mark (RXREV=0) for signal transmission on USART_RX pin.

12.5 DMA transfer introduction

Enable transmit data buffer and receive data buffer using DMA to achieve continuous high-speed transmission for USART, which is detailed in subsequent sections. For more information on specific DMA configuration, refer to DMA chapter

12.5.1 Transmission using DMA

1. Select a DMA channel: Select a DMA channel from DMA channel map table described in DMA chapter.
2. Configure the destination of DMA transfer: Configure the USART_DT register address as the destination address bit of DMA transfer in the DMA control register. Data will be sent to this address after transmit request is received by DMA.
3. Configure the source of DMA transfer: Configure the memory address as the source of DMA transfer in the DMA control register. Data will be loaded into the USART_DT register from the memory address after transmit request is received by DMA.
4. Configure the total number of bytes to be transferred in the DMA control register.
5. Configure the channel priority of DMA transfer in the DMA control register.
6. Configure DMA interrupt generation after half or full transfer in the DMA control register.
7. Enable DMA transfer channel in the DMA control register.

12.5.2 Reception using DMA

1. Select a DMA transfer channel: Select a DMA channel from DMA channel map table described in DMA chapter.
2. Configure the destination of DMA transfer: Configure the memory address as the destination of DMA transfer in the DMA control register. Data will be loaded from the USART_DT register to the programmed destination after reception request is received by DMA
3. Configure the source of DMA transfer: Configure the USART_DT register address as the source of DMA transfer in the DMA control register. Data will be loaded from the USART_DT register to the programmed destination after reception request is received by DMA.
4. Configure the total number of bytes to be transferred in the DMA control register.
5. Configure the channel priority of DMA transfer in the DMA control register.
6. Configure DMA interrupt generation after half or full transfer in the DMA control register.
7. Enable a DMA transfer channel in the DMA control register.

12.6 Baud rate generation

12.6.1 Introduction

USART baud rate generator uses an internal counter based on PCLK. The DIV (USART_BAUDR [15:0] register) represents the overflow value of the counter. Each time the counter is full, it denotes one-bit data. Thus each data bit width refers to PCLK cycles x DIV. The receiver and transmitter of USART share the same baud rate generator, and the receiver splits each data bit into 16 equal parts to achieve oversampling, so the data bit width should not be less than 16 PCLK periods, that is, the DIV value must be greater than or equal to 16.

12.6.2 Configuration

User can program the desired baud rate by setting different system clocks and writing different values into the USART_BAUDR register. The calculation format is as follows:

$$\frac{\text{TX}}{\text{RX}}\ \text{baud rate} = \frac{f_{CK}}{\text{DIV}}$$

Where, f_{CK} refers to the system clock of USART (PCLK1 for USART2, 3 and USART4, and PCLK2 for USART1).

Note: 1. Write access to the USART_BAUDR register before UEN. The baud rate register value should not be altered when UEN=1

2. When USART receiver or transmitter is disabled, the internal counter will be reset, and baud rate interrupt will occur.

Table 12-1 Error calculation for programmed baud rate

Baud rate		**fPCLK=36MHz**			**fPCLK=72MHz**		
No.	Kbps	Actual	Value programmed in the baud register	Error %	Actual	Value programmed in the baud register	Error %
1	2.4	2.4	15000	0%	2.4	30000	0%
2	9.6	9.6	3750	0%	9.6	7500	0%
3	19.2	19.2	1875	0%	19.2	3750	0%
4	57.6	57.6	625	0%	57.6	1250	0%
5	115.2	115.384	312	0.15%	115.2	625	0%
6	230.4	230.769	156	0.16%	230.769	312	0.16%
7	460.8	461.538	78	0.16%	461.538	156	0.16%
8	921.6	923.076	39	0.16%	923.076	78	0.16%
9	2250	2250	16	0%	2250	32	0%
10	4500	NA	NA	NA	4500	16	0%

If the baud rate is 115.2 Kbps, when fPCLK=36MHz, the baud register should be set as 312(0x138), and the calculated result is 36000000 / 312 = 115384 = 115.384Kbps.
Error: (actual value – theoretical value) / theoretical value * 100%:(115.384 - 115.2) / 115.2 * 100% = 0.15%.

12.7 Transmitter

12.7.1 Transmitter introduction

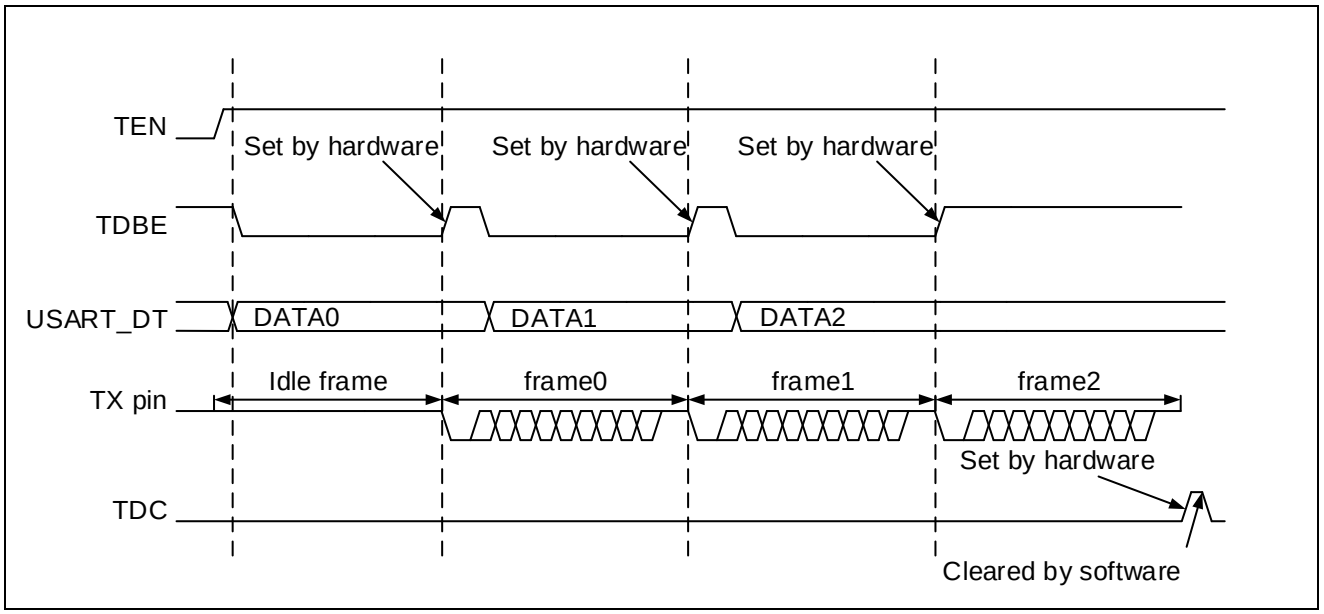
USART transmitter has its individual TEN control bit. The transmitter and receiver share the same baud rate that is programmable. There is a transmit data buffer (TDR) and a transmit shift register in the USART. The TDBE bit is set whenever the TDR is empty, and an interrupt is generated if the TDBEIEN is set. The data written by software is stored in the TDR register. When the shift register is empty, the data will be moved from the TDR register to the shift register so that the data in the transmit shift register is output on the TX pin in LSB mode. The output format depends on the programmed frame format. If synchronous transfer or clock output is selected, the clock pulse is output on the CK pin. If the hardware flow control is selected, the control signal is input on the CTS pin.

Note: 1. The TEN bit cannot be reset during data transfer, or the data on the TX pin will be corrupted.
2. After the TEN bit is enabled, the USART will automatically send an idle frame.

12.7.2 Transmitter configuration

1. USART enable: Set the UEN bit.
2. Full-duplex/half-duplex configuration: Refer to full-duplex/half-duplex selector for more information (Section 12.2).
3. Mode configuration: Refer to mode selector for more information (Section 12.3).
4. Frame format configuration: Refer to frame format for more information (Section 12.4).
5. Interrupt configuration: Refer to interrupt generation for more information (Section 12.11).
6. DMA transmission configuration: If the DMA mode is selected, the DMATEN bit (bit 7 in the USART_CTRL3register) is set, and configure DMA register accordingly.
7. Baud rate configuration: Refer to baud rate generation for details (Section 12.6).
8. Transmitter enable: When the TEN bit is set, the USART transmitter will send an idle frame.
9. Write operation: Wait until the TDBE bit is set, the data to be transferred will be loaded into the USART_DT register (This operation will clear the TDBE bit). Repeat this step in non-DMA mode.
10. After the last data expected to be transferred is written, wait until the TDC is set, indicating the end of transfer. The USART cannot be disabled before the flag is set, or transfer error will occur.
11. When TDC=1, read access to the USART_STS register and write access to the USART_DT register will clear the TDC bit; This bit can also be cleared by writing “0”, but this is valid only in DMA mode.

Figure 12-10 Variations when transmitting TDC/TDBE



Note: There are two USART_CLK cycles (fixed) of idle level between two data during consecutive data transfer (USART). For example, when USART clock = 72 MHz and clock period = 13.88 ns, the data transfer is completed at the stop bit, and the next data will be transmitted through TX pin after an idle period of 13.88 * 2 = 27.76 ns.

12.8 Receiver

12.8.1 Receiver introduction

USART receiver has its individual REN control bit (bit 2 in the USART_CTRL1 register). The transmitter and receiver share the same baud rate that is programmable. There is a receive data buffer (RDR) and a receive shift register in the USART.

The data is input on the RX pin of the USART. When a valid start bit is detected, the receiver ports the data received into the receive shift register in LSB mode. After a full data frame is received, based on the programmed frame format, it will be moved from the receive shift register to the receive data buffer, and the RDBF is set accordingly. An interrupt is generated if the RDBFIEN is set. If hardware flow control is selected, the control signal is output on the RTS pin.

During data reception, the USART receiver will detect whether there are errors to occur, including framing error, overrun error, parity check error or noise error, depending on software configuration, and whether there are interrupts to generate using the interrupt enable bits.

12.8.2 Receiver configuration

Configuration procedure:

1. USART enable: UEN bit is set.
2. Full-duplex/half-duplex configuration: Refer to full-duplex/half-duplex selector for more information (Section 12.2).
3. Mode configuration: Refer to mode selector for more information (Section 12.3).
4. Frame format configuration: Refer to frame format for more information (Section 12.4).
5. Interrupt configuration: Refer to interrupt generation for more information (Section 12.11).
6. Reception using DMA: If the DMA mode is selected, the DMAREN bit is set, and configure DMA register accordingly.
7. Baud rate configuration: Refer to baud rate generation for details (Section 12.6).
8. Receiver enable: REN bit is set

Character reception:

- The RDBF bit is set. It indicates that the content of the shift register is transferred to the RDR (Receiver Data Register). In other words, data is received and can be read (including its associated error flags)
- An interrupt is generated when the RDBFIEN is set.
- The error flag is set when a framing error, noise error or overrun error is detected during reception.
- In DMA mode, the RDNE bit is set after every byte is received, and it is cleared when the data register is read by DMA.
- In non-DMA mode, the RDBF bit is cleared when read access to the USART_DT register by software. The RDBF flag can also be cleared by writing 0 to it. The RDBF bit must be cleared before the end of next frame reception to avoid overrun error.

Break frame reception:

- Non-LIN mode: It is handled as a framing error, and the FERR is set. An interrupt is generated if the corresponding interrupt bit is enabled. Refer to framing error described below for details.
- LIN mode: It is handled as a break frame, and the BFF bit is set. An interrupt is generated if the BFIEN is set.
- USART receiver: It is handled as a data frame, and the IDLEF bit is set. An interrupt is generated if the IDLEIEN is set.

When a framing error occurs:

- The FERR bit is set.
- The USART receiver moves the invalid data from the receive shift register to the receive data buffer.
- In non-DMA mode, both FERR and RDBF are set at the same time. The latter will generate an interrupt. In DMA mode, an interrupt is generated if the ERRIEN.

When an overrun error occurs:

- The ROERR bit is set.
- The data in the receive data buffer is not lost. The previous data is still available when the USART_DT register is read.
- The content in the receive shift register is overwritten. Afterwards, any data received will be lost.
- An interrupt is generated if the RDBFIEN is set or both ERRIEN and DMAREN are set.

- The ROERR bit is cleared by reading the USART_STS register and then USART_DT register in order

Note: If ROERR is set, it indicates that at least one piece of data is lost, with two possibilities:
- *If RDBF=1, it indicates that the last valid data is still stored in the receive data buffer, and can be read.*
- *If RDBF=0, it indicates that the last valid data in the receive data buffer has already been read.*

Note: The REN bit cannot be reset during data reception, or the byte that is currently being received will be lost.

12.8.3 Start bit and noise detection

A start bit detection occurs when the REN bit is set. With the oversampling techniques, the USART receiver samples data on the 3rd, 5th, 7th, 8th, 9th and 10th bits to detect the valid start bit and noise.

The table below shows the data sampling over start bit and noise detection.

Table 12-2 Data sampling over start bit and noise detection

Sampled value (3·5·7)	Sampled value (8·9·10)	NERR bit	Start bit validity
000	000	0	Valid
001/010/100	001/010/100	1	Valid
001/010/100	000	1	Valid
000	001/010/100	1	Valid
111/110/101/011	Any value	0	Valid
Any value	111/110/101/011	0	Valid

Note: If the sampling values on the 3rd, 5th, 7th, 8th, 9th, and 10th bits do not match the above mentioned requirements, the USART receiver does not think that a correct start bit is received, and thus it will abort the start bit detection and return to idle state waiting for a falling edge.

The USART receiver has the ability to detect noise. In the non-synchronous mode, the USART receiver samples data on the 7th, 8th and 9th bits, with its oversampling techniques, to distinguish valid data input from noise based on different sampling values, and recover data as well as set NERR (Noise Error Flag) bit.

Table 12-3 Data sampling over valid data and noise detection

Sampled value	NERR bit	Received bit value	Data validity
000	0	0	Valid
001	1	0	Invalid
010	1	0	Invalid
011	1	1	Invalid
100	1	0	Invalid
101	1	1	Invalid
110	1	1	Invalid
111	0	1	Valid

USART is able to receive data under the maximum allowable deviation condition. Its value depends on the DBN[1:0] bit of the USART_CTRL1 register and the DIV[3: 0] of the USART_BAUDR register.

Note: The maximum allowable deviations stated in the table below are calculated based on 115.2Kbps. The actual deviations may vary with the settings of baud rate. In other words, the greater the baud rate is, the smaller the maximum allowable deviation; in contrast, when the baud rate gets smaller, the maximum allowable deviation will get bigger.

Table 12-4 Maximum allowable deviation

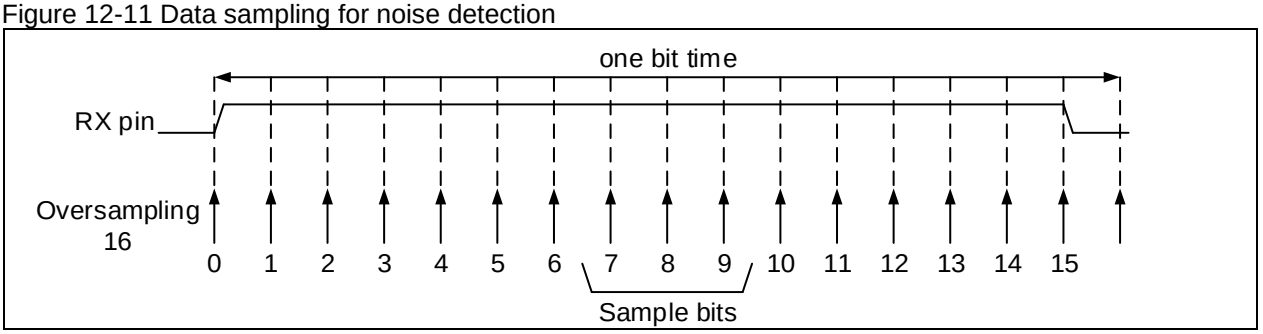
DBN[1:0]	DIV[3:0] = 0	DIV[3:0] != 0
00	3.75%	3.33%
01	3.41%	3.03%
10	4.16%	3.7%

When noise is detected in a data frame:

- The NERR bit is set at the same time as the RDBF bit
- The invalid data is transferred from the receive shift register to the receive data buffer.
- No interrupt is generated in non-DMA mode. However, since the NERR bit is set at the same time as the RDBF bit, the RDBF bit will generate an interrupt. In DMA mode, an interrupt will be issued if the ERRIEN is set.

The NERR bit is cleared by read access to USART_STS register followed by the USART_DT read operation.

Figure 12-11 Data sampling for noise detection



12.9 Low-power wakeup

USART supports low-power wakeup. Before entering DEEPSLEEP mode, the software should guarantee that the USART_CLK is clocked by HICK and LEXT, confirm no transmission by checking the OCCUPY bit, verify the completion of USART receiver initialization by checking the RXON bit, and finally set SMUSEN=1 to enable USART in DEEPSLEEP mode.

After entering the DEEPSLEEP mode, the USART_CLK is disabled, and USART detects the falling edge on the receiver line. Once a falling edge is detected, USART will request MCU to enable USART_CLK (in enabled state until the USART returns to Idle state). If a wakeup source is detected in this process, USART will generate an interrupt to wake up MCU; if no wakeup source is detected, USART will request MCU to disable USART_CLK and wait for the next falling edge).

USART supports three wakeup modes depending on the LPWUM[1:0], i.e., ID match (LPWUM=00), start bit (LPWUM=10) and RDBF flag (LPWUM=11). If the programmed wakeup source is detected in DEEPSLEEP mode, the LPWUF bit is set. Setting the LPWUFIE bit generates an interrupt. It should be noted that this interrupt is only valid in DEEPSLEEP mode. In addition, if the RDBF flag is selected for wakeup, setting RDBFIE can enable an interrupt.

The system clock is disabled after entering the DEEPSLEEP; therefore, the software needs to configure the wakeup mode and set the corresponding interrupt enable bits in advance.

When USART in silent mode enters DEEPSLEEP mode:

1. *Do not use idle bus for silent mode wakeup.*
2. *If the ID match is selected for silent mode wakeup, the MCU low-power mode wakeup mode should be ID match mode. If the RDBF bit is set before entering DEEPSLEEP mode, for the ID match mode, even if MCU exits DEEPSLEEP mode, the USART is still in silent mode.*
3. *If the Start bit is selected to wake up MCU to exit* DEEPSLEEP *mode, the LPWUF bit is set, while the RDBF bit is not.*

Note: For USART to wake up Deepsleep mode, it is necessary to clear USART flag bits and EXINT's pending flag by software.
When using USART to wake up Deepsleep mode, it is recommended to configure rising edge trigger for EXINT, because if rising edge and falling edge (both edges) are configured for EXINT, it is necessary to clear EXINT pending flag individually.

12.10 Tx/Rx swap

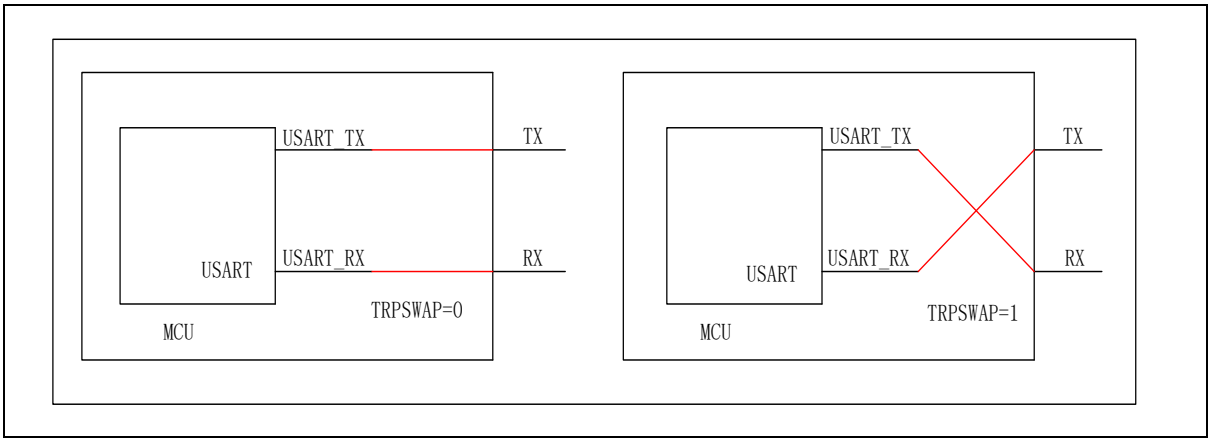
When the TRPSWAP bit (USART_CTRL2[15]) is set, Tx/Rx pin can be swapped. Two common scenes are listed below:

- If the Tx/Rx were reversed while the user attempts to connect the device externally to a RS-

232 chip, they can be swapped through the TRPSWAP bit, without the need of hardware intervention.

- If the user only connected the master Tx to the slave Rx in full-duplex mode, the Tx/Rx can be interchangeable with the TRPSWAP bit , after the master and slave are swapped, without the need of hardware intervention.

Figure 12-12 Tx/Rx swap



Note: The SWAP (USART_CTRL2[15]) can be modified only when the USART is disabled (UEN=0)

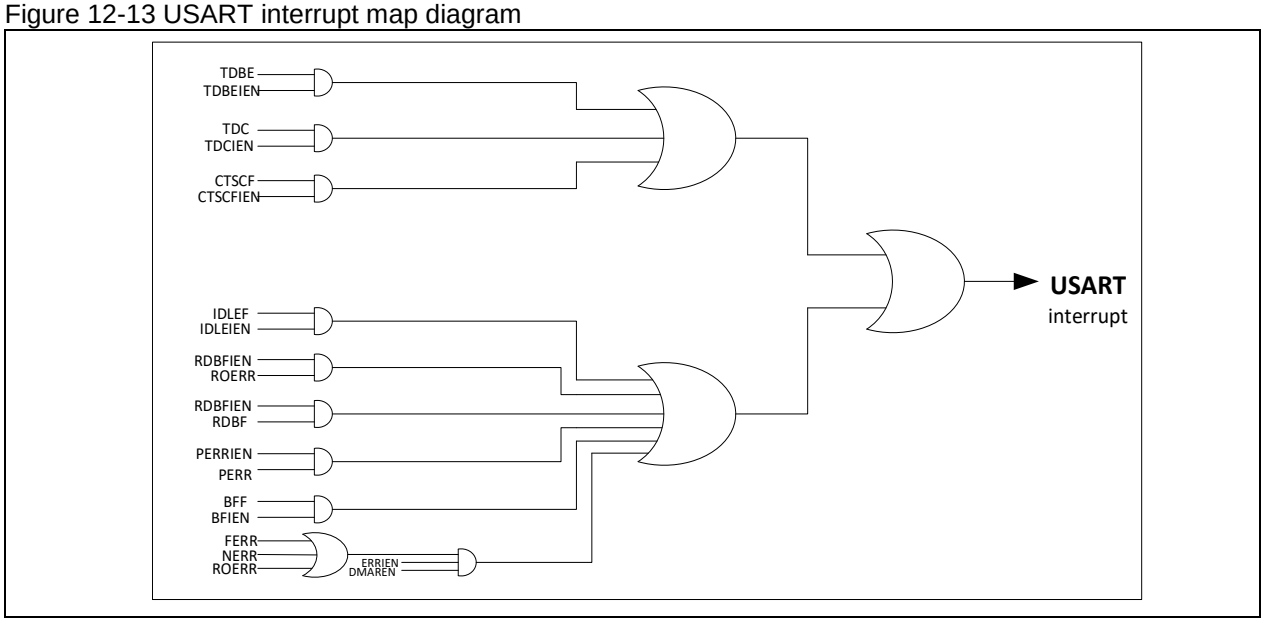
12.11 Interrupt requests

USART interrupt generator serves as a control center of USART interrupts. It is used to monitor the interrupt source inside the USART in real time and the generation of interrupts according to the programmed interrupt control bits. The table below shows the USART interrupt source and interrupt enable control bit. An interrupt will be generated over an event when the corresponding interrupt enable bit is set.

Table 12-5 USART interrupt requests

Interrupt event	Event flag	Enable bit
Transmit data register empty	TDBE	TDBEIEN
CTS flag	CTSCF	CTSCFIEN
Transmit complete	TDC	TDCIEN
Receive data buffer full	RDBF	RDBFIEN
Receive overflow error	ROERR	
Idle flag	IDLEF	IDLEIEN
Parity error	PERR	PERRIEN
Break frame flag	BFF	BFIEN
Noise error, overflow error or framing error	NERR or ROERR or FERR	ERRIEN (1)

Figure 12-13 USART interrupt map diagram



12.12 I/O pin control

The following five interfaces are used for USART communication.

RX: Serial data input.

TX: Serial data output. In single-wire half-duplex and Smartcard mode, the TX pin is used as an I/O for data transmission and reception.

CK: Transmitter clock output. The output CLK phase, polarity and frequency can be programmable.

CTS: Transmitter input. Send enable signal in hardware flow control mode.

RTS: Receiver output. Send request signal in hardware flow control mode.

12.13 USART registers

Table 12-6 USART register map and reset value

Register	Offset	Reset value
USART_STS	0x00	0x00C0
USART_DT	0x04	0x0000
USART_BAUDR	0x08	0x0000
USART_CTRL1	0x0C	0x0000
USART_CTRL2	0x10	0x0000
USART_CTRL3	0x14	0x0000
USART_GDIV	0x18	0x0000
USART_RTOV	0x1C	0x0000
USART_IFC	0x20	0x0000

12.13.1 Status register (USART_STS)

Bit	Name	Reset value	Type	Description
Bit 31:23 Bit 19:18 Bit 15:12	Reserved	0x000000	resd	Forced 0 by hardware.
Bit 22	RXON	0		Receiver enable flag 0: Disabled 1: Enabled Note: USART3 and USART4 do not support this bit (0 by default)
Bit 21	TXON	0		Transmitter enable flag 0: Disabled 1: Enabled Note: USART3 and USART4 do not support this bit (0 by default)
Bit 20	LPWUF	0	r	Low-power mode wakeup flag When a wakeup event is detected, this bit is set. It is cleared by software. 0: No wakeup event detected 1: Wakeup event detected Note: USART3 and USART4 do not support this bit (0 by default)
Bit 17	CMDF	0	r	Byte match detection flag This bit is set by hardware when the byte defined by ID[7:0] is received. It is cleared by software. 0: No byte received 1: Byte received
Bit 16	OCCUPY	0		Receiver occupied flag 0: Not occupied 1: Occupied Note: USART3 and USART4 do not support this bit (0 by default)
Bit 11	RTODF	0	r	Receiver timeout detection flag This bit is set by hardware when the timeout value reaches the programmed value in RTOV register and without any communication. It is cleared by software. 0: No timeout detected 1: Timeout detected
Bit 10	Reserved	0	resd	Forced 0 by hardware.
Bit 9	CTSCF	0x0	rw0c	CTS change flag This bit is set by hardware when the CTS status line changes. It is cleared by software. 0: No change on the CTS status line 1: A change occurs on the CTS status line
Bit 8	BFF	0x0	rw0c	Break frame flag This bit is set by hardware when a break frame is

				detected. It is cleared by software. 0: Break frame is not detected. 1: Break frame is detected.
Bit 7	TDBE	0x1	ro	Transmit data buffer empty This bit is set by hardware when the transmit data buffer is empty. It is cleared by a USART_DT register write operation. 0: Data is not transferred to the shift register. 1: Data is transferred to the shift register.
Bit 6	TDC	0x1	rw0c	Transmit data complete This bit is set by hardware at the end of transmission. It is cleared by software. (Option 1: read access to USART_STS register followed by a USART_DT write operation; Option 2: Write “0” to this bit) 0: Transmission is not completed. 1: Transmission is completed.
Bit 5	RDBF	0x0	rw0c	Receive data buffer full This bit is set by hardware when the data is transferred from the shift register to the USART_DT register. It is cleared by software. (Option 1: read USART_DT register; Option 2: write “0” to this bit) 0: Data is not received. 1: Data is received.
Bit 4	IDLEF	0x0	ro	Idle flag This bit is set by hardware when an idle line is detected. It is cleared by software. (Read USART_DT register followed by a USART_DT read operation) 0: No idle line is detected. 1: Idle line is detected
Bit 3	ROERR	0x0	ro	Receiver overflow error This bit is set by hardware when the data is received while the RDNE is still set. It is cleared by software. (Read USART_STS register followed by a USART_DT read operation) 0: No overflow error 1: Overflow error is detected. Note: When this bit is set, the DT register content will not be lost, but the subsequent data will be overwritten.
Bit 2	NERR	0x0	ro	Noise error This bit is set by hardware when noise is detect on a received frame. It is cleared by software. (Read USART_STS register followed by a USART_DT read operation) 0: No noise is detected 1: Noise is detected
Bit 1	FERR	0x0	ro	Framing error This bit is set by hardware when a stop bit error (low), excessive noise or break frame is detected. It is cleared by software. USART_STS register followed by a USART_DT read operation) 0: No framing error is detected 1: Framing error is detected
Bit 0	PERR	0x0	ro	Parity error This bit is set by hardware when parity error occurs. It is cleared by software. USART_STS register followed by a USART_DT read operation) 0: No parity error occurs 1: Parity error occurs

12.13.2 Data register (USART_DT)

Bit	Name	Reset value	Type	Description
Bit 31:9	Reserved	0x000000	resd	Forced to 0 by hardware.
Bit 8:0	DT	0x000	rw	Data value This register provides read and write function. When transmitting with the parity bit enabled, the value written in the MSB bit will be replaced by the parity bit. When receiving with the parity bit enabled, the value in the MSB bit is the received parity bit.

12.13.3 Baud rate register (USART_BAUDR)

Note: If the TE and RE are disabled respectively, the baud counter stops counting.

Bit	Name	Reset value	Type	Description
Bit 31:16	Reserved	0x0000	resd	Forced to 0 by hardware.
Bit 15:0	DIV	0x0000	rw	Divider This field defines the USART divider.

12.13.4 Control register 1 (USART_CTRL1)

Bit	Name	Reset value	Type	Description
Bit 31:29	Reserved	0x0	resd	Forced to 0 by hardware.
Bit 28	DBN1	0x0	rw	Data bit num This bit, along with the DBN0 bit, is used to program the number of data bits. 10: 7 data bits 00: 8 data bits 01: 9 data bits 11: Write operation forbidden.
Bit 27	RTODEN	0	rw	Receiver time out detection enable 0: Disabled 1: Enabled
Bit 26	RETODIE	0	rw	Receiver time out detection interrupt enable 0: Disabled 1: Enabled
Bit 25:21	TSDT	0x00	rw	Transmit start delay time In RS485 mode, the first data (in sequential transmit mode) is transmitted after a period of time of being written so as to ensure that the transfer direction of the external transmitter/receiver to switch back to transmit. This time depends on the TSDT value, in unit of 1/16 baud rate.
Bit 20:16	TCDT	0x00	rw	Transmit complete delay time In RS485 mode, a period of time (delay) is needed before the last data transfer is complete even if the last STOP bit has been transferred. This time duration allows the transfer direction of the external receiver/transmitter to switch back to receive. This time depends on the TCDT value, in unit of 1/16 baud rate.
Bit 15	Reserved	0	resd	Kept at its default value.
Bit 14	CMDIE	0	rw	Character match detection interrupt enable 0: Disabled 1: Enabled
Bit 13	UEN	0x0	rw	USART enable 0: Disabled 1: Enabled
Bit 12	DBN0	0x0	rw	Data bit num This bit, along with DBN1, is used to program the number of data bits 10: 7 data bits 00: 8 data bits 01: 9 data bits 11: Write operation forbidden.
Bit 11	WUM	0x0	rw	Wakeup mode This bit determines the way to wake up silent mode. 0: Waken up by idle line 1: Waken up by ID match
Bit 10	PEN	0x0	rw	Parity enable This bit is used to enable hardware parity control (generation of parity bit for transmission; detection of parity bit for reception). When this bit is enabled, the MSB bit of the transmitted data is replaced with the parity bit; Check whether the parity bit of the received data is correct. 0: Parity control is disabled 1: Parity control is enabled
Bit 9	PSEL	0x0	rw	Parity selection

				This bit selects the odd or even parity after the parity control is enabled. 0: Even parity 1: Odd parity
Bit 8	PERRIEN	0x0	rw	PERR interrupt enable 0: Interrupt is disabled. 1: Interrupt is enabled.
Bit 7	TDBEIEN	0x0	rw	TDBE interrupt enable 0: Interrupt is disabled. 1: Interrupt is enabled.
Bit 6	TDCIEN	0x0	rw	TDC interrupt enable 0: Interrupt is disabled. 1: Interrupt is enabled.
Bit 5	RDBFIEN	0x0	rw	RDBF interrupt enable 0: Interrupt is disabled. 1: Interrupt is enabled.
Bit 4	IDLEIEN	0x0	rw	IDLE interrupt enable 0: Interrupt is disabled. 1: Interrupt is enabled.
Bit 3	TEN	0x0	rw	Transmitter enable This bit enables the transmitter. 0: Transmitter is disabled 1: Transmitter is enabled
Bit 2	REN	0x0	rw	Receiver enable This bit enables the receiver. 0: Receiver is disabled. 1: Receiver is enabled.
Bit 1	RM	0x0	rw	Receiver mute This bit determines if the receiver is in mute mode or not. It is set or cleared by software. When the idle line is used to wake up from mute mode, this bit is cleared by hardware after wake up. When the address match is used to wake up from mute mode, it is cleared by hardware after wake up. When address mismatches, this bit is set by hardware to enter mute mode again. 0: Receiver is in active mode 1: Receiver is in mute mode
Bit 0	SBF	0x0	rw	Send break frame This bit is used to send a break frame. It can be set or cleared by software. Generally speaking, it is set by software and cleared by hardware at the end of break frame transmission. 0: No break frame is transmitted. 1: Break frame is transmitted.

12.13.5 Control register 2 (USART_CTRL2)

Bit	Name	Reset value	Type	Description
Bit 31:28	IDH	0x0	rw	USART identification This field holds the upper four bits of USART ID. It is configurable
Bit 27:20	Reserved	0x000	resd	Kept at its default value.
Bit 19	MTF	0	rw	MSB transmit first This bit is used to select MSB transmit first or LSB transmit first. 0: LSB first 1: MSB first Note: The parity check is not supported when MTF is enabled.
Bit 18	DTREV	0	rw	DT register polarity reverse 0: 1=H, 0=L 1: 1=L, 0=H
Bit 17	TXREV	0	rw	TX polarity reverse 0: VDD=1/idle,Gnd=0/mark 1: VDD=0/mark,Gnd=1/idle
Bit 16	RXREV	0	rw	RX polarity reverse 0: VDD=1/idle,Gnd=0/mark 1: VDD=0/mark,Gnd=1/idle

Bit 15	TRPSWAP	0x0	rw	Transmit/receive pin swap 0: Transmit/receive pin is not swappable 1: Transmit/receive pin is swappable
Bit 14	LINEN	0x0	rw	LIN mode enable 0: LIN mode is disabled. 1: LIN mode is enabled
Bit 13:12	STOPBN	0x0	rw	STOP bit num These bits are used to program the number of stop bits. 00: 1 stop bit 01: 0.5 stop bit 10: 2 stop bits 11: 1.5 stop bits
Bit 11	CLKEN	0x0	rw	Clock enable This bit is used to enable the clock pin for synchronous mode or Smartcard mode. 0: Clock is disabled. 1: Clock is enabled.
Bit 10	CLKPOL	0x0	rw	Clock polarity In synchronous mode or Smartcard mode, this bit is used to select the polarity of the clock output on the clock pin in idle state. 0: Clock output low 1: Clock output high
Bit 9	CLKPHA	0x0	rw	Clock phase This bit is used to select the phase of the clock output on the clock pin in synchronous mode or Smartcard mode. 0: Data capture is done on the first clock edge. 1: Data capture is done on the second clock edge.
Bit 8	LBCP	0x0	rw	Last bit clock pulse This bit is used to select whether the clock pulse of the last data bit transmitted is output on the clock pin in synchronous mode. 0: The clock pulse of the last data bit is no output on the clock pin. 1: The clock pulse of the last data bit is output on the clock pin.
Bit 7	Reserved	0x0	resd	Kept at its default value.
Bit 6	BFIEN	0x0	rw	Break frame interrupt enable 0: Disabled 1: Enabled
Bit 5	BFBN	0x0	rw	Break frame bit num This bit is used to select 11-bit or 10-bit break frame. 0: 10-bit break frame 1: 11-bit break frame
Bit 4	IDBN	0	rw	Identification bit num This bit is used to select ID bit number. 0: 4 bit 1: Data bit - 1 bit Note: When this bit is set, in 7, 8 or 9-bit data mode, the ID bit number is the lower 6, 7 or 8 bits (5, 6 or 7 bits when parity check is enabled), respectively.
Bit 3:0	IDL	0x0	rw	USART identification This field holds the lower four bits of USART ID. It is configurable.

Note: These three bits (CLKPOL, CLKPHA and LBCP) should not be changed while the transmission is enabled.

12.13.6 Control register 3 (USART_CTRL3)

Bit	Name	Reset value	Type	Description
Bit 31:18 Bit 12	Reserved	0x000000	resd	Forced to 0 by hardware.
Bit 17:16	LPWUM	0x0	rw	Low power wakeup method 00: ID match 01: Reserved 10: Start bit 11: RDBF Note: USART3 and USART4 do not support this bit (0 by

				default)
Bit 15	DEP	0	rw	DE polarity selection 0: High level active 1: Low level active
Bit 14	RS485EN	0	rw	RS485 enable This bit is used to enable RS485 mode. In RS485 mode, the USART controls the transfer direction of the external receiver/transmitter through the DE signal. 0: RS485 mode disabled. The control signal DE output is disabled. RTS pin is used in RS232 mode. 1: RS485 mode enabled. The control signal DE outputs on the RTS pin
Bit 13	LPWUFIE	0	rw	Low power wakeup flag interrupt enable 0: Disabled 1: Enabled Note: USART3 and USART4 do not support this bit (0 by default)
Bit 11	SMUSEN	0	rw	Deepsleep mode USART enable 0: Disabled 1: Enabled Note: USART3 and USART4 do not support this bit (0 by default)
Bit 10	CTSCFIEN	0x0	rw	CTSCF interrupt enable 0: CTSCF interrupt disabled 1: CTSCF interrupt enabled
Bit 9	CTSEN	0x0	rw	CTS enable 0: CTS is disabled. 1: CTS is enabled.
Bit 8	RTSEN	0x0	rw	RTS enable 0: RTS is disabled. 1: RTS is enabled.
Bit 7	DMATEN	0x0	rw	DMA transmitter enable 0: DMA transmitter is disabled. 1: DMA transmitter is enabled.
Bit 6	DMAREN	0x0	rw	DMA receiver enable 0: DMA receiver is disabled. 1: DMA receiver is enabled.
Bit 5	SCMEN	0x0	rw	Smartcard mode enable 0: Smartcard mode is disabled. 1: Smartcard mode is enabled.
Bit 4	SCNACKEN	0x0	rw	Smartcard NACK enable This bit is used to send NACK when parity error occurs. 0: NACK is disabled when parity error occurs. 1: NACK is enabled when parity error occurs.
Bit 3	SLBEN	0x0	rw	Single line bidirectional half-duplex enable 0: Single-wire bidirectional half-duplex is disabled. 1: Single-wire bidirectional half-duplex is enabled.
Bit 2	IRDALP	0x0	rw	IrDA low-power mode This bit is used to configure IrDA low-power mode. 0: IrDA low-power mode is disabled. 1: IrDA low-power mode is enabled.
Bit 1	IRDAEN	0x0	rw	IrDA enable 0: IrDA is disabled. 1: IrDA is enabled.
Bit 0	ERRIEN	0x0	rw	Error interrupt enable An interrupt is generated when a framing error, overflow error or noise error occurs. 0: Error interrupt is disabled. 1: Error interrupt is enabled.

12.13.7 Guard time and divider register (GDIV)

Bit	Name	Reset value	Type	Description
Bit 31:16	Reserved	0x00	resd	Forced to 0 by hardware.
Bit 15:8	SCGT	0x00	rw	Smart card guard time This field specifies the guard time value. The transmission complete flag is set after this guard time in smartcard mode.
Bit 7:0	ISDIV	0x00	rw	IrDA/Smartcard division In IrDA mode: 8 bit [7: 0] is valid. It is valid in common mode and must be set to 00000001. In low-power mode, it divides the peripheral clock to serve as the period base of the pulse width; 00000000: Reserved–Do not write. 00000001: Divided by 1 00000010: Divided by 2 …… Smartcard mode: the lower 5 bit [4: 0] is valid. This division is used to divide the peripheral clock to provide clock for the Smartcard. Configured as follows: 00000: Reserved–Do not write. 00001: Divided by 2 00010: Divided by 4 00011: Divided by 6 ……

12.13.8 Receiver timeout detection register (RTOV)

Bit	Name	Reset value	Type	Description
Bit 31:24	Reserved	0x00	resd	Forced to 0 by hardware.
Bit 23:0	RTOV	0x00	rw	Receiver time out value The unit is 1 bit width.

12.13.9 Interrupt flag clear register (IFC)

Bit	Name	Reset value	Type	Description
Bit 31:21 Bit 19:18 Bit 16:12 Bit 10:0	Reserved	0x00	resd	Forced to 0 by hardware.
Bit 20	LPWUFC	0	w1	Low power wake up flag clear Note: USART3 and USART4 do not support this bit (0 by default)
Bit 17	CMDFC	0	w1	Character match detection flag clear
Bit 11	RTODFC	0	w1	Receiver time out detection flag clear

13 Serial peripheral interface (SPI)

13.1 SPI introduction

The SPI interface supports both SPI protocol and I2S protocol (depending on software configuration). This chapter gives an introduction of the main features and configuration procedures of SPI used as SPI and I2S.

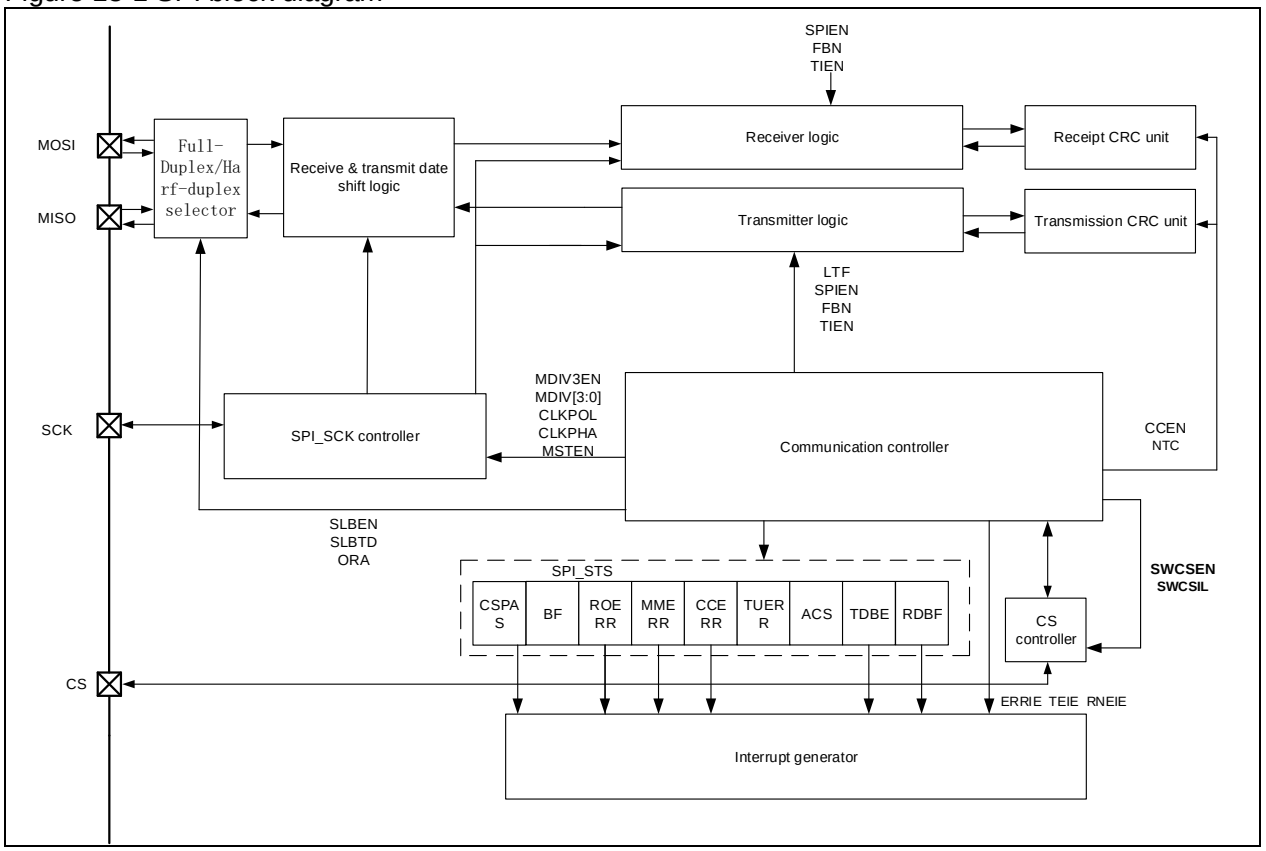
13.2 Function overview

13.2.1 SPI description

The SPI can be configured in host or slave mode depending on software configuration. It is also capable of operating in full-duplex, full-duplex receive-only, half-duplex transmit-only and receive-only modes, with DMA capability and automatic CRC calculation and check function. The SPI interface can be configured by software to be compatible with TI protocol.

SPI block diagram:

Figure 13-1 SPI block diagram



Main features as SPI:

- Programmable full-duplex or half-duplex communication
  - — Full-duplex synchronous communication (supporting full-duplex receive-only mode to free the transmit IO for other purposes)
  - — Half-duplex synchronous communication (transfer direction is configurable by software: receive or transmit)
- Programmable master or slave mode
- Programmable CS signal handling
  - — CS signal handling by hardware
  - — CS signal handling by software
- Programmable 8-bit or 16-bit frame format

- Programmable communication frequency and prescaler (prescaler up to $f_{PCLK}/2$)
- Programmable clock polarity and phase
- Programmable data transfer order (MSB-first or LSB-first)
- Programmable error interrupt flags (CS pulse error, receiver overflow error, master mode error and CRC error)
- Programmable transmit data buffer empty interrupt and receive data buffer full interrupt
- Support transmission and reception using DMA
- Support hardware CRC transmission and check
- Busy status flag
- Compatible with the TI protocol

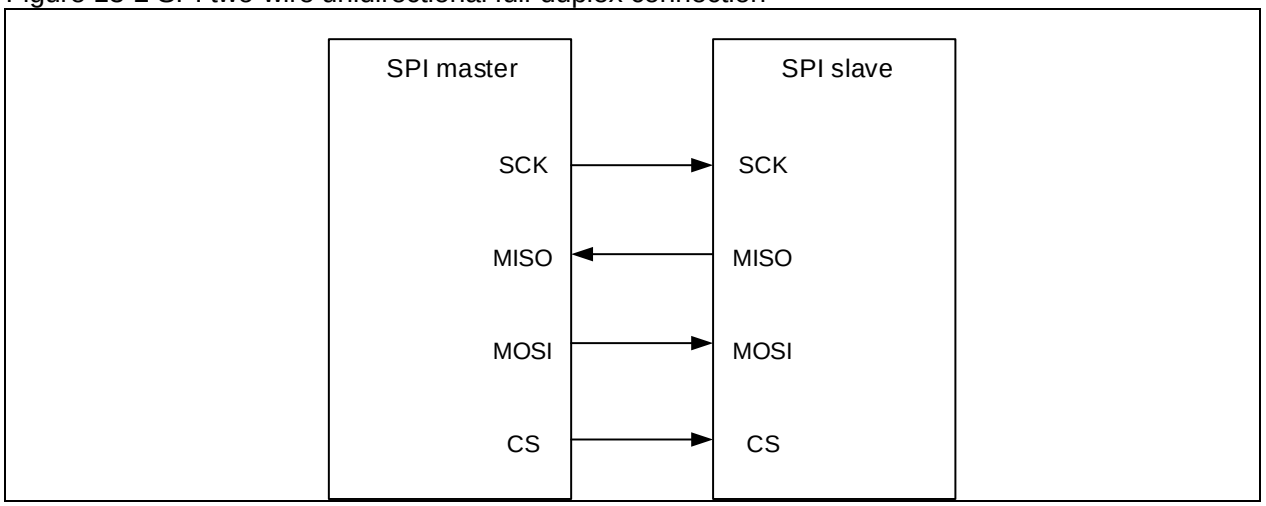
13.2.2 Full-duplex/half-duplex selector

When used as an SPI interface (through software configuration), it is capable of operating in four synchronous modes: two-wire unidirectional full-duplex, single-wire unidirectional receive only, single-wire bidirectional half-duplex transmit and single-wire bidirectional half-duplex receive.

Figure 13-2 shows the two-wire unidirectional full-duplex mode and SPI IO connection:

The SPI operates in two-wire unidirectional full-duplex mode when SLBEN=0 and ORA=0. In this case, the SPI supports simultaneous data transmission and reception. IOs are connected as follows:

Figure 13-2 SPI two-wire unidirectional full-duplex connection



In both master and slave mode, it is required to wait until the RDBF bit and TDBE bit is set, and BF=0 before disabling the SPI or entering power-saving mode (or disabling SPI system clock).

Figure 13-3 shows the single-wire unidirectional receive-only mode and SPI IO connection.

The SPI operates in single-wire unidirectional receive-only mode when SLBEN=0 and ORA=1. In this case, the SPI can be used as data receiver (transmission is not available) only. In master mode, the MISO pin receives data and the IO mapped onto MOSI is released. In slave mode, the MOSI pin receives data and the IO mapped onto MISO is released.

Figure 13-3 Single-wire unidirectional receive only in SPI master mode

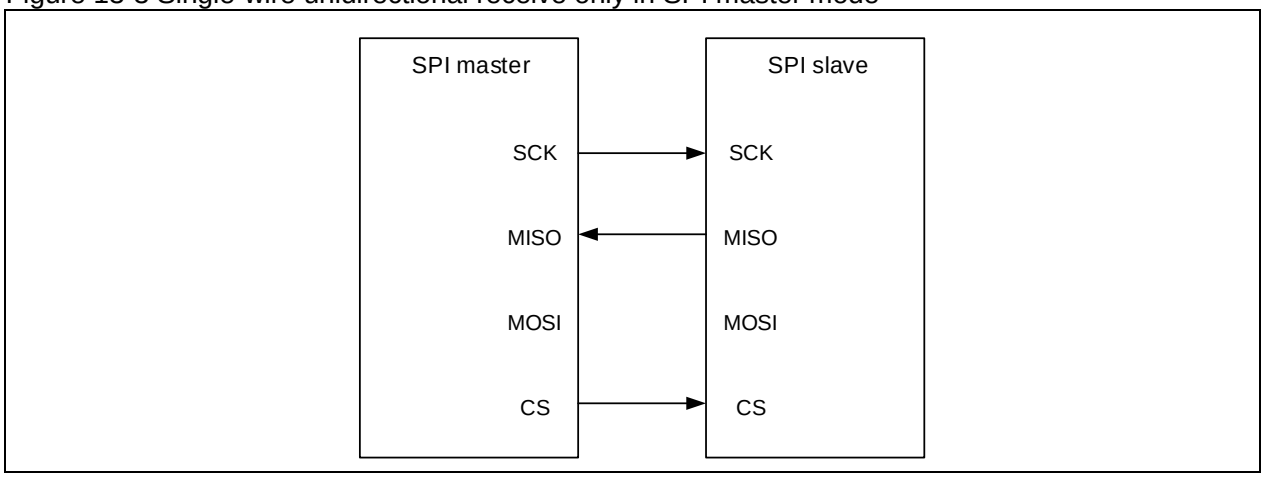
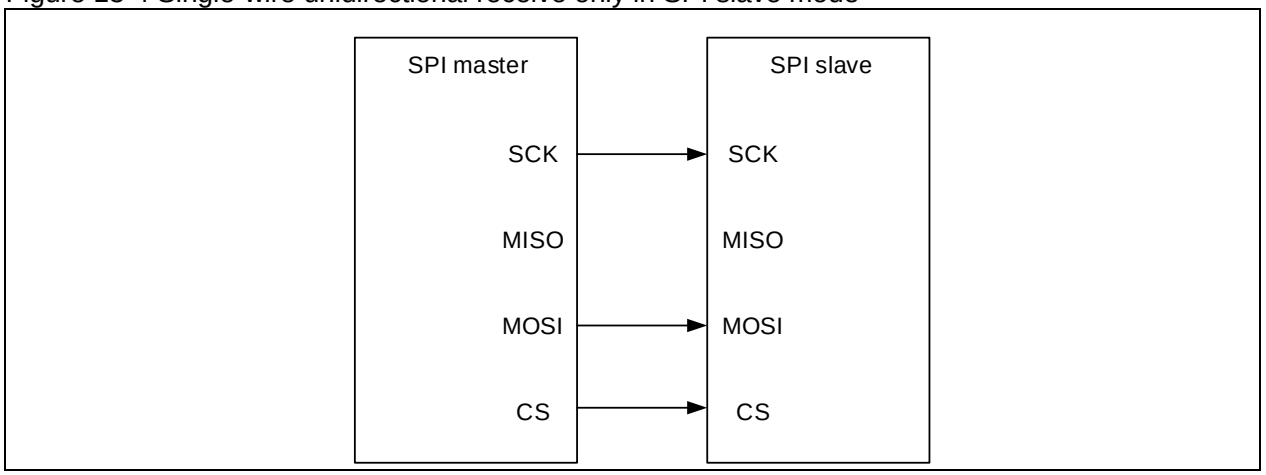


Figure 13-4 Single-wire unidirectional receive only in SPI slave mode



In master mode, it is required to wait until the second to last RDBF bit is set and then one SPI_CPK clock before disabling the SPI. The last RDBF must be set to 1 before entering power-saving mode (or disabling SPI system clock).

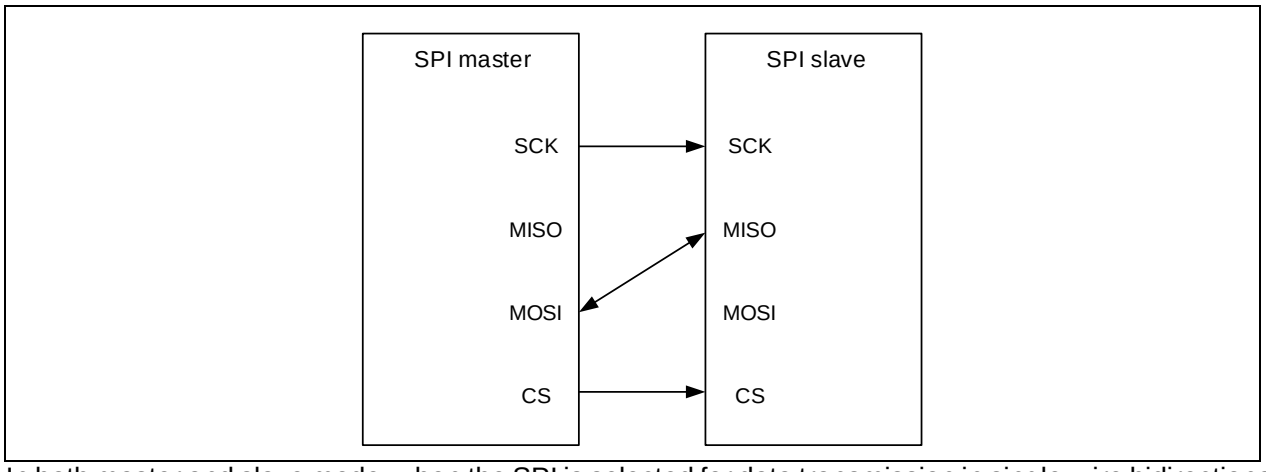
In slave mode, there is no need to check any flag before disabling the SPI. However, it is still necessary to wait until the BF becomes 0 before entering power-saving mode.

Figure 13-5 shows single-wire bidirectional half-duplex mode and SPI IO connection.

When SLBEN=1, the SPI operates in single-wire bidirectional half-duplex mode. In this case, the SPI supports data reception and transmission alternately. In master mode, the MOSI pin transmits/ receives data, and IO mapped onto MISO pin is released. In slave mode, the MISO pin transmits/receives data, and IO mapped onto MOSI pin is released.

The SLBTD bit is used by software to configure transfer direction. When SLBTD=1, the SPI can be used only for data transmission; when SLBTD=0, the SPI can be only used to receive data.

Figure 13-5 Single-wire bidirectional half-duplex mode



In both master and slave mode, when the SPI is selected for data transmission in single-wire bidirectional half-duplex mode, the TDBE bit must be set, and the BF must be 0 before disabling the SPI. The power-saving mode (or disabling SPI system clock) cannot be entered unless the SPI is disabled.

In master mode, when the SPI is selected for data reception in single-wire bidirectional half-duplex mode, it is required to wait until the second to last RDBF is set and then one SPI_SCK cycle before disabling the SPI. And the last RDBF must be set to 1 before entering power-saving mode (or disabling SPI system clock).

In slave mode, when the SPI is selected for data reception in single-wire bidirectional half-duplex mode, there is no need to check any flag before disabling the SPI. However, the BT must be 0 before entering power-saving mode (or disabling SPI system clock).

13.2.3 Chip select controller

The Chip select controller (CS) is used to enable hardware or software control for chip select signals through software configuration. This controller is used to select master/slave device in multi-processor mode, and to avoid conflicts on data lines by enabling the SCK signal output followed by CS signal. The hardware and software configuration procedure is detailed as follows, along with their respective input/output in master and slave mode.

CS hardware configuration procedure:

In master mode with CS being as an output, the CS hardware control is enabled by setting HWCSOE=1 and SWCSEN=0. If the SPI is enabled, low level is output on the CS pin. The CS signal is then released after the SPI is disabled and the transmission is complete.

In master mode with CS being as an input, the CS hardware control is enabled by setting HWCSOE=0 and SWCSEN=0. At this point, the SPI is automatically disabled by hardware and enters slave mode as soon as the CS pin low is detected by master SPI. The mode error flag (MMERR bit) is set accordingly. An interrupt is generated if ERRIE=1. During the period of MMERR being set, the SPIEN and MSTEN cannot be set by software. The MMERR is cleared by read or write access to the SPI_STS register followed by write operation to the SPI_CTRL1 register.

In slave mode with CS being as an input, the CS hardware control is enabled by setting HWCSOE=0 and SWCSEN=0. The slave determines whether to transmit / receive data based on the level on the CS pin. The slave is selected for data reception and transmission only when the CS pin is low.

CS software configuration procedure:

In master mode with CS being as an input, SWCSEN=1, the CS software control is enabled. When SWCSIL=0, the SPI is automatically disabled by hardware and enters slave mode. The mode error flag (MMERR bit) is set accordingly. An interrupt is generated if ERRIE=1. When the MMERR bit is set, the SPIEN and MSTEN bits cannot be set by software. The MMERR bit is cleared by read or write access to the SPI_STS register followed by write operation to the SPI_CTRL1 register.

In slave mode with CS being as an input, SWCSEN=1, the CS software control is enabled. The SPI judges the CS signal with the SWCSIL bit, instead of CS pin. When SWCSIL=0, the slave is selected for data reception and transmission.

13.2.4 SPI_SCK controller

The SPI protocol adopts synchronous transmission. In master mode with the SPI being used as SPI, it is required to generate a communication clock for data reception and transmission via the SPI interface, and the communication clock should be output to the slave via IO for data reception and transmission. In slave mode, the communication clock is provided by peripherals, and is input to the SPI via IO. In all, the SPI_SCK controller is used for the generation and configuration of SPI_SCK, with the configuration procedure detailed as follows:

SPI_SCK controller configuration procedure:

- Clock polarity and clock phase selection: by setting the CLKPOL and CLKPHA bit.
- Clock prescaler selection: Select the desired PCLK frequency by setting the CRM bit, and select the desired prescaler by setting the MDIV[3: 0] bit.
- Master/slave selection: Select SPI as master or slave by setting the MSTEN bit.

Note that the clock output is activated after the SPI is enabled in master reception-only mode, and it remains there until when the SPI is disabled and the reception is complete.

13.2.5 CRC

The SPI interface provides separate CRC calculation unit for transmission and reception. When used as SPI through software configuration, the automatic CRC calculation and check is performed while the user is reading or writing through DMA or CPU. During transmission, if the received data is not consistent with, detected by hardware, the data in the SPI_RCRC register, and such data is exactly the CRC value, then the CCERR bit will be set. An interrupt is generated if ERRIE=1.

The CRC function and configuration procedure of the SPI are described as follows.

CRC configuration procedure

- CRC calculation polynomial is configured by setting the SPI_CPOLY register.
- CRC enable: The CRC calculation is enabled by setting the CCEN bit. This operation will reset the SPI_RCRC and SPI_TCRC registers.
- Select if or when the NTC bit is set, depending on DMA or CPU data register. See the following descriptions.

Transmission using DMA

When DMA is used to write the data to be transmitted, if the CCEN bit is enabled, the hardware calculates the CRC value automatically according to the value in the SPI_CPOLY register and each transmitted data, and sends the CRC value at the end of the last data transmission. This CRC value is the value of the SPI_TCRC register.

Reception using DMA

When DMA is used to read the data to be received, if the CCEN bit is enabled, the hardware calculates the CRC value automatically according to the value in the SPI_CPOLY register and each received data, and waits until the completion of CRC data reception at the end of the last data reception before comparing the received CRC value with the value of the SPI_RCRC register. If check error occurs, the CCERR flag is set. An interrupt is generated if the ERRIE bit is enabled.

Transmission using CPU

Unlike DMA mode, after writing the last data to be transmitted, the CPU mode requires the NTC bit to be set by software before the end of the last data transmission.

Reception using CPU

In two-wire unidirectional full-duplex mode, the CRC calculation and check in CPU reception mode will be completed automatically by following CPU transmission mode to operate the NTC bit,

In single-wire unidirectional reception-only mode and single-wire bidirectional reception-only mode, it is required to set the NTC bit before the software receives the last data when the second-to-last data is already received.

13.2.6 DMA transfer

The SPI supports write and read operations with DMA. Refer to the following configuration procedure.

Special attention should be paid to: when the CRC calculation and check is enabled, the number of data transferred by DMA is configured as the number of the data to be transferred plus 1. The number of data read with DMA is configured as the number of the data to be received. In this case, the hardware will send CRC automatically at the end of full transfer, and the receiver will continue to perform CRC automatic check. Note that the received CRC data will be moved into the SPI_DT register by hardware, the RDBF is set, and DMA read request will be issued if then DAM transfer feature is enabled. Hence, it is recommended to read the SPI_DT register by software to get the CRC value at the end of CRC reception in order to avoid the upcoming transfer error.

Transmission with DMA

- Select DMA channel: Select a DMA channel for the current SPI from DMA channel map table described in DMA chapter.
- Configure the destination of DMA transfer: Configure the SPI_DT register address as the destination address bit of DMA transfer in the DMA control register. Data will be sent to this address after transmit request is received by DMA.
- Configure the source of DMA transfer: Configure the memory address as the source of DMA transfer in the DMA control register. Data will be loaded into the SPI_DT register from the memory address after transmit request is received by DMA.
- Configure the total number of bytes to be transferred in the DMA control register.
- Configure the channel priority of DMA transfer in the DMA control register.
- Configure DMA interrupt generation after half or full transfer in the DMA control register.
- Enable DMA transfer channel in the DMA control register.

Reception with DMA

- Select DMA transfer channel: Select a DMA channel for the current SPI from DMA channel map table described in DMA chapter.
- Configure the destination of DMA transfer: Configure the memory address as the destination of DMA transfer in the DMA control register. Data will be loaded from the SPI_DT register to the programmed destination after reception request is received by DMA.
- Configure the source of DMA transfer: Configure the SPI_DT register address as the source of DMA transfer in the DMA control register. Data will be loaded from the SPI_DT register to the programmed destination after reception request is received by DMA.
- Configure the total number of bytes to be transferred in the DMA control register.
- Configure the total number of bytes to be transferred in the DMA control register.
- Configure DMA interrupt generation after half or full transfer in the DMA control register
- Enable DMA transfer channel in the DMA control register.

13.2.7 TI mode

The SPI interface is compatible with the TI protocol. The TI mode is enabled by setting the TIEN bit to 1.

In this mode, the SPI interface will generates a communication clock SPI_CLK in accordance with the TI protocol. This means that the SPI_CLK polarity and phase are forced to conform to the TI protocol requirements, without the need of the intervention of CLKPOL and CLKPHA bits. Thus the CLKPOL and CLKPHA bits cannot be used to change the polarity and phase of the SPI_CLK either.

In this mode, the SPI interface will generate a CS signal in accordance with the TI protocol, meaning that the CS input and output are forced to conform to the TI protocol requirements, without the need of the intervention of SWCSEN, SWCSIL and HWCSOE bits. Thus, the SWCSEN, SWCSIL and HWCSOE bits cannot be used for CS signal management either.

In slave mode, once the TI mode is enabled, the SPI slave controls the MISO pin only during data transmission, meaning that the MISO pin state remains Hi-Z in idle state.

In slave mode, once the TI mode is enabled, the SPI interface is capable of detecting CS pulse errors during data transmission, and setting the CSPAS bit (It is cleared by reading the SPI_STS) as soon as

a CS pulse error is detected. At this point, the detected erroneous pulse will be ignored by the SPI. However, since there is something wrong with the CS signal, the software should disable the SPI slave and re-configure the SPI master before re-enabling the SPI slave for communication.

13.2.8 Transmitter

The SPI transmitter is clocked by SPI_SCK controller. It can output different data frame formats, depending on software configuration. There is a SPI_DT register available in the SPI which is used to be written with the data to be transmitted. When the transmitter is clocked, the contents in the SPI_DT register are copied into the data buffer (Unlike SPI_DT, it is driven by SPI_SCK, and controlled by hardware, instead of software), and sent out in order based on the programmed frame format.

Both DMA and CPU can be used for write operation. For DMA transfer, refer to DMA transfer section for more details. For CPU transfer, attention should be paid to the TDBE bit. The reset value of this bit is 1, indicating that the SPI_DT register is empty. If the TDBEIE bit is set, an interrupt is generated. After the data is written, the TDBE is pulled low until the data is moved to the transmit data buffer before the TDBE is set once again. This means that the user can be allowed to write the data to be transmitted only when the TDBE is set.

After the transmitter is configured and the SPI is enabled, the SPI is ready for data transmission. Before going forward, it is necessary for the users to refer to full-duplex / half-duplex chapter to get detailed configuration information, go to the Chip select controller chapter for specific chip select mode, check the SPI_SCK controller chapter for information on communication clock, and refer to CRC and DMA transfer chapter to configure CRC and DMA (if necessary). The recommended configuration procedure are as follows.

Transmitter configuration procedure:

- Configure full-duplex/half-duplex selector
- Configure chip select controller
- Configure SPI_SCK controller
- Configure CRC (if necessary)
- Configure DMA transfer (if necessary)
- If the DMA transfer mode is not used, the software will check whether to enable transmit data interrupt (TDBEIE =1) through the TDBE bit.
- Configure frame format: select MSB/LSB mode with the LTF bit, and select 8/16-bit data with the FBN bit
- Enable SPI by setting the SPIEN

13.2.9 Receiver

The SPI receiver is clocked by the SPI_SCK controller. It can output different data frame formats through software configuration. There is a receive data buffer register, driven by the SPI_SCK, in the SPI receiver.

At the last CLK of each transfer, the data is moved from the shift register to the receive data buffer register. Then the transmitter sets the receive data complete flag to the SPI logic. When the flag is detected by the SPI logic, the data in the receive data buffer is copied into the SPI_DT register, with the RDBF being set. This means that the data is received, and it is already stored into the SPI_DT. In this case, read access to the SPI_DT register will clear the RDBF bit.

Both DMA and CPU can be used for read operation. For DMA transfer, refer to DMA transfer section for more details. For CPU transfer, attention should be paid to the RDBE bit. The reset value of this bit is 0, indicating that the SPI_DT register is empty. If the data is received and moved into the SPI_DT, the RDBF is set, meaning that there are some data to be read in the SPI_DT register. An interrupt is generated if the RDBFIE bit is set.

When the next received data is ready to be moved to the SPI_DT register while the previously received data has not been read yet (RDBF=1), then the data overflow occurs. The previously receive data are not lost, but the next received data will do. At this point, the ROERR is set. An interrupt is generated if the ERRIE is set. Read SPI_DT register and then the SPI_STS register will clear the ROERR bit. The recommended configuration procedure is as follows.

Receiver configuration procedure:

- Configure full-duplex/half-duplex selector
- Configure chip select controller
- Configure SPI_SCK controller
- Configure CRC (if necessary)
- Configure DMA transfer (if necessary)
- If the DMA transfer mode is not used, the software will check whether to enable receive data interrupt (RDBEIE =1) through the RDBE bit.
- Configure frame format: select MSB/LSB mode with the LTF bit, and select 8/16-bit data with the FBN bit
- Enable SPI by setting the SPIEN

13.2.10 Motorola mode

This section describes the SPI communication timings, which includes full-duplex and half-duplex master/slave timings.

Full-duplex communication – master mode

For host side, configured as follows:

MSTEN=1: Master enable

SLBEN=0: Full-duplex mode

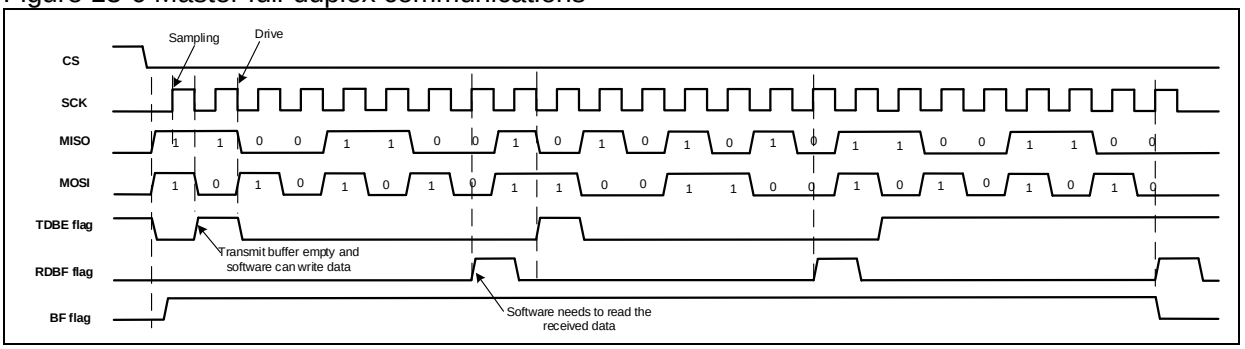
CLKPOL=0, CLKPHA=0: SCK idle output low, use the first edge for sampling

FBN=0: 8-bit frame

Master transmit (MOSI): 0xaa, 0xcc, 0xaa

Slave transmit (MISO): 0xcc, 0xaa, 0xcc

Figure 13-6 Master full-duplex communications



Full-duplex communication – slave mode

For slave side, configured as follows:

MSTEN=0: Slave enable

SLBEN=0: Full-duplex mode

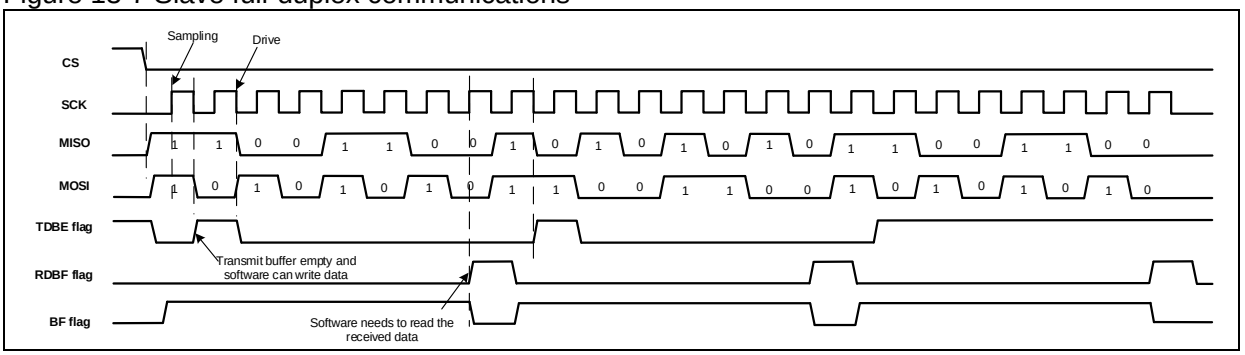
CLKPOL=0, CLKPHA=0: SCK idle output low, use the first edge for sampling

FBN=0: 8-bit frame

Master transmit (MOSI): 0xaa, 0xcc, 0xaa

Slave transmit (MISO): 0xcc, 0xaa, 0xcc

Figure 13-7 Slave full-duplex communications



Half-duplex communication – master transmit timing

Configured as follows:

MSTEN=1: Master enable

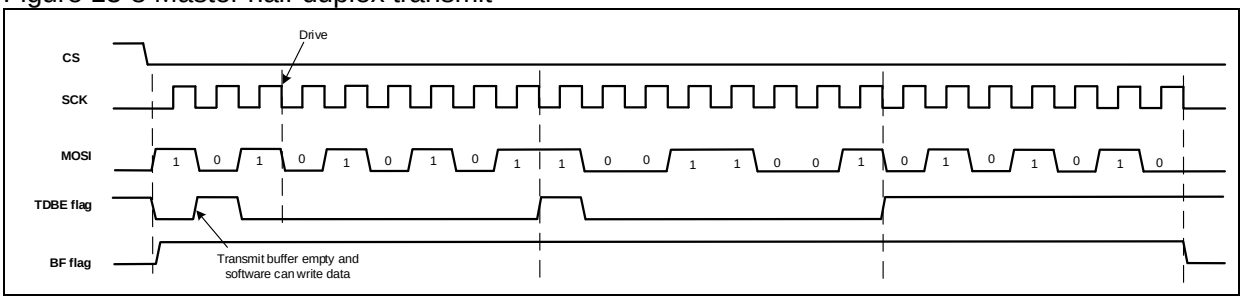
SLBEN=1: Single line bidirectional mode

CLKPOL=0, CLKPHA=0: SCK idle output low, use the first edge for sampling

FBN=0: 8-bit frame

Master transmit (MOSI): 0xaa, 0xcc, 0xaa

Figure 13-8 Master half-duplex transmit



Half-duplex communication – slave receive

Configured as follows:

MSTEN=0: Slave enable

SLBEN=1: Single line bidirectional mode

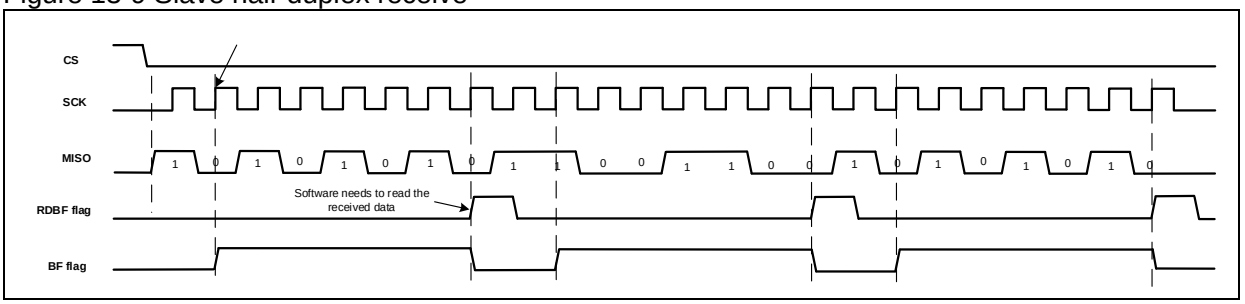
SLBTD=0: Receive mode

CLKPOL=0, CLKPHA=0: SCK idle output low, use the first edge for sampling

FBN=0: 8-bit frame

Slave receive: 0xaa, 0xcc, 0xaa

Figure 13-9 Slave half-duplex receive



Half-duplex communication – slave transmit

Configured as follows:

MSTEN=0: Slave enable

SLBEN=1: Single line bidirectional mode

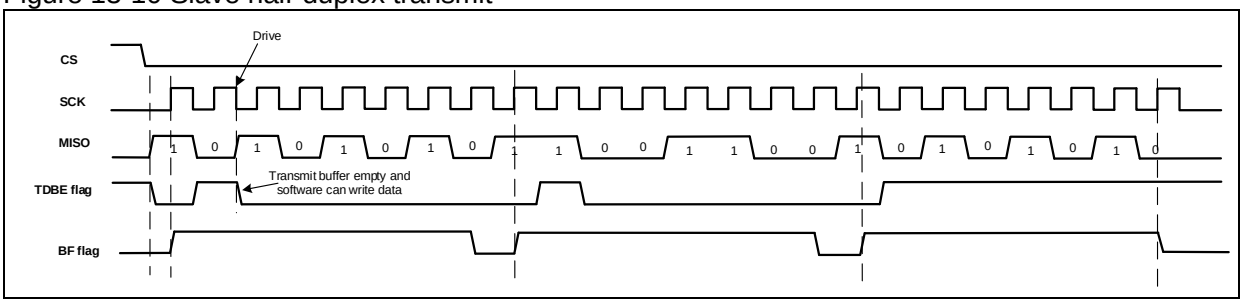
SLBTD=1: Transmit enable

CLKPOL=0, CLKPHA=0: SCK idle output low, use the first edge for sampling

FBN=0: 8-bit frame

Slave transmit: 0xaa, 0xcc, 0xaa

Figure 13-10 Slave half-duplex transmit



Half-duplex communication – master receive

Configured as follows:

MSTEN=1: Master enable

SLBEN=1: Single line bidirectional mode

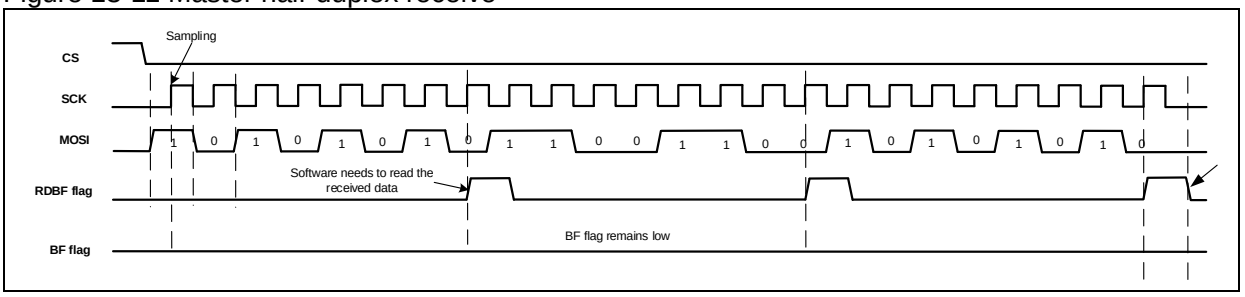
SLBTD=0: Receive enable

CLKPOL=0, CLKPHA=0: SCK idle output low, use the first edge for sampling

FBN=0: 8-bit frame

Master receive: 0xaa, 0xcc, 0xaa

Figure 13-11 Master half-duplex receive

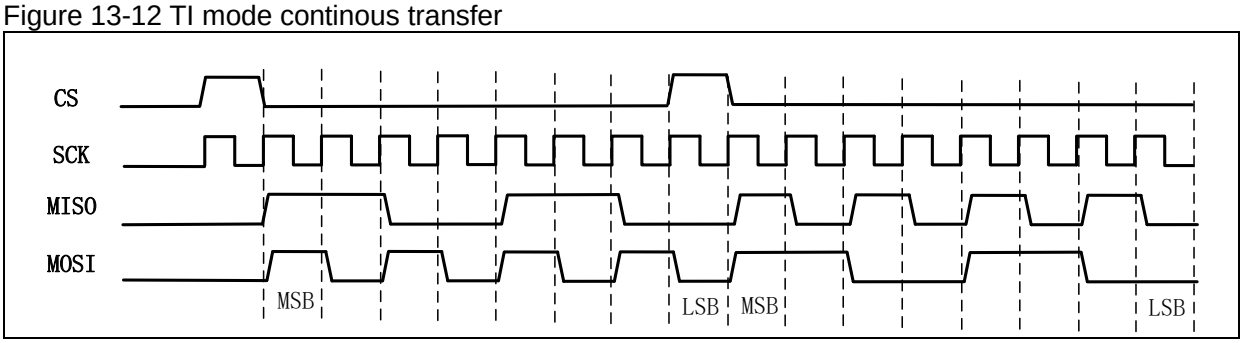


13.2.11 TI mode

The SPI interface supports TI mode. This mode is enabled by setting TIEN=1.

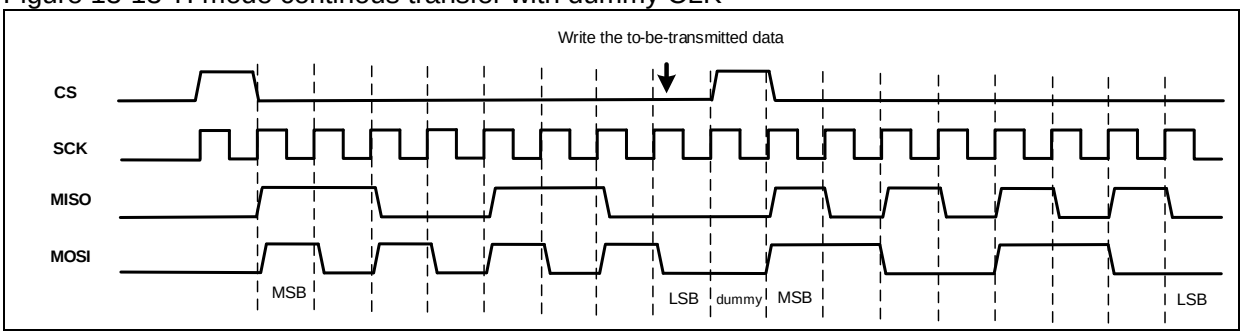
In TI mode, a slight difference is present between continuous and discontinuous communication timings. When the to-be transmitted data is written before the rising SCK edge corresponding to the last data of the current transmit frame, it is a continuous communication, without dummy CLK between data, and the host sends a valid CS pulse while transmitting the last data of the current frame.

Figure 13-12 TI mode continous transfer



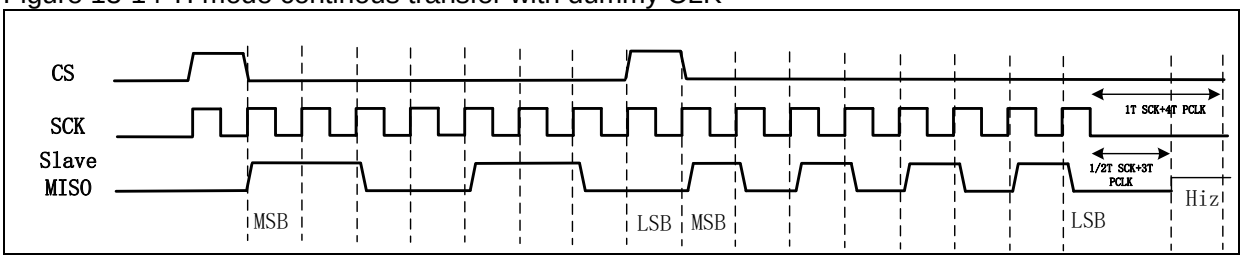
In TI mode, when the to-be-transmitted data is written between the rising and falling SCK edge corresponding to the last data of the current transmit frame, a dummy CLK exists between data.

Figure 13-13 TI mode continous transfer with dummy CLK



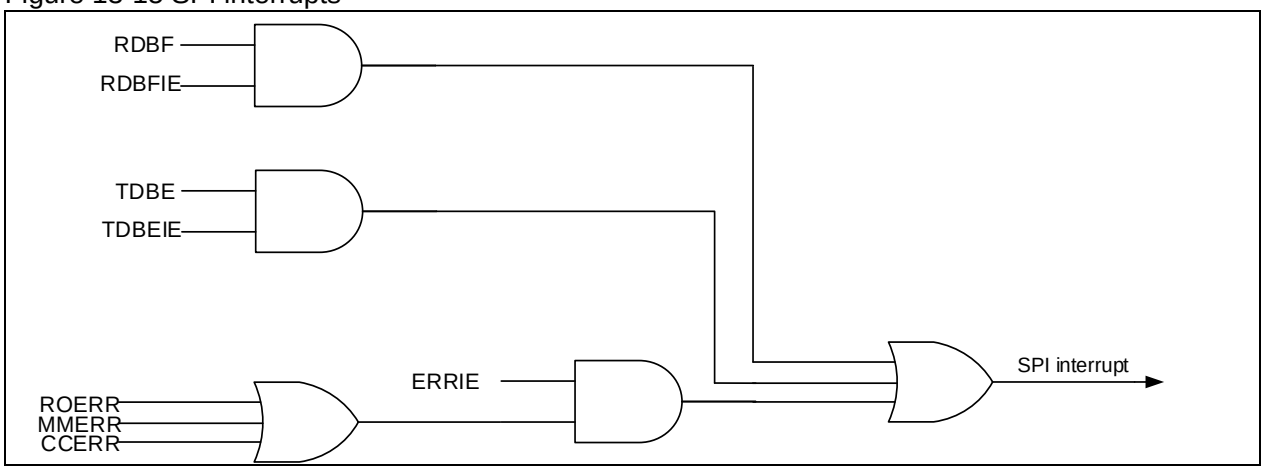
In TI mode, when the to-be-transmitted data is written after the falling SCK edge corresponding to the last data of the current transmit frame, the host always issues a valid SCK clock after 1T SCK + 4T PCLK cycles. If the slave still does not detect a valid CS pulse at the end of the current data reception, it disables MISO output after 1/2T SCK + 3T PCLK cycles to control MISO floating.

Figure 13-14 TI mode continous transfer with dummy CLK



13.2.12 Interrupts

Figure 13-15 SPI interrupts



13.2.13 IO pin control

When used as SPI, the SPI interface is connected to external devices through four pins.

- MISO: Master In/Slave Out. The pin is used to receive data from slave in SPI master mode, and transmit data in slave mode.
- MOSI: Master Out/Slave In. The pin is used to transmit data in SPI master mode, and receive data in slave mode.
- SCK: SPI communication clock. The pin serves as output (communication clock is sent to peripheral via this pin) in SPI master mode, and as input (communication clock from master is input to SPI via this pin) in SPI slave mode.
- CS: Chip Select. This is an optional pin which is used to select master/slave mode. Refer to CS section for more information.

13.2.14 Precautions

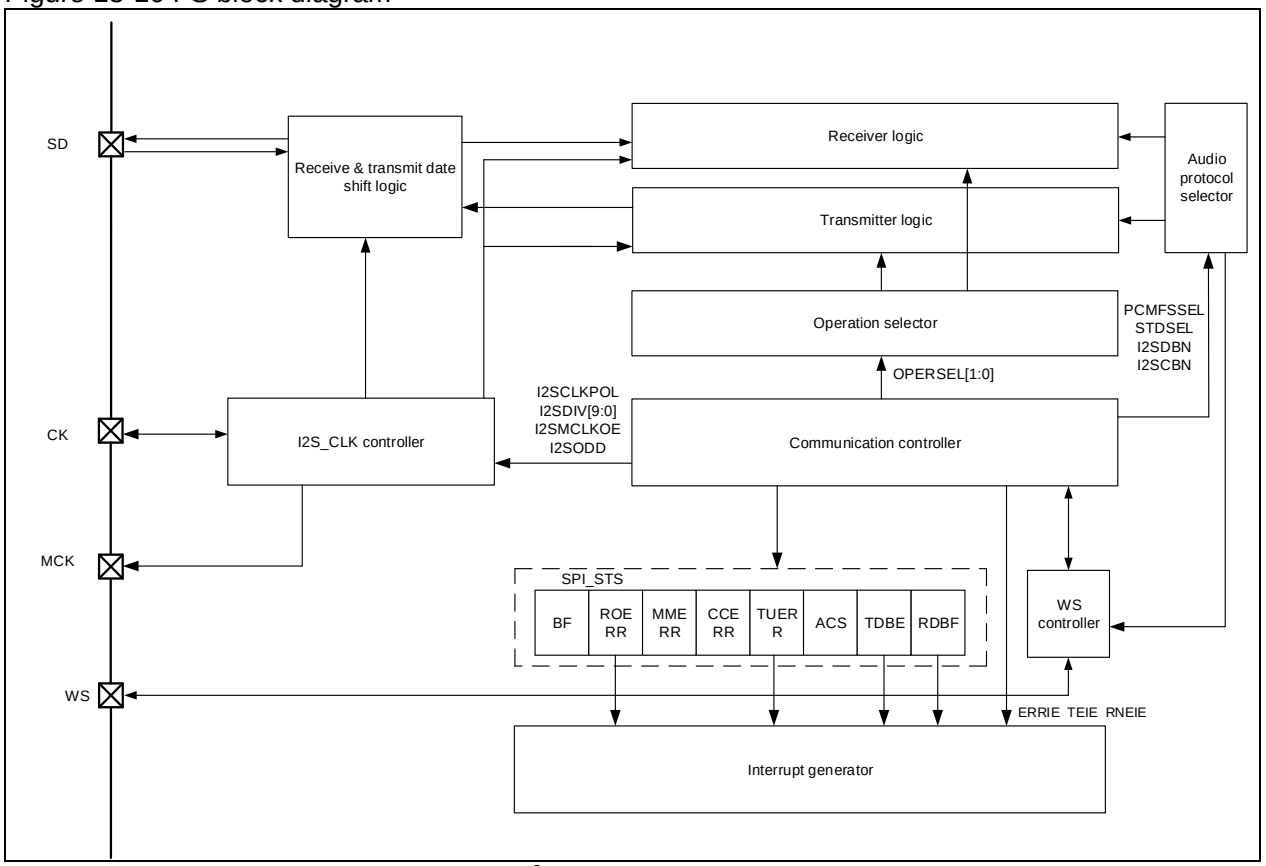
- CRC value should be obtained by software reading DT register at the end of CRC reception

13.3 I²S functional description

13.3.1 I²S introduction

The I²S is capable of operating in master receive, master transmit, and slave receive and slave transmit, depending on software configuration. These four operating modes support four audio protocols including Philips standard, MSB-aligned standard, LSB-aligned standard and PCM standard, respectively. The DMA transfer is also supported.

Figure 13-16 I²S block diagram



Main features when the SPI is used as I²S:

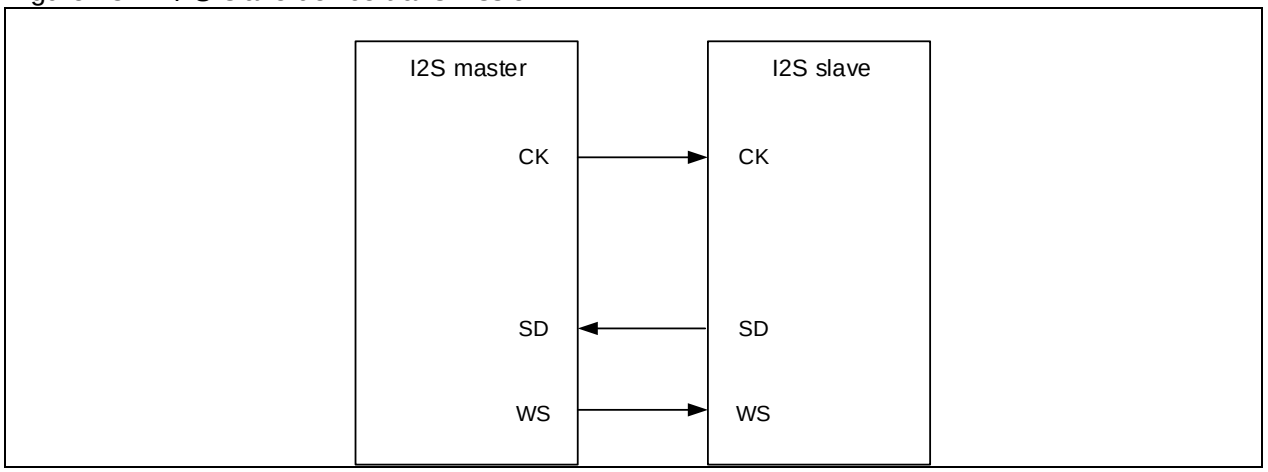
- Programmable operating modes
  - Slave device transmission
  - Slave device reception
  - Master device transmission
  - Master device reception
- Programmable clock polarity
- Programmable clock frequency (8 KHz to 192 KHz)
- Programmable data bits (16 bits, 24 bits, 32 bits)
- Programmable channel bits (16 bits, 32 bits)
- Programmable audio protocol
  - I²S Philips standard
  - MSB-aligned standard (left-aligned)
  - LSB-aligned standard (right-aligned)
  - PCM standard (channel frame with short and long frame synchronization)
- DMA transfer
- Main peripheral clock with a fixed frequency of 256x Fs (audio sampling frequency)

13.3.2 Operating mode selection

The SPI, used as I²S selector, offers multiple operating modes for selection, namely, slave device transmit, slave device receive, master device transmit and master device receive. This is done by software configuration.

Slave device transmission:
Set the I2SMSEL bit, and OPERSEL[1:0] =00, the I²S will work in slave device transmission mode.

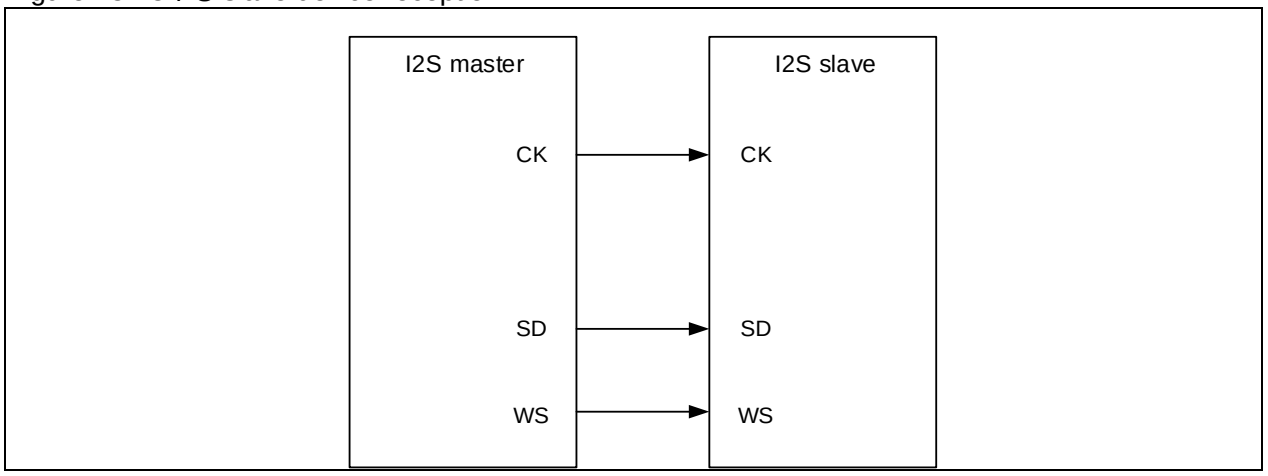
Figure 13-17 I²S slave device transmission



Slave device reception:

Set the I2SMSEL bit, and OPERSEL[1:0]=01, the I²S will work in slave device reception mode.

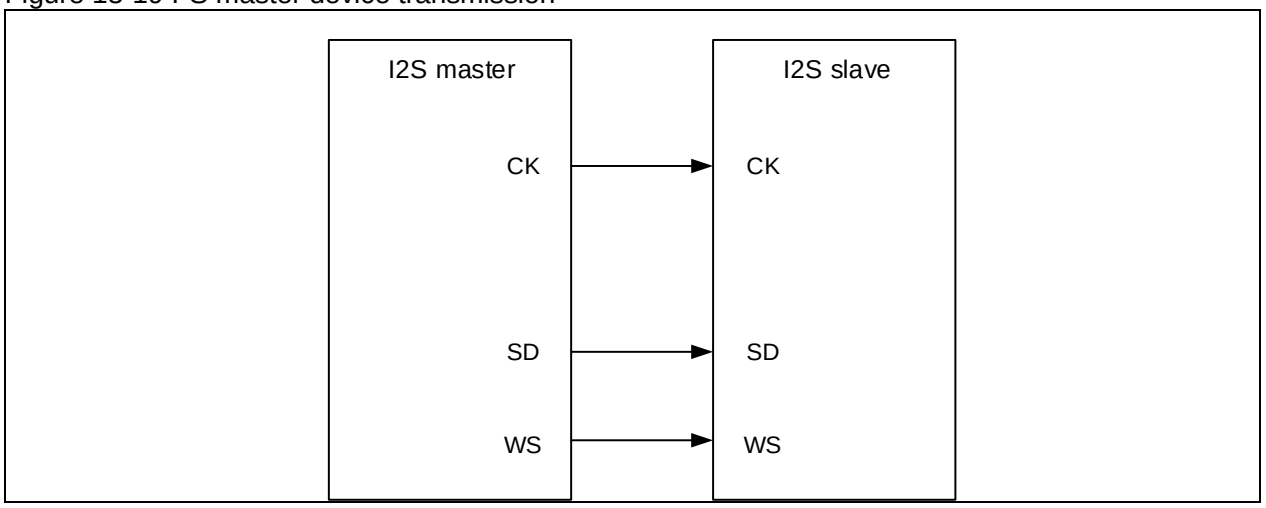
Figure 13-18 I²S slave device reception



Master device transmission:

Set the I2SMSEL bit, and OPERSEL[1:0]=10, the I²S will work in master device transmission mode.

