N32H473/474/475/480/481/482/487/488 Series

32-bit ARM Cortex^{TM}-M4F Microcontroller

User Manual V1.2.0

Contents

List of tables

List of figures

1 Abbreviations

1.1 Describes The List Of Abbreviations Used In The Register Table

The following abbreviations are used in the description of registers:

read/write(rw)	Software can read and write this bit.
read-only(r)	Software can only read this bit.
write-only(w)	Software can only write this bit, and reading this bit will return the reset value.
read/clear(rc_w1)	Software can read this bit or clear it by writing' 1', and writing' 0' has no effect on this bit.
read/clear(rc_w0)	Software can read this bit or clear it by writing' 0', and writing' 1' has no effect on this bit.
read/clear by read(rc_r)	Software can read this bit. Reading this bit will automatically clear it to' 0'. Writing' 0' has no effect on this bit.
read/set(rs)	Software can read or set this bit. Writing' 0' has no effect on this bit.
read-only write trigger(rt_w)	Software can read this bit and write' 0' or' 1' to trigger an event, but it has no effect on this bit value.
toggle(t)	Software can only flip this bit by writing' 1', and writing' 0' has no effect on this bit.
Reserved(Res.)	Reserved bits, the default value must be kept unchanged.

1.2 Available Peripherals