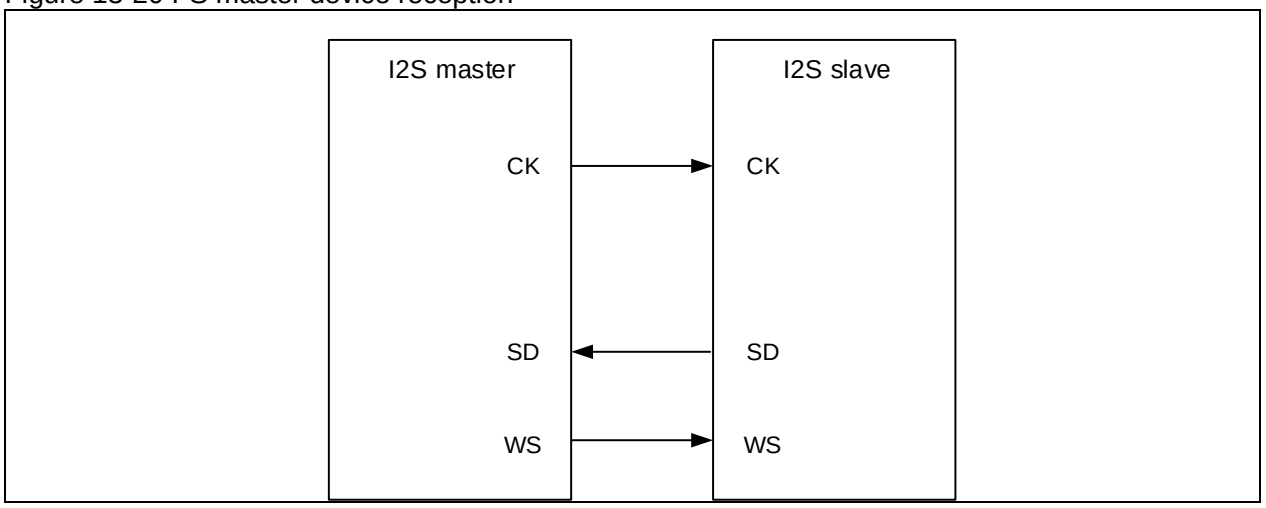
Figure 13-19 I²S master device transmission



Master device reception:

Set the I2SMSEL bit, and OPERSEL[1: 0]=11, the I²S will work in master device reception mode.

Figure 13-20 I²S master device reception



13.3.3 Audio protocol selector

As I²S interface, the SPI supports multiple audio protocols. The user is able to select the desired audio protocol, the number of data bits and of channel bits through the audio protocol selector by software. By controlling the WS controller automatically, the audio protocol selector outputs or detects WS signals that conform to the protocol requirements.

- Select audio protocol by setting the STDSEL bit
  STDSLE=00: Philips standard
  STDSLE=01: MSB-aligned standard (left-aligned)
  STDSLE=10: LSB-aligned standard (right-aligned)
  STDSLE=11: PCM standard
- Select PCM frame synchronization format: PCMFSSEL=1 for PCM long frame synchronization, PCMFSSEL=0 for short frame synchronization (this step is required when selecting PCM protocol)
- Select the number of data bits by setting the I2SDBN bit
  I2SDBN=00: 16 bits
  I2SDBN =01: 24 bits
  I2SDBN =10: 32 bits

- Select the number of channel bits by setting the I2SCBN bit
  I2SDBN =0: 16 bits
  I2SDBN =1: 32 bits

Note: Read/Write operation mode depends on the selected audio protocol, data bits and channel bits. The following lists all possible configuration combinations and their respective read and write operation mode.

- **Philips standard, PCM standard, MSB-aligned or LSB-aligned standard, 16-bit data and 16-bit channel**

  The data bits are the same as the channel bits. Each channel requires only one read/write operation from/ to the SPI_DT register, and the number of DMA transfer is 1.

- **Philips standard, PCM standard or MSB-aligned standard, 16-bit data and 32-bit channel**

  The data bits are different from the channel bits. Each channel requires only one read/write operation from/to the SPI_DT register, and the number of DMA transfer is 1. The first 16 bits are valid data, and the remaining 16-bit are forced to 0 by hardware.

- **Philips standard, PCM standard or MSB-aligned standard, 24-bit data and 32-bit channel**

  The data bits are different from the change bits. Each channel requires two read/write operations from/to the SPI_DT register, and the number of DMA transfer is 2. The first 16-bit channel transmits and receives the first 16-bit data, while the last 16-bit channel transmits and receives the upper 8-bit data, and the lower 8-bit data are forced to 0 by hardware.

- **Philips standard, PCM standard, MSB-aligned or LSB-aligned standard, 32-bit data and 32-bit channel**

  The data bits are the same as the channel bits. Each channel requires two read/write operations from/to the SPI_DT register, and the number of DMA transfer is 2. These 32-bit data are proceeded (transmit and reception) in two times, with 16-bit data each time.

- **LSB-aligned standard, 16-bit data and 32-bit channel**

  The data bits are different from the channel bits. Each channel requires only one read/write operation from/to the SPI_DT register, and the number of DMA transfer is 1. The last 16 bits (LSB) are valid data, while the first 16-bit data (MSB) are forced to 0 by hardware.

- **LSB-aligned standard, 24-bit data and 32-bit channel**

  The data bits are different from the channel bits. Each channel requires two read/write operations from/to the SPI_DT register, and the number of DMA transfer is 2. Of the first 16-bit data, its lower 8 bits are valid data, and the upper 8 bits are forced to 0 by software; the last 16 bits transmit and receive the second 16-bit data.

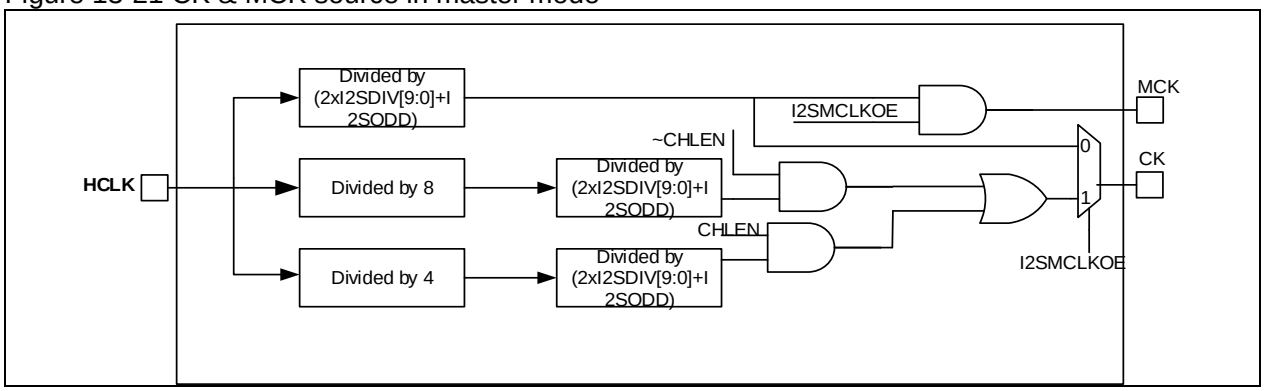
13.3.4 I2S_CLK controller

As I^2S, The audio protocols the SPI supports adopts synchronous transmission. In master mode, it is required to generate a communication clock for data reception and transmission on the SPI, and the communication clock should be output to the slave via IO for data reception and transmission. In slave mode, the communication clock is provided by master, and is input to the SPI via IO. In all, the I2S_SCK controller is used for the generation and configuration of I2S_SCK, with the configuration procedure detailed as follows:

In I^2S master mode, the SPI provides communication clock (CK) and main peripheral clock (MCK) shown in *Figure 13-21*. The CK and MCK are generated by HCLK divider, and the MCK frequency division factor depends on the I2SDIV and I2SODD. The calculation formula is seen in *Figure 13-21*.

The CK frequency division factor depends on whether to provide the main clock for peripherals. To ensure that the main clock is always 256 times the audio sampling frequency, the provision of main clock and the number of channel bits should be taken into account. When the main clock is needed, the CK should be divided by 8 (I2SCBN=0) or 4 (I2SCBN=1), then divided again by the same frequency division factor as that of the MCK, that is the final communication clock; When the main clock is not needed, the CK frequency division factor is determined by I2SDIV and I2SODD, shown in *Figure 13-21*.

Figure 13-21 CK & MCK source in master mode



In addition to the above configuration, the following table lists the values of I2SDIV and I2SODD corresponding to some specific frequencies, as well as their respective error for the users to configure the I2SDIV and I2SODD.

Table 13-1 Audio frequency precision using system clock

SCLK (MHz)	MCLK	Target Fs (Hz)	16bit				32bit			
			I2S DIV	I2S_ODD	RealFs	Error	I2S DIV	I2S_ODD	RealFs	Error
72	No	192000	6	0	187500	2.34%	3	0	187500	2.34%
72	No	96000	11	1	97826.09	1.90%	6	0	93750	2.34%
72	No	48000	32	1	34615.38	27.88%	11	1	48913.04	1.90%
72	No	44100	25	1	44117.65	0.04%	13	0	43269.23	1.88%
72	No	32000	35	0	32142.86	0.45%	17	1	32142.86	0.45%
72	No	22050	51	0	22058.82	0.04%	25	1	22058.82	0.04%
72	No	16000	70	1	15957.45	0.27%	35	0	16071.43	0.45%
72	No	11025	102	0	11029.41	0.04%	51	0	11029.41	0.04%
72	No	8000	140	1	8007.117	0.09%	70	1	7978.723	0.27%
72	Yes	96000	2	0	70312.5	26.76%	2	0	70312.5	26.76%
72	Yes	48000	3	0	46875	2.34%	3	0	46875	2.34%
72	Yes	44100	3	0	46875	6.29%	3	0	46875	6.29%
72	Yes	32000	4	1	31250	2.34%	4	1	31250	2.34%
72	Yes	22050	6	1	21634.62	1.88%	6	1	21634.62	1.88%
72	Yes	16000	9	0	15625	2.34%	9	0	15625	2.34%
72	Yes	11025	13	0	10817.31	1.88%	13	0	10817.31	1.88%
72	Yes	8000	17	1	8035.714	0.45%	17	1	8035.714	0.45%

13.3.5 DMA transfer

The SPI supports write and read operations with DMA. Whether used as SPI or I²S, read/write request using DMA comes from the same peripheral. As a result, their configuration procedure are the same, described as follows.

Transmission with DMA

- Select a DMA channel: Select a DMA channel for the current SPI from DMA channel map table described in DMA chapter.
- Configure the destination of DMA transfer: Configure the SPI_DT register address as the destination address bit of DMA transfer in the DMA control register. Data will be sent to this address after transmit request is received by DMA.
- Configure the source of DMA transfer: Configure the memory address as the source of DMA transfer in the DMA control register. Data will be loaded into the SPI_DT register from the memory address after transmit request is received by DMA.
- Configure the total number of bytes to be transferred in the DMA control register.

- Configure the channel priority of DMA transfer in the DMA control register.
- Configure DMA interrupt generation after half or full transfer in the DMA control register.
- Enable DMA transfer channel in the DMA control register.

Reception with DMA

- Select DMA transfer channel: Select a DMA channel for the current SPI from DMA channel map table described in DMA chapter.
- Configure the destination of DMA transfer: Configure the memory address as the destination of DMA transfer in the DMA control register. Data will be loaded from the SPI_DT register to the programmed destination after reception request is received by DMA.
- Configure the source of DMA transfer: Configure the SPI_DT register address as the source of DMA transfer in the DMA control register. Data will be loaded from the SPI_DT register to the programmed destination after reception request is received by DMA.
- Configure the total number of bytes to be transferred in the DMA control register.
- Configure the total number of bytes to be transferred in the DMA control register.
- Configure DMA interrupt generation after half or full transfer in the DMA control register
- Enable DMA transfer channel in the DMA control register.

13.3.6 Transmitter/Receiver

Whether used as SPI or I2S, there is no difference for CPU. The SPI (in whatever mode) shares the same base address, the same SPI_DT register, the same transmitter and receiver. The SPI transmitter and receiver is responsible for sending and receiving the desired data frame according to the configuration of the communication controller. Thus their status flags such as TDBE, RDBF and ROERR, and their interrupt enable bits including TDBEIE, RDBFIE and ERRIE are identical.

Special attention must be paid to:

- CRC check is not available on the I^2S. Any operations related to CRC, including CCERR flag and corresponding interrupts, are not supported.
- I^2S protocol needs decode the current channel status. The ACS bit is used to judge whether the current transfer occurs on the left channel (ACS=0) or the right channel (ACS=1).
- TUERR bit indicates whether an underrun occurs. TUERR=1 means an underrun error occurs on the transmitter. An interrupt is generated when the ERRIE is set.
- Read/write operation to the SPI_DT register is different under different audio protocols, data bits and channel bits. Refer to the audio protocol selector section for more information.
- Pay more attention to the I^2S disable operation under different configurations, shown as follows:
  — I2SDBN=00, I2SCBN=1, STDSLE=10: wait for the second-to-last RDBF=1 and 17 CK cycles before disabling the I^2S.
  — I2SDBN=00, I2SCBN=1, STDSLE=00 or STDSLE=01 or STDSLE=11: wait for the last RDBF=1 and one CK cycle before disabling I^2S.
  — I2SDBN, I2SCBN, STDSLE combination: wait for the second-to-last RDBF=1 and one CK cycle before disabling the I^2S.

I^2S transmitter configuration procedure:

- Configure operation mode selector
- Configure audio protocol selector
- Configure I2S_SCK controller
- Configure DMA transfer (if necessary)
- Set the I2SEN bit to enable I^2S

I^2S receiver configuration procedure:

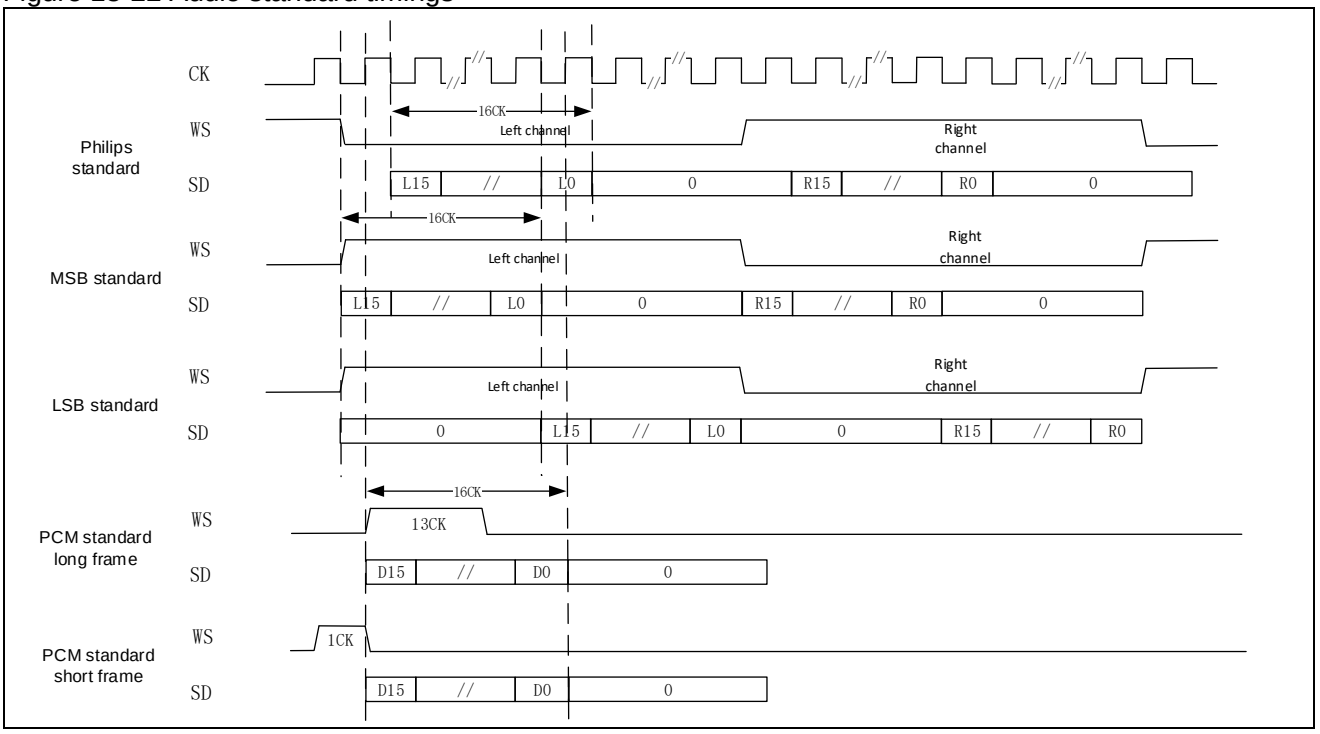
- Configure operation mode selector
- Configure audio protocol selector
- Configure I2S_SCK controller
- Configure DMA transfer (if necessary)

- Set the I2SEN bit to enable I²S

13.3.7 I²S communication timings

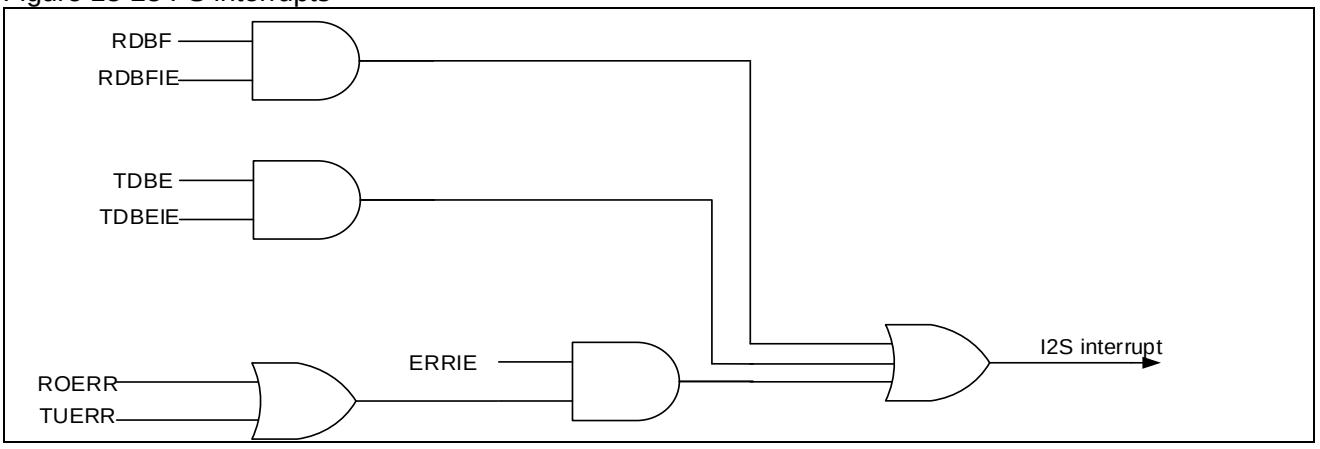
I²S supports four audio protocols including Philips standard, MSB-aligned standard, LSB-aligned standard and PCM standard, respectively. Figure 13-23 shows their respective timings.

Figure 13-22 Audio standard timings



13.3.8 Interrupts

Figure 13-23 I²S interrupts



13.3.9 IO pin control

The I²S needs three pins for transfer operation, namely, the SD, WS and CK. The MCLK pin is also required if there is a need to provide main clock for peripherals. Considering the SPI interface cannot be used as I²S and SPI at the same time, the I²S shares some pins with SPI, described as follows:

- SD: Data pin (share the same GPIO with MOSI pin) for data transmission and reception.
- WS: Word select (share the same GPIO with CS pin) for output in master mode, and input in slave mode.
- CK: Communication clock (share the same GPIO with SCK pin) for output in master mode, and input in slave mode.

- MCLK: Master clock (it is used in I_2S mode only, share the same GPIO with MISO) is used to provide main clock for peripherals. The frequency of output clock signal is set to 256x Fs (audio sampling frequency)

13.4 SPI registers

These peripheral registers must be accessed by words (32 bits).

Table 13-2 SPI register map and reset value

Register	Offset	Reset value
SPI_CTRL1	0x00	0x0000
SPI_CTRL2	0x04	0x0000
SPI_STS	0x08	0x0002
SPI_DT	0x0C	0x0000
SPI_CPOLY	0x10	0x0007
SPI_RCRC	0x14	0x0000
SPI_TCRC	0x18	0x0000
SPI_I2SCTRL	0x1C	0x0000
SPI_I2SCLKP	0x20	0x0002

13.4.1 SPI control register1 (SPI_CTRL1) (Not used in I^2S mode)

Bit	Name	Reset value	Type	Description
Bit 15	SLBEN	0x0	rw	Single line bidirectional half-duplex enable 0: Disabled 1: Enabled
Bit 14	SLBTD	0x0	rw	Single line bidirectional half-duplex transmission direction This bit and the SLBEN bit together determine the data output direction in “Single line bidirectional half-duplex” mode. 0: Receive-only mode 1: Transmit-only mode
Bit 13	CCEN	0x0	rw	CRC calculation enable 0: Disabled 1: Enabled
Bit 12	NTC	0x0	rw	Transmit CRC next When this bit is set, it indicates that the next data transferred is CRC value. 0: Next transmitted data is normal data 1: Next transmitted data is CRC value
Bit 11	FBN	0x0	rw	Frame bit num This bit is used to configure the number of data frame bit for transmission/reception. 0: 8-bit data frame 1: 16-bit data frame
Bit 10	ORA	0x0	rw	Receive-only active In two-wire unidirectional mode, when this bit is set, it indicates that Receive-only is active, but the transmit is not allowed. 0: Transmission and reception 1: Receive-only mode
Bit 9	SWCSEN	0x0	rw	Software CS enable When this bit is set, the CS pin level is determined by the SWCSIL bit. The status of I/O level on the CK pin is invalid. 0: Disabled 1: Enabled
Bit 8	SWCSIL	0x0	rw	Software CS internal level This bit is valid only when the SWCSEN is set. It

				determines the level on the CS pin. In master mode, this bit must be set. 0: Low level 1: High level
Bit 7	LTF	0x0	rw	LSB transmit first This bit is used to select for MST transfer first or LSB transfer first. 0: MSB 1: LSB
Bit 6	SPIEN	0x0	rw	SPI enable 0: Disabled 1: Enabled
Bit 5: 3	MDIV	0x0	rw	Master clock frequency division In master mode, the peripheral clock divided by the prescaler is used as SPI clock. The MDIV[3] bit is in the SPI_CTRL2 register, MDIV[3: 0]: 0000: Divided by 2 0001: Divided by 4 0010: Divided by 8 0011: Divided by 16 0100: Divided by 32 0101: Divided by 64 0110: Divided by 128 0111: Divided by 256 1000: Divided by 512 1001: Divided by 1024
Bit 2	MSTEN	0x0	rw	Master enable 0: Disabled (Slave) 1: Enabled (Master)
Bit 1	CLKPOL	0x0	rw	Clock polarity Indicates the polarity of clock output in idle state. 0: Low level 1: High level
Bit 0	CLKPHA	0x0	rw	Clock phase 0: Data capture starts from the first clock edge 1: Data capture starts from the second clock edge

Note: The SPI_CTRL1 register must be 0 in I^2S mode.

13.4.2 SPI control register 2 (SPI_CTRL2)

Bit	Name	Reset value	Type	Description
Bit 15: 10	Reserved	0x00	resd	Forced to 0 by hardware.
Bit 9	MDIV3EN	0x0	rw	Master clock frequency divided by 3 enable 0: Disabled 1: Enabled Note: When this bit is set, the MDIV[3: 0] becomes invalid, and the SPI clock is forced to be PCLK/3.
Bit 8	MDIV[3]	0x0	rw	Master clock frequency division Refer to the MDIV[2: 0] of the SPI_CTRL1 register.
Bit 7	TDBEIE	0x0	rw	Transmit data buffer empty interrupt enable 0: Disabled 1: Enabled
Bit 6	RDBFIE	0x0	rw	Receive data buffer full interrupt enable 0: Disabled 1: Enabled
Bit 5	ERRIE	0x0	rw	Error interrupt enable This bit controls interrupt generation when errors occur (CCERR, MMERR, ROERR and TUERR) 0: Disabled 1: Enabled

Bit 4	TIEN	0x0	rw	TI mode enable 0: TI mode disabled (Motorola mode) 1: TI mode enabled (TI mode) Note: This mode is not used in I2S mode. It must be 0 in I2S mode.
Bit 3	Reserved	0x0	resd	Kept at default value
Bit 2	HWCSOE	0x0	rw	Hardware CS output enable This bit is valid only in master mode. When this bit is set, the I/O output on the CS pin is low; when this bit is 0, the I/O input on the CS pin must be set high. 0: Disabled 1: Enabled
Bit 1	DMATEN	0x0	rw	DMA transmit enable 0: Disabled 1: Enabled
Bit 0	DMAREN	0x0	rw	DMA receive enable 0: Disabled 1: Enabled

13.4.3 SPI status register (SPI_STS)

Bit	Name	Reset value	Type	Description
Bit 15: 9	Reserved	0x00	resd	Forced to 0 by hardware
Bit 8	CSPAS	0x0	ro	CS pulse abnormal setting flag 0: CS pulse flag normal 1: CS pulse flag is set abnormally Note: This bit is used for TI slave mode. It is cleared by reading the STS register.
Bit 7	BF	0x0	ro	Busy flag 0: SPI is not busy. 1: SPI is busy.
Bit 6	ROERR	0x0	ro	Receiver overflow error 0: No overflow error 1: Overflow error occurs.
Bit 5	MMERR	0x0	ro	Master mode error This bit is set by hardware and cleared by software (read/write access to the SPI_STS register, followed by write operation to the SPI_CTRL1 register) 0: No mode error 1: Mode error occurs.
Bit 4	CCERR	0x0	rw0c	CRC error Set by hardware, and cleared by software. 0: No CRC error 1: CRC error occurred
Bit 3	TUERR	0x0	ro	Transmitter underload error Set by hardware, and cleared by software (read the SPI_STS register). 0: No underload error 1: Underload error occurs. Note: This bit is only used in I^2S mode.
Bit 2	ACS	0x0	ro	Audio channel state This bit indicates the status of the current audio channel. 0: Left channel 1: Right channel Note: This bit is only used in I^2S mode.

Bit 1	TDBE	0x1	ro	Transmit data buffer empty 0: Transmit data buffer is not empty. 1: Transmit data buffer is not empty.
Bit 0	RDBF	0x0	ro	Receive data buffer full 0: Transmit data buffer is not full. 1: Transmit data buffer is full.

13.4.4 SPI data register (SPI_DT)

Bit	Name	Reset value	Type	Description
Bit 15: 0	DT	0x0000	rw	Data value This register controls read and write operations. When the data bit is set as 8 bit, only the 8-bit LSB [7: 0] is valid.

13.4.5 SPICRC register (SPI_CPOLY) (Not used in I²S mode)

Bit	Name	Reset value	Type	Description
Bit 15: 0	CPOLY	0x0007	rw	CRC polynomial This register contains the polynomial used for CRC calculation. Note: This register is valid only in SPI mode.

13.4.6 SPIRxCRC register (SPI_RCRC) (Not used in I²S mode)

Bit	Name	Reset value	Type	Description
Bit 15: 0	RCRC	0x0000	ro	Receive CRC When CRC calculation is enabled, this register contains the CRC value computed based on the received data. This register is reset when the CCEN bit in the SPI_CTRL1 register is cleared. When the data frame format is set to 8-bit data, only the 8-bit LSB ([7: 0]) are calculated based on CRC8 standard; when 16-bit data bit is selected, follow CRC16 standard. Note: This register is only used in SPI mode.

13.4.7 SPITxCRC register (SPI_TCRC)

Bit	Name	Reset value	Type	Description
Bit 15: 0	TCRC	0x0000	ro	Transmit CRC When CRC calculation is enabled, this register contains the CRC value computed based on the transmitted data. This register is reset when the CCEN bit in the SPI_CTRL1 register is cleared. When the data frame format is set to 8-bit data, only the 8-bit LSB ([7: 0]) are calculated based on CRC8 standard; when 16-bit data bit is selected, follow CRC16 standard. Note: This register is only used in SPI mode.

13.4.8 SPI_I2S register (SPI_I2SCTRL)

Bit	Name	Reset value	Type	Description
Bit 15: 12	Reserved	0x0	resd	Forced to 0 by hardware.
Bit 11	I2SMSEL	0x0	rw	I²S mode select 0: SPI mode 1: I²S mode
Bit 10	I2SEN	0x0	rw	I²S enable 0: Disabled 1: Enabled
Bit 9: 8	OPERSEL	0x0	rw	I²S operation mode select 00: Slave transmission 01: Slave reception 10: Master transmission 11: Master reception
Bit 7	PCMFSSEL	0x0	rw	PCM frame synchronization This bit is valid only when the PCM standard is used. 0: Short frame synchronization 1: Long frame synchronization
Bit 6	Reserved	0x0	resd	Kept at default value
Bit 5: 4	STDSEL	0x0	rw	I²S standard select 00: Philips standard 01: MSB-aligned standard (left-aligned) 10: LSB-aligned standard (right-aligned) 11: PCM standard
Bit 3	I2SCLKPOL	0x0	rw	I²S clock polarity This bit indicates the clock polarity on the clock pin in idle state. 0: Low 1: High
Bit 2: 1	I2SDBN	0x0	rw	I²S data bit num 00: 16-bit data length 01: 24-bit data length 10: 32-bit data length 11: Not allowed.
Bit 0	I2SCBN	0x0	rw	I²S channel bit num This bit can be configured only when the I²S is set to 16-bit data; otherwise, it is fixed to 32-bit by hardware. 0: 16-bit wide 1: 32-bit wide

13.4.9 SPI_I2S prescaler register (SPI_I2SCLKP)

Bit	Name	Reset value	Type	Description
Bit 15: 12	Reserved	0x0	resd	Forced 0 by hardware.
Bit 9	I2SMCLKOE	0x0	rw	I²S Master clock output enable 0: Disabled 1: Enabled
Bit 8	I2SODD	0x0	rw	Odd factor for I²S division 0: Actual divider factor =I2SDIV*2 1: Actual divider factor =(I2SDIV*2)+1
Bit 11: 10 Bit 7: 0	I2SDIV	0x02	rw	I²S division It is not allowed to configure I2SDIV[9: 0]=0 or I2SDIV[9: 0]=1

14 Timer

AT32L021 timers include basic timers, general-purpose timers, and advanced timers. Please refer to *Section 14.1* ~ *Section 14.6* for details. Table 14-1 gives a list of timers of various types.

Table 14-1 TMR functional comparison

Timer type	Timer	Counter bit	Count mode	Repetition	Prescaler	DMA requests	Capture/compare channel	PWM input mode	EXT input	Brake input
Advanced-control timer	TMR1	16	Up Down Up/Down	16-bit	1~65536	O	4	O	O	O
General-purpose timer	TMR3	16	Up Down Up/Down	X	1~65536	O	4	O	O	X
	TMR14	16	Up	X	1~65536	X	1	X	X	X
	TMR15	16	Up	8-bit	1~65536	O	2	O	X	O
	TMR16 TMR17	16	Up	8-bit	1~65536	O	1	X	X	O
Basic timer	TMR6	16	Up	X	1~65536	O	X	X	X	X

Timer type	Timer	Counter bit	Count mode	PWM output	Single pulse output	Complementary output	Dead-time	Encoder interface connection	Interfacing with hall sensors	Linkage peripheral
Advanced-control timer	TMR1	16	Up Down Up/Down	O	O	O	O	O	O	Timer synchronization
General-purpose time	TMR3	16	Up Down Up/Down	O	O	X	X	O	O	Timer synchronization
	TMR14	16	Up	O	O	X	X	X	X	NA
	TMR15	16	Up	O	O	O	O	X	X	Timer synchronization
	TMR16 TMR17	16	Up	O	O	O	O	X	X	NA
Basic timer	TMR6 TMR7	16	Up	X	X	X	X	X	X	NA

14.1 Basic timer (TMR6)

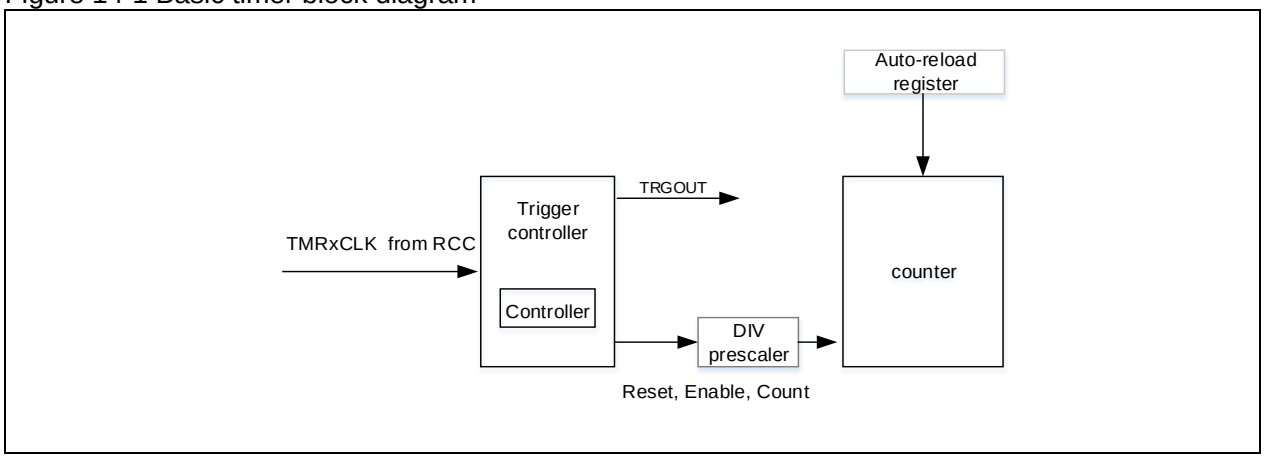
14.1.1 TMR6 introduction

TMR6 basic timer includes a 16-bit upcounter and corresponding control logic, without being connected to external I/Os. It can be used for basic timing function.

14.1.2 TMR6 main features

- 16-bit auto reload upcounter
- 16-bit prescaler used to divide TMR_CLK clock frequency by any factor between 1 and 65536

Figure 14-1 Basic timer block diagram

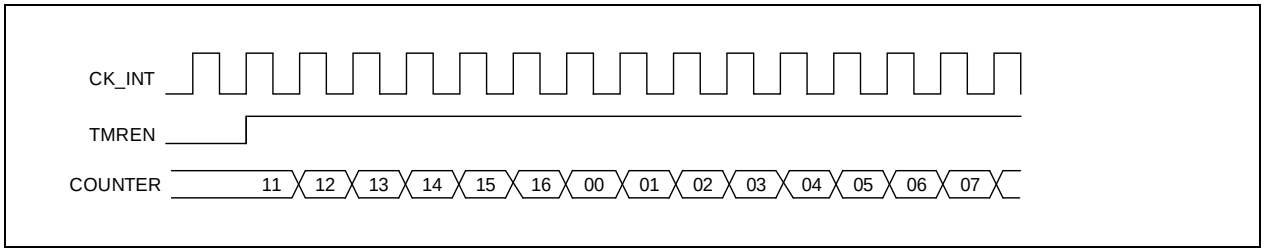


14.1.3 TMR6 function overview

14.1.3.1 Counting clock

The counter clock of TMR6 is provided by an internal clock source (CK_INT) divided by prescaler. When TMR's APB clock prescaler factor is 1, the CK_INT frequency is equal to that of APB; otherwise, it doubles the APB clock frequency.

Figure 14-2 Control circuit with CK_INT divided by 1



14.1.3.2 Counting mode

The basic timer only supports upcounting mode. It has an internal 16-bit counter.

The TMRx_PR register is used to set counting cycles of the counter. The value in the TMRx_PR is immediately moved to the shadow register by default. When the periodic buffer is enabled (PRBEN=1), the value in the TMRx_PR register is transferred to the shadow register only upon an overflow event.

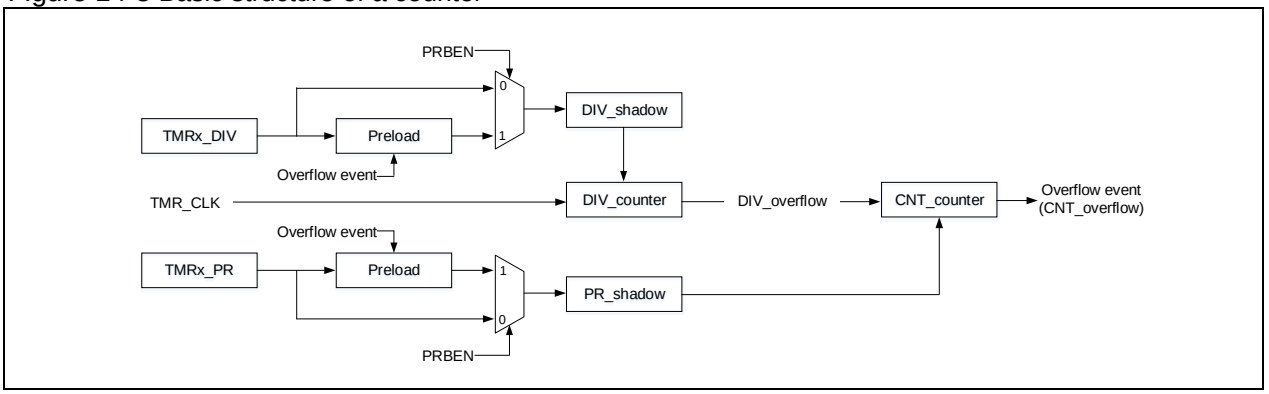
TMRx_DIV register is used to define the counting frequency of the counter. The counter counts once every DIV[15:0]+1 clock cycle. Similar to TMRx_PR register, after the periodic buffer is enabled, the value of the TMRx_DIV register is transferred into the shadow register upon an overflow event.

Reading the TMRx_CNT register returns the current counter value. Writing the TMRx_CNT register will update the current counter value.

An overflow event is enabled by default. It can be disabled by setting OVFEN=1 in the TMRx_CTRL1 register. The OVFS bit in the TMRx_CTRL1 register is used to select an overflow event source, which includes, by default, counter overflow or underflow, setting OVFSWTR, reset signal generated by slave mode timer controller in reset mode. Once the OVFS is set, an overflow event is generated only when overflow or underflow occurs.

Setting the TMREN bit (TMREN=1) enables the timer to start counting. Base on synchronization logic, however, the actual counter enable signal TMR_EN is set 1 clock cycle after the TMREN is set.

Figure 14-3 Basic structure of a counter



Upcounting mode

In upcounting mode, the counter counts from 0 to the value programmed in the TMRx_PR register, then restarts from 0 and generates a counter overflow event with setting OVFIF=1 at the same time. If the overflow event is disabled, the counter is no longer reloaded with a prescaler value and a periodic value when a counter overflow event occurs, otherwise, the counter is updated with prescaler and periodic values at an overflow event.

Figure 14-4 Overflow event when PRBEN=0

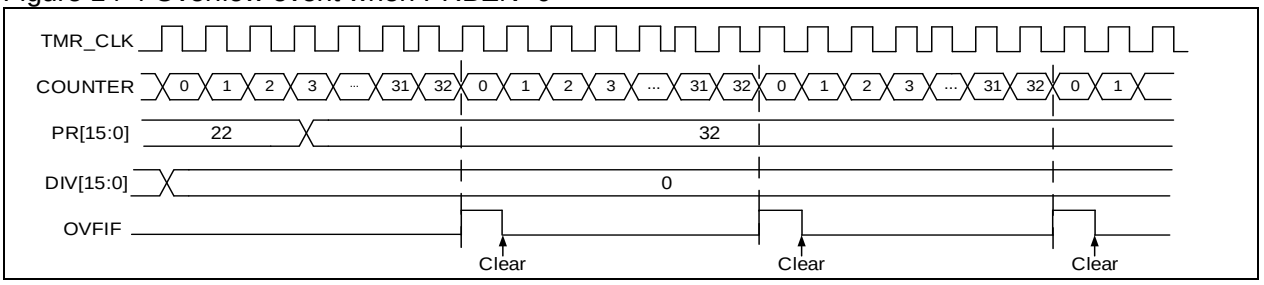


Figure 14-5 Overflow event when PRBEN=1

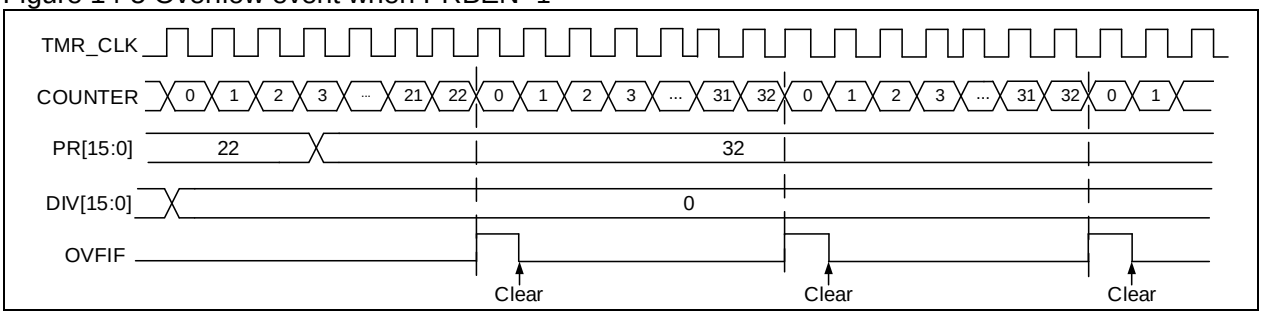
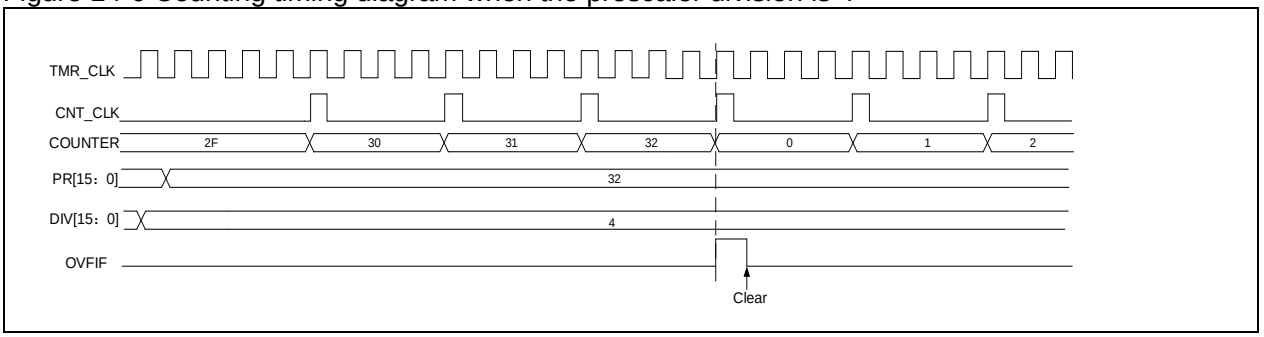


Figure 14-6 Counting timing diagram when the prescaler division is 4



14.1.3.3 Debug mode

When the microcontroller enters debug mode (Cortex®-M0+ core halted), the TMRx counter stops counting when the TMRx_PAUSE bit is set to 1.

14.1.4 TMR6 registers

These peripheral registers can be accessed by half-words (16 bits) or words (32 bits).

In Table 14-2, all the TMR6 registers are mapped to a 16-bit addressable space.

Table 14-2 TMR6 register table and reset value

Register	Offset	Reset value
TMRx_CTRL1	0x00	0x0000
TMRx_CTRL2	0x04	0x0000
TMRx_IDEN	0x0C	0x0000
TMRx_ISTS	0x10	0x0000
TMRx_SWEVT	0x14	0x0000
TMRx_CVAL	0x24	0x0000
TMRx_DIV	0x28	0x0000
TMRx_PR	0x2C	0x0000

14.1.4.1 TMR6 control register 1 (TMRx_CTRL1)

Bit	Name	Reset value	Type	Description
Bit 15: 8	Reserved	0x0	resd	Kept at default value.
Bit 7	PRBEN	0x0	rw	Period buffer enable 0: Period buffer is disabled. 1: Period buffer is enabled.
Bit 6: 4	Reserved	0x0	resd	Kept at default value.
Bit 3	OCMEN	0x0	rw	One cycle mode enable This bit is used to select whether to stop the counter at overflow event. 0: Disabled 1: Enabled
Bit 2	OVFS	0x0	rw	Overflow event source This bit is used to select overflow event or DMA request sources. 0: Counter overflow, setting the OVFSWTR bit or overflow event generated from the slave controller 1: Only counter overflow generates an overflow event.
Bit 1	OVFEN	0x0	rw	Overflow event enable This bit is used to enable or disable OEV event generation. 0: OEV event is enabled. An overflow event is generated by any of the following events: - Counter overflow - Setting the OVFSWTR bit to 1 - Overflow event generated from the slave controller 1: OEV event is disabled. If the OVFSWTR bit is set to 1, or if a hardware reset is generated from the slave mode controller, the counter and the prescaler are reinitialized. Note: This bit is set and cleared by software.
Bit 0	TMREN	0x0	rw	TMR enable 0: Disabled 1: Enabled

14.1.4.2 TMR6 control register 2 (TMRx_CTRL2)

Bit	Name	Reset value	Type	Description
Bit 15: 7	Reserved	0x0	resd	Kept at default value.
Bit 6: 4	PTOS	0x0	rw	Master TMR output selection This field is used to select the signals in master mode to be sent to slave timers. 000: Reset

				001: Enable 010: Update
Bit 3: 0	Reserved	0x0	resd	Kept at default value.

14.1.4.3 TMR6 DMA/interrupt enable register (TMRx_IDEN)

Bit	Name	Reset value	Type	Description
Bit 15: 9	Reserved	0x0	resd	Kept at default value.
Bit 8	OVFDEN	0x0	rw	Overflow event DMA request enable 0: Disabled 1: Enabled
Bit 7: 1	Reserved	0x0	resd	Kept at default value.
Bit 0	OVFIEN	0x0	rw	Overflow interrupt enable 0: Disabled 1: Enabled

14.1.4.4 TMR6 interrupt status register (TMRx_ISTS)

Bit	Name	Reset value	Type	Description
Bit 15: 1	Reserved	0x0	resd	Kept at default value.
Bit 0	OVFIF	0x0	rw0c	Overflow interrupt flag This bit is set by hardware at an overflow event. It is cleared by software. 0: No overflow event occurred. 1: Overflow event occurred, and if OVFEN=0, and OVFS=0 in the TMRx_CTRL1 register: − An overflow event occurred when OVFG=1 in the TMRx_SWEVE register − An overflow event occurred when the counter value (CVAL) is reinitialized by a trigger event.

14.1.4.5 TMR6 software event register (TMRx_SWEVT)

Bit	Name	Reset value	Type	Description
Bit 15: 1	Reserved	0x0	resd	Kept at default value.
Bit 0	OVFSWTR	0x0	rw0c	Overflow event triggered by software An overflow event is trigged by software. 0: No effect 1: Generate an overflow event by software

14.1.4.6 TMR6 counter value (TMRx_CVAL)

Bit	Name	Reset value	Type	Description
Bit 15: 0	CVAL	0x0	rw	Counter value

14.1.4.7 TMR6 divider value (TMRx_DIV)

Bit	Name	Reset value	Type	Description
Bit 15: 0	DIV	0x0	rw	Divider value The counter clock frequency fCK_CNT = fTMR_CLK / (DIV[15: 0]+1). At each overflow event, DIV value is written to the DIV register.

14.1.4.8 TMR6 period register (TMRx_PR)

Bit	Name	Reset value	Type	Description
Bit 15: 0	PR	0x0	rw	Period value This indicates the period value of the TMRx counter. The timer stops working when the period value is 0.

14.2 General-purpose timer (TMR3)

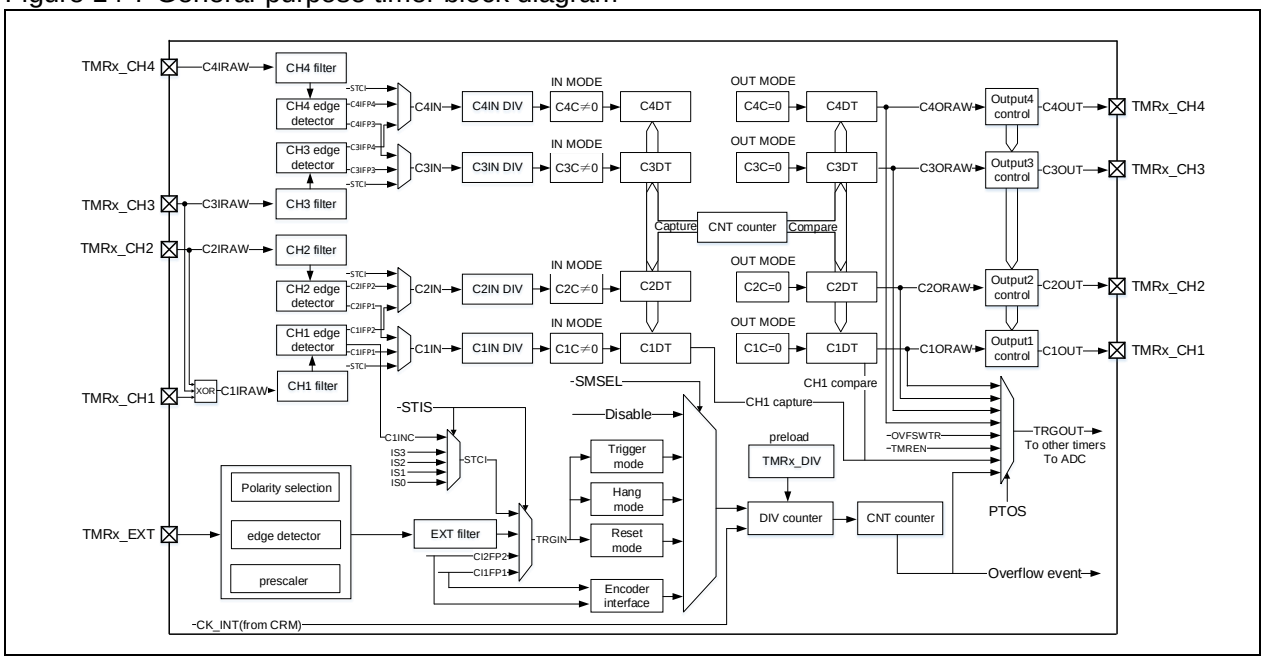
14.2.1 TMR3 introduction

The general-purpose timer (TMR3) consists of a 16-bit counter supporting up, down, up/down counting modes, four capture/compare registers, and four independent channels. It can be used for input capture and programmable PWM output.

14.2.2 TMR3 main features

- Clock source: internal clock, external clock and internal trigger inputs
- 16-bit up, down, up/down and encoder mode counter
- 4 independent channels for input capture, output compare, PWM generation and one-pulse mode output
- Synchronization control between master and slave timers
- Interrupt/DMA generation at overflow event, trigger event and channel event
- Support TMR burst DMA transfer

Figure 14-7 General-purpose timer block diagram

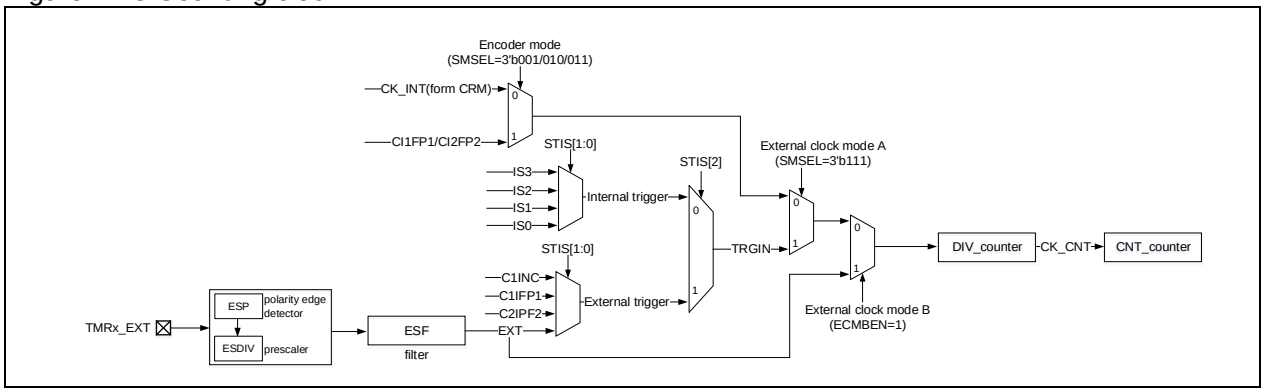


14.2.3 TMR3 functional overview

14.2.3.1 Counting clock

The counter clock of TMR3 can be provided by an internal clock (CK_INT), external clocks (external clock mode A and B) or internal trigger input (ISx)

Figure 14-8 Counting clock



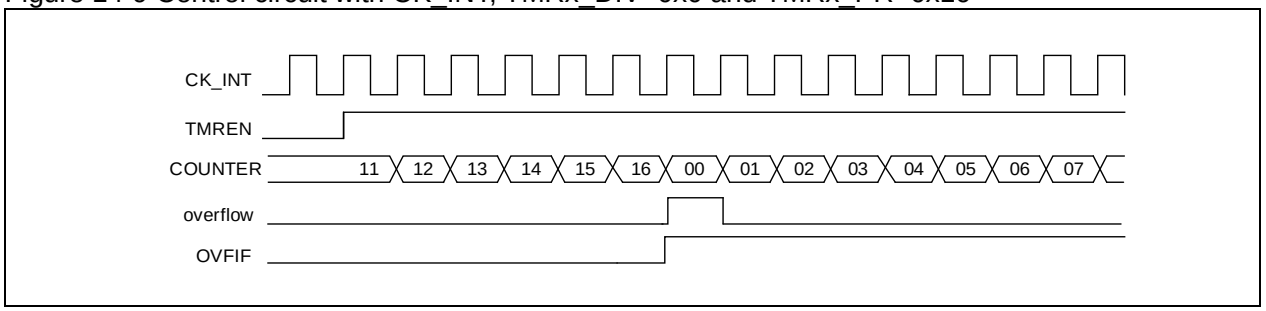
Internal clock (CK_INT)

By default, the CK_INT, which is divided by a prescaler, is used to drive the counter to count. When TMR's APB clock prescaler factor is 1, the CK_INT frequency is equal to that of APB; otherwise, it doubles the APB clock frequency.

Follow the procedures below:

- Select a counting mode by setting the TWCMSEL[1:0] in TMRx_CTRL1 register.
  If a unidirectional aligned counting mode is selected, it is necessary to select a counting direction through the OWCDIR in TMRx_CTRL1 register.
- Set counting frequency through TMRx_DIV register
- Set counting cycles through TMRx_PR register
- Enable the counter by setting the TMREN bit in the TMRx_CTRL1 register

Figure 14-9 Control circuit with CK_INT, TMRx_DIV=0x0 and TMRx_PR=0x16



External clocks (**TRGIN/EXT**)

The counter clock can be provided by two external clock sources, namely, TRGIN and EXT signals.

SMSEL=3'b111: External clock mode A is selected. Select an external clock source TRGIN signal by setting the STIS[2: 0] bit to drive the counter.

The external clock sources include:
C1INC (STIS=3'b100, channel 1 rising edge and falling edge)
C1IFP1 (STIS=3'b101, channel 1 signal which goes through filtering and polarity selection)
C2IFP2 (STIS=3'b110, channel 2 signal which goes through filtering and polarity selection)
EXT (STIS=3'b111, external input signal which goes through filtering, polarity selection and frequency division).

ECMBEN=1: External clock mode B is selected. The counter is driven by external input that has gone through polarity selection, frequency division and filtering. The external clock mode B, equivalent to the external clock mode A, selects EXT signal as an external force TRGIN.

To use external clock mode A, follow the steps below:

- Set external source TRGIN parameters

  If the TMRx_CH1 is used as a source of TRGIN, it is necessary to configure channel 1 input filter (C1DF[3:0] in TMRx_CM1 register) and channel 1 input polarity (C1P/C1CP in TMRx_CCTRL register);

  If the TMRx_CH2 is used as source of TRGIN, it is necessary to configure channel 2 input filter (C2DF[3:0] in TMRx_CM1 register) and channel 2 input polarity (C2P/C2CP in TMRx_CCTR register);

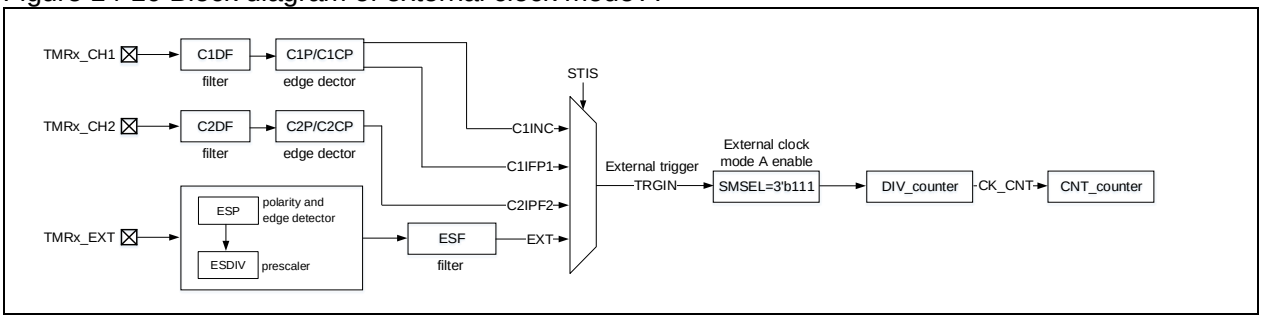
  If the TMRx_EXT is used as a source of TRGIN, it is necessary to configure the external signal polarity (ESP in TMRx_STCTRL register), external signal frequency division (ESDIV[1:0] in TMRx_STCTRL) and external signal filter (ESF[3:0] in TMRx_STCTRL register).
- Set TRGIN signal source using the STIS[1:0] bit in TMRx_STCTRL register
- Enable external clock mode A by setting SMSEL=3'b111 in TMRx_STCTR register
- Set counting frequency through the DIV[15:0] in TMRx_DIV register
- Set counting period through the PR[15:0] in TMRx_PR register
- Enable counter through the TMREN bit in TMRx_CTRL1 register

To use external clock mode B, follow the steps below:

- Set external signal polarity through the ESP bit in TMRx_STCTRL register

- Set external signal frequency division through the ESDIV[1:0] bit in TMRx_STCTRL register
- Set external signal filter through the ESF[3:0] bit in TMRx_STCTRL register
- Enable external clock mode B through the ECMBEN bit in TMRx_STCTR register
- Set counting frequency through the DIV[15:0] bit in TMRx_DIV register
- Set counting period through the PR[15:0] bit in TMRx_PR register
- Enable counter through the TMREN in TMRx_CTRL1 register

Figure 14-10 Block diagram of external clock mode A



Note: The delay between the signal on the input side and the actual clock of the counter is due to the synchronization circuit.

Figure 14-11 Counting in external clock mode A, PR=0x32 and DIV=0x0

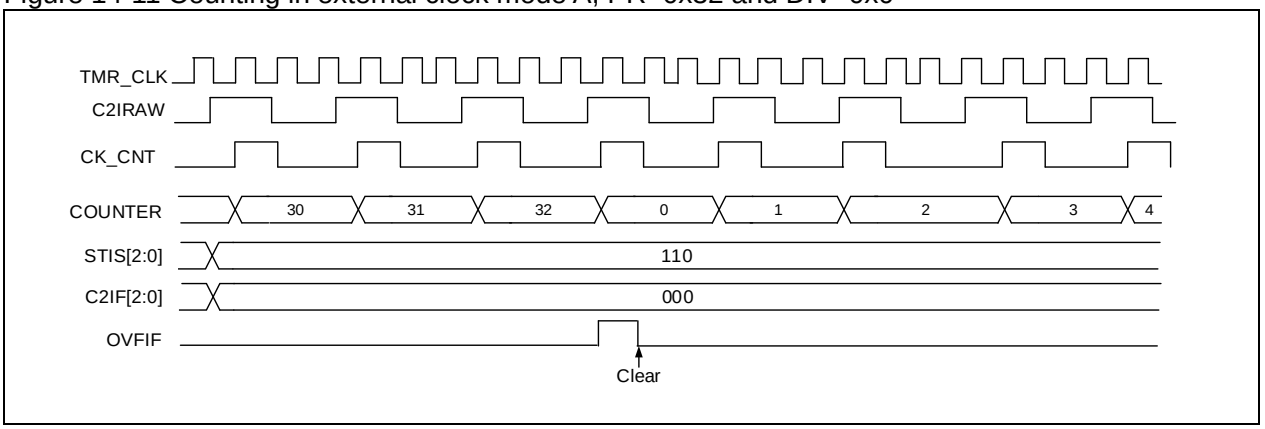
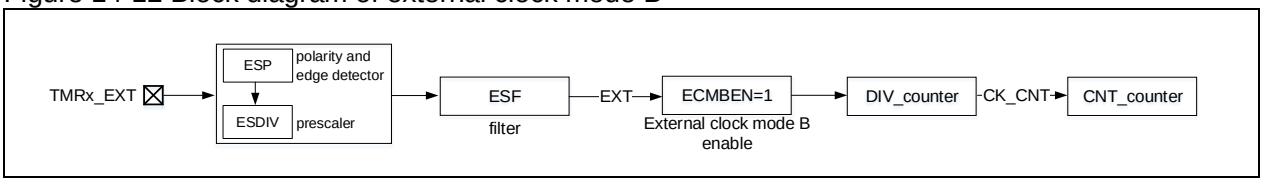
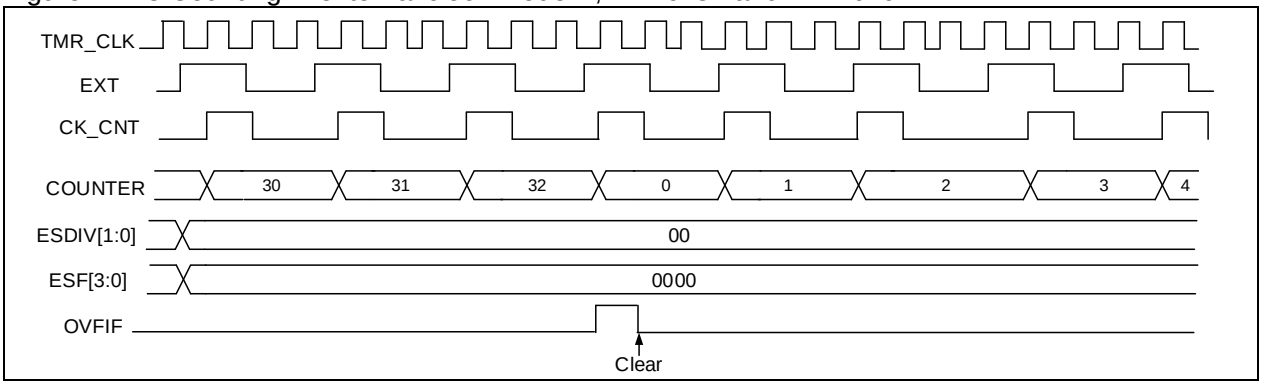


Figure 14-12 Block diagram of external clock mode B



Note: The delay between the EXT signal on the input side and the actual clock of the counter is due to the synchronization circuit.

Figure 14-13 Counting in external clock mode B, PR=0x32 and DIV=0x0



Internal trigger input (ISx)

Timer synchronization allows interconnection between several timers. The TMR_CLK of one timer can be provided by TRGOUT signal from another timer. The internal trigger signal is selected by setting the

STIS[2: 0] bit to enable counting.

TMR3 consists of a 16-bit prescaler, which is used to generate CK_CNT that enables the counter to count. The frequency division relationship between the CK_CNT and TMR_CLK can be adjusted by setting the TMRx_DIV register. The prescaler value can be modified at any time, but the new prescaler value is taken into account when the next overflow event occurs.

Below is the configuration procedure for internal trigger input:

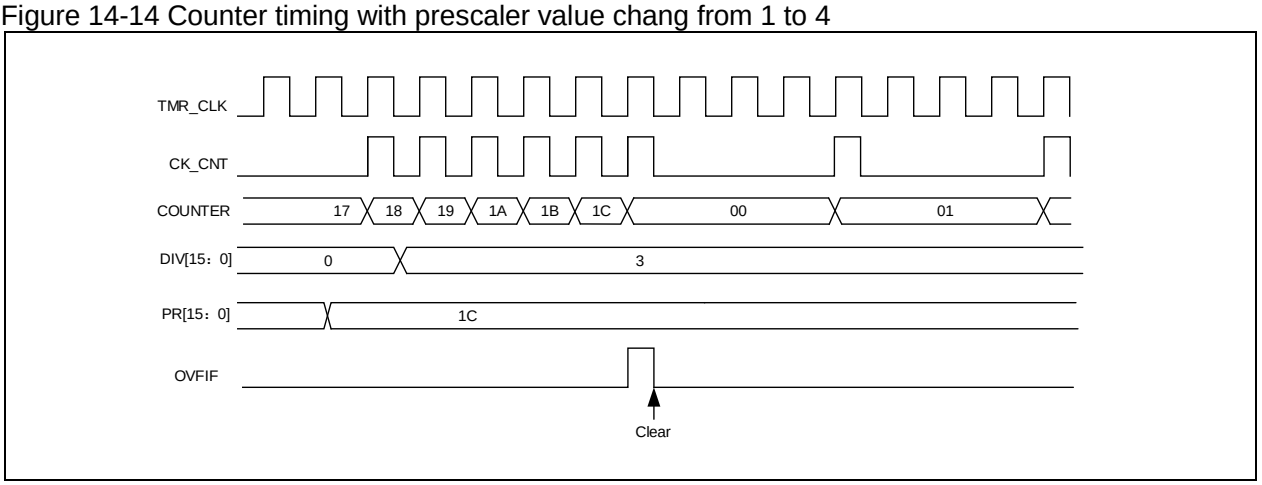
- Set counting cycles through TMRx_PR register
- Set counting frequency through TMRx_DIV register
- Set counting modes through the TWCMSEL[1:0] in TMRx_CTRL1 register
- Select internal trigger by setting STIS[2:0]= 3'b000~3'b011 in TMRx_STCTRL register
- Select external clock mode A by setting SMSEL[2:0]=3'b111 in TMRx_STCTRL register
- Enable TMRx to start counting through the TMREN in TMRx_CTRL1 register

Table 14-3 TMRx internal trigger connection

Slave controller	IS0 (STIS = 000)	IS1 (STIS = 001)	IS2 (STIS = 010)	IS3 (STIS = 011)
TMR1	TMR15		TMR3	
TMR3	TMR1		TMR15	
TMR15		TMR3	TMR16_OC	TMR17_OC

Note 1: If there is no corresponding timer in a device, the corresponding trigger signal ISx is not present.

Figure 14-14 Counter timing with prescaler value chang from 1 to 4



14.2.3.2 Counting mode

The timer TMR3 supports several counting modes to meet different application scenarios as it has an internal 16-bit counter supporting up, down, up/down counting.

The TMRx_PR register is used to set counting cycles of the counter. The value in the TMRx_PR is immediately moved to the shadow register by default. When the periodic buffer is enabled (PRBEN=1), the value in the TMRx_PR register is transferred to the shadow register only upon an overflow event.

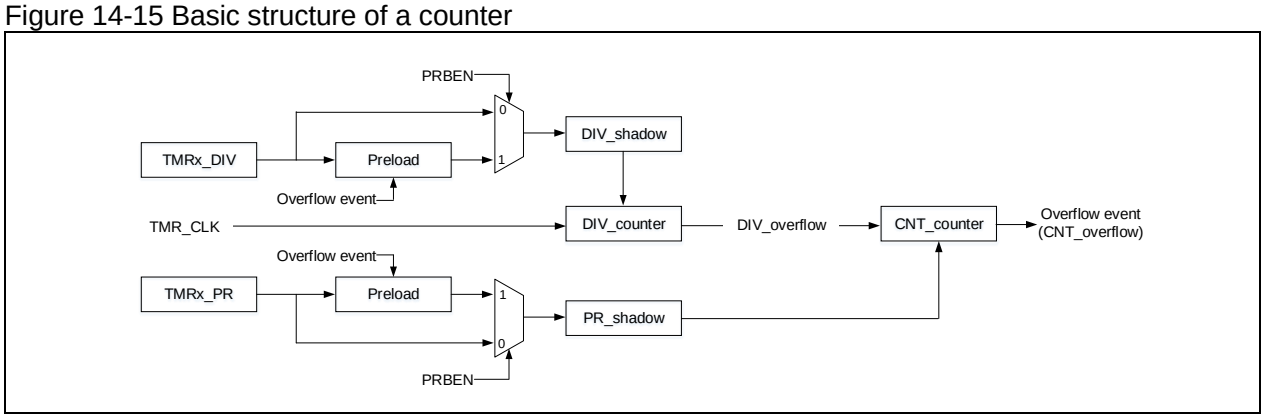
TMRx_DIV register is used to define the counter frequency of the counter. The counter counts once every DIV[15:0]+1 clock cycle. Similar to TMRx_PR register, after enabling periodic buffer, the value of the TMRx_DIV register are transferred into the shadow register upon an overflow event.

Reading the TMRx_CNT register returns the current counter value. Writing the TMRx_CNT register will update the current counter value.

An overflow event is enabled by default. It can be disabled by setting OVFEN=1 in the TMRx_CTRL1 register. The OVFS bit in the TMRx_CTRL1 register is used to select an overflow event source, which is, by default, counter overflow or underflow, setting OVFSWTR, or reset signal generated by slave mode timer controller reset mode. Once the OVFS is set, an overflow event is generated only when overflow or underflow occurs.

Setting the TMREN bit (TMREN=1) enables the timer to start counting. Base on synchronization logic, however, the actual enable signal TMR_EN is set 1 clock cycle after the TMREN is set.

Figure 14-15 Basic structure of a counter



Upcounting mode

This mode is enabled by setting CMSEL[1:0]=2'b00 and OWCDIR=1'b0 in the TMRx_CTRL1 register.

In upcounting mode, the counter counts from 0 to the value programmed in the TMRx_PR register, restarts from 0, and generates a counter overflow event, with setting OVFIF=1. If the overflow event is disabled, the register is no longer reloaded with the prescaler and periodic value after counter overflow occurs, otherwise, the prescaler and periodic value will be updated at an overflow event.

Figure 14-16 Overflow event when PRBEN=0

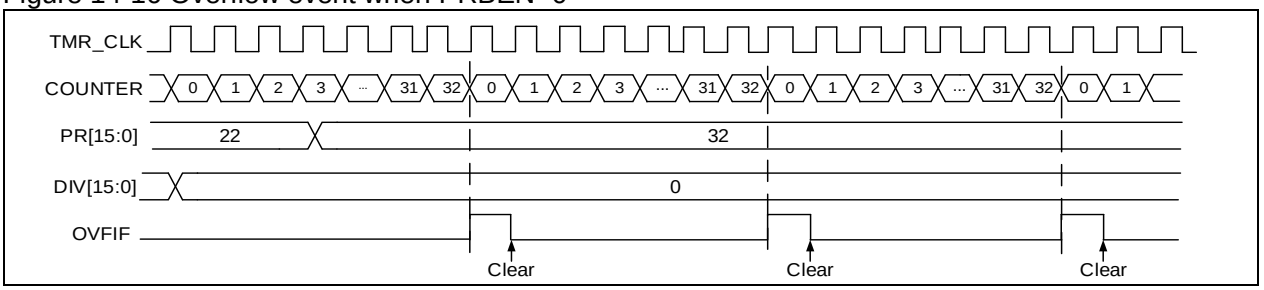
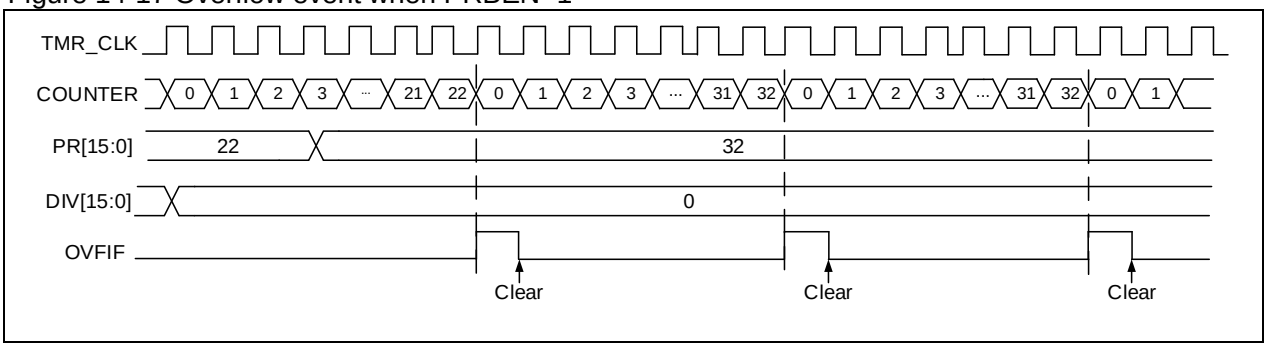


Figure 14-17 Overflow event when PRBEN=1

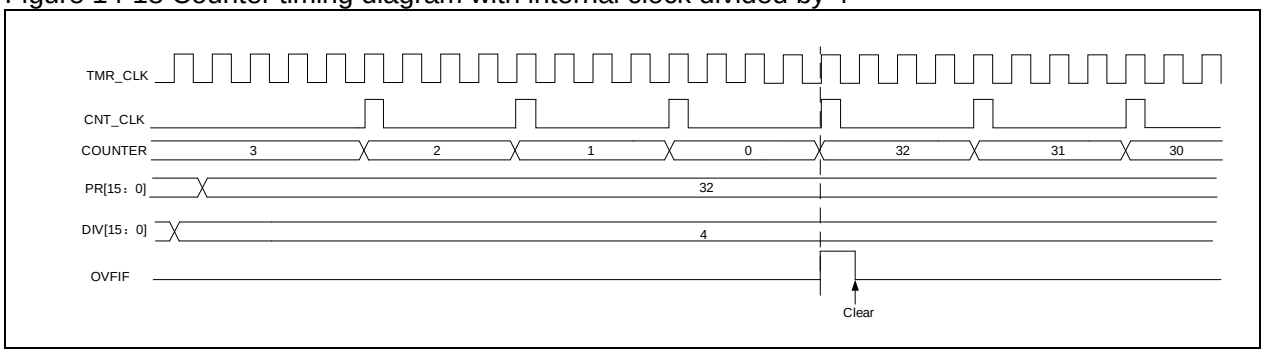


Downcounting mode

This mode is enabled by setting CMSEL[1:0]=2'b00 and OWCDIR=1'b1 in the TMRx_CTRL1 register.

In downcounting mode, the counter counts from the value programmed in the TMRx_PR register down to 0, and restarts from the value programmed, and generates a counter underflow event.

Figure 14-18 Counter timing diagram with internal clock divided by 4



Up/down counting mode (center-aligned mode)

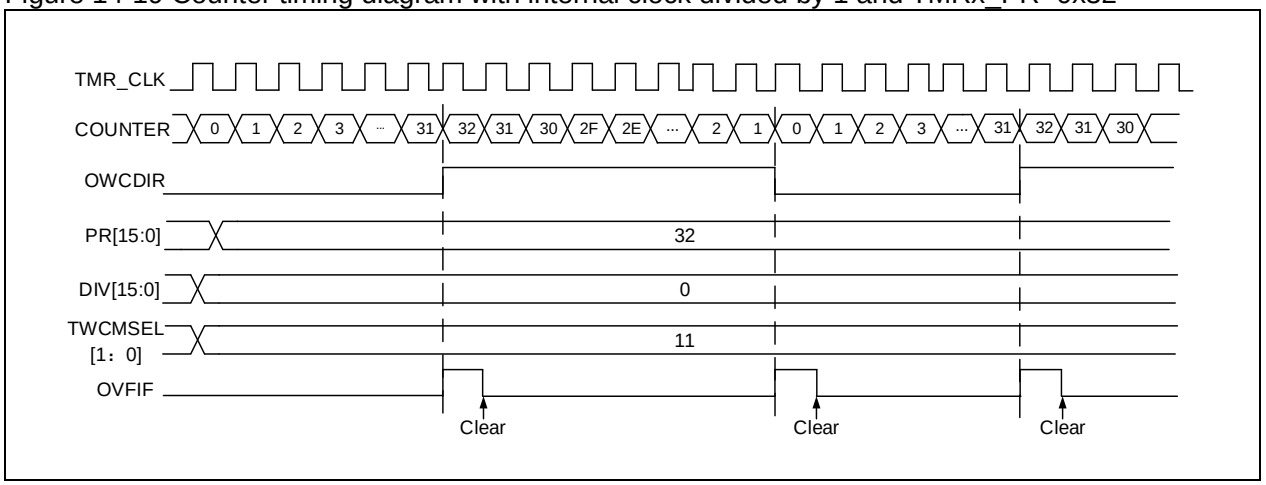
This mode is selected by setting CMSEL[1:0]≠2'b00 in the TMRx_CTRL1 register.

In up/down counting mode, the counter counts up/down alternatively. When the counter counts from the value programmed in the TMRx_PR register down to 1, an underflow event is generated, and then restarts counting from 0; when the counter counts from 0 to the value of the TMRx_PR register -1, an overflow event is generated, and then restarts counting from the value of the TMRx_PR register. The OWCDIR bit indicates the current counting direction.

The TWCMSEL[1:0] bit in the TMRx_CTRL1 register is used to select the condition under which the CxIF flag is set in two-way counting mode. In other words, when TWCMSEL[1:0]=2'b01 (counting mode 1) is selected, the CxIF flag is set only when the counter counts down; when TWCMSEL[1:0]=2'b10 (counting mode 2) is selected, the CxIF flag is set only when the counter counts up; when TWCMSEL[1:0]=2'b11 (counting mode 3) is selected, the CxIF flag is set when the counter counts up and down.

Note: The OWCDIR is ready-only in up/down counting mode.

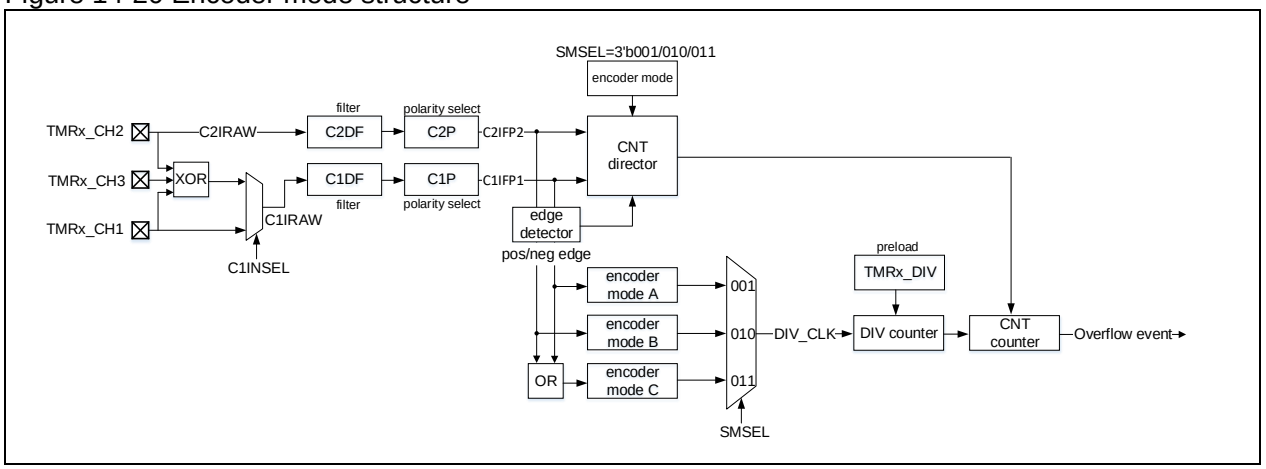
Figure 14-19 Counter timing diagram with internal clock divided by 1 and TMRx_PR=0x32



Encoder interface mode

In this mode, the two input (TMRx_CH1 and TMRx_CH2) signals are required. Depending on the level on one input, the counter counts up or down on the edge of the other input signal. The OWCDIR bit indicates the direction of the counter, as shown in the table below:

Figure 14-20 Encoder mode structure



Encoder mode A: SMSEL=3'b001. The counter counts on the selected C1IFP1 edge (rising and falling edges), and the counting direction is dependent on the edge direction of C1IFP1 and the level of C2IFP2.

Encoder mode B: SMSEL=3'b010. The counter counts on the selected C2IFP2 edge (rising and falling edges), and the counting direction is dependent on the edge direction of C2IFP2 and the level of C1IFP1.

Encoder mode C: SMSEL=3'b011. The counter counts on both C1IFP1 and C2IFP2 edges (rising and falling edges). The counting direction is dependent on the C1IFP1 edge direction and C2IFP2 level, and C2IFP2 edge direction and C1IFP1 level.

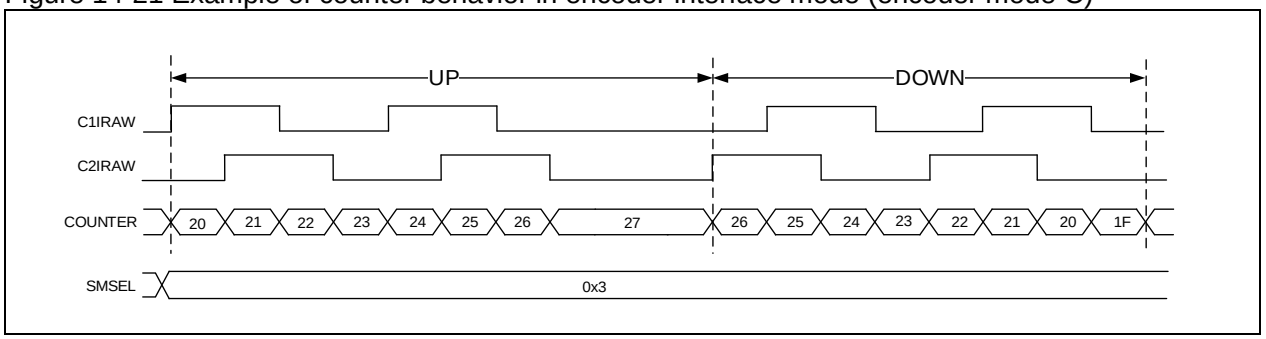
To use encoder mode, follow the procedures below:

- Set channel 1 input signal filtering through the C1DF[3:0] bit in the TMRx_CM1 register;
  Set channel 1 input signal active level through the C1P bit in the TMRx_CCTRL register
- Set channel 2 input signal filtering through the C2DF[3:0] bit in the TMRx_CM1 register;
  Set channel 2 input signal active level through the C2P bit in the TMRx_CCTRL register
- Set channel 1 as input mode through the C1C[1:0] bit in the TMRx_CM1 register;
  Set channel 2 as input mode through the C2C[1:0] bit in the TMRx_CM1 register
- Select encoder mode A (SMSEL=3'b001), encoder mode B (SMSEL=3'b010), or encoder mode C (SMSEL=3'b011) by setting the SMSEL[2:0] bit in the TMRx_STCTRL register
- Set counting cycles through the PR[15:0] bit in the TMRx_PR register
- Set counting frequency through the DIV[15:0] bit in the TMRx_DIV register
- Configure the corresponding IOs of TMRx_CH1 and TMRx_CH2 as multiplexed mode
- Enable counter through the TMREN bit in the TMRx_CTRL1 register

Table 14-4 Counting direction versus encoder signals

Active edge	Level on opposite signal (C1IFP1 to C2IFP2, C2IFP2 to C1IFP1)	C1IFP1 signal		C2IFP2 signal	
		Rising	Falling	Rising	Falling
Count on C1IFP1 only	High	Down	Up	No count	No count
	Low	Up	Down	No count	No count
Count on C2IFP2 only	High	No count	No count	Up	Down
	Low	No count	No count	Down	Up
Count on both C1IFP1 and C2IFP2	High	Down	Up	Up	Down
	Low	Up	Down	Down	Up

Figure 14-21 Example of counter behavior in encoder interface mode (encoder mode C)



14.2.3.3 TMR input function

TMR3 has four independent channels, with each being configured as input or output. As input, each channel input signal is processed as follows:

- TMRx_CHx outputs the pre-processed CxIRAW. The C1INSE bit is used to select the source of C1IRAW from TMRx_CH1, or XOR-ed TMRx_CH1, TMRx_CH2 and TMRx_CH3.

  The sources of C2IRAW, C3IRAW and C4IRAW are TMRx_CH2, TMRx_CH3 and TMRx_CH4 respectively.
- CxIRAW passes digital filter and outputs a filtered CxIF signal. The digital filter uses the CxDF bit to set sampling frequency and sampling times.
- CxIF passes edge detector, and outputs the CxIFPx signal with edge selection. The edge selector is controlled by CxP and CxCP bits. It provides input rising edge, falling edge and both edges for selection.
- CxIFPx passes capture signal selector, and outputs the CxIN signal with capture signal selection. The capture signal selector is controlled by CxC bit. The CxIN source can be from CxIFPx, CyIFPx or STCI. Of those, CyIFPx (x≠y) refers to the CyIFPy signal which is from Y channel and has passed channel-x edge detector (for example, C1IFP2 refers to the C1IFP1 on channel 1 that has passed channel 2 edge detector). The STCI comes from slave timer controller, and its source is defined by STIS bit.
- CxIN passes input divider and outputs CxIPS signal. The divider factor can be defined as No division, divided/2, divided /4 or divided /8, through the CxIDIV bit.

Figure 14-22 Input/output channel 1 main circuit

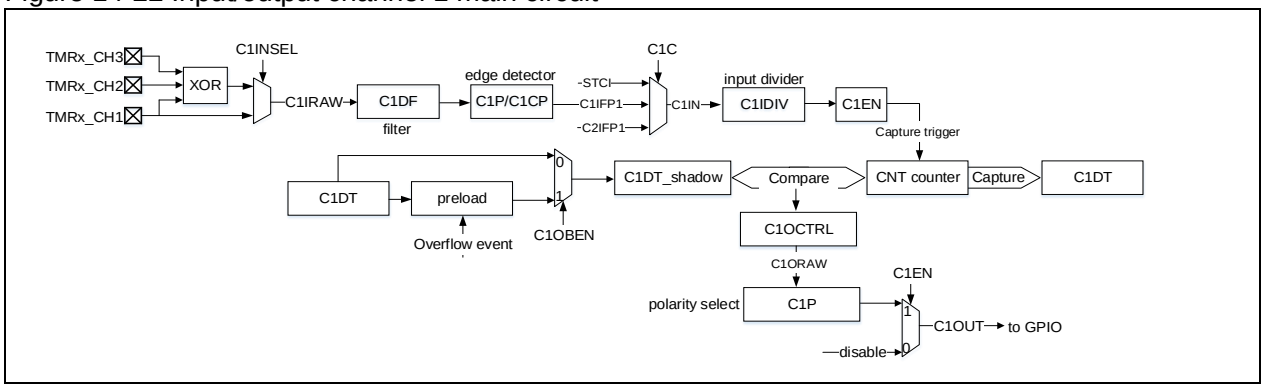
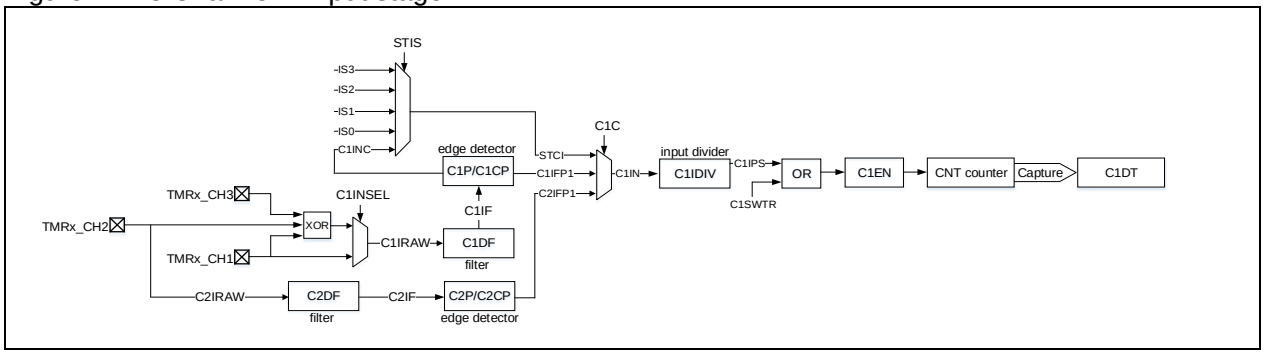


Figure 14-23 Channel 1 input stage



Input mode

In input mode, the TMRx_CxDT register latches the current counter values after the selected trigger signal is detected, and the capture compare interrupt flag bit (CxIF) is set to 1. An interrupt or a DMA request will be generated if the CxIEN and CxDEN bits are enabled. If the selected trigger signal is detected after the CxIF is set to 1, a capture overflow event is generated, and the previous counter value will be overwritten by the current counter value, setting CxRF to 1 at the same time.

To capture the rising edge of C1IN input, follow the procedure below

- Set C1C=01 in the TMRx_CM1 register to select the C1IN (channel 1 input)
- Set C1IN signal filter bandwidth (using CxDF[3: 0])
- Set the active edge on the C1IN channel by writing C1P=0 (rising edge) in the TMRx_CCTRL register
- Program C1IN signal capture frequency divider (C1DIV[1: 0])
- Enable channel 1 input capture (C1EN=1)
- If needed, enable the relevant interrupt or DMA request by setting the C1IEN bit in the TMRx_IDEN register or the C1DEN bit in the TMRx_IDEN register

Timer input XOR function

The 3 timer input pins (TMRx_CH1, TMRx_CH2 and TMRx_CH3) are connected to the channel 1 through an XOR gate. This function is selected by setting the C1INSE in the TMRx_CTRL2 register.

The XOR gate can be connect to Hall sensors. For example, connect the three XOR-ed inputs to the three Hall sensors respectively so as to calculate the position and speed of the rotation by analyzing three Hall sensor signals.

PWM input

PWM input mode is applied to channel 1 and 2. To use this mode, both C1IN and C2IN have to be mapped to the same TMRx_CHx, and the CxIFPx of either channel 1 or channel 2 must be configured as slave timer controller reset mode.

The PWM input mode can be used to measure the period and duty cycle of PWM input signals. For example, the user can measure the period and duty cycle of the PWM applied on channel 1 using the following procedures:

- Set C1C=2‘b01: select C1IN for C1IFP1
- Set C1P=1’b0, select C1IFP1 rising edge active
- Set C2C=2‘b10, select C2IN for C1IFP2
- Set C2P=1’b1, select C1IFP2 falling edge active
- Set STIS=3’b101, select C1IFP1 as slave timer trigger signal
- Set SMSEL=3‘b100: configure the slave timer controller reset mode
- Set C1EN=1’b1 and C2EN=1’b1. Enable channel 1 and input capture

After above configuration, the rising edge of channel 1 input signal will trigger the capture and stores the capture value into C1DT register, while the channel 1 input signal rising edge resets the counter. The falling edge of the channel 1 input signal triggers the capture and stores the capture value into C2DT register. The period of the channel 1 input signal is calculated through C1DT, while its duty cycle is obtained through C2DT.

Figure 14-24 PWM input mode configuration example

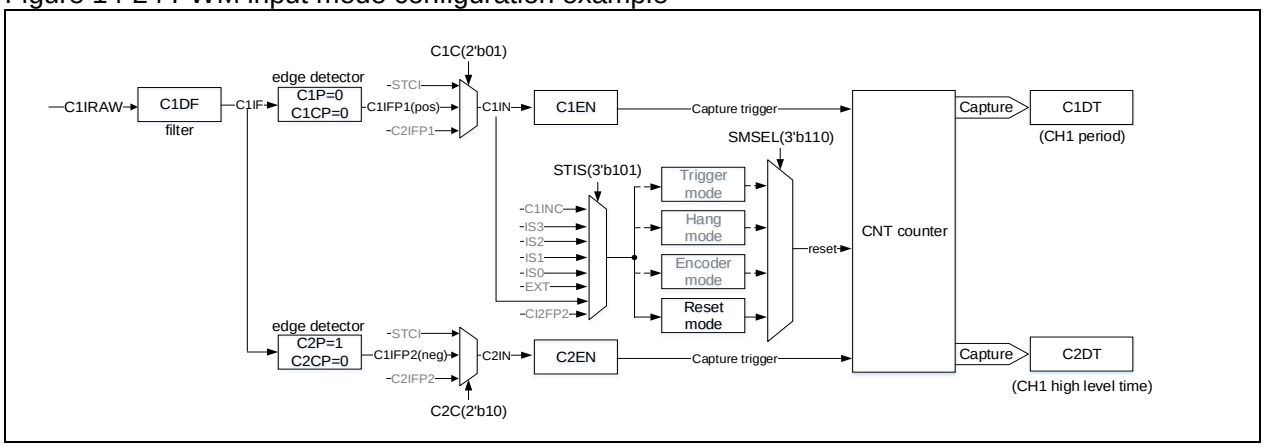
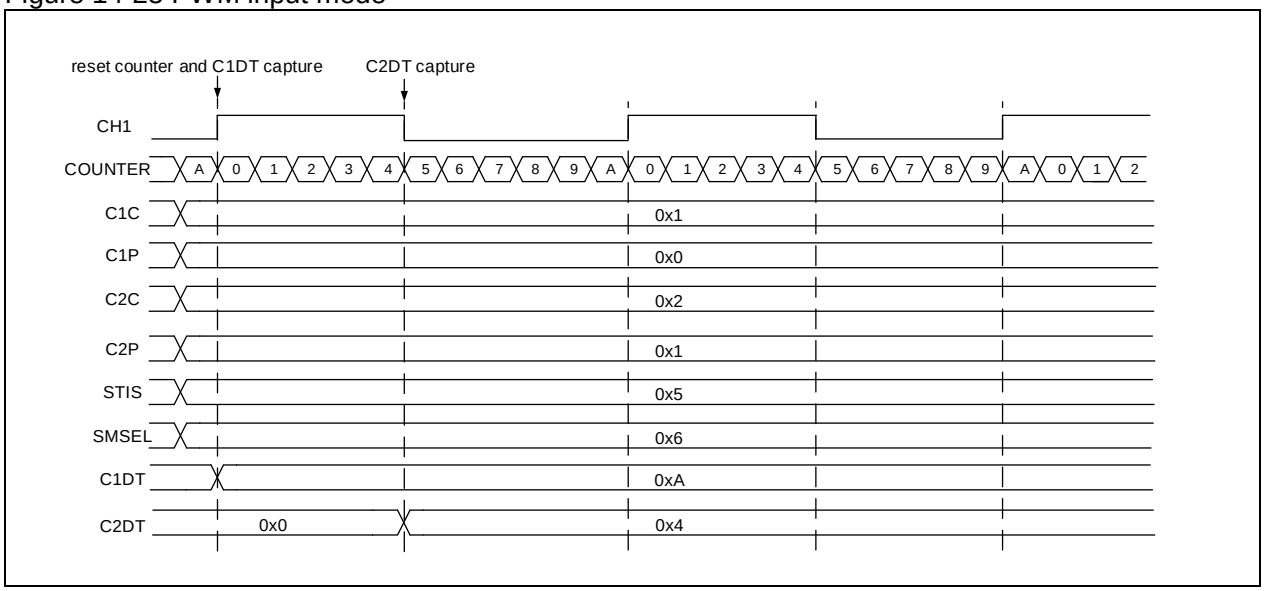


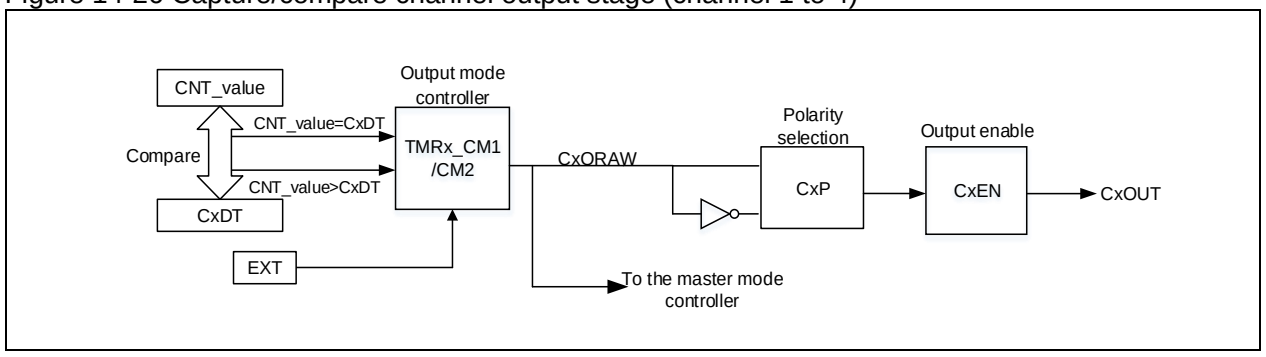
Figure 14-25 PWM input mode



14.2.3.4 TMR output function

The TMR output consists of a comparator and an output controller. It is used to program the period, duty cycle and polarity of output signals.

Figure 14-26 Capture/compare channel output stage (channel 1 to 4)



Output mode

CxC[1: 0]≠2'b00 is used to configure the channel as output mode. In this case, the counter value is compared with that in the CxDT register, and the intermediate signal CxORAW is generated according to the output mode selected by CxOCTRL[2: 0], and this signal is sent to IO after being processed by the output control circuit. The period of the output signal is configured by the TMRx_PR register, while the duty cycle is configured by the CxDT register.

Output compare modes include:

PWM mode A:

Enable PWM mode A by setting CxOCTRL=3'b110. In upcounting mode, C1ORAW outputs high

when TMRx_C1DT>TMRx_CVAL, otherwise, it is low; in downcounting mode, C1ORAW outputs low when TMRx_C1DT<TMRx_CVAL, otherwise, it is high.

To use PWM mode A, the following procedures are recommended:

- Set PWM period through TMRx_PR register
- Set PWM duty cycles through TMRx_CxDT
- Select PWM mode A by setting CxOCTRL=3'b110 in the TMRx_CM1/CM2 register
- Set counting frequency through TMRx_DIV register
- Select counting mode by setting the TWCMSEL[1:0] bit in the TMRx_CTRL1 register
- Select output polarity through the CxP and CxCP bits in the TMRx_CCTRL register
- Enable channel output through the CxEN and CxCEN bits in the TMRx_CCTRL register
- Enable TMRx output through the OEN bit in the TMRx_BRK register
- Configure GPIOs corresponding to TMR output channels as multiplexed mode
- Enable TMRx to start counting through the TMREN bit in the TMRx_CTRL1 register.

PWM mode B:

Enable PWM mode B by setting CxOCTRL=3'b111. In upcounting mode, C1ORAW outputs low when TMRx_C1DT>TMRx_CVAL, otherwise, it is high; in downcounting mode, C1ORAW outputs high when TMRx_C1DT<TMRx_CVAL, otherwise, it is low.

Forced output mode:

Enable forced output mode by setting CxOCTRL=3'b100/101. In this mode, the CxORAW is forced to be the programmed level, regardless of the counter value. Despite this, the channel flag bit and DMA request still depend on the compare result.

Output compare mode:

Enable output compare mode by setting CxOCTRL=3'b001/010/011. In this mode, when the counter value matches the value of the CxDT register, the CxORAW is forced high (CxOCTRL=3'b001), low (CxOCTRL=3'b010) or toggled (CxOCTRL=3'b011).

One-pulse mode:

This is a particular case of PWM mode. Enable one-pulse by setting OCMEN=1. In this mode, the comparison match is performed in the current counting period. The TMREN bit is cleared as soon as the current counting is completed. Therefore, only one pulse is output. In upcounting mode, the configuration must follow the rule: CVAL<CxDT≤PR; in downcounting mode, CVAL>CxDT is respected.

Fast output mode:

Enable this mode by setting CxOIEN=1. If enabled, the CxORAW signal will not change when the counter value matches the CxDT, but change at the start of the current counting period. In other words, the comparison result is advanced, so the comparison result between the counter value and the TMRx_CxDT register will determine the level of CxORAW in advance.

Figure 14-27 gives an example of output compare mode (toggle) with C1DT=0x3. When the counter value is equal to 0x3, C1OUT is toggled.

Figure 14-28 gives an example of the combination between upcounting mode and PWM mode A. The output signal behaves when PR=0x32 but CxDT is configured with a different value.

Figure 14-29 gives an example of the combination between up/down counting mode and PWM mode A. The output signal behaves when PR=0x32 but CxDT is configured with a different value.

Figure 14-30 gives an example of the combination between upcounting mode and one-pulse PWM mode B. The counter only counts only one cycle, and the output signal sends only one pulse.

Figure 14-27 C1ORAW toggles when counter value matches the C1DT value

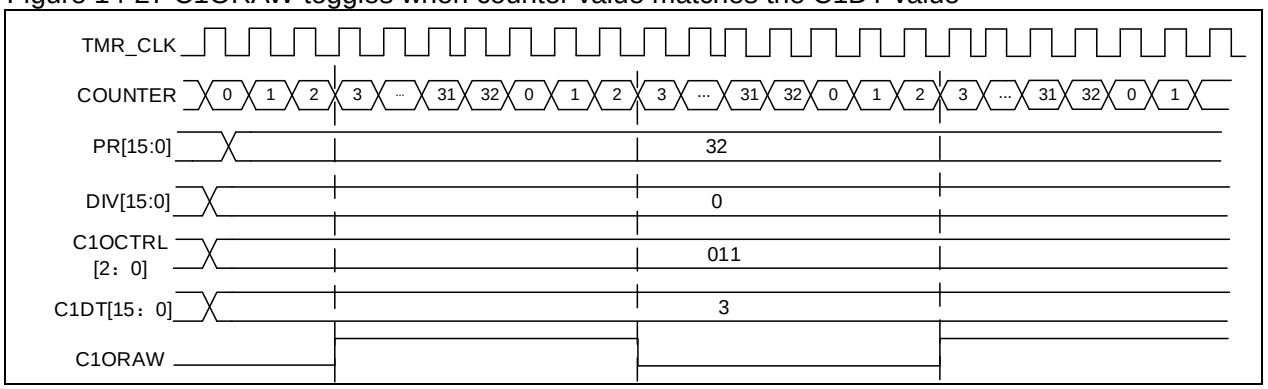


Figure 14-28 Upcounting mode and PWM mode A

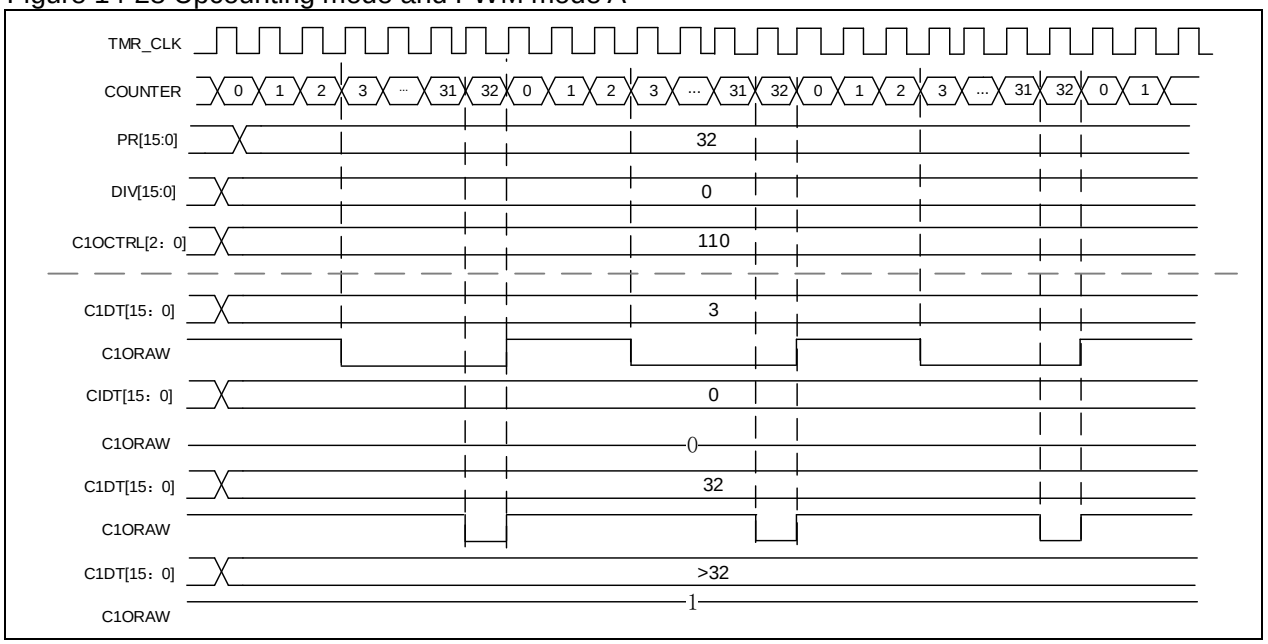


Figure 14-29 Up/down counting mode and PWM mode A

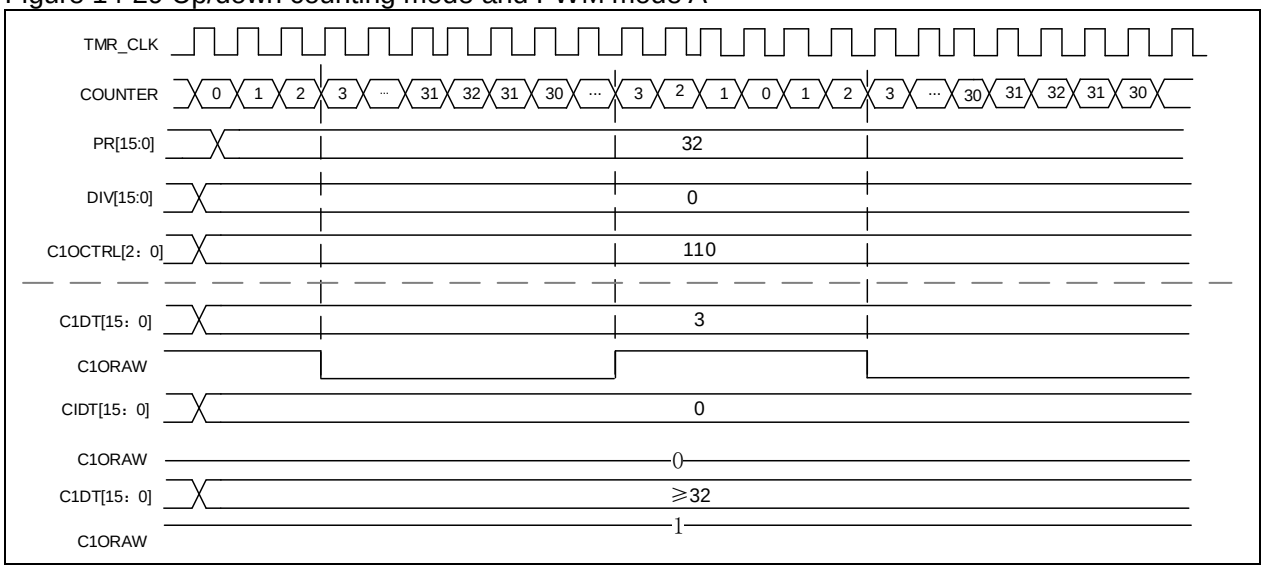
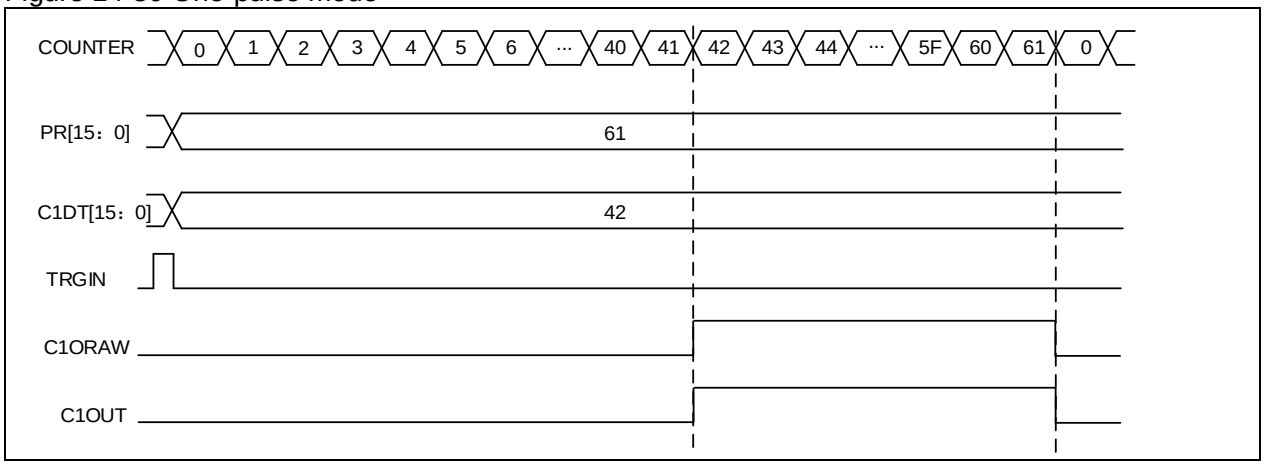


Figure 14-30 One-pulse mode



Master mode timer event output

When TMR is used as a master timer, one of the following source of signals can be selected as TRGOUT output to a slave mode timer. This is done by setting the PTOS bit in the TMRxCTRL2 register.

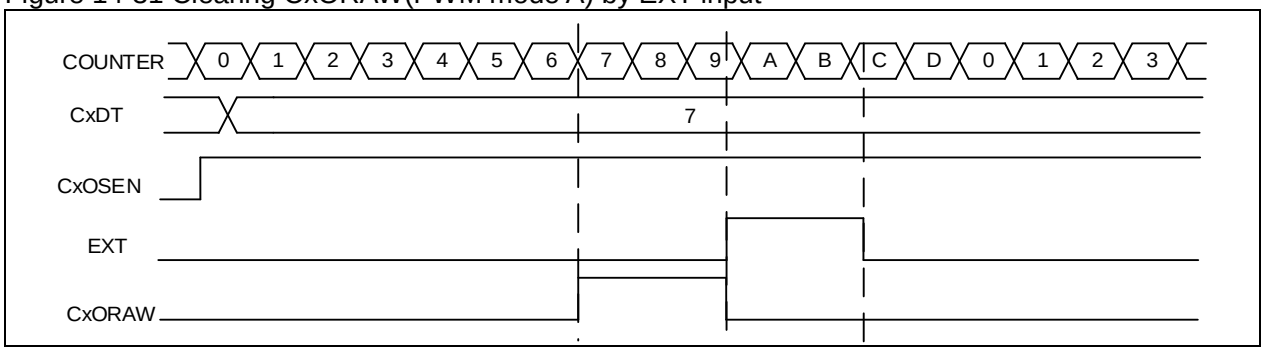
- PTOS=3’b000, TRGOUT output software overflow event (OVFSWTR bit/TMRx_SWEVT register)
- PTOS=3’b001, TRGOUT output counter enable
- PTOS=3’b010, TRGOUT output counter overflow event
- PTOS=3’b011, TRGOUT output capture and compare event
- PTOS=3’b100, TRGOUT output C1ORAW
- PTOS=3’b101, TRGOUT output C2ORAW
- PTOS=3’b110, TRGOUT output C3ORAW
- PTOS=3’b111, TRGOUT output C4ORAW

CxORAW clear

After the CxOSEN bit is set to 1, the CxORAW signal for a given channel is cleared by applying a high level to the EXT input. The CxORAW signal remains unchanged until the next overflow event.

This function can only be used in output capture or PWM modes, and does not work in forced mode. *Figure 14-31* shows the example of clearing CxORAW signal. When the EXT input is high, the CxORAW signal, which was originally high, is driven low; when the EXT is low, the CxORAW signal outputs the corresponding level according to the comparison result between the counter value and CxDT value.

Figure 14-31 Clearing CxORAW(PWM mode A) by EXT input



14.2.3.5 TMR synchronization

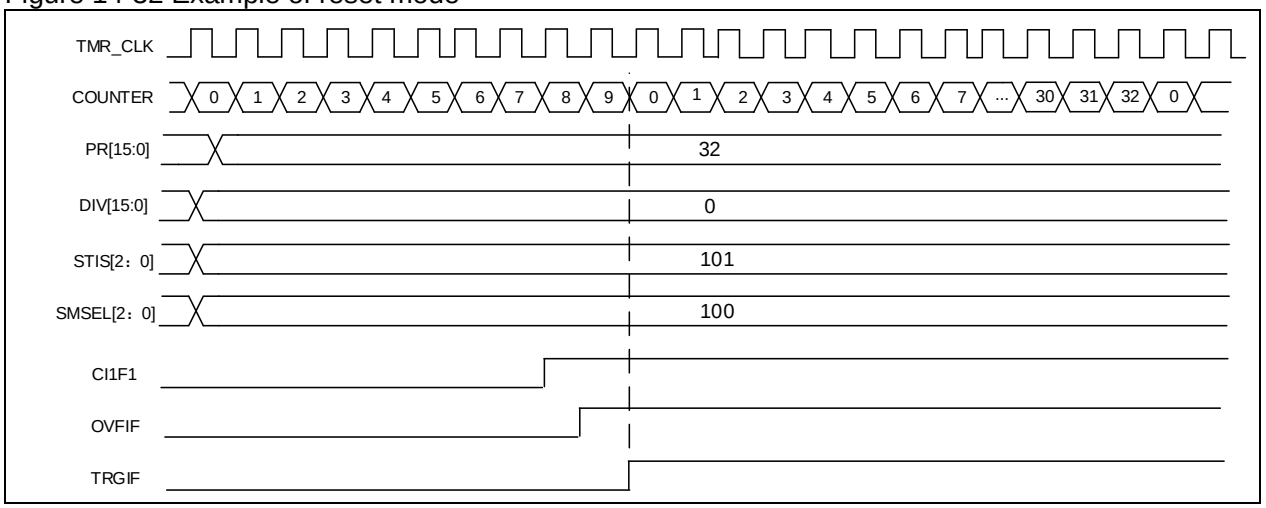
The timers are linked together internally for timer synchronization. Master timer is selected by setting the PTOS[2: 0] bit; Slave timer is selected by setting the SMSEL[2: 0] bit.

Slave modes include:

Slave mode: Reset mode

The counter and its prescaler can be reset by a selected trigger signal. An overflow event is generated when OVFS=0.

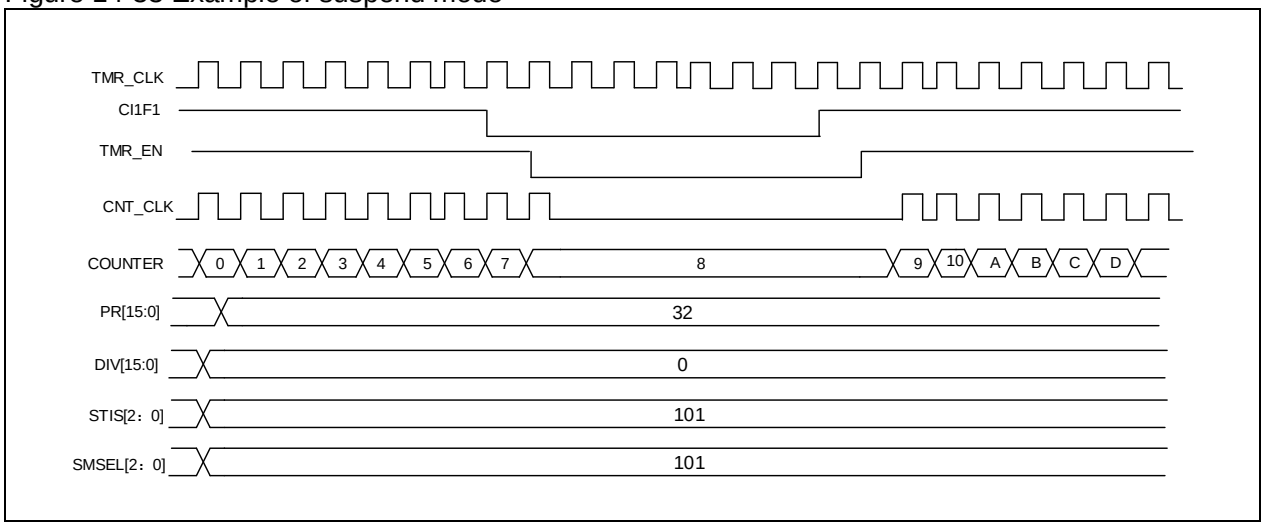
Figure 14-32 Example of reset mode



Slave mode: Suspend mode

In this mode, the counter is controlled by a selected trigger input. The counter starts counting when the trigger input is high and stops as soon as the trigger input is low.

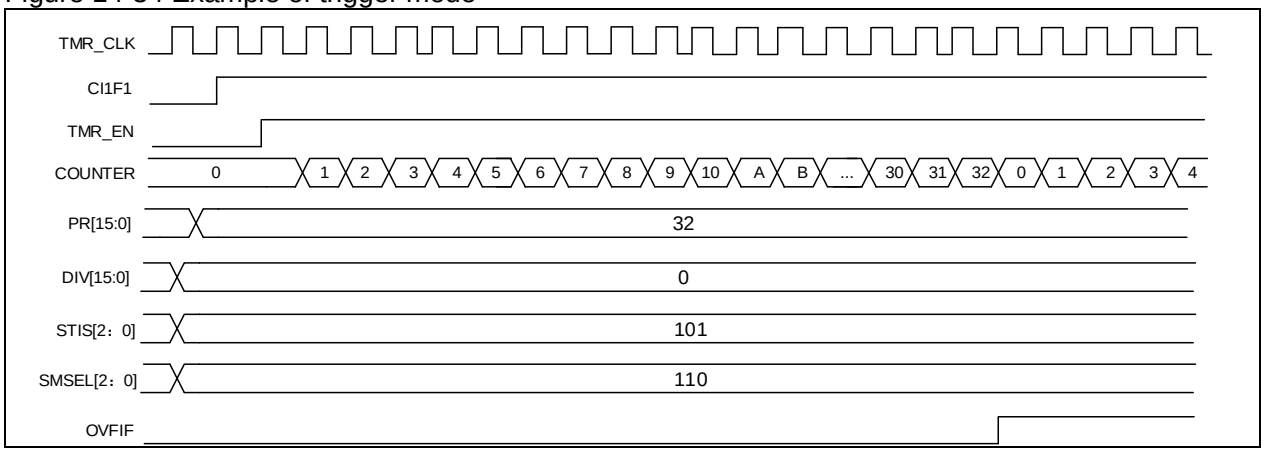
Figure 14-33 Example of suspend mode



Slave mode: Trigger mode

The counter starts counting on the rising edge of a selected trigger input (TMR_EN=1)

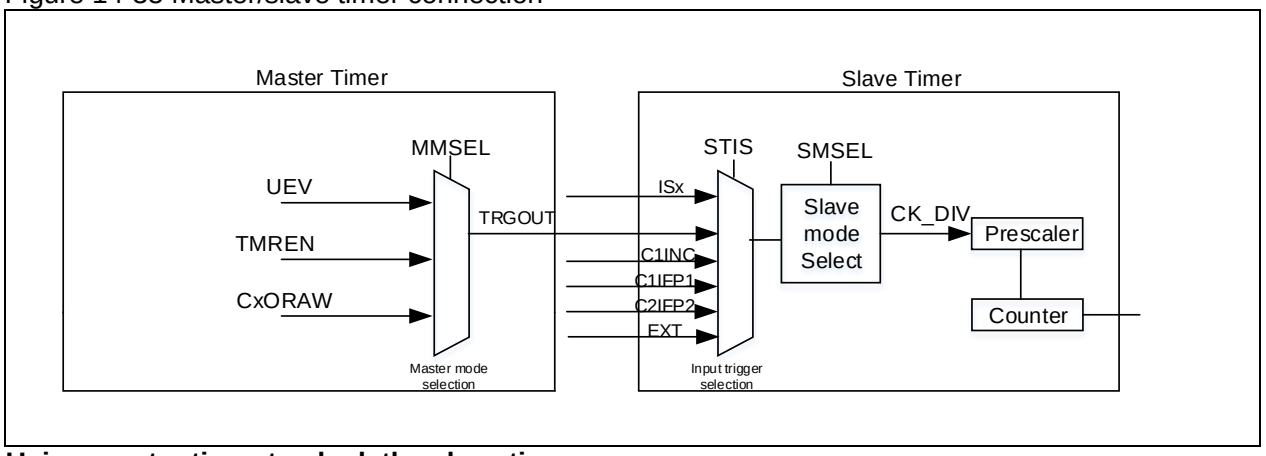
Figure 14-34 Example of trigger mode



Master/slave timer interconnection

Both Master and slave timer can be configured in different master and slave modes respectively. The combination of both can be used for various purposes. *Figure 14-35* provides an example of interconnection between master timer and slave timer.

Figure 14-35 Master/slave timer connection



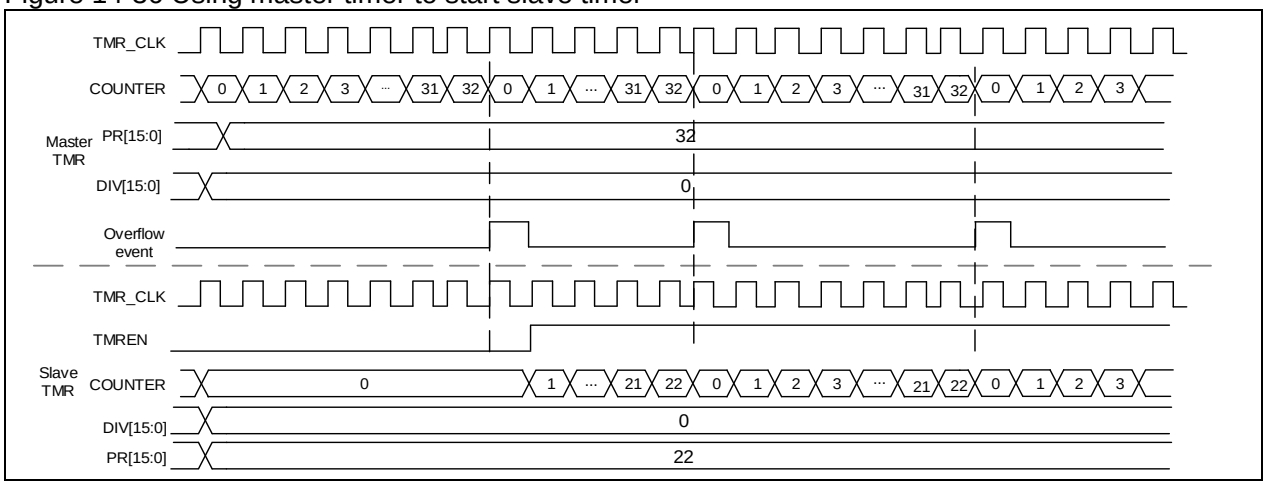
Using master timer to clock the slave timer:

- Configure master timer output signal TRGOUT as an overflow event (PTOS[2: 0]=3’b010). The master timer outputs a pulse signal at each counter overflow event, which is used as the counting clock of a slave timer.
- Configure the master timer counting period (TMRx_PR register)
- Configure the slave timer trigger input signal TRGIN as master timer output (STIS[2: 0] in the TMRx_STCTRL register)
- Configure the slave timer to use external clock mode A (SMSEL[2: 0]=3’b111 in the TMRx_STCTRL register)
- Set TMREN =1 in both master timer and slave timer to enable them

Using master timer to start slave timer:

- Configure master timer output signal TRGOUT as an overflow event (PTOS[2: 0]=3’b010). The master timer outputs a pulse signal at each counter overflow event, which is used as the counting clock of the slave timer.
- Configure master timer counting period (TMRx_PR registers)
- Configure slave timer trigger input signal TRGIN as master timer input
- Configure slave timer as trigger mode (SMSEL=3’b110 in the TMR2_STCTRL register)
- Set TMREN=1 to enable master timer.

Figure 14-36 Using master timer to start slave timer

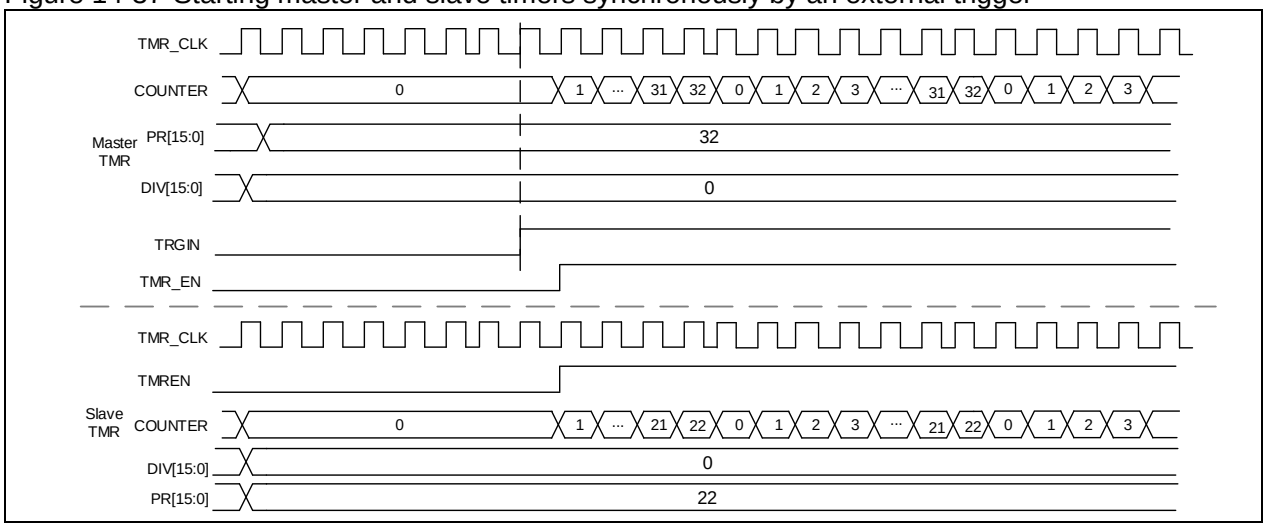


Starting master and slave timers synchronously by an external trigger:

In this example, configure master timer as master mode and slave mode synchronously and enable its slave timer synchronization function. This mode is used for synchronization between master timer and slave timer.

- Set STS=1 of the master timer.
- Configure master timer output signal TRGOUT as an overflow event (PTOS[2: 0]=3'b010). The master timer outputs a pulse signal at each counter overflow event, which is used as the counting clock of the slave timer.
- Configure the slave timer mode of the master timer as trigger mode, and select C1IN as trigger source
- Configure slave timer trigger input signal TRGIN as master timer output
- Configure slave timer as trigger mode (SMSEL=3'b110 in the TMR2_STCTRL register)

Figure 14-37 Starting master and slave timers synchronously by an external trigger



14.2.3.6 Debug mode

When the microcontroller enters debug mode (Cortex®-M0+ core halted), the TMRx counter stops counting by setting the TMRx_PAUSE to 1 in the DEBUG module.

14.2.4 TMR3 registers

These peripheral registers must be accessed by half-words (16 bits) or words (32 bits).

TMR3 registers are mapped into a 32-bit addressable space.

Table 14-5 TMR3 register map and reset value

Register	Offset	Reset value
TMRx_CTRL1	0x00	0x0000
TMRx_CTRL2	0x04	0x0000
TMRx_STCTRL	0x08	0x0000
TMRx_IDEN	0x0C	0x0000
TMRx_ISTS	0x10	0x0000
TMRx_SWEVT	0x14	0x0000
TMRx_CM1	0x18	0x0000
TMRx_CM2	0x1C	0x0000
TMRx_CCTRL	0x20	0x0000
TMRx_CVAL	0x24	0x0000
TMRx_DIV	0x28	0x0000
TMRx_PR	0x2C	0x0000
TMRx_C1DT	0x34	0x0000
TMRx_C2DT	0x38	0x0000
TMRx_C3DT	0x3C	0x0000
TMRx_C4DT	0x40	0x0000
TMRx_DMACTRL	0x48	0x0000
TMRx_DMADT	0x4C	0x0000

14.2.4.1 TMR3 control register 1 (TMRx_CTRL1)

Bit	Name	Reset value	Type	Description
Bit 15: 10	Reserved	0x0	resd	Kept at default value.
Bit 9: 8	CLKDIV	0x0	rw	Clock divider This field is used to define the division ratio between digital filter sampling frequency (f_{DTS}) and timer clock frequency (f_{CK_INT}). 00: No division, $f_{DTS}=f_{CK\_INT}$ 01: Divided by 2, $f_{DTS}=f_{CK\_INT}/2$ 10: Divided by 4, $f_{DTS}=f_{CK\_INT}/4$ 11: Reserved
Bit 7	PRBEN	0x0	rw	Period buffer enable 0: Period buffer is disabled 1: Period buffer is enabled
Bit 6: 5	TWCMSEL	0x0	rw	Two-way counting mode selection 00: One-way counting mode, depending on the OWCDIR bit 01: Two-way counting mode1, count up and down alternately, the CxIF bit is set only when the counter counts down 10: Two-way counting mode2, count up and down alternately, the CxIF bit is set only when the counter counts up 11: Two-way counting mode3, count up and down alternately, the CxIF bit is set when the counter counts up / down
Bit 4	OWCDIR	0x0	rw	One-way count direction 0: Up 1: Down
Bit 3	OCMEN	0x0	rw	One cycle mode enable This bit is use to select whether to stop counting at an

				overflow event 0: The counter does not stop at an overflow event 1: The counter stops at an overflow event
Bit 2	OVFS	0x0	rw	Overflow event source This bit is used to select overflow event or DMA request sources. 0: Counter overflow, setting the OVFSWTR bit or overflow event generated by slave timer controller 1: Only counter overflow generates an overflow event
Bit 1	OVFEN	0x0	rw	Overflow event enable 0: Enabled 1: Disabled
Bit 0	TMREN	0x0	rw	TMR enable 0: Disabled 1: Enabled

14.2.4.2 TMR3 control register 2 (TMRx_CTRL2)

Bit	Name	Reset value	Type	Description
Bit 15: 8	Reserved	0x0	resd	Kept at its default value.
Bit 7	C1INSEL	0x0	rw	C1IN selection 0: CH1 pin is connected to C1IRAW input 1: The XORed result of CH1, CH2 and CH3 pins is connected to C1IRAW input
Bit 6: 4	PTOS	0x0	rw	Master TMR output selection This field is used to select the TMRx signal sent to the slave timer. 000: Reset 001: Enable 010: Update 011: Compare pulse 100: C1ORAW signal 101: C2ORAW signal 110: C3ORAW signal 111: C4ORAW signal
Bit 3	DRS	0x0	rw	DMA request source 0: Capture/compare event 1: Overflow event
Bit 2: 0	Reserved	0x0	resd	Kept at its default value.

14.2.4.3 TMR3 slave timer control register (TMRx_STCTRL)

Bit	Name	Reset value	Type	Description
Bit 15	ESP	0x0	rw	External signal polarity 0: High or rising edge 1: Low or falling edge
Bit 14	ECMBEN	0x0	rw	External clock mode B enable This bit is used to enable external clock mode B 0: Disabled 1: Enabled
Bit 13: 12	ESDIV	0x0	rw	External signal divider This field is used to select the frequency division factor of an external trigger 00: Normal 01: Divided by 2 10: Divided by 4 11: Divided by 8
Bit 11: 8	ESF	0x0	rw	External signal filter This field is used to filter an external signal. The external signal can be sampled only after it has been generated N times 0000: No filter, sampling by 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
Bit 7	STS	0x0	rw	Subordinate TMR synchronization If enabled, master and slave timer can be synchronized. 0: Disabled 1: Enabled
Bit 6: 4	STIS	0x0	rw	Subordinate TMR input selection This field is used to select the input of subordinate TMR. 000: Internal selection 0 (IS0) 001: Internal selection 1 (IS1) 010: Internal selection 2 (IS2) 011: Internal selection 3 (IS3) 100: C1IRAW input detector (C1INC) 101: Filtered input 1 (C1IF1) 110: Filtered input 2 (C1IF2) 111: External input (EXT) Please refer to Table 14-4 for more information on ISx for each timer.
Bit 3	Reserved	0x0	resd	Kept at default value
Bit 2: 0	SMSEL	0x0	rw	Subordinate TMR mode selection 000: Slave mode is disabled 001: Encoder mode A 010: Encoder mode B 011: Encoder mode C 100: Reset mode —Rising edge of the TRGIN input reinitializes the counter 101: Suspend mode — The counter starts counting when the TRGIN is high 110: Trigger mode — A trigger event is generated at the rising edge of the TRGIN input 111: External clock mode A — Rising edge of the TRGIN input clocks the counter Note: Please refer to count mode section for the details on encoder mode A/B/C.

14.2.4.4 TMR3 DMA/interrupt enable register (TMRx_IDEN)

Bit	Name	Reset value	Type	Description
Bit 15	Reserved	0x0	resd	Kept at default value
Bit 14	TDEN	0x0	rw	Trigger DMA request enable 0: Disabled 1: Enabled
Bit 13	Reserved	0x0	resd	Kept at default value
Bit 12	C4DEN	0x0	rw	Channel 4 DMA request enable 0: Disabled 1: Enabled
Bit 11	C3DEN	0x0	rw	Channel 3 DMA request enable 0: Disabled 1: Enabled
Bit 10	C2DEN	0x0	rw	Channel 2 DMA request enable 0: Disabled 1: Enabled

Bit 9	C1DEN	0x0	rw	Channel 1 DMA request enable 0: Disabled 1: Enabled
Bit 8	OVFDEN	0x0	rw	Overflow event DMA request enable 0: Disabled 1: Enabled
Bit 7	Reserved	0x0	resd	Kept at default value
Bit 6	TIEN	0x0	rw	Trigger interrupt enable 0: Disabled 1: Enabled
Bit 5	Reserved	0x0	resd	Kept at default value
Bit 4	C4IEN	0x0	rw	Channel 4 interrupt enable 0: Disabled 1: Enabled
Bit 3	C3IEN	0x0	rw	Channel 3 interrupt enable 0: Disabled 1: Enabled
Bit 2	C2IEN	0x0	rw	Channel 2 interrupt enable 0: Disabled 1: Enabled
Bit 1	C1IEN	0x0	rw	Channel 1 interrupt enable 0: Disabled 1: Enabled
Bit 0	OVFIEN	0x0	rw	Overflow interrupt enable 0: Disabled 1: Enabled

14.2.4.5 TMR3 interrupt status register (TMRx_ISTS)

Bit	Name	Reset value	Type	Description
Bit 15: 13	Reserved	0x0	resd	Kept at default value
Bit 12	C4RF	0x0	rw0c	Channel 4 recapture flag Please refer to C1RF description.
Bit 11	C3RF	0x0	rw0c	Channel 3 recapture flag Please refer to C1RF description.
Bit 10	C2RF	0x0	rw0c	Channel 2 recapture flag Please refer to C1RF description.
Bit 9	C1RF	0x0	rw0c	Channel 1 recapture flag This bit indicates whether a recapture is detected when C1IF=1. This bit is set by hardware, and cleared by writing “0”. 0: No capture is detected 1: Capture is detected.
Bit 8: 7	Reserved	0x0	resd	Kept at default value
Bit 6	TRGIF	0x0	rw0c	Trigger interrupt flag This bit is set by hardware on a trigger event. It is cleared by writing “0”. 0: No trigger event occurred 1: Trigger event is generated. Trigger event: an active edge is detected on TRGIN input, or any edge in suspend mode.
Bit 5	Reserved	0x0	resd	Kept at default value
Bit 4	C4IF	0x0	rw0c	Channel 4 interrupt flag Please refer to C1IF description.
Bit 3	C3IF	0x0	rw0c	Channel 3 interrupt flag Please refer to C1IF description.
Bit 2	C2IF	0x0	rw0c	Channel 2 interrupt flag Please refer to C1IF description.
Bit 1	C1IF	0x0	rw0c	Channel 1 interrupt flag If the channel 1 is configured as input mode: This bit is set by hardware on a capture event. It is cleared by software or read access to the TMRx_C1DT 0: No capture event occurred 1: Capture event is generated

				If the channel 1 is configured as output mode: This bit is set by hardware on a compare event. It is cleared by software. 0: No compare event occurred 1: Compare event is generated
Bit 0	OVFIF	0x0	rw0c	Overflow interrupt flag This bit is set by hardware on an overflow event. It is cleared by software. 0: No overflow event occurred 1: Overflow event is generated. If OVFEN=0 and OVFS=0 in the TMRx_CTRL1 register: − An overflow event is generated when OVFG= 1 in the TMRx_SWEVE register; − An overflow event is generated when the counter CVAL is reinitialized by a trigger event.

14.2.4.6 TMR3 software event register (TMRx_SWEVT)

Bit	Name	Reset value	Type	Description
Bit 15: 7	Reserved	0x0	resd	Kept at default value.
Bit 6	TRGSWTR	0x0	rw	Trigger event triggered by software This bit is set by software to generate a trigger event. 0: No effect 1: Generate a trigger event.
Bit 5	Reserved	0x0	resd	Kept at default value.
Bit 4	C4SWTR	0x0	wo	Channel 4 event triggered by software Please refer to C1M description.
Bit 3	C3SWTR	0x0	wo	Channel 3 event triggered by software Please refer to C1M description.
Bit 2	C2SWTR	0x0	wo	Channel 2 event triggered by software Please refer to C1M description
Bit 1	C1SWTR	0x0	wo	Channel 1 event triggered by software This bit is set by software to generate a channel 1 event. 0: No effect 1: Generate a channel 1 event.
Bit 0	OVFSWTR	0x0	wo	Overflow event triggered by software This bit is set by software to generate an overflow event. 0: No effect 1: Generate an overflow event.

14.2.4.7 TMR3 channel mode register 1 (TMRx_CM1)

Output compare mode:

Bit	Name	Reset value	Type	Description
Bit 15	C2OSEN	0x0	rw	Channel 2 output switch enable
Bit 14: 12	C2OCTRL	0x0	rw	Channel 2 output control
Bit 11	C2OBEN	0x0	rw	Channel 2 output buffer enable
Bit 10	C2OIEN	0x0	rw	Channel 2 output enable immediately
Bit 9: 8	C2C	0x0	rw	Channel 2 configuration This field is used to define the direction of the channel 2 (input or output), and the selection of input pin when C2EN='0': 00: Output 01: Input, C2IN is mapped on C2IFP2 10: Input, C2IN is mapped on C1IFP2 11: Input, C2IN is mapped on STCI. This mode works only when the internal trigger input is selected by STIS register.
Bit 7	C1OSEN	0x0	rw	Channel 1 output switch enable 0: C1ORAW is not affected by EXT 1: Once high level is detect on EXT input, C1ORAW is cleared.
Bit 6: 4	C1OCTRL	0x0	rw	Channel 1 output control This field defines the behavior of the original signal C1ORAW.

				000: Disconnected. C1ORAW is disconnected from C1OUT; 001: C1ORAW is high when TMRx_CVAL=TMRx_C1DT 010: C1ORAW is low when TMRx_CVAL=TMRx_C1DT 010: Switch C1ORAW level when TMRx_CVAL=TMRx_C1DT 100: C1ORAW is forced low 101: C1ORAW is forced high. 110: PWM mode A －OWCDIR=0, C1ORAW is high once TMRx_C1DT>TMRx_CVAL, else low; －OWCDIR=1, C1ORAW is low once TMRx_ C1DT <TMRx_CVAL, else high; 111: PWM mode B －OWCDIR=0, C1ORAW is low once TMRx_ C1DT >TMRx_CVAL, else high; －OWCDIR=1, C1ORAW is high once TMRx_ C1DT <TMRx_CVAL, else low. *Note: In the configurations other than 000', the C1OUT is connected to C1ORAW. The C1OUT output level is not only subject to the changes of C1ORAW, but also the output polarity set by CCTRL.*
Bit 3	C1OBEN	0x0	rw	Channel 1 output buffer enable 0: Buffer function of TMRx_C1DT is disabled. The new value written to the TMRx_C1DT takes effect immediately. 1: Buffer function of TMRx_C1DT is enabled. The value to be written to the TMRx_C1DT is stored in the buffer register, and can be sent to the TMRx_C1DT register only on an overflow event.
Bit 2	C1OIEN	0x0	rw	Channel 1 output enable immediately In PWM mode A or B, this bit is used to accelerate the channel 1 output's response to the trigger event. 0: Need to compare the CVAL with C1DT before generating an output 1: No need to compare the CVAL and C1DT. An output is generated immediately when a trigger event occurs.
Bit 1: 0	C1C	0x0	rw	Channel 1 configuration This field is used to define the direction of the channel 1 (input or output), and the selection of input pin when C1EN='0': 00: Output 01: Input, C1IN is mapped on C1IFP1 10: Input, C1IN is mapped on C2IFP1 11: Input, C1IN is mapped on STCI. This mode works only when the internal trigger input is selected by STIS.

Input capture mode:

Bit	Name	Reset value	Type	Description
Bit 15: 12	C2DF	0x0	rw	Channel 2 digital filter
Bit 11: 10	C2IDIV	0x0	rw	Channel 2 input divider
Bit 9: 8	C2C	0x0	rw	Channel 2 configuration This field is used to define the direction of the channel 2 (input or output), and the selection of input pin when C2EN='0': 00: Output 01: Input, C2IN is mapped on C2IFP2 10: Input, C2IN is mapped on C1IFP2 11: Input, C2IN is mapped on STCI. This mode works only when the internal trigger input is selected by STIS.
Bit 7: 4	C1DF	0x0	rw	Channel 1 digital filter This field defines the digital filter of the channel 1. "N" stands for the number of filtering, indicating that the input edge can pass the filter only after N sampling events.

				0000: No filter, sampling is done at f_{DTS} 1000: $f_{SAMPLING}=f_{DTS}/8$, N=6 0001: $f_{SAMPLING}=f_{CK\_INT}$, N=2 1001: $f_{SAMPLING}=f_{DTS}/8$, N=8 0010: $f_{SAMPLING}=f_{CK\_INT}$, N=4 1010: $f_{SAMPLING}=f_{DTS}/16$, N=5 0011: $f_{SAMPLING}=f_{CK\_INT}$, N=8 1011: $f_{SAMPLING}=f_{DTS}/16$, N=6 0100: $f_{SAMPLING}=f_{DTS}/2$, N=6 1100: $f_{SAMPLING}=f_{DTS}/16$, N=8 0101: $f_{SAMPLING}=f_{DTS}/2$, N=8 1101: $f_{SAMPLING}=f_{DTS}/32$, N=5 0110: $f_{SMPLING}=f_{DTS}/4$, N=6 1110: $f_{SAMPLING}=f_{DTS}/32$, N=6 0111: $f_{SAMPLING}=f_{DTS}/4$, N=8 1111: $f_{SAMPLING}=f_{DTS}/32$, N=8
Bit 3: 2	C1IDIV	0x0	rw	Channel 1 input divider This field defines Channel 1 input divider. 00: No divider. An input capture is generated at each active edge. 01: An input compare is generated every 2 active edges 10: An input compare is generated every 4 active edges 11: An input compare is generated every 8 active edges Note: the divider is reset once C1EN='0'
Bit 1: 0	C1C	0x0	rw	Channel 1 configuration This field is used to define the direction of the channel 1 (input or output), and the selection of input pin when C1EN='0': 00: Output 01: Input, C1IN is mapped on C1IFP1 10: Input, C1IN is mapped on C2IFP1 11: Input, C1IN is mapped on STCI. This mode works only when the internal trigger input is selected by STIS.

14.2.4.8 TMR3 channel mode register 2 (TMRx_CM2)

Output compare mode:

Bit	Name	Reset value	Type	Description
Bit 15	C4OSEN	0x0	rw	Channel 4 output switch enable
Bit 14: 12	C4OCTRL	0x0	rw	Channel 4 output control
Bit 11	C4OBEN	0x0	rw	Channel 4 output buffer enable
Bit 10	C4OIEN	0x0	rw	Channel 4 output enable immediately
Bit 9: 8	C4C	0x0	rw	Channel 4 configuration This field is used to define the direction of the channel 1 (input or output), and the selection of input pin when C4EN='0': 00: Output 01: Input, C4IN is mapped on C4IFP4 10: Input, C4IN is mapped on C3IFP4 11: Input, C4IN is mapped on STCI. This mode works only when the internal trigger input is selected by STIS.
Bit 7	C3OSEN	0x0	rw	Channel 3 output switch enable
Bit 6: 4	C3OCTRL	0x0	rw	Channel 3 output control
Bit 3	C3OBEN	0x0	rw	Channel 3 output buffer enable
Bit 2	C3OIEN	0x0	rw	Channel 3 output enable immediately
Bit 1: 0	C3C	0x0	rw	Channel 3 configuration This field is used to define the direction of the channel 1 (input or output), and the selection of input pin when C3EN='0': 00: Output 01: Input, C3IN is mapped on C3IFP3 10: Input, C3IN is mapped on C4IFP3 11: Input, C3IN is mapped on STCI. This mode works only when the internal trigger input is selected by STIS.

Input capture mode:

Bit	Name	Reset value	Type	Description
Bit 15: 12	C4DF	0x0	rw	Channel 4 digital filter
Bit 11: 10	C4IDIV	0x0	rw	Channel 4 input divider
Bit 9: 8	C4C	0x0	rw	Channel 4 configuration This field is used to define the direction of the channel 1 (input or output), and the selection of input pin when C4EN='0': 00: Output 01: Input, C4IN is mapped on C4IFP4 10: Input, C4IN is mapped on C3IFP4 11: Input, C4IN is mapped on STCI. This mode works only when the internal trigger input is selected by STIS.
Bit 7: 4	C3DF	0x0	rw	Channel 3 digital filter
Bit 3: 2	C3IDIV	0x0	rw	Channel 3 input divider
Bit 1:0	C3C	0x0	rw	Channel 3 configuration This field is used to define the direction of the channel 1 (input or output), and the selection of input pin when C3EN='0': 00: Output 01: Input, C3IN is mapped on C3IFP3 10: Input, C3IN is mapped on C4IFP3 11: Input, C3IN is mapped on STCI. This mode works only when the internal trigger input is selected by STIS.

14.2.4.9 TMR3 channel control register (TMRx_CCTRL)

Bit	Name	Reset value	Type	Description
Bit 15: 14	Reserved	0x0	resd	Kept at default value.
Bit 13	C4P	0x0	rw	Channel 4 polarity Please refer to C1P description.
Bit 12	C4EN	0x0	rw	Channel 4 enable Please refer to C1EN description.
Bit 11	C3CP	0x0	rw	Channel 3 complementary polarity This bit defines the valid edge of input signals. Refer to C1P bit for details.
Bit 10	Reserved	0x0	resd	Kept at default value.
Bit 9	C3P	0x0	rw	Channel 3 polarity Please refer to C1P description.
Bit 8	C3EN	0x0	rw	Channel 3 enable Please refer to C1EN description.
Bit 7	C2CP	0x0	rw	Channel 2 complementary polarity This bit defines the valid edge of input signals. Refer to C1P bit for details.
Bit 6	Reserved	0x0	resd	Kept at default value.
Bit 5	C2P	0x0	rw	Channel 2 polarity Please refer to C1P description.
Bit 4	C2EN	0x0	rw	Channel 2 enable Please refer to C1EN description.
Bit 3	C1CP	0x0	rw	Channel 1 complementary polarity This bit defines the valid edge of input signals. Refer to C1P bit for details.
Bit 2	Reserved	0x0	resd	Kept at default value.
Bit 1	C1P	0x0	rw	Channel 1 polarity When the channel 1 is configured as output mode: 0: C1OUT is active high 1: C1OUT is active low When the channel 1 is configured as input mode: C1CP/C1P are used to define the valid edge of input signals. 00: C1IN active edge is on its rising edge. When used as external trigger, C1IN is not inverted. 01: C1IN active edge is on its falling edge. When used as external trigger, C1IN is inverted.

				10: Reserved 11: C1IN rising and falling edges active. When used as external trigger, C1IN is not inverted.
Bit 0	C1EN	0x0	rw	Channel 1 enable 0: Input or output is disabled 1: Input or output is enabled

Table 14-6 Standard CxOUT channel output control bit

CxEN bit	CxOUT output state
0	Output disabled (CxOUT=0, Cx_EN=0)
1	CxOUT = CxORAW + polarity, Cx_EN=1

Note: The state of the external I/O pins connected to the standard CxOUT channel depends on the CxOUT channel state and the GPIO and IOMUX registers.

14.2.4.10 TMR3 counter value (TMRx_CVAL)

Bit	Name	Reset value	Type	Description
Bit 31: 16	Reserved	0x0	resd	Kept at its default value.
Bit 15: 0	CVAL	0x0	rw	Counter value

14.2.4.11 TMR3 division (TMRx_DIV)

Bit	Name	Reset value	Type	Description
Bit 15: 0	DIV	0x0	rw	Divider value The counter clock frequency $f_{CK\_CNT} = f_{TMR\_CLK}$ /(DIV[15: 0]+1). DIV contains the value written at an overflow event.

14.2.4.12 TMR3 period register (TMRx_PR)

Bit	Name	Reset value	Type	Description
Bit 31: 16	Reserved	0x0	resd	Kept at its default value.
Bit 15: 0	PR	0x0	rw	Period value This defines the period value of the TMRx counter. The timer stops working when the period value is 0.

14.2.4.13 TMR3 channel 1 data register (TMRx_C1DT)

Bit	Name	Reset value	Type	Description
Bit 31: 16	Reserved	0x0	resd	Kept at its default value.
Bit 15: 0	C1DT	0x0000	rw	Channel 1 data register When the channel 1 is configured as input mode: The C1DT is the CVAL value stored by the last channel 1 input event (C1IN) When the channel 1 is configured as output mode: C1DT is the value to be compared with the CVAL value. Whether the written value takes effective immediately depends on the C1OBEN bit, and the corresponding output is generated on C1OUT as configured.

14.2.4.14 TMR3 channel 2 data register (TMRx_C2DT)

Bit	Name	Reset value	Type	Description
Bit 31: 16	Reserved	0x0	resd	Kept at its default value.
Bit 15: 0	C2DT	0x0	rw	Channel 2 data register When the channel 2 is configured as input mode: The C2DT is the CVAL value stored by the last channel 2 input event (C1IN) When the channel 2 is configured as output mode: C2DT is the value to be compared with the CVAL value. Whether the written value takes effective immediately depends on the C2OBEN bit, and the corresponding output is generated on C2OUT as configured.

14.2.4.15 TMR3 channel 3 data register (TMRx_C3DT)

Bit	Name	Reset value	Type	Description
Bit 31: 16	Reserved	0x0	resd	Kept at its default value.
Bit 15: 0	C3DT	0x0	rw	Channel 3 data register When the channel 3 is configured as input mode: The C3DT is the CVAL value stored by the last channel 3 input event (C1IN) When the channel 3 is configured as output mode: C3DT is the value to be compared with the CVAL value. Whether the written value takes effective immediately depends on the C3OBEN bit, and the corresponding output is generated on C3OUT as configured.

14.2.4.16 TMR3 channel 4 data register (TMRx_C4DT)

Bit	Name	Reset value	Type	Description
Bit 31: 16	Reserved	0x0	resd	Kept at its default value.
Bit 15: 0	C4DT	0x0	rw	Channel 4 data register When the channel 4 is configured as input mode: The C4DT is the CVAL value stored by the last channel 4 input event (C1IN) When the channel 4 is configured as output mode: C4DT is the value to be compared with the CVAL value. Whether the written value takes effective immediately depends on the C4OBEN bit, and the corresponding output is generated on C4OUT as configured.

14.2.4.17 TMR3 DMA control register (TMRx_DMACTRL)

Bit	Name	Reset value	Type	Description
Bit 15: 13	Reserved	0x0	resd	Kept at default value.
Bit 12: 8	DTB	0x00	rw	DMA transfer bytes This field defines the number of DMA transfers: 00000: 1 byte 00001: 2 bytes 00010: 3 bytes 00011: 4 bytes .... 10000: 17 bytes 10001: 18 bytes
Bit 7: 5	Reserved	0x0	resd	Kept at default value.
Bit 4: 0	ADDR	0x00	rw	DMA transfer address offset ADDR is defined as an offset starting from the address of the TMRx_CTRL1 register. 00000: TMRx_CTRL1 00001: TMRx_CTRL2 00010: TMRx_STCTRL .....

14.2.4.18 TMR3 DMA data register (TMRx_DMADT)

Bit	Name	Reset value	Type	Description
Bit 15: 0	DMADT	0x0	rw	DMA data register A read or write operation to the DMADT register accesses the TMR registers at the following address: TMRx peripheral address + ADDR*4 to TMRx peripheral address + ADDR*4 + DTB*4.

14.3 General-purpose timer (TMR14)

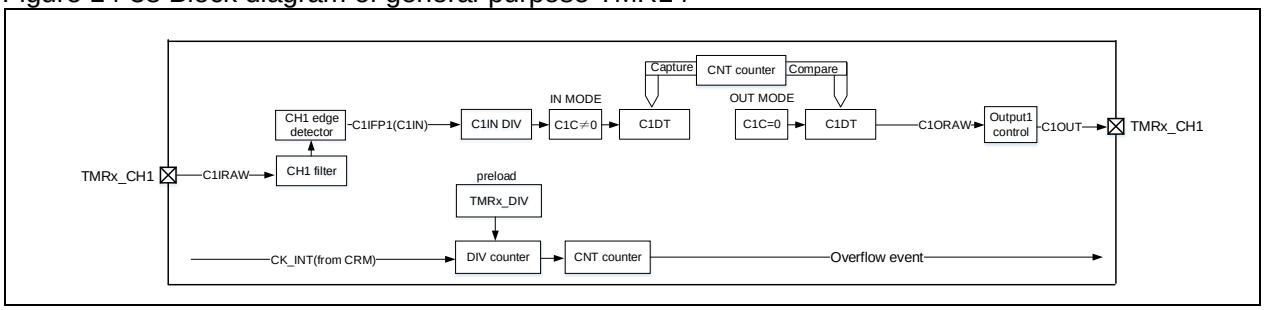
14.3.1 TMR14 introduction

The general-purpose timer (TMR14) consists of a 16-bit counter supporting upcounting mode.

14.3.2 TMR14 main features

- Clock source of counter: internal clock
- 16-bit up counter
- 1x independent channel for input capture, output compare, PWM generation
- Synchronization circuit to interconnect several timers together
- Interrupt generation on at overflow and channel events

Figure 14-38 Block diagram of general-purpose TMR14

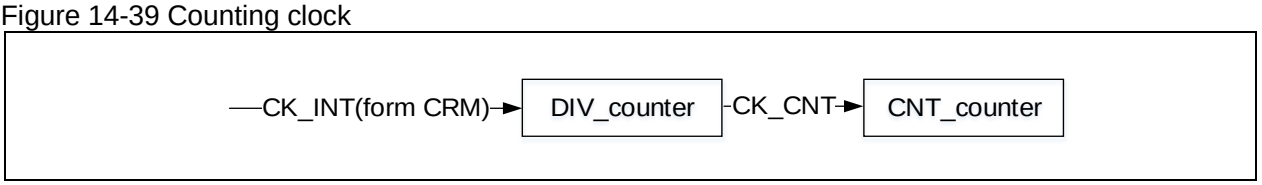


14.3.3 TMR14 functional overview

14.3.3.1 Counting clock

The counter clock of TMR14 is provided by an internal clock (CK_INT).

Figure 14-39 Counting clock



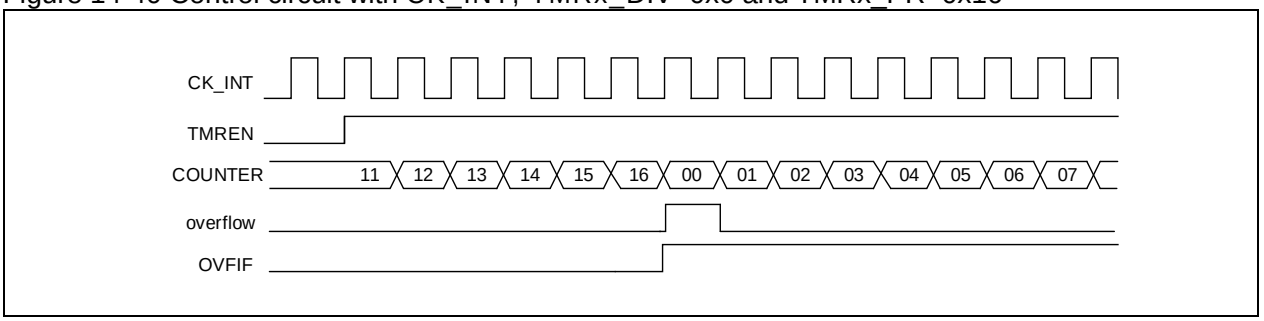
Internal clock (CK_INT)

By default, the CK_INT, which is divided by a prescaler, is used to drive the counter to count. When TMR's APB clock prescaler factor is 1, the CK_INT frequency is equal to that of APB; otherwise, it doubles the APB clock frequency.

Follow the procedures below:

- Set counting frequency through TMRx_DIV register
- Set counting cycles through TMRx_PR register
- Enable the counter by setting the TMREN bit in the TMRx_CTRL1 register

Figure 14-40 Control circuit with CK_INT, TMRx_DIV=0x0 and TMRx_PR=0x16



14.3.3.2 Counting mode

The general-purpose timer (TMR14) consists of a 16-bit counter supporting upcounting mode.

The TMRx_PR register is used to define counting period of counter. The value in the TMRx_PR is immediately moved to the shadow register by default. When the periodic buffer is enabled (PRBEN=1), the value in the TMRx_PR register is transferred to the shadow register only at an overflow event.

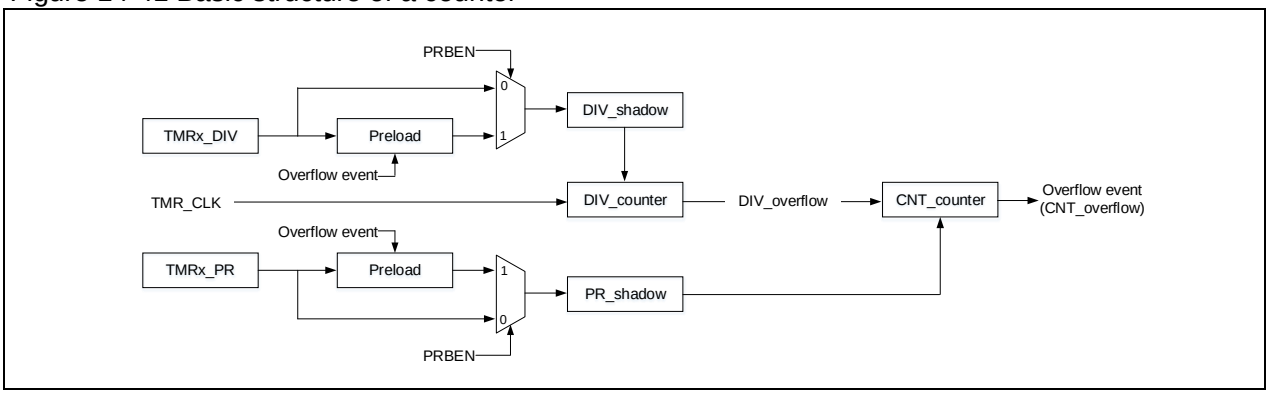
TMRx_DIV register is used to define the counter frequency of the counter. The counter counts once every DIV[15:0]+1 clock cycle. Similar to TMRx_PR register, after a periodic buffer is enabled, the value of the TMRx_DIV register is transferred into the shadow register upon an overflow event.

Reading the TMRx_CNT register returns the current counter value. Writing the TMRx_CNT register will update the current counter value.

An overflow event is enabled by default. It can be disabled by setting OVFEN=1 in the TMRx_CTRL1 register. The OVFS bit in the TMRx_CTRL1 register is used to select an overflow event source, which is, by default, counter overflow or underflow, setting OVFSWTR, or reset signal generated by slave timer controller reset mode. Once the OVFS is set, an overflow event is generated only when overflow or underflow occurs.

Setting the TMREN bit (TMREN=1) enables the timer to start counting. Base on synchronization logic, however, the actual enable signal TMR_EN is set 1 clock cycle after the TMREN is set.

Figure 14-41 Basic structure of a counter



Upcounting mode

This mode is enabled by setting CMSEL[1:0]=2'b00 and OWCDIR=1'b0 in the TMRx_CTRL1 register.

In upcounting mode, the counter counts from 0 to the value programmed in the TMRx_PR register, restarts from 0, and generates a counter overflow event, setting OVFIF bit to 1. If the overflow event is disabled, the counter is no longer reloaded with the prescaler and period value on counter overflow, otherwise, the prescaler and period value will be updated on an overflow event.

Figure 14-42 Overflow event when PRBEN=0

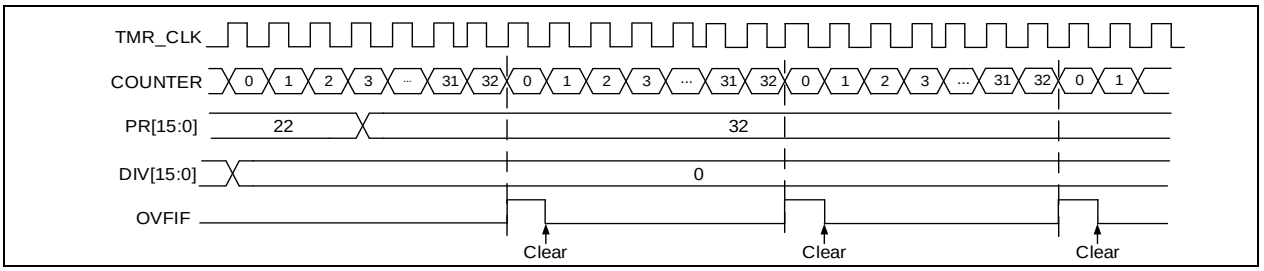
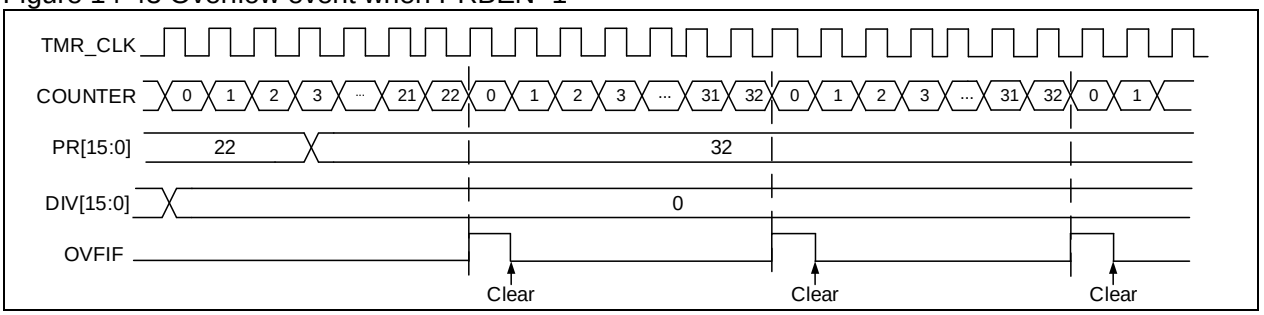

TMR_CLK
COUNTER
0
1
2
3
...
31
32
PR[15:0]
22
32
DIV[15:0]
0
OVFIF
Clear
Clear
Clear

Figure 14-43 Overflow event when PRBEN=1



14.3.3.3 TMR input function

TMR14 has one independent channel that can be configured as input or output.

As input, each channel input signal is processed as follows:

- TMRx_CHx outputs the pre-processed CxIRAW. The C1INSE bit is used to select TMRx_CHx as the source of C1IRAW
- CxIRAW passes digital filter and outputs a filtered CxIF signal. The digital filter uses the CxDF bit to set sampling frequency and sampling times.
- CxIF passes edge detector, and outputs the CxIFPx signal with edge selection. The edge selector is controlled by CxP and CxCP bits. It provides input rising edge, falling edge and both edges for selection.
- CxIFPx passes capture signal selector, and outputs the CxIN signal with capture signal selection. The capture signal selector is controlled by CxC bit. The CxIN source can be from CxIFPx.
- CxIN passes input divider and outputs CxIPS signal. The divider factor can be defined as No division, divided/2, divided /4 or divided /8, through the CxIDIV bit.

Figure 14-44 Input/output channel 1 main circuit

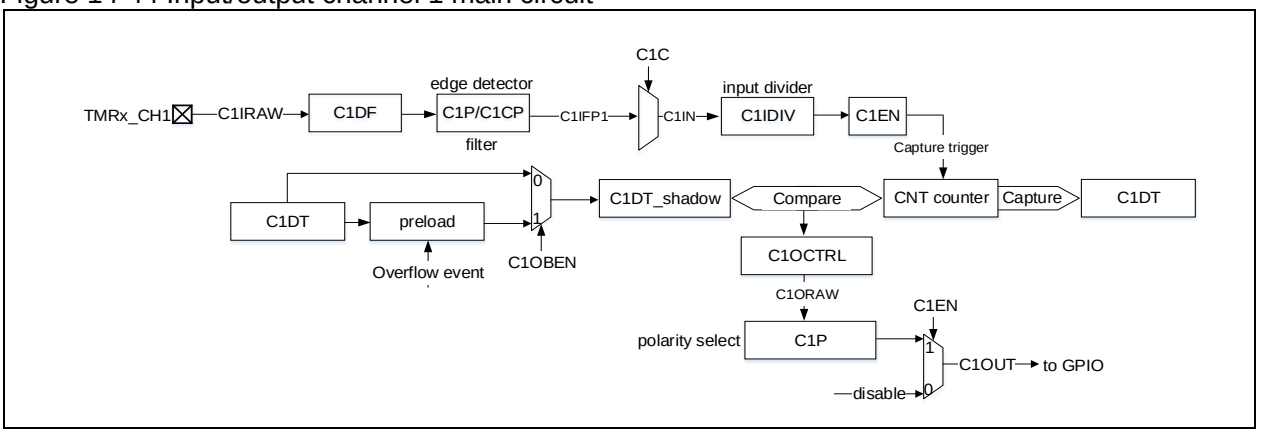
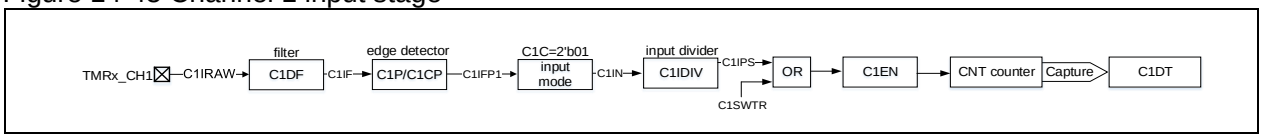


Figure 14-45 Channel 1 input stage



Input mode

In input mode, the TMRx_CxDT register latches the current counter values after the selected trigger signal is detected, and the capture compare interrupt flag bit (CxIF) is set to 1. An interrupt or DMA request will be generated if the CxIEN bit or CxDEN bit is enabled. If the selected trigger signal is detected when CxIF=1, a capture overflow event is generated. The previous counter value will be overwritten by the current counter value, and the CxRF is set to 1.

To capture the rising edge of C1IN input, follow the procedure below:

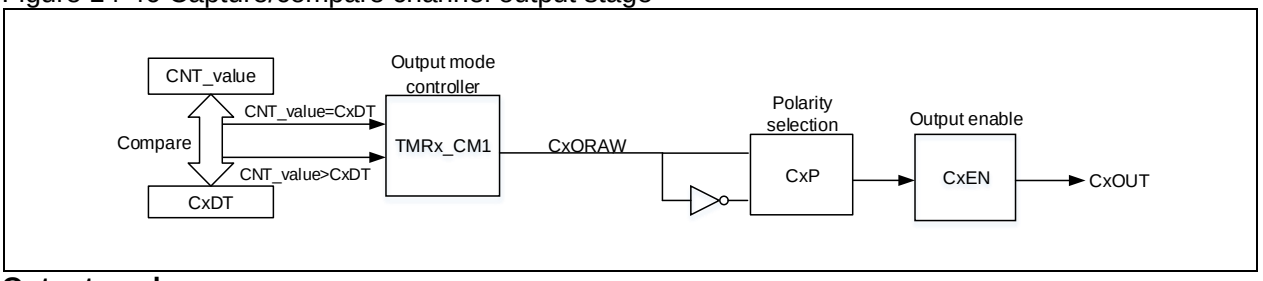
- Set C1C=01 in the TMRx_CM1 register to select C1IN (channel 1 input)
- Set C1IN signal filter bandwidth (CxDF[3: 0])
- Set the active edge on the C1IN channel by writing C1P=0 (rising edge) in the TMRx_CCTR register

- Program C1IN signal capture frequency divider (C1DIV[1:0])
- Enable channel 1 input capture (C1EN=1)
- If needed, enable the relevant interrupt by setting the C1IEN bit in the TMRx_IDEN register

14.3.3.4 TMR output function

The TMR output consists of a comparator and an output controller. It is used to program the period, duty cycle and polarity of output signals.

Figure 14-46 Capture/compare channel output stage



Output mode

CxC[1: 0]≠2'b00 is used to configure the channel as output mode. In this case, the counter value is compared with that in the CxDT register, and the intermediate signal CxORAW is generated according to the output mode selected by CxOCTRL[2: 0], and this signal is sent to IO after being processed by the output control circuit. The period of the output signal is configured by the TMRx_PR register, while the duty cycle is configured by the CxDT register.

Output compare modes include:

PWM mode A:

Enable PWM mode A by setting CxOCTRL=3'b110. In upcounting mode, C1ORAW outputs high when TMRx_C1DT>TMRx_CVAL, otherwise, it is low; in downcounting mode, C1ORAW outputs low when TMRx_C1DT<TMRx_CVAL, otherwise, it is high.

To use PWM mode A, the following procedures are recommended:

- Set PWM period through TMRx_PR register
- Set PWM duty cycles through TMRx_CxD
- Select PWM mode A by setting CxOCTRL=3'b110 in the TMRx_CM1/CM2 register
- Set counting frequency through TMRx_DIV register
- Select counting mode by setting the TWCMSEL[1:0] bit in the TMRx_CTRL1 register
- Select output polarity through the CxP and CxCP bits in the TMRx_CCTRL register
- Enable channel output through the CxEN and CxCEN bits in the TMRx_CCTRL register
- Enable TMRx output through the OEN bit in the TMRx_BRK register
- Configure GPIOs corresponding to TMR output channels as multiplexed mode
- Enable TMRx to start counting through the TMREN bit in the TMRx_CTRL1 register.

PWM mode B:

Enable PWM mode B by setting CxOCTRL=3'b111. In upcounting mode, C1ORAW outputs low when TMRx_C1DT>TMRx_CVAL, otherwise, it is high; In downcounting mode, C1ORAW outputs high when TMRx_C1DT<TMRx_CVAL, otherwise, it is low.

Forced output mode:

Enable forced output mode by setting CxOCTRL=3'b100/101. In this case, the CxORAW is forced to be the programmed level, regardless of the counter value. Despite this, the channel flag bit and DMA request still depend on the compare result.

Output compare mode:

Enable output compare mode by setting CxOCTRL=3'b001/010/011. In this case, when the counter value matches the value of the CxDT register, the CxORAW is forced high (CxOCTRL=3'b001), low (CxOCTRL=3'b010) or toggling (CxOCTRL=3'b011).

One-pulse mode:

This is a particular case of PWM mode. Enable one-pulse by setting OCMEN=1. In this mode, the comparison match is performed in the current counting period. The TMREN bit is cleared as soon as

the current counting is completed. Therefore, only one pulse is output. When in upcounting mode, the configuration must follow the rule: CVAL<CxDT≤PR; in downcounting mode, CVAL>CxDT is required.

Fast output mode:

Enable this mode by setting CxOIEN=1. If enabled, the CxORAW signal will not change when the counter value matches the CxDT, but change at the beginning of the current counting period. In other words, the comparison result is advanced, so the comparison result between the counter value and the TMRx_CxDT register will determine the level of CxORAW in advance.

Figure 14-47 gives an example of output compare mode (toggle) with C1DT=0x3. When the counter value is equal to 0x3, C1OUT toggles.

Figure 14-48 gives an example of the combination between upcounting mode and PWM mode A. The output signal behaves when PR=0x32 but CxDT is configured with a different value.

Figure 14-49 gives an example of the combination between upcounting mode and one-pulse PWM mode B. The counter only counts only one cycle, and the output signal sends only one pulse.

Figure 14-47 C1ORAW toggles when counter value matches the C1DT value

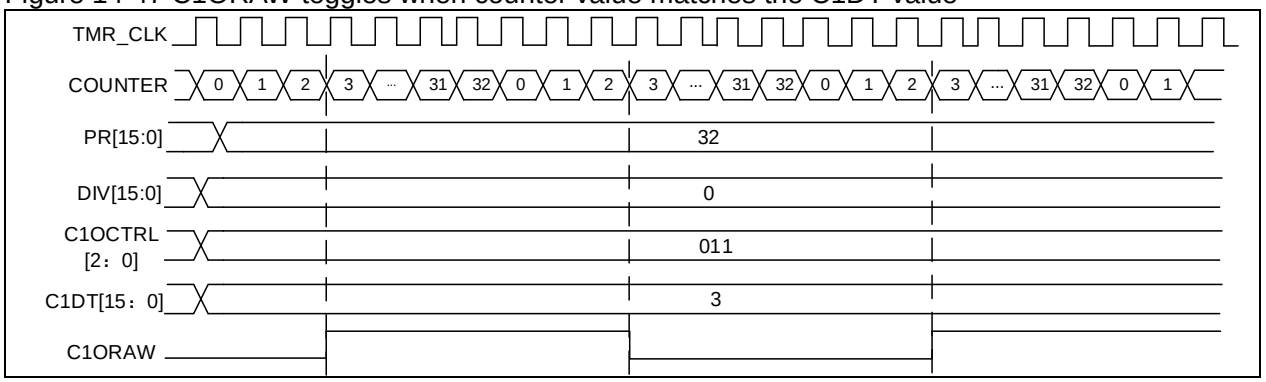


Figure 14-48 Upcounting mode and PWM mode A

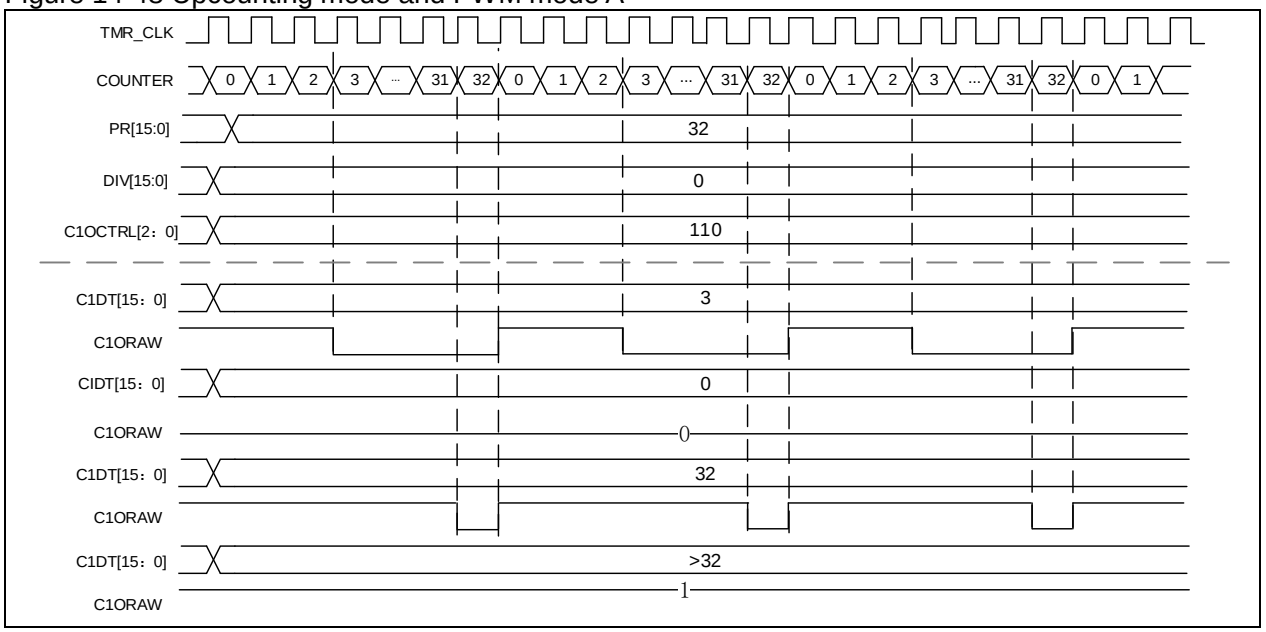
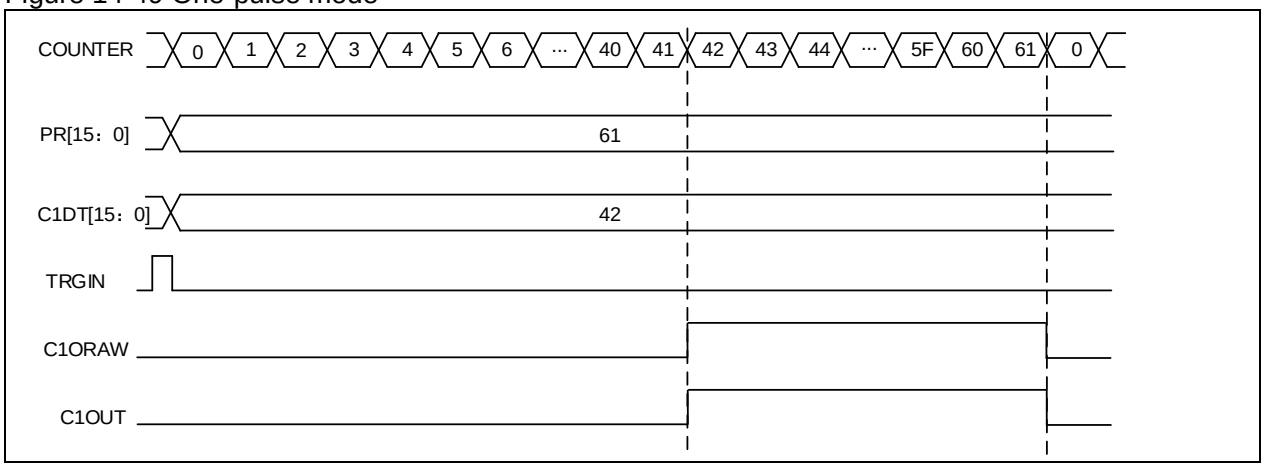


Figure 14-49 One-pulse mode



14.3.3.5 Debug mode

When the microcontroller enters debug mode (Cortex®-M0+ core halted), the TMRx counter stops counting by setting the TMRx_PAUSE in the DEBUG module.

14.3.4 TMR14 registers

These peripheral registers must be accessed by half-words (16 bits) or words (32 bits).

TMR14 registers are mapped into a 16-bit addressable space.

Table 14-7 TMR14 register map and reset value

Register	Register	Reset value
TMRx_CTRL1	0x00	0x0000
TMRx_IDEN	0x0C	0x0000
TMRx_ISTS	0x10	0x0000
TMRx_SWEVT	0x14	0x0000
TMRx_CM1	0x18	0x0000
TMRx_CCTRL	0x20	0x0000
TMRx_CVAL	0x24	0x0000
TMRx_DIV	0x28	0x0000
TMRx_PR	0x2C	0x0000
TMRx_C1DT	0x34	0x0000
TMR14_RMP	0x50	0x0000

14.3.4.1 TMR14 control register 1 (TMRx_CTRL1)

Bit	Name	Reset value	Type	Description
Bit 15: 10	Reserved	0x0	resd	Kept at default value
Bit 9: 8	CLKDIV	0x0	rw	Clock divider This field is used to define the division ratio between digital filter sampling frequency (f_{DTS}) and timer clock frequency (f_{CK_INT}). 00: No division, $f_{DTS}=f_{CK\_INT}$ 01: Divided by 2, $f_{DTS}=f_{CK\_INT}/2$ 10: Divided by 4, $f_{DTS}=f_{CK\_INT}/4$ 11: Reserved
Bit 7	PRBEN	0x0	rw	Period buffer enable 0: Period buffer is disabled 1: Period buffer is enabled
Bit 6: 3	Reserved	0x0	resd	Kept at default value
Bit 2	OVFS	0x0	rw	Overflow event source

				This bit is used to select overflow event or DMA request sources. 0: Counter overflow, setting the OVFSWTR bit or overflow event generated by slave timer controller 1: Only counter overflow generates an overflow event
Bit 1	OVFEN	0x0	rw	Overflow event enable 0: Enabled 1: Disabled
Bit 0	TMREN	0x0	rw	TMR enable 0: Enabled 1: Disabled

14.3.4.2 TMR14 DMA/interrupt enable register (TMRx_IDEN)

Bit	Name	Reset value	Type	Description
Bit 15:2	Reserved	0x0	resd	Kept at default value.
Bit 1	C1IEN	0x0	rw	Channel 1 interrupt enable 0: Disabled 1: Enabled
Bit 0	OVFIEN	0x0	rw	Overflow interrupt enable 0: Disabled 1: Enabled

14.3.4.3 TMR14 interrupt status register (TMRx_ISTS)

Bit	Register	Reset value	Type	Description
Bit 15: 10	Reserved	0x0	resd	Kept at default value.
Bit 9	C1RF	0x0	rw0c	Channel 1 recapture flag This bit indicates whether a recapture is detected when C1IF=1. This bit is set by hardware, and cleared by writing “0”. 0: No capture is detected 1: Capture is detected.
Bit 8:2	Reserved	0x0	resd	Default value
Bit 1	C1IF	0x0	rw0c	Channel 1 interrupt flag If the channel 1 is configured as input mode: This bit is set by hardware on a capture event. It is cleared by software or read access to the TMRx_C1DT 0: No capture event occurred 1: Capture event is generated If the channel 1 is configured as output mode: This bit is set by hardware on a compare event. It is cleared by software. 0: No compare event occurred 1: Compare event is generated
Bit 0	OVFIF	0x0	rw0c	Overflow interrupt flag This bit is set by hardware on an overflow event. It is cleared by software. 0: No overflow event occurs 1: Overflow event is generated. If OVFEN=0 and OVFS=0 in the TMRx_CTRL1 register: − An overflow event is generated when OVFG= 1 in the TMRx_SWEVE register; − An overflow event is generated when the counter CVAL is reinitialized by a trigger event.

14.3.4.4 TMR14 software event register (TMRx_SWEVT)

Bit	Name	Reset value	Type	Description
Bit 15: 2	Reserved	0x0	resd	Kept at default value.
Bit 1	C1SWTR	0x0	wo	Channel 1 event triggered by software This bit is set by software to generate a channel 1 event. 0: No effect 1: Generate a channel 1 event.

Bit 0	OVFSWTR	0x0	wo	Overflow event triggered by software This bit is set by software to generate an overflow event. 0: No effect 1: Generate an overflow event.

14.3.4.5 TMR14 channel mode register 1 (TMRx_CM1)

The channel can be used in input (capture mode) or output (compare mode). The direction of a channel is defined by the corresponding CxC bits. All the other bits of this register have different functions in input and output modes. The CxOx describes its function in output mode when the channel is in output mode, while the CxIx describes its function in output mode when the channel is in input mode. Attention must be given to the fact that the same bit can have different functions in input mode and output mode.

Output compare mode:

Bit	Name	Reset value	Type	Description
Bit 15:7	Reserved	0x0	resd	Kept at default value.
Bit 6: 4	C1OCTRL	0x0	rw	Channel 1 output control This field defines the behavior of the original signal C1ORAW. 000: Disconnected. C1ORAW is disconnected from C1OUT; 001: C1ORAW is high when TMRx_CVAL=TMRx_C1DT 010: C1ORAW is low when TMRx_CVAL=TMRx_C1DT 011: Switch C1ORAW level when TMRx_CVAL=TMRx_C1DT 100: C1ORAW is forced low 101: C1ORAW is forced high. 110: PWM mode A －OWCDIR=0, C1ORAW is high once TMRx_C1DT>TMRx_CVAL, else low; －OWCDIR=1, C1ORAW is low once TMRx_ C1DT <TMRx_CVAL, else high; 111: PWM mode B － OWCDIR=0, C1ORAW is low once TMRx_ C1DT >TMRx_CVAL, else high; － OWCDIR=1, C1ORAW is high once TMRx_ C1DT <TMRx_CVAL, else low. *Note: In the configurations other than 000', the C1OUT is connected to C1ORAW. The C1OUT output level is not only subject to the changes of C1ORAW, but also the output polarity set by CCTRL.*
Bit 3	C1OBEN	0x0	rw	Channel 1 output buffer enable 0: Buffer function of TMRx_C1DT is disabled. The new value written to the TMRx_C1DT takes effect immediately. 1: Buffer function of TMRx_C1DT is enabled. The value to be written to the TMRx_C1DT is stored in the buffer register, and can be sent to the TMRx_C1DT register only on an overflow event.
Bit 2	C1OIEN	0x0	rw	Channel 1 output enable immediately In PWM mode A or B, this bit is used to accelerate the channel 1 output's response to the trigger event. 0: Need to compare the CVAL with C1DT before generating an output 1: No need to compare the CVAL and C1DT. An output is generated immediately when a trigger event occurs.
Bit 1: 0	C1C	0x0	rw	Channel 1 configuration This field is used to define the direction of the channel 1 (input or output), and the selection of input pin when C1EN='0': 00: Output 01: Input, C1IN is mapped on C1IFP1 10: Input, C1IN is mapped on C2IFP1

				11: Input, C1IN is mapped on STCI. This mode works only when the internal trigger input is selected by STIS.

Input capture mode:

Bit	Name	Reset value	Type	Description
Bit 15: 8	Reserved	0x0	resd	Kept at default value.
Bit 7: 4	C1DF	0x0	rw	Channel 1 digital filter This field defines the digital filter of the channel 1. N stands for the number of filtering, indicating that the input edge can pass the filter only after N sampling events. 0000: No filter, sampling is done at f_{DTS} 1000: $f_{SAMPLING}=f_{DTS}/8$, N=6 0001: $f_{SAMPLING}=f_{CK\_INT}$, N=2 1001: $f_{SAMPLING}=f_{DTS}/8$, N=8 0010: $f_{SAMPLING}=f_{CK\_INT}$, N=4 1010: $f_{SAMPLING}=f_{DTS}/16$, N=5 0011: $f_{SAMPLING}=f_{CK\_INT}$, N=8 1011: $f_{SAMPLING}=f_{DTS}/16$, N=6 0100: $f_{SAMPLING}=f_{DTS}/2$, N=6 1100: $f_{SAMPLING}=f_{DTS}/16$, N=8 0101: $f_{SAMPLING}=f_{DTS}/2$, N=8 1101: $f_{SAMPLING}=f_{DTS}/32$, N=5 0110: $f_{SMPLING}=f_{DTS}/4$, N=6 1110: $f_{SAMPLING}=f_{DTS}/32$, N=6 0111: $f_{SAMPLING}=f_{DTS}/4$, N=8 1111: $f_{SAMPLING}=f_{DTS}/32$, N=8
Bit 3: 2	C1IDIV	0x0	rw	Channel 1 input divider This field defines Channel 1 input divider. 00: No divider. An input capture is generated at each active edge. 01: An input compare is generated every 2 active edges 10: An input compare is generated every 4 active edges 11: An input compare is generated every 8 active edges Note: the divider is reset once C1EN='0'
Bit 1: 0	C1C	0x0	rw	Channel 1 configuration This field is used to define the direction of the channel 1 (input or output), and the selection of input pin when C1EN='0': 00: Output 01: Input, C1IN is mapped on C1IFP1 10: Reserved 11: Reserved

14.3.4.6 TMR14 channel control register (TMRx_CCTRL)

Bit	Name	Reset value	Type	Description
Bit 15: 14	Reserved	0x0	resd	Kept at default value.
Bit 3	C1CP	0x0	rw	Channel 1 complementary polarity 0: C1COUT is active high. 1: C1COUT is active low.
Bit 2	Reserved	0x0	resd	Kept at default value.
Bit 1	C1P	0x0	rw	Channel 1 polarity When the channel 1 is configured as output mode: 0: C1OUT is active high 1: C1OUT is active low When the channel 1 is configured as input mode: The active edge of input signals are defined by C1LP/C1P. 00: C1IN active edge is on its rising edge. When used as external trigger, C1IN is not reversed. 01: C1IN active edge is on its falling edge. When used as external trigger, C1IN is reversed. 10: Reserved

				11: C1IN active edge is on its rising edge and falling edge. When used as external trigger, C1IN is not reversed.
Bit0	C1EN	0x0	rw	Channel 1 enable 0: Input or output is disabled 1: Input or output is enabled

Table 14-8 Standard CxOUT channel output control bits

CxEN bit	CxOUT output status
0	Output disabled (CxOUT=0)
1	CxOUT = CxORAW + polarity

Note: The state of the external I/O pins connected to the standard CxOUT channel depends on the CxOUT channel state and the GPIO and IOMUX registers.

14.3.4.7 TMR14 counter value (TMRx_CVAL)

Bit	Name	Reset value	Type	Description
Bit 15: 0	CVAL	0x0	rw	Counter value

14.3.4.8 TMR14 division value (TMRx_DIV)

Bit	Name	Reset value	Type	Description
Bit 15: 0	DIV	0x0	rw	Divider value The counter clock frequency fCK_CNT = fTMR_CLK /(DIV[15:0]+1). The value of this register is transferred to the actual prescaler register when an overflow event occurs.

14.3.4.9 TMR14 period register (TMRx_PR)

Bit	Name	Reset value	Type	Description
Bit 15: 0	PR	0x0	rw	Period value This defines the period value of the TMRx counter. The timer stops working when the period value is 0.

14.3.4.10 TMR14 channel 1 data register (TMRx_C1DT)

Bit	Name	Reset value	Type	Description
Bit 15: 0	C1DT	0x0	rw	Channel 1 data register When the channel 1 is configured as input mode: The C1DT is the CVAL value stored by the last channel 1 input event (C1IN) When the channel 1 is configured as output mode: C1DT is the value to be compared with the CVAL value. Whether the written value takes effective immediately depends on the C1OBEN bit, and the corresponding output is generated on C1OUT as configured.

14.3.4.11 TMR14 channel input remap register (TMR14_RMP)

Bit	Name	Reset value	Type	Description
Bit 15: 2	Reserved	0x00	resd	Kept at default value.
Bit 1: 0	TMR14_CH1_IRMP	0x0	rw	TMR14 channel 1 input remap 00: TMR14 channel 1 input is connected to GPIO 01: ERTC_CLK 10: HEXT/32 11: CLK_OUT

14.4 General-purpose timer (TMR15)

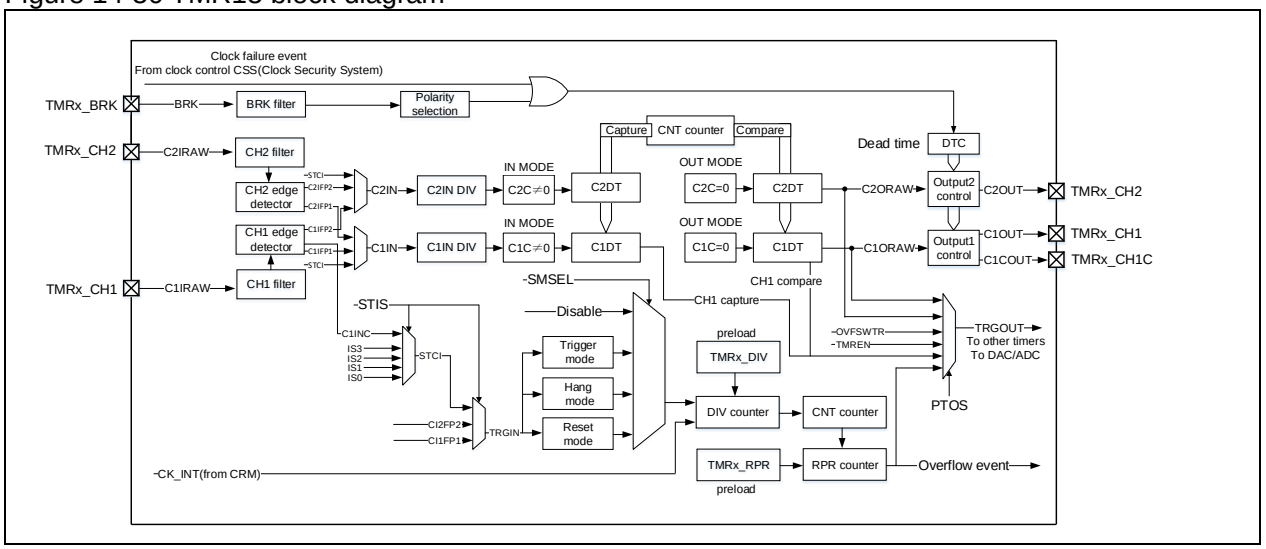
14.4.1 TMR15 introduction

The general-purpose timer (TMR15) consists of a 16-bit upcounter, two capture/compare registers, and two independent channels. They can be used for dead-time insertion, input capture and programmable PWM output.

14.4.2 TMR15 main features

- Clock source of counter: internal clock, external input and internal trigger input
- 16-bit upcounter, and 8-bit repetition counter
- Two independent channels for input capture, output compare, PWM generation, one-pulse mode output and dead-time insertion
- One independent channel for complementary output
- TMR brake function
- Synchronization control between master and slave timers
- Interrupt/DMA generation at overflow event, trigger event, brake input and channel event
- Support TMR burst DMA transfer

Figure 14-50 TMR15 block diagram

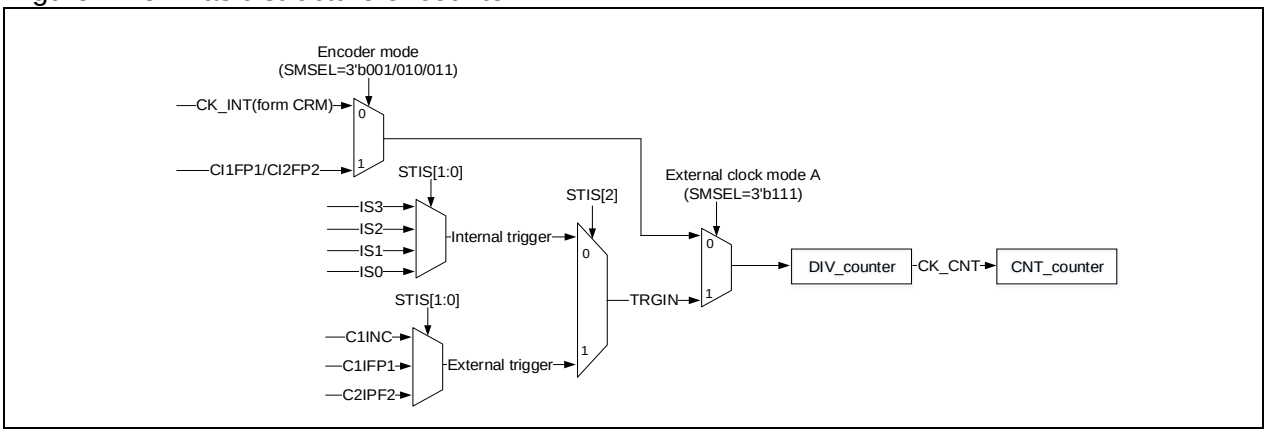


14.4.3 TMR15 functional overview

14.4.3.1 Counting clock

The counting clock of TMR15 can be provided by internal clock (CK_INT), external clock (external clock mode A) or internal trigger input (ISx).

Figure 14-51 Basic structure of counter

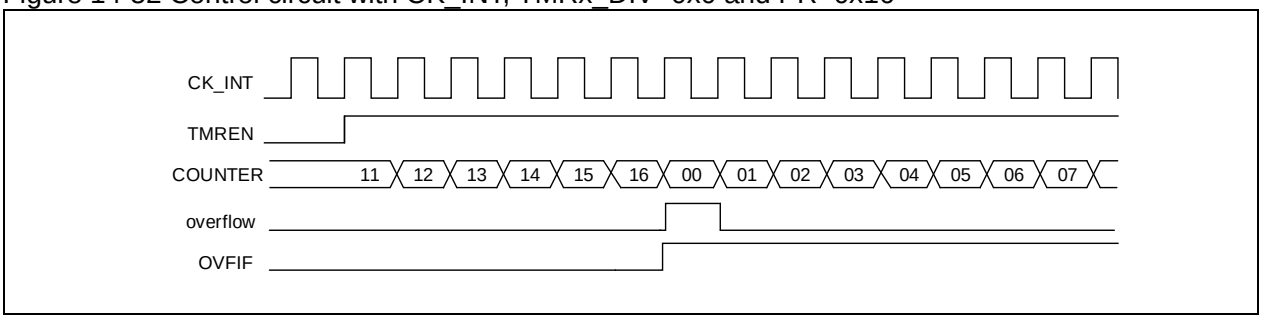


Internal clock (CK_INT)

By default, the CK_INT which is divided by the prescaler, is used to drive the counter to start counting. When TMR's APB clock prescaler factor is 1, the CK_INT frequency is equal to that of APB; otherwise, it doubles the APB clock frequency.

- Set counting frequency through TMRx_DIV register
- Set counting cycles through TMRx_PR register
- Enable a counter by setting the TMREN bit in the TMRx_CTRL1 register

Figure 14-52 Control circuit with CK_INT, TMRx_DIV=0x0 and PR=0x16



External clock (**TRGIN/EXT**)

The counter clock can be provided by the external clock source TRGIN.

SMSEL=3'b111: External clock mode A is selected. Select an external clock source TRGIN signal by setting the STIS[2: 0] bit to drive the counter to start counting.

The external clock sources include:

C1INC (STIS=3'b100, rising edge and falling edge of channel 1)

C1IFP1 (STIS=3'b101, the channel 1 signal which goes through filtering and polarity selection)

C2IFP2 (STIS=3'b110, a channel 2 signal which goes through filtering and polarity selection).

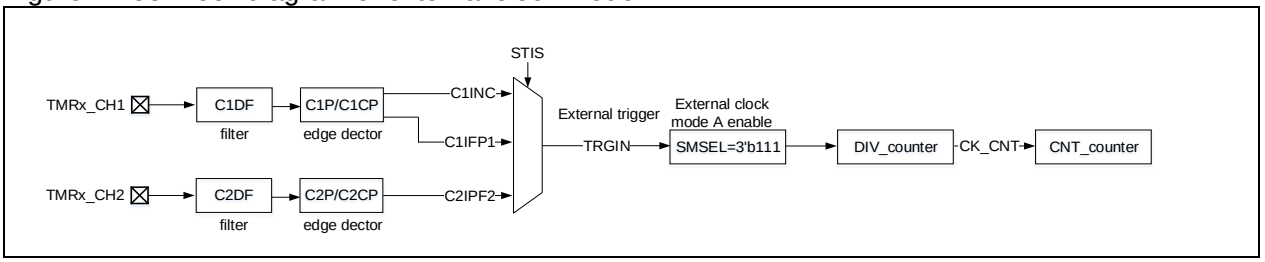
To use external clock mode A, follow the steps below:

- Set external source TRGIN parameters

  If the TMRx_CH1 is used as a source of TRGIN, it is necessary to configure channel 1 input filter (C1DF[3:0] in TMRx_CM1 register) and channel 1 input polarity (C1P/C1CP in TMRx_CCTRL register);

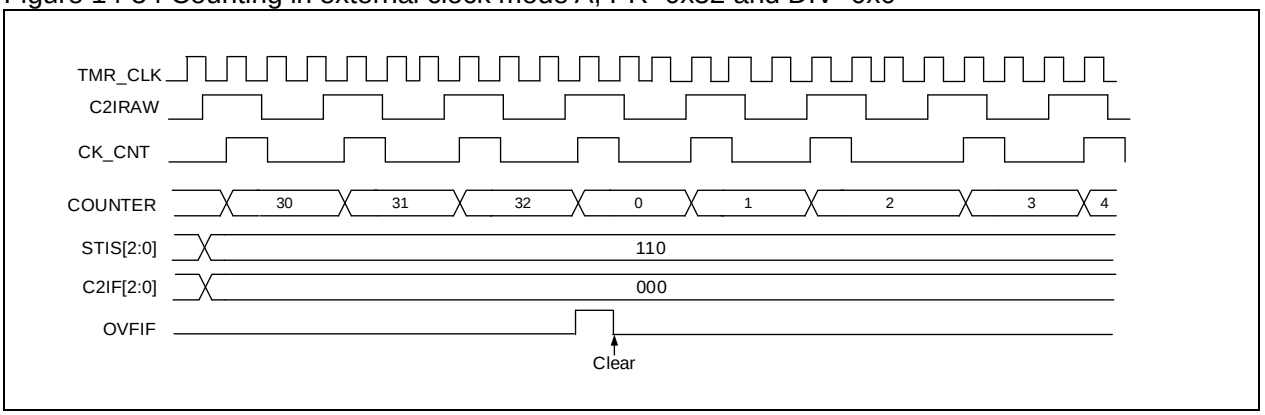
  If the TMRx_CH2 is used as source of TRGIN, it is necessary to configure channel 2 input filter (C2DF[3:0] in TMRx_CM1 register) and channel 2 input polarity (C2P/C2CP in TMRx_CCTR register);
- Set TRGIN signal source using the STIS[1:0] bit in TMRx_STCTRL register
- Enable external clock mode A by setting SMSEL=3'b111 in TMRx_STCTR register
- Set counting frequency through the DIV[15:0] in TMRx_DIV register
- Set counting period through the PR[15:0] in TMRx_PR register
- Enable counter through the TMREN bit in TMRx_CTRL1 register

Figure 14-53 Block diagram of external clock mode A



Note: The delay between the signal on the input side and the actual clock of the counter is due to the synchronization circuit

Figure 14-54 Counting in external clock mode A, PR=0x32 and DIV=0x0



Internal trigger input (ISx)

Timer synchronization allows interconnection between several timers. The TMR_CLK of one timer can be provided by the TRGOUT signal from another timer. The internal trigger signal is selected by setting the STIS[2: 0] bit to enable counting.

TMR15 consists of a 16-bit prescaler, which is used to generate the CK_CNT that enables the counter to count. The frequency division relationship between the CK_CNT and TMR_CLK can be adjusted by setting the TMR15_DIV register. The prescaler value can be modified at any time, but the new prescaler value is taken into account when the next overflow event occurs.

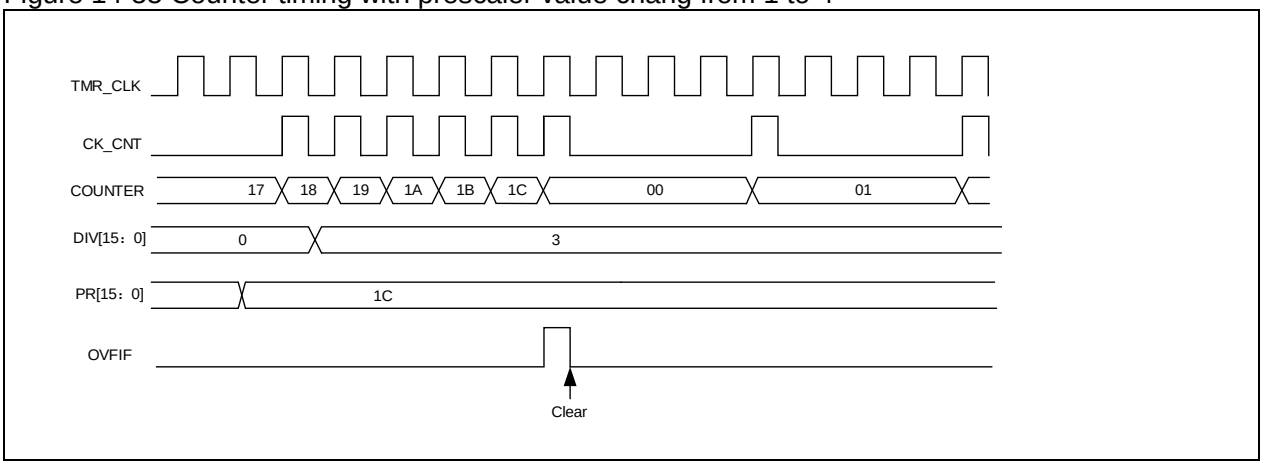
Below is the configuration procedure for internal trigger input:

- Set counting cycles through TMRx_PR register
- Set counting frequency through TMRx_DIV register
- Set counting modes through the TWCMSEL[1:0] in TMRx_CTRL1 register
- Select internal trigger by setting STIS[2:0]= 3’b000~3’b011 in TMRx_STCTRL register
- Select external clock mode A by setting SMSEL[2:0]=3’b111 in TMRx_STCTRL register
- Enable TMRx to start counting through the TMREN in TMRx_CTRL1 register

Table 14-9 TMRx internal trigger connection

Slave controller	IS0 (STIS = 000)	IS1 (STIS = 001)	IS2 (STIS = 010)	IS3 (STIS = 011)
TMR1	TMR15	-	TMR3	-
TMR3	TMR1	-	TMR15	-
TMR15	-	TMR3	TMR16	TMR17_OC

Figure 14-55 Counter timing with prescaler value chang from 1 to 4



14.4.3.2 Counting mode

The TMR15 supports multiple counting modes to meet various application scenarios. It features a 16-bit upcounter.

The TMRx_PR register is used to define counting period of counter. The value in the TMRx_PR is immediately moved to the shadow register by default. When the periodic buffer is enabled (PRBEN=1), the value in the TMRx_PR register is transferred to the shadow register only at an overflow event.

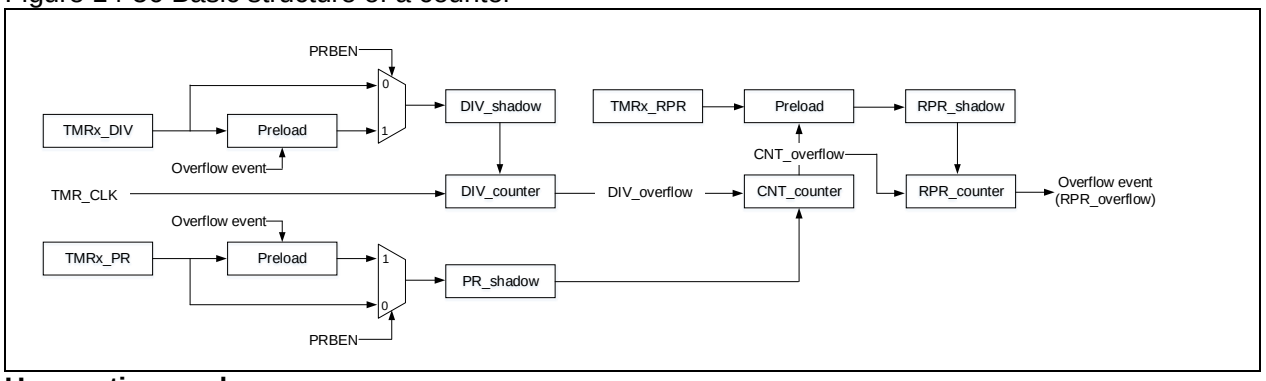
TMRx_DIV register is used to define the counter frequency of the counter. The counter counts once every DIV[15:0]+1 clock cycle. Similar to TMRx_PR register, after the periodic buffer is enabled, the value of the TMRx_DIV register is transferred into the shadow register upon an overflow event.

Reading the TMRx_CNT register returns the current counter value. Writing the TMRx_CNT register will update the current counter value.

An overflow event is enabled by default. It can be disabled by setting OVFEN=1 in the TMRx_CTRL1 register. The OVFS bit in the TMRx_CTRL1 register is used to select the source of an overflow event, which is, by default, counter overflow or underflow, setting OVFSWTR, reset signal generated by slave mode timer controller in reset mode. Once the OVFS is set, an overflow event is generated only when overflow or underflow occurs.

Setting the TMREN bit (TMREN=1) enables the timer to start counting. Base on synchronization logic, however, the actual enable signal TMR_EN is set 1 clock cycle after the TMREN is set.

Figure 14-56 Basic structure of a counter



Upcounting mode

This mode is enabled by setting CMSEL[1:0]=2'b00 and OWCDIR=1'b0 in the TMRx_CTRL1 register.

In upcounting mode, the counter counts from 0 to the value programmed in the TMRx_PR register, restarts from 0, and generates a counter overflow event, with setting OVFIF bit to 1. If the overflow event is disabled, the counter is no longer reloaded with the prescaler and period value on counter overflow, otherwise, the prescaler and period value will be updated on an overflow event.

Figure 14-57 Overflow event when PRBEN=0

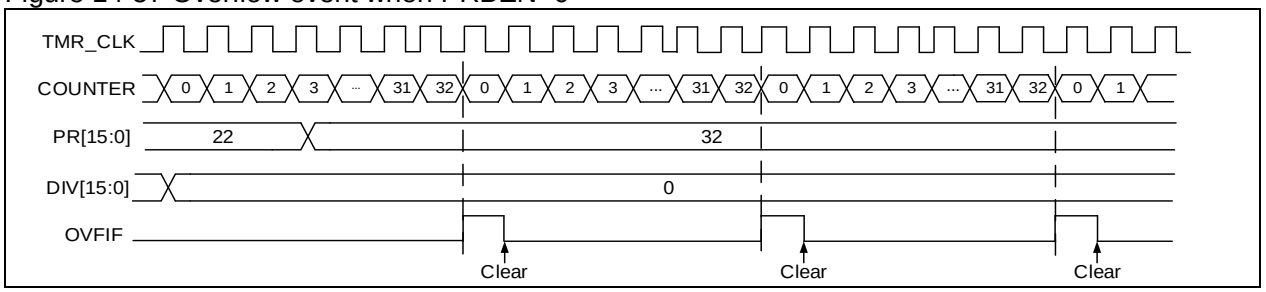
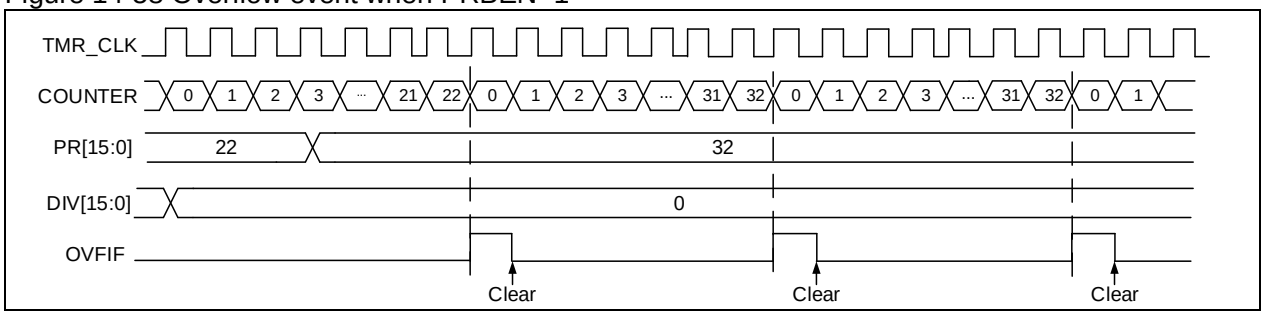


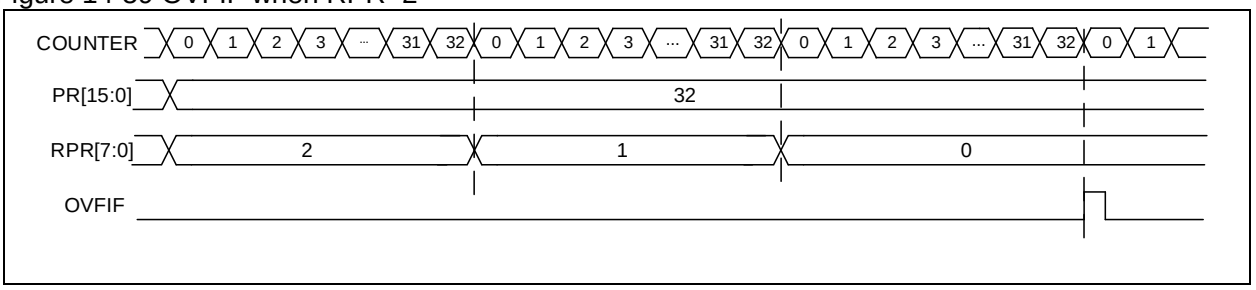
Figure 14-58 Overflow event when PRBEN=1



Repetition counter mode:

The TMRx_RPR register is used to enable repetition counting mode. This mode is enabled when the repetition counter value is not equal to 0. In this mode, an overflow event is generated when a counter overflow occurs (RPR[7:0]+1). The repetition counter is decremented at each counter overflow. An overflow event is generated when the repetition counter reaches 0. The frequency of the overflow event can be adjusted by setting the repetition counter value.

Figure 14-59 OVFIF when RPR=2



14.4.3.3 TMR input function

TMR15 has two independent channels that can be configured in input or output mode.

As input, the channel input signal is processed as follows:

- TMRx_CHx outputs the pre-processed CxIRAW. The C1INSE bit is used to select TMRx_CHx as the source of C1IRAW
- CxIRAW passes digital filter and outputs a filtered CxIF signal. The digital filter uses the CxDF bit to program sampling frequency and sampling times.
- CxIF passes edge detector, and outputs the CxIFPx signal with edge selection. The edge selector is controlled by CxP and CxCP bits. It provides input rising edge, falling edge and both edges for selection.
- CxIFPx passes capture signal selector, and outputs the CxIN signal with capture signal selection. The capture signal selector is controlled by CxC bit. The CxIN source can be from CxIFPx, CyIFPx or STCI. Of those, CyIFPx (x≠y) refers to the CyIFPy signal which is from Y channel and has passed channel-x edge detector (for example, C1IFP2 refers to the C1IFP1 on channel 1 that has passed channel 2 edge detector). The STCI comes from slave timer controller, and its source is defined by STIS bit.
- CxIN passes input divider and outputs CxIPS signal. The divider factor can be defined as No division, divided/2, divided /4 or divided /8, through the CxIDIV bit.

Figure 14-60 Input/output channel 1 main circuit

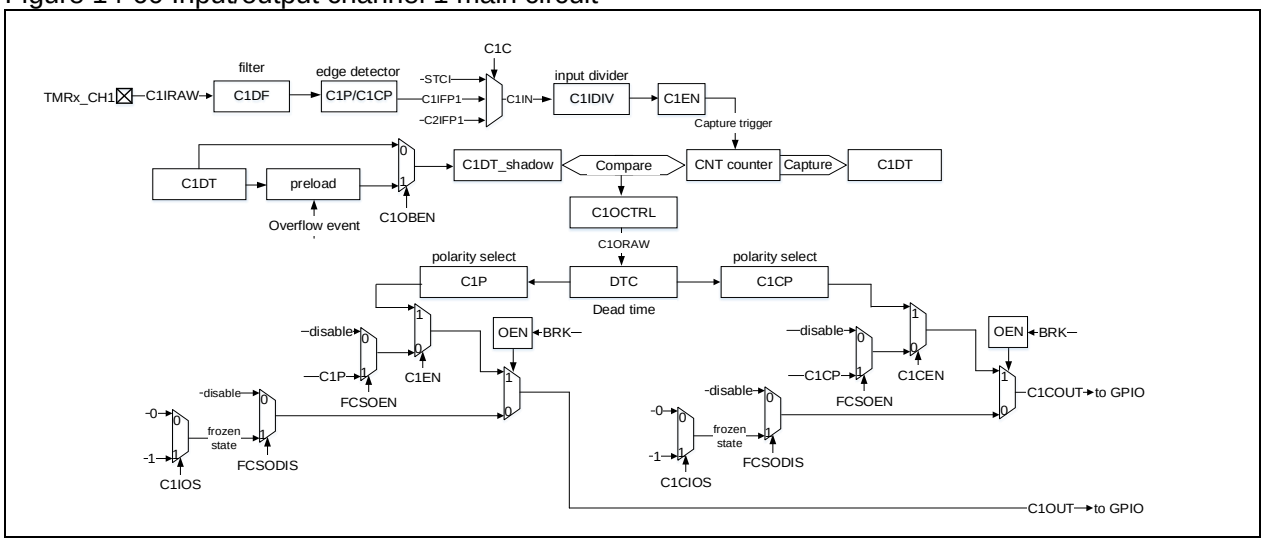
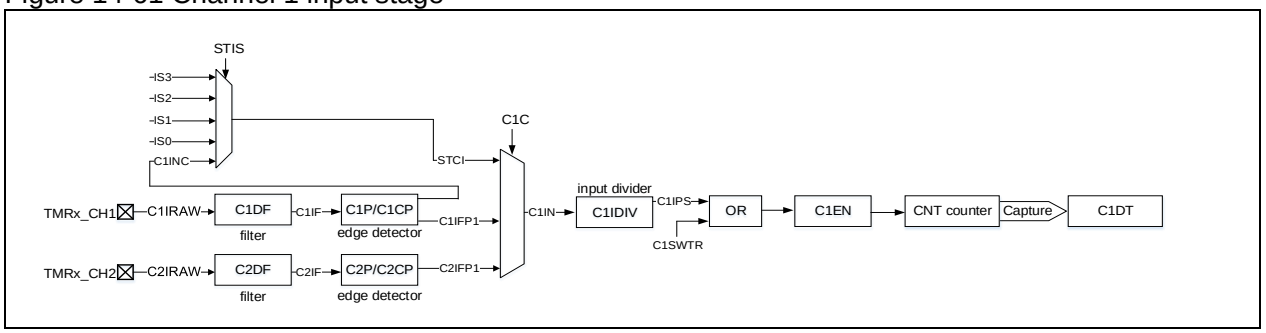


Figure 14-61 Channel 1 input stage



Input mode

In input mode, the TMRx_CxDT registers latches the current counter values after the selected trigger signal is detected, and the capture compare interrupt flag bit (CxIF) is set to 1. An interrupt/DMA request will be generated if the CxIEN bit and CxDEN bit are enabled. If the selected trigger signal is detected when the CxIF is set to 1, a capture overflow event is generated, the previous counter value will be overwritten with the current counter value, and the CxRF is set to 1.

To capture the rising edge of C1IN input, following the procedure below:

- Set C1C=01 in the TMR15_CM1 register to select the C1IN as channel 1 input
- Set C1IN signal filter bandwidth (CxDF[3: 0])
- Set the active edge of C1IN channel by writing C1P=0 (rising edge) in the TMR15_CCTRL register
- Program C1IN signal capture frequency divider (C1DIV[1: 0])
- Enable channel 1 input capture (C1EN=1)
- If needed, enable the relevant interrupt or DMA request by setting the C1IEN bit in the TMR15_IDEN register or the C1DEN bit in the TMR15_IDEN register

PWM input

PWM input mode is applied to channel 1 and 2. To use this mode, both C1IN and C2IN are mapped on the same TMRx_CHx, and the CxIFPx of either channel 1 or channel 2 must be configured as trigger input and slave mode controller is configured in reset mode.

The PWM input mode can be used to measure the period and duty cycle of the PWM input signal. For example, the user can measure the period and duty cycle of the PWM applied on channel 1 using the following procedures:

- Set C1C=2‘b01: select C1IN for C1IFP1
- Set C1P=1’b0, select C1IFP1 rising edge active
- Set C2C=2‘b10, select C2IN for C1IFP2
- Set C2P=1’b1, select C1IFP2 falling edge active
- Set STIS=3’b101, select the slave mode timer trigger signal as C1IFP1

- Set SMSEL=3‘b100: configure the slave mode controller in reset mode
- Set C1EN=1’b1 and C2EN=1’b1. Enable channel 1 and input capture

After above configuration, the rising edge of channel 1 input signal will trigger the capture and stores the capture value into C1DT register, and it will reset the counter at the same time. The falling edge of the channel 1 input signal triggers the capture and stores the capture value into C2DT register. The period of the channel 1 input signal is calculated through C1DT, while its duty cycle is obtained through C2DT.

Figure 14-62 PWM input mode configuration example

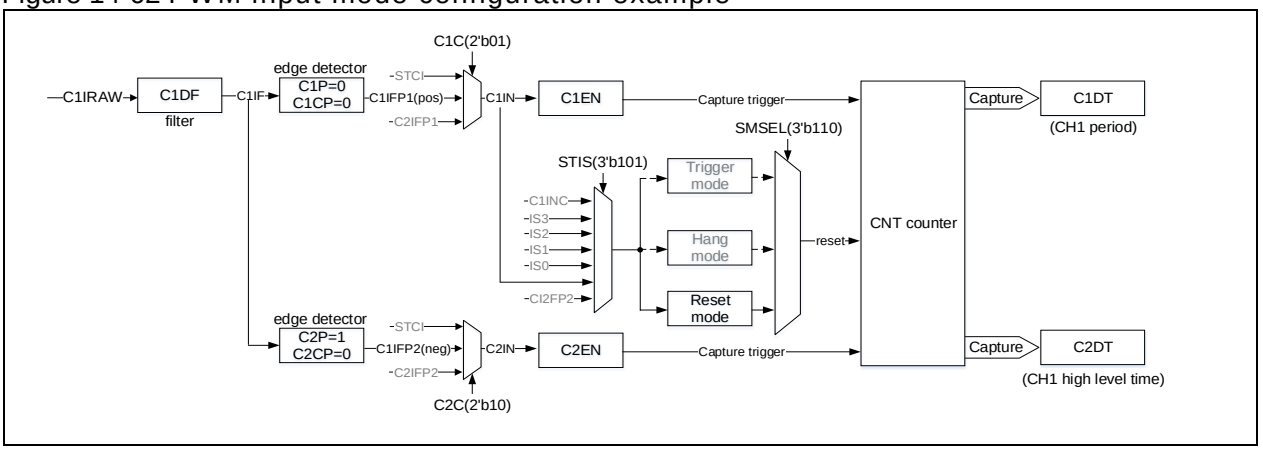
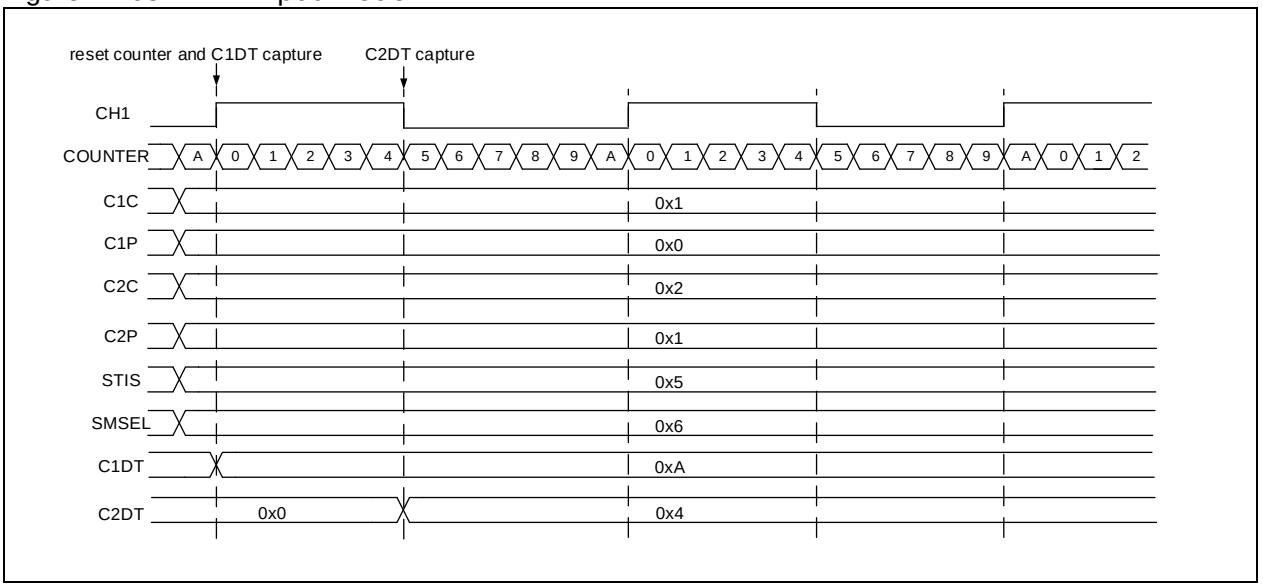


Figure 14-63 PWM input mode



14.4.3.4 TMR output function

The TMR output consists of a comparator and an output controller. It is used to program the period, duty cycle and polarity of output signals.

Figure 14-64 Channel 1 output stage

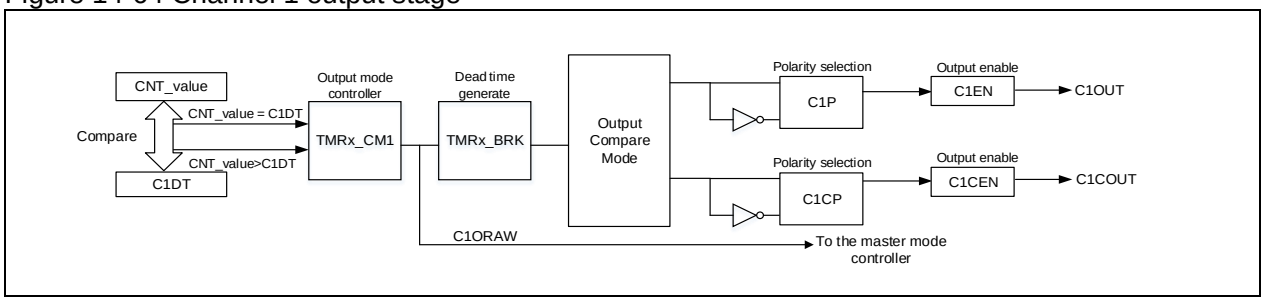
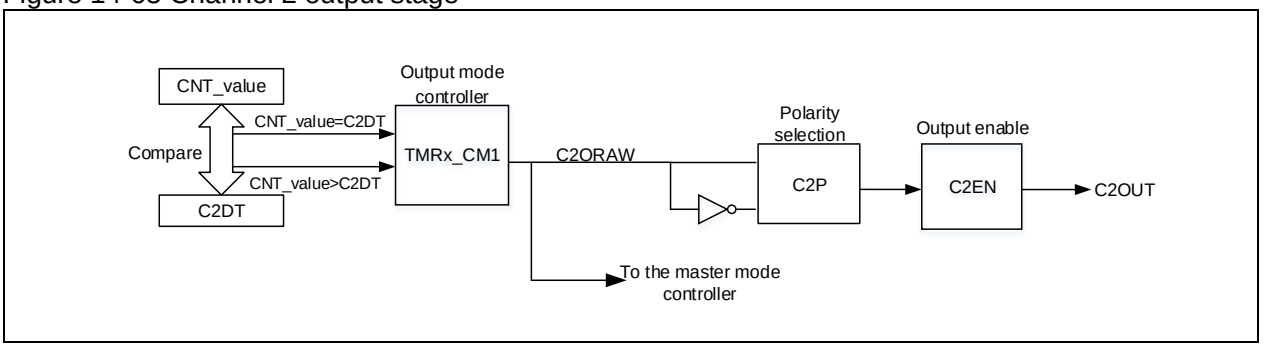


Figure 14-65 Channel 2 output stage



Output mode

CxC[1: 0]≠2'b00 is used to configure the channel as output mode. In this case, the counter value is compared with that in the CxDT register, and the intermediate signal CxORAW is generated according to the output mode selected by CxOCTRL[2: 0], and this signal is sent to IO after being processed by the output control circuit. The period of the output signal is configured by the TMR15_PR register, while the duty cycle is configured by the CxDT register.

Output compare modes include:

PWM mode A:

Enable PWM mode A by setting CxOCTRL=3'b110. In upcounting mode, C1ORAW outputs high when TMRx_C1DT>TMRx_CVAL, otherwise, it is low; in downcounting mode, C1ORAW outputs low when TMRx_C1DT<TMRx_CVAL, otherwise, it is high.

To use PWM mode A, the following procedures are recommended:

- Set PWM periods through TMRx_PR register
- Set PWM duty cycles through TMRx_CxDT register
- Select PWM mode A by setting CxOCTRL=3'b110 in the TMRx_CM1/CM2 register
- Set counting frequency through TMRx_DIV register
- Select counting mode by setting the TWCMSEL[1:0] bit in the TMRx_CTRL1 register
- Select output polarity through the CxP and CxCP bits in the TMRx_CCTRL register
- Enable channel output through the CxEN and CxCEN bits in the TMRx_CCTRL register
- Enable TMRx output through the OEN bit in the TMRx_BRK register
- Configure GPIOs corresponding to TMR output channels as multiplexed mode
- Enable TMRx to start counting through the TMREN bit in the TMRx_CTRL1 register.

PWM mode B:

Enable PWM mode B by setting CxOCTRL=3'b111. In upcounting mode, C1ORAW outputs low when TMRx_C1DT>TMRx_CVAL, otherwise, it is high; in downcounting mode, C1ORAW outputs high when TMRx_C1DT<TMRx_CVAL, otherwise, it is low.

Forced output mode:

Enable forced output mode by setting CxOCTRL=3'b100/101. In this mode, the CxORAW is forced to be the programmed level, regardless of the counter value. Despite this, the channel flag bit and DMA request still depend on the compare result.

Output compare mode:

Enable output compare mode by setting CxOCTRL=3'b001/010/011. In this mode, when the counter value matches the value of the CxDT register, the CxORAW is forced high (CxOCTRL=3'b001), low (CxOCTRL=3'b010) or toggled (CxOCTRL=3'b011).

One-pulse mode:

This is a particular case of PWM mode. Enable one-pulse by setting OCMEN=1. In this mode, the comparison match is performed in the current counting period. The TMREN bit is cleared as soon as the current counting is completed. Therefore, only one pulse is output. When in upcounting mode, the configuration must follow the rule: CVAL<CxDT≤PR; in downcounting mode, CVAL>CxDT is required.

Fast output mode:

Enable this mode by setting CxOIEN=1. If enabled, the CxORAW signal will not change when the counter value matches the CxDT, but change at the beginning of the current counting period. In other

words, the comparison result is advanced, so the comparison result between the counter value and the TMRx_CxDT register will determine the level of CxORAW in advance.

Figure 14-66 gives an example of output compare mode (toggle) with C1DT=0x3. When the counter value is equal to 0x3, C1OUT toggles.

Figure 14-67 gives an example of the combination between upcounting mode and PWM mode A. The output signal behaves when PR=0x32 but CxDT is configured with a different value.

Figure 14-68 gives an example of the combination between upcounting mode and one-pulse PWM mode B. The counter only counts only one cycle, and the output signal sends only one pulse.

Figure 14-66 C1ORAW toggles when counter value matches the C1DT value

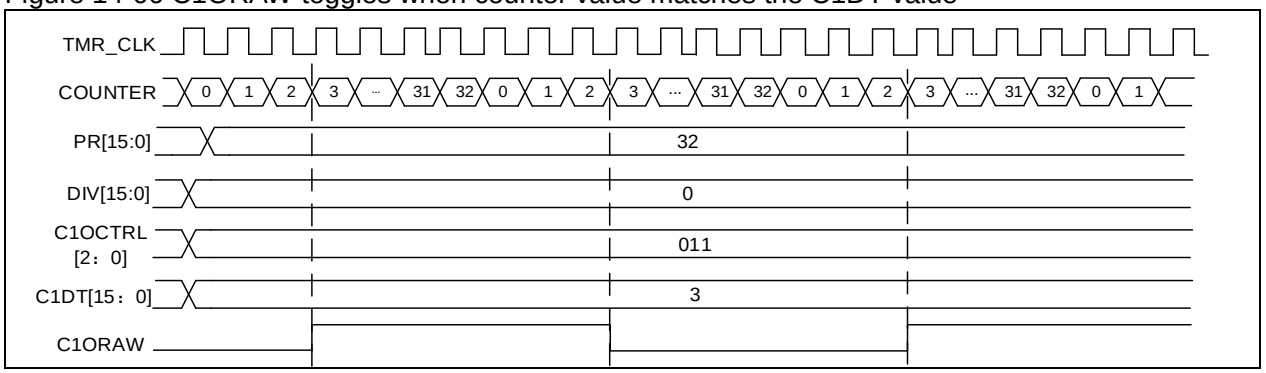


Figure 14-67 Upcounting mode and PWM mode A

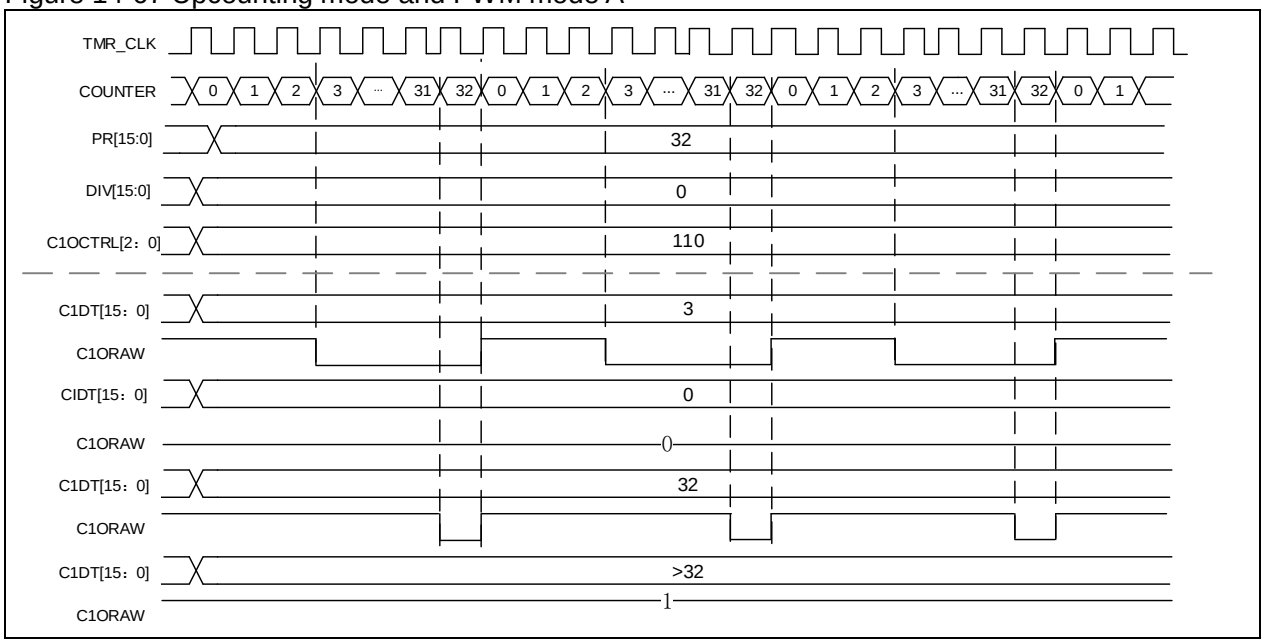
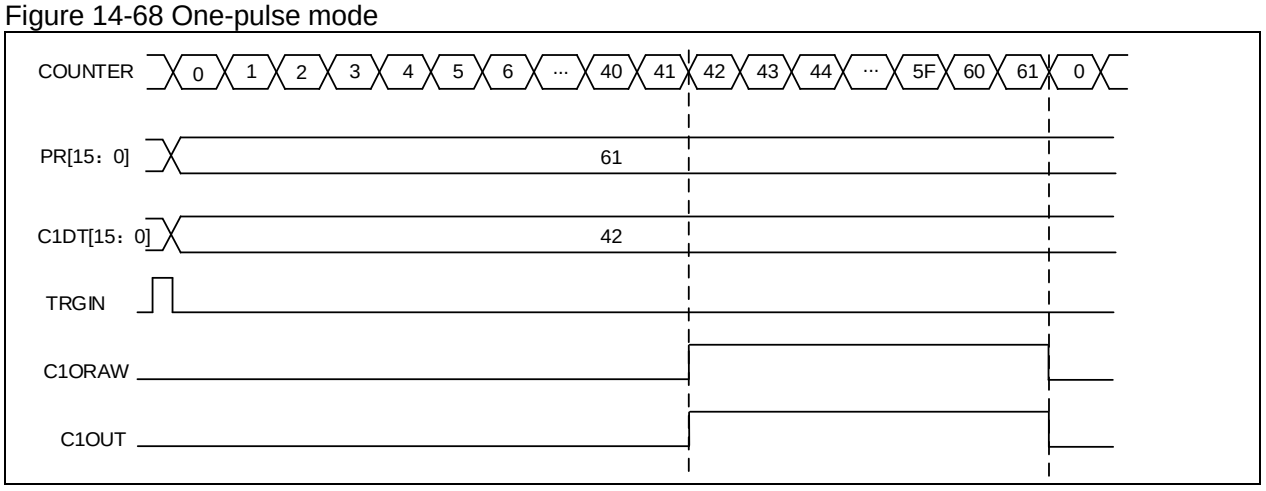


Figure 14-68 One-pulse mode



Master mode timer event output

When TMR is used as a master timer, one of the following source of signals can be selected as TRGOUT output to a slave mode timer. This is done by setting the PTOS bit in the TMRxCTRL2 register.

- PTOS=3'b000, TRGOUT output software overflow event (OVFSWTR bit in TMRx_SWEVT register)
- PTOS=3'b001, TRGOUT output counter enable
- PTOS=3'b010, TRGOUT output counter overflow event
- PTOS=3'b011, TRGOUT output capture and compare event
- PTOS=3'b100, TRGOUT output C1ORAW
- PTOS=3'b101, TRGOUT output C2ORAW

Dead-time insertion

The TMR15 contains a set of reverse channel output. This function is enabled by the CxCEN bit and its polarity is selected by CxCP. Refer to Table 14-11 for more information about the output state of CxOUT and CxCOUT.

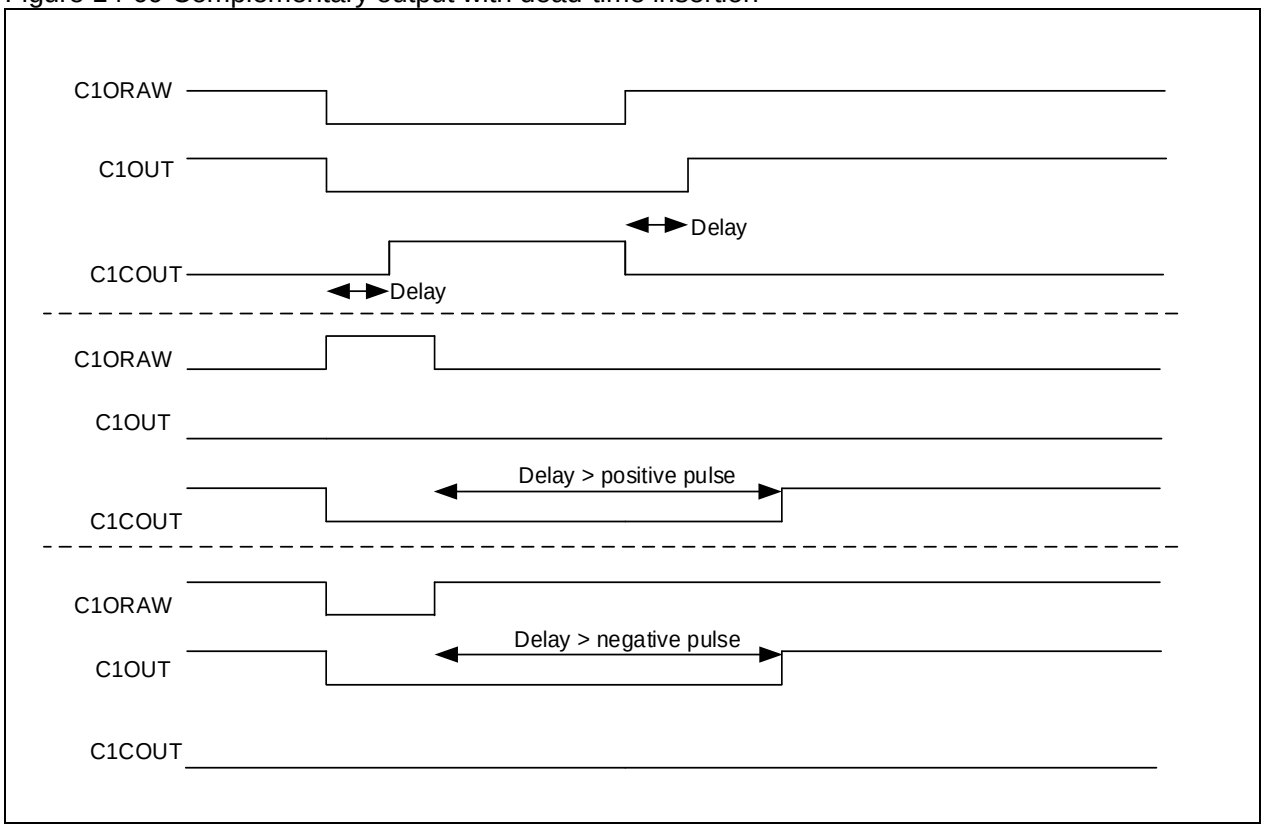
The dead-time is activated when switching to IDLEF state (OEN falling down to 0).

Set both CxEN and CxCEN bits to 1, and use DTC[7:0] bit to insert dead-time of different durations. After the dead-time insertion, the rising edge of the CxOUT is delayed compared to the rising edge of the reference signal; the rising edge of the CxCOU is delayed compared to the falling edge of the reference signal.

If the delay is greater than the width of the active output width, C1OUT and C1COUT will not generate corresponding pulses. Therefore, the dead-time should be less than the width of the active output.

Figure 14-69 gives an example of dead-time insertion when CxP=0, CxCP=0, OEN=1, CxEN=1 and CxCEN=1.

Figure 14-69 Complementary output with dead-time insertion



14.4.3.5 TMR brake function

When the brake function is enabled (BRKEN=1), the CxOUT and CxCOUT are jointly controlled by OEN, FCSODIS, FCSOEN, CxIOS and CxCIOS. But, CxOUT and CxCOUT cannot always be set to active levels at the same time. Please refer to Table 14-17 for more details.

The brake source can be from brake input pin or a clock failure event. The brake input polarity is defined by the BRKV bit.

When a brake event occurs, there are the following actions:

- The OEN bit is cleared asynchronously, and the channel output state is selected by setting the FCSODIS bit. This function works even if the MCU oscillator is off.
- After OEN is cleared, the channel output level is defined by the CxIOS bit. If FCSODIS=0, the timer output is disabled, otherwise, the output enable remains high.
- When a complementary output mode is used:
  - The output is first put in reset state, that is, inactive state (depending on the polarity). This is done asynchronously so that it works even if no clock is provided to the timer.
  - If the timer clock is still active, then the dead-time generator is activated. The CxIOS and CxCIOS bits are used to program the level after dead-time. Even in this case, the CxIOS and CxCIOS cannot be driven to their active levels at the same time.

Note: Because of synchronization logic on OEN bit, the dead-time duration is usually longer than usual (around 2 clk_tmr clock cycles)

  - If FCSODIS=0, the timer releases the enable output, otherwise, it keeps the enable output; the enable output becomes high as soon as one of the CxEN and CxCEN bits becomes high.
- If the brake interrupt or DMA request is enabled, the brake statue flag is set, and a break interrupt or DMA request can be generated.
- If AOEN=1, the OEN bit is automatically set to 1 at the next overflow event.

Note: When the brake input is active, the OEN cannot be set, nor the status flag BRKIF can be cleared.

Figure 14-70 TMR output control

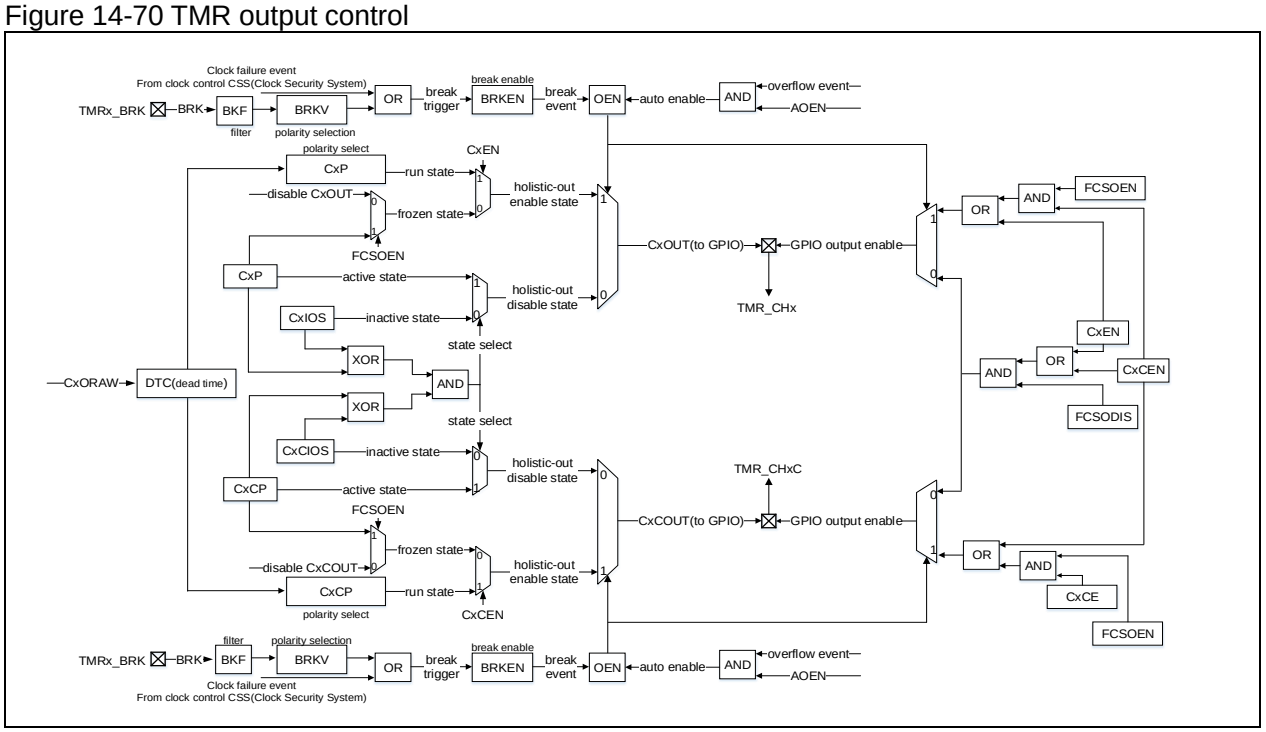
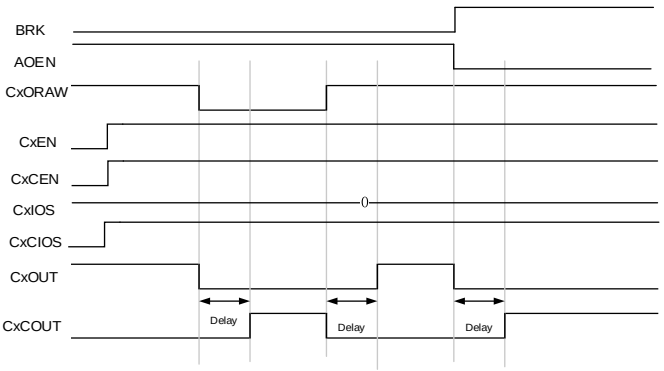


Figure 14-71 Example of TMR brake function



14.4.3.6 TMR synchronization

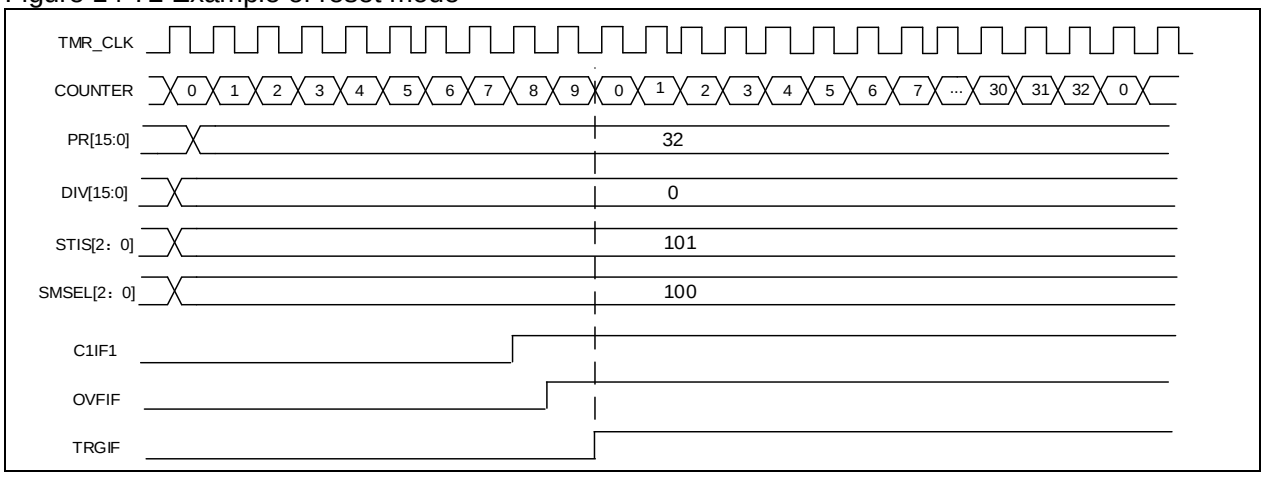
The master and slave timers are linked together internally for timer synchronization. Master mode timer is selected by setting the PTOS[2: 0] bit; Slave timer is selected by setting the SMSEL[2: 0] bit.

Slave modes include:

Slave mode: Reset mode

The counter and its prescaler can be reset by a selected trigger signal. An overflow event can be generated when OVFS=0.

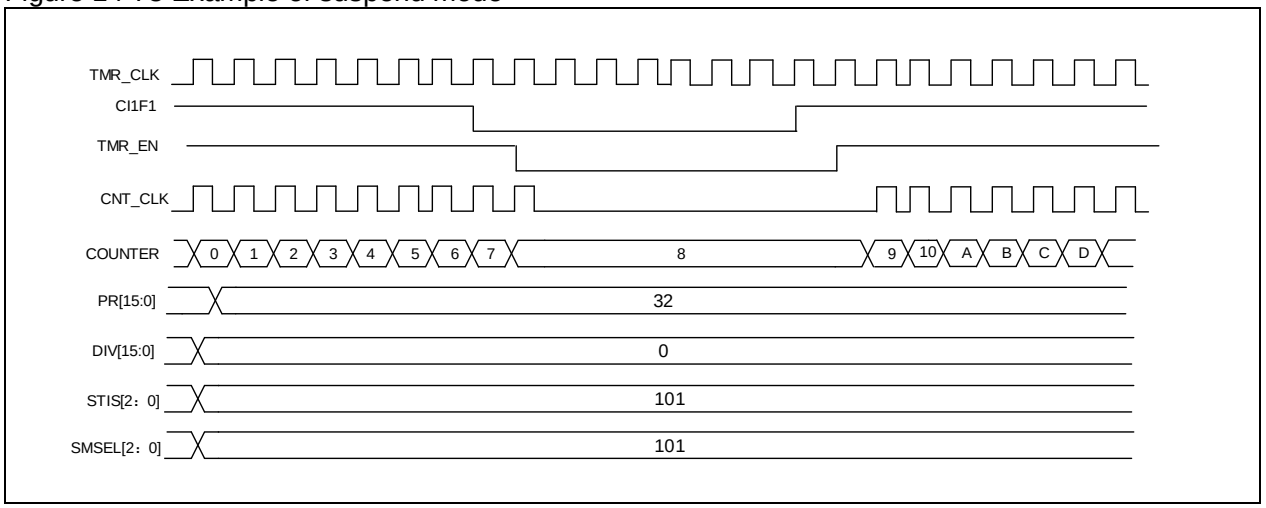
Figure 14-72 Example of reset mode



Slave mode: Suspend mode

In this mode, the counter is controlled by a selected trigger input. The counter starts counting when the trigger input is high and stops as soon as the trigger input is low.

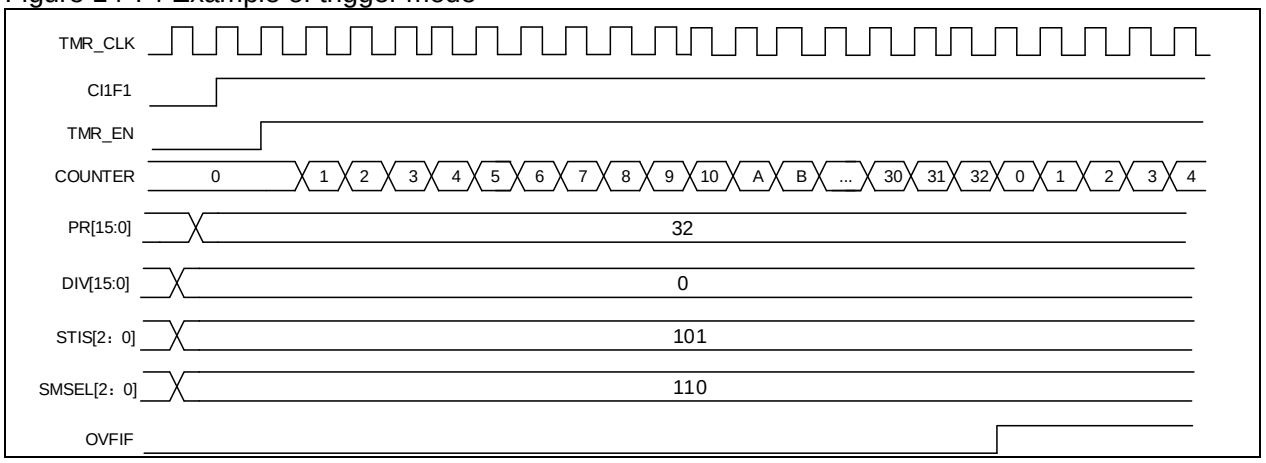
Figure 14-73 Example of suspend mode



Slave mode: Trigger mode

The counter starts counting on the rising edge of a selected trigger input (TMR_EN=1)

Figure 14-74 Example of trigger mode



See *Chapter 14.2.3.5* for more information about timer synchronization.

14.4.3.7 Debug mode

When the microcontroller enters debug mode (Cortex®-M0+ core halted), the TMR15 counter stops counting by setting the TMR15_PAUSE in the DEBUG module.

14.4.4 TMR15 registers

These peripheral registers must be accessed by half-words (16 bits) or words (32 bits).

TMR14 registers are mapped into a 16-bit addressable space.

Table 14-10 TMR15 register map and reset value

Register	Offset	Reset value
TMR15_CTRL1	0x00	0x0000
TMR15_CTRL2	0x04	0x0000
TMR15_STCTRL	0x08	0x0000
TMR15_IDEN	0x0C	0x0000
TMR15_ISTS	0x10	0x0000
TMR15_SWEVT	0x14	0x0000
TMR15_CM1	0x18	0x0000
TMR15_CCTRL	0x20	0x0000
TMR15_CVAL	0x24	0x0000
TMR15_DIV	0x28	0x0000
TMR15_PR	0x2C	0x0000
TMR15_RPR	0x30	0x0000
TMR15_C1DT	0x34	0x0000
TMR15_C2DT	0x38	0x0000
TMR15_BRK	0x44	0x0000
TMR15_DMACTRL	0x48	0x0000
TMR15_DMADT	0x4C	0x0000

14.4.4.1 TMR15 control register1 (TMR15_CTRL1)

Bit	Name	Reset value	Type	Description
Bit 15: 10	Reserved	0x0	resd	Kept at default value
Bit 9: 8	CLKDIV	0x0	rw	Clock divider This field is used to define the division ratio between digital filter sampling frequency (f_{DTS}) and timer clock frequency (f_{CK_INT}). It is also used to set the division ratio between dead time base (T_{DTS}) and timer clock period (T_{CK_INT}). 00: No division, $f_{DTS}=f_{CK\_INT}$ 01: Divided by 2, $f_{DTS}=f_{CK\_INT}/2$ 10: Divided by 4, $f_{DTS}=f_{CK\_INT}/4$ 11: Reserved
Bit 7	PRBEN	0x0	rw	Period buffer enable 0: Period buffer is disabled 1: Period buffer is enabled
Bit 6: 4	Reserved	0x0	resd	Kept at default value
Bit 3	OCMEN	0x0	rw	One cycle mode enable This bit is use to select whether to stop counting at an update event 0: The counter does not stop at an update event 1: The counter stops at an update event
Bit 2	OVFS	0x0	rw	Overflow event source This bit is used to select overflow event or DMA request sources. 0: Counter overflow, setting the OVFSWTR bit or overflow event generated by slave timer controller 1: Only counter overflow generates an overflow event

Bit 1	OVFEN	0x0	rw	Overflow event enable 0: Enabled 1: Disabled
Bit 0	TMREN	0x0	rw	TMR enable 0: Enabled 1: Disabled

14.4.4.2 TMR15 control register 2 (TMR15_CTRL2)

Bit	Name	Reset value	Type	Description
Bit 31: 11	Reserved	0x0	resd	Kept at default value.
Bit 10	C2IOS	0x0	rw	Channel 2 idle output state
Bit 9	C1CIOS	0x0	rw	Channel 1 complementary idle output state Output disabled (OEN= 0) after dead-time: 0: C1OUTL=0 1: C1OUTL=1
Bit 8	C1IOS	0x0	rw	Channel 1 idle output state Output disabled (OEN = 0), after dead-time: 0: C1OUT=0 1: C1OUT=1
Bit 7	Reserved	0x0	resd	Kept at default value.
Bit 6: 4	PTOS	0x0	rw	Master TMR output selection This field is used to select the TMRx signal sent to the slave timer. 000: Reset 001: Enable 010: Overflow 011: Compare pulse 100: C1ORAW signal 101: C2ORAW signal
Bit 3	DRS	0x0	rw	DMA request source 0: Channel event 1: Overflow event
Bit 2	CCFS	0x0	rw	Channel control bit flash selection This bit only acts on channels with complementary output. If the channel control bits are buffered: 0: Control bits are updated by setting the HALL bit 1: Control bits are updated by setting the HALL bit or a rising edge on TRGIN.
Bit 1	Reserved	0x0	resd	Kept at default value.
Bit 0	CBCTRL	0x0	rw	Channel buffer control This bit acts on channels that have complementary output. 0: CxEN, CxCEN and CxOCTRL bits are not buffered. 1: CxEN, CxCEN and CxOCTRL bits are buffered.

14.4.4.3 TMR15 slave timer control register (TMR15_STCAL)

Bit	Name	Reset value	Type	Description
Bit 31: 8	Reserved	0x0	resd	Kept at default value.
Bit 7	STS	0x0	rw	Subordinate TMR synchronization If enabled, master and slave timer can be synchronized. 0: Disabled 1: Enabled
Bit 6: 4	STIS	0x0	rw	Subordinate TMR input selection This field is used to select the subordinate TMR input. 000: Internal selection 0 (IS0) 001: Internal selection 1 (IS1) 010: Internal selection 2 (IS2) 011: Internal selection 3 (IS3) 100: C1IRAW input detector (C1INC) 101: Filtered input 1 (C1IF1) 110: Filtered input 2 (C1IF2) Please refer to Table 14-9 for more information on ISx for

				each timer.
Bit 3	Reserved	0x0	resd	Kept at default value.
Bit 2: 0	SMSEL	0x0	rw	Subordinate TMR mode selection 000: Slave mode is disabled 100: Reset mode — Rising edge of the TRGIN input reinitializes the counter 101: Suspend mode — The counter starts counting when the TRGIN is high 110: Trigger mode — A trigger event is generated at the rising edge of the TRGIN input 111: External clock mode A — Rising edge of the TRGIN input clocks the counter

14.4.4.4 TMR15 DMA/interrupt enable register (TMR15_IDEN)

Bit	Name	Reset value	Type	Description
Bit 15	Reserved	0x0	resd	Kept at default value.
Bit 14	TDEN	0x0	rw	Trigger DMA request enable 0: Disabled 1: Enabled
Bit 13	HALLDE	0x0	rw	HALL DMA request enable 0: Disabled 1: Enabled
Bit 12: 11	Reserved	0x0	resd	Kept at default value.
Bit 10	C2DEN	0x0	rw	Channel 2 DMA request enable 0: Disabled 1: Enabled
Bit 9	C1DEN	0x0	rw	Channel 1 DMA request enable 0: Disabled 1: Enabled
Bit 8	OVFDEN	0x0	rw	Overflow event DMA request enable 0: Disabled 1: Enabled
Bit 7	BRKIE	0x0	rw	Brake interrupt enable 0: Disabled 1: Enabled
Bit 6	TIEN	0x0	rw	Trigger interrupt enable 0: Disabled 1: Enabled
Bit 5	HALLIEN	0x0	rw	HALL interrupt enable 0: Disabled 1: Enabled
Bit 4: 3	Reserved	0x0	resd	Kept at default value.
Bit 2	C2IEN	0x0	rw	Channel 2 interrupt enable 0: Disabled 1: Enabled
Bit 1	C1IEN	0x0	rw	Channel 1 interrupt enable 0: Disabled 1: Enabled
Bit 0	OVFIEN	0x0	rw	Overflow interrupt enable 0: Disabled 1: Enabled

14.4.4.5 TMR15 interrupt status register (TMR15_ISTS)

Bit	Name	Reset value	Type	Description
Bit 15: 11	Reserved	0x0	resd	Kept at default value.
Bit 10	C2RF	0x0	rw0c	Channel 2 recapture flag Please refer to C1RF description.
Bit 9	C1RF	0x0	rw0c	Channel 1 recapture flag This bit indicates whether a recapture is detected when C1IF=1. This bit is set by hardware, and cleared by writing “0”. 0: No capture is detected

				1: Capture is detected.
Bit 8	Reserved	0x0	resd	Kept at default value.
Bit 7	BRKIF	0x0	rw0c	Brake interrupt flag This bit indicates whether the brake input is active or not. It is set by hardware and cleared by writing “0” 0: Inactive level 1: Active level
Bit 6	TRGIF	0x0	rw0c	Trigger interrupt flag This bit is set by hardware on a trigger event. It is cleared by writing “0”. 0: No trigger event occurs 1: Trigger event is generated. Trigger event: an active edge is detected on TRGIN input, or any edge in suspend mode.
Bit 5	HALLIF	0x0	rw0c	HALL interrupt flag This bit is set by hardware on HALL event. It is cleared by writing “0”. 0: No Hall event occurred. 1: Hall event is detected. HALL event: CxEN, CxCEN and CxOCTRL are updated.
Bit 4: 3	Reserved	0x0	resd	Kept at default value.
Bit 2	C2IF	0x0	rw0c	Channel 2 interrupt flag Please refer to C1IF description.
Bit 1	C1IF	0x0	rw0c	Channel 1 interrupt flag If the channel 1 is configured as input mode: This bit is set by hardware on a capture event. It is cleared by software or read access to the TMRx_C1DT 0: No capture event occurred 1: Capture event is generated If the channel 1 is configured as output mode: This bit is set by hardware on a compare event. It is cleared by software. 0: No compare event occurred 1: Compare event is generated
Bit 0	OVFIF	0x0	rw0c	Overflow interrupt flag This bit is set by hardware on an overflow event. It is cleared by software. 0: No overflow event occurred 1: Overflow event is generated. If OVFEN=0 and OVFS=0 in the TMR15_CTRL1 register: − An overflow event is generated when OVFG= 1 in the TMR15_SWEVE register; − An overflow event is generated when the counter CVAL is reinitialized by a trigger event.

14.4.4.6 TMR15 software event register (TMR15_SWEVT)

Bit	Name	Reset value	Type	Description
Bit 15: 8	Reserved	0x0	resd	Kept at default value.
Bit 7	BRKSWTR	0x0	wo	Brake event triggered by software This bit is set by software to generate a brake event. 0: No effect 1: Generate a brake event.
Bit 6	TRGSWTR	0x0	rw	Trigger event triggered by software This bit is set by software to generate a trigger event. 0: No effect 1: Generate a trigger event.
Bit 5	HALLSWTR	0x0	wo	HALL event triggered by software This bit is set by software to generate a HALL event. 0: No effect 1: Generate a HALL event. Note: This bit acts only on channels with complementary output.
Bit 4: 3	Reserved	0x0	resd	Kept at default value.
Bit 2	C2SWTR	0x0	wo	Channel 2 event triggered by software

				Please refer to C1M description
Bit 1	C1SWTR	0x0	wo	Channel 1 event triggered by software This bit is set by software to generate a channel 1 event. 0: No effect 1: Generate a channel 1 event.
Bit 0	OVFSWTR	0x0	wo	Overflow event triggered by software This bit is set by software to generate an overflow event. 0: No effect 1: Generate an overflow event.

14.4.4.7 TMR15 channel mode register1 (TMR15_CM1)

The channel can be used in input (capture mode) or output (compare mode). The direction of a channel is defined by the corresponding CxC bits. All the other bits of this register have different functions in input and output modes. The CxOx describes its function in output mode when the channel is in output mode, while the CxIx describes its function in output mode when the channel is in input mode. Attention must be given to the fact that the same bit can have different functions in input mode and output mode.

Output compare mode:

Bit	Name	Reset value	Type	Description
Bit 15	C2OSEN	0x0	rw	Channel 2 output switch enable
Bit 14: 12	C2OCTRL	0x0	rw	Channel 2 output control
Bit 11	C2OBEN	0x0	rw	Channel 2 output buffer enable
Bit 10	C2OIEN	0x0	rw	Channel 2 output enable immediately
Bit 9: 8	C2C	0x0	rw	Channel 2 configuration This field is used to define the direction of the channel 2 (input or output), and the selection of input pin when C2EN='0': 00: Output 01: Input, C2IN is mapped on C2IFP2 10: Input, C2IN is mapped on C1IFP2 11: Input, C2IN is mapped on STCI. This mode works only when the internal trigger input is selected by STIS register.
Bit 7	C1OSEN	0x0	rw	Channel 1 output switch enable 0: EXT input does not affect C1ORAW 1: When EXT input is high, C1ORAW is cleared.
Bit 6: 4	C1OCTRL	0x0	rw	Channel 1 output control This field defines the behavior of the original signal C1ORAW. 000: Disconnected. C1ORAW is disconnected from C1OUT; 001: C1ORAW is high when TMR15_CVAL=TMR15_C1DT 010: C1ORAW is low when TMR15_CVAL=TMR15_C1DT 011: Switch C1ORAW level when TMR15_CVAL=TMR15_C1DT 100: C1ORAW is forced low 101: C1ORAW is forced high. 110: PWM mode A －OWCDIR=0, C1ORAW is high once TMR15_C1DT>TMR15_CVAL, else low; －OWCDIR=1, C1ORAW is low once TMR15_ C1DT <TMR15_CVAL, else high; 111: PWM mode B － OWCDIR=0, C1ORAW is low once TMR15_ C1DT >TMR15_CVAL, else high; － OWCDIR=1, C1ORAW is high once TMR15_ C1DT <TMR15_CVAL, else low. *Note: In the configurations other than 000', the C1OUT is connected to C1ORAW. The C1OUT output level is not only subject to the changes of C1ORAW, but also the output polarity set by CCTRL.*

Bit 3	C1OBEN	0x0	rw	Channel 1 output buffer enable 0: Buffer function of TMR15_C1DT is disabled. The new value written to the TMR15_C1DT takes effect immediately. 1: Buffer function of TMR15_C1DT is enabled. The value to be written to the TMR15_C1DT is stored in the buffer register, and can be sent to the TMR15_C1DT register only on an overflow event.
Bit 2	C1OIEN	0x0	rw	Channel 1 output enable immediately In PWM mode A or B, this bit is used to accelerate the channel 1 output's response to the trigger event. 0: Need to compare the CVAL with C1DT before generating an output 1: No need to compare the CVAL and C1DT. An output is generated immediately when a trigger event occurs.
Bit 1: 0	C1C	0x0	rw	Channel 1 configuration This field is used to define the direction of the channel 1 (input or output), and the selection of input pin when C1EN='0': 00: Output 01: Input, C1IN is mapped on C1IFP1 10: Input, C1IN is mapped on C2IFP1 11: Input, C1IN is mapped on STCI. This mode works only when the internal trigger input is selected by STIS.

Input capture mode:

Bit	Name	Reset value	Type	Description
Bit 15: 12	C2DF	0x0	rw	Channel 2 digital filter
Bit 11: 10	C2IDIV	0x0	rw	Channel 2 input divider
Bit 9: 8	C2C	0x0	rw	Channel 2 configuration This field is used to define the direction of the channel 2 (input or output), and the selection of input pin when C2EN='0': 00: Output 01: Input, C2IN is mapped on C2IFP2 10: Input, C2IN is mapped on C1IFP2 11: Input, C2IN is mapped on STCI. This mode works only when the internal trigger input is selected by STIS.
Bit 7: 4	C1DF	0x0	rw	Channel 1 digital filter This field defines the digital filter of the channel 1. N stands for the number of filtering, indicating that the input edge can pass the filter only after N sampling events. 0000: No filter, sampling is done at f_{DTS} 1000: $f_{SAMPLING}=f_{DTS}/8$, N=6 0001: $f_{SAMPLING}=f_{CK\_INT}$, N=2 1001: $f_{SAMPLING}=f_{DTS}/8$, N=8 0010: $f_{SAMPLING}=f_{CK\_INT}$, N=4 1010: $f_{SAMPLING}=f_{DTS}/16$, N=5 0011: $f_{SAMPLING}=f_{CK\_INT}$, N=8 1011: $f_{SAMPLING}=f_{DTS}/16$, N=6 0100: $f_{SAMPLING}=f_{DTS}/2$, N=6 1100: $f_{SAMPLING}=f_{DTS}/16$, N=8 0101: $f_{SAMPLING}=f_{DTS}/2$, N=8 1101: $f_{SAMPLING}=f_{DTS}/32$, N=5 0110: $f_{SMPLING}=f_{DTS}/4$, N=6 1110: $f_{SAMPLING}=f_{DTS}/32$, N=6 0111: $f_{SAMPLING}=f_{DTS}/4$, N=8 1111: $f_{SAMPLING}=f_{DTS}/32$, N=8
Bit 3: 2	C1IDIV	0x0	rw	Channel 1 input divider This field defines Channel 1 input divider. 00: No divider. An input capture is generated at each active edge. 01: An input compare is generated every 2 active edges 10: An input compare is generated every 4 active edges

				11: An input compare is generated every 8 active edges Note: the divider is reset once C1EN='0'
Bit 1: 0	C1C	0x0	rw	Channel 1 configuration This field is used to define the direction of the channel 1 (input or output), and the selection of input pin when C1EN='0': 00: Output 01: Input, C1IN is mapped on C1IFP1 10: Input, C1IN is mapped on C2IFP1 11: Input, C1IN is mapped on STCI. This mode works only when the internal trigger input is selected by STIS.

14.4.4.8 TMR15 channel control register (TMR15_CCTRL)

Bit	Name	Reset value	Type	Description
Bit 15: 8	Reserved	0x0	resd	Kept at default value.
Bit 7	C2CP	0x0	rw	Channel 2 complementary polarity Refer to C1CP descriptions.
Bit 6	Reserved	0x0	resd	Kept at default value.
Bit 5	C2P	0x0	rw	Channel 2 polarity Refer to C1P descriptions.
Bit 4	C2EN	0x0	rw	Channel 2 enable Refer to C1EN descriptions.
Bit 3	C1CP	0x0	rw	Channel 1 complementary polarity 0: C1COUT is active high. 1: C1COUT is active low.
Bit 2	C1CEN	0x0	rw	Channel 1 complementary enable 0: Output is disabled. 1: Output is enabled.
Bit 1	C1P	0x0	rw	Channel 1 polarity When the channel 1 is configured in output mode: 0: C1OUT is active high 1: C1OUT is active low When the channel 1 is configured in input mode: The active edge of the input signal is defined by C1CP/C1P. 00: C1IN active edge is on its rising edge. When used as external trigger, C1IN is not inverted. 01: C1IN active edge is on its falling edge. When used as external trigger, C1IN is inverted. 10: Reserved 11: C1IN active edge is on its falling edge and rising edge. When used as external trigger, C1IN is not inverted.
Bit 0	C1EN	0x0	rw	Channel 1 enable 0: Input or output is disabled 1: Input or output is enabled

Table 14-11 Complementary output channel CxOUT and CxCOUT control bits with brake function

Control bit					Output state (1)	
OEN bit	FCSODIS bit	FCSOEN bit	CxEN bit	CxCEN bit	CxOUT output state	CxCOUT output state
1	X	0	0	0	Output disabled (no driven by the timer) CxOUT=0, Cx_EN=0	Output disabled (no driven by the timer) CxCOUT=0, CxCEN=0
		0	0	1	Output disabled (no driven by the timer) CxOUT=0, Cx_EN=0	CxORAW + polarity, CxCOUT= CxORAW xor CxCP, CxCEN=1
		0	1	0	CxORAW+ polarity CxOUT= CxORAW xor CxP, Cx_EN=1	Output disabled (no driven by the timer) CxCOUT=0, CxCEN=0
		0	1	1	CxORAW+polarity+ dead-time, Cx_EN=1	CxORAW inverted+polarity+dead-time, CxCEN=1
		1	0	0	Output disabled (no driven by the timer) CxOUT=CxP, Cx_EN=0	Output disabled (no driven by the timer) CxCOUT=CxCP, CxCEN=0
		1	0	1	Off-state (Output enabled with inactive level) CxOUT=CxP, Cx_EN=1	CxORAW + polarity, CxCOUT= CxORAW xor CxCP, CxCEN=1
		1	1	0	CxORAW + polarity, CxOUT= CxORAW xor CxP, Cx_EN=1	Off-state (Output enabled with inactive level) CxCOUT=CxCP, CxCEN=1
		1	1	1	CxORAW+ polarity+dead-time, Cx_EN=1	CxORAW inverted+polarity+dead-time, CxCEN=1
0	0	X	0	0	Output disabled (corresponding IO is not driven by the timer, IO floating)	
	0		0	1	Asynchronously: CxOUT=CxP, Cx_EN=0, CxCOUT=CxCP, CxCEN=0;	
	0		1	0	If the clock is present: after a dead-time, CxOUT=CxIOS, CxCOUT=CxCIOS, assuming that CxIOS and CxCIOS do not correspond to CxOUT and CxCOUT active level.	
	0		1	1		
	1		0	0	CxEN=CxCEN=0: output disabled (corresponding IO is not driven by the timer, IO floating)	
					In other cases: off-state (corresponding channel output invalid level)	
					Asynchronously: CxOUT =CxP, Cx_EN=1, CxCOUT=CxCP, CxCEN=1;	
					If the clock is present: after a dead-time, CxOUT=CxIOS, CxCOUT=CxCIOS, assuming that CxIOS and CxCIOS do not correspond to CxOUT and CxCOUT active level.	

Note: If the two outputs of a channel are not used (CxEN = CxCEN = 0), CxIOS, CxCIOS, CxP and CxCP must be cleared.

Note: The state of the external I/O pins connected to the complementary CxOUT and CxCOUT channels depends on the CxOUT and CxCOUT channel state and the GPIO and the IOMUX registers.

14.4.4.9 TMR15 counter value (TMR15_CVAL)

Bit	Name	Reset value	Type	Description
Bit 15: 0	CVAL	0x0	rw	Counter value

14.4.4.10 TMR15 divider (TMR15_DIV)

Bit	Name	Reset value	Type	Description
Bit 15: 0	DIV	0x0	rw	Divider value The counter clock frequency fCK_CNT = fTMR_CLK /(DIV[15:0]+1). The value of this register is transferred to the actual prescaler register when an overflow event occurs.

14.4.4.11 TMR15 period register (TMR15_PR)

Bit	Name	Reset value	Type	Description
Bit 15: 0	PR	0x0	rw	Period value This defines the period value of the TMRx counter. The timer stops working when the period value is 0.

14.4.4.12 TMR15 repetition period register (TMR15_RPR)

Bit	Name	Reset value	Type	Description
Bit 15: 8	Reserved	0x0	resd	Kept at default value
Bit 7: 0	RPR	0x00	rw	Repetition of period value This field is used to reduce the generation rate of overflow events. An overflow event is generated when the repetition counter reaches 0.

14.4.4.13 TMR15 channel 1 data register (TMR15_C1DT)

Bit	Name	Reset value	Type	Description
Bit 15: 0	C1DT	0x0	rw	Channel 1 data register When the channel 1 is configured as input mode: The C1DT is the CVAL value stored by the last channel 1 input event (C1IN) When the channel 1 is configured as output mode: C1DT is the value to be compared with the CVAL value. Whether the written value takes effective immediately depends on the C1OBEN bit, and the corresponding output is generated on C1OUT as configured.

14.4.4.14 TMR15 channel 2 data register (TMR15_C2DT)

Bit	Name	Reset value	Type	Description
Bit 15: 0	C2DT	0x0	rw	Channel 2 data register When the channel 2 is configured as input mode: The C2DT is the CVAL value stored by the last channel 2 input event (C1IN) When the channel 2 is configured as output mode: C2DT is the value to be compared with the CVAL value. Whether the written value takes effective immediately depends on the C2OBEN bit, and the corresponding output is generated on C2OUT as configured.

14.4.4.15 TMR15 brake register (TMR15_BRK)

Bit	Name	Reset value	Type	Description
Bit 31: 18	Reserved	0x0	resd	Kept at its default value.
Bit 19: 16	BKF	0x0	rw	Brake input filter This field is used to set the filter for brake input. The filter number N indicates that the input edge can pass through filter only after N sampling events. 0000: $f_{SAMPLING}=f_{DTS}$ (no filter) 1000: $f_{SAMPLING}=f_{DTS}$ /8, N=6

				0001: $f_{SAMPLING}=f_{CK\_INT}$, N=2 1001: $f_{SAMPLING}=f_{DTS}/8$, N=8 0010: $f_{SAMPLING}=f_{CK\_INT}$, N=4 1010: $f_{SAMPLING}=f_{DTS}/16$, N=5 0011: $f_{SAMPLING}=f_{CK\_INT}$, N=8 1011: $f_{SAMPLING}=f_{DTS}/16$, N=6 0100: $f\_{SAMPLING}=f_{DTS}/2$, N=6 1100: $f_{SAMPLING}=f_{DTS}/16$, N=8 0101: $f_{SAMPLING}=f_{DTS}/2$, N=8 1101: $f_{SAMPLING}=f_{DTS}/32$, N=5 0110: $f_{SMPLING}=f_{DTS}/4$, N=6 1110: $f_{SAMPLING}=f_{DTS}/32$, N=6 0111: $f_{SAMPLING}=f_{DTS}/4$, N=8 1111: $f_{SAMPLING}=f_{DTS}/32$, N=8
Bit 15	OEN	0x0	rw	Output enable This bit acts on the channels as output. It is used to enable CxOUT and CxCOUT outputs. 0: Disabled 1: Enabled
Bit 14	AOEN	0x0	rw	Automatic output enable OEN is set automatically at an overflow event. 0: Disabled 1: Enabled
Bit 13	BRKV	0x0	rw	Brake input validity This bit is used to select the active level of a brake input. 0: Brake input is active low. 1: Brake input is active high.
Bit 12	BRKEN	0x0	rw	Brake enable This bit is used to enable brake input. 0: Brake input is disabled. 1: Brake input is enabled.
Bit 11	FCSOEN	0x0	rw	Frozen channel status when holistic output enable This bit acts on the channels that have complementary output. It is used to set the channel state when the timer is inactive and OEN=1. 0: CxOUT/CxCOUT outputs are disabled. 1: CxOUT/CxCOUT outputs are enabled. Output inactive level.
Bit 10	FCSODIS	0x0	rw	Frozen channel status when holistic output disable This bit acts on the channels that have complementary output. It is used to set the channel state when the timer is inactive and OEN=0. 0: CxOUT/CxCOUT outputs are disabled. 1: CxOUT/CxCOUT outputs are enabled. Output idle level.
Bit 9: 8	WPC	0x0	rw	Write protection configuration This field is used to enable write protection. 00: Write protection is OFF. 01: Write protection level 3, and the following bits are write protected: TMR1_STOP: DTC, STPEN, STPV and HOAEN TMRx_CTRL2: CxIOS and CxCIOS 10: Write protection level 2. The following bits and all bits in level 3 are write protected: TMR1_CCTRL: CxP and CxCP TMR1_STOP: FCSODIS and FCSOEN 11: Write protection level 1. The following bits and all bits in level 2 are write protected: TMR1_CMx: C2OCTRL and C2OBEN Note: Once WPC>0, its content remains frozen until the next system reset.
Bit 7: 0	DTC	0x00	rw	Dead-time configuration This field defines the duration of the dead-time insertion. The 3-bit MSB of DTC[7: 0] is used for function selection:

				0xx: DT = DTC [7: 0] * TDTS 10x: DT = (64+ DTC [5: 0]) * TDTS * 2 110: DT = (32+ DTC [4: 0]) * TDTS * 8 111: DT = (32+ DTC [4: 0]) * TDTS * 16

Note: Based on lock configuration, AOEN, BRKV, BRKEN, FCSODIS, FCSOEN and DTC[7:0] can be write protected. Thus it is necessary to configure write protection when writing to the TMR15_BRK register for the first time.

14.4.4.16 TMR15 DMA control register (TMR15_DMACTRL)

Bit	Name	Reset value	Type	Description
Bit 15:13	Reserved	0x0	resd	Kept at default value.
Bit 12:8	DTB	0x00	rw	DMA transfer bytes This field defines the number of DMA transfers: 00000: 1 byte 00001: 2 bytes 00010: 3 bytes 00011: 4 bytes .... 10000: 17 bytes 10001: 18 bytes
Bit 7:5	Reserved	0x0	resd	Kept at default value.
Bit 4: 0	ADDR	0x00	rw	DMA transfer address offset ADDR is defined as an offset starting from the address of the TMR15_CTRL1 register: 00000: TMR15_CTRL1 00001: TMR15_CTRL2 00010: TMR15_STCTRL .....

14.4.4.17 TMR15 DMA data register (TMR15_DMADT)

Bit	Name	Reset value	Type	Description
Bit 15: 0	DMADT	0x0	rw	DMA data register A write/read operation to the DMADT register accesses any TMR register located at the following address: TMR15 peripheral address + ADDR*4 to TMR15 peripheral address + ADDR*4 + DTB*4

14.5 General-purpose timers (TMR16 and TMR17)

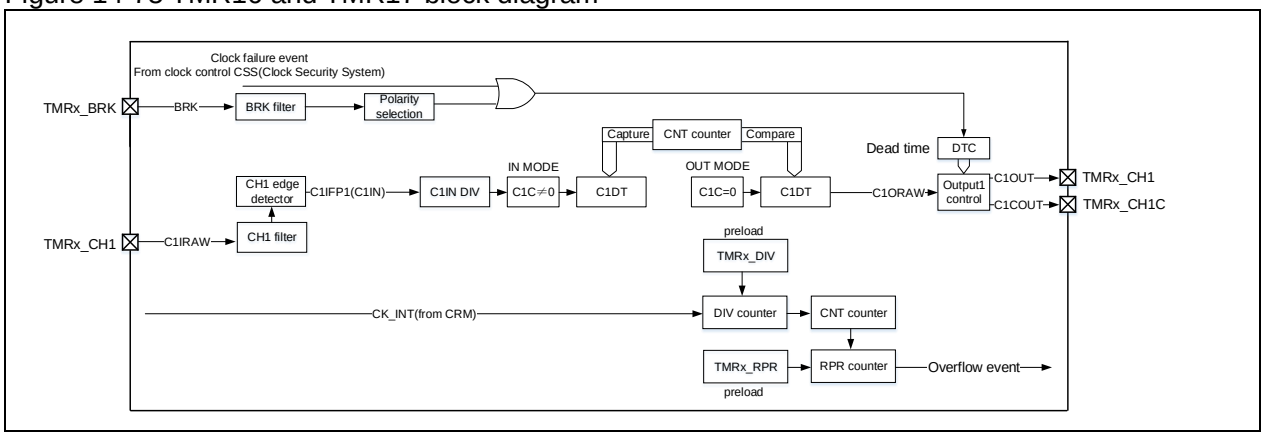
14.5.1 TMR16 and TMR17 introduction

The general-purpose timers (TMR16 and TMR17) consist of a 16-bit upcounter, one capture/compare register, and one independent channel. They can be used for dead-time insertion, input capture and programmable PWM output.

14.5.2 TMR16 and TMR17 main features

- Clock source of counter clock: internal clock, external input and internal trigger inputs
- 16-bit upcounter, and 8-bit repetition counter
- One independent channel for input capture, output compare, PWM generation, one-pulse mode output and dead-time insertion
- One independent channel for complementary output
- TMR brake function
- Synchronization control between master and slave timers
- Interrupt/DMA generation at overflow event, brake signal input and channel events
- Support TMR burst DMA transfer

Figure 14-75 TMR16 and TMR17 block diagram

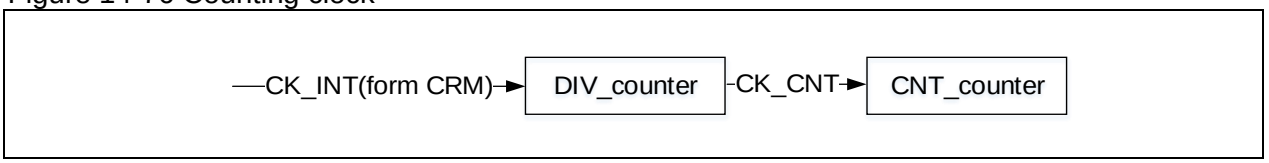


14.5.3 TMR16 and TMR17 functional overview

14.5.3.1 Counting clock

The counting clock of TMR16 and TMR17 can only be from the internal clock (CK_INT).

Figure 14-76 Counting clock

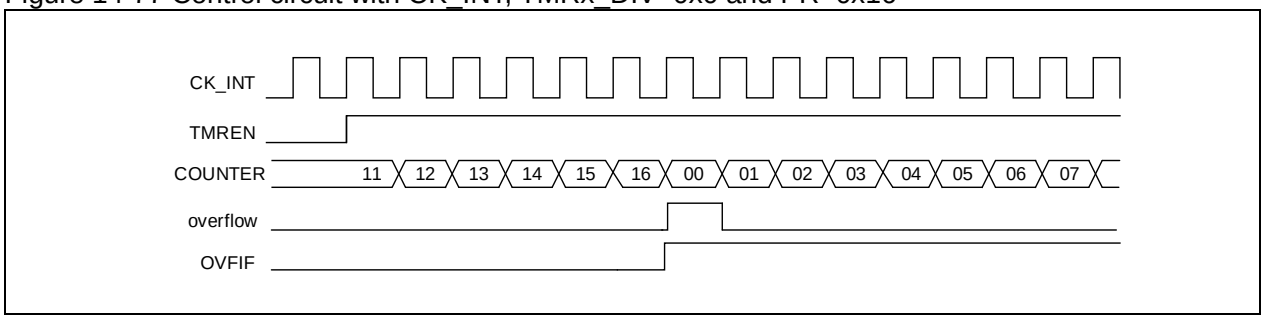


Internal clock (CK_INT)

By default, the CK_INT, which is divided by the prescaler, is used to drive the counter to start counting. When TMR's APB clock prescaler factor is 1, the CK_INT frequency is equal to that of APB; otherwise, it doubles the APB clock frequency.

- Set counting frequency through TMRx_DIV register
- Set counting cycles through TMRx_PR register
- Enable a counter by setting the TMREN bit in the TMRx_CTRL1 register

Figure 14-77 Control circuit with CK_INT, TMRx_DIV=0x0 and PR=0x16



14.5.3.2 Counting mode

The TMR16 and TMR17 supports multiple counting modes to meet various application scenarios. Each of them has a 16-bit upcounter.

The TMRx_PR register is used to define counting period of counter. The value in the TMRx_PR is immediately moved to the shadow register by default. When the periodic buffer is enabled (PRBEN=1), the value in the TMRx_PR register is transferred to the shadow register only at an overflow event.

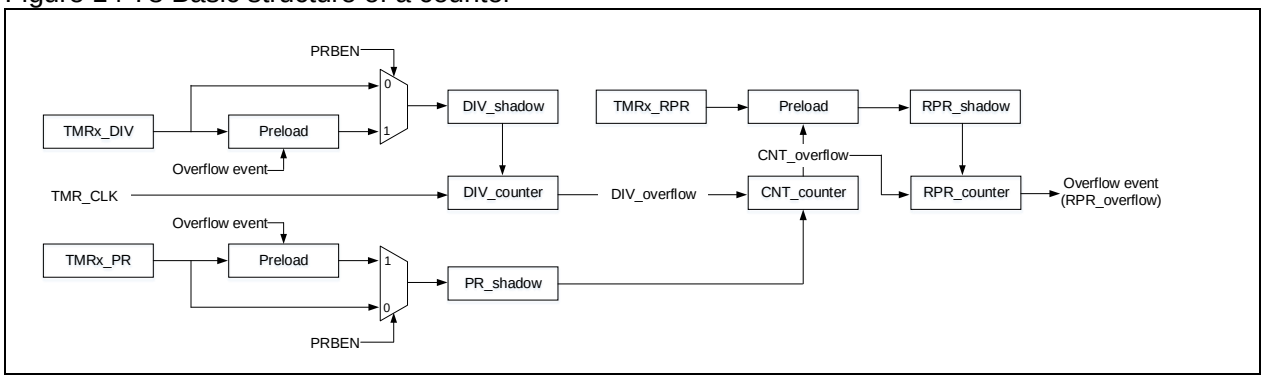
TMRx_DIV register is used to define the counter frequency of the counter. The counter counts once every DIV[15:0]+1 clock cycle. Similar to TMRx_PR register, after a periodic buffer is enabled, the value of the TMRx_DIV register are transferred into the shadow register upon an overflow event.

Reading the TMRx_CNT register returns the current counter value. Writing the TMRx_CNT register will update the current counter value.

An overflow event is enabled by default. It can be disabled by setting OVFEN=1 in the TMRx_CTRL1 register. The OVFS bit in the TMRx_CTRL1 register is used to select the source of an overflow event, which is, by default, counter overflow or underflow, setting OVFSWTR, reset signal generated by slave mode timer controller in reset mode. Once the OVFS is set, an overflow event is generated only when overflow or underflow occurs.

Setting the TMREN bit (TMREN=1) enables the timer to start counting. Base on synchronization logic, however, the actual enable signal TMR_EN is set 1 clock cycle after the TMREN is set.

Figure 14-78 Basic structure of a counter



Upcounting mode

This mode is enabled by setting CMSEL[1:0]=2'b00 and OWCDIR=1'b0 in the TMRx_CTRL1 register.

In upcounting mode, the counter counts from 0 to the value programmed in the TMRx_PR register, restarts from 0, and generates a counter overflow event, with setting OVFIF bit to 1. If the overflow event is disabled, the counter is no longer reloaded with the prescaler and period value on counter overflow, otherwise, the prescaler and period value will be updated on an overflow event.

Figure 14-79 Overflow event when PRBEN=0

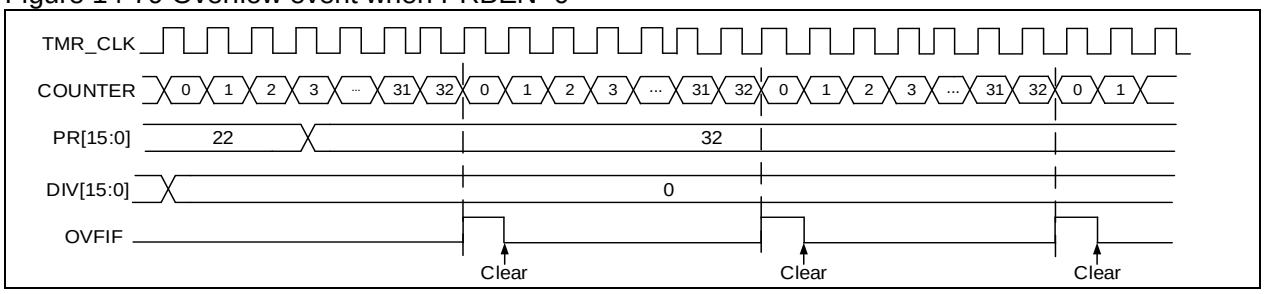
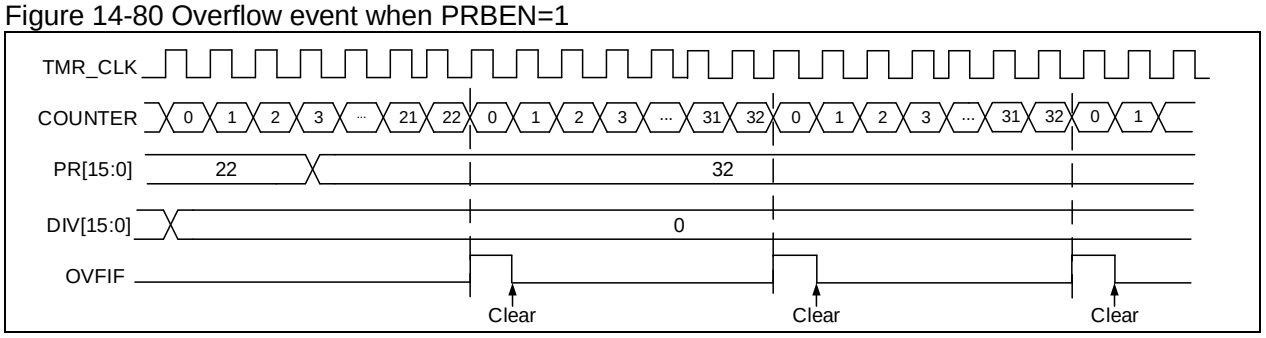


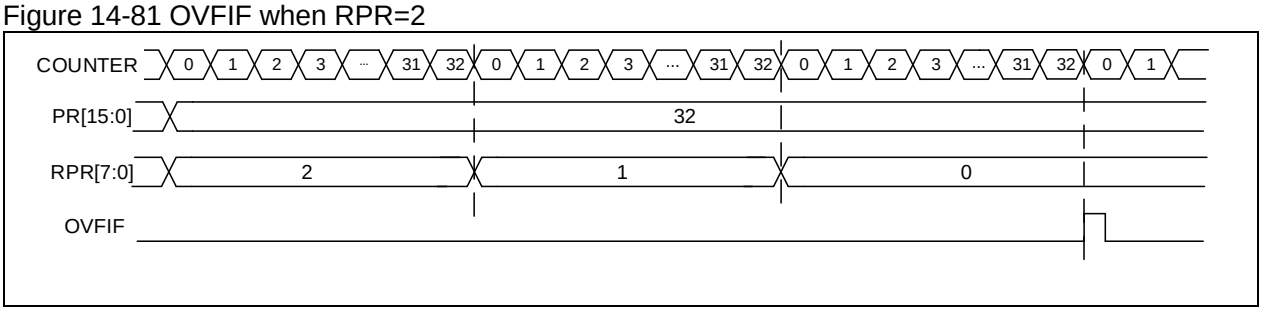
Figure 14-80 Overflow event when PRBEN=1



Repetition counter mode:

The TMRx_RPR register is used to enable repetition counting mode. This mode is enabled when the repetition counter value is not equal to 0. In this mode, an overflow event is generated when a counter overflow occurs (RPR[7:0]+1). The repetition counter is decremented at each counter overflow. An overflow event is generated when the repetition counter reaches 0. The frequency of the overflow event generation can be adjusted by setting the repetition counter value.

Figure 14-81 OVFIF when RPR=2



14.5.3.3 TMR input function

Each of TMR16 and TMR17 has an independent channel that can be configured in input or output mode. As input, it can be used for filtering, selection and division of input signals, as well as for input capture.

Figure 14-82 Input/output channel 1 main circuit

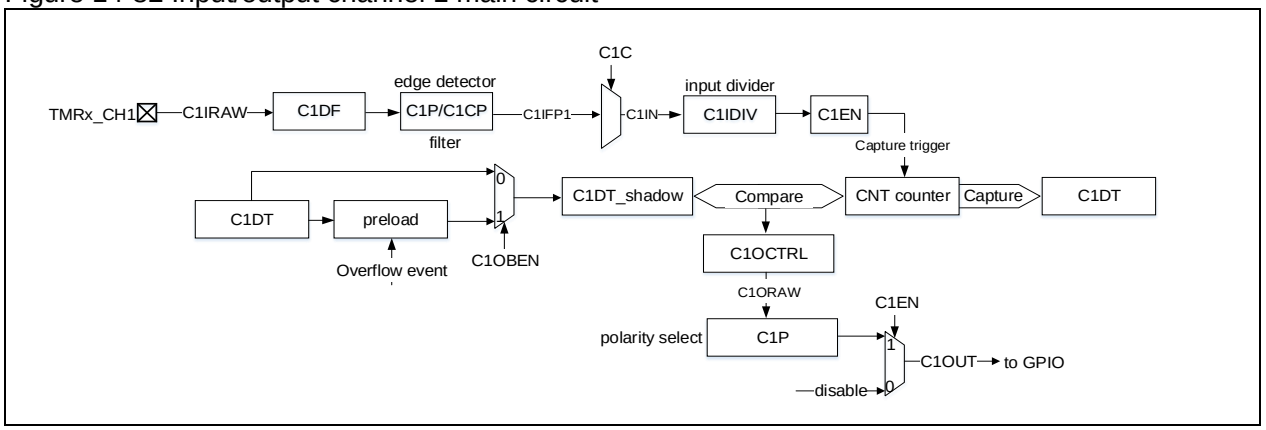
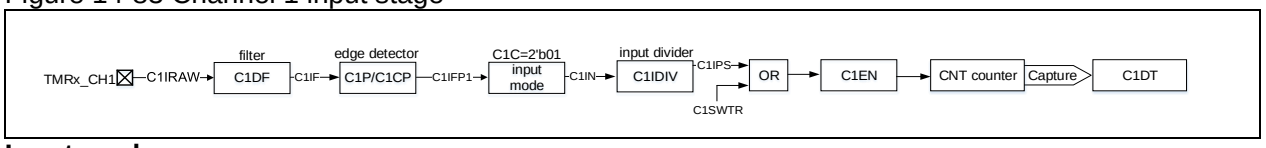


Figure 14-83 Channel 1 input stage



Input mode

In input mode, the TMRx_CxDT registers latches the current counter values after the selected trigger signal is detected, and the capture compare interrupt flag bit (CxIF) is set to 1. An interrupt/DMA request will be generated if the CxIEN bit and CxDEN bit are enabled. If the selected trigger signal is detected when the CxIF is set to 1, a capture overflow event is generated, the previous counter value will be overwritten with the current counter value, and the CxRF is set to 1.

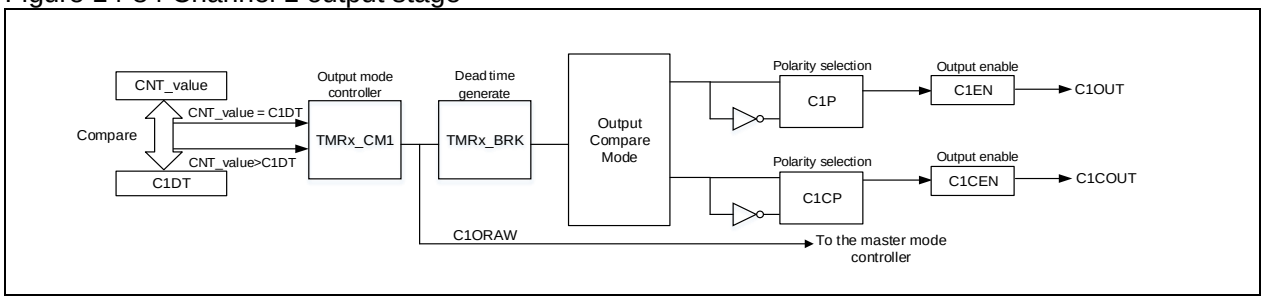
To capture the rising edge of C1IN input, following the procedure below:

- Set C1C=01 in the TMRx_CM1 register to select C1IN (channel 1 input)
- Set C1IN signal filter bandwidth (CxDF[3: 0])
- Set the active edge of C1IN channel by writing C1P=0 (rising edge) in the TMRx_CCTRL register
- Program C1IN signal capture frequency divider (C1DIV[1: 0])
- Enable channel 1 input capture (C1EN=1)
- If needed, enable the relevant interrupt or DMA request by setting the C1IEN bit in the TMRx_IDEN register or the C1DEN bit in the TMRx_IDEN register

14.5.3.4 TMR output function

The TMR output consists of a comparator and an output controller. It is used to program the period, duty cycle and polarity of output signals.

Figure 14-84 Channel 1 output stage



Output mode

CxC[1: 0]≠2'b00 is used to configure the channel as output mode. In this case, the counter value is compared with that in the CxDT register, and the intermediate signal CxORAW is generated according to the output mode selected by CxOCTRL[2: 0], and this signal is sent to IO after being processed by the output control circuit. The period of the output signal is configured by the TMRx_PR register, while the duty cycle is configured by the CxDT register.

Output compare modes include:

PWM mode A:

Enable PWM mode A by setting CxOCTRL=3'b110. In upcounting mode, C1ORAW outputs high when TMRx_C1DT>TMRx_CVAL, otherwise, it is low; in downcounting mode, C1ORAW outputs low when TMRx_C1DT<TMRx_CVAL, otherwise, it is high.

To use PWM mode A, the following procedures are recommended:

- Set PWM periods through TMRx_PR register
- Set PWM duty cycles through TMRx_CxDT register
- Select PWM mode A by setting CxOCTRL=3'b110 in the TMRx_CM1/CM2 register
- Set counting frequency through TMRx_DIV register
- Select counting mode by setting the TWCMSEL[1:0] bit in the TMRx_CTRL1 register
- Select output polarity through the CxP and CxCP bits in the TMRx_CCTRL register

- Enable channel output through the CxEN and CxCEN bits in the TMRx_CCTRL register
- Enable TMRx output through the OEN bit in the TMRx_BRK register
- Configure GPIOs corresponding to TMR output channels as multiplexed mode
- Enable TMRx to start counting through the TMREN bit in the TMRx_CTRL1 register.

PWM mode B:

Enable PWM mode B by setting CxOCTRL=3'b111. In upcounting mode, C1ORAW outputs low when TMRx_C1DT>TMRx_CVAL, otherwise, it is high; in downcounting mode, C1ORAW outputs high when TMRx_C1DT<TMRx_CVAL, otherwise, it is low.

Forced output mode:

Enable forced output mode by setting CxOCTRL=3'b100/101. In this case, the CxORAW is forced to be the programmed level, regardless of the counter value. Despite this, the channel flag bit and DMA request still depend on the compare result.

Output compare mode:

Enable output compare mode by setting CxOCTRL=3'b001/010/011. In this case, when the counter value matches the value of the CxDT register, the CxORAW is forced high (CxOCTRL=3'b001), low (CxOCTRL=3'b010) or toggling (CxOCTRL=3'b011).

One-pulse mode:

This is a particular case of PWM mode. Enable one-pulse by setting OCMEN=1. In this mode, the comparison match is performed in the current counting period. The TMREN bit is cleared as soon as the current counting is completed. Therefore, only one pulse is output. When in upcounting mode, the configuration must follow the rule: CVAL<CxDT≤PR; in downcounting mode, CVAL>CxDT is required.

Fast output mode:

Enable this mode by setting CxOIEN=1. If enabled, the CxORAW signal will not change when the counter value matches the CxDT, but change at the beginning of the current counting period. In other words, the comparison result is advanced, so the comparison result between the counter value and the TMRx_CxDT register will determine the level of CxORAW in advance.

Figure 14-85 gives an example of output compare mode (toggle) with C1DT=0x3. When the counter value is equal to 0x3, C1OUT toggles.

Figure 14-86 gives an example of the combination between upcounting mode and PWM mode A. The output signal behaves when PR=0x32 but CxDT is configured with a different value.

Figure 14-87 gives an example of the combination between upcounting mode and one-pulse PWM mode B. The counter only counts only one cycle, and the output signal sends only one pulse.

Figure 14-85 C1ORAW toggles when counter value matches the C1DT value

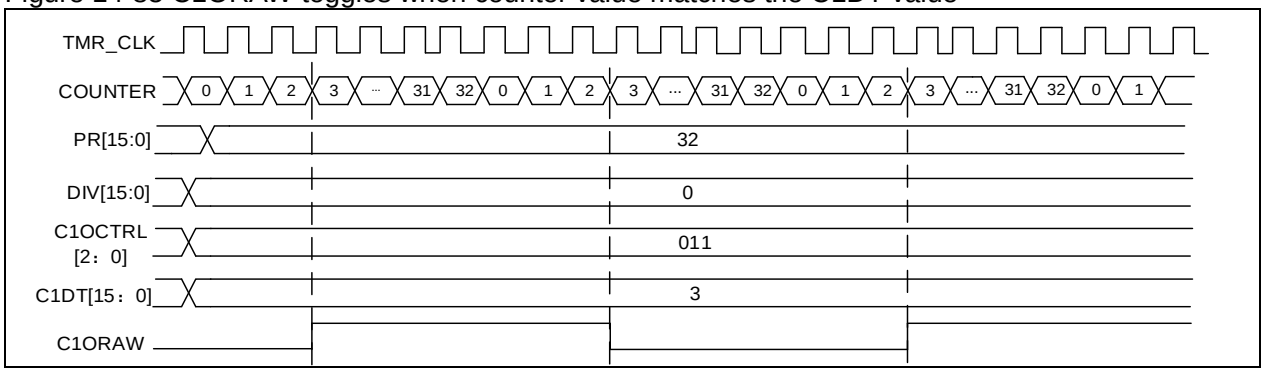


Figure 14-86 Upcounting mode and PWM mode A

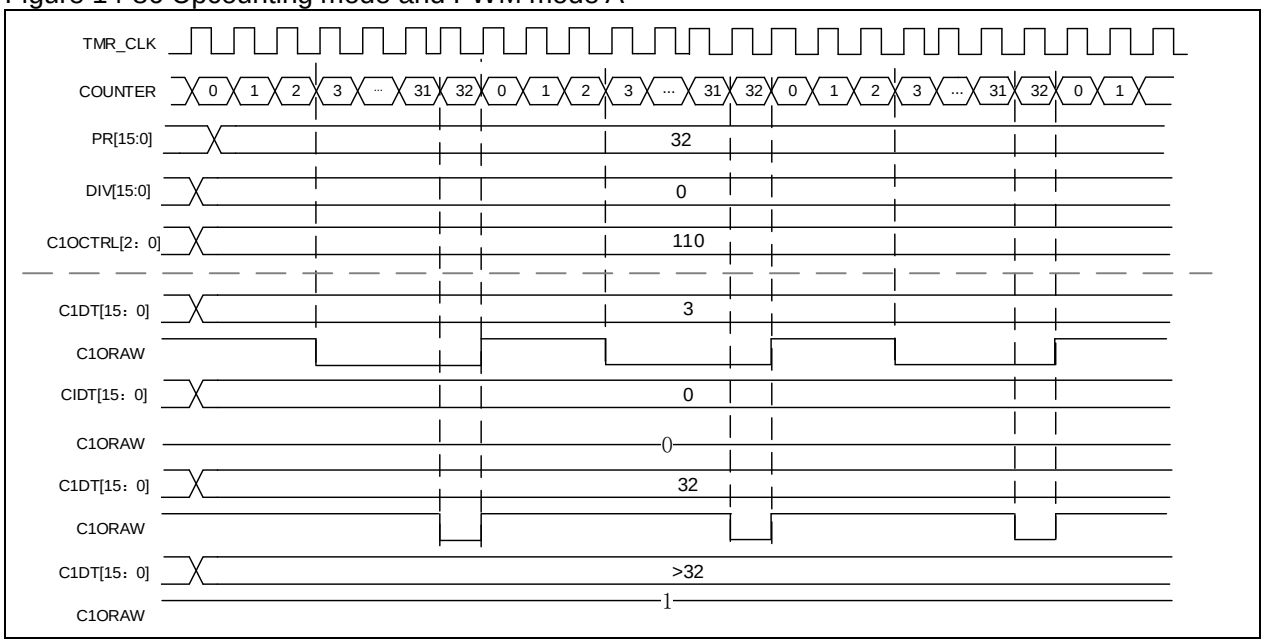
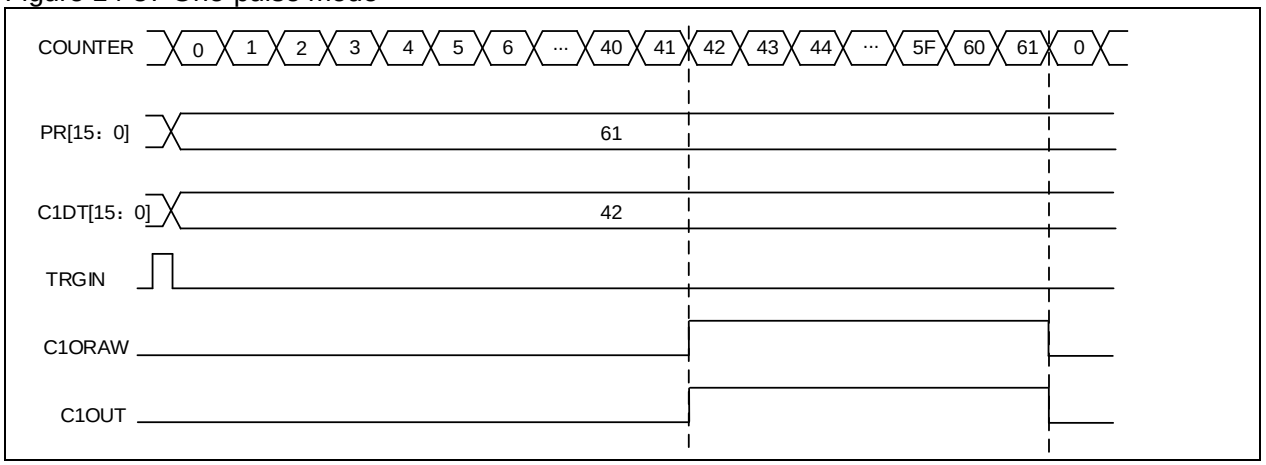


Figure 14-87 One-pulse mode



Dead-time insertion

The TMR16 and TMR17 contain a set of reverse channel output. This function is enabled by the CxCEN bit and its polarity is selected by CxCP. Refer to Table 14-17 for more information about the output state of CxOUT and CxCOUT.

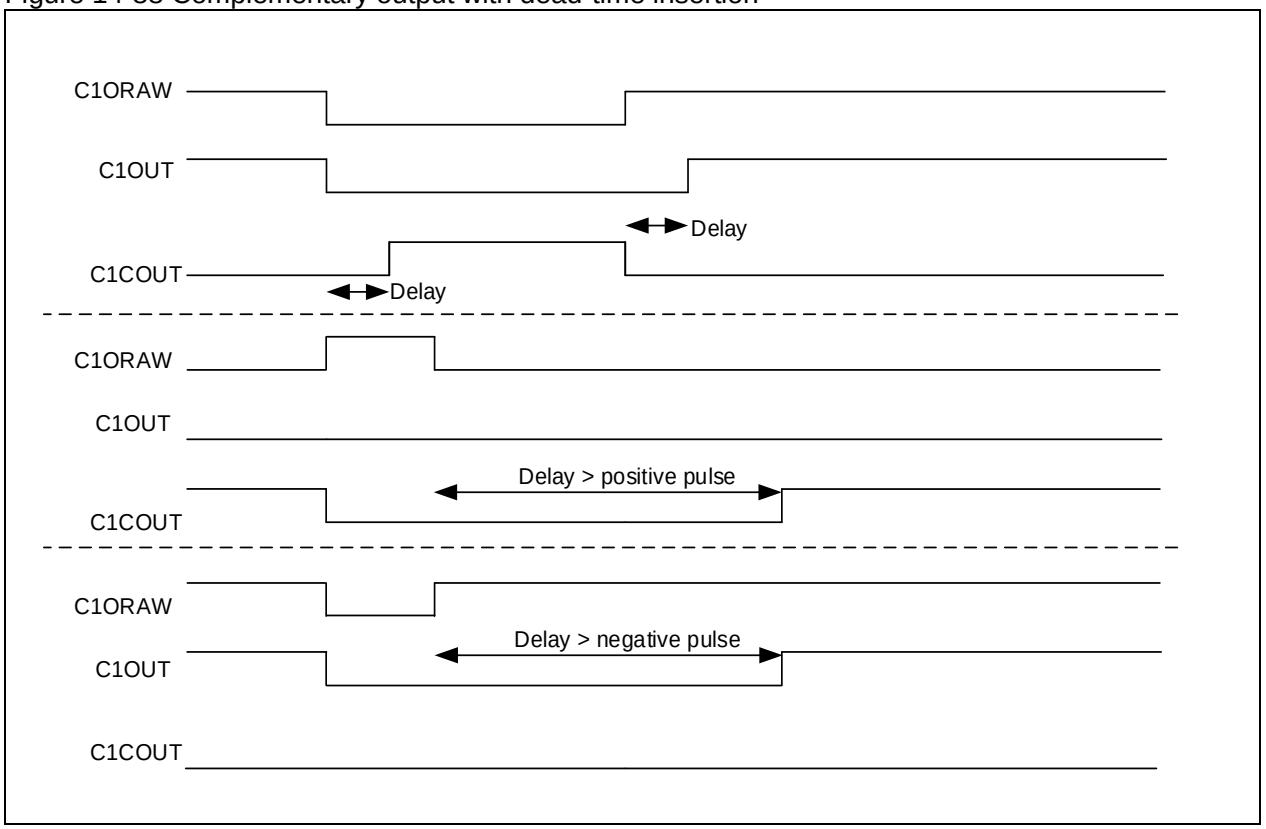
The dead-time is activated when switching to IDLEF state (OEN falling down to 0).

Set both CxEN and CxCEN bits to 1, and use DTC[7:0] bit to insert dead-time of different durations. After the dead-time insertion, the rising edge of the CxOUT is delayed compared to the rising edge of the reference signal; the rising edge of the CxCOU is delayed compared to the falling edge of the reference signal.

If the delay is greater than the width of the active output, C1OUT and C1COUT will not generate corresponding pulses. Therefore, the dead-time should be less than the width of the active output.

Figure 14-88 gives an example of dead-time insertion when CxP=0, CxCP=0, OEN=1, CxEN=1 and CxCEN=1.

Figure 14-88 Complementary output with dead-time insertion



14.5.3.5 TMR brake function

When the brake function is enabled (BRKEN=1), the CxOUT and CxCOUT are jointly controlled by OEN, FCSODIS, FCSOEN, CxIOS and CxCIOS. But, CxOUT and CxCOUT cannot always be set to active levels at the same time. Please refer to Table 14-13 for more details.

The brake source can be from a brake input pin or a clock failure event. The polarity is controlled by the BRKV bit.

When a brake event occurs, there are the following actions:

- The OEN bit is cleared asynchronously, and the channel output state is selected by setting the FCSODIS bit. This function works even if the MCU oscillator is off.
- After OEN is cleared, the channel output level is defined by the CxIOS bit. If FCSODIS=0, the timer output is disabled, otherwise, the output enable remains high.
- When complementary output mode is used:
  - The output is first put in reset state, that is, inactive state (depending on the polarity). This is done asynchronously so that it works even if no clock is provided to the timer.
  - If the timer clock is still active, then the dead-time generator is activated. The CxIOS and CxCIOS bits are used to program the level after dead-time. Even in this case, the CxIOS and CxCIOS cannot be driven to their active level at the same time.

Note: Because of synchronization logic on OEN bit, the dead-time duration is usually longer than usual (around 2 clk_tmr clock cycles)

  - If FCSODIS=0, the timer releases the enable output, otherwise, it keeps the enable output; the enable output becomes high as soon as one of the CxEN and CxCEN bits becomes high.
- If the brake interrupt or DMA request is enabled, the brake statue flag is set, and a brake interrupt or DMA request can be generated.
- If AOEN=1, the OEN bit is automatically set to 1 at the next overflow event.

Note: When brake input is active, OEN cannot be set, nor the status flag BRKIF can be cleared.

Figure 14-89 Example of TMR output control

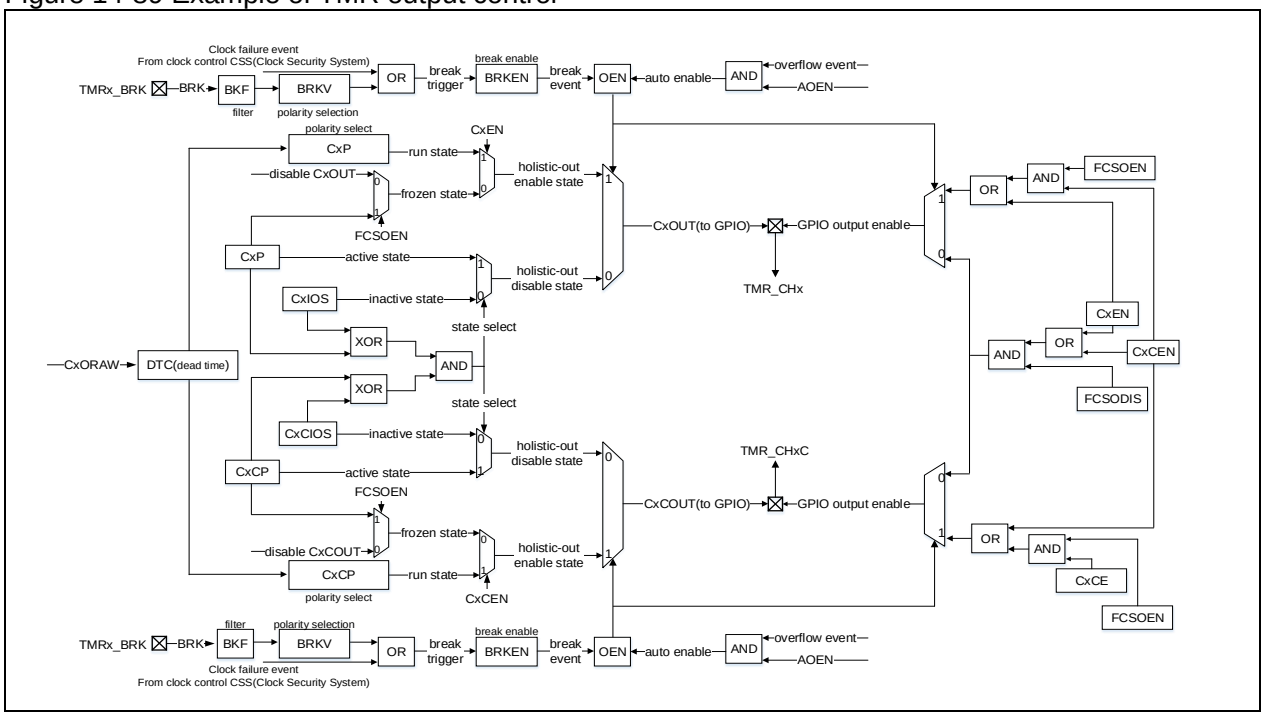
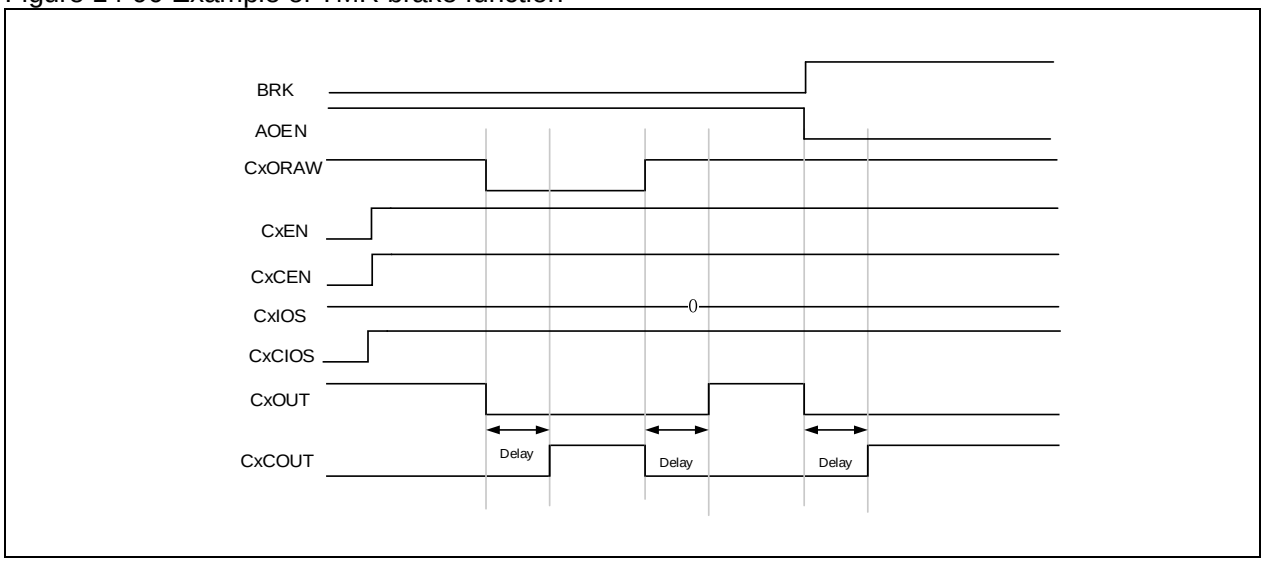


Figure 14-90 Example of TMR brake function



14.5.3.6 Debug mode

When the microcontroller enters debug mode (Cortex®-M0+ core halted), the TMRx counter stops counting by setting the TMRx_PAUSE in the DEBUG module.

14.5.4 TMR16 and TMR17 registers

These peripheral registers must be accessed by half-words (16 bits) or words (32 bits).

TMR16 and TMR17 registers are mapped into a 16-bit addressable space.

Table 14-12 TMR16 and TMR17 register map and reset value

Register	Offset	Reset value
TMRx_CTRL1	0x00	0x0000
TMRx_CTRL2	0x04	0x0000
TMRx_IDEN	0x0C	0x0000
TMRx_ISTS	0x10	0x0000
TMRx_SWEVT	0x14	0x0000
TMRx_CM1	0x18	0x0000
TMRx_CCTRL	0x20	0x0000
TMRx_CVAL	0x24	0x0000
TMRx_DIV	0x28	0x0000
TMRx_PR	0x2C	0x0000
TMRx_RPR	0x30	0x0000
TMRx_C1DT	0x34	0x0000
TMRx_BRK	0x44	0x0000
TMRx_DMACTRL	0x48	0x0000
TMRx_DMADT	0x4C	0x0000

14.5.4.1 TMR16 and TMR17 control register 1 (TMRx_CTRL1)

Bit	Name	Reset value	Type	Description
Bit 15: 10	Reserved	0x0	resd	Kept at default value
Bit 9: 8	CLKDIV	0x0	rw	Clock divider This field is used to define the division ratio between digital filter sampling frequency (f_{DTS}) and timer clock frequency (f_{CK_INT}). it is also used to set the division ratio between dead time base (T_{DTS}) and timer clock period (T_{CK_INT}). 00: No division, $f_{DTS}=f_{CK\_INT}$ 01: Divided by 2, $f_{DTS}=f_{CK\_INT}/2$ 10: Divided by 4, $f_{DTS}=f_{CK\_INT}/4$ 11: Reserved
Bit 7	PRBEN	0x0	rw	Period buffer enable 0: Period buffer is disabled 1: Period buffer is enabled
Bit 6: 4	Reserved	0x0	resd	Kept at default value
Bit 3	OCMEN	0x0	rw	One cycle mode enable This bit is use to select whether to stop counting at an overflow event 0: The counter does not stop at an overflow event 1: The counter stops at an overflow event
Bit 2	OVFS	0x0	rw	Overflow event source This bit is used to select overflow event or DMA request sources. 0: Counter overflow, setting the OVFSWTR bit or overflow event generated by slave timer controller 1: Only counter overflow generates an overflow event
Bit 1	OVFEN	0x0	rw	Overflow event enable 0: Enabled

				1: Disabled
Bit 0	TMREN	0x0	rw	TMR enable 0: Enabled 1: Disabled

14.5.4.2 TMR16 and TMR17 control register 2 (TMRx_CTRL2)

Bit	Name	Reset value	Type	Description
Bit 30: 10	Reserved	0x0	resd	Kept at default value.
Bit 9	C1CIOS	0x0	rw	Channel 1 complementary idle output state Output disabled (OEN = 0), after dead-time: 0: C1OUTL=0 1: C1OUTL=1
Bit 8	C1IOS	0x0	rw	Channel 1 idle output state Output disabled (OEN = 0), after dead-time: 0: C1OUT=0 1: C1OUT=1
Bit 7:4	Reserved	0x0	resd	Kept at default value.
Bit 3	DRS	0x0	rw	DMA request source 0: Channel event 1: Overflow event
Bit 2	CCFS	0x0	rw	Channel control bit flash selection This bit only acts on channels with complementary output. If the channel control bits are buffered: 0: Control bits are updated by setting the HALL bit 1: Control bits are updated by setting the HALL bit or a rising edge on TRGIN.
Bit 1	Reserved	0x0	resd	Kept at default value.
Bit 0	CBCTRL	0x0	rw	Channel buffer control This bit acts on channels that have complementary output. 0: CxEN, CxCEN and CxOCTRL bits are not buffered. 1: CxEN, CxCEN and CxOCTRL bits are buffered.

14.5.4.3 TMR16 and TMR17 DMA/interrupt enable register (TMRx_IDEN)

Bit	Name	Reset value	Type	Description
Bit 15:10	Reserved	0x0	resd	Kept at default value.
Bit 9	C1DEN	0x0	rw	Channel 1 DMA request enable 0: Disabled 1: Enabled
Bit 8	OVFDEN	0x0	rw	Overflow event DMA request enable 0: Disabled 1: Enabled
Bit 7	BRKIE	0x0	rw	Brake interrupt enable 0: Disabled 1: Enabled
Bit 6	Reserved	0x0	resd	Kept at default value.
Bit 5	HALLIEN	0x0	rw	HALL interrupt enable 0: Disabled 1: Enabled
Bit 4: 2	Reserved	0x0	resd	Kept at default value.
Bit 1	C1IEN	0x0	rw	Channel 1 interrupt enable 0: Disabled 1: Enabled
Bit 0	OVFIEN	0x0	rw	Overflow interrupt enable 0: Disabled 1: Enabled

14.5.4.4 TMR16 and TMR17 interrupt status register (TMRx_ISTS)

Bit	Name	Reset value	Type	Description
Bit 15: 10	Reserved	0x0	resd	Kept at default value.
Bit 9	C1RF	0x0	rw0c	Channel 1 recapture flag

				This bit indicates whether a recapture is detected when C1IF=1. This bit is set by hardware, and cleared by writing “0”. 0: No capture is detected 1: Capture is detected.
Bit 8	Reserved	0x0	resd	Kept at default value.
Bit 7	BRKIF	0x0	rw0c	Brake interrupt flag This bit indicates whether the brake input is active or not. It is set by hardware and cleared by writing “0” 0: Inactive level 1: Active level
Bit 6	Reserved	0x0	resd	Kept at default value.
Bit 5	HALLIF	0x0	rw0c	HALL interrupt flag This bit is set by hardware on HALL event. It is cleared by writing “0”. 0: No Hall event occurred. 1: Hall event is detected. HALL even: CxEN, CxCEN and CxOCTRL are updated.
Bit 4: 2	Reserved	0x0	resd	Kept at default value.
Bit 1	C1IF	0x0	rw0c	Channel 1 interrupt flag If the channel 1 is configured as input mode: This bit is set by hardware on a capture event. It is cleared by software or read access to the TMRx_C1DT 0: No capture event occurred 1: Capture event is generated If the channel 1 is configured as output mode: This bit is set by hardware on a compare event. It is cleared by software. 0: No compare event occurred 1: Compare event is generated
Bit 0	OVFIF	0x0	rw0c	Overflow interrupt flag This bit is set by hardware on an overflow event. It is cleared by software. 0: No overflow event occurred 1: Overflow event is generated. If OVFEN=0 and OVFS=0 in the TMRx_CTRL1 register: − An overflow event is generated when OVFG= 1 in the TMRx_SWEVE register; − An overflow event is generated when the counter CVAL is reinitialized by a trigger event.

14.5.4.5 TMR16 and TMR17 software event register (TMRx_SWEVT)

Bit	Name	Reset value	Type	Description
Bit 15: 8	Reserved	0x0	resd	Kept at default value.
Bit 7	BRKSWTR	0x0	wo	Brake event triggered by software This bit is set by software to generate a brake event. 0: No effect 1: Generate a brake event.
Bit 6	Reserved	0x0	resd	Kept at default value.
Bit 5	HALLSWTR	0x0	wo	HALL event triggered by software This bit is set by software to generate a HALL event. 0: No effect 1: Generate a HALL event. Note: This bit acts only on channels with complementary output.
Bit 4: 2	Reserved	0x0	resd	Kept at default value.
Bit 1	C1SWTR	0x0	wo	Channel 1 event triggered by software This bit is set by software to generate a channel 1 event. 0: No effect 1: Generate a channel 1 event.
Bit 0	OVFSWTR	0x0	wo	Overflow event triggered by software This bit is set by software to generate an overflow event. 0: No effect 1: Generate an overflow event.

14.5.4.6 TMR16 and TMR17 channel mode register 1 (TMRx_CM1)

The channel can be used in input (capture mode) or output (compare mode). The direction of a channel is defined by the corresponding CxC bits. All the other bits of this register have different functions in input and output modes. The CxOx describes its function in output mode when the channel is in output mode, while the CxIx describes its function in output mode when the channel is in input mode. Attention must be given to the fact that the same bit can have different functions in input mode and output mode.

Output compare mode:

Bit	Name	Reset value	Type	Description
Bit 15:8	Reserved	0x0	resd	Kept at default value.
Bit 7	C1OSEN	0x0	rw	Channel 1 output switch enable 0: EXT input does not affect C1ORAW 1: When EXT input is high, C1ORAW is cleared.
Bit 6: 4	C1OCTRL	0x0	rw	Channel 1 output control This field defines the behavior of the original signal C1ORAW. 000: Disconnected. C1ORAW is disconnected from C1OUT; 001: C1ORAW is high when TMRx_CVAL=TMRx_C1DT 010: C1ORAW is low when TMRx_CVAL=TMRx_C1DT 011: Switch C1ORAW level when TMRx_CVAL=TMRx_C1DT 100: C1ORAW is forced low 101: C1ORAW is forced high. 110: PWM mode A －OWCDIR=0, C1ORAW is high once TMRx_C1DT>TMRx_CVA, else low; －OWCDIR=1, C1ORAW is low once TMRx_ C1DT <TMRx_CVA, else high; 111: PWM mode B － OWCDIR=0, C1ORAW is low once TMRx_ C1DT >TMRx_CVAL, else high; － OWCDIR=1, C1ORAW is high once TMRx_ C1DT <TMRx_CVAL, else low. *Note: In the configurations other than 000', the C1OUT is connected to C1ORAW. The C1OUT output level is not only subject to the changes of C1ORAW, but also the output polarity set by CCTRL.*
Bit 3	C1OBEN	0x0	rw	Channel 1 output buffer enable 0: Buffer function of TMRx_C1DT is disabled. The new value written to the TMRx_C1DT takes effect immediately. 1: Buffer function of TMRx_C1DT is enabled. The value to be written to the TMRx_C1DT is stored in the buffer register, and can be sent to the TMRx_C1DT register only on an overflow event.
Bit 2	C1OIEN	0x0	rw	Channel 1 output enable immediately In PWM mode A or B, this bit is used to accelerate the channel 1 output's response to the trigger event. 0: Need to compare the CVAL with C1DT before generating an output 1: No need to compare the CVAL and C1DT. An output is generated immediately when a trigger event occurs.
Bit 1: 0	C1C	0x0	rw	Channel 1 configuration This field is used to define the direction of the channel 1 (input or output), and the selection of input pin when C1EN='0': 00: Output 01: Input, C1IN is mapped on C1IFP1 10: Reserved 11: Reserved

Input capture mode:

Bit	Name	Reset value	Type	Description
Bit 15: 8	Reserved	0x0	resd	Kept at default value.
Bit 7: 4	C1DF	0x0	rw	Channel 1 digital filter This field defines the digital filter of the channel 1. N stands for the number of filtering, indicating that the input edge can pass the filter only after N sampling events. 0000: No filter, sampling is done at f_{DTS} 1000: $f_{SAMPLING}=f_{DTS}/8$, N=6 0001: $f_{SAMPLING}=f_{CK\_INT}$, N=2 1001: $f_{SAMPLING}=f_{DTS}/8$, N=8 0010: $f_{SAMPLING}=f_{CK\_INT}$, N=4 1010: $f_{SAMPLING}=f_{DTS}/16$, N=5 0011: $f_{SAMPLING}=f_{CK\_INT}$, N=8 1011: $f_{SAMPLING}=f_{DTS}/16$, N=6 0100: $f_{SAMPLING}=f_{DTS}/2$, N=6 1100: $f_{SAMPLING}=f_{DTS}/16$, N=8 0101: $f_{SAMPLING}=f_{DTS}/2$, N=8 1101: $f_{SAMPLING}=f_{DTS}/32$, N=5 0110: $f_{SMPLING}=f_{DTS}/4$, N=6 1110: $f_{SAMPLING}=f_{DTS}/32$, N=6 0111: $f_{SAMPLING}=f_{DTS}/4$, N=8 1111: $f_{SAMPLING}=f_{DTS}/32$, N=8
Bit 3: 2	C1IDIV	0x0	rw	Channel 1 input divider This field defines Channel 1 input divider. 00: No divider. An input capture is generated at each active edge. 01: An input compare is generated every 2 active edges 10: An input compare is generated every 4 active edges 11: An input compare is generated every 8 active edges Note: the divider is reset once C1EN='0'
Bit 1: 0	C1C	0x0	rw	Channel 1 configuration This field is used to define the direction of the channel 1 (input or output), and input rempapping when C1EN='0': 00: Output 01: Input, C1IN is mapped on C1IFP1 10: Reserved 11: Reserved

14.5.4.7 TMR16 and TMR17 channel control register (TMRx_CCTRL)

Bit	Name	Reset value	Type	Description
Bit 15: 4	Reserved	0x0	resd	Kept at default value.
Bit 3	C1CP	0x0	rw	Channel 1 complementary polarity 0: C1COUT is active high. 1: C1COUT is active low.
Bit 2	C1CEN	0x0	rw	Channel 1 complementary enable 0: Output is disabled. 1: Output is enabled.
Bit 1	C1P	0x0	rw	Channel 1 polarity When the channel 1 is configured in output mode: 0: C1OUT is active high 1: C1OUT is active low When the channel 1 is configured in input mode: The active edge of the input signal is defined by C1CP/C1P. 00: C1IN active edge is on its rising edge. When used as external trigger, C1IN is not inverted. 01: C1IN active edge is on its falling edge. When used as external trigger, C1IN is inverted. 10: Reserved 11: C1IN active edge is on its falling edge and rising edge. When used as external trigger, C1IN is not inverted.

Bit 0	C1EN	0x0	rw	Channel 1 enable 0: Input or output is disabled 1: Input or output is enabled

Table 14-13 Complementary output channel CxOUT and CxCOUT control bits with brake function

Control bit					Output state (1)	
OEN bit	FCSODIS bit	FCSOEN bit	CxEN bit	CxCEN bit	CxOUT output state	CxCOUT output state
1	X	0	0	0	Output disabled (no driven by the timer) CxOUT=0, Cx_EN=0	Output disabled (no driven by the timer) CxCOUT=0, CxCEN=0
		0	0	1	Output disabled (no driven by the timer) CxOUT=0, Cx_EN=0	CxORAW + polarity, CxCOUT= CxORAW xor CxCP, CxCEN=1
		0	1	0	CxORAW+ polarity CxOUT= CxORAW xor CxP, Cx_EN=1	Output disabled (no driven by the timer) CxCOUT=0, CxCEN=0
		0	1	1	CxORAW+polarity+ dead-time, Cx_EN=1	CxORAW inverted+polarity+dead-time, CxCEN=1
		1	0	0	Output disabled (no driven by the timer) CxOUT=CxP, Cx_EN=0	Output disabled (no driven by the timer) CxCOUT=CxCP, CxCEN=0
		1	0	1	Off-state (Output enabled with inactive level) CxOUT=CxP, Cx_EN=1	CxORAW + polarity, CxCOUT= CxORAW xor CxCP, CxCEN=1
		1	1	0	CxORAW + polarity, CxOUT= CxORAW xor CxP, Cx_EN=1	Off-state (Output enabled with inactive level) CxCOUT=CxCP, CxCEN=1
		1	1	1	CxORAW+ polarity+dead-time, Cx_EN=1	CxORAW inverted+polarity+dead-time, CxCEN=1
0	0	X	0	0	Output disabled (corresponding IO is not driven by the timer, IO floating)	
	0		0	1	Asynchronously: CxOUT=CxP, Cx_EN=0, CxCOUT=CxCP, CxCEN=0;	
	0		1	0	If the clock is present: after a dead-time, CxOUT=CxIOS, CxCOUT=CxCIOS, assuming that CxIOS and CxCIOS do not correspond to CxOUT and CxCOUT active level.	
	0		1	1		
	1		0	0	CxEN=CxCEN=0: output disabled (corresponding IO is not driven by the timer, IO floating)	
	1		0	1	In other cases: off-state (corresponding channel output invalid level)	
	1		1	0	Asynchronously: CxOUT =CxP, Cx_EN=1, CxCOUT=CxCP, CxCEN=1;	
	1		1	1	If the clock is present: after a dead-time, CxOUT=CxIOS, CxCOUT=CxCIOS, assuming that CxIOS and CxCIOS do not correspond to CxOUT and CxCOUT active level.	

Note: If the two outputs of a channel are not used (CxEN = CxCEN = 0), CxIOS, CxCIOS, CxP and CxCP must be cleared.

Note: The state of the external I/O pins connected to the complementary CxOUT and CxCOUT channels depends on the CxOUT and CxCOUT channel state and the GPIO and the IOMUX registers.

14.5.4.8 TMR16 and TMR17 counter value (TMRx_CVAL)

Bit	Name	Reset value	Type	Description
Bit 15: 0	CVAL	0x0	rw	Counter value

14.5.4.9 TMR16 and TMR17 divider (TMRx_DIV)

Bit	Name	Reset value	Type	Description
Bit 15: 0	DIV	0x0	rw	Divider value The counter clock frequency fCK_CNT = fTMR_CLK /(DIV[15:0]+1). The value of this register is transferred to the actual prescaler register when an overflow event occurs.

14.5.4.10 TMR16 and TMR17 period register (TMRx_PR)

Bit	Name	Reset value	Type	Description
Bit 15: 0	PR	0x0	rw	Period value This defines the period value of the TMR counter. The timer stops working when the period value is 0.

14.5.4.11 TMR16 and TMR17 repetition period register (TMRx_RPR)

Bit	Name	Reset value	Type	Description
Bit 15: 8	Reserved	0x0	resd	Kept at default value
Bit 7: 0	RPR	0x0	rw	Repetition of period value This field is used to reduce the generation rate of overflow events. An overflow event is generated when the repetition counter reaches 0.

14.5.4.12 TMR16 and TMR17 channel 1 data register (TMRx_C1DT)

Bit	Name	Reset value	Type	Description
Bit 15: 0	C1DT	0x0	rw	Channel 1 data register When the channel 1 is configured as input mode: The C1DT is the CVAL value stored by the last channel 1 input event (C1IN) When the channel 1 is configured as output mode: C1DT is the value to be compared with the CVAL value. Whether the written value takes effective immediately depends on the C1OBEN bit, and the corresponding output is generated on C1OUT as configured.

14.5.4.13 TMR16 and TMR17 brake register (TMRx_BRK)

Bit	Name	Reset value	Type	Description
Bit 31: 18	Reserved	0x0	resd	Kept at its default value.
Bit 19: 16	BKF	0x0	rw	Brake input filter This field is used to set the filter for brake input. The filter number N indicates that the input edge can pass through filter only after N sampling events. 0000: $f_{SAMPLING}=f_{DTS}$ (no filter) 1000: $f_{SAMPLING}=f_{DTS}$ /8, N=6 0001: $f_{SAMPLING}=f_{CK\_INT}$, N=2 1001: $f_{SAMPLING}=f_{DTS}$/8, N=8 0010: $f_{SAMPLING}=f_{CK\_INT}$, N=4 1010: $f_{SAMPLING}=f_{DTS}$/16, N=5 0011: $f_{SAMPLING}=f_{CK\_INT}$, N=8 1011: $f_{SAMPLING}=f_{DTS}$/16, N=6 0100: $f_{\_SAMPLING}=f_{DTS}$/2, N=6 1100: $f_{SAMPLING}=f_{DTS}$/16, N=8 0101: $f_{SAMPLING}=f_{DTS}$/2, N=8 1101: $f_{SAMPLING}=f_{DTS}$/32, N=5 0110: $f_{SMPLING}=f_{DTS}$/4, N=6 1110: $f_{SAMPLING}=f_{DTS}$/32, N=6 0111: $f_{SAMPLING}=f_{DTS}$/4, N=8

				1111: $f_{SAMPLING}=f_{DTS}/32$, N=8
Bit 15	OEN	0x0	rw	Output enable This bit acts on the channels as output. It is used to enable CxOUT and CxCOUT outputs. 0: Disabled 1: Enabled
Bit 14	AOEN	0x0	rw	Automatic output enable OEN is set automatically at an overflow event. 0: Disabled 1: Enabled
Bit 13	BRKV	0x0	rw	Brake input validity This bit is used to select the active level of a brake input. 0: Brake input is active low. 1: Brake input is active high.
Bit 12	BRKEN	0x0	rw	Brake enable This bit is used to enable brake input. 0: Brake input is disabled. 1: Brake input is enabled.
Bit 11	FCSOEN	0x0	rw	Frozen channel status when holistic output enable This bit acts on the channels that have complementary output. It is used to set the channel state when the timer is inactive and OEN=1. 0: CxOUT/CxCOUT outputs are disabled. 1: CxOUT/CxCOUT outputs are enabled. Output inactive level.
Bit 10	FCSODIS	0x0	rw	Frozen channel status when holistic output disable This bit acts on the channels that have complementary output. It is used to set the channel state when the timer is inactive and OEN=0. 0: CxOUT/CxCOUT outputs are disabled. 1: CxOUT/CxCOUT outputs are enabled. Output idle level.
Bit 9: 8	WPC	0x0	rw	Write protection configuration This field is used to enable write protection. 00: Write protection is OFF. 01: Write protection level 3, and the following bits are write protected: TMR1_STOP: DTC, STPEN, STPV and HOAEN TMRx_CTRL2: CxIOS and CxCIOS 10: Write protection level 2. The following bits and all bits in level 3 are write protected: TMR1_CCTRL: CxP and CxCP TMR1_STOP: FCSODIS and FCSOEN 11: Write protection level 1. The following bits and all bits in level 2 are write protected: TMR1_CMx: C2OCTRL and C2OBEN Note: When WPC>0, its content remains frozen until the next system reset.
Bit 7: 0	DTC	0x00	rw	Dead-time configuration This field defines the duration of the dead-time insertion. The 3-bit MSB of DTC[7: 0] is used for function selection: 0xx: DT = DTC [7: 0] * TDTS 10x: DT = (64+ DTC [5: 0]) * TDTS * 2 110: DT = (32+ DTC [4: 0]) * TDTS * 8 111: DT = (32+ DTC [4: 0]) * TDTS * 16

Note: Based on lock configuration, AOEN, BRKV, BRKEN, FCSODIS, FCSOEN and DTC[7:0] can all be write protected. Thus it is necessary to configure write protection when writing to the TMRx_BRK register for the first time.

14.5.4.14 TMR16 and TMR17 DMA control register (TMRX_DMACTRL)

Bit	Name	Reset value	Type	Description
Bit 15:13	Reserved	0x0	resd	Kept at default value.
Bit 12:8	DTB	0x00	rw	DMA transfer bytes This field defines the number of DMA transfers: 00000: 1 byte 00001: 2 bytes 00010: 3 bytes 00011: 4 bytes .... 10000: 17 bytes 10001: 18 bytes
Bit 7:5	Reserved	0x0	resd	Kept at default value.
Bit 4: 0	ADDR	0x00	rw	DMA transfer address offset ADDR is defined as an offset starting from the address of the TMRx_CTRL1 register: 00000: TMRx_CTRL1 00001: TMRx_CTRL2 00010: TMRx_STCTRL .....

14.5.4.15 TMR16 and TMR17 DMA data register (TMRx_DMADT)

Bit	Name	Reset value	Type	Description
Bit 15: 0	DMADT	0x0	rw	DMA data register A write/read operation to the DMADT register accesses any TMR register located at the following address: TMRx peripheral address + ADDR*4 to TMRx peripheral address + ADDR*4 + DTB*4

14.6 Advanced-control timers (TMR1)

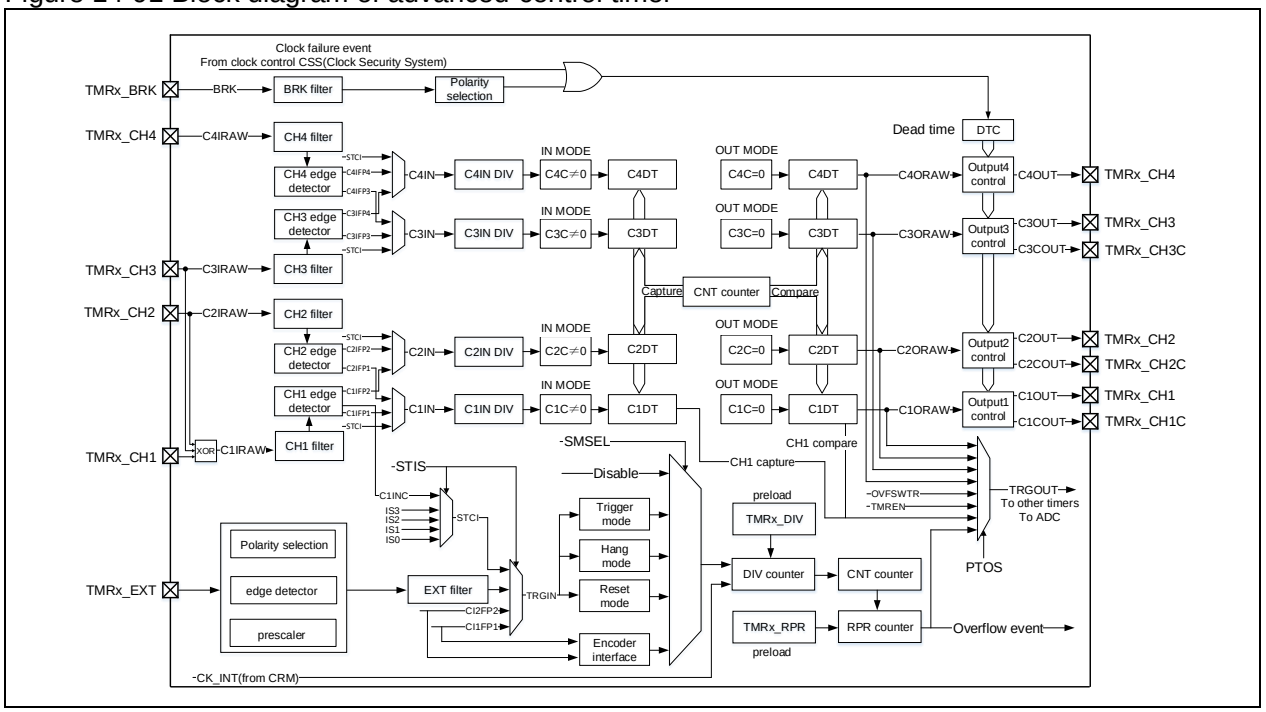
14.6.1 TMR1 introduction

The advanced-control timer TMR1 consists of a 16-bit counter supporting up and down counting modes, four channel registers, and four independent channels. It can be used for dead-time insertion, input capture and programmable PWM output.

14.6.2 TMR1 main features

- Clock source of counter: internal clock, external clock and internal trigger input
- 16-bit up, down, up/down, repetition and encoder mode counter
- Five independent channels for input capture, output compare, PWM generation, one-pulse mode output and dead-time insertion
- Three independent channels for complementary output
- TMR brake function
- Synchronization control between master and slave timers
- Brake signal input and filtering support
- Interrupt/DMA generation at overflow event, trigger event, brake input and channel event
- Support TMR burst DMA transfer

Figure 14-91 Block diagram of advanced-control timer

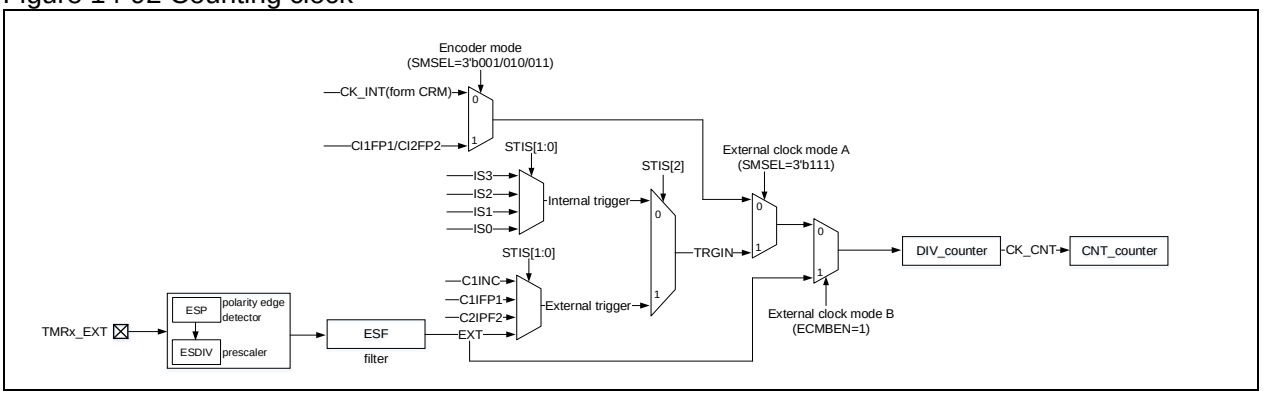


14.6.3 TMR1 functional overview

14.6.3.1 Counting clock

The count clock of TMR1 can be provided by an internal clock (CK_INT), external clock (external clock mode A and B) or internal trigger input (ISx).

Figure 14-92 Counting clock



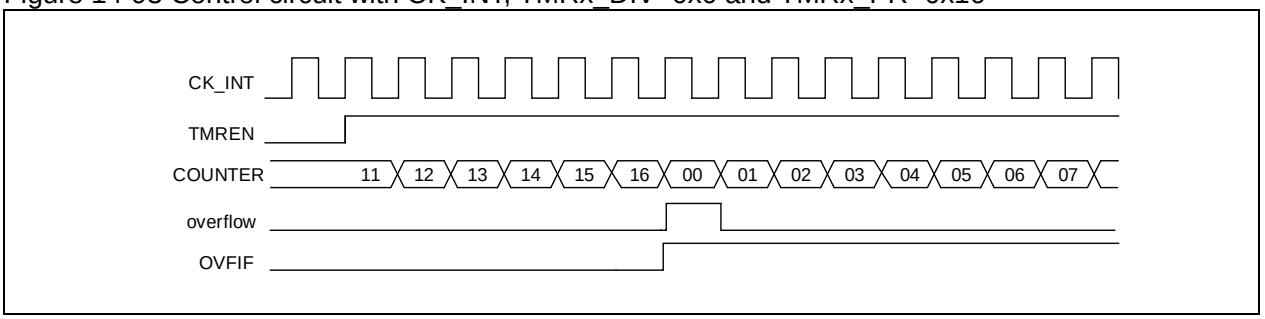
Internal clock (CK_INT)

By default, the CK_INT, which is divided by the prescaler, is used to drive the counter to start counting.

When TMR's APB clock prescaler factor is 1, the CK_INT frequency is equal to that of APB; otherwise, it doubles the APB clock frequency.

- Select counting mode by setting the TWCMSEL[1:0] in TMRx_CTRL1 register. If a unidirectional aligned counting mode is selected, it is necessary to select a counting direction through the OWCDIR in TMRx_CTRL1 register.
- Set counting frequency through TMRx_DIV register
- Set counting cycles through TMRx_PR register
- Enable counter by setting the TMREN bit in the TMRx_CTRL1 register

Figure 14-93 Control circuit with CK_INT, TMRx_DIV=0x0 and TMRx_PR=0x16



External clock (**TRGIN/EXT**)

The counter clock can be provided by two external clock sources, namely, TRGIN and EXT signals.

SMSEL=3'b111: External clock mode A is selected. By setting the STIS[2: 0] bit, select an external clock source TRGIN signal to drive the counter to start counting.

The external clock sources include:

C1INC (STIS=3'b100, rising edge and falling edge signal of channel 1)

C1IFP1 (STIS=3'b101, the channel 1 signal which goes through filtering and polarity selection)

C2IFP2 (STIS=3'b110, a channel 2 signal which goes through filtering and polarity selection)

EXT (STIS=3'b111, external input signal which goes through polarity selection, division and filtering)

ECMBEN=1: External clock mode B is selected. The counter is driven by EXT that has gone through polarity selection, frequency division and filtering. The external clock mode B, equivalent to the external clock mode A, selects EXT signal as an external clock source TRGIN.

To use external clock mode A, follow the steps below:

- Set external source TRGIN parameters

  If the TMRx_CH1 is used as a source of TRGIN, it is necessary to configure channel 1 input filter (C1DF[3:0] in TMRx_CM1 register) and channel 1 input polarity (C1P/C1CP in TMRx_CCTRL register);

  If the TMRx_CH2 is used as source of TRGIN, it is necessary to configure channel 2 input filter (C2DF[3:0] in TMRx_CM1 register) and channel 2 input polarity (C2P/C2CP in TMRx_CCTR register);

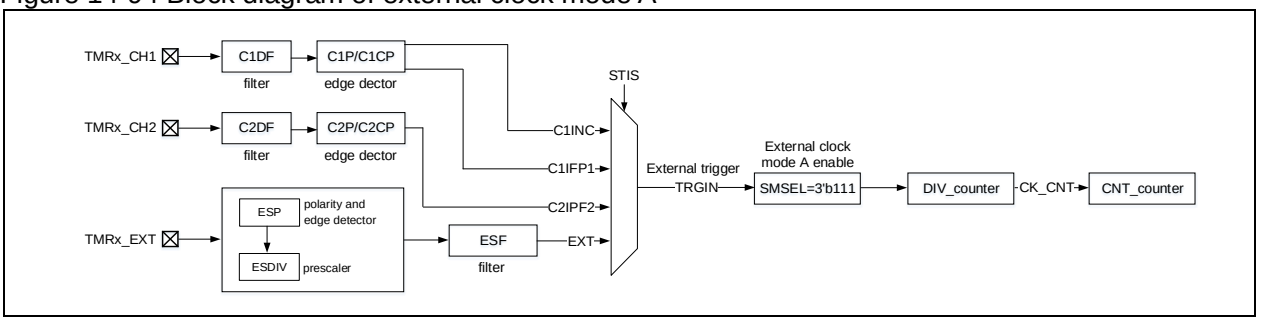
If the TMRx_EXT is used as a source of TRGIN, it is necessary to configure the external signal polarity (ESP in TMRx_STCTRL register), its frequency division (ESDIV[1:0] in TMRx_STCTRL) and filtering (ESF[3:0] in TMRx_STCTRL register).

- Set TRGIN signal source through the STIS[1:0] bit in TMRx_STCTRL register
- Enable external clock mode A by setting SMSEL=3'b111 in TMRx_STCTR register
- Set counting frequency through the DIV[15:0] in TMRx_DIV register
- Set counting period through the PR[15:0] in TMRx_PR register
- Enable counter through the TMREN bit in TMRx_CTRL1 register

To use external clock mode B, follow the steps below:

- Set external signal polarity through the ESP bit in TMRx_STCTRL register
- Set external signal frequency division through the ESDIV[1:0] bit in TMRx_STCTRL register
- Set external signal filtering through the ESF[3:0] bit in TMRx_STCTRL register
- Enable external clock mode B through the ECMBEN bit in TMRx_STCTR register
- Set counting frequency through the DIV[15:0] bit in TMRx_DIV register
- Set counting period through the PR[15:0] bit in TMRx_PR register
- Enable counter through the TMREN in TMRx_CTRL1 register

Figure 14-94 Block diagram of external clock mode A



Note: The delay between the signal on the input side and the actual clock of the counter is due to the synchronization circuit.

Figure 14-95 Counting in external clock mode A, PR=0x32 and DIV=0x0

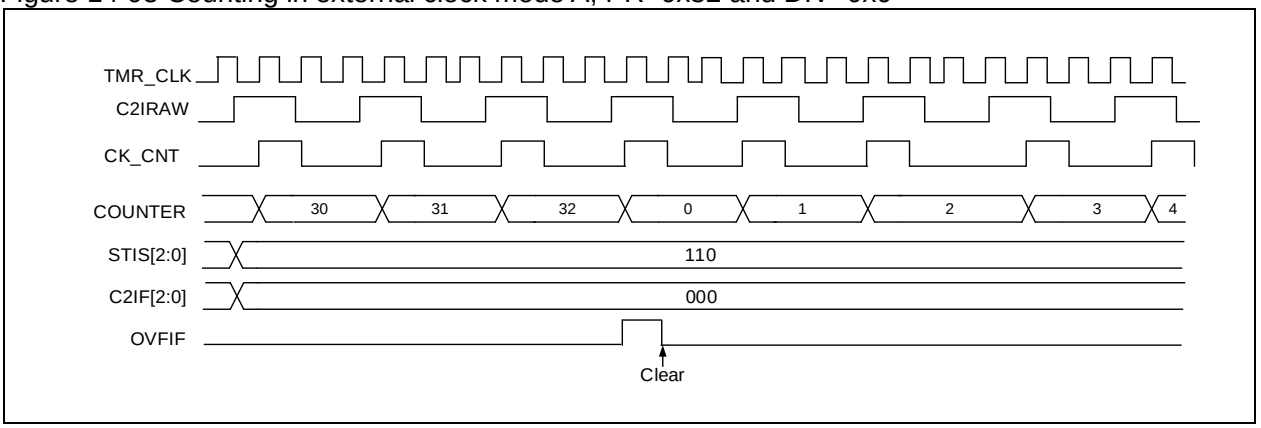
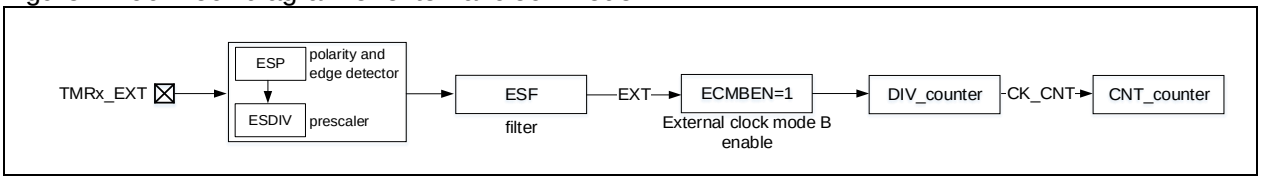
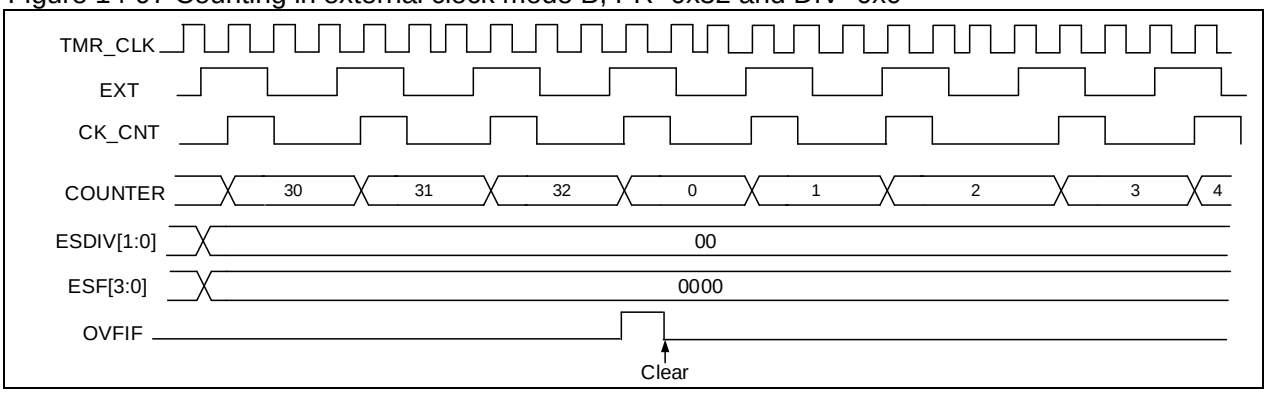


Figure 14-96 Block diagram of external clock mode B



Note: The delay between the EXT signal on the input side and the actual clock of the counter is due to the synchronization circuit.

Figure 14-97 Counting in external clock mode B, PR=0x32 and DIV=0x0



Internal trigger input (ISx)

Timer synchronization allows interconnection between several timers. The TMR_CLK of one timer can be provided by the TRGOUT output of another timer. Set the STIS[2: 0] bit to select internal trigger signal to enable counting.

The advanced-control timer consists of a 16-bit prescaler, which is used to generate the CK_CNT that enables the counter to count. The frequency division relationship between the CK_CNT and TMR_CLK can be adjusted by setting the value of the TMR1_DIV register. The prescaler value can be modified at any time, but it takes effect only when the next overflow event occurs.

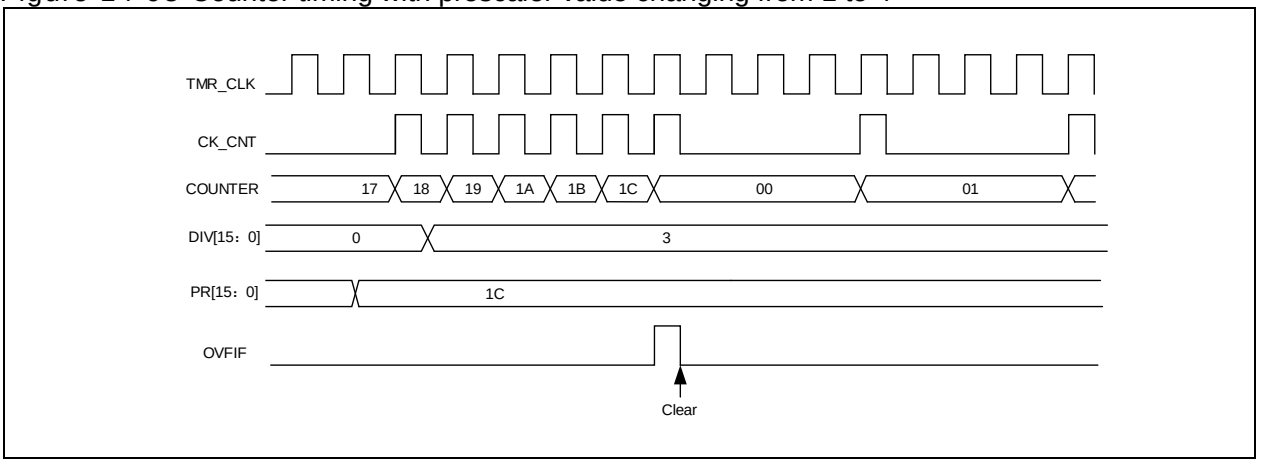
Below is the configuration procedure for internal trigger input:

- Set counting cycles through TMRx_PR register
- Set counting frequency through TMRx_DIV register
- Set counting modes through the TWCMSEL[1:0] in TMRx_CTRL1 register
- Select internal trigger by setting STIS[2:0]= 3'b000~3'b011 in TMRx_STCTRL register
- Select external clock mode A by setting SMSEL[2:0]=3'b111 in TMRx_STCTRL register
- Enable TMRx to start counting through the TMREN in TMRx_CTRL1 register

Table 14-14 TMRx internal trigger connection

Slave timer	IS0 (STIS=000)	IS1 (STIS=001)	IS2 (STIS=010)	IS3 (STIS=011)
TMR1	TMR15	-	TMR3	-
TMR3	TMR1	-	TMR15	-
TMR15	-	TMR3	TMR16	TMR17_OC

Figure 14-98 Counter timing with prescaler value changing from 1 to 4



14.6.3.2 Counting mode

The advanced-control timer consists of a 16-bit counter supporting up, down, up/down counting modes.

The TMRx_PR register is used to define counting period. The value in the TMRx_PR is immediately moved to the shadow register by default. When the periodic buffer is enabled (PRBEN=1), the value in the TMRx_PR register is transferred to the shadow register only at an overflow event.

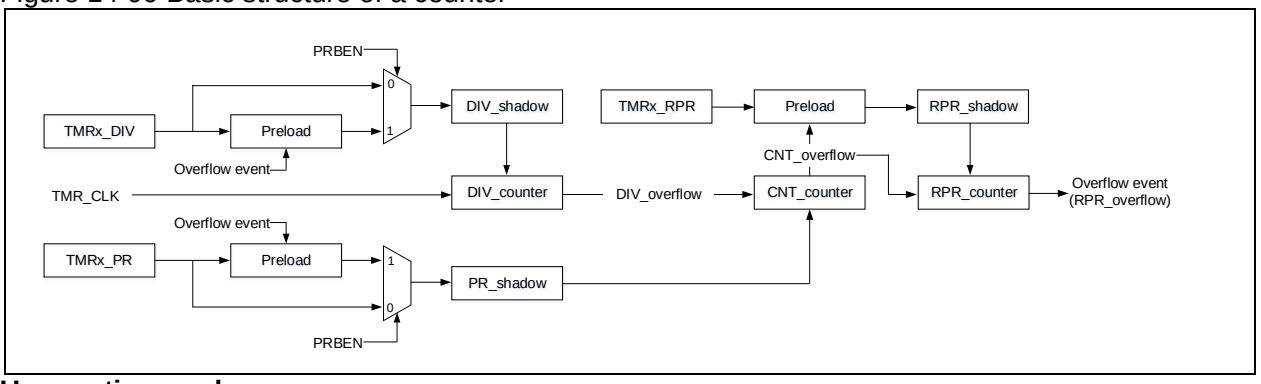
TMRx_DIV register is used to define the counter frequency. The counter counts once every DIV[15:0]+1 clock cycle. Similar to TMRx_PR register, after periodic buffer is enabled, the value of the TMRx_DIV register is transferred into the shadow register upon an overflow event.

Reading the TMRx_CNT register returns the current counter value. Writing the TMRx_CNT register will update the current counter value.

An overflow event is enabled by default. It can be disabled by setting OVFEN=1 in the TMRx_CTRL1 register. The OVFS bit in the TMRx_CTRL1 register is used to select the source of an overflow event, which is, by default, counter overflow or underflow, setting OVFSWTR, reset signal generated by slave mode timer controller in reset mode. Once the OVFS is set, an overflow event is generated only when overflow or underflow occurs.

Setting the TMREN bit (TMREN=1) enables the timer to start counting. Base on synchronization logic, however, the actual enable signal TMR_EN is set 1 clock cycle after the TMREN is set.

Figure 14-99 Basic structure of a counter



Upcounting mode

This mode is enabled by setting CMSEL[1:0]=2'b00 and OWCDIR=1'b0 in the TMRx_CTRL1 register.

In upcounting mode, the counter counts from 0 to the value programmed in the TMR1_PR register, restarts from 0, and generates a counter overflow event, with setting OVFIF bit to 1. If the overflow event is disabled, the counter is no longer reloaded with the prescaler and re-loaded value on counter overflow, otherwise, the prescaler and re-loaded value will be updated on an overflow event.

Figure 14-100 Overflow event when PRBEN=0

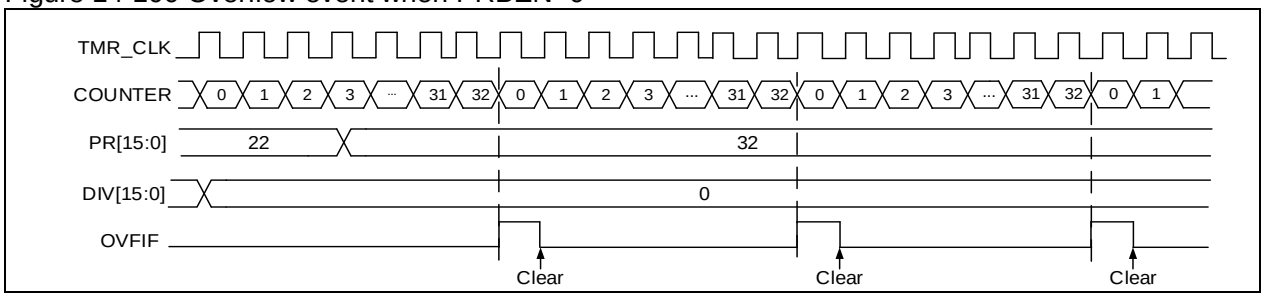
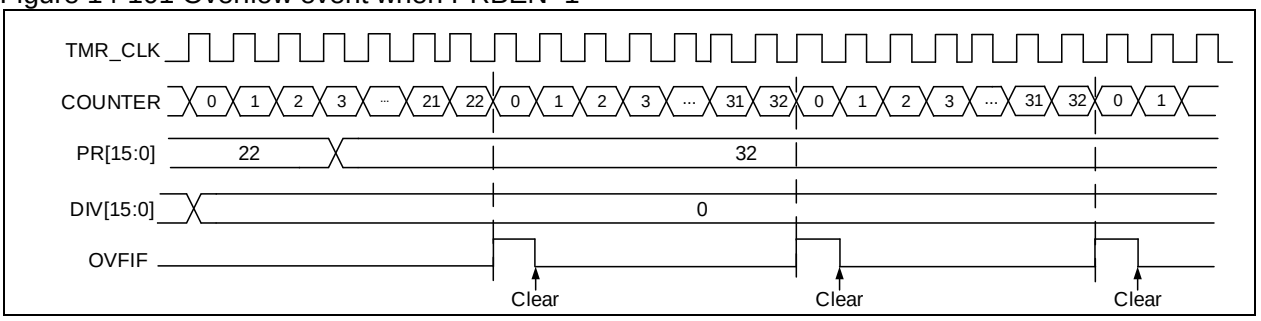


Figure 14-101 Overflow event when PRBEN=1

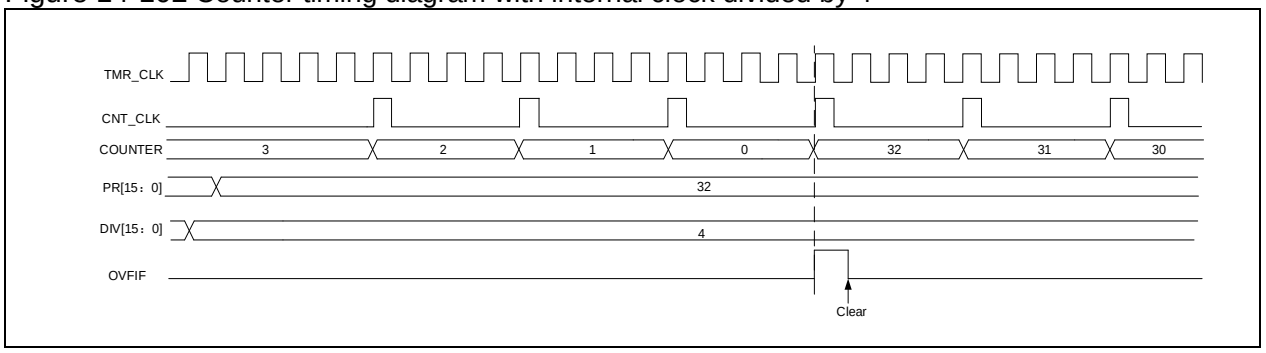


Downcounting mode

This mode is enabled by setting CMSEL[1:0]=2'b00 and OWCDIR=1'b1 in the TMRx_CTRL1 register.

In downcounting mode, the counter counts from the value programmed in the TMRx_PR register down to 0, and restarts from the value programmed in the TMRx_PR register, and generates a counter underflow event.

Figure 14-102 Counter timing diagram with internal clock divided by 4



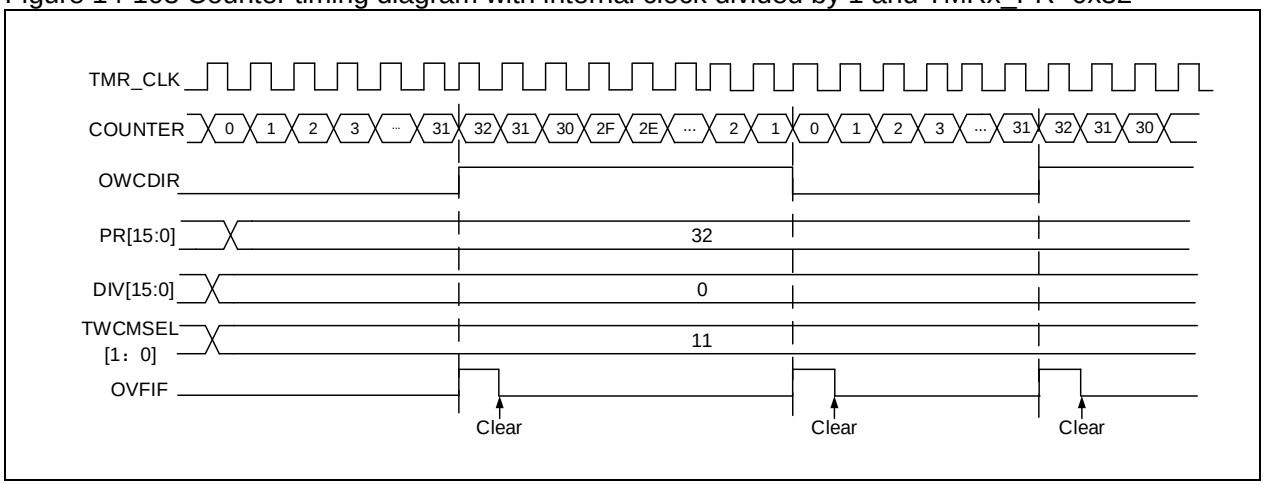
Up/down counting mode (center-aligned mode)

Up/down counting mode can be enabled by setting CMSEL[1:0]≠2'b00 in the TMRx_CTRL1 register. In up/down counting mode, the counter counts up/down alternatively. When the counter counts from the value programmed in the TMRx_PR register down to 1, an underflow event is generated, and then restarts counting from 0; when the counter counts from 0 to the value of the TMRx_PR register -1, an overflow event is generated, and then restarts counting from the value of the TMRx_PR register. The OWCDIR bit indicates the current counting direction.

The TWCMSEL[1:0] bit in the TMRx_CTRL1 register is used to select the condition under which the CxIF flag is set in two-way counting mode. In other words, when TWCMSEL[1:0]=2'b01 (counting mode 1) is selected, the CxIF flag is set only when the counter counts down; when TWCMSEL[1:0]=2'b10 (counting mode 2) is selected, the CxIF flag is set only when the counter counts up; when TWCMSEL[1:0]=2'b11 (counting mode 3) is selected, the CxIF flag is set when the counter counts up and down.

Note: The OWCDIR is ready-only in up/down counting mode.

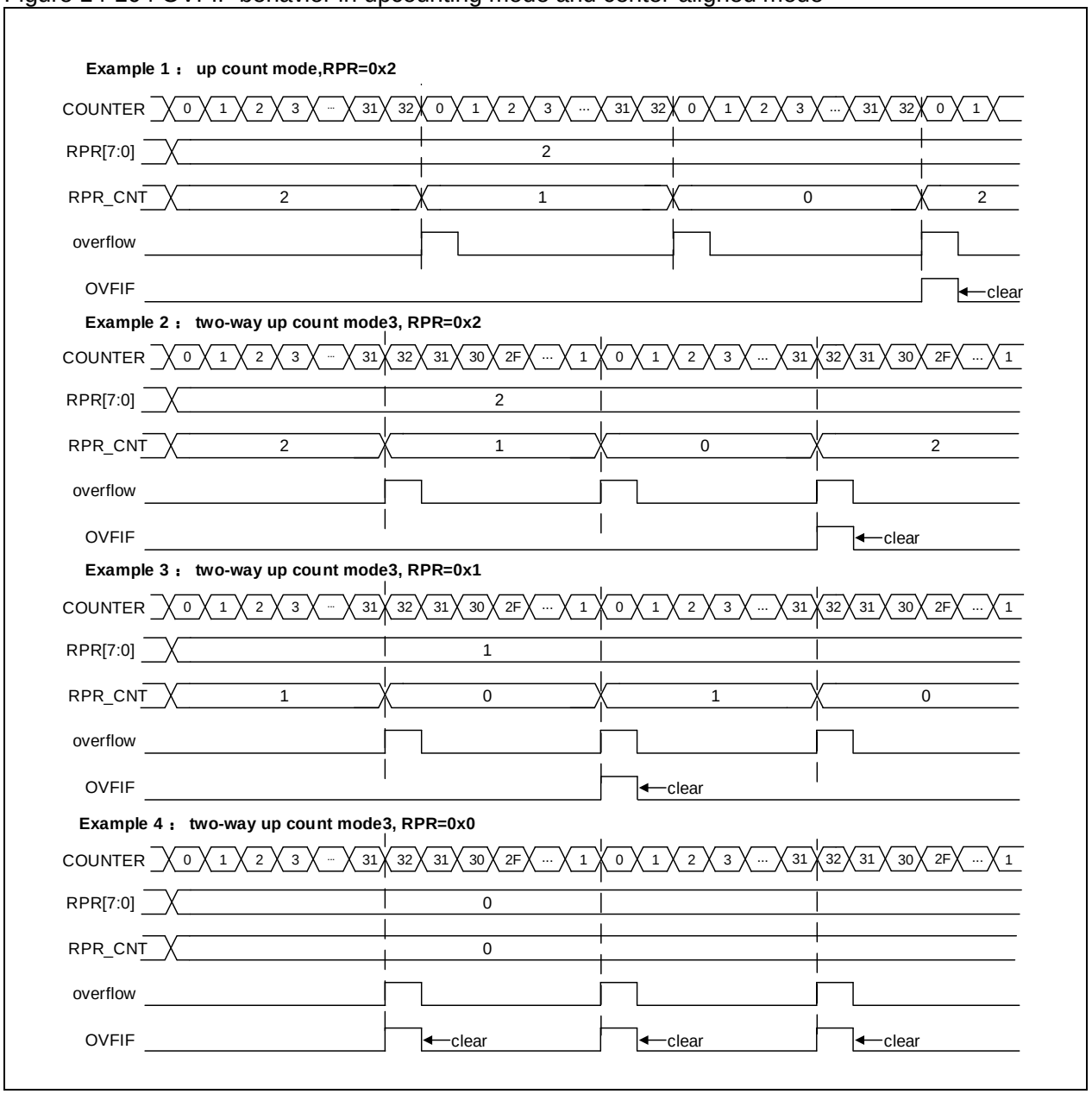
Figure 14-103 Counter timing diagram with internal clock divided by 1 and TMRx_PR=0x32



Repetition counter mode:

The TMRx_RPR register is used to set repetition counting mode. This mode is enabled when the repetition counter value is not equal to 0. In this mode, an overflow event is generated when a counter overflow occurs (RPR[7:0]+1). The repetition counter is decremented at each counter overflow. An overflow event is generated when the repetition counter reaches 0. The frequency of the overflow event can be adjusted by setting the repetition counter value.

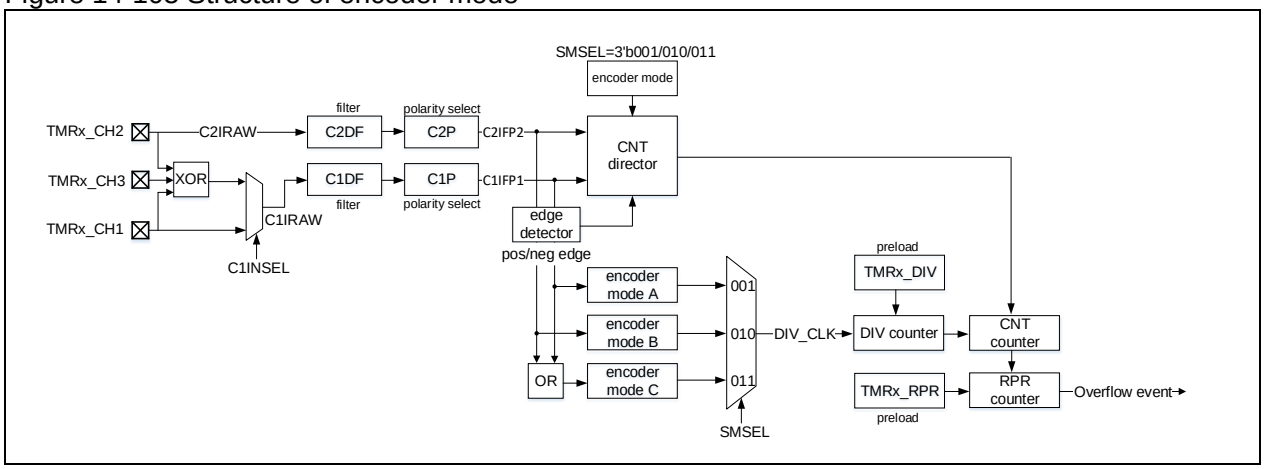
Figure 14-104 OVFIF behavior in upcounting mode and center-aligned mode



Encoder interface mode

In this mode, the two inputs (TMRx_CH1/ TMRx_CH2) are required. Depending on the level on one input, the counter counts up or down on the edge of the other input. The OWCDIR bit indicates the direction of the counter, as shown in the figure below:

Figure 14-105 Structure of encoder mode



Encoder mode A: SMSEL=3'b001. The counter counts on the selected C1IFP1 edge (rising and falling edges), and the counting direction is dependent on the edge direction of C1IFP1 and the level of C2IFP2.

Encoder mode B: SMSEL=3'b010. The counter counts on the selected C2IFP2 edge (rising and falling edges), and the counting direction is dependent on the edge direction of C2IFP2 and the level of C1IFP1.

Encoder mode C: SMSEL=3'b011. The counter counts on both C1IFP1 and C2IFP2 edges (rising and falling edges). The counting direction is dependent on the C1IFP1 edge direction and C2IFP2 level, and C2IFP2 edge direction and C1IFP1 level.

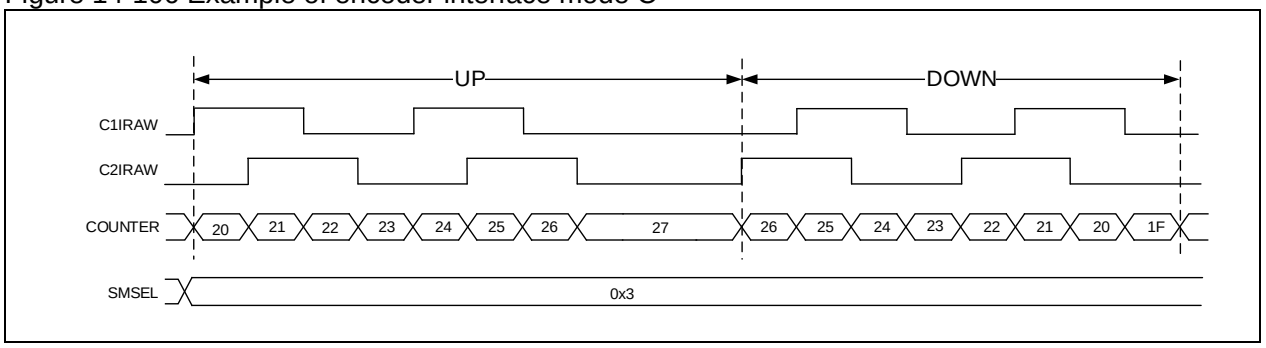
To use encoder mode, follow the procedures below:

- Set channel 1 input signal filtering through the C1DF[3:0] bit in the TMRx_CM1 register;
  Set channel 1 input signal active level through the C1P bit in the TMRx_CCTRL register
- Set channel 2 input signal filtering through the C2DF[3:0] bit in the TMRx_CM1 register;
  Set channel 2 input signal active level through the C2P bit in the TMRx_CCTRL register
- Set channel 1 as input mode through the C1C[1:0] bit in the TMRx_CM1 register;
  Set channel 2 as input mode through the C2C[1:0] bit in the TMRx_CM1 register
- Select encoder mode A (SMSEL=3'b001), encoder mode B (SMSEL=3'b010), or encoder mode C (SMSEL=3'b011) by setting the SMSEL[2:0] bit in the TMRx_STCTRL register
- Set counting cycles through the PR[15:0] bit in the TMRx_PR register
- Set counting frequency through the DIV[15:0] bit in the TMRx_DIV register
- Configure the corresponding IOs of TMRx_CH1 and TMRx_CH2 as multiplexed mode
- Enable counter through the TMREN bit in the TMRx_CTRL1 register

Table 14-15 Counting direction versus encoder signals

Active edge	Level on opposite signal (C1IFP1 to C2IFP2, C2IFP2 to C1IFP1)	C1IFP1 signal		C2IFP2 signal	
		Rising	Falling	Rising	Falling
Count on C1IFP1 only	High	Down	Up	No count	No count
	Low	Up	Down	No count	No count
Count on C2IFP2 only	High	No count	No count	Up	Down
	Low	No count	No count	Down	Up
Count on both C1IFP1 and C2IFP2	High	Down	Up	Up	Down
	Low	Up	Down	Down	Up

Figure 14-106 Example of encoder interface mode C



14.6.3.3 TMR input function

The TMR1 has four independent channels. Each channel can be configured as input or output. As input, each channel input signal is processed as follows:

- TMRx_CHx outputs the pre-processed CxIRAW. The C1INSE bit is used to select the source of C1IRAW from TMRx_CH1, or XOR-ed TMRx_CH1, TMRx_CH2 and TMRx_CH3.

  The sources of C2IRAW, C3IRAW and C4IRAW are TMRx_CH2, TMRx_CH3 and TMRx_CH4 respectively.
- CxIRAW passes digital filter and outputs a filtered CxIF signal. The digital filter uses the CxDF bit to set sampling frequency and sampling times.
- CxIF passes edge detector, and outputs the CxIFPx signal with edge selection. The edge selector is controlled by CxP and CxCP bits. It provides input rising edge, falling edge and both edges for selection.
- CxIFPx passes capture signal selector, and outputs the CxIN signal with capture signal selection. The capture signal selector is controlled by CxC bit. The CxIN source can be from CxIFPx, CyIFPx or STCI. Of those, CyIFPx (x≠y) refers to the CyIFPy signal which is from Y channel and has passed channel-x edge detector (for example, C1IFP2 refers to the C1IFP1 on channel 1 that has passed channel 2 edge detector). The STCI comes from slave timer controller, and its source is defined by STIS bit.
- CxIN passes input divider and outputs CxIPS signal. The divider factor can be defined as No division, divided/2, divided /4 or divided /8, through the CxIDIV bit.

Figure 14-107 Input/output channel 1 main circuit

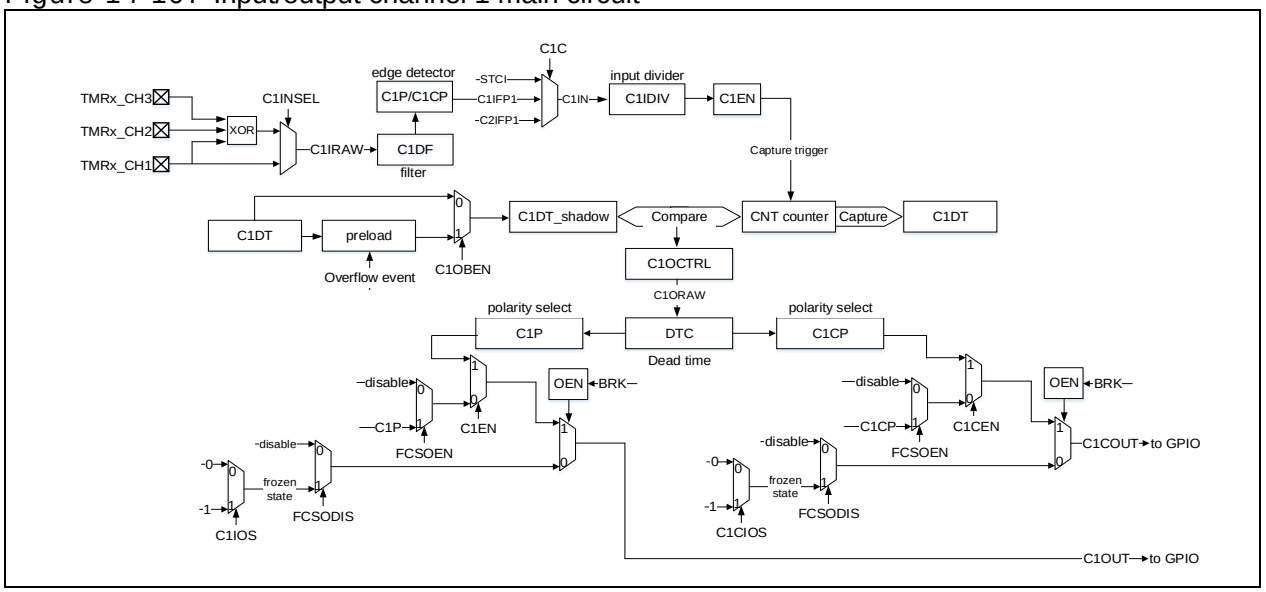
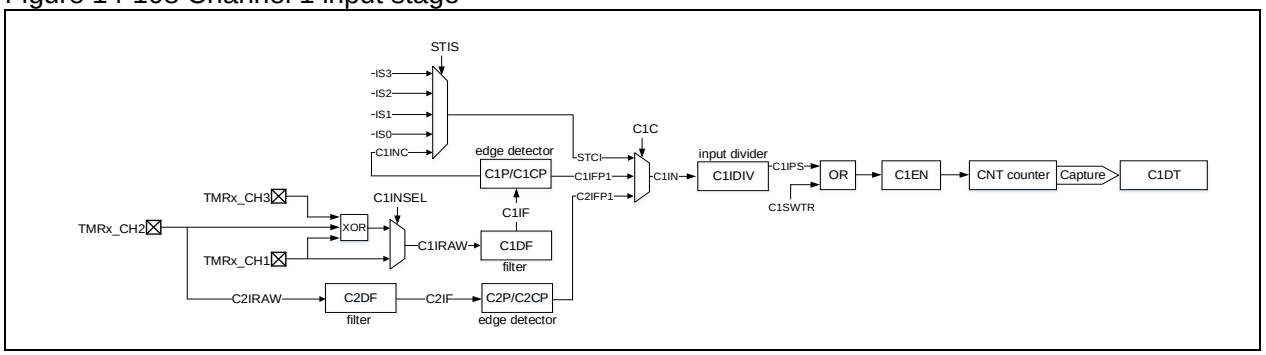


Figure 14-108 Channel 1 input stage



Input mode

In input mode, the TMRx_CxDT registers latch the current counter values after the selected trigger signal is detected, and the capture compare interrupt flag bit (CxIF) is set to 1. An interrupt/DMA request will be generated if the CxIEN bit and CxDEN bit are enabled. If the selected trigger signal is detected when the CxIF is set to 1, a capture overflow event is generated. The previous counter value will be overwritten with the current counter value, and the CxRF is set to 1.

To capture the rising edge of C1IN input, following the procedure below:

- Set C1C=01 in the TMRx_CM1 register to select the C1IN as channel 1 input
- Set C1IN signal filter bandwidth (CxDF[3: 0])
- Set the active edge of C1IN channel by writing C1P=0 (rising edge) in the TMR1_CCTRL register
- Program C1IN signal capture frequency divider (C1DIV[1: 0])
- Enable channel 1 input capture (C1EN=1)
- If needed, enable the relevant interrupt or DMA request by setting the C1IEN bit in the TMRx_IDEN register or the C1DEN bit in the TMRx_IDEN register

Timer input XOR function

The timer input pins (TMR1_CH1, TMR1_CH2 and TMR1_CH3) are connected to the channel 1 (selected by setting the C1INSE in the TMR1_CTRL2 register) through an XOR gate.

The XOR gate can be used to connect Hall sensors. For example, connect the three XOR inputs to the three Hall sensors respectively so as to calculate the position and speed of the rotation by analyzing three Hall sensor signals.

PWM input

PWM input mode is applied to channel 1 and 2. To use this mode, both C1IN and C2IN are mapped on the same TMRx_CHx, and the CxIFPx of either channel 1 or channel 2 must be configured as slave timer controller reset mode.

The PWM input mode can be used to measure the period and duty cycle of PWM input signals. For example, the user can measure the period and duty cycle of the PWM applied on channel 1 using the following procedures:

- Set C1C=2‘b01: select C1IN for C1IFP1
- Set C1P=1’b0, select C1IFP1 rising edge active
- Set C2C=2‘b10, select C2IN for C1IFP2
- Set C2P=1’b1, select C1IFP2 falling edge active
- Set STIS=3’b101, select C1IFP1 as the slave mode timer trigger signal
- Set SMSEL=3‘b100: configure the slave timer reset mode
- Set C1EN=1’b1 and C2EN=1’b1. Enable channel 1 and input capture

After above configuration, the rising edge of channel 1 input signal will trigger the capture and stores the capture value into C1DT register, and it will reset the counter at the same time. The falling edge of the channel 1 input signal triggers the capture and stores the capture value into C2DT register. The period of the channel 1 input signal is calculated through C1DT, and its duty cycle is obtained through C2DT.

Figure 14-109 PWM input mode configuration example

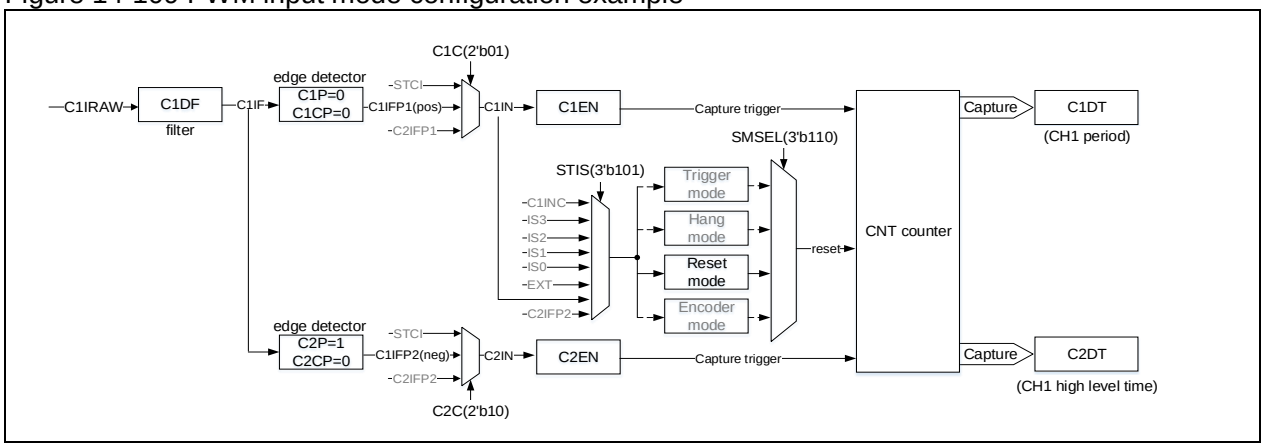
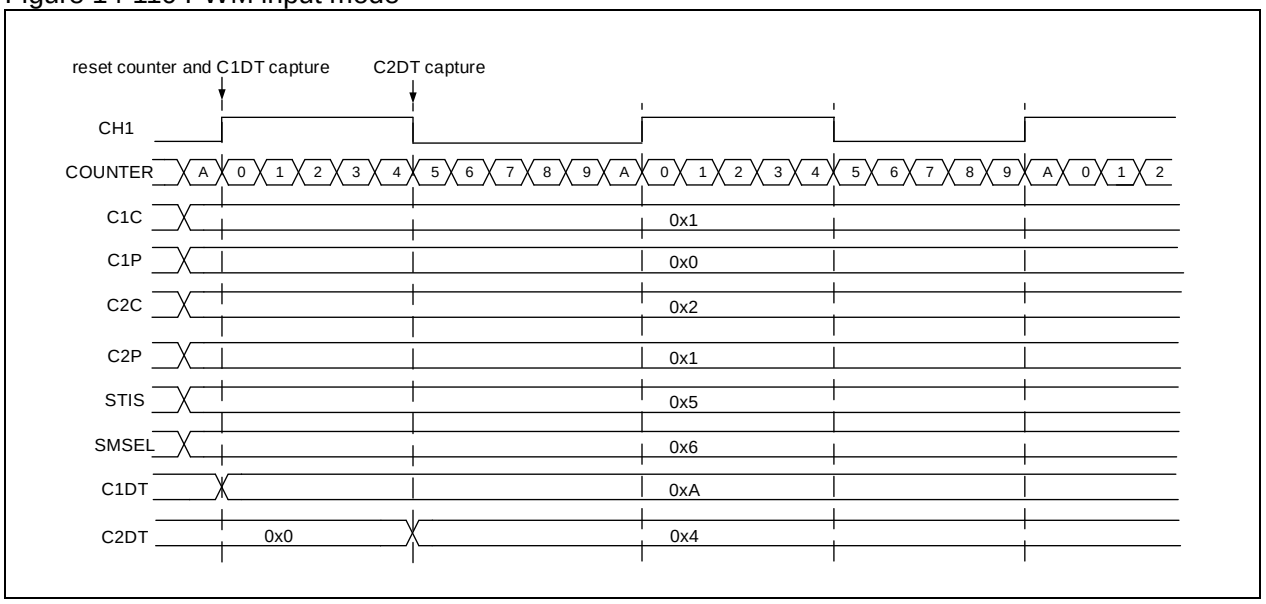


Figure 14-110 PWM input mode



14.6.3.4 TMR output function

The TMR output consists of a comparator and an output controller. It is used to program the period, duty cycle and polarity of output signals. The advanced-control timer output function varies from one channel to one channel.

Figure 14-111 Channel output stage (channel 1 to 3)

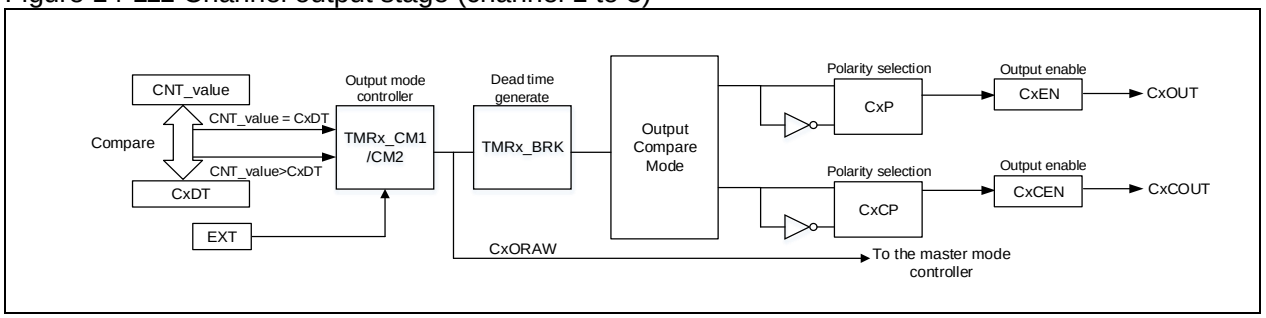
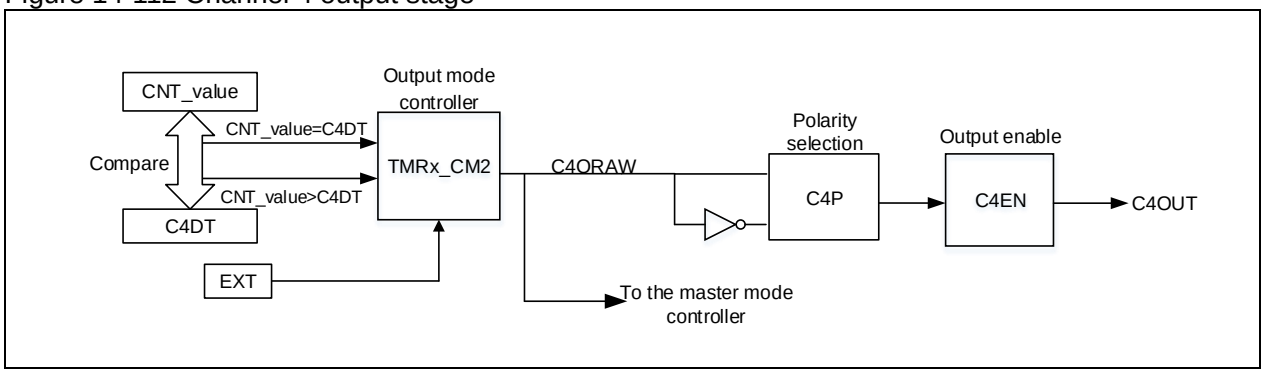


Figure 14-112 Channel 4 output stage



Output mode

CxC[1: 0]≠2'b00 is used to configure the channel as output mode. In this case, the counter value is compared with that in the CxDT register, and the intermediate signal CxORAW is generated according to the output mode selected by CxOCTRL[2: 0], and this signal is sent to IO after being processed by the output control circuit. The period of the output signal is configured by the TMR1_PR register, while the duty cycle is configured by the CxDT register.

Output compare modes include:

PWM mode A:

Enable PWM mode A by setting CxOCTRL=3'b110. In upcounting mode, C1ORAW outputs high when TMRx_C1DT>TMRx_CVAL, otherwise, it is low; In downcounting mode, C1ORAW outputs low when TMRx_C1DT<TMRx_CVAL, otherwise, it is high.

To use PWM mode A, the following procedures are recommended:

- Set PWM periods through TMRx_PR register
- Set PWM duty cycles through TMRx_CxDT register
- Select PWM mode A by setting CxOCTRL=3'b110 in the TMRx_CM1/CM2 register
- Set counting frequency through TMRx_DIV register
- Select counting mode by setting the TWCMSEL[1:0] bit in the TMRx_CTRL1 register
- Select output polarity through the CxP and CxCP bits in the TMRx_CCTRL register
- Enable channel output through the CxEN and CxCEN bits in the TMRx_CCTRL register
- Enable TMRx output through the OEN bit in the TMRx_BRK register
- Configure GPIOs corresponding to TMR output channels as multiplexed mode
- Enable TMRx to start counting through the TMREN bit in the TMRx_CTRL1 register.

PWM mode B:

Enable PWM mode B by setting CxOCTRL=3'b111. In upcounting mode, C1ORAW outputs low when TMRx_C1DT>TMRx_CVAL, otherwise, it is high; In downcounting mode, C1ORAW outputs high when TMRx_C1DT<TMRx_CVAL, otherwise, it is low.

Forced output mode:

Enable forced output mode by setting CxOCTRL=3'b100/101. In this case, the CxORAW is forced to be the programmed level, regardless of the counter value. Despite this, the channel flag bit and DMA request still depend on the compare result.

Output compare mode:

Enable output compare mode by setting CxOCTRL=3'b001/010/011. In this case, when the counter value matches the value of the CxDT register, the CxORAW is forced high (CxOCTRL=3'b001), low (CxOCTRL=3'b010) or toggling (CxOCTRL=3'b011).

One-pulse mode:

This is a particular case of PWM mode. Enable one-pulse by setting OCMEN=1. In this mode, the comparison match is performed in the current counting period. The TMREN bit is cleared as soon as the current counting is completed. Therefore, only one pulse is output. When in upcounting mode, the configuration must follow the rule: CVAL<CxDT≤PR; in downcounting mode, CVAL>CxDT is required.

Fast output mode:

Enable this mode by setting CxOIEN=1. If enabled, the CxORAW signal will not change when the counter value matches the CxDT, but change at the beginning of the current counting period. In other words, the comparison result is advanced, so the comparison result between the counter value and the TMRx_CxDT register will determine the level of CxORAW in advance.

Figure 14-113 gives an example of output compare mode (toggle) with C1DT=0x3. When the counter value is equal to 0x3, C1OUT toggles.

Figure 14-114 gives an example of the combination between upcounting mode and PWM mode A. The output signal behaves when PR=0x32 but CxDT is configured with a different value.

Figure 14-115 gives an example of the combination between up/down counting mode and PWM mode A. The output signal behaves when PR=0x32 but CxDT is configured with a different value.

Figure 14-116 gives an example of the combination between upcounting mode and one-pulse PWM mode B. The counter only counts only one cycle, and the output signal sends only one pulse.

Figure 14-113 C1ORAW toggles when counter value matches the C1DT value

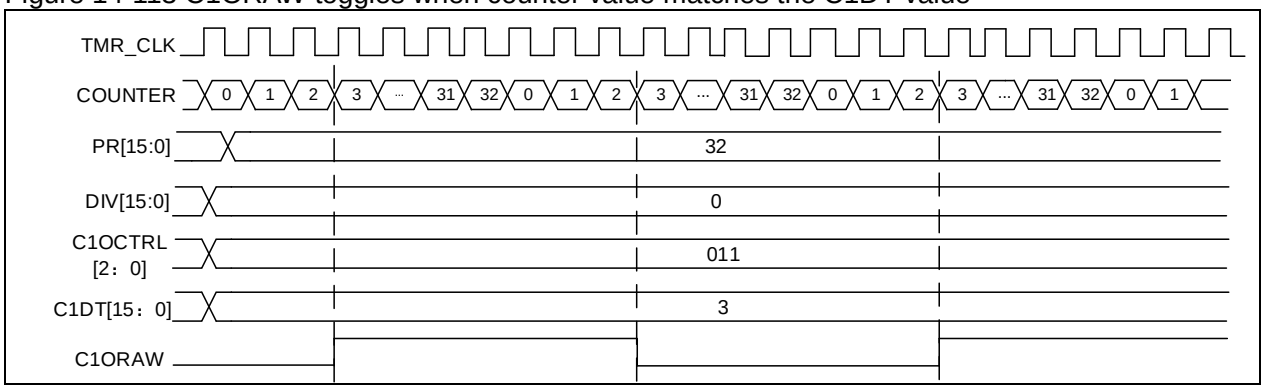


Figure 14-114 Upcounting mode and PWM mode A

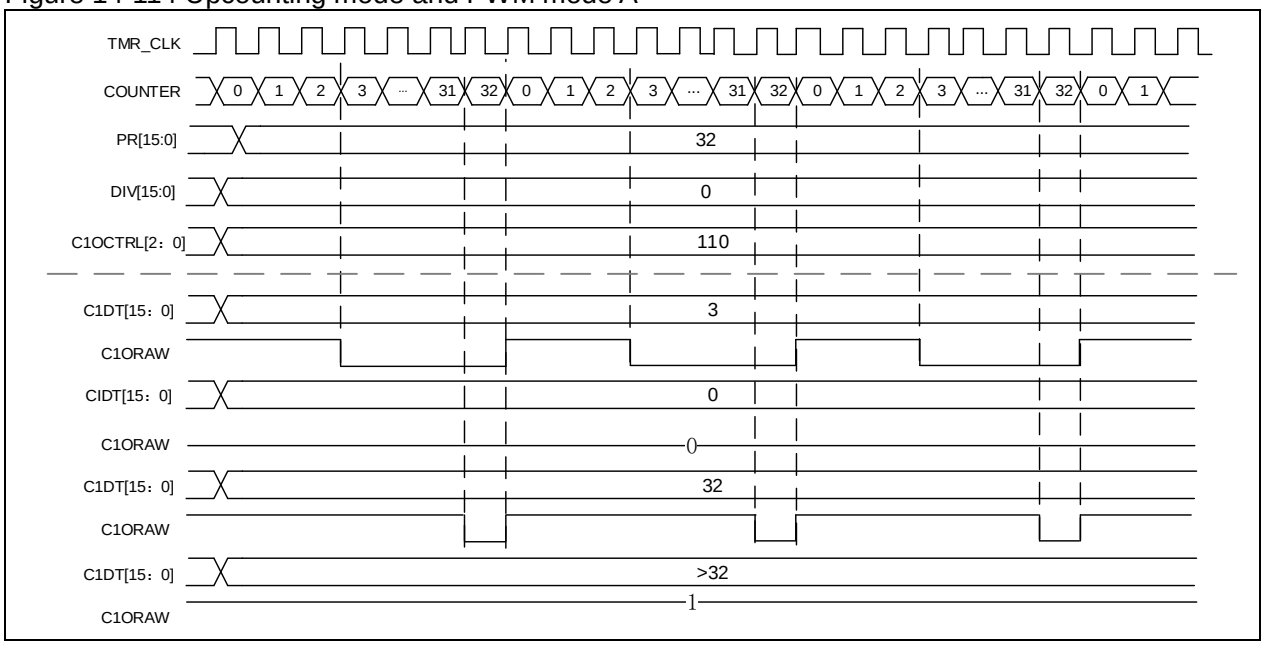


Figure 14-115 Up/down counting mode and PWM mode

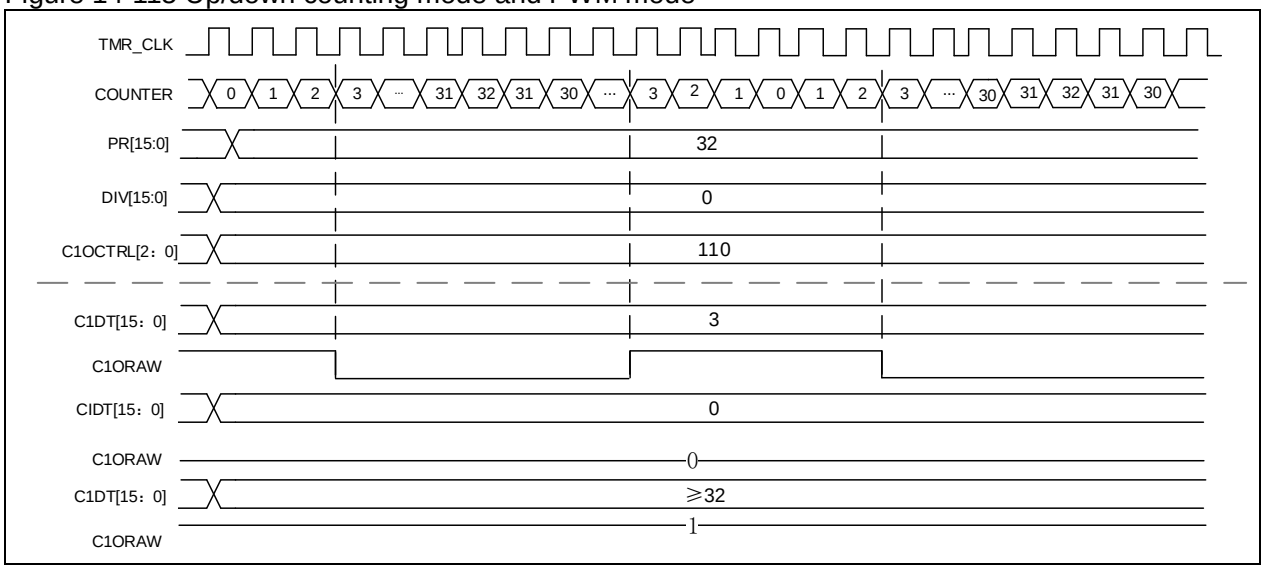
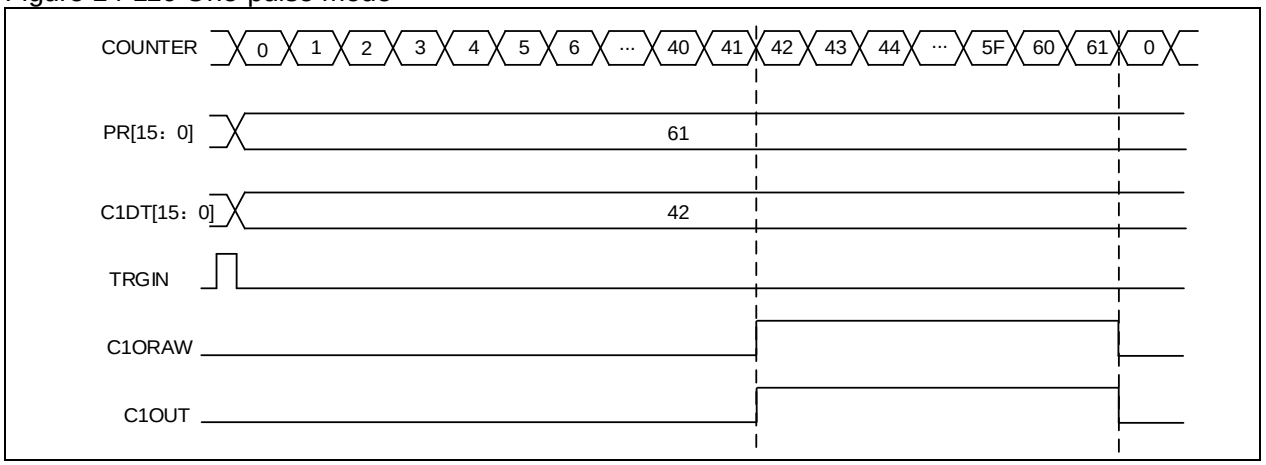


Figure 14-116 One-pulse mode



Master mode timer event output

When TMR is used as a master timer, one of the following source of signals can be selected as TRGOUT output to a slave mode timer. This is done by setting the PTOS bit in the TMRxCTRL2 register.

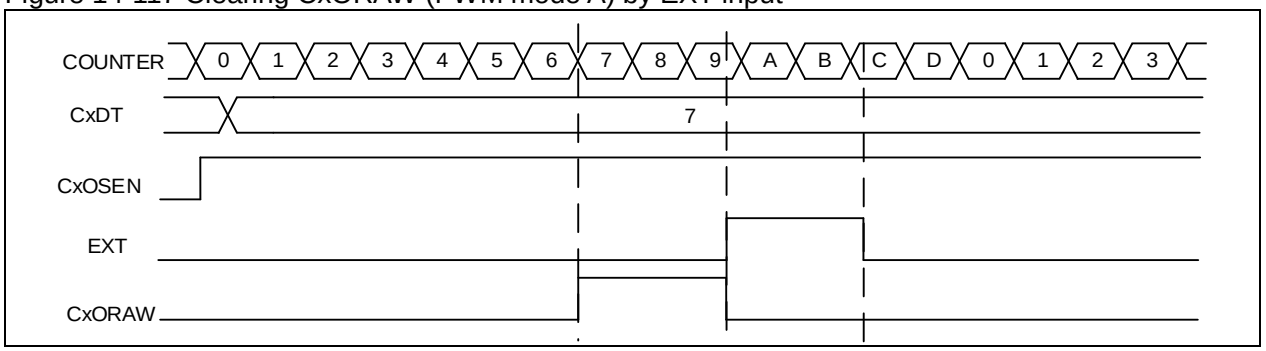
- PTOS=3’b000, TRGOUT output software overflow event (OVFSWTR bit in TMRx_SWEVT register)
- PTOS=3’b001, TRGOUT output counter enable
- PTOS=3’b010, TRGOUT output counter overflow event
- PTOS=3’b011, TRGOUT output capture and compare event
- PTOS=3’b100, TRGOUT output C1ORAW
- PTOS=3’b101, TRGOUT output C2ORAW
- PTOS=3’b110, TRGOUT output C3ORAW
- PTOS=3’b111, TRGOUT output C4ORAW

CxORAW clear

When the CxOSEN bit is set, the CxORAW signal for a given channel is cleared by applying a high level to the EXT input. The CxORAW signal remains unchanged until the next overflow event.

This function can only be used in output capture or PWM modes, and does not work in forced output mode. *Figure 14-117* shows the example of clearing CxORAW. When the EXT input is high, the CxORAW signal, which was originally high, is driven low; when the EXT is low, the CxORAW signal outputs the corresponding level according to the comparison result between the counter value and CxDT value.

Figure 14-117 Clearing CxORAW (PWM mode A) by EXT input



Dead-time insertion

The channel 1 to 3 of the advanced-control timers contains a set of reverse channel output. This function is enabled by the CxCEN bit and its polarity is defined by CxCP. Refer to Table 14-17 for more information about the output state of CxOUT and CxCOUT.

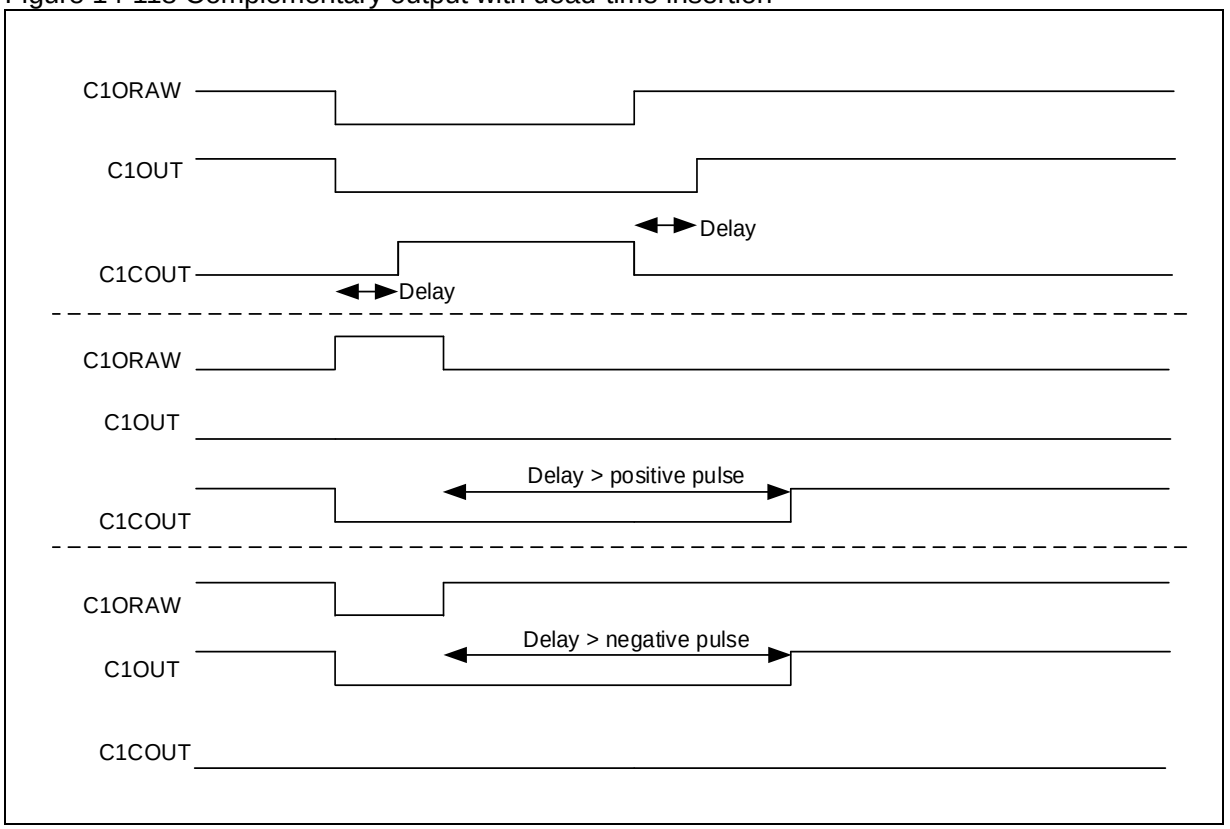
The dead-time is activated when switching to IDLEF state (OEN falling down to 0).

Setting both CxEN and CxCEN bits, and using DTC[7:0] bit to insert dead-time of different durations. After the dead-time insertion, the rising edge of the CxOUT is delayed compared to the rising edge of the reference signal; the rising edge of the CxCOU is delayed compared to the falling edge of the reference signal.

If the delay is greater than the width of the active output, then theC1OUT and C1COUT will not generate corresponding pulses. Therefore the dead-time should be less than the width of the active output.

Figure 14-118 gives an example of dead-time insertion when CxP=0, CxCP=0, OEN=1, CxEN=1 and CxCEN=1.

Figure 14-118 Complementary output with dead-time insertion



14.6.3.5 TMR brake function

When the brake function is enabled (BRKEN=1), the CxOUT and CxCOUT are jointly controlled by OEN, FCSODIS, FCSOEN, CxIOS and CxCIOS. But, CxOUT and CxCOUT cannot be set both to active level at the same time. Please refer to Table 14-17 for more details.

The brake source can be a brake input pin or a clock failure event. The polarity is controlled by the BRKV bit.

When a brake event occurs, there are the following actions:

- The OEN bit is cleared asynchronously, and the channel output state is selected by setting the FCSODIS bit. This function works even if the MCU oscillator is off.
- After OEN is cleared, the channel output level is defined by the CxIOS bit. If FCSODIS=0, the timer output is disabled, otherwise, the output enable remains high.
- When complementary output mode is used:
  - The output is first put in reset state, that is, inactive state (depending on the polarity). This is done asynchronously so that it works even if no clock is provided to the timer.
  - If the timer clock is still active, then the dead-time generator is activated. The CxIOS and CxCIOS bits are used to program the level after dead-time. Even in this case, the CxIOS and CxCIOS cannot be driven to their active level at the same time.

    Note: because of synchronization on OEN, the dead-time duration is usually longer than usual (around 2 clk_tmr clock cycles)
  - If FCSODIS=0, the timer releases the enable output, otherwise, it keeps the enable output; the enable output becomes high as soon as one of the CxEN and CxCEN bits becomes high.
- If the brake interrupt or DMA request is enabled, the brake statue flag is set, and a brake interrupt or DMA request can be generated.
- If AOEN=1, the OEN bit is automatically set again at the next overflow event.

Note: When the brake input is active, the OEN cannot be set, nor the status flag, BRKIF can be cleared.

Figure 14-119 Example of TMR output control

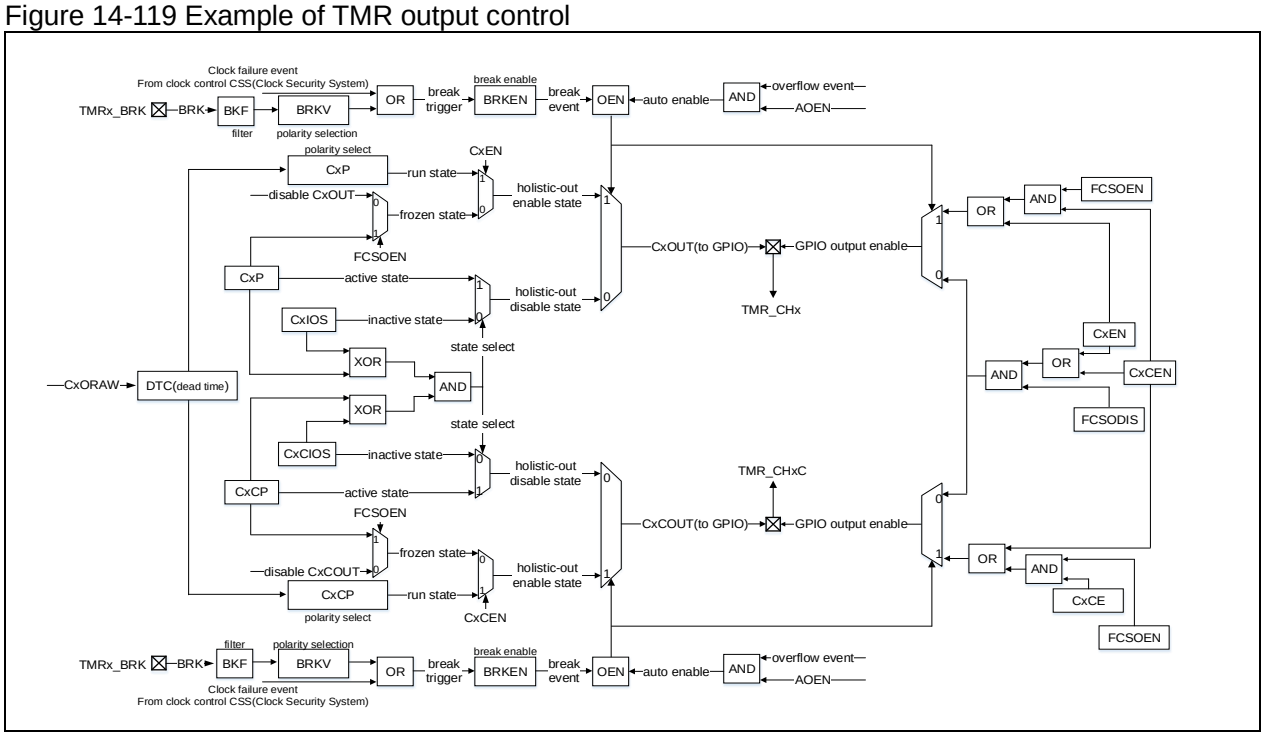
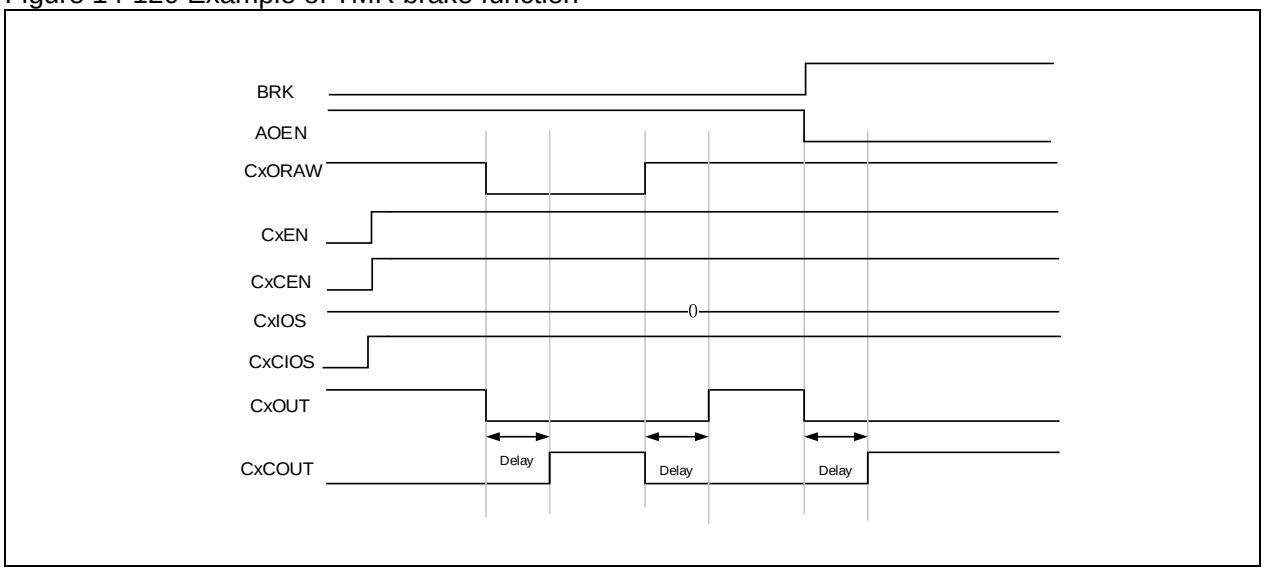


Figure 14-120 Example of TMR brake function



14.6.3.6 TMR synchronization

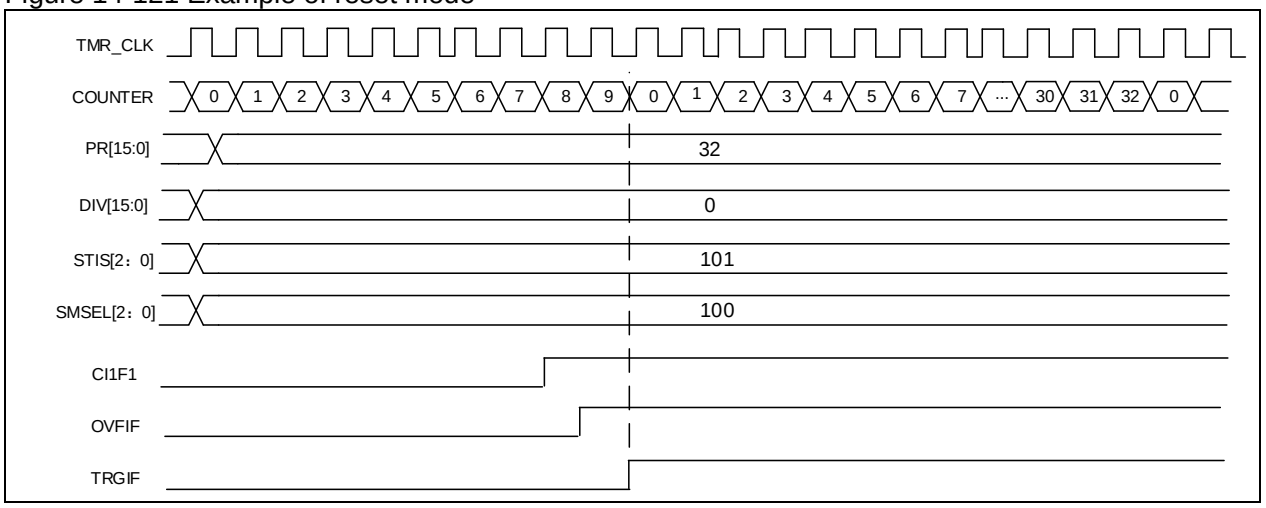
The timers are linked together internally for timer synchronization. Master timer is selected by setting the PTOS[2: 0] bit; Slave timer is selected by setting the SMSEL[2: 0] bit.

Slave modes include:

Slave mode: Reset mode

The counter and its prescaler can be reset by a selected trigger signal. An overflow event can be generated when OVFS=0.

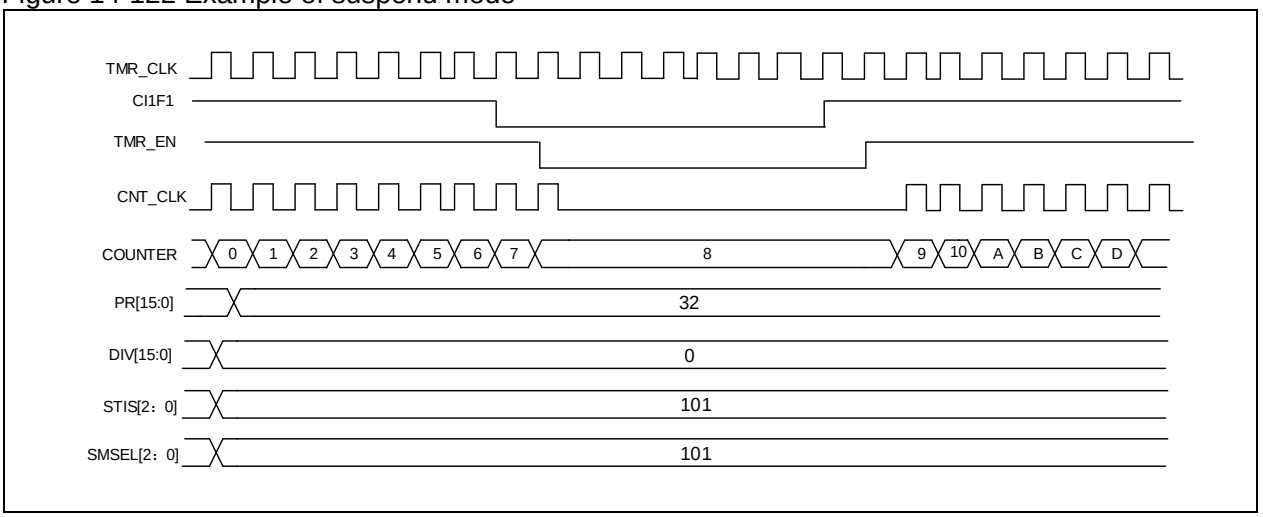
Figure 14-121 Example of reset mode



Slave mode: Suspend mode

In this mode, the counter is controlled by a selected trigger input. The counter starts counting when the trigger input is high and stops as soon as the trigger input is low.

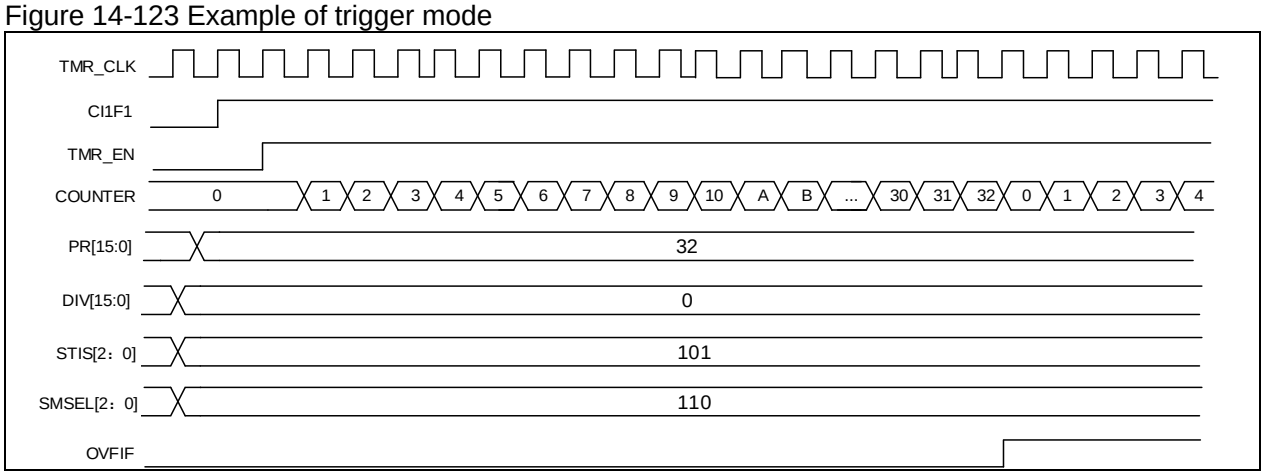
Figure 14-122 Example of suspend mode



Slave mode: Trigger mode

The counter can start counting on the rising edge of a selected trigger input (TMR_EN=1)

Figure 14-123 Example of trigger mode



For more examples on timer synchronization, please refer to Section *14.2.3.5.*

14.6.3.7 Debug mode

When the microcontroller enters debug mode (Cortex®-M0+ core halted), the TMR1 counter stops counting by setting the TMR1_PAUSE in the DEBUG module.

14.6.4 TMR1 registers

These peripheral registers must be accessed by half-words (16 bits) or words (32 bits).

TMR1 registers are mapped into a 16-bit addressable space.

Table 14-16 TMR1 register map and reset value

Register	Offset	Reset value
TMR1_CTRL1	0x00	0x0000
TMR1_CTRL2	0x04	0x0000
TMR1_STCTRL	0x08	0x0000
TMR1_IDEN	0x0C	0x0000
TMR1_ISTS	0x10	0x0000
TMR1_SWEVT	0x14	0x0000
TMR1_CM1	0x18	0x0000
TMR1_CM2	0x1C	0x0000
TMR1_CCTRL	0x20	0x0000
TMR1_CVAL	0x24	0x0000
TMR1_DIV	0x28	0x0000
TMR1_PR	0x2C	0x0000
TMR1_RPR	0x30	0x0000
TMR1_C1DT	0x34	0x0000
TMR1_C2DT	0x38	0x0000
TMR1_C3DT	0x3C	0x0000
TMR1_C4DT	0x40	0x0000
TMR1_BRK	0x44	0x0000
TMR1_DMACTRL	0x48	0x0000
TMR1_DMADT	0x4C	0x0000
TMR1_CM3	0x70	0x0000
TMR1_C5DT	0x74	0x0000

14.6.4.1 TMR1 control register1 (TMR1_CTRL1)

Bit	Name	Reset value	Type	Description
Bit 15: 10	Reserved	0x0	resd	Kept at default value.
Bit 9: 8	CLKDIV	0x0	rw	Clock division This field is used to define the division ratio between digital filter sampling frequency (f_{DTS}) and timer clock frequency (f_{CK_INT}). it is also used to set the division ratio between dead time base (T_{DTS}) and timer clock period (T_{CK_INT}). 00: No division, $f_{DTS}=f_{CK\_INT}$ 01: Divided by 2, $f_{DTS}=f_{CK\_INT}/2$ 10: Divided by 4, $f_{DTS}=f_{CK\_INT}/4$ 11: Reserved
Bit 7	PRBEN	0x0	rw	Period buffer enable 0: Period buffer is disabled 1: Period buffer is enabled

Bit 6: 5	TWCMSEL	0x0	rw	Two-way counting mode selection 00: One-way counting mode, depending on the OWCDIR bit 01: Two-way counting mode 1, count up and down alternately, the CxIF bit is set only when the counter counts down 10: Two-way counting mode 2, count up and down alternately, the CxIF bit is set only when the counter counts up 11: Two-way counting mode 3, count up and down alternately, the CxIF bit is set when the counter counts up / down
Bit 4	OWCDIR	0x0	rw	One-way count direction 0: Up 1: Down
Bit 3	OCMEN	0x0	rw	One cycle mode enable This bit is use to select whether to stop counting at an update event 0: The counter does not stop at an update event 1: The counter stops at an update event
Bit 2	OVFS	0x0	rw	Overflow event source This bit is used to select overflow event or DMA request sources. 0: Counter overflow, setting the OVFSWTR bit or overflow event generated by slave timer controller 1: Only counter overflow generates an overflow event
Bit 1	OVFEN	0x0	rw	Overflow event enable 0: Enabled 1: Disabled
Bit 0	TMREN	0x0	rw	TMR enable 0: Disabled 1: Enabled

14.6.4.2 TMR1 control register 2 (TMR1_CTRL2)

Bit	Name	Reset value	Type	Description
Bit 31	TRGOUT2EN	0x0	rw	TRGOUT2 enable 0: TRGOUT2 disabled 1: TRGOUT2 enabled
Bit 30: 15	Reserved	0x0	resd	Kept at default value.
Bit 14	C4IOS	0x0	rw	Channel 4 idle output state
Bit 13	C3CIOS	0x0	rw	Channel 3 complementary idle output state
Bit 12	C3IOS	0x0	rw	Channel 3 idle output state
Bit 11	C2CIOS	0x0	rw	Channel 2 complementary idle output state
Bit 10	C2IOS	0x0	rw	Channel 2 idle output state
Bit 9	C1CIOS	0x0	rw	Channel 1 complementary idle output state Output disabled (OEN = 0), after dead-time: 0: C1OUTL=0 1: C1OUTL=1
Bit 8	C1IOS	0x0	rw	Channel 1 idle output state Output disabled (OEN = 0), after dead-time: 0: C1OUT=0 1: C1OUT=1
Bit 7	C1INSEL	0x0	rw	C1IN selection 0: CH1 pin is connected to C1IRAW input 1: The XORed result of CH1, CH2 and CH3 pins is connected to C1IRAW input
Bit 6: 4	PTOS	0x0	rw	Master TMR output selection This field is used to select the TMRx signal sent to the slave timer. 000: Reset 001: Enable 010: Overflow 011: Compare pulse 100: C1ORAW signal

				101: C2ORAW signal 110: C3ORAW signal 111: C4ORAW signal
Bit 3	DRS	0x0	rw	DMA request source 0: Channel event 1: Overflow event
Bit 2	CCFS	0x0	rw	Channel control bit flash selection This bit only acts on channels that have complementary output. If the channel control bits are buffered: 0: Control bits are updated by setting the HALL bit 1: Control bits are updated by setting the HALL bit or a rising edge on TRGIN.
Bit 1	Reserved	0x0	resd	Kept at default value.
Bit 0	CBCTRL	0x0	rw	Channel buffer control This bit acts on channels that have complementary output. 0: CxEN, CxCEN and CxOCTRL bits are not buffered. 1: CxEN, CxCEN and CxOCTRL bits are buffered.

14.6.4.3 TMR1 slave timer control register (TMR1_STCAL)

Bit	Name	Reset value	Type	Description
Bit 15	ESP	0x0	rw	External signal polarity 0: High or rising edge 1: Low or falling edge
Bit 14	ECMBEN	0x0	rw	External clock mode B enable This bit is used to enable external clock mode B 0: Disabled 1: Enabled
Bit 13: 12	ESDIV	0x0	rw	External signal divide This field is used to select the frequency division of an external trigger 00: Normal 01: Divided by 2 10: Divided by 4 11: Divided by 8
Bit 11: 8	ESF	0x0	rw	External signal filter This field is used to filter an external signal. The external signal can be sampled only after it has been generated N times 0000: No filter, sampling by 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
Bit 7	STS	0x0	rw	Subordinate TMR synchronization If enabled, master and slave timer can be synchronized. 0: Disabled 1: Enabled
Bit 6: 4	STIS	0x0	rw	Subordinate TMR input selection This field is used to select the subordinate TMR input. 000: Internal selection 0 (IS0)

				001: Internal selection 1 (IS1) 010: Internal selection 2 (IS2) 011: Internal selection 3 (IS3) 100: C1IRAW input detector (C1INC) 101: Filtered input 1 (C1IF1) 110: Filtered input 2 (C1IF2) 111: External input (EXT) Please refer to Table 14-14 for more information on ISx for each timer.
Bit 3	Reserved	0x0	resd	Kept at its default value.
Bit 2: 0	SMSEL	0x0	rw	Subordinate TMR mode selection 000: Slave mode is disabled 001: Encoder mode A 010: Encoder mode B 011: Encoder mode C 100: Reset mode — Rising edge of the TRGIN input reinitializes the counter 101: Suspend mode — The counter starts counting when the TRGIN is high 110: Trigger mode — A trigger event is generated at the rising edge of the TRGIN input 111: External clock mode A — Rising edge of the TRGIN input clocks the counter Note: Please refer to count mode section for the details on encoder mode A/B/C.

14.6.4.4 TMR1 DMA/interrupt enable register (TMR1_IDEN)

Bit	Name	Reset value	Type	Description
Bit 15	Reserved	0x0	resd	Kept at default value.
Bit 14	TDEN	0x0	rw	Trigger DMA request enable 0: Disabled 1: Enabled
Bit 13	HALLDE	0x0	rw	HALL DMA request enable 0: Disabled 1: Enabled
Bit 12	C4DEN	0x0	rw	Channel 4 DMA request enable 0: Disabled 1: Enabled
Bit 11	C3DEN	0x0	rw	Channel 3 DMA request enable 0: Disabled 1: Enabled
Bit 10	C2DEN	0x0	rw	Channel 2 DMA request enable 0: Disabled 1: Enabled
Bit 9	C1DEN	0x0	rw	Channel 1 DMA request enable 0: Disabled 1: Enabled
Bit 8	OVFDEN	0x0	rw	Overflow event DMA request enable 0: Disabled 1: Enabled
Bit 7	BRKIE	0x0	rw	Brake interrupt enable 0: Disabled 1: Enabled
Bit 6	TIEN	0x0	rw	Trigger interrupt enable 0: Disabled 1: Enabled
Bit 5	HALLIEN	0x0	rw	HALL interrupt enable 0: Disabled 1: Enabled
Bit 4	C4IEN	0x0	rw	Channel 4 interrupt enable 0: Disabled 1: Enabled
Bit 3	C3IEN	0x0	rw	Channel 3 interrupt enable

				0: Disabled 1: Enabled
Bit 2	C2IEN	0x0	rw	Channel 2 interrupt enable 0: Disabled 1: Enabled
Bit 1	C1IEN	0x0	rw	Channel 1 interrupt enable 0: Disabled 1: Enabled
Bit 0	OVFIEN	0x0	rw	Overflow interrupt enable 0: Disabled 1: Enabled

14.6.4.5 TMR1 interrupt status register (TMR1_ISTS)

Bit	Name	Reset value	Type	Description
Bit 31: 17	Reserved	0x0	resd	Kept at default value.
Bit 16	C5IF	0x0	rw0c	Channel 5 interrupt flag When a compare event occurred, this bit is set by hardware, and cleared by writing “0”. 0: No compare event is detected 1: Compare event is detected.
Bit 15: 13	Reserved	0x0	resd	Kept at default value.
Bit 12	C4RF	0x0	rw0c	Channel 4 recapture flag Please refer to C1RF description.
Bit 11	C3RF	0x0	rw0c	Channel 3 recapture flag Please refer to C1RF description.
Bit 10	C2RF	0x0	rw0c	Channel 2 recapture flag Please refer to C1RF description.
Bit 9	C1RF	0x0	rw0c	Channel 1 recapture flag This bit indicates whether a recapture is detected when C1IF=1. This bit is set by hardware, and cleared by writing “0”. 0: No capture is detected 1: Capture is detected.
Bit 8	Reserved	0x0	resd	Kept at default value.
Bit 7	BRKIF	0x0	rw0c	Brake interrupt flag This bit indicates whether the brake input is active or not. It is set by hardware and cleared by writing “0” 0: Inactive level 1: Active level
Bit 6	TRGIF	0x0	rw0c	Trigger interrupt flag This bit is set by hardware on a trigger event. It is cleared by writing “0”. 0: No trigger event occurs 1: Trigger event is generated. Trigger event: an active edge is detected on TRGIN input, or any edge in suspend mode.
Bit 5	HALLIF	0x0	rw0c	HALL interrupt flag This bit is set by hardware on HALL event. It is cleared by writing “0”. 0: No Hall event occurs. 1: Hall event is detected. HALL event: CxEN, CxCEN and CxOCTRL are updated.
Bit 4	C4IF	0x0	rw0c	Channel 4 interrupt flag Please refer to C1IF description.
Bit 3	C3IF	0x0	rw0c	Channel 3 interrupt flag Please refer to C1IF description.
Bit 2	C2IF	0x0	rw0c	Channel 2 interrupt flag Please refer to C1IF description.
Bit 1	C1IF	0x0	rw0c	Channel 1 interrupt flag If the channel 1 is configured as input mode: This bit is set by hardware on a capture event. It is cleared by software or read access to the TMRx_C1DT 0: No capture event occurs

				1: Capture event is generated If the channel 1 is configured as output mode: This bit is set by hardware on a compare event. It is cleared by software. 0: No compare event occurs 1: Compare event is generated
Bit 0	OVFIF	0x0	rw0c	Overflow interrupt flag This bit is set by hardware on an overflow event. It is cleared by software. 0: No overflow event occurs 1: Overflow event is generated. If OVFEN=0 and OVFS=0 in the TMRx_CTRL1 register: − An overflow event is generated when OVFG= 1 in the TMRx_SWEVE register; − An overflow event is generated when the counter CVAL is reinitialized by a trigger event.

14.6.4.6 TMR1 software event register (TMR1_SWEVT)

Bit	Name	Reset value	Type	Description
Bit 15: 8	Reserved	0x0	resd	Kept at default value.
Bit 7	BRKSWTR	0x0	wo	Brake event triggered by software This bit is set by software to generate a brake event. 0: No effect 1: Generate a brake event.
Bit 6	TRGSWTR	0x0	rw	Trigger event triggered by software This bit is set by software to generate a trigger event. 0: No effect 1: Generate a trigger event.
Bit 5	HALLSWTR	0x0	wo	HALL event triggered by software This bit is set by software to generate a HALL event. 0: No effect 1: Generate a HALL event. Note: This bit acts only on channels that have complementary output.
Bit 4	C4SWTR	0x0	wo	Channel 4 event triggered by software Please refer to C1M description.
Bit 3	C3SWTR	0x0	wo	Channel 3 event triggered by software Please refer to C1M description.
Bit 2	C2SWTR	0x0	wo	Channel 2 event triggered by software Please refer to C1M description
Bit 1	C1SWTR	0x0	wo	Channel 1 event triggered by software This bit is set by software to generate a channel 1 event. 0: No effect 1: Generate a channel 1 event.
Bit 0	OVFSWTR	0x0	wo	Overflow event triggered by software This bit is set by software to generate an overflow event. 0: No effect 1: Generate an overflow event.

14.6.4.7 TMR1 channel mode register1 (TMR1_CM1)

The channel can be used in input (capture mode) or output (compare mode). The direction of a channel is defined by the corresponding CxC bits. All the other bits of this register have different functions in input and output modes. The CxOx describes its function in output mode when the channel is in output mode, while the CxIx describes its function in output mode when the channel is in input mode. Attention must be given to the fact that the same bit can have different functions in input mode and output mode.

Output compare mode:

Bit	Name	Reset value	Type	Description
Bit 15	C2OSEN	0x0	rw	Channel 2 output switch enable
Bit 14: 12	C2OCTRL	0x0	rw	Channel 2 output control
Bit 11	C2OBEN	0x0	rw	Channel 2 output buffer enable
Bit 10	C2OIEN	0x0	rw	Channel 2 output enable immediately
Bit 9: 8	C2C	0x0	rw	Channel 2 configuration

				This field is used to define the direction of the channel 2 (input or output), and the selection of input pin when C2EN='0': 00: Output 01: Input, C2IN is mapped on C2IFP2 10: Input, C2IN is mapped on C1IFP2 11: Input, C2IN is mapped on STCI. This mode works only when the internal trigger input is selected by STIS register.
Bit 7	C1OSEN	0x0	rw	Channel 1 output switch enable 0: C1ORAW is not affected by EXT input. 1: Once a high level is detected on EXT input, clear C1ORAW.
Bit 6: 4	C1OCTRL	0x0	rw	Channel 1 output control This field defines the behavior of the original signal C1ORAW. 000: Disconnected. C1ORAW is disconnected from C1OUT; 001: C1ORAW is high when TMR1_CVAL=TMR1_C1DT 010: C1ORAW is low when TMR1_CVAL=TMR1_C1DT 011: Switch C1ORAW level when TMR1_CVAL=TMR1_C1DT 100: C1ORAW is forced low 101: C1ORAW is forced high. 110: PWM mode A — OWCDIR=0, C1ORAW is high once TMR1_C1DT>TMR1_CVAL, else low; — OWCDIR=1, C1ORAW is low once TMR1_C1DT <TMR1_CVAL, else high; 111: PWM mode B — OWCDIR=0, C1ORAW is low once TMR1_C1DT >TMR1_CVAL, else high; — OWCDIR=1, C1ORAW is high once TMR1_C1DT <TMR1_CVAL, else low. *Note: In the configurations other than 000', the C1OUT is connected to C1ORAW. The C1OUT output level is not only subject to the changes of C1ORAW, but also the output polarity set by CCTRL.*
Bit 3	C1OBEN	0x0	rw	Channel 1 output buffer enable 0: Buffer function of TMR1_C1DT is disabled. The new value written to the TMR1_C1DT takes effect immediately. 1: Buffer function of TMR1_C1DT is enabled. The value to be written to the TMR1_C1DT is stored in the buffer register, and can be sent to the TMR1_C1DT register only on an overflow event.
Bit 2	C1OIEN	0x0	rw	Channel 1 output enable immediately In PWM mode A or B, this bit is used to accelerate the channel 1 output's response to the trigger event. 0: Need to compare the CVAL with C1DT before generating an output 1: No need to compare the CVAL and C1DT. An output is generated immediately when a trigger event occurs.
Bit 1: 0	C1C	0x0	rw	Channel 1 configuration This field is used to define the direction of the channel 1 (input or output), and the selection of input pin when C1EN='0': 00: Output 01: Input, C1IN is mapped on C1IFP1 10: Input, C1IN is mapped on C2IFP1 11: Input, C1IN is mapped on STCI. This mode works only when the internal trigger input is selected by STIS.

Input capture mode:

Bit	Name	Reset value	Type	Description
Bit 15: 12	C2DF	0x0	rw	Channel 2 digital filter
Bit 11: 10	C2IDIV	0x0	rw	Channel 2 input divider
Bit 9: 8	C2C	0x0	rw	Channel 2 configuration This field is used to define the direction of the channel 2 (input or output), and the selection of input pin when C2EN='0': 00: Output 01: Input, C2IN is mapped on C2IFP2 10: Input, C2IN is mapped on C1IFP2 11: Input, C2IN is mapped on STCI. This mode works only when the internal trigger input is selected by STIS.
Bit 7: 4	C1DF	0x0	rw	Channel 1 digital filter This field defines the digital filter of the channel 1. N stands for the number of filtering, indicating that the input edge can pass the filter only after N sampling events. 0000: No filter, sampling is done at f_{DTS} 1000: $f_{SAMPLING}=f_{DTS}/8$, N=6 0001: $f_{SAMPLING}=f_{CK\_INT}$, N=2 1001: $f_{SAMPLING}=f_{DTS}/8$, N=8 0010: $f_{SAMPLING}=f_{CK\_INT}$, N=4 1010: $f_{SAMPLING}=f_{DTS}/16$, N=5 0011: $f_{SAMPLING}=f_{CK\_INT}$, N=8 1011: $f_{SAMPLING}=f_{DTS}/16$, N=6 0100: $f_{SAMPLING}=f_{DTS}/2$, N=6 1100: $f_{SAMPLING}=f_{DTS}/16$, N=8 0101: $f_{SAMPLING}=f_{DTS}/2$, N=8 1101: $f_{SAMPLING}=f_{DTS}/32$, N=5 0110: $f_{SMPLING}=f_{DTS}/4$, N=6 1110: $f_{SAMPLING}=f_{DTS}/32$, N=6 0111: $f_{SAMPLING}=f_{DTS}/4$, N=8 1111: $f_{SAMPLING}=f_{DTS}/32$, N=8
Bit 3: 2	C1IDIV	0x0	rw	Channel 1 input divider This field defines Channel 1 input divider. 00: No divider. An input capture is generated at each active edge. 01: An input compare is generated every 2 active edges 10: An input compare is generated every 4 active edges 11: An input compare is generated every 8 active edges Note: the divider is reset once C1EN='0'
Bit 1: 0	C1C	0x0	rw	Channel 1 configuration This field is used to define the direction of the channel 1 (input or output), and the selection of input pin when C1EN='0': 00: Output 01: Input, C1IN is mapped on C1IFP1 10: Input, C1IN is mapped on C2IFP1 11: Input, C1IN is mapped on STCI. This mode works only when the internal trigger input is selected by STIS.

14.6.4.8 TMR1 channel mode register2 (TMR1_CM2)

The channel can be used in input (capture mode) or output (compare mode). The direction of a channel is defined by the corresponding CxC bits. All the other bits of this register have different functions in input and output modes. The CxOx describes its function in output mode when the channel is in output mode, while the CxIx describes its function in output mode when the channel is in input mode. Attention must be given to the fact that the same bit can have different functions in input mode and output mode.

Output compare mode:

Bit	Name	Reset value	Type	Description
Bit 15	C4OSEN	0x0	rw	Channel 4 output switch enable
Bit 14: 12	C4OCTRL	0x0	rw	Channel 4 output control
Bit 11	C4OBEN	0x0	rw	Channel 4 output buffer enable

Bit 10	C4OIEN	0x0	rw	Channel 4 output enable immediately
Bit 9: 8	C4C	0x0	rw	Channel 4 configuration This field is used to define the direction of the channel 1 (input or output), and the selection of input pin when C4EN='0': 00: Output 01: Input, C4IN is mapped on C4IFP4 10: Input, C4IN is mapped on C3IFP4 11: Input, C4IN is mapped on STCI. This mode works only when the internal trigger input is selected by STIS.
Bit 7	C3OSEN	0x0	rw	Channel 3 output switch enable
Bit 6: 4	C3OCTRL	0x0	rw	Channel 3 output control
Bit 3	C3OBEN	0x0	rw	Channel 3 output buffer enable
Bit 2	C3OIEN	0x0	rw	Channel 3 output enable immediately
Bit 1: 0	C3C	0x0	rw	Channel 3 configuration This field is used to define the direction of the channel 1 (input or output), and the selection of input pin when C3EN='0': 00: Output 01: Input, C3IN is mapped on C3IFP3 10: Input, C3IN is mapped on C4IFP3 11: Input, C3IN is mapped on STCI. This mode works only when the internal trigger input is selected by STIS.

Input capture mode:

Bit	Name	Reset value	Type	Description
Bit 15: 12	C4DF	0x0	rw	Channel 4 digital filter
Bit 11: 10	C4IDIV	0x0	rw	Channel 4 input divider
Bit 9: 8	C4C	0x0	rw	Channel 4 configuration This field is used to define the direction of the channel 1 (input or output), and the selection of input pin when C4EN='0': 00: Output 01: Input, C4IN is mapped on C4IFP4 10: Input, C4IN is mapped on C3IFP4 11: Input, C4IN is mapped on STCI. This mode works only when the internal trigger input is selected by STIS.
Bit 7: 4	C3DF	0x0	rw	Channel 3 digital filter
Bit 3: 2	C3IDIV	0x0	rw	Channel 3 input divider
Bit 1:0	C3C	0x0	rw	Channel 3 configuration This field is used to define the direction of the channel 1 (input or output), and the selection of input pin when C3EN='0': 00: Output 01: Input, C3IN is mapped on C3IFP3 10: Input, C3IN is mapped on C4IFP3 11: Input, C3IN is mapped on STCI. This mode works only when the internal trigger input is selected by STIS.

14.6.4.9 TMR1 channel control register (TMR1_CCTRL)

Bit	Name	Reset value	Type	Description
Bit 15: 14	Reserved	0x0	resd	Kept at default value.
Bit 13	C4P	0x0	rw	Channel 4 polarity Please refer to C1P description.
Bit 12	C4EN	0x0	rw	Channel 4 enable Please refer to C1EN description.
Bit 11	C3CP	0x0	rw	Channel 3 complementary polarity Please refer to C1P description.
Bit 10	C3CEN	0x0	rw	Channel 3 complementary enable Please refer to C1EN description.
Bit 9	C3P	0x0	rw	Channel 3 polarity Please refer to C1P description.
Bit 8	C3EN	0x0	rw	Channel 3 enable Please refer to C1EN description.

Bit 7	C2CP	0x0	rw	Channel 2 complementary polarity Please refer to C1P description.
Bit 6	C2CEN	0x0	rw	Channel 2 complementary enable Please refer to C1EN description.
Bit 5	C2P	0x0	rw	Channel 2 polarity Please refer to C1P description.
Bit 4	C2EN	0x0	rw	Channel 2 enable Please refer to C1EN description.
Bit 3	C1CP	0x0	rw	Channel 1 complementary polarity 0: C1COUT is active high. 1: C1COUT is active low.
Bit 2	C1CEN	0x0	rw	Channel 1 complementary enable 0: Output is disabled. 1: Output is enabled.
Bit 1	C1P	0x0	rw	Channel 1 polarity When the channel 1 is configured in output mode: 0: C1OUT is active high 1: C1OUT is active low When the channel 1 is configured in input mode: The active edge of the input signal is defined by C1CP/C1P. 00: C1IN active edge is on its rising edge. When used as external trigger, C1IN is not inverted. 01: C1IN active edge is on its falling edge. When used as external trigger, C1IN is inverted. 10: Reserved 11: C1IN active edge is on its falling edge and rising edge. When used as external trigger, C1IN is not inverted.
Bit0	C1EN	0x0	rw	Channel 1 enable 0: Input or output is disabled 1: Input or output is enabled

Table 14-17 Complementary output channel CxOUT and CxCOUT control bits with brake function

Control bit					Output state (1)	
OEN bit	FCSODIS bit	FCSOEN bit	CxEN bit	CxCEN bit	CxOUT output state	CxCOUT output state
1	X	0	0	0	Output disabled (no driven by the timer) CxOUT=0, Cx_EN=0	Output disabled (no driven by the timer) CxCOUT=0, CxCEN=0
		0	0	1	Output disabled (no driven by the timer) CxOUT=0, Cx_EN=0	CxORAW + polarity, CxCOUT= CxORAW xor CxCP, CxCEN=1
		0	1	0	CxORAW+ polarity CxOUT= CxORAW xor CxP, Cx_EN=1	Output disabled (no driven by the timer) CxCOUT=0, CxCEN=0
		0	1	1	CxORAW+polarity+dead-time, Cx_EN=1	CxORAW inverted+polarity+dead-time, CxCEN=1
		1	0	0	Output disabled (no driven by the timer) CxOUT=CxP, Cx_EN=0	Output disabled (no driven by the timer) CxCOUT=CxCP, CxCEN=0
		1	0	1	Off-state (Output enabled with inactive level) CxOUT=CxP, Cx_EN=1	CxORAW + polarity, CxCOUT= CxORAW xor CxCP, CxCEN=1

		1	1	0	CxORAW + polarity, CxOUT= CxORAW xor CxP, Cx_EN=1	Off-state (Output enabled with inactive level) CxCOUT=CxCP, CxCEN=1
		1	1	1	CxORAW+ polarity+dead-time, Cx_EN=1	CxORAW inverted+polarity+dead-time, CxCEN=1
	0		0	0	Output disabled (corresponding IO is not driven by the timer, IO floating) Asynchronously: CxOUT=CxP, Cx_EN=0, CxCOUT=CxCP, CxCEN=0; If the clock is present: after a dead-time, CxOUT=CxIOS, CxCOUT=CxCIOS, assuming that CxIOS and CxCIOS do not correspond to CxOUT and CxCOUT active level.	
	0		0	1		
	0		1	0		
	0		1	1		
0	1	X	0	0	CxEN=CxCEN=0: Output disabled (corresponding IO is not driven by the timer, IO floating) In other cases: Off-state (Output enabled with inactive level) Asynchronously: CxOUT =CxP, Cx_EN=1, CxCOUT=CxCP, CxCEN=1; If the clock is present: after a dead-time, CxOUT=CxIOS, CxCOUT=CxCIOS, assuming that CxIOS and CxCIOS do not correspond to CxOUT and CxCOUT active level.	
	1		0	1		
	1		1	0		
	1		1	1		

Note: If the two outputs of a channel are not used (CxEN = CxCEN = 0), CxIOS, CxCIOS, CxP and CxCP must be cleared.

Note: The state of the external I/O pins connected to the complementary CxOUT and CxCOUT channels depends on the CxOUT and CxCOUT channel state and the GPIO and the IOMUX registers.

14.6.4.10 TMR1 counter value (TMR1_CVAL)

Bit	Name	Reset value	Type	Description
Bit 15: 0	CVAL	0x0	rw	Counter value

14.6.4.11 TMR1 divider (TMR1_DIV)

Bit	Name	Reset value	Type	Description
Bit 15: 0	DIV	0x0	rw	Divider value The counter clock frequency fCK_CNT = fTMR_CLK /(DIV[15:0]+1) The value of this register is transferred to the actual prescaler register when an overflow event occurs.

14.6.4.12 TMR1 period register (TMR1_PR)

Bit	Name	Reset value	Type	Description
Bit 15: 0	PR	0x0	rw	Period value This defines the period value of the timer. The timer stops working when the period value is 0.

14.6.4.13 TMR1 repetition period register (TMR1_RPR)

Bit	Name	Reset value	Type	Description
Bit 15: 0	RPR	0x0	rw	Repetition of period value This field is used to reduce the generation rate of overflow events. An overflow event is generated when the repetition counter reaches 0.

14.6.4.14 TMR1 channel 1 data register (TMR1_C1DT)

Bit	Name	Reset value	Type	Description
Bit 15: 0	C1DT	0x0	rw	Channel 1 data register When the channel 1 is configured as input mode:

				The C1DT is the CVAL value stored by the last channel 1 input event (C1IN) When the channel 1 is configured as output mode: C1DT is the value to be compared with the CVAL value. Whether the written value takes effective immediately depends on the C1OBEN bit, and the corresponding output is generated on C1OUT as configured.

14.6.4.15 TMR1 channel 2 data register (TMR1_C2DT)

Bit	Name	Reset value	Type	Description
Bit 15: 0	C2DT	0x0	rw	Channel 2 data register When the channel 2 is configured as input mode: The C2DT is the CVAL value stored by the last channel 2 input event (C1IN) When the channel 2 is configured as output mode: C2DT is the value to be compared with the CVAL value. Whether the written value takes effective immediately depends on the C2OBEN bit, and the corresponding output is generated on C2OUT as configured.

14.6.4.16 TMR1 channel 3 data register (TMR1_C3DT)

Bit	Name	Reset value	Type	Description
Bit 15: 0	C3DT	0x0	rw	Channel 3 data register When the channel 3 is configured as input mode: The C3DT is the CVAL value stored by the last channel 3 input event (C1IN) When the channel 3 is configured as output mode: C3DT is the value to be compared with the CVAL value. Whether the written value takes effective immediately depends on the C3OBEN bit, and the corresponding output is generated on C3OUT as configured.

14.6.4.17 TMR1 channel 4 data register (TMRx_C4DT)

Bit	Name	Reset value	Type	Description
Bit 15: 0	C4DT	0x0	rw	Channel 4 data register When the channel 4 is configured as input mode: The C4DT is the CVAL value stored by the last channel 4 input event (C1IN) When the channel 3 is configured as output mode: C4DT is the value to be compared with the CVAL value. Whether the written value takes effective immediately depends on the C4OBEN bit, and the corresponding output is generated on C4OUT as configured.

14.6.4.18 TMR1 brake register (TMR1_BRK)

Bit	Name	Reset value	Type	Description
Bit 19: 16	BKF	0x0	rw	Brake input filter This field is used to set the filter for brake input. The filter number N indicates that the input edge can pass through filter only after N sampling events. 0000: $f_{SAMPLING}=f_{DTS}$ (no filter) 1000: $f_{SAMPLING}=f_{DTS}/8$, N=6 0001: $f_{SAMPLING}=f_{CK\_INT}$, N=2 1001: $f_{SAMPLING}= f_{DTS}/8$, N=8 0010: $f_{SAMPLING}=f_{CK\_INT}$, N=4 1010: $f_{SAMPLING}=f_{DTS}/16$, N=5 0011: $f_{SAMPLING}=f_{CK\_INT}$, N=8 1011: $f_{SAMPLING}=f_{DTS}/16$, N=6 0100: $f_{SAMPLING}=f_{DTS}/2$, N=6 1100: $f_{SAMPLING}=f_{DTS}/16$, N=8 0101: $f_{SAMPLING}=f_{DTS}/2$, N=8 1101: $f_{SAMPLING}=f_{DTS}/32$, N=5

				0110: $f_{SAMPLING}=f_{DTS}/4$, N=6 1110: $f_{SAMPLING}=f_{DTS}/32$, N=6 0111: $f_{SAMPLING}=f_{DTS}/4$, N=8 1111: $f_{SAMPLING}=f_{DTS}/32$, N=8
Bit 15	OEN	0x0	rw	Output enable This bit acts on the channels as output. It is used to enable CxOUT and CxCOUT outputs. 0: Disabled 1: Enabled
Bit 14	AOEN	0x0	rw	Automatic output enable OEN is set automatically at an overflow event. 0: Disabled 1: Enabled
Bit 13	BRKV	0x0	rw	Brake input validity This bit is used to select the active level of a brake input. 0: Brake input is active low. 1: Brake input is active high.
Bit 12	BRKEN	0x0	rw	Brake enable This bit is used to enable brake input. 0: Brake input is disabled. 1: Brake input is enabled.
Bit 11	FCSOEN	0x0	rw	Frozen channel status when holistic output enable This bit acts on the channels that have complementary output. It is used to set the channel state when the timer is inactive and OEN=1. 0: CxOUT/CxCOUT outputs are disabled. 1: CxOUT/CxCOUT outputs are enabled. Output inactive level.
Bit 10	FCSODIS	0x0	rw	Frozen channel status when holistic output disable This bit acts on the channels that have complementary output. It is used to set the channel state when the timer is inactive and OEN=0. 0: CxOUT/CxCOUT outputs are disabled. 1: CxOUT/CxCOUT outputs are enabled. Output idle level.
Bit 9: 8	WPC	0x0	rw	Write protection configuration This field is used to enable write protection. 00: Write protection is OFF. 01: Write protection level 3, and the following bits are write protected: TMR1_STOP: DTC, STPEN, STPV and HOAEN TMR1_CTRL2: CxIOS and CxCIOS 10: Write protection level 2. The following bits and all bits in level 3 are write protected: TMR1_CCTRL: CxP and CxCP TMR1_STOP: FCSODIS and FCSOEN 11: Write protection level 1. The following bits and all bits in level 2 are write protected: TMR1_CMx: C2OCTRL and C2OBEN Note: When WPC>0, its content remains frozen until the next system reset.
Bit 7: 0	DTC	0x00	rw	Dead-time configuration This field defines the duration of the dead-time insertion. The 3-bit MSB of DTC[7: 0] is used for function selection: 0xx: DT = DTC [7: 0] * TDTS 10x: DT = (64+ DTC [5: 0]) * TDTS * 2 110: DT = (32+ DTC [4: 0]) * TDTS * 8 111: DT = (32+ DTC [4: 0]) * TDTS * 16

Note: Based on lock configuration, AOEN, BRKV, BRKEN, FCSODIS, FCSOEN and DTC[7:0] can all be write protected. Thus it is necessary to configure write protection when writing to the TMRx_BRK register for the first time.

14.6.4.19 TMR1 DMA control register (TMR1_DMACTRL)

Bit	Name	Reset value	Type	Description
Bit 15:13	Reserved	0x0	resd	Kept at default value.
Bit 12:8	DTB	0x0	rw	DMA transfer bytes This field defines the number of DMA transfers: 00000: 1 byte 00001: 2 bytes 00010: 3 bytes 00011: 4 bytes .... 10000: 17 bytes 10001: 18 bytes
Bit 7:5	Reserved	0x0	resd	Kept at default value.
Bit 4: 0	ADDR	0x0	rw	DMA transfer address offset ADDR is defined as an offset starting from the address of the TMR1_CTRL1 register: 00000: TMR1_CTRL1 00001: TMR1_CTRL2 00010: TMR1_STCTRL .....

14.6.4.20 TMR1 DMA data register (TMR1_DMADT)

Bit	Name	Reset value	Type	Description
Bit 15: 0	DMADT	0x0	rw	DMA data register A write/read operation to the DMADT register accesses any TMR register located at the following address: TMR1 peripheral address + ADDR*4 to TMR1 peripheral address + ADDR*4 + DTB*4

14.6.4.21 TMR1 channel mode register3 (TMR1_ CM3)

Bit	Name	Reset value	Type	Description
Bit 15: 6	Reserved	0x0	resd	Kept at default value.
Bit 7	C5OSEN	0x0	rw	Channel 5 output switch enable
Bit 6: 4	C5OCTRL	0x0	rw	Channel 5 output control
Bit 3	C5OBEN	0x0	rw	Channel 5 output buffer enable
Bit 2	C5OIEN	0x0	rw	Channel 5 output immediately enable
Bit 1: 0	Reserved	0x0	resd	Kept at default value.

14.6.4.22 TMR1 channel 5 data register (TMR1_C5DT)

Bit	Name	Reset value	Type	Description
Bit 15: 0	C5DT	0x0	rw	Channel 5 data register C5DT holds the value that is to be compared with the CVAL. Whether the written data will takes effect immediately depends on the C5OBEN bit, and the corresponding output generates on the C5OUT bit.

15 Window watchdog timer (WWDT)

15.1 WWDT introduction

The window watchdog downcounter must be reloaded within a limited time to prevent the watchdog circuits from generating a system reset. The window watch dog is used to detect the occurrence of system malfunctions.

The window watchdog timer is clocked by a divided APB1_CLK. The precision of the APB1_CLK enables the window watchdog to take accurate control of the limited time window.

15.2 WWDT main features

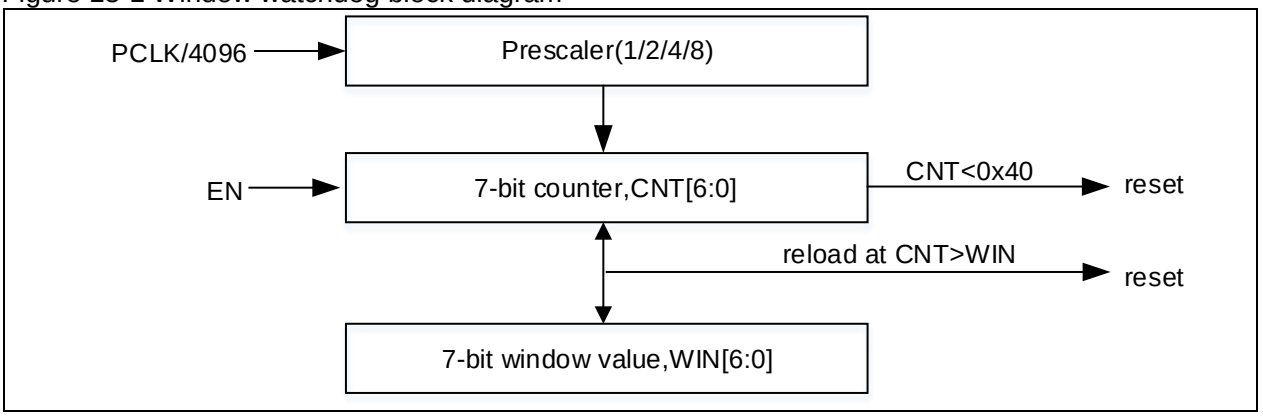
- 7-bit downcounter
- If the watchdog is enabled, a system reset is generated when the value of the downcounter is less than 0x40 or when the downcounter is reloaded outside the time window.
- The downcounter can be reloaded by enabling the reload counter interrupt.

15.3 WWDT functional overview

If the watchdog is enabled, a system reset is generated at the following conditions:

- When the 7-bit downcounter scrolls from 0x40 to 0x3F;
- When the 7-bit downcounter is greater than the 7-bit window value programmed.

Figure 15-1 Window watchdog block diagram



To prevent a system reset while reloading the counter value, the counter must be reloaded only when its value is less than the value stored in the window register and greater than 0x40.

The WWDT counter is clocked by a divided APB1_CLK, with the division factor being defined by the DIV[1: 0] bit in the WWDT_CFG register.

The counter value determines the maximum counter cycles before the watchdog generates a reset. It can be used together with the WIN[6: 0] bit to adjust the reload window.

WWDT offers reload counter interrupt feature. If enabled, the WWDT will set the RLDF flag when the counter value reaches 0x40h, and an interrupt is generated accordingly. The interrupt service routine (ISTS) can be used to reload the counter to prevent a system reset. Note that if CNT[6]=0, setting the WWDTEN=1 will generate a system reset, so the CNT[6] bit must be always set (CNT[6]=1) while writing to the WWDT_CTRL register to prevent the occurrence of an immediate reset once the window watchdog is enabled.

The formula to calculate the window watchdog time out:

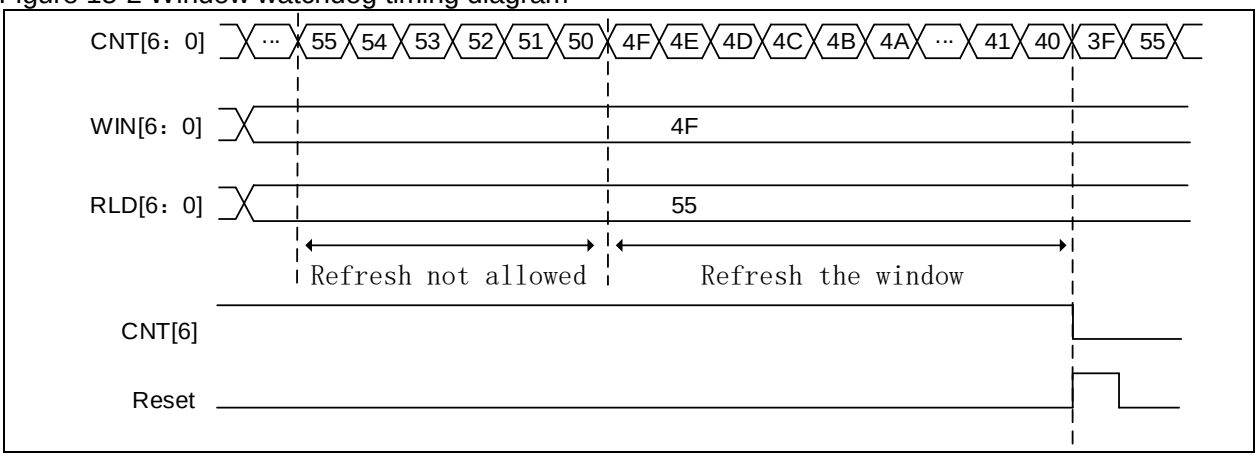
$$T_{WWDT} = T_{PCLK1} \times 4096 \times 2^{DIV[1:0]} \times (CNT[5:0] + 1); (ms)$$

Where: T_{PCLK1} refers to APB1 clock period, in ms.

Table 15-1 Minimum and maximum timeout value when PCLK1=80 MHz

Prescaler	Min. Timeout value	Max. Timeout value
0	51.2µs	3.28ms
1	102.4µs	6.55ms
2	204.8µs	13.11ms
3	409.6µs	26.21ms

Figure 15-2 Window watchdog timing diagram



15.4 Debug mode

When the microcontroller enters debug mode (Cortex®-M0+ core halted), the WWDT counter stops counting by setting the WWDT_PAUSE in the DEBUG module.

15.5 WWDT registers

These peripheral registers must be accessed by half words (16 bits) or words (32 bits).

Register	Offset	Reset value
WWDT_CTRL	0x00	0x7F
WWDT_CFG	0x04	0x7F
WWDT_STS	0x08	0x00

15.5.1 Control register (WWDT_CTRL)

Bit	Name	Reset value	Type	Description
Bit 31: 8	Reserved	0x000000	resd	Kept at default value.
Bit 7	WWDTEN	0x0	rw1s	Window watchdog enable 0: Disabled 1: Enabled This bit is set by software, but can be cleared only after reset.
Bit 6: 0	CNT	0x7F	rw	Downcounter When the counter counts down to 0x3F, a reset is generated.

15.5.2 Configuration register (WWDT_CFG)

Bit	Name	Reset value	Type	Description
Bit 31: 10	Reserved	0x000000	resd	Kept at default value.
Bit 9	RLDIEN	0x0	rw	Reload counter interrupt 0: Disabled 1: Enabled
Bit 8: 7	DIV	0x0	rw	Clock division value 00: PCLK1 divided by 4096 01: PCLK1 divided by 8192 10: PCLK1 divided by 16384 11: PCLK1 divided by 32768
Bit 6: 0	WIN	0x7F	rw	Window value If the counter is reloaded while its value is greater than the window register value, a reset will be generated. The counter must be reloaded between 0x40 and WIN[6: 0].

15.5.3 Status register (WWDT_STS)

Bit	Name	Reset value	Type	Description
Bit 31: 1	Reserved	0x0000 0000	resd	Kept at default value.
Bit 0	RLDF	0x0	rw0c	Reload counter interrupt flag This flag is set when the downcounter reaches 0x40. This bit is set by hardware and cleared by software.

16 Watchdog timer (WDT)

16.1 WDT introduction

The WDT is driven by a dedicated low-speed clock (LICK). Due to the lower clock accuracy of LICK, the WDT is best suited to the applications that have lower timing accuracy and can run independently outside the main application.

16.2 WDT main features

- 12-bit downcounter
- The counter is clocked by LICK (can work in Deepsleep mode)
- The counter can be configured to stop counting in Deepsleep or Standby mode
- A system reset is generated under the following circumstances:
  - When the counter value is decremented to 0
  - When the counter is reloaded outside the window

16.3 WDT functional overview

WDT enable:

Both software and hardware operations can be used to enable WDT. In other words, the WDT can be enabled by writing 0xCCCC to the WDT_CMD register; or when the user enables the hardware watchdog through user system data area, the WDT will be automatically enabled after power-on reset.

WDT reset:

When the counter value of the WDT counts down to 0, a WDT reset will be generated. Thus the WDT_CMD register must be written with the reload value 0xAAAA at a regular interval to avoid WDT-triggered reset. Besides, setting WIN[11:0]= 0xFFF (not default value) will enable window watchdog feature. In this case, reloading counter value while the counter value is greater than the window value will trigger a system reset.

WDT write-protected:

The WDT_DIV, WDT_RLD and WDT_WIN registers are write-protected. Writing the value 0x5555 to the WDT_CMD register will unlock write protection. The update status of these three registers are indicated by the DIVF, RLDF and WINF bits in the WDT_STS register. If a different value is written to the WDT_CMD register, these three registers will be write-protected again. Writing the value 0xAAAA to the WDT_CMD register also enables write protection.

WDT clock:

The WDT counter is clocked by LICK. The LICK is an internal RC clock in the range of 30kHz~60kHz. Therefore, the timeout period is also within a certain range, so a margin should be taken into account when configuring timeout period. The LICK can be calibrated to obtain a relatively accurate WDT timeout. For more information about LICK, see *Section 4.1.1*

WDT low power counting mode:

WDT can work in Sleep, Deepsleep and Standby modes. Users can choose to stop counting in Deepsleep and Standby modes by setting the nWDT_DEPSLP and nWDT_STDBY bits in the User System Data area.

If the counter is disabled, it will stop decrementing as soon as the Deepsleep and Standby modes are entered. This means that the WDT would not generate a system reset in both low power modes. After waking up from these two modes, it continues downcounting from the value at the time of the entry of these modes.

Figure 16-1 WDT block diagram

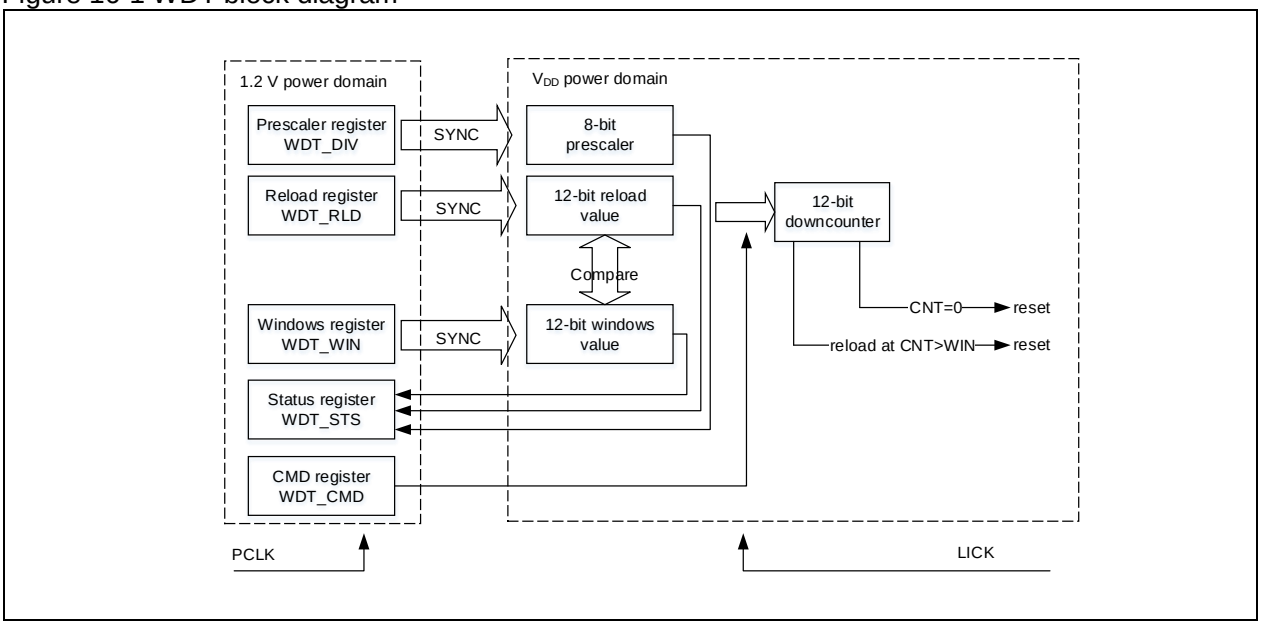


Table 16-1 WDT timeout period (LICK=40kHz)

Prescaler divider	DIV[2:0] bits	Min.timeout (ms) RLD[11:0] = 0x000	Max. timeout (ms) RLD[11:0] = 0xFFF
/4	0	0.1	409.6
/8	1	0.2	819.2
/16	2	0.4	1638.4
/32	3	0.8	3276.8
/64	4	1.6	6553.6
/128	5	3.2	13107.2
/256	(6 or 7)	6.4	26214.4

16.4 Debug mode

When the microcontroller enters debug mode (Cortex®-M0+ core halted), the WDT counter stops counting by setting the WDT_PAUSE in the DEBUG module.

16.5 WDT registers

These peripheral registers must be accessed by half-words (16 bits) or words (32 bits).

Register	Offset	Reset value
WDT_CMD	0x00	0x0000 0000
WDT_DIV	0x04	0x0000 0000
WDT_RLD	0x08	0x0000 0FFF
WDT_STS	0x0C	0x0000 0000
WDT_WIN	0x10	0x0000 0FFF

16.5.1 Command register (WDT_CMD)

(Reset in Standby mode)

Bit	Name	Reset value	Type	Description
Bit 31: 16	Reserved	0x0000	resd	Kept at default value.
Bit 15: 0	CMD	0x0000	wo	Command register 0xAAAA: Reload counter 0x5555: Unlock the write-protected WDT_DIV, WDT_RLD and WDT_WIN 0xCCCC: Enable WDT. If the hardware watchdog has been enabled, ignore this operation.

16.5.2 Divider register (WDT_DIV)

(Not reset in Standby mode)

Bit	Name	Reset value	Type	Description
Bit 31: 3	Reserved	0x0000 0000	resd	Kept at default value.
Bit 2: 0	DIV	0x0	rw	Clock division value 000: LICK divided by 4 001: LICK divided by 8 010: LICK divided by 16 011: LICK divided by 32 100: LICK divided by 64 101: LICK divided by 128 110: LICK divided by 256 111: LICK divided by 256 The write protection must be unlocked in order to enable write access to the register. The register can be read only when DIVF=0.

16.5.3 Reload register (WDT_RLD)

(Not reset in Standby mode)

Bit	Name	Reset value	Type	Description
Bit 31: 12	Reserved	0x00000	resd	Kept at default value.
Bit 11: 0	RLD	0xFFF	rw	Reload value The write protection must be unlocked in order to enable write access to the register. The register can be read only when RLDF=0.

16.5.4 Status register (WDT_STS)

(Reset in Standby mode)

Bit	Name	Reset value	Type	Description
Bit 31: 3	Reserved	0x00000000	resd	Kept at default value.
Bit 2	WINF	0x0	ro	Window value update complete flag 0: Window value update complete 1: Window value update is in process. The WDT_WIN register can be written only when RLDF=0
Bit 2	RLDF	0x0	ro	Reload value update complete flag 0: Reload value update complete 1: Reload value update is in process. The reload register WDT_RLD can be written only when RLDF=0
Bit 0	DIVF	0x0	ro	Division value update complete flag 0: Division value update complete 1: Division value update is in process. The divider register WDT_DIV can be written only when DIVF=0

16.5.5 Window register (WDT_WIN)

(Not reset in Standby mode)

Bit	Name	Reset value	Type	Description
Bit 31: 12	Reserved	0x000000	resd	Kept at default value.
Bit 11: 0	WIN	0xFFF	ro	Window value When the counter value is greater than the window value, reloading the counter will trigger a reset. The reload counter value falls between 0 and the window value.

17 Enhanced real-time clock (ERTC)

17.1 ERTC introduction

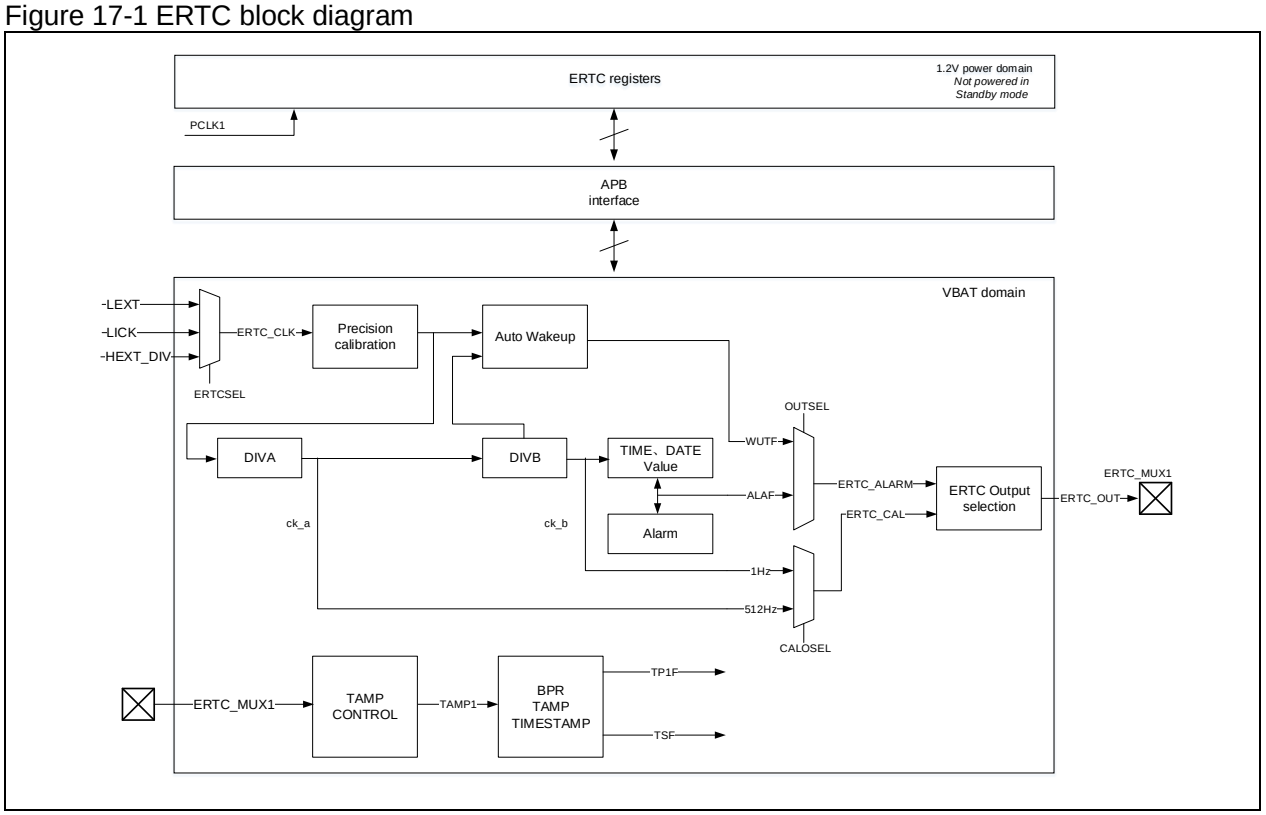
The real-time clock provides a calendar clock function. The time and date can be changed by modifying the ERTC_TIME and ERTC_DATE register.

The ERTC module is in battery powered domain, which means that it keeps running and free from the influence of system reset and VDD power off as long as VBAT is powered.

17.2 ERTC main features

- Real-time calendar, one alarm. Compensations for 28-, 29- (leap year), 30-, and 31-day months are performed. When the year register is a multiple of 4, it represents a leap year, with clock support
- Periodic auto-wakeup
- Reference clock detection
- 1x programmable tamper detection supporting time stamp feature
- Supports fine calibration
- 5 x battery powered registers
- 4 x interrupts: alarm A, periodic auto-wakeup, tamper detection and time stamp
- Multiplexed function output, calibration clock output, alarm event or wakeup event
- Multiplexed function input, reference clock input, one-channel tamper detection and time stamp

Figure 17-1 ERTC block diagram



17.3 ERTC functional overview

17.3.1 ERTC clock

ERTC clock source (ERTC_CLK) is selected via clock controller from a LEXT, LICK, and divided-by-32 HEXT.

The ERTC embeds two dividers: A and B, defined by the DIVA[6: 0] and DIVB[14: 0] bits respectively. It is recommended that the DIVA is configured with a higher value in order to minimum power consumption. After being divided by dividers A and B, the ERTC_CLK generates ck_a and ck_b clocks, respectively. The ck_a is used for subsecond update, while the ck_b is used for calendar update and periodic auto wakeup. The clock frequencies of ck_a and ck_b are obtained from the following equation:

$$F_{ck\_a} = \frac{f_{ERTC\_CLK}}{DIVA + 1}$$

$$F_{ck\_b} = \frac{f_{ERTC\_CLK}}{(DIVB + 1) \times (DIVA + 1)}$$

To obtain ck_b with frequency of 1 Hz, DIVA=127, DIVB=255, and 32.768 kH LEXT should be used. This ck_b is then used for calendar update.

17.3.2 ERTC initialization

ERTC register write protection

After a power-on reset, all ERTC registers are write protected. Such protection mechanism is not affected by the system reset. Write access to the ERTC registers (except the ERTC_STS[14: 8], ERTC_TAMP and ERTC_BPRx registers) can be enabled by unlocking write protection.

To unlock write protection of ERTC registers, the steps below should be respected:

1. Enable power interface clock by setting PWCEN=1 in the CRM_APB1EN register
2. Unlock write protection of battery powered domain by setting BPWEN=1 in the PWC_CTRL register
3. Write 0xCA and 0x53 to the ERTC_WP register in sequence. Writing an incorrect key will activate the write protection again.

Table 17-1 lists the ERTC registers that can be configured only after the write protection is unlocked and after the initialization mode is entered.

Table 17-1 RTC register map and reset values

Register	ERTC_WP enabled	Whether to enter initialization mode	Others
ERTC_TIME	Y	Y	-
ERTC_DATE	Y	Y	-
ERTC_CTRL	Y	Bit 6, and 4 only	-
ERTC_STS	Y, except [13: 8]	-	-
ERTC_DIV	Y	Y	-
ERTC_WAT	Y	N	Configurable when WATWF=1
ERTC_ALA	Y	N	Configurable when ALAWF =1
ERTC_WP	-	-	-
ERTC_SBS	-	-	-
ERTC_TADJ	Y	N	Configurable when TADJF=0
ERTC_TSTM	-	-	-
ERTC_TSDT	-	-	-

ERTC_TSSBS	-	-	-
ERTC_SCAL	Y	N	Configurable when CALUPDF=0
ERTC_TAMP	N	N	-
ERTC_ALASBS	Y	N	Configurable when ALAWF =1
ERTC_BPRx	N	N	-

Clock and calendar initialization

After the register write protection is unlocked, follow the procedures below for clock and calendar initialization:

1. Enter initialization mode by setting IMEN =1.
2. Wait until the initialization flag INITF bit is set to 1.
3. Set DIVB and DIVA.
4. Set the clock and calendar values.
5. Leave the initialization mode by clearing the IMEN bit. Wait until the UPDF bit is set to 1, indicating the completion of the calendar update. The calendar starts counting.

The ERTC also allows the fine-tuning for daylight saving time and clock.

Daylight saving time feature: It is used to increase (ADD1H=1) or decrease (DEC1H=1) one hour in the calendar, without the need of another initialization process.

Clock calibration: It is used for the fine calibration of the current clock. If only DECSBS[14: 0] is configured, the value will be added to the DIVB counter and a clock latency will be generated. If only ADD1S=1 is activated, the current clock will increase by one second. If both DECSBS[14: 0] and ADDIS bit are configured, the clock will increase by a fraction of a second.

Time latency (ADD1S=0): DECSBS/(DIVB+1)

Time advance (ADD1S=1): 1-(DECSBS/(DIVB+1)

Note: To avoid subsecond overflow, SBS[15]=0 must be asserted before setting the ERTC_TADJ register. Clock adjustment and clock synchronization can not be made at the same time. When RCD=1, clock adjustment is not supported.

Reading the calendar

The ERTC offers two different ways to read the calendar, namely, synchronous read (DREN=0) and asynchronous read (DREN=1).

DREN=0: The clock and calendar values can be obtained by reading the synchronous shadow register via the PCLK1. The UPDF bit is set each time the shadow register is synchronized with the ERTC calendar value located in the battery powered domain. The synchronization is performed every two ERTC_CLK cycles. The shadow register is reset by a system reset. To ensure consistency between the 3 values (ERTC_SBS, ERTC_TIME and ERTC_DATE registers), reading lower-order registers will lock the values in the higher-order registers until the ERTC_DATE register is read. For example, reading the ERTC_SBS register will lock the values in the ERTC_TIME and ERTC_DATE registers.

DREN=1, the ERTC will perform direct read access to the ERTC clock and calendar located in the battery powered domain with the PCLK1, avoiding the occurrence of errors caused by time synchronization. In this mode, the UPDF flag is cleared by hardware. To ensure data accuracy when reading clock and calendar, the software must read the clock and calendar registers twice, and compare the results of two read operations. If the result is not aligned, read again until that the results of two read accesses are matched. Besides, it is also possible to compare the least significant bits of the two read operations to determine their consistency.

Note: In Standby and Deepsleep modes, the current calendar values are not copied into the shadow register.. When waking up from these two modes, UPDF=0 must be asserted, and then wait until UPDF=1, to ensure that the latest calendar value can be read. In synchronous read (DREN=0) mode, the frequency of the PCLK1 must be at least seven times the ERTC_CLK frequency. In asynchronous

read (DREN=1), an additional APB cycle is required to complete the read operations of the calendar register.

Alarm clock initialization

The ERTC contains an alarm clock A, and relevant respective interrupt.

The alarm clock value is programmed with the ERTC_ALASBS and ERTC_ALA. When the programmed alarm value matches the calendar value, an alarm event is generated if an alarm clock is enabled. The MASKx bit can be used to selectively mask calendar fields. The calendar fields, which are masked, are not allocated with an alarm clock.

To configure the alarm clocks, the following steps should be respected:

1. Disable alarm clock A (by setting ALAEN=0);
2. Wait until the ALAWF bit is set to enable write access to the alarm clock A;
3. Configure alarm clock A registers (ERTC_ALA and ERTC_ALASBS);
4. Enable alarm clock A by setting ALAEN=1.

Note: If MASK1=0 in the ERTC_ALA register, the alarm clock can work normally only when the DIVB value is at least equal to 3.

17.3.3 Periodic automatic wakeup

Periodic automatic wakeup unit is used to wake up ERTC from low power mode automatically. The period is programmed with the VAL[15: 0] bi (When WATCLK[2]=1, it is extended to 17 bits, and the wakeup counter value is VAL+2^{16}). When the wakeup counter value drops from the VAL to zero, the WATF bit is set to 1, and a wakeup event is generated, with the wakeup counter being reloaded with the VAL value. An interrupt is also generated if a periodic wakeup interrupt is enabled.

The WATCLK[2: 0] bit can be used to select a wakeup timer clock, including ERTC_CLK/16, ERTC_CLK/8, ERTC_CLK/4, ERTC_CLK/2 or ck_b (usually 1Hz). The cooperation between wakeup timer clocks and wakeup counter values enable users to adjust the wakeup period freely.

To enable a periodic automatic wakeup, the following steps should be respected:

1. Disable a periodic automatic wakeup by setting WATEN=0;
2. Wait until WATWF=1 to enable write access to the wakeup reload timer and WATCLK[2: 0];
3. Configure the wakeup timer counter value and wakeup timer with VAL[15: 0] and WATCLK[2: 0] bits;
4. Enable a timer by setting WATEN=1.

Note: A wakeup timer is not affected by a system reset and low power consumption modes (Sleep, Deepsleep and Standby modes)

Note: In debug mode, if the ERTC_CLK is selected as wakeup clock, the counter which is used for periodic wakeup works normally.

17.3.4 ERTC calibration

Smooth digital calibration:

Smooth digital calibration has a higher and well-distributed performance than the coarse digital calibration. The calibration is performed by increasing or decreasing ERTC_CLK in an evenly manner.

The smooth digital calibration period is around 2^{20} ERTC_CLK (32 seconds) when the ERTC_CLK is 32.768 kHz. The DEC[8: 0] bit specifies the number of pulses to be masked during the 2^{20} ERTC_CLK cycles. A maximum of 511 pulses can be removed. When the ADD bit is set, 512 pulses can be inserted during the 2^{20} ERTC_CLK cycles. When DEC[8: 0] and ADD are sued together, a deviation ranging from -511 to +512 ERTC_CLK cycles can be added during the 2^{20} ERTC_CLK cycles.

The effective calibrated frequency (F_{SCAL}):

$$F_{SCAL} = F_{ERTC\_CLK} \times [\,1 + \frac{ADD \times 512 - DEC}{2^{20} + DEC - ADD \times 512}\,]$$

When the divider A is less than 3, the calibration operates as if ADD was equal to 0. The divider B value should be reduced so that each second is accelerated by 8 ERTC_CLK cycles, which means that 256 ERTC_CLK cycles are added every 32 seconds. When DEC[8: 0] and ADD are used together, a deviation ranging from -255 to +256 ERTC_CLK cycles can be added during the 2^{20} ERTC_CLK cycles.

At this point, the effective calibrated frequency (F_{SCAL})

$$F_{SCAL} = F_{ERTC\_CLK} \times [\ 1 + \frac{256 - DEC}{2^{20} + DEC - 256}\ ]$$

It is also possible to select 8 or 16-second digital calibration period through the CAL8 and CAL16 bits. The 8-second period takes priority over 16-second. In other words, when both 8-second and 16-second are enabled, 8-second calibration period prevails.

The CALUPDF flag in the ERTC indicates the calibration status. During the configuration of ERTC_SCAL register, the CALUPDF bit is set, indicating that the calibration value is being updated; Once the calibration value is successfully applied, this bit is cleared automatically, indicating the completion of the calibration value update.

17.3.5 Reference clock detection

The calendar update can be synchronized (not used in low-power modes) to a reference clock (usually the mains 50 or 60 Hz) with a higher precision. This reference clock is used to calibrate the precision of the calendar update frequency (1 Hz)

When it is enabled, the reference clock edge detection is performed during the first 7 ck_a periods around each of the calendar updates. When detected, the edge is used to update calendar values, and 3 ck_a periods are used for subsequent calendar updates. Each time the reference clock edge is detected, the divider A value is forced to reload, making the reference clock aligned with the 1 Hz clock. If the 1 Hz clock has a slight shift, a more accurate reference clock can be used to fine-tune the 1 Hz clock so that it is aligned with the reference clock. If no reference clock edge is detected, the calendar is updated based on ERTC's original clock source.

Note: Once the reference clock detection is enabled, the DIVA and DIVB must be kept at its respective reset value (0x7F and 0xFF respectively).

17.3.6 Time stamp function

When a time stamp event is detected on the tamper pin (valid edge is detected), the current calendar value will be stored to the time stamp register.

When a time stamp event occurs, the time stamp flag bit (TSF) will be set to 1 in the ERTC_STS register. If a new time stamp event is detected when time stamp flag (TSF) is already set, then the time stamp overflow flag (TSOF) will be set, but the time stamp registers will remain the result of the last event. By setting the TSIEN bit, an interrupt can be generated when a time stamp event occurs.

Usage of time stamp:

1. How to enable time stamp when a valid edge is detected on a tamper pin

- Select a rising edge or falling edge to trigger time stamp by setting the TSEDG bit
- Enable time stamp by setting TSEN=1

2. How to save time stamp on a tamper event

- Configure tamper detection registers
- Enable tamper detection time stamp by setting TPTSEN=1

Note: The TSF bit will be set to 1 two ck_a cycles after a time stamp event. It is suggested that users poll TSOF bit when the TSF is set.

17.3.7 Tamper detection

The ERTC has a tamper detection mode: TAMP1. It can be configured as a level detection with filter or edge detection. TAMP1 is mapped onto the ERTC_MUX1 pin.

The TP1F will be set to 1 when a valid tamper event is detected. An interrupt will also be generated if a tamper detection interrupt is enabled. If the TPTSEN bit is already set to 1, a time stamp event will be generated accordingly. Once a tamper event occurs, the battery powered registers will be reset so as to ensure data security in the battery powered domain.

How to configure edge detection

1. Select edge detection by setting TPFLT=00, and select a valid edge (TP1EDG)
2. According to actual needs, configure whether to activate a time stamp upon a tamer event (TPTSEN=1)

3. According to actual needs, enable a tamper detection interrupt (TPIEN=1)
4. Enable TAMP1 by setting TP1EN=1

How to configure level detection with filtering

1. Select level detection with filtering, and valid level sampling times (TPFLT≠00)
2. Select tamper detection valid level (TP1EDG)
3. Select tamper detection sampling frequency (TPFREQ\)
4. According to actual needs, enable tamper detection pull-up (setting TPPU=1). When TPPU=1 is asserted, tamper detection pre-charge time must be configured through the TPPR bit
5. According to actual needs, configure whether to activate a time stamp on a tamper event (TPTSEN=1)
6. According to actual needs, enable a tamper interrupt (TPIEN=1)
7. Enable TAMP1 by setting TP1EN=1

While setting edge detection mode, the following two points worth our attention:

1. If a rising edge is configured to enable tamper detection, and the tamper detection pin turns to high level before tamper detection is enabled, then a tamper event will be detected right after the tamper detection is enabled;
2. If a falling edge is configured to enable tamper detection, and the tamper detection pin turns to low level before tamper detection is enabled, then a tamper event will be detected right after the tamper detection is enabled;

17.3.8 Multiplexed function output

ERTC provides a set of multiplexed function outputs for the following events:

1. Clocks calibrated (OUTSEL=0 and CALOEN=1)
- Output 512Hz (CALOSEL=0)
- Output 1Hz (CALOSEL=1)

2. Alarm clock A (OUTSEL=1)
3. Wakeup event (OUTSEL=3)

When alarm clock or wakeup event is selected (OUTSEL≠0), it is possible to select output type (open-drain or push-pull) with the OUTTYPE bit, and output polarity with the OUTP bit.

17.3.9 ERTC wakeup

ERTC can be woken up by alarm clock, periodic auto wakeup, time stamp or tamper event. To enable an ERTC interrupt, follow the procedure below:

1. Configure the EXINT line corresponding to ERTC interrupts as an interrupt mode and enable it, and select a rising edge
2. Enable a NVIC channel corresponding to ERTC interrupts
3. Enable an ERTC interrupt

Table 17-2 lists the ERTC clock sources, events and interrupts that are able to wakeup low-power modes.

Table 17-2 ERTC low-power mode wakeup

Clock sources	Events	Wake up Sleep	Wake up Deepsleep	Wakeup Standby
HEXT	Alarm clock A	√	×	×
	Periodic automatic wakeup	√	×	×
	Time stamp	√	×	×
	Tamper event	√	×	×
LICK	Alarm clock A	√	√	√
	Periodic automatic wakeup	√	√	√
	Time stamp	√	√	√
	Tamper event	√	√	√
LEXT	Alarm clock A	√	√	√
	Periodic automatic wakeup	√	√	√
	Time stamp	√	√	√
	Tamper event	√	√	√

Table 17-3 Interrupt control bits

Interrupt events	Event flag	Interrupt enable bit	EXINT line
Alarm clock A	ALAF	ALAIEN	17
Periodic automatic wakeup	WATF	WATIEN	20
Time stamp	TSF	TSIEN	19
Tamper event	TP1F/TP2F	TPIEN	19

17.4 ERTC registers

These peripheral registers must be accessed by words (32 bits).

ERTC registers are 32-bit addressable registers.

Register	Offset	Reset value
ERTC_TIME	0x00	0x0000 0000
ERTC_DATE	0x04	0x0000 2101
ERTC_CTRL	0x08	0x0000 0000
ERTC_STS	0x0C	0x0000 0007
ERTC_DIV	0x10	0x007F 00FF
ERTC_WAT	0x14	0x0000 FFFF
ERTC_ALA	0x1C	0x0000 0000
ERTC_WP	0x24	0x0000 0000
ERTC_SBS	0x28	0x0000 0000
ERTC_TADJ	0x2C	0x0000 0000
ERTC_TSTM	0x30	0x0000 0000
ERTC_TSDT	0x34	0x0000 000D
ERTC_TSSBS	0x38	0x0000 0000
ERTC_SCAL	0x3C	0x0000 0000
ERTC_TAMP	0x40	0x0000 0000
ERTC_ALASBS	0x44	0x0000 0000
ERTC_BPRx	0x50-0x60	0x0000 0000

17.4.1 ERTC time register (ERTC_TIME)

Bit	Name	Reset value	Type	Description
Bit 31: 23	Reserved	0x000	resd	Kept at default value.
Bit 22	AMPM	0x0	rw	AM/PM 0: AM 1: PM Note: This bit is applicable for 12-hr format only. It is 0 for 24-hr format instead.
Bit 21: 20	HT	0x0	rw	Hour tens
Bit 19: 16	HU	0x0	rw	Hour units
Bit 15	Reserved	0x0	resd	Kept at its default value.
Bit 14: 12	MT	0x0	rw	Minute tens
Bit 11: 8	MU	0x0	rw	Minute units
Bit 7	Reserved	0x0	resd	Kept at default value.
Bit 6: 4	ST	0x0	rw	Second tens
Bit 3: 0	SU	0x0	rw	Second units

17.4.2 ERTC date register (ERTC_DATE)

Bit	Name	Reset value	Type	Description
Bit 31: 24	Reserved	0x00	resd	Kept at default value.
Bit 23: 20	YT	0x0	rw	Year tens
Bit 19: 16	YU	0x0	rw	Year units
Bit 15: 13	WK	0x1	rw	Week day 0: Forbidden 1: Monday 2: Tuesday 3: Wednesday 4: Thursday 5: Friday 6: Saturday 7: Sunday
Bit 12	MT	0x0	rw	Month tens
Bit 11: 8	MU	0x1	rw	Month units
Bit 7: 6	Reserved	0x0	resd	Kept at default value.
Bit 5: 4	DT	0x0	rw	Date tens
Bit 3: 0	DU	0x1	rw	Date units

17.4.3 ERTC control register (ERTC_CTRL)

Bit	Name	Reset value	Type	Description
Bit 31: 24	Reserved	0x00	resd	Kept at default value.
Bit 23	CALOEN	0x0	rw	Calibration output enable 0: Calibration output disabled 1: Calibration output enabled
Bit 22: 21	OUTSEL	0x0	rw	Output source selection 00: Output source disabled 01: Alarm clock A 10: Output source disabled 11: Wakeup event
Bit 20	OUTP	0x0	rw	Output polarity 0: High 1: Low
Bit 19	CALOSEL	0x0	rw	Calibration output selection 0: 512Hz 1: 1Hz
Bit 18	BPR	0x0	rw	Battery powered domain data register This bit in the battery powered domain is not affected by a system reset. It is used to store the daylight saving time change or others that need to be saved permanently.
Bit 17	DEC1H	0x0	wo	Decrease 1 hour 0: No effect 1: Subtract 1 hour Note: This bit is applicable only when the current hour is not 0. The next second takes effect when this bit is set (don't set this bit when the hour is being incremented)
Bit 16	ADD1H	0x0	wo	Add 1 hour 0: No effect 1: Add 1 hour Note: The next second takes effect when this bit is set (don't set this bit when the hour is being incremented)
Bit 15	TSIEN	0x0	rw	Timestamp interrupt enable 0: Timestamp interrupt disabled 1: Timestamp interrupt enabled
Bit 14	WATIEN	0x0	rw	Wakeup timer interrupt enable 0: Wakeup timer interrupt disable 1: Wakeup timer interrupt enabled

Bit 13	Reserved	0x0	resd	Kept at default value.
Bit 12	ALAIEN	0x0	rw	Alarm A interrupt enable 0: Alarm A interrupt disabled 1: Alarm A interrupt enabled
Bit 11	TSEN	0x0	rw	Timestamp enable 0: Timestamp disabled 1: Timestamp enabled
Bit 10	WATEN	0x0	rw	Wakeup timer enable 0: Wakeup timer disabled 1: Wakeup timer enabled
Bit 9	Reserved	0x0	resd	Kept at default value.
Bit 8	ALAEN	0x0	rw	Alarm A enable 0: Alarm A disabled 1: Alarm A enabled
Bit 7	Reserved	0x0	resd	Kept at default value.
Bit 6	HM	0x0	rw	Hour mode 0: 24-hour format 1: 12-hour format
Bit 5	DREN	0x0	rw	Date/time register direct read enable 0: Date/time register direct read disabled. ERTC_TIME, ERTC_DATE and ERTC_SBS values are taken from the synchronized registers, which are updated once every two ERTC_CLK cycles 1: Date/time register direct read enabled. ERTC_TIME, ERTC_DATE and ERTC_SBS values are taken from the battery powered domain.
Bit 4	RCDEN	0x0	rw	Reference clock detection enable 0: Reference clock detection disabled 1: Reference clock detection enabled
Bit 3	TSEDG	0x0	rw	Timestamp trigger edge 0: Rising edge 1: Falling edge
Bit 2: 0	WATCLK	0x0	rw	Wakeup timer clock selection 000: ERTC_CLK/16 001: ERTC_CLK/8 010: ERTC_CLK/4 011: ERTC_CLK/2 10x: ck_a 11x: ck_a is selected. 2^{16} is added to the wakeup counter value, and wakeup time =ERTC_WAT+2^{16}. *Note: The write access to this field is supported when WATEN=0 and WATWF=1.*

17.4.4 ERTC initialization and status register (ERTC_STS)

Bit	Name	Reset value	Type	Description
Bit 31: 17	Reserved	0x0000	resd	Kept at default value.
Bit 16	CALUPDF	0x0	ro	Calibration value update complete flag 0: Calibration value update is complete 1: Calibration value update is in progress This bit is automatically set when software writes to the ERTC_SCAL register. It is automatically cleared when a new calibration value is taking into account. When this bit is set, the write access to the ERTC_SCAL register is not allowed.
Bit 15:14	Reserved	0x0	resd	Kept at default value.
Bit 13	TP1F	0x0	rw0c	Tamper detection 1 flag 0: No tamper event 1: Tamper event occurred
Bit 12	TSOF	0x0	rw0c	Timestamp overflow flag 0: No timestamp overflow 1: Timestamp overflow occurs If a new time stamp event is detected when time stamp flag (TSF) is already set, this bit will be set by hardware.
Bit 11	TSF	0x0	rw0c	Timestamp flag

				0: No timestamp event 1: Timestamp event occurs It is recommended to double check the TSOF flag after reading a timestamp and clearing the TSF. Otherwise, a new timestamp event may be detected while clearing the TSF. *Note: The clearing operation of this bit takes effect after two APB_CLK cycles.*
Bit 10	WATF	0x0	rw0c	Wakeup timer flag 0: No wakeup timer event 1: Wakeup timer event occurs *Note: The clearing operation of this bit takes effect after two APB_CLK cycles.*
Bit 9	Reserved	0x0	resd	Kept at default value.
Bit 8	ALAF	0x0	rw0c	Alarm clock A flag 0: No alarm clock event 1: Alarm clock event occurred *Note: The clearing operation of this bit takes effect after two APB_CLK cycles.*
Bit 7	IMEN	0x0	rw	Initialization mode enable 0: Initialization mode disabled 1: Initialization mode enabled When an initialization mode is entered, the calendar stops running.
Bit 6	IMF	0x0	ro	Enter initialization mode flag 0: Initialization mode is not entered 1: Initialization mode is entered The ERTC_TIME, ERTC_DATE and ERTC_DIV registers can be modified only when an initialization mode is enabled (INITEN=1) and entered (INITEF=1).
Bit 5	UPDF	0x0	rw0c	Calendar update flag 0: Calendar update is in progress 1: Calendar update is complete The UPDF bit is set each time ERTC_TIME, ERTC_DATE and ERTC_SBS are synchronized with the ERTC calendar value located in the battery powered domain. The synchronization is performed every two ERTC_CLK cycles.
Bit 4	INITF	0x0	ro	Calendar initialization flag 0: Calendar has not been initialized 1: Calendar has been initialized This bit is set when the calendar year filed (ERTC_DATE) is different from 0. It is cleared when the year is 0.
Bit 3	TADJF	0x0	ro	Time adjustment flag 0: No time adjustment 1: Time adjustment is in progress This bit is automatically set when a write access to the ERTC_TADJ register is performed. It is automatically cleared at the end of time adjustment.
Bit 2	WATWF	0x1	ro	Wakeup timer register allows write flag 0: Wakeup timer register configuration not allowed 1: Wakeup timer register configuration allowed
Bit 1	Reserved	0x0	resd	Kept at default value.
Bit 0	ALAWF	0x1	ro	Alarm A register allows write flag 0: Alarm A register write operation not allowed 1: Alarm A register write operation allowed

17.4.5 ERTC divider register (ERTC_DIV)

Bit	Name	Reset value	Type	Description
Bit 31: 23	Reserved	0x000	resd	Kept at default value.
Bit 22: 16	DIVA	0x7F	rw	Divider A
Bit 15	Reserved	0x0	resd	Kept at default value.
Bit 14: 0	DIVB	0x00FF	rw	Divider B Calendar clock = ERTC_CLK/((DIVA+1)x(DIVB+1))

17.4.6 ERTC wakeup timer register (ERTC_WAT)

Bit	Name	Reset value	Type	Description
Bit 31: 16	Reserved	0x0000	resd	Kept at default value.
Bit 15: 0	VAL	0xFFFF	rw	Wakeup timer reload value

17.4.7 ERTC alarm clock A register (ERTC_ALA)

Bit	Name	Reset value	Type	Description
Bit 31	MASK4	0x0	rw	Date/week day mask 0: Date/week day is not masked 1: Alarm clock doesn't care about date/week day
Bit 30	WKSEL	0x0	rw	Date/week day select 0: Date 1: Week day (DT[1: 0] is not used)
Bit 29: 28	DT	0x0	rw	Date tens
Bit 27: 24	DU	0x0	rw	Date/week day units
Bit 23	MASK3	0x0	rw	Hour mask 0: No hour mask 1: Alarm clock doesn't care about hours
Bit 22	AMPM	0x0	rw	AM/PM 0: AM 1: PM *Note: This bit is applicable for 12-hour format only. It is 0 for 24-hour format.*
Bit 21: 20	HT	0x0	rw	Hour tens
Bit 19: 16	HU	0x0	rw	Hour units
Bit 15	MASK2	0x0	rw	Minute mask 0: No minute mask 1: Alarm clock doesn't care about minutes
Bit 14: 12	MT	0x0	rw	Minute tens
Bit 11: 8	MU	0x0	rw	Minute units
Bit 7	MASK1	0x0	rw	Second mask 0: No second mask 1: Alarm clock doesn't care about seconds
Bit 6: 4	ST	0x0	rw	Second tens
Bit 3: 0	SU	0x0	rw	Second units

17.4.8 ERTC write protection register (ERTC_WP)

Bit	Name	Reset value	Type	Description
Bit 31: 8	Reserved	0x000000	resd	Kept at default value
Bit 7: 0	CMD	0x00	wo	Command register All ERTC register write protection is unlocked by writing 0xCA and then 0x53. Writing any other value will re-activate write protection.

17.4.9 ERTC subsecond register (ERTC_SBS)

Bit	Name	Reset value	Type	Description
Bit 31: 16	Reserved	0x0000	resd	Kept at default value
Bit 15: 0	SBS	0x0000	ro	Sub-second value

				Subsecond is the value in the DIVB counter. Clock frequency = ERTC_CLK/(DIVA+1)

17.4.10 ERTC time adjustment register (ERTC_TADJ)

Bit	Name	Reset value	Type	Description
Bit 31	ADD1S	0x0	wo	Add 1 second 0: No effect 1: Add one second This bit can be written only when TADJF=0. It is intended to be used with DECSBS in order to fine-tune the time.
Bit 30: 15	Reserved	0x0000	resd	Kept at default value
Bit 14: 0	DECSBS	0x0000	wo	DECSBS[14: 0]: Decrease sub-second value Delay (ADD1S=0): Delay = DECSBS/(DIVB+1) Advance (ADD1S=1): Advance =1-(DECSBS/(DIVB+1))

17.4.11 ERTC time stamp time register (ERTC_TSTM)

Bit	Name	Reset value	Type	Description
Bit 31: 23	Reserved	0x000	resd	Kept at default value
Bit 22	AMPM	0x0	ro	AM/PM 0: AM 1: PM *Note: This bit is applicable for 12-hour format only. It is 0 for 24-hour format.*
Bit 21: 20	HT	0x0	ro	Hour tens
Bit 19: 16	HU	0x0	ro	Hour units
Bit 15	Reserved	0x0	resd	Kept at default value
Bit 14: 12	MT	0x0	ro	Minute tens
Bit 11： 8	MU	0x0	ro	Minute units
Bit 7	Reserved	0x0	resd	Kept at its default value
Bit 6: 4	ST	0x0	ro	Second tens
Bit 3: 0	SU	0x0	ro	Second units

Note: The content of this register is valid only when the TSF=1 in the ERTC_STS register. It is cleared when TSF bit is reset.

17.4.12 ERTC time stamp date register (ERTC_TSDT)

Bit	Name	Reset value	Type	Description
Bit 31: 16	Reserved	0x0000	resd	Kept at default value
Bit 15: 13	WK	0x0	ro	Week day
Bit 12	MT	0x0	ro	Month tens
Bit 11: 8	MU	0x0	ro	Month units
Bit 7: 6	Reserved	0x0	resd	Kept at default value
Bit 5: 4	DT	0x0	ro	Date tens
Bit 3: 0	DU	0x0	ro	Date units

Note: The content of this register is valid only when the TSF=1 in the ERTC_STS register. It is cleared when TSF bit is reset.

17.4.13 ERTC time stamp subsecond register (ERTC_TSSBS)

Bit	Name	Reset value	Type	Description
Bit 31: 16	Reserved	0x0000	resd	Kept at default value
Bit 15: 0	SBS	0x0000	ro	Sub-second value

Note: The content of this register is valid only when the TSF is set in the ERTC_STS register. It is cleared when TSF bit is reset.

17.4.14 ERTC smooth calibration register (ERTC_SCAL)

Bit	Name	Reset value	Type	Description
Bit 31: 16	Reserved	0x0000	resd	Kept at default value
Bit 15	ADD	0x0	rw	Add ERTC clock 0: No ERTC clock added 1: One ERTC_CLK is inserted every 2^{11} ERTC_CLK cycles
Bit 14	CAL8	0x0	rw	8-second calibration period 0: No effect 1: 8-second calibration
Bit 13	CAL16	0x0	rw	16 second calibration period 0: No effect 1: 16-second calibration
Bit 12: 9	Reserved	0x0	resd	Kept at default value
Bit 8: 0	DEC	0x000	rw	Decrease ERTC clock DEC out of ERTC_CLK cycles are masked during the 2^{20} ERTC_CLK cycles. This bit is usually used with ADD. When the ADD is set, the actual number of ERTC_CLK is equal to 2^{20}+512-DEC during the 2^{20} ERTC_CLK cycles.

17.4.15 ERTC tamper configuration register (ERTC_TAMP)

Bit	Name	Reset value	Type	Description
Bit 31: 19	Reserved	0x0000	resd	Kept at default value
Bit 18	OUTTYPE	0x0	rw	Output type 0: Open-drain output 1: Push-pull output
Bit 17:16	Reserved	0x0	resd	Kept at default value
Bit 15	TPPU	0x0	rw	Tamper detection pull-up 0: Tamper detection pull-up enabled 1: Tamper detection pull-up disabled
Bit 14: 13	TPPR	0x0	rw	Tamper detection pre-charge time 0: 1 ERTC_CLK cycle 1: 2 ERTC_CLK cycles 2: 4 ERTC_CLK cycles 3: 8 ERTC_CLK cycles
Bit 12: 11	TPFLT	0x0	rw	Tamper detection filter time 0: No filter 1: Tamper is detected after 2 consecutive samples 2: Tamper is detected after 4 consecutive samples 3: Tamper is detected after 8 consecutive samples
Bit 10: 8	TPFREQ	0x0	rw	Tamper detection frequency 0: ERTC_CLK/32768 1: ERTC_CLK/16384 2: ERTC_CLK/8192 3: ERTC_CLK/4096 4: ERTC_CLK/2048 5: ERTC_CLK/1024 6: ERTC_CLK/512 7: ERTC_CLK/256
Bit 7	TPTSEN	0x0	rw	Tamper detection timestamp enable 0: Tamper detection timestamp disabled 1: Tamper detection timestamp enabled. Save timestamp on a tamper event.

Bit 6: 3	Reserved	0x0	resd	Kept at default value
Bit 2	TPIEN	0x0	rw	Tamper detection interrupt enable 0: Tamper detection interrupt disabled 1: Tamper detection interrupt enabled
Bit 1	TP1EDG	0x0	rw	Tamper detection 1 valid edge If TPFLT=0: 0: Rising edge 1: Falling edge If TPFLT>0: 0: Low 1: High
Bit 0	TP1EN	0x0	rw	Tamper detection 1 enable 0: Tamper detection 1 disabled 1: Tamper detection 1 enabled

17.4.16 ERTC alarm clock A subsecond register (ERTC_ALASBS)

Bit	Name	Reset value	Type	Description
Bit 31: 28	Reserved	0x0	resd	Kept at default value
Bit 27: 24	SBSMSK	0x0	rw	Sub-second mask 0: No comparison. Alarm A doesn't care about subseconds. 1: SBS[0] is compared 2: SBS[1: 0] are compared 3: SBS[2: 0] are compared ... 14: SBS[13: 0] are compared 15: SBS[14: 0] are compared
Bit 23: 15	Reserved	0x000	rw	Kept at default value
Bit 14: 0	SBS	0x0000	rw	Sub-second value

17.4.17 ERTC battery powered domain data register (ERTC_BPRx)

Bit	Name	Reset value	Type	Description
Bit 31: 0	DT	0x0000 0000	rw	Battery powered domain data BPR_DTx registers are powered on by V_{BAT} so that they are not reset by a system reset. They are reset on a tamper event or when a battery powered domain is reset.

18 Analog-to-digital converter (ADC)

18.1 ADC introduction

The ADC is a peripheral that converts an analog input signal into a 12-bit/10-bit/8-bit/6-bit digital signal. Its sampling rate is up to 2 MSPS, with up to 19 channels for sampling and conversion.

18.2 ADC main features

Analog:

- 12-bit, 10-bit, 8-bit, 6-bit configurable resolution
- Self-calibration time: 154 ADC clock cycles
- ADC conversion time
  - ADC conversion time is 0.50 µs at max. 28 MHz (in 12-bit resolution)
  - ADC conversion time is 0.28 µs at max. 28 MHz (in 6-bit resolution)
- ADC supply requirement: refer to datasheet
- ADC input range: $V_{SSA} \leq V_{IN} \leq V_{DDA}$

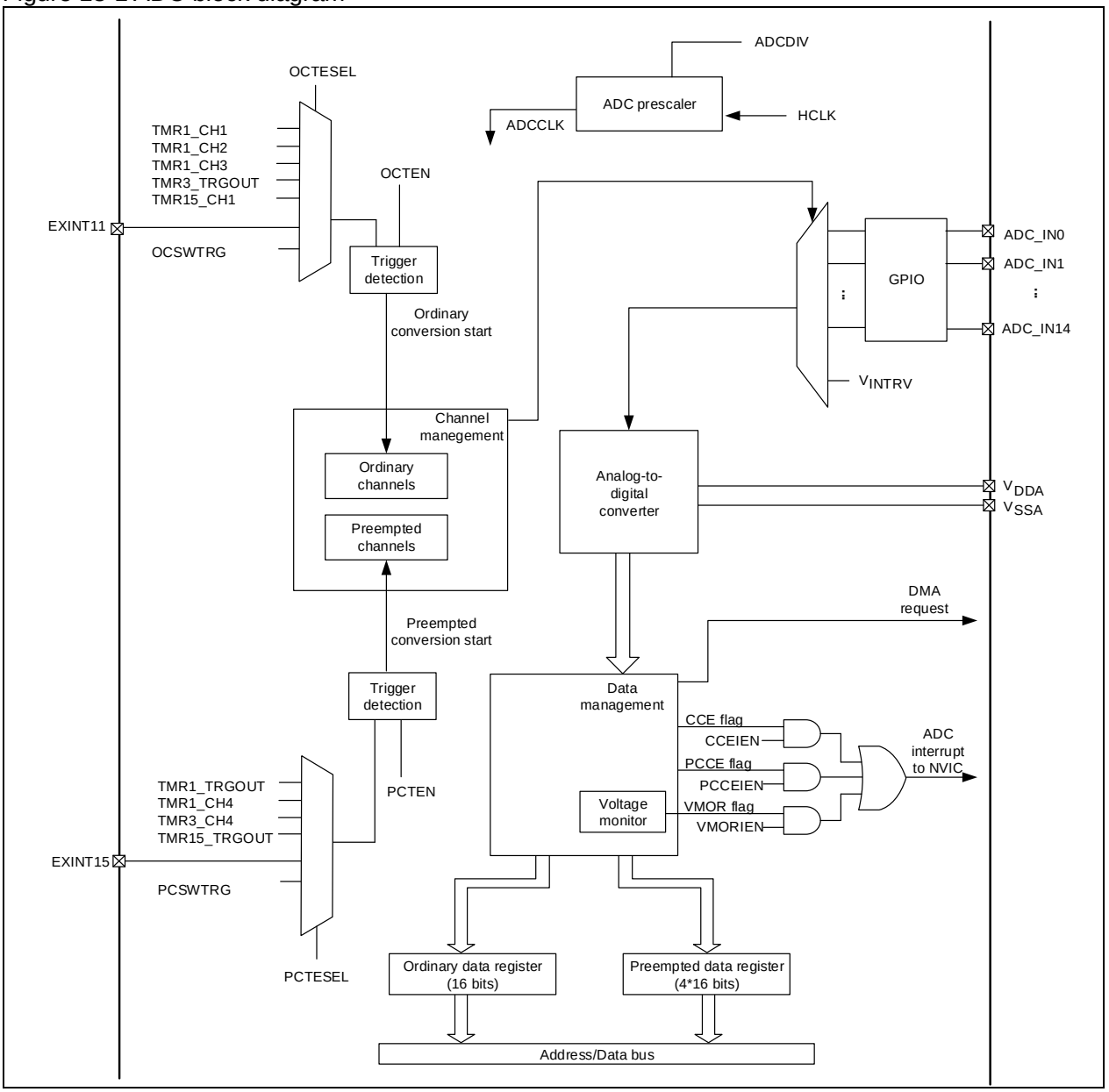
Digital control:

- Manage regular channels and preempted channels based on priority rule
- Regular channels and preempted channels with respective trigger detect circuits
- Programmable sampling time for each channel
- Supports multi-channel conversion mode
- Oversampling: hardware oversampling up to 16-bit resolution
- Optional data alignment format
- Programmable voltage monitoring threshold
- Regular channels with DMA transfers
- Interrupt generation at one of the following events:
  - End of the conversion of preempted channels
  - End of the conversion of regular channels
  - Voltage outside the programmed threshold

18.3 ADC structure

Figure 18-1 shows the block diagram of ADC.

Figure 18-1 ADC block diagram



Input pin description:

- V_{DDA}: ADC analog supply
- V_{SSA}: ADC analog supply ground
- ADCx_IN: Analog input signal channel

Refer to the AT32L021 datasheet for more information.

18.4 ADC functional overview

18.4.1 Channel management

Analog signal channel input:

There are 19 analog signal channel inputs for the ADC, expressed by ADC_Inx (x=0 to 18).

- ADC_IN0 to ADC_IN14: external analog input
- ADC_IN15 and ADC_IN16: $V_{SSA,}$
- ADC_IN17: internal reference voltage
- ADC_IN18: $V_{DDA.}$

Channel conversion

The conversions are divided into two groups: ordinary and preempted channels. Preempted group has priority over ordinary group.

If a preempted channel trigger occurs in the course of ordinary channel conversion, the ordinary channel conversion is interrupted, giving priority to the preempted channel, and the ordinary channel continues its conversion at the end of the preempted channel conversion. If the ordinary channel trigger occurs during preempted channel conversion, the ordinary channel conversion won't start until the end of the preempted channel conversion.

Program the ADC_Inx into the ordinary channel sequence (ADC_OSQx) and the preempted channel sequence (ADC_PSQ), and the same channels can be repeated. The total number of sequences is determined by OCLEN and PCLEN, then it is ready to enable the ordinary channel or preempted channel conversion.

18.4.1.1 Internal reference voltage

The internal reference voltage of a 1.2 V typical value is connected to ADC1_IN17. It is required to enable the ITSRVEN bit in the ADC_CTRL2 register before converting the internal reference channel. The converted data of this channel can be used to calculate an external reference voltage.

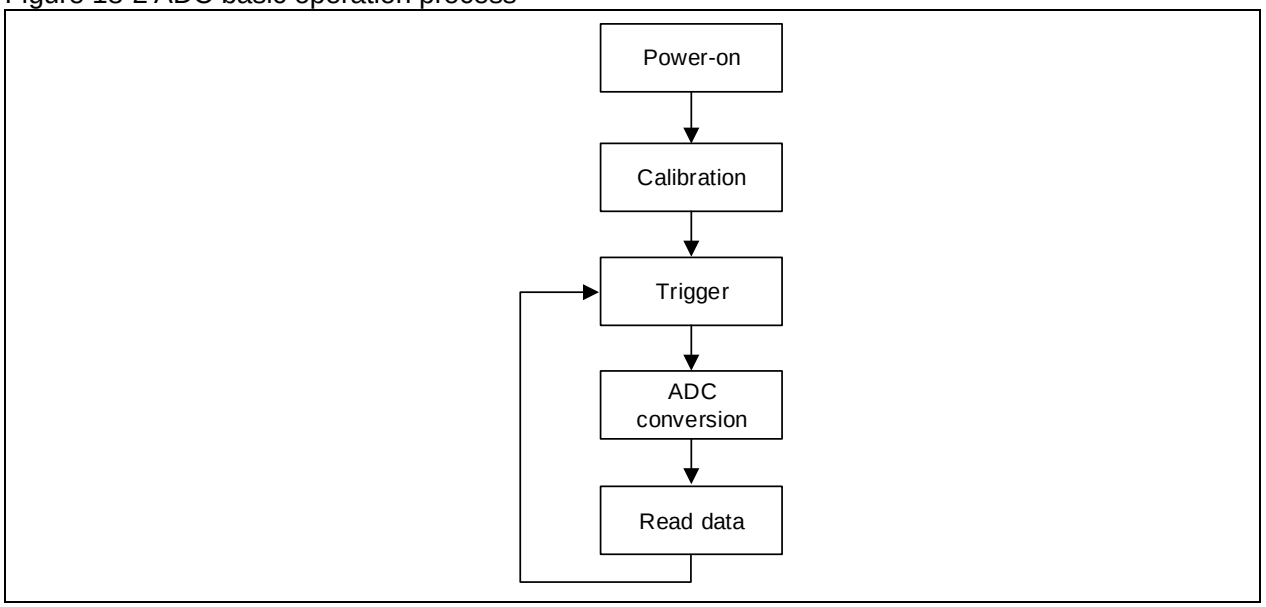
18.4.1.2 ADC internal sampling capacitor precharge

Precharge mode can be enabled through the PRECEN bit in the ADC_CTRL1 register. Once enabled, the internal sampling capacitor of ADC can be automatically charged and discharged to VREF+/2 during ADC non-sampling period.

18.4.2 ADC operation process

Figure 18-2 shows a basic ADC flowchart. It is recommended to do calibration after the initial power-on in order to improve the accuracy of sampling and conversion. After the calibration, a trigger event can activate ADC sampling and conversion. Then read data at the end of the conversion.

Figure 18-2 ADC basic operation process



18.4.2.1 Power-on and calibration

Power-on

Set the ADCEN bit in the CRM_APB2EN register to enable ADC clocks: PCLK2 and ADCCLK.

Program the desired ADCCLK frequency by setting the ADCDIV bit in the ADC_CCTRL register.

ADCCLK uses a divided-frequency HCLK as its clock source.

Note: ADCCLK must be less than 28 MHz.

Then set the ADCEN bit in the ADC_CTRL2 register to supply the ADC, and wait until the t_{STAB} flag is set before starting ADC conversion. Clearing the ADCEN bit will stop ADC conversion and result in a reset. In the meantime, the ADC is switched off to save power consumption.

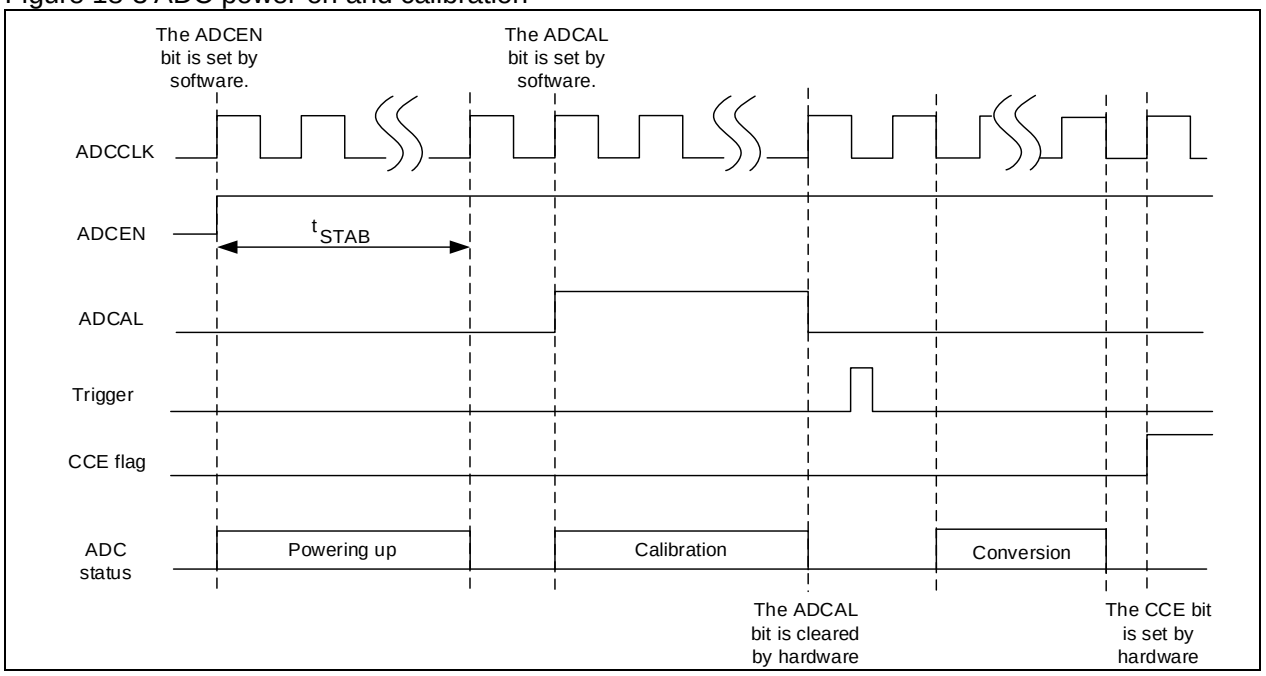
Calibration

After power-on, enable calibration by setting the ADCAL bit in the ADC_CTRL2 register. When the calibration is complete, the ADCAL bit is cleared by hardware and the conversion is triggered by software.

After each calibration, the calibration value is stored in the ADC_ODT register, and then it is automatically sent back to the ADC so as to eliminate capacitance errors. The storage of the calibration value will not set the CCE flag, generate an interrupt or DMA request.

Note: The calibration can only be done when the 12-bit resolution is used and PRECEN bit is disabled in the ADC_CTRL1 register. The switch of the resolution and precharge can be allowed only after the completion of the calibration. Then it is necessary to wait until the RDY flag is set before triggering resolution switching.

Figure 18-3 ADC power-on and calibration



18.4.2.2 Trigger

The ADC trigger involves ordinary channels and preempted channels. The ordinary channel conversion is triggered by ordinary channels, while the preempted channel conversion is triggered by preempted ones. After the OCTEN or PCTEN bit is enabled in the ADC_CTRL2 register, ADC will detect the rising edge of a trigger source and start conversion.

The conversion can be triggered by software writing the OCSWTRG and PCSWTRG bits in the ADC_CTRL2 register, or by an external event. The external events come from timers and pins, depending on the OCTESEL and PCTESEL bits in the ADC_CTRL2 register, as shown in Table 18-1.

For regular group, it also has a special trigger source, which is to repeatedly enable ADCEN for triggering ADC conversion, without the need of enabling OCTEN bit in the ADC_CTRL2 register.

Table 18-1 Trigger sources for ordinary and preempted channels

OCTESEL	Trigger source	PCTESEL	Trigger source
000	TMR1_CH1 event	000	TMR1_TRGOUT event
001	TMR1_CH2 event	001	TMR1_CH4 event
010	TMR1_CH3 event	010	Reserved
011	Reserved	011	Reserved
100	TMR3_TRGOUT event	100	TMR3_CH4 event
101	TMR15_CH1 event	101	TMR15_TRGOUT event
110	EXINT line11 external pin	110	EXINT line15 external pin
111	OCSWTRG bit	111	PCSWTRG bit

18.4.2.3 Sampling and conversion sequence

The sampling period can be configured by setting the CSPTx bit in the ADC_SPT1 and ADC_SPT2 registers. The duration required for a single conversion is calculated based on the following formula:

$$\text{A single conversion time (ADCCLK period)} = \text{sampling time} + \text{resolution bits} + 0.5$$

Example:

If the CSPTx selects 1.5 cycles, CRSEL selects 12 bits, then a single conversion needs 1.5+12.5=14 ADCCLK cycles

If the CSPTx selects 7.5 cycles, CRSEL selects 10 bits, then a conversion needs 7.5+10.5=18 ADCCLK cycles

18.4.3 Conversion sequence management

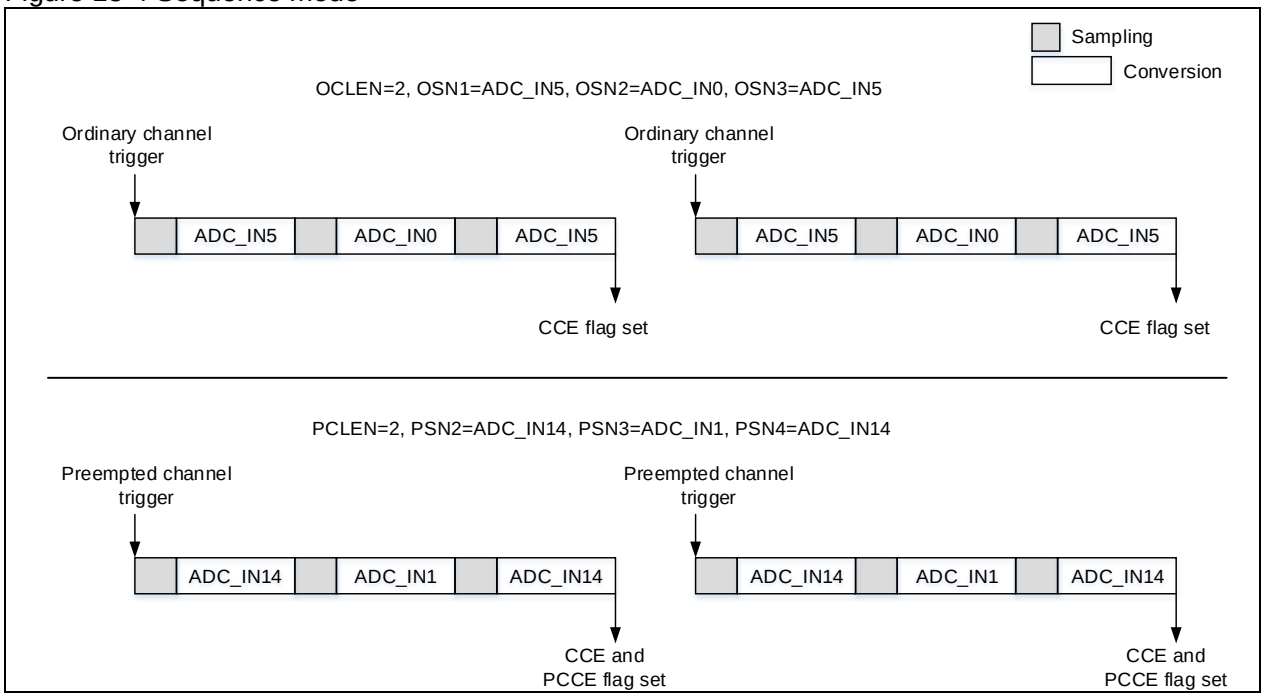
By default, only one channel is converted at each trigger, which is either OSN1-defined channel (regular channel) or PSN4-defined channel (preempted channel).

The following section describes various conversion sequence modes in detail. This mode enables multiple channels to be converted in a specific sequence.

18.4.3.1 Sequence mode

The sequence mode is enabled by setting the SQEN bit in the ADC_CTRL1 register. The ADC_OSQx registers are used to configure the sequence and the number of ordinary channels, while the ADC_PSQ register is used to define the sequence and the number of preempted channels. After the sequence mode is enabled, a single trigger event enables the conversion of a group of channels in order. The ordinary channels start converting from the QSN1 while the preempted channels start from the PSNx (x=4-PCLEN). Figure 18-4 shows an example of the behavior in sequence mode.

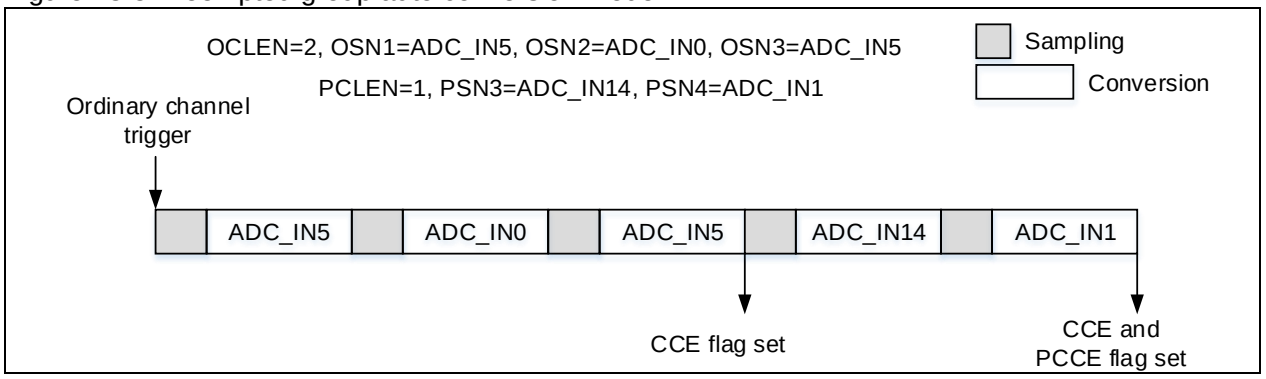
Figure 18-4 Sequence mode



18.4.3.2 Preempted group automatic conversion mode

The automatic conversion mode for preempted group is enabled by setting the PCAUTOEN bit in the ADC_CTRL1 register. In this mode, once the ordinary channel conversion is over, the preempted group will automatically continue its conversion. This mode can work in conjunction with the sequence mode. The preempted group conversion starts automatically at the end of the conversion of the ordinary group. *Figure 18-5* shows an example of the behavior when the automatic preempted group conversion mode works together with the ordinary group.

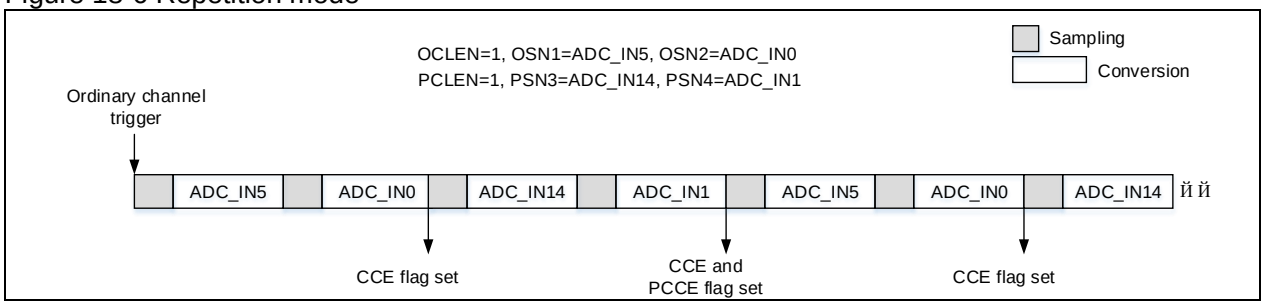
Figure 18-5 Preempted group auto conversion mode



18.4.3.3 Repetition mode

The repetition mode is enabled by setting the RPEN bit in the ADC_CTRL2 register. When a trigger signal is detected, the ordinary channels will be converted repeatedly. This mode can work in conjunction with the ordinary channel conversion in sequence mode to enable the repeated conversion of the ordinary group. Such mode can also work with the preempted group auto conversion mode to repeatedly convert the ordinary group and preempted group in sequence. *Figure 18-6* shows an example of the behavior when the repetition mode works together with the sequence mode and preempted group auto conversion mode.

Figure 18-6 Repetition mode



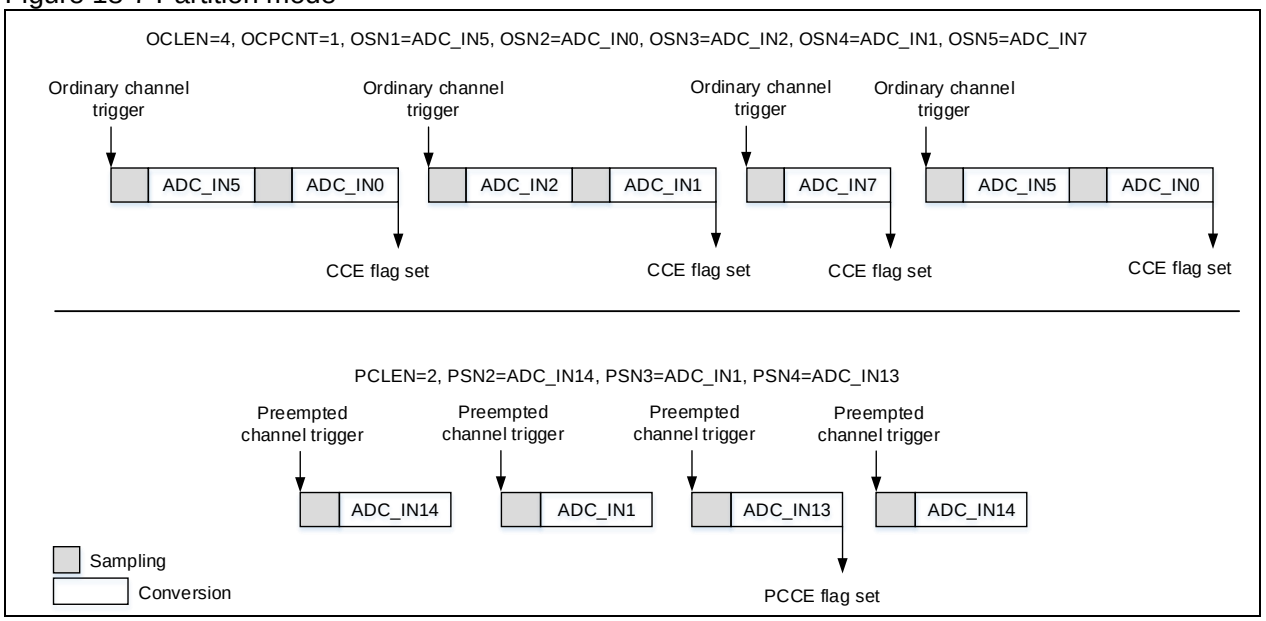
18.4.3.4 Partition mode

The partition mode of the ordinary group can be enabled by setting the OCPEN bit in the ADC_CTRL1 register. In this mode, the ordinary group conversion sequence length (OCLEN bit in the ADC_OSQ1 register) is divided into subgroups. The number of channels in a subgroup is specified through the OCPCNT bit in the ADC_CTRL1 register. When a trigger occurs, all channels in the subgroup will be converted. Each trigger selects a different subgroup in order.

Setting the PCPEN bit in the ADC_CTRL1 register enables the partition mode of preempted group. In this mode, the preempted group conversion sequence length (PCLEN bit in the ADC_PSQ register) is divided into subgroups, each of which has only one channel. When a trigger occurs, the channel in the subgroup will be converted. Each trigger event selects a different sub-group in order.

The partition mode and repetition mode cannot be used at the same time. *Figure 18-7* shows an example of the behavior in partition mode for ordinary group and preempted group.

Figure 18-7 Partition mode



18.4.4 Oversampling

The converted data of a single oversampling are obtained by enabling multiple conversions of the same channel and then averaging the accumulated conversion data.

- Oversampling ratio is defined through the OSRSEL bit in the ADC_OVSP register. This bit is used to define the oversampling multiples (times), which is achieved by converting the same channel many times
- Oversampling shift is selected through the OSSSEL bit in the ADC_OVSP register. This bit defines the average coefficient, which is done by right shift.

If the averaged data is greater than 16 bits, then only the right-aligned 16-bit data is taken and put into a 16-bit data register, shown in Table 18-2.

Example:

If 4x oversampling is selected through the OSRSEL bit, then the same channel is converted by four times in a single oversampling conversion, and the converted data derived from 4 conversions is put together. If 6-bit resolution is selected through the OSSSE bit, then the accumulated data is divided by 2^6 and rounded up.

Table 18-2 Correlation between maximum cumulative data, oversampling multiple and shift digits

Oversampling multiple	**2x**	**4x**	**8x**	**16x**	**32x**	**64x**	**128x**	**256x**
Max cumulative data	**0x1FFE**	**0x3FFC**	**0x7FF8**	**0xFFF0**	**0x1FFE0**	**0x3FFC0**	**0x7FF80**	**0xFFF00**
No shift	0x1FFE	0x3FFC	0x7FF8	0xFFF0	0xFFE0	0xFFC0	0xFF80	0xFF00
Shift 1 bit	0x0FFF	0x1FFE	0x3FFC	0x7FF8	0xFFF0	0xFFE0	0xFFC0	0xFF80
Shift 2 bit	0x0800	0x0FFF	0x1FFE	0x3FFC	0x7FF8	0xFFF0	0xFFE0	0xFFC0
Shift 3 bit	0x0400	0x0800	0x0FFF	0x1FFE	0x3FFC	0x7FF8	0xFFF0	0xFFE0
Shift 4 bit	0x0200	0x0400	0x0800	0x0FFF	0x1FFE	0x3FFC	0x7FF8	0xFFF0
Shift 5 bit	0x0100	0x0200	0x0400	0x0800	0x0FFF	0x1FFE	0x3FFC	0x7FF8
Shift 6 bit	0x0080	0x0100	0x0200	0x0400	0x0800	0x0FFF	0x1FFE	0x3FFC
Shift 7 bit	0x0040	0x0080	0x0100	0x0200	0x0400	0x0800	0x0FFF	0x1FFE
Shift 8 bit	0x020	0x0040	0x0080	0x0100	0x0200	0x0400	0x0800	0x0FFF

In oversampling conversion mode, the DTALIGN and PCDTOx are ignored, and data must be right aligned.

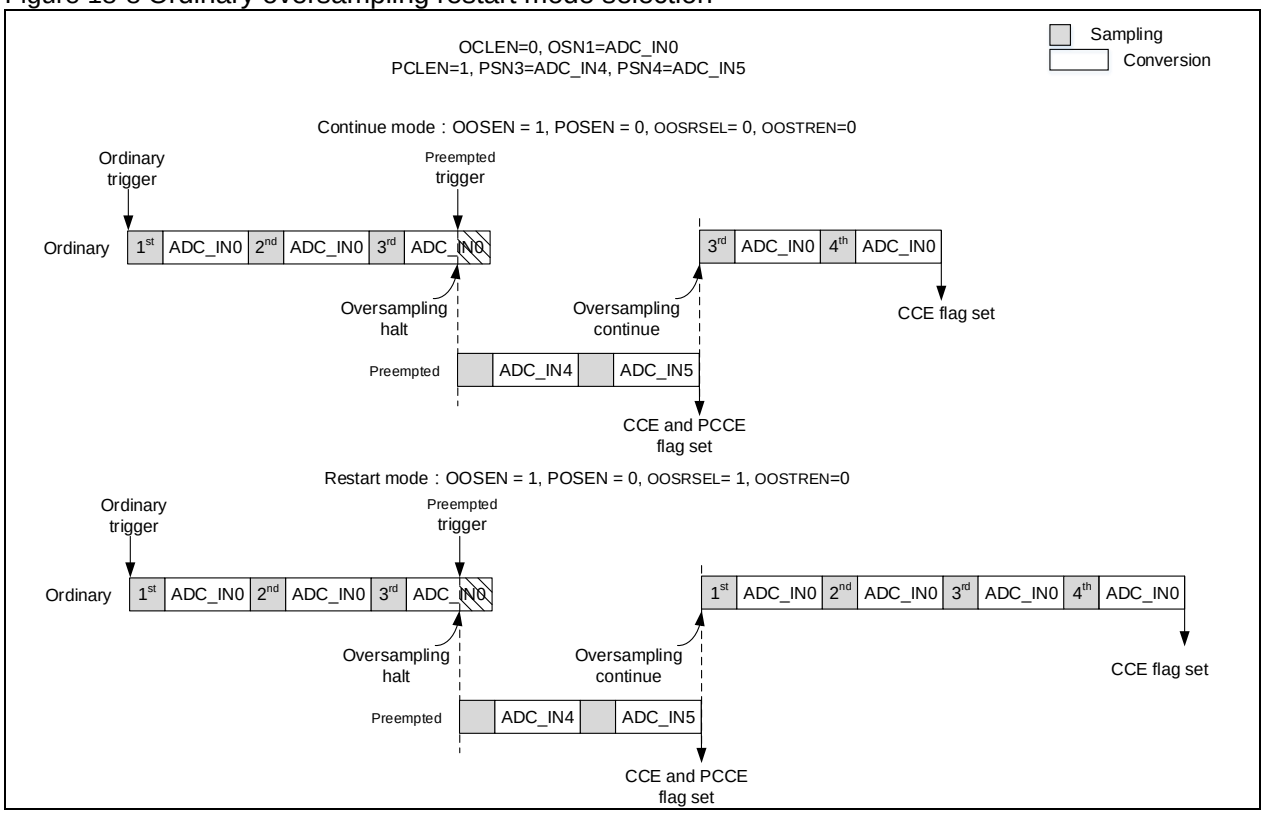
18.4.4.1 Oversampling of ordinary channels

The OOSRSEL bit in the ADC_OVSP register is used to resume ordinary oversampling mode.

- OOSRSEL=0: continuous conversion mode. Ordinary group of channels, after being interrupted by preempted channels during oversampling, will retain its converted data and resume from the last interrupted location.
- OOSRSEL=1: restart mode. Ordinary group of channels, after being interrupted by preempted channels during oversampling, will be reset and restart the ordinary conversion from the beginning. It means that its previous converted data before being interrupted are cleared up.

Figure 18-8 shows the differences between ordinary continuous mode and restart mode in 4x oversampling rate and sequence mode.

Figure 18-8 Ordinary oversampling restart mode selection

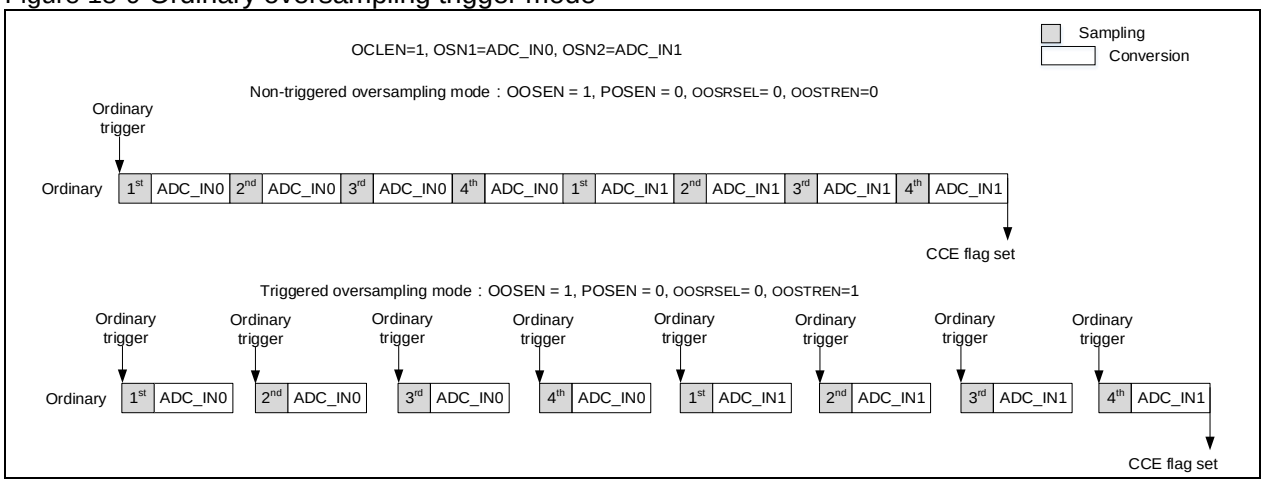


Trigger mode can be enabled by setting the OOSTREN bit in the ADC_OVSP register. In this mode, ordinal channels must be triggered for conversion. Once the ordinary conversion is interrupted by preempted channels, it is necessary to re-trigger ordinary group of channels to resume its oversampling.

When the trigger mode works in conjunction with conversion sequence management mode, trigger mode is applied, and the conversion complete flag follows the conversion sequence management mode. *Figure 18-9* shows the behavior when the ordinary trigger mode works together with resume mode in 4x oversampling rate and sequential mode.

Note: The trigger mode and repetition mode cannot be used concurrently.

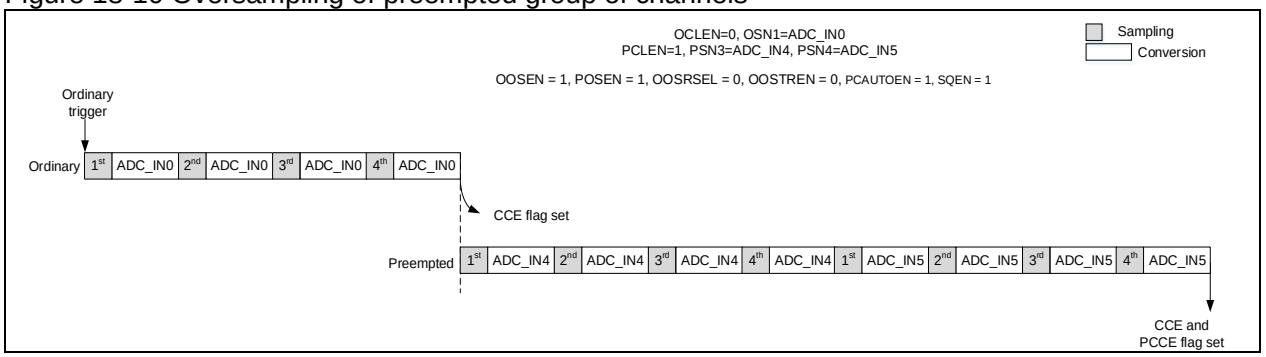
Figure 18-9 Ordinary oversampling trigger mode



18.4.4.2 Oversampling of preempted group of channels

The preempted oversampling and ordinary oversampling can be used simultaneously or individually. The oversampling of preempted group of channels does not affect ordinary oversampling modes. *Figure 18-10* shows the behavior when the preempted oversampling and ordinary oversampling trigger mode are used simultaneously in 4x oversampling rate and auto-conversion of preempted group.

Figure 18-10 Oversampling of preempted group of channels



18.4.5 Data management

At the end of the conversion of the ordinary group, the converted data are stored in the ADC_ODT register. Once the preempted group conversion ends, their converted data are stored in the ADC_PDTx register.

18.4.5.1 Data alignment

DTALIGN bit in the ADC_CTRL2 register selects the alignment of data (right-aligned or left-aligned). As the offset of the ADC_PCDTOx register is subtracted from the converted data of preempted group, the result may be a negative value, marked by SIGN.

The data are aligned on a half-word basis expect when the resolution is set to 6-bit. In 6-bit resolution mode, the data are aligned on a byte basis, as shown in *Figure 18-11.*

Figure 18-11 Data alignment

Ordinary channel data 12 bits

Right-alignment

0	0	0	0	DT[11]	DT[10]	DT[9]	DT[8]	DT[7]	DT[6]	DT[5]	DT[4]	DT[3]	DT[2]	DT[1]	DT[0]

Left-alignment

DT[11]	DT[10]	DT[9]	DT[8]	DT[7]	DT[6]	DT[5]	DT[4]	DT[3]	DT[2]	DT[1]	DT[0]	0	0	0	0

Ordinary channel data 6 bits

Right-alignment

0	0	0	0	0	0	0	0	0	0	DT[5]	DT[4]	DT[3]	DT[2]	DT[1]	DT[0]

Left-alignment

0	0	0	0	0	0	0	0	DT[5]	DT[4]	DT[3]	DT[2]	DT[1]	DT[0]	0	0

Preempted channel data 12 bits

Right-alignment

SIGN	SIGN	SIGN	SIGN	DT[11]	DT[10]	DT[9]	DT[8]	DT[7]	DT[6]	DT[5]	DT[4]	DT[3]	DT[2]	DT[1]	DT[0]

Left-alignment

SIGN	DT[11]	DT[10]	DT[9]	DT[8]	DT[7]	DT[6]	DT[5]	DT[4]	DT[3]	DT[2]	DT[1]	DT[0]	0	0	0

Preempted channel data 6 bits

Right-alignment

SIGN	SIGN	SIGN	SIGN	SIGN	SIGN	SIGN	SIGN	SIGN	SIGN	DT[5]	DT[4]	DT[3]	DT[2]	DT[1]	DT[0]

Left-alignment

SIGN	SIGN	SIGN	SIGN	SIGN	SIGN	SIGN	SIGN	SIGN	DT[5]	DT[4]	DT[3]	DT[2]	DT[1]	DT[0]	0

18.4.5.2 Data read

Read access to the ADC_ODT register using CPU or DMA gets the converted data of ordinary group. Read access to the ADC_PDTx register using CPU gets the converted data of preempted group.

When the OCDMAEN bit is set in the ADC_CTRL2 register, the ADC will issue a DMA request each time the ADC_OTD register is updated.

18.4.6 Voltage monitoring

The OCVMEN bit or PCVMEN bit in the ADC_CTRL1 register is used to enable voltage monitoring based on the converted data.

The VMOR bit will be set if the converted result is outside the high threshold (ADC_VMHB register) or less than the low threshold (ADC_VMLB register).

The VMSGEN bit in the ADC_CTRL1 register is used to enable voltage monitoring on either a single channel or all the channels. The VMCSEL bit is used to select a specific channel for voltage monitoring.

Voltage monitoring is based on the comparison result between the original converted data and the 12-bit voltage monitor boundary register, regardless of the CRSEL, PCDTOx and DTALIGN bits.

When using an oversampler, voltage monitoring is based on the comparison result between the 16-bit registers (ADC_VMHB[15:0] and ADC_VMLB[15:0]) and the oversampled data.

18.4.7 Status flag and interrupts

The ADC has its dedicated ADCx_STS registers, namely, ordinary channel conversion start flag (OCCS), preempted channel conversion start flag (PCCS), preempted channel conversion end flag (PCCE), channel conversion end flag (CCE) and voltage monitor out of range (VMOR).

PCCE, CCE and VMOR have their respective interrupt enable bits. Once these interrupt bits are enabled, the corresponding flag is set and an interrupt is sent to CPU.

18.5 ADC registers

Table 18-3 lists ADC register map and their reset values.

These peripheral registers must be accessed by words (32 bits).

Table 18-3 ADC register map and reset values

Register name	Offset	Reset value
ADC_STS	0x000	0x0000 0000
ADC_CTRL1	0x004	0x0000 0000
ADC_CTRL2	0x008	0x0000 0000
ADC_SPT1	0x00C	0x0000 0000
ADC_SPT2	0x010	0x0000 0000
ADC_PCDTO1	0x014	0x0000 0000
ADC_PCDTO2	0x018	0x0000 0000
ADC_PCDTO3	0x01C	0x0000 0000
ADC_PCDTO4	0x020	0x0000 0000
ADC_VMHB	0x024	0x0000 0FFF
ADC_VMLB	0x028	0x0000 0000
ADC_OSQ1	0x02C	0x0000 0000
ADC_OSQ2	0x030	0x0000 0000
ADC_OSQ3	0x034	0x0000 0000
ADC_PSQ	0x038	0x0000 0000
ADC_PDT1	0x03C	0x0000 0000
ADC_PDT2	0x040	0x0000 0000
ADC_PDT3	0x044	0x0000 0000
ADC_PDT4	0x048	0x0000 0000
ADC_ODT	0x04C	0x0000 0000
ADC_OVSP	0x080	0x0000 0000

18.5.1 ADC status register (ADC_STS)

Accessed by words.

Bit	Name	Reset value	Type	Description
Bit 31: 7	Reserved	0x0000000	resd	Kept at default value.
Bit 6	RDY	0x0	rw0c	ADC conversion ready flag This bit is read only. It is set by hardware when the ADC is powered on. 0: Not ready for ADC conversion 1: Ready for ADC conversion
Bit 5	Reserved	0x0	resd	Kept at default value.
Bit 4	OCCS	0x0	rw0c	Ordinary channel conversion start flag This bit is set by hardware and cleared by software (writing 0). 0: No ordinary channel conversion started 1: Ordinary channel conversion has started
Bit 3	PCCS	0x0	rw0c	Preempted channel conversion start flag This bit is set by hardware and cleared by software (writing 0). 0: No preempted channel conversion started 1: Preempted channel conversion has started
Bit 2	PCCE	0x0	rw0c	Preempted channel end of conversion flag This bit is set by hardware and cleared by software (writing 0). 0: Conversion is not complete 1: Conversion is complete
Bit 1	OCCE	0x0	rw0c	End of conversion flag This bit is set by hardware. It is cleared by software (writing 0) or by reading the ADC_ODT register. 0: Conversion is not complete 1: Conversion is complete
Bit 0	VMOR	0x0	rw0c	Voltage monitoring out of range flag This bit is set by hardware and cleared by software (writing 0). 0: Voltage is within the value programmed 1: Voltage is outside the value programmed

18.5.2 ADC control register1 (ADC_CTRL1)

Accessed by words.

Bit	Name	Reset value	Type	Description
Bit 31	PRECEN	0x0	rw	Precharge enable This bit is used to precharge/discharge to VREF/2 after each ADC conversion in order to cut the time for the next sampling. 0: Disabled 1: Enabled
Bit 30: 26	Reserved	0x00	resd	Kept at default value.
Bit 25: 24	CRSEL	0x0	rw	Conversion resolution select 00: 12-bit 01: 10-bit 10: 8-bit 11: 6-bit
Bit 23	OCVMEN	0x0	rw	Voltage monitoring enable on ordinary channels 0: Voltage monitoring disabled on ordinary channels 1: Voltage monitoring enabled on ordinary channels
Bit 22	PCVMEN	0x0	rw	Voltage monitoring enable on preempted channels 0: Voltage monitoring disabled on preempted channels 1: Voltage monitoring enabled on preempted channels
Bit 21: 16	Reserved	0x0	resd	Kept at default value.
Bit 15: 13	OCPCNT	0x0	rw	Partitioned mode conversion count of ordinary channels 000: 1 channel 001: 2 channels …… 111: 8 channels Note: In this mode, the preempted group converts only one channel at each trigger.

Bit 12	PCPEN	0x0	rw	Partitioned mode enable on preempted channels 0: Partitioned mode disabled on preempted channels 1: Partitioned mode enabled on preempted channels
Bit 11	OCPEN	0x0	rw	Partitioned mode enable on ordinary channels This is set and cleared by software to enable or disable partitioned mode on ordinary channels. 0: Partitioned mode disabled on ordinary channels 1: Partitioned mode enabled on ordinary channels
Bit 10	PCAUTOEN	0x0	rw	Preempted group automatic conversion enable after ordinary group 0: Preempted group automatic conversion disabled 1: Preempted group automatic conversion enabled
Bit 9	VMSGEN	0x0	rw	Voltage monitoring enable on a single channel 0: Disabled (Voltage monitoring enabled on all channels) 1: Enabled (Voltage monitoring enabled a single channel)
Bit 8	SQEN	0x0	rw	Sequence mode enable 0: Sequence mode disabled (a single channel is converted) 1: Sequence mode enabled (the selected multiple channels are converted) Note: If this mode is enabled and the CCEIEN/PCCEIEN is set, a CCE or PCCE interrupt is generated only at the end of conversion of the last channel.
Bit 7	PCCEIEN	0x0	rw	Conversion end interrupt enable on Preempted channels 0: Conversion end interrupt disabled on Preempted channels 1: Conversion end interrupt enabled on Preempted channels
Bit 6	VMORIEN	0x0	rw	Voltage monitoring out of range interrupt enable 0: Voltage monitoring out of range interrupt disabled 1: Voltage monitoring out of range interrupt enabled
Bit 5	CCEIEN	0x0	rw	Channel conversion end interrupt enable 0: Channel conversion end interrupt disabled 1: Channel conversion end interrupt enabled
Bit 4: 0	VMCSEL	0x00	rw	Voltage monitoring channel select This filed is valid only when the VMSGEN is enabled. 00000: ADC_IN0 channel 00001: ADC_IN1 channel …… 01111: ADC_IN15 channel 10000: ADC_IN16 channel 10001: ADC_IN17 channel 10010: ADC_IN18 channel 11011~11111: Unused, do not configure.

18.5.3 ADC control register2 (ADC_CTRL2)

Accessed by words.

Bit	Name	Reset value	Type	Description
Bit 30: 24	Reserved	0x00	resd	Kept at default value
Bit 23	ITSRVEN	0x0	rw	Internal VINTRV enable 0: Disabled 1: Enabled
Bit 22	OCSWTRG	0x0	rw	Ordinary channels conversion triggered by software 0: Ordinary channels conversion not triggered 1: Ordinary channels conversion triggered (it can be triggered by software, or cleared by hardware as soon as the conversion starts)
Bit 21	PCSWTRG	0x0	rw	Preempted channels conversion triggered by software 0: Preempted channels conversion not triggered 1: Preempted channels conversion triggered (This bit is cleared by software or by hardware as soon as the conversion starts)

Bit 20	OCTEN	0x0	rw	Trigger mode enable for ordinary channels conversion 0: Disabled 1: Enabled
Bit 19: 17	OCTESEL	0x0	rw	Trigger event select for ordinary channels conversion 000: TMR1 CH1 event 001: TMR1 CH2 event 010: TMR1 CH3 event 011: Unused. Do not configure. 100: TMR3 TRGOUT event 101: TMR15 CH1 event 110: EXINT 11 111: OCSWTRG
Bit 16	Reserved	0x0	resd	Kept at default value.
Bit 15	PCTEN	0x0	rw	Trigger mode enable for preempted channels conversion 0: Disabled 1: Enabled
Bit 14: 12	PCTESEL	0x0	rw	Trigger event select for preempted channels conversion 000: TMR1 TRGOUT event 001: TMR1 CH4 event 010: Unused. Do not configure. 011: Unused. Do not configure. 100: TMR3 CH4 event 101: TMR15 TRGOUT event 110: EXINT 15 111: PCSWTRG
Bit 11	DTALIGN	0x0	rw	Data alignment 0: Right alignment 1: Left alignment
Bit 10:9	Reserved	0x0	resd	Kept at default value.
Bit 8	OCDMAEN	0x0	rw	DMA transfer enable of ordinary channels 0: Disabled 1: Enabled
Bit 7: 4	Reserved	0x0	resd	Kept at default value.
Bit 3	ADCALINIT	0x0	rw	Initialize A/D calibration This bit is set by software and cleared by hardware. It is cleared after the calibration registers are initialized. 0: No initialization occurred or initialization completed 1: Enable initialization or initialization is ongoing
Bit 2	ADCAL	0x0	rw	A/D Calibration 0: No calibration occurred or calibration completed 1: Enable calibration or calibration is in process
Bit 1	RPEN	0x0	rw	Repetition mode enable 0: Repetition mode disabled When SQEN=0, a single channel is converted each time when a trigger event arrives; when SQEN=1, a group of channels are converted each timer when a trigger event arrives. 1: Repetition mode enabled When SQEN =0, continuous conversion mode on a single channel is enabled at each trigger event; when SQEN =1, continuous conversion mode on a group of channels is enabled at each trigger event. Until ADCEN is cleared.
Bit 0	ADCEN	0x0	rw	A/D converter enable 0: A/D converter disabled (ADC goes to power-down mode) 1: A/D converter enabled Note: When this bit is OFF, writing an ON command can wake up the ADC from power-down mode.

				Writing ON command but without chaning other bits in the register while this bit is ON will trigger the conversion of ordinary group. The application should pay attention to the fact that there is a delay of t_{STAB} between power up and start of conversion.

18.5.4 ADC sampling time register 1 (ADC_SPT1)

Accessed by words.

Bit	Name	Reset value	Type	Description
Bit 31:27	Reserved	0x00	resd	Kept at default value.
Bit 26: 24	CSPT18	0x0	rw	Sample time selection of channel ADC_IN18 000: 1.5 cycles 001: 7.5 cycles 010: 13.5 cycles 011: 28.5 cycles 100: 41.5 cycles 101: 55.5 cycles 110: 71.5 cycles 111: 239.5 cycles
Bit 23: 21	CSPT17	0x0	rw	Sample time selection of channel ADC_IN17 000: 1.5 cycles 001: 7.5 cycles 010: 13.5 cycles 011: 28.5 cycles 100: 41.5 cycles 101: 55.5 cycles 110: 71.5 cycles 111: 239.5 cycles
Bit 20: 18	CSPT16	0x0	rw	Sample time selection of channel ADC_IN16 000: 1.5 cycles 001: 7.5 cycles 010: 13.5 cycles 011: 28.5 cycles 100: 41.5 cycles 101: 55.5 cycles 110: 71.5 cycles 111: 239.5 cycles
Bit 17: 15	CSPT15	0x0	rw	Sample time selection of channel ADC_IN15 000: 1.5 cycles 001: 7.5 cycles 010: 13.5 cycles 011: 28.5 cycles 100: 41.5 cycles 101: 55.5 cycles 110: 71.5 cycles 111: 239.5 cycles
Bit 14: 12	CSPT14	0x0	rw	Sample time selection of channel ADC_IN14 000: 1.5 cycles 001: 7.5 cycles 010: 13.5 cycles 011: 28.5 cycles 100: 41.5 cycles 101: 55.5 cycles

				110: 71.5 cycles 111: 239.5 cycles
Bit 11: 9	CSPT13	0x0	rw	Sample time selection of channel ADC_IN13 000: 1.5 cycles 001: 7.5 cycles 010: 13.5 cycles 011: 28.5 cycles 100: 41.5 cycles 101: 55.5 cycles 110: 71.5 cycles 111: 239.5 cycles
Bit 8: 6	CSPT12	0x0	rw	Sample time selection of channel ADC_IN12 000: 1.5 cycles 001: 7.5 cycles 010: 13.5 cycles 011: 28.5 cycles 100: 41.5 cycles 101: 55.5 cycles 110: 71.5 cycles 111: 239.5 cycles
Bit 5: 3	CSPT11	0x0	rw	Sample time selection of channel ADC_IN11 000: 1.5 cycles 001: 7.5 cycles 010: 13.5 cycles 011: 28.5 cycles 100: 41.5 cycles 101: 55.5 cycles 110: 71.5 cycles 111: 239.5 cycles
Bit 2: 0	CSPT10	0x0	rw	Sample time selection of channel ADC_IN10 000: 1.5 cycles 001: 7.5 cycles 010: 13.5 cycles 011: 28.5 cycles 100: 41.5 cycles 101: 55.5 cycles 110: 71.5 cycles 111: 239.5 cycles

18.5.5 ADC sampling time register 2 (ADC_SPT2)

Accessed by words.

Bit	Name	Reset value	Type	Description
Bit 31: 30	Reserved	0x0	resd	Kept at default value
Bit 29: 27	CSPT9	0x0	rw	Sample time selection of channel ADC_IN9 000: 1.5 cycles 001: 7.5 cycles 010: 13.5 cycles 011: 28.5 cycles 100: 41.5 cycles 101: 55.5 cycles 110: 71.5 cycles 111: 239.5 cycles
Bit 26: 24	CSPT8	0x0	rw	Sample time selection of channel ADC_IN8 000: 1.5 cycles 001: 7.5 cycles 010: 13.5 cycles 011: 28.5 cycles 100: 41.5 cycles 101: 55.5 cycles 110: 71.5 cycles 111: 239.5 cycles
Bit 23: 21	CSPT7	0x0	rw	Sample time selection of channel ADC_IN7 000: 1.5 cycles 001: 7.5 cycles 010: 13.5 cycles 011: 28.5 cycles 100: 41.5 cycles 101: 55.5 cycles 110: 71.5 cycles 111: 239.5 cycles
Bit 20: 18	CSPT6	0x0	rw	Sample time selection of channel ADC_IN6 000: 1.5 cycles 001: 7.5 cycles 010: 13.5 cycles 011: 28.5 cycles 100: 41.5 cycles 101: 55.5 cycles 110: 71.5 cycles 111: 239.5 cycles
Bit 17: 15	CSPT5	0x0	rw	Sample time selection of channel ADC_IN5 000: 1.5 cycles 001: 7.5 cycles 010: 13.5 cycles 011: 28.5 cycles 100: 41.5 cycles 101: 55.5 cycles 110: 71.5 cycles 111: 239.5 cycles

Bit 14: 12	CSPT4	0x0	rw	Sample time selection of channel ADC_IN4 000: 1.5 cycles 001: 7.5 cycles 010: 13.5 cycles 011: 28.5 cycles 100: 41.5 cycles 101: 55.5 cycles 110: 71.5 cycles 111: 239.5 cycles
Bit 11: 9	CSPT3	0x0	rw	Sample time selection of channel ADC_IN3 000: 1.5 cycles 001: 7.5 cycles 010: 13.5 cycles 011: 28.5 cycles 100: 41.5 cycles 101: 55.5 cycles 110: 71.5 cycles 111: 239.5 cycles
Bit 8: 6	CSPT2	0x0	rw	Sample time selection of channel ADC_IN2 000: 1.5 cycles 001: 7.5 cycles 010: 13.5 cycles 011: 28.5 cycles 100: 41.5 cycles 101: 55.5 cycles 110: 71.5 cycles 111: 239.5 cycles
Bit 5: 3	CSPT1	0x0	rw	Sample time selection of channel ADC_IN1 000: 1.5 cycles 001: 7.5 cycles 010: 13.5 cycles 011: 28.5 cycles 100: 41.5 cycles 101: 55.5 cycles 110: 71.5 cycles 111: 239.5 cycles
Bit 2: 0	CSPT0	0x0	rw	Sample time selection of channel ADC_IN0 000: 1.5 cycles 001: 7.5 cycles 010: 13.5 cycles 011: 28.5 cycles 100: 41.5 cycles 101: 55.5 cycles 110: 71.5 cycles 111: 239.5 cycles

18.5.6 ADC preempted channel data offset register x (ADC_ PCDTOx) (x=1..4)

Accessed by words.

Bit	Name	Reset value	Type	Description
Bit 31: 12	Reserved	0x00000	resd	Kept at default value
Bit 11: 0	PCDTOx	0x000	rw	Data offset for Preempted channel x Converted data stored in the ADC_PDTx = Raw converted data – ADC_ PCDTOx

18.5.7 ADC voltage monitoring high threshold register (ADC_VWHB)

Accessed by words.

Bit	Name	Reset value	Type	Description
Bit 31: 16	Reserved	0x00000	resd	Kept at default value
Bit 15: 0	VMHB	0xFFF	rw	Voltage monitoring high boundary

18.5.8 ADC voltage monitor low threshold register (ADC_ VWLB)

Accessed by words.

Bit	Name	Reset value	Type	Description
Bit 31: 12	Reserved	0x00000	resd	Kept at default value
Bit 11: 0	VMLB	0x000	rw	Voltage monitoring low boundary

18.5.9 ADC ordinary sequence register 1 (ADC_ OSQ1)

Accessed by words.

Bit	Name	Reset value	Type	Description
Bit 31: 24	Reserved	0x00	resd	Kept at default value
Bit 23: 20	OCLEN	0x0	rw	Ordinary conversion sequence length 0000: 1 conversion 0001: 2 conversions …… 11111: 16 conversions
Bit 19: 15	OSN16	0x00	rw	Number of 16th conversion in ordinary sequence
Bit 14: 10	OSN15	0x00	rw	Number of 15th conversion in ordinary sequence
Bit 9: 5	OSN14	0x00	rw	Number of 14th conversion in ordinary sequence
Bit 4: 0	OSN13	0x00	rw	Number of 13th conversion in ordinary sequence Note: The number can be 0~18,20~27. For example, if the number is set to 3, it means that the 13th conversion is ADC_IN3 channel.

18.5.10 ADC ordinary sequence register 2 (ADC_ OSQ2)

Accessed by words.

Bit	Name	Reset value	Type	Description
Bit 31: 30	Reserved	0x0	resd	Kept at default value
Bit 29: 25	OSN12	0x00	rw	Number of 12th conversion in ordinary sequence
Bit 24: 20	OSN11	0x00	rw	Number of 11th conversion in ordinary sequence
Bit 19: 15	OSN10	0x00	rw	Number of 10th conversion in ordinary sequence
Bit 14: 10	OSN9	0x00	rw	Number of 9th conversion in ordinary sequence
Bit 9: 5	OSN8	0x00	rw	Number of 8th conversion in ordinary sequence

Bit 4: 0	OSN7	0x00	rw	Number of 7th conversion in ordinary sequence Note: The number can be 0~18, 20~27. For example, if the number is set to 8, it means that the 7th conversion is ADC_IN8 channel.

18.5.11 ADC ordinary sequence register 3 (ADC_ OSQ3)

Accessed by words.

Bit	Name	Reset value	Type	Description
Bit 31: 30	Reserved	0x0	resd	Kept at default value
Bit 29: 25	OSN6	0x00	rw	Number of 6th conversion in ordinary sequence
Bit 24: 20	OSN5	0x00	rw	Number of 5th conversion in ordinary sequence
Bit 19: 15	OSN4	0x00	rw	Number of 4th conversion in ordinary sequence
Bit 14: 10	OSN3	0x00	rw	Number of 3rd conversion in ordinary sequence
Bit 9: 5	OSN2	0x00	rw	Number of 2nd conversion in ordinary sequence
Bit 4: 0	OSN1	0x00	rw	Number of 1st conversion in ordinary sequence Note: The number can be 0~18, 20~27. For example, if the number is set to 8, it means that the 1st conversion is ADC_IN17 channel.

18.5.12 ADC preempted sequence register (ADC_PSQ)

Accessed by words.

Bit	Name	Reset value	Type	Description
Bit 31: 30	Reserved	0x0	resd	Kept at default value
Bit 21: 20	PCLEN	0x0	rw	Preempted conversion sequence length 00: 1 conversion 01: 2 conversions 10: 3 conversions 11: 4 conversions
Bit 19: 15	PSN4	0x00	rw	Number of 4th conversion in preempted sequence
Bit 14: 10	PSN3	0x00	rw	Number of 3rd conversion in preempted sequence
Bit 9: 5	PSN2	0x00	rw	Number of 2nd conversion in preempted sequence
Bit 4: 0	PSN1	0x00	rw	Number of 1st conversion in preempted sequence Note: The number can be 0~18. For example, if the number is set to 3, it represents ADC_IN3 channel. If PCLEN is less than 4, the conversion sequence starts from 4-PCLEN. For example, when ADC_PSQ ([21: 0]) =10 00110 00101 00100 00011, it indicates that the scan conversion follows the sequence: 4, 5, 6, not 3, 4, 5.

18.5.13 ADC preempted data register x (ADC_ PDTx) (x=1..4)

Accessed by words.

Bit	Name	Reset value	Type	Description
Bit 31: 16	Reserved	0x0000	resd	Kept at default value
Bit 15: 0	PDTx	0x0000	rw	Conversion data from preempted channel

18.5.14 ADC ordinary data register (ADC_ ODT)

Accessed by words.

Bit	Name	Reset value	Type	Description
Bit 31: 16	Reserved	0x0000	resd	Kept at default value
Bit 15: 0	ODT	0x0000	ro	Conversion data of ordinary channel

18.5.15 ADC oversampling register (ADC_OVSP)

Accessed by words.

Bit	Name	Reset value	Type	Description
Bit 31: 11	Reserved	0x0000	resd	Kept at default value.
Bit 10	OOSRSEL	0x0	rw	Ordinary oversampling restart mode select When the ordinary oversampling is interrupted by preempted conversions, this bit can be used to select where to resume ordinary conversions. 0: Continuous mode (ordinary oversampling buffer will be reserved) 1: Restart mode (ordinary oversampling buffer will be cleared, that is, the previously oversampled times are reset)
Bit 9	OOSTREN	0x0	rw	Ordinary oversampling trigger mode enable 0: Disabled (only one trigger is needed for all oversampling conversions) 1: Enabled (Each oversampling conversion needs a trigger)
Bit 8: 5	OSSSEL	0x0	rw	Oversampling shift select This field is used to define the number of right shift used in the oversampling results. 0000: No shift 0001: Shift 1 bit 0010: Shift 2 bits 0011: Shift 3 bits 0100: Shift 4 bits 0101: Shift 5 bits 0110: Shift 6 bits 0111: Shift 7 bits 1000: Shift 8 bits 1001~1111: Unused. Do not configure.
Bit 4: 2	OSRSEL	0x0	rw	Oversampling ratio select 000: 2x 001: 4x 010: 8x 011: 16x 100: 32x 101: 64x 110: 128x 111: 256x
Bit 1	POSEN	0x0	rw	Preempted oversampling enable 0: Preempted oversampling disabled 1: Preempted oversampling enabled
Bit 0	OOSEN	0x0	rw	Ordinary oversampling enable 0：Ordinary oversampling disabled 1: Ordinary oversampling enabled

19 Controller area network (CAN)

19.1 CAN introduction

CAN (Controller Area Network) is a serial communication protocol for real-time and reliable data communication among various nodes. It supports the CAN protocol version 2.0A and 2.0B.

19.2 CAN main features

- Baud rates up to 1M bit/s
- Supports time-triggered communication
- Interrupt enable and mask
- Configurable automatic retransmission mode

Transmission

- Three transmit mailboxes
- Configurable transmit priority
- Supports time stamp on transmission

Reception

- Two FIFOs with three-level depth
- 14 filter banks
- Supports identifier list mode
- Supports identifier mask mode
- FIFO overflow management

Time-triggered communication mode

- 16-bit timer
- Time stamp on transmission

19.3 Baud rate

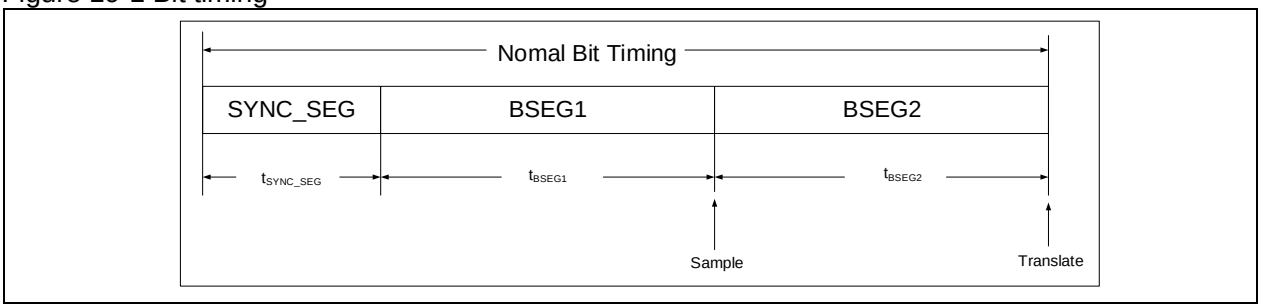
The nominal bit time of the CAN bus consists of three parts as follows:

Synchronization segment (SYNC_SEG): This segment occupies one time unit, and its time length is defined by the BRDIV[11: 0] bit in the CAN_BTMG register.

Bit segment 1 (BIT SEGMENT 1): It is referred to BSEG1 that includes PROP_SEG and PHASE_SEG1 defined in the CAN standard. Its length is configurable between 1 and 16 time units, depending on the BTS1[3: 0] bit.

Big segment 2 (BIT SEGMENT 2): It is referred to BSEG2 that includes PHASE_SEG2 defined in the CAN standard. Its length is configurable between 1 and 8 time units, depending on the BTS2[2: 0] bit.

Figure 19-1 Bit timing



Baud rate formula:

$$BaudRate = \frac{1}{\text{Nomal Bit Timimg}}$$

$$Nomal\ Bit\ Timing = t_{SYNC\_SEG} + t_{BSEG1} + t_{BSEG2}$$

with

$$t_{SYNC\_SEG} = 1 \text{ x } t_q$$

$$t_{BSEG1} = (1 + \text{BTS1}[3:0]) \text{ x } t_q$$

$$t_{BSEG2} = (1 + \text{BTS2}[2:0]) \text{ x } t_q$$

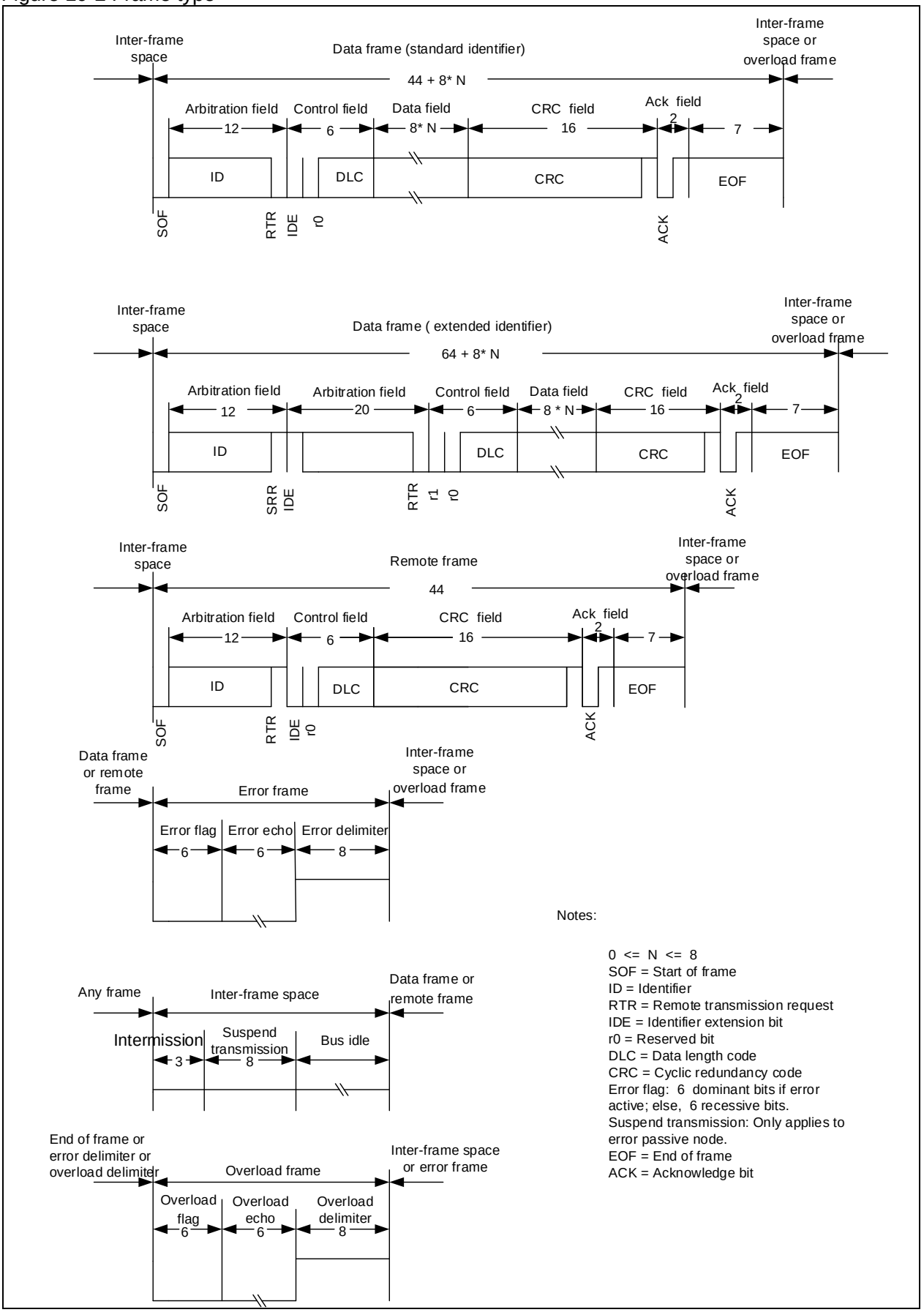
$$t_q = (1 + \text{BRDIV}[11:0]) \text{ x } t_{pclk}$$

Hard synchronization and resynchronization

The start location of each bit in CAN nodes is always in synchronization segment by default. The sampling is performed at the edge location of bit segment 1 and big segment 2 simultaneously.

During the actual transmission, there are certain phase drifts among the bits of CAN nodes due to the oscillator drifts of nodes, transmission latency and noise interference. Therefore it is necessary to avoid the impact of such phase error on communications by synchronizing or resynchronizing on the start-bit edge and the subsequent falling edges. The length of synchronization to compensate for phase drifts must not be greater than the resynchronization adjustment width (it is programmable between 1 and 4 time units through the RSAW[1:0] bit).

Figure 19-2 Frame type



Notes:

0 <= N <= 8
SOF = Start of frame
ID = Identifier
RTR = Remote transmission request
IDE = Identifier extension bit
r0 = Reserved bit
DLC = Data length code
CRC = Cyclic redundancy code
Error flag: 6 dominant bits if error active; else, 6 recessive bits.
Suspend transmission: Only applies to error passive node.
EOF = End of frame
ACK = Acknowledge bit

19.4 Interrupt management

The CAN controller has four global interrupt sources. Interrupts cam be enabled or disabled by setting the CAN_INTEN register.

Figure 19-3 Transmit interrupt generation

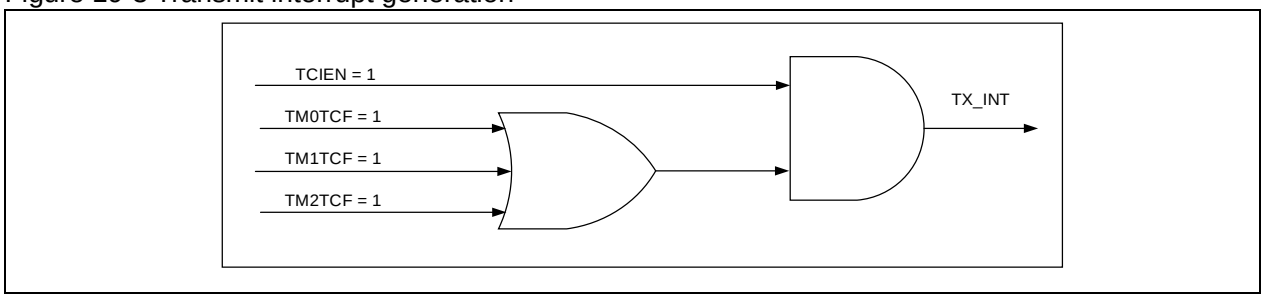


Figure 19-4 Receive interrupt 0 generation

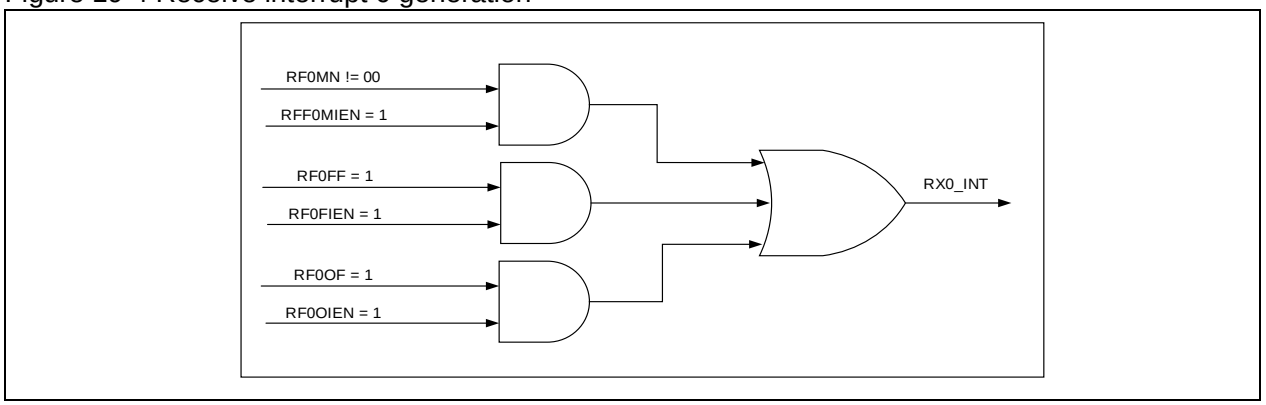


Figure 19-5 Receive interrupt 1 generation

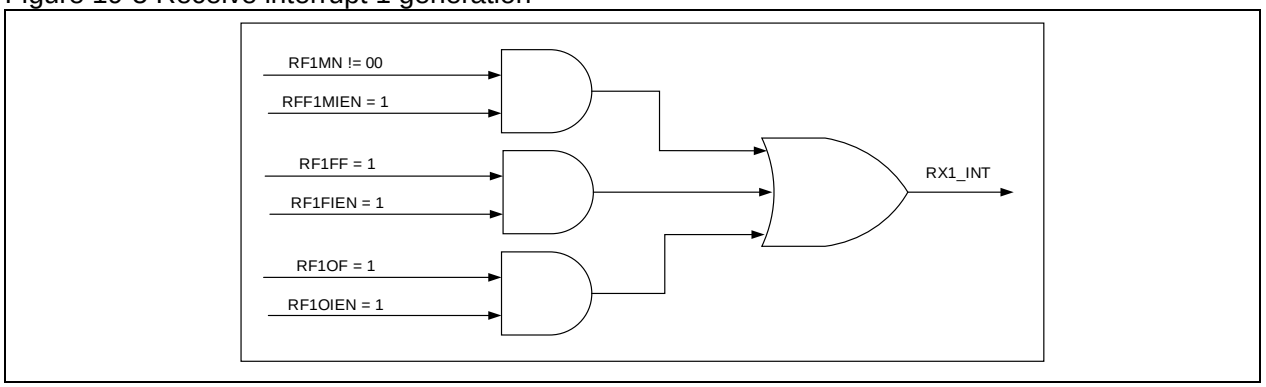
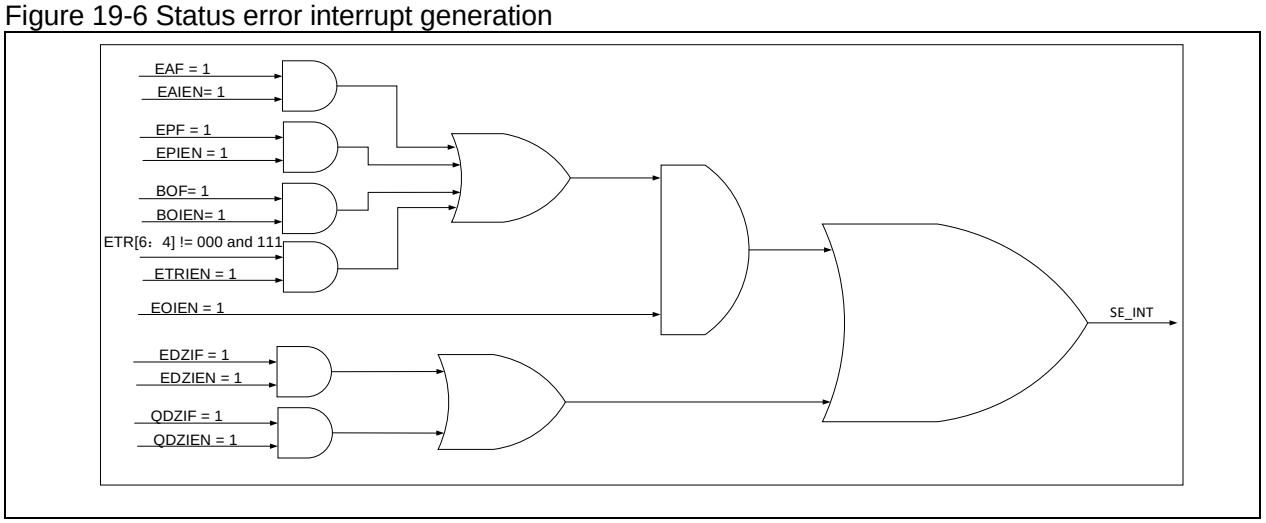


Figure 19-6 Status error interrupt generation



19.5 Design tips

The following information can be used as reference for CAN application development:

- **Debug control**
  When the system enters the debug mode, the CAN controller can be disabled (stop working) or enabled (continue working) through the CANx_PAUSE bit in the DEBUG_CTRL register and the PTD bit in the CAN_MCTRL register.
- **Time triggered communication**
  The timer triggered communication is used to improve real-time performance so as to avoid bus conflict. It is activated by setting TTCEN=1 in the CAN_MCTRL register. The internal 16-bit timer is incremented at each CAN bit time, and it is sampled on the Start of Frame bit to generate a time stamp value, which is stored in the CAN_RFCx and CAN_TMCx register.
- **Register access protection**
  The CAN_BTMG register can be modified only when the CAN is in frozen mode.

  Although the transmission of incorrect data will not affect the network level, it can have severe impact on the application. Thus this register can be modified only when a transmit mailbox is empty.
  The filter configuration in the CAN_FMCFG (filter mode configuration register), CAN_FBWCFG (filter bit width configuration register) and CAN_FRF (filter bit register) registers can be altered only when FCS=1.

  The CAN_FiFBx register can be modified only when FCS=1 (in filter configuration mode) or FAENx=0 (filter is disabled).

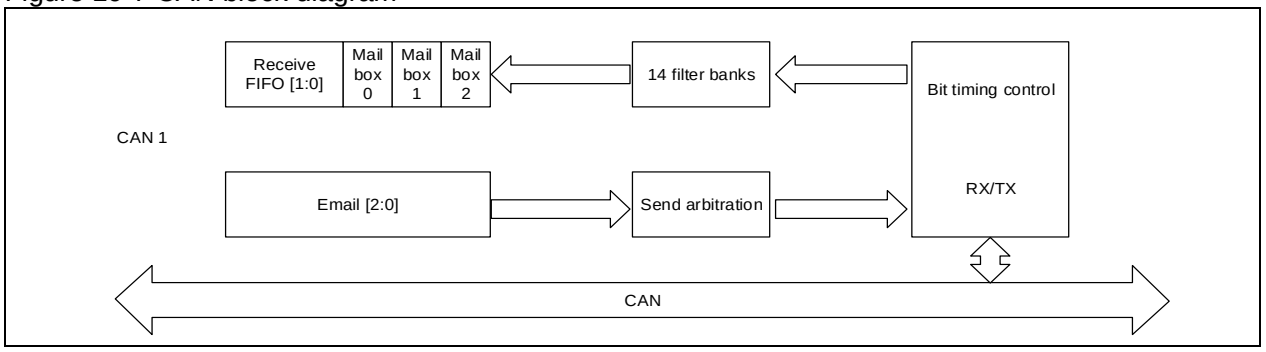
19.6 Functional overview

19.6.1 General description

As the number of nodes in the CAN network and the number of messages grows, an enhanced filtering mechanism is needed to handle all types of messages in order to reduce the processing time of message reception. One FIFO scheme is used to ensure that the CPU can concentrate on application tasks for a long period of time without message loss. In the meantime, the priority order of the messages to be transmitted is configured by hardware. Standard identifiers (11-bit) and extended identifiers (29-bit) are supported by hardware.

Based on the conditions above, the CAN controller provides 14 scalable/configurable identifier filter banks, 2 receive FIFOs, each capable of storing 3 full messages and managed by hardware. There are 3 transmit mailboxes with their transmit priority order defined by a transmit scheduler.

Figure 19-7 CAN block diagram



19.6.2 Operating modes

The CAN controller has three operating modes:

- **Sleep mode**

  After a system reset, the CAN controller is in Sleep mode. In this mode, the CAN clock is stopped to reduce power consumption and an internal pull-up resistance is disabled. However, the software can still access to mailbox registers.

  The software requests the CAN controller to enter Sleep mode by setting the DZEN bit in the CAN_MCTRL register. The hardware confirms the request by setting the DZC bit in the CAN_MSTS register.

  Exit Sleep mode in two ways:
  The CAN controller can be woke up by hardware clearing the DZEN bit when the AEDEN=1 in the CAN_MCTRL register and the CAN bus activity is detected. Sleep mode can also be woke up by software clearing the DZEN bit.

  Sleep mode to Frozen mode:
  The CAN controller switches from Sleep mode to Frozen mode when the FZEN=1 in the CAN_MCTRL register and the DZEN bit is cleared. Such switch operation is confirmed by hardware setting the FZC bit in the CAN_MSTS register.

  Sleep mode to Communication mode:
  The CAN controller enters Communication mode when the FZEN and DZEN bits are both cleared and the CAN controller has synchronized with the bus. In other words, it must wait for 11 consecutive recessive bits to be detected on the CANRX pin.

- **Frozen mode**

  The software initialization can be done only in Frozen mode, including the CAN_BTMG (CAN bit timing register) and CAN_MCTRL (CAN master control register) registers. But the initialization of the 14 CAN filter banks (mode, scale, FIFO association, activation and filter values) can be done in non-Frozen mode. When the CAN controller is in Frozen mode, message reception and transmission are both disabled.

  Frozen mode to Communication mode:
  The CAN controller leaves Frozen mode when the FZEN bit is cleared in the CAN_MCTRL register. This switch operation is confirmed by hardware clearing the FZC bit in the CAN_MSTS register. The CAN controller must be synchronized with the bus.

  Frozen mode to Sleep mode:
  The CAN controller enters Sleep mode if DZEN=1 and FZEN=0 in the CAN_MCTRL register. This switch operation is confirmed by hardware setting the DZC bit in the CAN_MSTS register.

- **Communication mode**

  After the CAN_BTMG and CAN_MCTRL registers are configured in Frozen mode, the CAN controller enters Communication mode and is ready for message reception and transmission.

  Communication mode to Sleep mode:
  The CAN controller switches to Sleep mode when the DZEN bit is set in the CAN_MCTRL register and the current CAN bus transmission is complete.

  Communication mode to Frozen mode: The CAN controller enters Frozen mode when the FZEN bit is set in the CAN_MCTRL register and the current CAN bus transmission is complete.

19.6.3 Test modes

The CAN controller defines three test modes, including Listen-only mode, Loop back mode and Listen-only combined with Loop back mode. Test mode can be selected by setting the LOEN and LBEN bits in the CAN_BTMG register.

- **Listen-only mode** is selected when the LOEN bit is set in the CAN_BTMG register. In this mode, the CAN is able to receive data, but it sends only recessive bits on the CANTX pin. In the meantime, the dominant bits on the CANTX can be detected by the receive side but without affecting the CAN bus.

- **Loop back mode** is selected by setting the LBEN bit in the CAN_BTMG register. In this mode, The CAN only receives the level signal on its CANTX pin of its own node. Meanwhile, the CAN can send data to the external bus. The Loop back mode is mainly used for self-test functions.
- **Loop back mode combined with Listen-only mode:** It is possible to combine the Listen-only and Loop back mode by setting [31:30] bits=11 in the CAN_BTMG register. In this mode, the CAN is disconnected from the bus network, the CANTX pin remains in recessive state, and the transmit side is connected to the receive side.

19.6.4 Message filtering

The received message has to go through filtering by its identifier. If passed, the message will be stored in the corresponding FIFOs. If not, the message will be discarded. The whole operation is done by hardware without using CPU resources.

Filter bit width

The CAN controller provides 14 configurable and scalable filter banks (0~13). Each filter bank has two 32-bit registers, CAN_FiFB1 and CAN_FiFB2. The filter bit width can be configured as two 16 bits or one 32 bits, depending on the corresponding bits in the CAN_FBWCFG register.

32-bit filter register CAN_FiFBx includes SID[10: 0], EID[17: 0], IDT and RTR bits.

CAN_FiFB1[31: 21]	CAN_FiFB1[20: 3]	CAN_FiFB1[2: 0]		
CAN_FiFB2[31: 21]	CAN_FiFB2[20: 3]	CAN_FiFB2[2: 0]		
SID[10: 0]/EID[28: 18]	EID[17: 0]	IDT	RTR	0

Two 16-bit filter register CAN_FiFBx includes SID[10: 0], IDT, RTR and EID[17: 15] bits

CAN_FiFB1[31: 21]	CAN_FiFB1 [20: 19]	CAN_FiFB1 [18: 16]	CAN_FiFB1[15: 5]	CAN_FiFB1 [4: 3]	CAN_FiFB1 [2: 0]	
CAN_FiFB2[31: 21]	CAN_FiFB2 [20: 19]	CAN_FiFB2 [18: 16]	CAN_FiFB2[15: 5]	AN_FiFB2 [4: 3]	CAN_FiFB2 [2: 0]	
SID[10: 0]	IDT	RTR	EID[17: 15]	SID[10: 0]	IDT	RTR	EID[17: 15]

Filtering mode

The filter can be configured in identifier mask mode or in identifier list mode by setting the FMSELx bit in the CAN_FMCFG register. The mask mode is used to specify which bits must match the pre-programmed identifiers, and which bits do not need. In identifier list mode, the identifier must match the pre-programmed identifier. The two modes can be used in conjunction with filter width to deliver four filtering modes below:

Figure 19-8 32-bit identifier mask mode

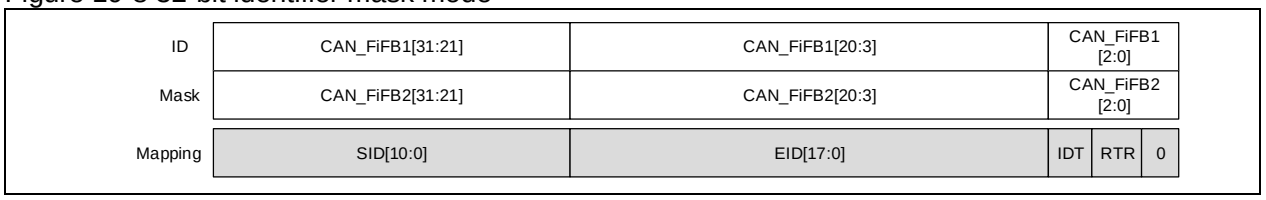


Figure 19-9 32-bit identifier list mode

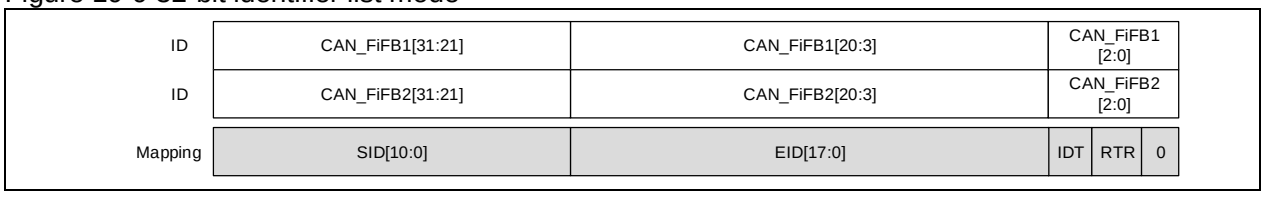


Figure 19-10 16-bit identifier mask mode

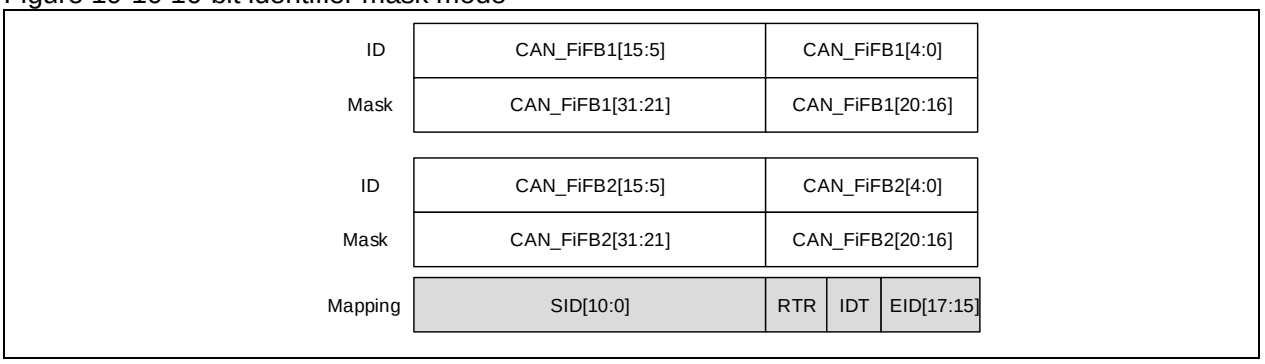
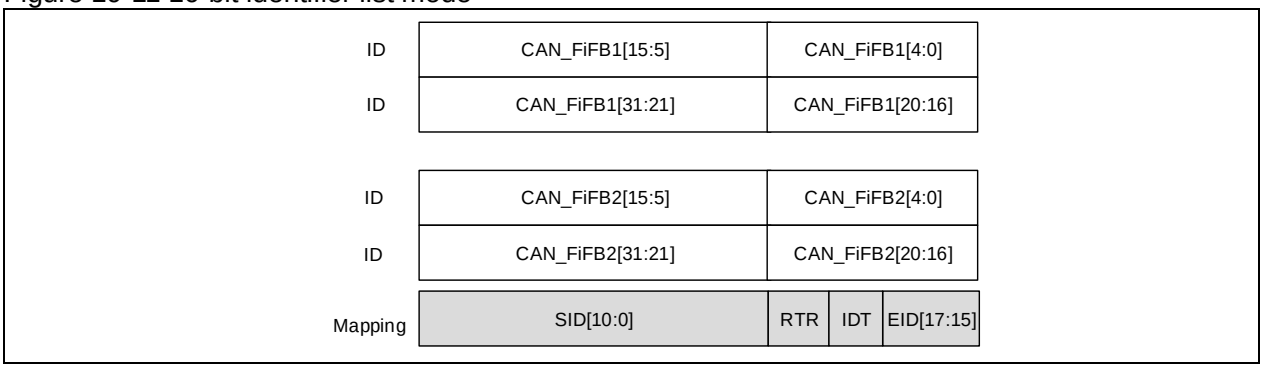


Figure 19-11 16-bit identifier list mode



Filter match number

14 filter banks have different filtering effects dependent on the bit width mode. For example, 32-bit identifier mask mode contains the filters numbered “n” while 16-bit identifier list mode contains the filters numbered “n, n+1, n+2 and n+3”. When a frame of message passes through the filter numbered “N”, the number N is stored in the RFFMN[7: 0] bit in the CAN_RFCx register. The distribution of the filter number does not take into account the activation state of the filter banks.

Filter bank	FIFO0	Active	Filter number
0	CAN_F0FB1[31:0]-ID	Yes	0
	CAN_F0FB2[31:0]-ID		1
1	CAN_F1FB1[15:0]-ID	Yes	2
	CAN_F1FB1[31:16]-ID		3
	CAN_F1FB2[15:0]-ID		4
	CAN_F1FB2[31:16]-ID		5
2	CAN_F2FB1[31:0]-ID	Yes	6
	CAN_F2FB2[31:0]-Mask		
6	CAN_F6FB1[15:0]-ID	No	7
	CAN_F6FB1[31:16]-Mask		
	CAN_F6FB2[15:0]-ID		8
	CAN_F6FB2[31:16]-Mask		
9	CAN_F9FB1[31:0]-ID	No	9
	CAN_F9FB2[31:0]-ID		10
10	CAN_F10FB1[15:0]-ID	Yes	11
	CAN_F10FB1[31:16]-Mask		
	CAN_F10FB2[15:0]-ID		12
	CAN_F10FB2[31:16]-Mask		
12	CAN_F12FB1[15:0]-ID	No	13
	CAN_F12FB1[31:16]-ID		14
	CAN_F12FB2[15:0]-ID		15
	CAN_F12FB2[31:16]-ID		16

Filter bank	FIFO1	Active	Filter number
3	CAN_F3FB1[15:0]-ID	Yes	0
	CAN_F3FB1[31:16]-ID		1
	CAN_F3FB2[15:0]-ID		2
	CAN_F3FB2[31:16]-ID		3
4	CAN_F4FB1[31:0]-ID	Yes	4
	CAN_F4FB2[31:0]-Mask		
5	CAN_F5FB1[15:0]-ID	No	5
	CAN_F5FB1[31:16]-Mask		
	CAN_F5FB2[15:0]-ID		6
	CAN_F5FB2[31:16]-Mask		
7	CAN_F7FB1[15:0]-ID	No	7
	CAN_F7FB1[31:16]-ID		8
	CAN_F7FB2[15:0]-ID		9
	CAN_F7FB2[31:16]-ID		10
8	CAN_F8FB1[31:0]-ID	Yes	11
	CAN_F8FB2[31:0]-Mask		
11	CAN_F11FB1[31:0]-ID	Yes	12
	CAN_F11FB2[31:0]-ID		13
13	CAN_F13FB1[15:0]-ID	Yes	14
	CAN_F13FB1[31:16]-ID		15
	CAN_F13FB2[15:0]-ID		16
	CAN_F13FB2[31:16]-ID		17

Priority rules

It may occur that CAN controller receives a frame of message that pass through several filters successfully. In this case, the filter match number stored in the receive mailbox is determined according to the following priority rules:

- A 32-bit filter has priority over a 16-bit filter
- For filters with equal bit width, the identifier list mode has priority over the identifier mask mode
- For filter with equal bit width and identifier mode, the lower number has priority over the higher number.

Filter configuration

- Configure CAN filter by setting the FCS bit in the CAN_FCTRL register.
- Select Identifier mask mode or identifier list mode by setting the FMSELx bit in the CAN_FMCFG register.
- Configure the filter bit width as two 16 bits or one 32 bits by setting the FBWSELx bit in the CAN_FBWCFG register.
- Associate the filter x with FIFO0 or FIFO1 by setting the FRFSELx bit in the CAN_FRF register.
- Activate the filter banks x by setting FAENx=1 in the CAN_FACFG register.
- Configure 0~13 filter banks by writing to the CAN_FiFBx register (i=0…13; x=1,2).
- Complete the CAN filter configuration by setting FCS=0 in the CAN_FCTRL register.

19.6.5 Message transmission

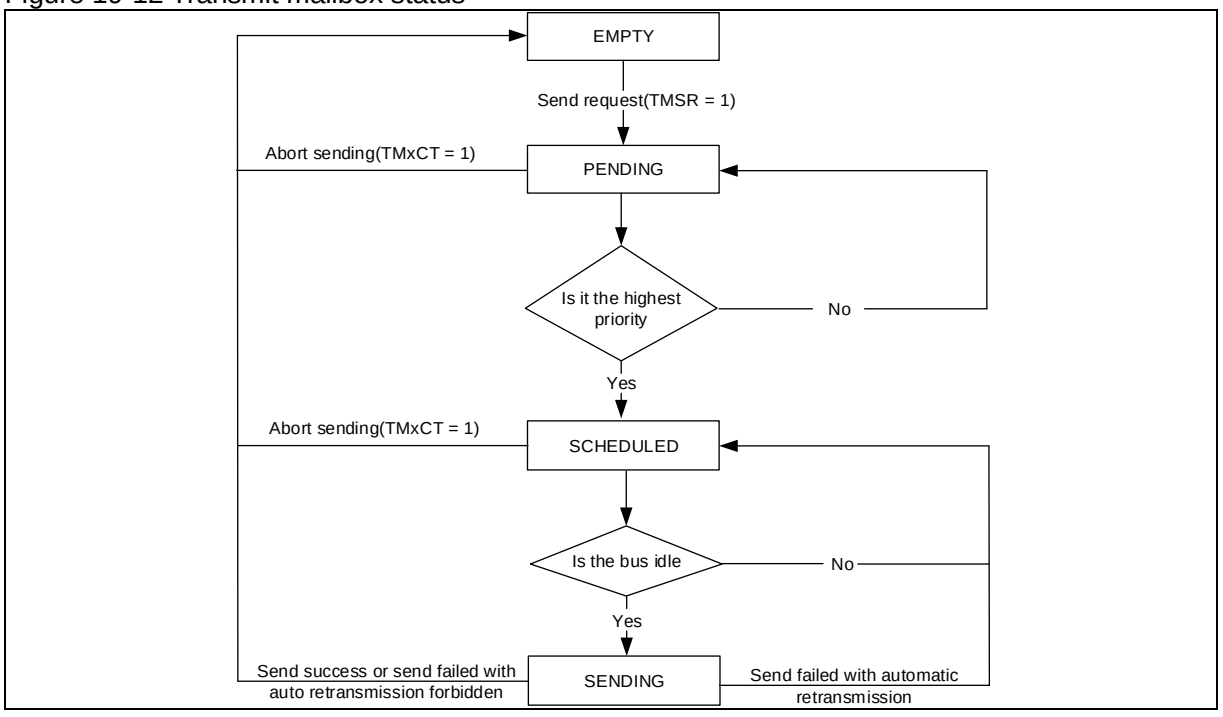
Register configuration

To transmit a message, the application must select one transmit mailbox and configure the CAN_TMIx, CAN_TMCx, CAN_TMDTLx and CAN_TMDTHx registers. Once the mailbox configuration is complete, setting the TMSR bit in the CAN_TMIx register can initiate CAN transmission.

Message transmission

The mailbox enters into pending state immediately after the mailbox is configured and the CAN controller receives a transmit request. At this point, the CAN controller will confirm whether the mailbox is given the highest priority or not. If yes, it will enter into SCHEDULED STATE, otherwise, it will wait to get the highest priority. The mailbox in SCHEDULED state will monitor the CAN bus state so that the messages in SCHEDULED mailbox can be transmitted as soon as the CAN bus becomes idle. The mailbox will enter EMPTY state at the end of the message transmission.

Figure 19-12 Transmit mailbox status



Transmit priority configuration

When two or more transmit boxes are in PENDING state, their transmit priority must be given.

By identifier:

When MMSSR=0 in the CAN_MCTRL register, the transmit order is defined by the identifier of the message in the mailbox. The message with lower identifier value has the highest priority. If the identifier values are the same, the message with lower mailbox number will be transmitted first.

By transmit request order:

When MMSSR=1 in the CAN_MCTRL register, the transmit priority is given by the transmit request order of mailboxes.

Transmit status and error status

The TMxTCF, TMxTSF, TMxALF, TMxTEF and TMxEF bits in the CAN_TSTS register are used to indicate transmit status and error status.

TMxTCF bit: Transmission complete flag, indicating that the data transmission is complete when TMxTCF=1.

TMxTSF bit: Transmission success flag, indicating that the data has been transmitted successfully when TMxTSF =1.

TMxALF bit: Transmission arbitration lost flag, indicating that the data transmission arbitration is lost when TMxALF=1.

TMxTEF bit: Transmission error flag, indicating that the data transmission failed due to bus error, and an error frame is sent when TMxTEF=1.

TMxEF bit: Mailbox empty flag, indicating that the data transmission is complete and the mailbox becomes empty when TMxEF=1.

Transmit abort

The TMxCT bit is set in the CAN_TSTS register to abort the transmission of the current mailbox, detailed as follows:

When the current transmission fails or arbitration is lost, if the automatic retransmission mode is disabled, the transmit mailbox become EMPTY; if the automatic retransmission mode is enabled, the transmit mailbox becomes SCHEDULED, the mailbox transmission then is aborted and becomes EMPTY.

When the current transmission is complete successfully, the mailbox becomes EMPTY.

19.6.6 Message reception

Register configuration

The CAN_RFIx (receive FIFO mailbox identifier register), CAN_RFCx (receive FIFO mailbox data length and time stamp register), CAN_RFDTLx (receive FIFO mailbox data register low) and CAN_RFDTHx (receive FIFO mailbox data register high) registers can be used by user applications to obtain valid messages.

Message reception

The CAN controller has two FIFO with three levels to receive messages. FIFO rule is adopted. When the message is received correctly and has passed the identifier filtering, it is considered as a valid message and is stored in the corresponding FIFO. The number of the received messages RFxMN[1: 0] will be incremented by one whenever the receive FIFO receives a valid message. If a valid message is received when RFxMN[1: 0]=3, the controller will select either to overwrite the previous messages or discard the new incoming message through the MDRSEL bit in the CAN_MCTRL register.

In the meantime, when the user reads a frame of message and the RFxR is set in the CAN_RFx register, one FIFO mailbox is released, and RFxMN[1: 0] bit is decremented by one in the CAN_RFx register.

Receive FIFO status

RFxMN[1: 0], RFxFF and RFxOF bits in the RFx register are used to indicate the status of receive FIFO.

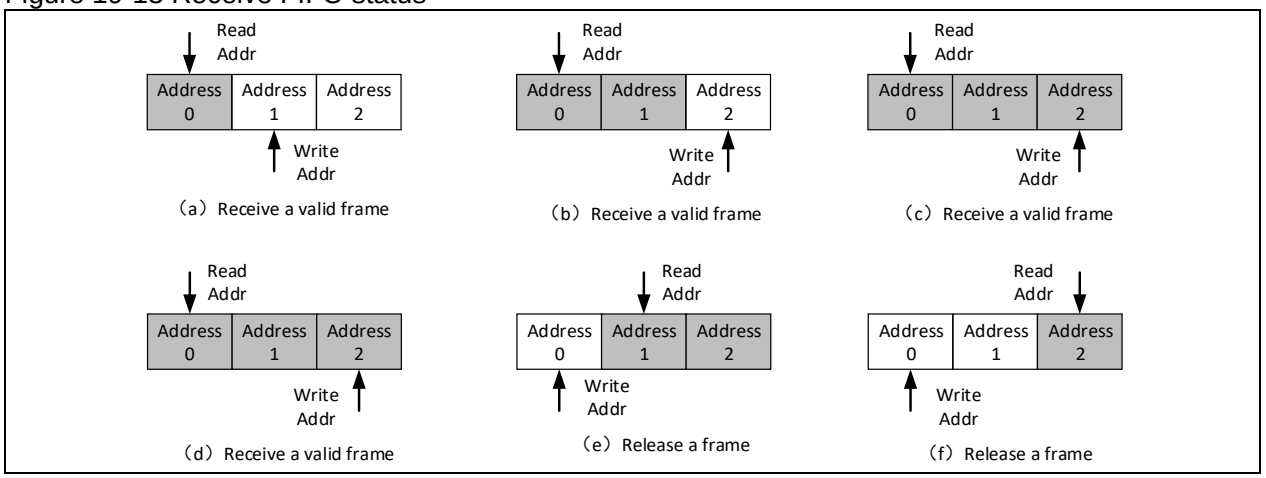
RFxMN[1: 0]: indicates the number of valid messages stored in the FIFOx.

RFxFF: indicates that three valid messages are stored in the FIFOx (i.e. the three mailboxes are full), as shown in (c) of *Figure 19-13*.

RFxOF: indicates that a new valid message has been received while the FIFOx is full, as shown in (d)

of *Figure 19-13*.

Figure 19-13 Receive FIFO status



19.6.7 Error management

The status of the current CAN node is indicated by the receive error counter (TEC) and transmit error counter (REC) bits in the CAN_ESTS register. The ETR[2: 0] bit in the CAN_ESTS register is used to record the last error source, and the corresponding interrupts will be generated when the CAN_INTEN register is enabled.

- **Error active flag:** When both TEC and REC are lower than 128, the system is in the error active state. An error active flag is set when an error is detected.
- **Error passive flag:** When either TEC or REC is greater than 127, the system is in the error passive state. An error passive flag is set when an error is detected.
- **Bus-off state:** The bus-off state is entered when TEC is greater than 255. In bus-off state, the node is not able to transmit and receive messages. The CAN recovers from bus-off state in two ways:

  Option 1:
  When AEBOEN=0 in the CAN_MCTRL register, in communication mode, the CAN will recover from bus-off state when the 128 occurrence of 11 consecutive recessive bits are detected on the CAN RX pin, and the software requests to enter Frozen mode and exit Frozen mode.

  Option 2:
  When AEBOEN=1 in the CAN_MCTRL register, in communication mode, the CAN will resume from bus-off state automatically after 128 occurrences of 11 consecutive recessive bits have been detected on the CAN RX pin.

19.7 CAN registers

These peripheral registers must be accessed by words (32 bits).

Table 19-1 CAN register map and reset values

Register name	Offset	Reset value
MCTRL	000h	0x0001 0002
MSTS	004h	0x0000 0C02
TSTS	008h	0x1C00 0000
RF0	00Ch	0x0000 0000
FR1	010h	0x0000 0000
INTEN	014h	0x0000 0000
ESTS	018h	0x0000 0000
BTMG	01Ch	0x0123 0000
Reserved	020h~17Fh	xx

TMI0	180h	0xXXXX XXXX
TMC0	184h	0xXXXX XXXX
TMDTL0	188h	0xXXXX XXXX
TMDTH0	18Ch	0xXXXX XXXX
TMI1	190h	0xXXXX XXXX
TMC1	194h	0xXXXX XXXX
TMDTL1	198h	0xXXXX XXXX
TMDTH1	19Ch	0xXXXX XXXX
TMI2	1A0h	0xXXXX XXXX
TMC2	1A4h	0xXXXX XXXX
TMDTL2	1A8h	0xXXXX XXXX
TMDTH2	1ACh	0xXXXX XXXX
RFI0	1B0h	0xXXXX XXXX
RFC0	1B4h	0xXXXX XXXX
RFDTL0	1B8h	0xXXXX XXXX
RFDTH0	1BCh	0xXXXX XXXX
RFI1	1C0h	0xXXXX XXXX
RFC1	1C4h	0xXXXX XXXX
RFDTL1	1C8h	0xXXXX XXXX
RFDTH1	1CCh	0xXXXX XXXX
Reserved	1D0h~1FFh	xx
FCTRL	200h	0x2A1C 0E01
FBWCFG	204h	0x0000 0000
Reserved	208h	xx
FSCFG	20Ch	0x0000 0000
Reserved	210h	xx
FRF	214h	0x0000 0000
Reserved	218h	xx
FACFG	21Ch	0x0000 0000
Reserved	220h~23Fh	xx
F0FB1	240h	0xXXXX XXXX
F0FB2	244h	0xXXXX XXXX
F1FB1	248h	0xXXXX XXXX
F1FB2	24Ch	0xXXXX XXXX
…	…	...
F13FB1	2A8h	0xXXXX XXXX
F13FB2	2ACh	0xXXXX XXXX

19.7.1 CAN control and status registers

19.7.1.1 CAN master control register (CAN_MCTRL)

Bit	Name	Reset value	Type	Description
Bit 31: 17	Reserved	0x0000	resd	Kept at default value.
Bit 16	PTD	0x1	rw	Prohibit transceive when debug 0: Transmission works during debug 1: Transmission is prohibited during debug. Receive-write and control FIFO can be still accessible. Note: Transmission can be disabled only when PTD and CANx_PAUSE bits in the DEBUG_CTRL register are set at the same time.
Bit 15	SPRST	0x0	rw1s	Software partial reset 0: Normal (No reset) 1: Software partial reset Note: SPRST only resets receive FIFO and MCTRL register. The CAN enters Sleep mode after reset. Then this bit is automatically cleared by hardware.
Bit 14: 8	Reserved	0x00	resd	Kept at default value.
Bit 7	TTCEN	0x0	rw	Time triggered communication mode enable 0: Time triggered communication mode disabled 1: Time triggered communication mode enabled
Bit 6	AEBOEN	0x0	rw	Automatic exit bus-off enable 0: Automatic exit bus-off disabled 1: Automatic exit bus-off enabled Note: When Automatic exit bus-off mode is enabled, the hardware will automatically leave bus-off mode as soon as an exit timing is detected on the CAN bus. When Automatic exit bus-off mode is disabled, the software must enter/leave the freeze mode once more, and then the bus-off state is left only when an exit timing is detected on the CAN bus.
Bit 5	AEDEN	0x0	rw	Automatic exit doze mode enable 0: Automatic exit sleep mode disabled 1: Automatic exit sleep mode enabled Note: When Automatic exit sleep mode is disabled, the Sleep mode is left by software clearing the sleep request command. When Automatic exit sleep mode is enabled, the sleep mode is left without the need of software intervention as soon as a message is monitored on the CAN bus.
Bit 4	PRSFEN	0x0	rw	Prohibit retransmission enable when sending fails enable 0: Retransmission is enabled. 1: Retransmission is disabled.
Bit 3	MDRSEL	0x0	rw	Message discard rule select when overflow 0: The previous message is discarded. 1: The new incoming message is discarded.
Bit 2	MMSSR	0x0	rw	Multiple message transmit sequence rule 0: The message with the smallest identifier is first transmitted. 1: The message with the first request order is first transmitted.
Bit 1	DZEN	0x1	rw	Doze mode enable 0: Sleep mode is disabled. 1: Sleep mode is enabled. Note: The hardware will automatically leave sleep mode when the AEDEN is set and a message is monitored on the CAN bus. After CAN reset or partial software reset, this bit is forced

				to be set by hardware, that is, the CAN will keep in sleep mode, by default.
Bit 0	FZEN	0x0	rw	Freeze mode enable 0: Freeze mode disabled 1: Freeze mode enabled Note: The CAN leaves Freeze mode once 11 consecutive recessive bits have been detected on the RX pin. For this reason, the software acknowledges the entry of Freeze mode after the FZC bit is cleared by hardware. The Freeze mode is entered only when the current CAN activity (transmission or reception) is completed. Thus the software acknowledges the exit of Freeze mode after the FZC bit is cleared by hardware.

19.7.1.2 CAN master status register (CAN_MSTS)

Bit	Name	Reset value	Type	Description
Bit 31: 12	Reserved	0x00000	resd	Kept at default value.
Bit 11	REALRX	0x1	ro	Real time level on RX pin 0: Low 1: High
Bit 10	LSAMPRX	0x1	ro	Last sample level on RX pin 0: Low 1: High Note: This value keeps updating with the REALRX.
Bit 9	CURS	0x0	ro	Current receive status 0: No reception occurred 1: Reception is in progress Note: This bit is set by hardware when the CAN reception starts, and it is cleared by hardware at the end of reception.
Bit 8	CUSS	0x0	ro	Current transmit status 0: No transmit occurred 1: Transmit is in progress Note: This bit is set by hardware when the CAN transmission starts, and it is cleared by hardware at the end of transmission.
Bit 7: 5	Reserved	0x0	resd	Kept at default value.
Bit 4	EDZIF	0x0	rw1c	Enter doze mode interrupt flag 0: Sleep mode is not entered or no condition for flag set. 1: Sleep mode is entered. Note: This bit is set by hardware only when EDZIEN=1 and the CAN enters Sleep mode. When set, this bit will generate a status change interrupt. This bit is cleared by software (writing 1 to itself) or by hardware when DZC is cleared.
Bit 3	QDZIF	0x0	rw1c	Exit doze mode interrupt flag 0: Sleep mode is not left or no condition for exit. 1: Sleep mode has been left or exit condition has generated. Note: This bit is cleared by software (writing 1 to itself) Sleep mode is left when a SOF is detected on the bus. When QDZIEN=1, this bit will generate a status change interrupt.
Bit 2	EOIF	0x0	rw1c	Error occur interrupt flag 0: No error interrupt or no condition for error interrupt flag 1: Error interrupt is generated. Note: This bit is cleared by software (writing 1 to itself). This bit is set by hardware only when the corresponding bit is set in the CAN_ESTS register and the corresponding interrupt enable bit in the CAN_INTEN register is enabled. When set, this bit will generate a

				status change interrupt.
Bit 1	DZC	0x1	ro	Doze mode acknowledge 0: The CAN is not in Sleep mode. 1: CAN is in Sleep mode. Note: This bit is used to decide whether the CAN is in Sleep mode or not. This bit acknowledges the Sleep mode request generated by software. The Sleep mode can be entered only when the current CAN activity (transmission or reception) is completed. For this reason, the software acknowledges the entry of Sleep mode after this bit is set by hardware. The Sleep mode is left only once 11 consecutive recessive bits have been detect on the CAN RX pin. For this reason, the software acknowledges the exit of Sleep mode after this bit is cleared by hardware.
Bit 0	FZC	0x0	ro	Freeze mode confirm 0: The CAN is not in Freeze mode. 1: The CAN is in Freeze mode. Note: This bit is used to decide whether the CAN is in Freeze mode or not. This bit acknowledges the Freeze mode request generated by software. The Freeze mode can be entered only when the current CAN activity (transmission or reception) is completed. For this reason, the software acknowledges the entry of Freeze mode after this bit is set by hardware. The Freeze mode is left only once 11 consecutive recessive bits have been detect on the CAN RX pin. For this reason, the software acknowledges the exit of Freeze mode after this bit is cleared by hardware.

19.7.1.3 CAN transmit status register (CAN_TSTS)

Bit	Name	Reset value	Type	Description
Bit 31	TM2LPF	0x0	ro	Transmit mailbox 2 lowest priority flag 0: Mailbox 2 is not given the lowest priority. 1: Lowest priority (This indicates that more than one mailboxes are pending for transmission, the mailbox 2 has the lowest priority.)
Bit 30	TM1LPF	0x0	ro	Transmit mailbox 1 lowest priority flag 0: Mailbox 1 is not given the lowest priority. 1: Lowest priority (This indicates that more than one mailboxes are pending for transmission, the mailbox 1 has the lowest priority.)
Bit 29	TM0LPF	0x0	ro	Transmit mailbox 0 lowest priority flag 0: Mailbox 0 is not given the lowest priority. 1: Lowest priority (This indicates that more than one mailboxes are pending for transmission, the mailbox 0 has the lowest priority.)
Bit 28	TM2EF	0x1	ro	Transmit mailbox 2 empty flag This bit is set by hardware when no transmission is pending in the mailbox 2.
Bit 27	TM1EF	0x1	ro	Transmit mailbox 1 empty flag This bit is set by hardware when no transmission is pending in the mailbox 1.
Bit 26	TM0EF	0x1	ro	Transmit mailbox 0 empty flag This bit is set by hardware when no transmission is pending in the mailbox 0.
Bit 25: 24	TMNR	0x0	ro	Transmit Mailbox number record Note: If the transmit mailbox is free, these two bits refer to the number of the next transmit mailbox free. For example, in case of free CAN, the value of these two bit becomes 01 after a message transmit request is

				written. If the transmit box is full, these two bits refer to the number of the transmit mailbox with the lowest priority. For example, when there are three messages are pending for transmission, the identifiers of mailbox 0, mailbox 1 and mailbox 2 are 0x400, 0x433 and 0x411 respectively, and the value of these two bits becomes 01.
Bit 23	TM2CT	0x0	ro	Transmit mailbox 2 cancel transmit 0: No effect 1: Transmission is cancelled. Note: Software sets this bit to abort the transmission of mailbox 2. This bit is cleared by hardware when the transmit message in the mailbox 2 is cleared. Setting this bit has no effect if the mailbox 2 is free.
Bit 22: 20	Reserved	0x0	resd	Kept at default value.
Bit 19	TM2TEF	0x0	rw1c	Transmit mailbox 2 transmission error flag 0: No error 1: Mailbox 2 transmission error Note: This bit is set when the mailbox 2 transmission error occurred. It is cleared by software writing 1 or by hardware at the start of the next transmission
Bit 18	TM2ALF	0x0	rw1c	Transmit mailbox 2 arbitration lost flag 0: No arbitration lost 1: Transmit mailbox 2 arbitration lost Note: This bit is set when the mailbox 2 transmission failed due to an arbitration lost. It is cleared by software writing 1 or by hardware at the start of the next transmission
Bit 17	TM2TSF	0x0	rw1c	Transmit mailbox 2 transmission success flag 0: Transmission failed 1: Transmission was successful. Note: This bit indicates whether the mailbox 2 transmission is successful or not. It is cleared by software writing 1.
Bit 16	TM2TCF	0x0	rw1c	Transmit mailbox 2 transmission completed flag 0: Transmission is in progress 1: Transmission is completed Note: This bit is set by hardware when the transmission/abort request on mailbox 2 has been completed. It is cleared by software writing 1 or by hardware when a new transmission request is received. Clearing this bit will clear the TM2TSF, TM2ALF and TM2TEF bits of mailbox 2 by hardware
Bit 15	TM1CT	0x0	rw1s	Transmit mailbox 1 cancel transmit 0: No effect 1: Mailbox 1 cancel transmit Note: This bit is set by software to abort the transmission request on mailbox 1. Clearing the message transmission on mailbox 1 will clear this bit. Setting by this software has no effect when the mailbox 1 is free.
Bit 14: 12	Reserved	0x0	resd	Kept at default value.
Bit 11	TM1TEF	0x0	rw1c	Transmit mailbox 1 transmission error flag 0: No error 1: Mailbox 1 transmission error Note: This bit is set when the mailbox 1 transmission error occurred. It is cleared by software writing 1 or by hardware at the start of the next transmission
Bit 10	TM1ALF	0x0	rw1c	Transmit mailbox 1 arbitration lost flag

				0: No arbitration lost 1: Transmit mailbox 1 arbitration lost Note: This bit is set when the mailbox 1 transmission failed due to an arbitration lost. It is cleared by software writing 1 or by hardware at the start of the next transmission
Bit 9	TM1TSF	0x0	rw1c	Transmit mailbox 1 transmission success flag 0: Transmission failed 1: Transmission was successful. Note: This bit indicates whether the mailbox 1 transmission is successful or not. It is cleared by software writing 1.
Bit 8	TM1TCF	0x0	rw1c	Transmit mailbox 1 transmission completed flag 0: Transmission is in progress 1: Transmission is completed Note: This bit is set by hardware when the transmission/abort request on mailbox 1 has been completed. It is cleared by software writing 1 or by hardware when a new transmission request is received. Clearing this bit will clear the TM1TSF, TM1ALF and TM1TEF bits of mailbox 1.
Bit 7	TM0CT	0x0	rw1s	Transmit mailbox 0 cancel transmit 0: No effect 1: Mailbox 0 cancel transmit Note: This bit is set by software to abort the transmission request on mailbox 0. Clearing the message transmission on mailbox 0 will clear this bit. Setting by this software has no effect when the mailbox 0 is free.
Bit 6: 4	Reserved	0x0	resd	Kept at default value.
Bit 3	TM0TEF	0x0	rw1c	Transmit mailbox 0 transmission error flag 0: No error 1: Mailbox 0 transmission error Note: This bit is set when the mailbox 0 transmission error occurred. It is cleared by software writing 0 or by hardware at the start of the next transmission
Bit 2	TM0ALF	0x0	rw1c	Transmit mailbox 0 arbitration lost flag 0: No arbitration lost 1: Transmit mailbox 0 arbitration lost Note: This bit is set when the mailbox 0 transmission failed due to an arbitration lost. It is cleared by software writing 1 or by hardware at the start of the next transmission
Bit 1	TM0TSF	0x0	rw1c	Transmit mailbox 0 transmission success flag 0: Transmission failed 1: Transmission was successful. Note: This bit indicates whether the mailbox 0 transmission is successful or not. It is cleared by software writing 1.
Bit 0	TM0TCF	0x0	rw1c	Transmit mailbox 0 transmission completed flag 0: Transmission is in progress 1: Transmission is completed Note: This bit is set by hardware when the transmission/abort request on mailbox 0 has been completed. It is cleared by software writing 1 or by hardware when a new transmission request is received. Clearing this bit will clear the TM0TSF, TM0ALF and TM0TEF bits of mailbox 0.

19.7.1.4 CAN receive FIFO 0 register (CAN_RF0)

Bit	Name	Reset value	Type	Description
Bit 31: 6	Reserved	0x0000000	resd	Kept at default value.
Bit 5	RF0R	0x0	rw1s	Receive FIFO 0 release 0: No effect 1: Release FIFO Note: This bit is set by software to release FIFO 0. It is cleared by hardware when the FIFO 0 is released. Setting this bit by software has no effect when the FIFO 0 is empty. If there are more than two messages pending in the FIFO 0, the software has to release the FIFO 0 to access the second message.
Bit 4	RF0OF	0x0	rw1c	Receive FIFO 0 overflow flag 0: No overflow 1: Receive FIFO 0 overflow Note: This bit is set by hardware when a new message has been received and passed the filter while the FIFO 0 is full. It is cleared by software by writing 1.
Bit 3	RF0FF	0x0	rw1c	Receive FIFO 0 full flag 0: Receive FIFO 0 is not full 1: Receive FIFO 0 is full Note: This bit is set by hardware when there are three messages pending in the FIFO 0. It is cleared by software by writing 1.
Bit 2	Reserved	0x0	resd	Kept at default value.
Bit 1: 0	RF0MN	0x0	ro	Receive FIFO 0 message num Note: These two bits indicate how many messages are pending in the FIFO 0. RF0ML bit is incremented by one each time a new message has been received and passed the filter while the FIFO 0 is not full. RF0ML bit is decremented by one each time the software releases the receive FIFO 0 by writing 1 to the RF0R bit.

19.7.1.5 CAN receive FIFO 1 register (CAN_RF1)

Bit	Name	Reset value	Type	Description
Bit 31: 6	Reserved	0x0000000	resd	Kept at default value.
Bit 5	RF1R	0x0	rw1s	Receive FIFO 1 release 0: No effect 1: Release FIFO Note: This bit is set by software to release receive FIFO 1. It is cleared by hardware when the FIFO 1 is released. Setting this bit by software has no effect when the FIFO 1 is empty. If there are more than two messages pending in the FIFO 0, the software has to release the FIFO 1 to access the second message.
Bit 4	RF1OF	0x0	rw1c	Receive FIFO 1 overflow flag 0: No overflow 1: Receive FIFO 1 overflow Note: This bit is set by hardware when a new message has been received and passed the filter while the FIFO 1 is full. It is cleared by software by writing 1.
Bit 3	RF1FF	0x0	rw1c	Receive FIFO 1 full flag

				0: Receive FIFO 1 is not full 1: Receive FIFO 1 is full Note: This bit is set by hardware when three messages are pending in the FIFO 1. It is cleared by software by writing 1.
Bit 2	Reserved	0x0	resd	Kept at default value.
Bit 1: 0	RF1MN	0x0	ro	Receive FIFO 1 message num Note: These two bits indicate how many messages are pending in the FIFO 1. RF1ML bit is incremented by one each time a new message has been received and passed the filter while the FIFO 1 is not full. RF1ML bit is decremented by one each time the software releases the receive FIFO 1 by writing 1 to the RF1R bit.

19.7.1.6 CAN interrupt enable register (CAN_INTEN)

Bit	Name	Reset value	Type	Description
Bit 31: 18	Reserved	0x0000	resd	Kept at default value.
Bit 17	EDZIEN	0x0	rw	Enter doze mode interrupt enable 0: Enter sleep mode interrupt disabled 1: Enter sleep mode interrupt enabled Note: EDZIF flag bit corresponds to this interrupt. An interrupt is generated when both this bit and EDZIF bit are set.
Bit 16	QDZIEN	0x0	rw	Quit doze mode interrupt enable 0: Quit sleep mode interrupt disabled 1: Quit sleep mode interrupt enabled Note: The flag bit of this interrupt is the QDZIF bit. An interrupt is generated when both this bit and QDZIF bit are set.
Bit 15	EOIEN	0x0	rw	Error occur interrupt enable 0: Error interrupt disabled 1: Error interrupt enabled Note: The flag bit of this interrupt is the EOIF bit. An interrupt is generated when both this bit and EOIF bit are set.
Bit 14: 12	Reserved	0x0	resd	Kept at default value.
Bit 11	ETRIEN	0x0	rw	Error type record interrupt enable 0: Error type record interrupt disabled 1: Error type record interrupt enabled Note: EOIF is set only when this interrupt is enabled and the ETR[2: 0] is set by hardware.
Bit 10	BOIEN	0x0	rw	Bus-off interrupt enable 0: Bus-off interrupt disabled 1: Bus-off interrupt enabled Note: EOIF is set only when this interrupt is enabled and the BOF is set by hardware.
Bit 9	EPIEN	0x0	rw	Error passive interrupt enable 0: Error passive interrupt disabled 1: Error passive interrupt enabled Note: EOIF is set only when this interrupt is enabled and the EPF is set by hardware.
Bit 8	EAIEN	0x0	rw	Error active interrupt enable 0: Error warning interrupt disabled 1: Error warning interrupt enabled Note: EOIF is set only when this interrupt is enabled and the EAF is set by hardware.
Bit 7	Reserved	0x0	resd	Kept at default value.
Bit 6	RF1OIEN	0x0	rw	Receive FIFO 1 overflow interrupt enable 0: Receive FIFO 1 overflow interrupt disabled 1: Receive FIFO 1 overflow interrupt enabled Note: The flag bit of this interrupt is the RF1OF bit. An

				interrupt is generated when this bit and RF1OF bit are set.
Bit 5	RF1FIEN	0x0	rw	Receive FIFO 1 full interrupt enable 0: Receive FIFO 1 full interrupt disabled 1: Receive FIFO 1 full interrupt enabled Note: The flag bit of this interrupt is the RF1FF bit. An interrupt is generated when this bit and RF1FF bit are set.
Bit 4	RF1MIEN	0x0	rw	FIFO 1 receive message interrupt enable 0: FIFO 1 receive message interrupt disabled 1: FIFO 1 receive message interrupt enabled Note: The flag bit of this interrupt is RF1MN bit, so an interrupt is generated when this bit and RF1MN bit are set.
Bit 3	RF0OIEN	0x0	rw	Receive FIFO 0 overflow interrupt enable 0: Receive FIFO 0 overflow interrupt disabled 1: Receive FIFO 0 overflow interrupt enabled Note: The flag bit of this interrupt is RF0OF bit, so an interrupt is generated when this bit and RF0OF bit are set.
Bit 2	RF0FIEN	0x0	rw	Receive FIFO 0 full interrupt enable 0: Receive FIFO 0 full interrupt disabled 1: Receive FIFO 0 full interrupt enabled Note: The flag bit of this interrupt is the RF0FF bit. An interrupt is generated when this bit and RF0FF bit are set
Bit 1	RF0MIEN	0x0	rw	FIFO 0 receive message interrupt enable 0: FIFO 0 receive message interrupt disabled 1: FIFO 0 receive message interrupt enabled Note: The flag bit of this interrupt is the RF0MN bit. An interrupt is generated when this bit and RF0MN bit are set
Bit 0	TCIEN	0x0	rw	Transmit mailbox empty interrupt enable 0: Transmit mailbox empty interrupt disabled 1: Transmit mailbox empty interrupt enabled Note: The flag bit of this interrupt is the TMxTCF bit. An interrupt is generated when this bit and TMxTCF bit are set

19.7.1.7 CAN error status register (CAN_ESTS)

Bit	Name	Reset value	Type	Description
Bit 31: 24	REC	0x00	ro	Receive error counter This counter is implemented in accordance with the receive part of the fault confinement mechanism of the CAN protocol.
Bit 23: 16	TEC	0x00	ro	Transmit error counter This counter is implemented in accordance with the transmit part of the fault confinement mechanism of the CAN protocol.
Bit 15: 7	Reserved	0x00	resd	Kept at default value.
Bit 6: 4	ETR	0x0	rw	Error type record 000: No error 001: Bit stuffing error 010: Format error 011: Acknowledgement error 100: Recessive bit error 101: Dominant bit error 110: CRC error 111: Set by software Note: This field is used to indicate the current error type. It is set by hardware according to the error condition detected on the CAN bus. It is cleared by hardware when a message has been transmitted or received successfully. If the error code 7 is not used by hardware, this field can be set by software to monitor the code update.
Bit 3	Reserved	0x0	resd	Kept at default value.
Bit 2	BOF	0x0	ro	Bus-off flag 0: Bus-off state is not entered. 1: Bus-off state is entered. Note: When the TEC is greater than 255, the bus-off state is entered, and this bit is set by hardware.
Bit 1	EPF	0x0	ro	Error passive flag 0: Error passive state is not entered 1: Error passive state is entered Note: This bit is set by hardware when the current error times has reached the Error passive state limit (Receive Error Counter or Transmit Error Counter >127)
Bit 0	EAF	0x0	ro	Error active flag 0: Error active state is not entered 1: Error active state is entered Note: This bit is set by hardware when the current error times has reached the Error active state limit (Receive Error Counter or Transmit Error Counter ≥96)

19.7.1.8 CAN bit timing register (CAN_BTMG)

Bit	Name	Reset value	Type	Description
Bit 31	LOEN	0x0	rw	Listen-Only mode 0: Listen-Only mode disabled 1: Listen-Only mode enabled
Bit 30	LBEN	0x0	rw	Loop back mode 0: Loop back mode disabled 1: Loop back mode enabled
Bit 29: 26	Reserved	0x0	resd	Kept at default value.
Bit 25: 24	RSAW	0x1	rw	Resynchronization width tRSAW = tCAN x (RSAW[1: 0] + 1) Note: This field defines the maximum of time unit that the CAN hardware is allowed to lengthen or shorten in a bit.
Bit 23	Reserved	0x0	resd	Kept at default value.
Bit 22: 20	BTS2	0x2	rw	Bit time segment 2 tBTS2 = tCAN x (BTS2[2: 0] + 1) Note: This field defines the number of time unit in Bit time

				segment 2.
Bit 19: 16	BTS1	0x3	rw	Bit time segment 1 tBTS1 = tCAN x (BTS1[3: 0] + 1) Note: This field defines the number of time unit in Bit time segment 1.
Bit 15: 12	Reserved	0x0	resd	Kept at default value.
Bit 11: 0	BRDIV	0x000	rw	Baud rate division tq = (BRDIV[11: 0]+1) x tPCLK Note: This field defines the length of a time unit (tq).

19.7.2 CAN mailbox registers

This section describes the registers of the transmit and receive mailboxes. Refer to *Section 19.6.5* for more information on register map.

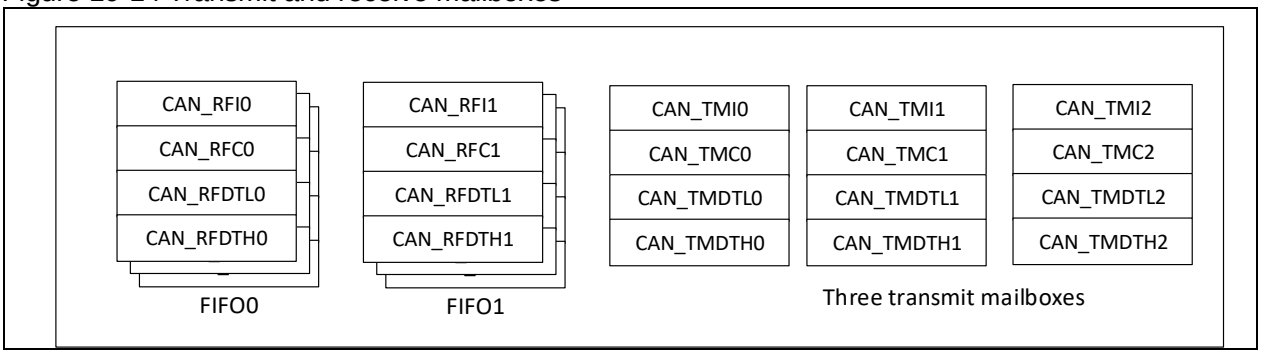
Transmit and receive mailboxes are the same except:

- RFFMN field in the CAN_RFCx register
- A receive mailbox is read only
- A transmit mailbox can be written only when empty. TMxEF=1 in the CAN_TSTS register indicates that the mailbox is empty.

There are three transmit mailboxes and two receive mailboxes. Each receive mailbox has 3-level depth of FIFO, and can only access to the first received message in the FIFO.

Each mailbox contains four registers.

Figure 19-14 Transmit and receive mailboxes



19.7.2.1 Transmit mailbox identifier register (CAN_TMIx) (x=0..2)

Note: 1. This register is write protected when its mailboxes are pending for transmission.

2.This register implements the Transmit Request control (bit 0) — reset value 0.

Bit	Name	Reset value	Type	Description
Bit 31: 21	TMSID/ TMEID	0xXXX	rw	Transmit mailbox standard identifier or extended identifier high bytes Note: This field defines the 11-bit high bytes of the standard identifier or extended identifier.
Bit 20: 3	TMEID	0xXXXXX	rw	Transmit mailbox extended identifier Note: This field defines the low 18 bits of the extended identifier.
Bit 2	TMIDSEL	0xX	rw	Transmit mailbox identifier type select 0: Standard identifier 1: Extended identifier
Bit 1	TMFRSEL	0xX	rw	Transmit mailbox frame type select 0: Data frame 1: Remote frame
Bit 0	TMSR	0x0	rw	Transmit mailbox send request 0: No effect 1: Transmit request Note: This bit is cleared by hardware when the transmission has been completed (The mailbox becomes empty)

19.7.2.2 Transmit mailbox data length and time stamp register (CAN_TMCx) (x=0..2)

All the bits in the register are write protected when the mailbox is not in empty state.

Bit	Name	Reset value	Type	Description
Bit 31: 16	TMTS	0xXXXX	rw	Transmit mailbox time stamp Note: This field contains the value of the CAN timer sampled at the SOF transmission.
Bit 15: 9	Reserved	0xXX	resd	Kept at default value
Bit 8	TMTSTEN	0xX	rw	Transmit mailbox time stamp transmit enable 0: Time stamp is not sent 1: Time stamp is sent Note: This bit is valid only when the time-triggered communication mode is enabled. In the time stamp MTS[15: 0], the MTS[7: 0] is stored in the TMDT7, and MTS[15: 8] in the TMDT6. The data length must be programmed as 8 to send time stamp.
Bit 7: 4	Reserved	0xX	resd	Kept at default value
Bit 3: 0	TMDTBL	0xX	rw	Transmit mailbox data byte length Note: This field defines the data length of a transmit message. A transmit message can contain from 0 to 8 data bytes.

19.7.2.3 Transmit mailbox data low register (CAN_TMDTLx) (x=0..2)

All the bits in the register are write protected when the mailbox is not in empty state.

Bit	Name	Reset value	Type	Description
Bit 31: 24	TMDT3	0xXX	rw	Transmit mailbox data byte 3
Bit 23: 16	TMDT2	0xXX	rw	Transmit mailbox data byte 2
Bit 15: 8	TMDT1	0xXX	rw	Transmit mailbox data byte 1
Bit 7: 0	TMDT0	0xXX	rw	Transmit mailbox data byte 0

19.7.2.4 Transmit mailbox data high register (CAN_TMDTHx) (x=0..2)

All the bits in the register are write protected when the mailbox is not in empty state.

Bit	Name	Reset value	Type	Description
Bit 31: 24	TMDT7	0xXX	rw	Transmit mailbox data byte 7
Bit 23: 16	TMDT6	0xXX	rw	Transmit mailbox data byte 6 Note: This field will be replaced with MTS[15: 8] when the time-triggered communication mode is enabled and the corresponding time stamp transmit is enabled.
Bit 15: 8	TMDT5	0xXX	rw	Transmit mailbox data byte 5
Bit 7: 0	TMDT4	0xXX	rw	Transmit mailbox data byte 4

19.7.2.5 Receive FIFO mailbox identifier register (CAN_RFIx) (x=0..1)

Note: All the receive mailbox registers are read only.

Bit	Name	Reset value	Type	Description
Bit 31: 21	RFSID/RFEID	0xXXX	ro	Receive FIFO standard identifier or receive FIFO extended identifier Note: This field defines the high 11 bits of the standard identifier or extended identifier.
Bit 20: 3	RFEID	0xXXXXX	ro	Receive FIFO extended identifier Note: This field defines the low 18 bits of the extended identifier.
Bit 2	RFIDI	0xX	ro	Receive FIFO identifier type indication 0: Standard identifier 1: Extended identifier
Bit 1	RFFRI	0xX	Ro	Receive FIFO frame type indication 0: Data frame 1: Remote frame
Bit 0	Reserved	0x0	resd	Kept at default value

19.7.2.6 Receive FIFO mailbox data length and time stamp register (CAN_RFCx) (x=0..1)

Note: All the receive mailbox registers are read only.

Bit	Name	Reset value	Type	Description
Bit 31: 16	RFTS	0xXXXX	ro	Receive FIFO time stamp Note: This field contains the value of the CAN timer sampled at the start of a receive frame.
Bit 15: 8	RFFMN	0xXX	ro	Receive FIFO filter match number Note: This field contains the filter number that a message has passed through.
Bit 7: 4	Reserved	0xX	resd	Kept at default value
Bit 3: 0	RFDTL	0xX	ro	Receive FIFO data length Note: This field defines the data length of a receive message. A transmit message can contain from 0 to 8 data bytes. For a remote frame, its data length RFDTl is fixed 0.

19.7.2.7 Receive FIFO mailbox data low register (CAN_RFDTLx) (x=0..1)

Note: All the receive mailbox registers are read only.

Bit	Name	Reset value	Type	Description
Bit 31: 24	RFDT3	0xXX	ro	Receive FIFO data byte 3
Bit 23: 16	RFDT2	0xXX	ro	Receive FIFO data byte 2
Bit 15: 8	RFDT1	0xXX	ro	Receive FIFO data byte 1
Bit 7: 0	RFDT0	0xXX	ro	Receive FIFO data byte 0

19.7.2.8 Receive FIFO mailbox data high register (CAN_RFDTHx) (x=0..1)

Note: All the receive mailbox registers are read only.

Bit	Name	Reset value	Type	Description
Bit 31: 24	RFDT7	0xXX	ro	Receive FIFO data byte 7
Bit 23: 16	RFDT6	0xXX	ro	Receive FIFO data byte 6
Bit 15: 8	RFDT5	0xXX	ro	Receive FIFO data byte 5
Bit 7: 0	RFDT4	0xXX	ro	Receive FIFO data byte 4

19.7.3 CAN filter registers

19.7.3.1 CAN filter control register (CAN_FCTRL)

Note: All the non-reserved bits of this register are controlled by software completely.

Bit	Name	Reset value	Type	Description
Bit 31: 1	Reserved	0x150E0700	resd	Kept at its default value
Bit 0	FCS	0x1	rw	Filter configuration switch 0: Disabled (Filter bank is active) 1: Enabled (Filter bank is in configuration mode) Note: The initialization of the filter bank can be configured only when it is in configuration mode.

19.7.3.2 CAN filter mode configuration register (CAN_FMCFG)

Note: This register can be written only when FCS=1 in the CAN_FCTRL register (The filter is in configuration mode)

Bit	Name	Reset value	Type	Description
Bit 31: 14	Reserved	0x00000	resd	Kept at default value
Bit 13: 0	FMSELx	0x0000	rw	Filter mode select Each bit corresponds to a filter bank. 0: Identifier mask mode 1: Identifier list mode

19.7.3.3 CAN filter bit width configuration register (CAN_ FBWCFG)

Note: This register can be written only when FCS=1 in the CAN_FCTRL register (The filter is in configuration mode)

Bit	Name	Reset value	Type	Description
Bit 31: 14	Reserved	0x00000	resd	Kept at default value
Bit 13: 0	FBWSELx	0x0000	rw	Filter bit width select Each bit corresponds to a filter bank. 0: Dual 16-bit 1: Single 32-bit

19.7.3.4 CAN filter FIFO association register (CAN_ FRF)

Note: This register can be written only when FCS=1 in the CAN_FCTRL register (The filter is in configuration mode)

Bit	Name	Reset value	Type	Description
Bit 31: 14	Reserved	0x00000	resd	Kept at default value
Bit 13: 0	FRFSELx	0x0000	rw	Filter relation FIFO select Each bit corresponds to a filter bank. 0: Associated with FIFO0 1: Associated with FIFO1

19.7.3.5 CAN filter activation control register (CAN_ FACFG)

Bit	Name	Reset value	Type	Description
Bit 31: 14	Reserved	0x00000	resd	Kept at default value
Bit 13: 0	FAENx	0x0000	rw	Filter active enable Each bit corresponds to a filter bank. 0: Disabled 1: Enabled

19.7.3.6 CAN filter bank i filter bit register (CAN_ FiFBx) (i=0..13; x=1..2)

Note: There are 14 filter banks (i=0..13). Each filter bank consists of two 32-bit registers, CAN_FiFB[2: 1]. This register can be modified only when the FAENx bit of the CAN_FACFG register is cleared or the FCS bit of the CAN_FCTRL register is set.

Bit	Name	Reset value	Type	Description
Bit 31: 0	FFDB	0x0000 0000	rw	Filters filter data bit Identifier list mode: The configuration value of the register matches with the level of the corresponding bit of the data received on the bus (If it is a standard frame, the value of the corresponding bit of the extended frame is neglected.) Identifier mark mode: Only the bit with its register configuration value being 1 can match with the level of the corresponding bit of the data received on the bus. It don't care when the register value is 0.

20 Hardware integer divider (HWDIV)

20.1 Introduction

Hardware-enabled 32-bit integer divider can be used for signed or unsigned division operations. Division operation begins with setting dividend and a divisor registers, without any other configurations. After configuration, quotient or remainder can be read. The calculations are returned by hardware after the completion of the division operation. There is no need for extra status check.

20.2 Main features

- Support signed and unsigned integer division operation
- Input 32-bit dividend and divisor, and output 32-bit quotient and remainder
- Support divided-by-0 error interrupt
- Division operations can be automatically performed after configuring dividend and divisor registers
- Reading quotient or remainder register returns the calculated result
- 9 system clocks for a division operation

20.3 Interrupts and interrupt control

The hardware integer divider (HWDIV), which is connected to CPU's NVIC, supports divided-by-zero error interrupts. When a divisor register is configured with 0 by software, the DIV0F bit is set in the HWDIV_STS register, and an interrupt is sent to CPU. In this case, in order to start a new division operation, it is necessary to clear the DIV0F bit by writing 1. If there is no need for CPU to receive such error interrupt, it is necessary to disable INTDIS bit in the HWDIV_CTRL register. However, error status can still be indicated by the DIV0F bit.

20.4 Configuration

- Set SIGN=0 (unsigned) or SIGN=1 (signed) in the HWDIV_CTRL register according to actual needs
- Set HWDIV_DVDD register (dividend)
- Set HWDIV_DVSR register (divisor). After setting HWDIV_DVDD and HWDIV_DVSR, division operation is then started.
- Read HWDIV_QUOT or HWDIV_REMD to get calculated results (no need extra wait time or status check)

20.5 Register descriptions

These peripheral registers must be accessed by words (32 bits).

Table 20-1 CAN register map and reset values

Register name	Offset	Reset value
HWDIV_DVDD	0x00	0x0000 0000
HWDIV_DVSR	0x04	0x0000 0001
HWDIV_QUOT	0x08	0x0000 0000
HWDIV_REMD	0x0C	0x0000 0000
HWDIV_CTRL	0x10	0x0000 0000
HWDIV_STS	0x14	0x0000 0000

20.5.1 Dividend register (HWDIV_DVDD)

Bit	Name	Reset value	Type	Description
Bit 31:0	DVDD	0x0000_0000	rw	Dividend

20.5.2 Divisor register (HWDIV_DVSR)

Bit	Name	Reset value	Type	Description
Bit 31:0	DVSR	0x0000_0001	rw	Divisor After configuring this register, a division operation is started automatically.

20.5.3 Quotient register (HWDIV_QUOT)

Bit	Name	Reset value	Type	Description
Bit 31:0	QUOT	0x0000_0000	ro	Quotient Reading this register will return a calculated result after the completion of a division operation.

20.5.4 Remainder register (HWDIV_REMD)

Bit	Name	Reset value	Type	Description
Bit 31:0	REMD	0x0000_0000	ro	Remainder Reading this register will return a calculated result after the completion of a division operation.

20.5.5 Hardware integer divider control register (HWDIV_CTRL)

Bit	Name	Reset value	Type	Description
Bit 31:3	Reserved	0x0000	resd	Kept at its default value.
Bit 2	INTDIS	0	rw	Error interrupt disable 0: Error interrupt enabled 1: Error interrupt disabled
Bit 1	SIGN	0	rw	Division signed operation 0: Non-signed division operation 1: Signed division operation
Bit 0	Reserved	0x0000	resd	Kept at its default value.

20.5.6 Divider status register (HWDIV_STS)

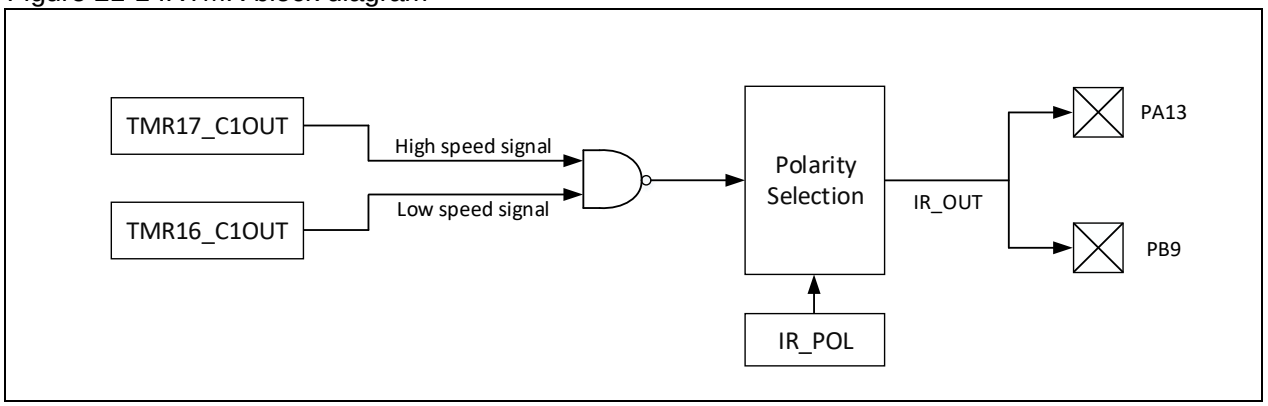
Bit	Name	Reset value	Type	Description
Bit 31:1	Reserved	0x00	resd	Kept at its default value.
Bit 0	DIV0F	0	rw1c	Division by zero fault status When this bit is set, it indicates that this is a divided-by-0 operation. It is cleared by writing 1.

21 Infrared timer (IRTMR)

The IRTMR (Infrared Timer) is used to generate IR_OUT signals for driving infrared LED to achieve infrared control.

The IR_OUT signals consists of a low-frequency modulation envelope and high-frequency carrier signals. The low-frequency modulation envelope signal comes from TMR16_C1OUT, while the high-frequency carrier signal is provided by the TMR17_C1OUT. The IR_POL bit in the SCFG_CFG1 register controls whether the IR_OUT output is reversed. The IR_OUT signal is output through multiplexed function PB9 or PA13 (multiplexed mode needs to be configured in advance).

Figure 21-1 IRTMR block diagram



22 Debug (DEBUG)

22.1 Debug introduction

22.2 Debug and Trace

It aims to support debugging various peripherals and configuring the status of peripherals in debug mode. For timers and watchdogs, the user can choose whether to stop or continue counting during debugging; For CAN, the user can define whether to stop or continue updating receive registers during debugging; For I2C, the user can choose whether to stop or continue SMBUS timeout counting.

In addition, code debugging is supported in Low-power mode. In Sleep mode, HICK and FCLK remain in run state. In Deepsleep mode, HICK oscillator is enabled to feed FCLK and HCLK.

Debugger is accessible through the address 0x40015800.

22.3 I/O pin control

The AT32L021 uses its two general-purpose I/O ports for SW-DP debugging. After reset, the SW-DP can be immediately used by debugger by default.

When uses need to switch between debug ports or disable debugging, these dedicated I/O ports can be released through SCFG registers for GPIO register to control.

22.4 DEGUB registers

Table 22-1 shows DEBUG register map and reset values.

These peripheral registers must be accessed by words (32 bits)

Table 22-1 DEBUG register address and reset value

Register name	Offset	Reset value
DEBUG_IDCODE	0x4001 5800	0xXXXX XXXX
DEBUG_CTRL	0x4001 5804	0x0000 0000
DEBUG_SER_ID	0x4001 5820	0x0000 XX0X

22.4.1 DEBUG device ID (DEBUG_IDCODE)

MCU integrates an ID code that is used to identify MCU's revision code. The DEBUG_IDCODE register is mapped on the external PPB bus at address 0x40015800. This code is accessible via SW port or by user code.

Bit	Name	Reset value	Type	Description
Bit 31: 0	PID	0xXXXX XXXX	ro	PID information

PID [31: 0]	AT32 part number	FLASH size	Packages
0x1001_2000	AT32L021G4U7	16KB	QFN28
0x1001_2001	AT32L021F4P7	16KB	TSSOP20
0x1001_2002	AT32L021F4U7	16KB	QFN20
0x1001_2003	AT32L021K4U7-4	16KB	QFN32
0x1001_2004	AT32L021K4U7	16KB	QFN32
0x1001_2005	AT32L021K4T7	16KB	LQFP32
0x1001_2006	AT32L021C4T7	16KB	LQFP48
0x1001_2087	AT32L021G6U7	32KB	QFN28
0x1001_2088	AT32L021F6P7	32KB	TSSOP20
0x1001_2089	AT32L021F6U7	32KB	QFN20

PID [31: 0]	AT32 part number	FLASH size	Packages
0x1001_208A	AT32L021K6U7-4	32KB	QFN32
0x1001_208B	AT32L021K6U7	32KB	QFN32
0x1001_208C	AT32L021K6T7	32KB	LQFP32
0x1001_208D	AT32L021C6T7	32KB	LQFP48
0x1001_210E	AT32L021G8U7	64KB	QFN28
0x1001_210F	AT32L021F8P7	64KB	TSSOP20
0x1001_2110	AT32L021F8U7	64KB	QFN20
0x1001_2111	AT32L021K8U7-4	64KB	QFN32
0x1001_2112	AT32L021K8U7	64KB	QFN32
0x1001_2113	AT32L021K8T7	64KB	LQFP32
0x1001_2114	AT32L021C8T7	64KB	LQFP48

22.4.2 DEBUG control register (DEBUG_CTRL)

This register is asynchronously reset by PORESET Reset (not reset by system reset). It can be written by debugger under reset.

Bit	Name	Reset value	Type	Description
Bit 31:28	Reserved	0x0	resd	Always 0.
Bit 27	TMR14_PAUSE	0	rw	TMR14 pause control bit 0: TMR14 works normally 1: TMR14 stops working
Bit 26: 25	Reserved	0	resd	Always 0.
Bit 24	TMR17_PAUSE	0	rw	TMR17 pause control bit 0: TMR17 works normally 1: TMR17 stops working
Bit 23	TMR16_PAUSE	0	rw	TMR16 pause control bit 0: TMR16 works normally 1: TMR16 stops working
Bit 22	TMR15_PAUSE	0	rw	TMR15 pause control bit 0: TMR15 works normally 1: TMR15 stops working
Bit 21	Reserved	0	resd	Always 0.
Bit 20	Reserved	0	resd	Always 0.
Bit 19	TMR6_PAUSE	0	rw	TMR16 pause control bit 0: TMR16 works normally 1: TMR16 stops working
Bit 18: 17	Reserved	0	resd	Always 0.
Bit 16	I2C2_SMBUS_TIMEOUT	0	rw	I2C2 pause control bit 0: I2C2 works normally 1: I2C2 SMBUS timeout control stops working
Bit 15	I2C1_SMBUS_TIMEOUT	0	rw	I2C1 pause control bit 0: I2C1 works normally 1: I2C1 SMBUS timeout control stops working
Bit 14	ERTC_PAUSE	0	rw	ERTC pause control bit 0: ERTC works normally 1: ERTC stops working
Bit 13	Reserved	0x0	resd	Always 0.
Bit 12	TMR3_PAUSE	0	rw	TMR3 pause control bit 0: TMR3 works normally 1: TMR3 stops working
Bit 11	Reserved	0	resd	Always 0.
Bit 10	TMR1_PAUSE	0	rw	TMR1 pause control bit 0: TMR1 works normally 1: TMR1 stops working
Bit 9	WWDT_PAUSE	0	rw	WWDT pause control bit 0: WWDT works normally 1: WWDT stops working
Bit 8	WDT_PAUSE	0	rw	WDT pause control bit 0: WDT works normally 1: WDT stops working
Bit 7:4	Reserved	0x0	resd	Always 0.

Bit 3	CAN_PAUSE	0	rw	CAN pause control bit 0: CAN works normally 1: CAN receive register stops receiving new data
Bit 2	STANDBY_DEBUG	0	rw	Standby mode pause control bit 0: The entire 1.2V digital circuit is powered off after Standby mode entry 1: The entire 1.2V digital circuit is still powered on after Standby mode entry. The system clock is provided by HICK.
Bit 1	DEEPSLEEP_DEBUG	0	rw	Deepsleep mode pause control bit 0: Entering Deepsleep mode will disable 1.2V domain clock. After leaving Deepsleep mode, HICK is activated and used as a clock source of system clock. Software should configure new system clock according to application needs. 1: After entering Deepsleep mode, system clock is provided by HICK. After leaving Deepsleep mode, HICK is used as system clock. Software should configure new system clock according to application needs.
Bit 0	SLEEP_DEBUG	0	rw	Sleep mode pause control bit 0: After entering Sleep mode, CPU HCLK is disabled, but other clocks still work. After leaving Sleep mode, there is no need to configure new clock system. 1: All clocks continue to run in Sleep mode.

22.4.3 DEBUG SERIES ID register (DEBUG_SER_ID)

DEBUG_SER_ID register is used to identify MCU series ID and revision ID. It is mapped to external PPB bus, and the base address is 0x40015820. This register is asynchronously reset by PORESET Reset (not reset by system reset). The MCU series ID is accessible through SW debug port or user code.

Bit	Name	Reset value	Type	Description
Bit 31:16	Reserved	0x0000	resd	Kept at its default value.
Bit 15:8	SER_ID	0xXX	ro	MCU series ID AT32L021: 0x10
Bit 7:3	Reserved	0x00	resd	Kept at its default value.
Bit 2:0	REV_ID	0xX	ro	Revision ID 0x0: Revision A

23 Revision history

Document Revision History

Date	Version	Revision Note
2024.02.23	2.00	Initial release.

IMPORTANT NOTICE – PLEASE READ CAREFULLY

Purchasers are solely responsible for the selection and use of ARTERY's products and services, and ARTERY assumes no liability whatsoever relating to the choice, selection or use of the ARTERY products and services described herein

No license, express or implied, to any intellectual property rights is granted under this document. If any part of this document deals with any third party products or services, it shall not be deemed a license granted by ARTERY for the use of such third party products or services, or any intellectual property contained therein, or considered as a warranty regarding the use in any manner of such third party products or services or any intellectual property contained therein.

Unless otherwise specified in ARTERY's terms and conditions of sale, ARTERY provides no warranties, express or implied, regarding the use and/or sale of ARTERY products, including but not limited to any implied warranties of merchantability, fitness for a particular purpose (and their equivalents under the laws of any jurisdiction), or infringement on any patent, copyright or other intellectual property right.

Purchasers hereby agree that ARTERY's products are not designed or authorized for use in: (A) any application with special requirements of safety such as life support and active implantable device, or system with functional safety requirements; (B) any aircraft application; (C) any aerospace application or environment; (D) any weapon application, and/or (E) or other uses where the failure of the device or product could result in personal injury, death, property damage. Purchasers' unauthorized use of them in the aforementioned applications, even if with a written notice, is solely at purchasers' risk, and Purchasers are solely responsible for meeting all legal and regulatory requirements in such use.

Resale of ARTERY products with provisions different from the statements and/or technical characteristics stated in this document shall immediately void any warranty grant by ARTERY for ARTERY's products or services described herein and shall not create or expand any liability of ARTERY in any manner whatsoever.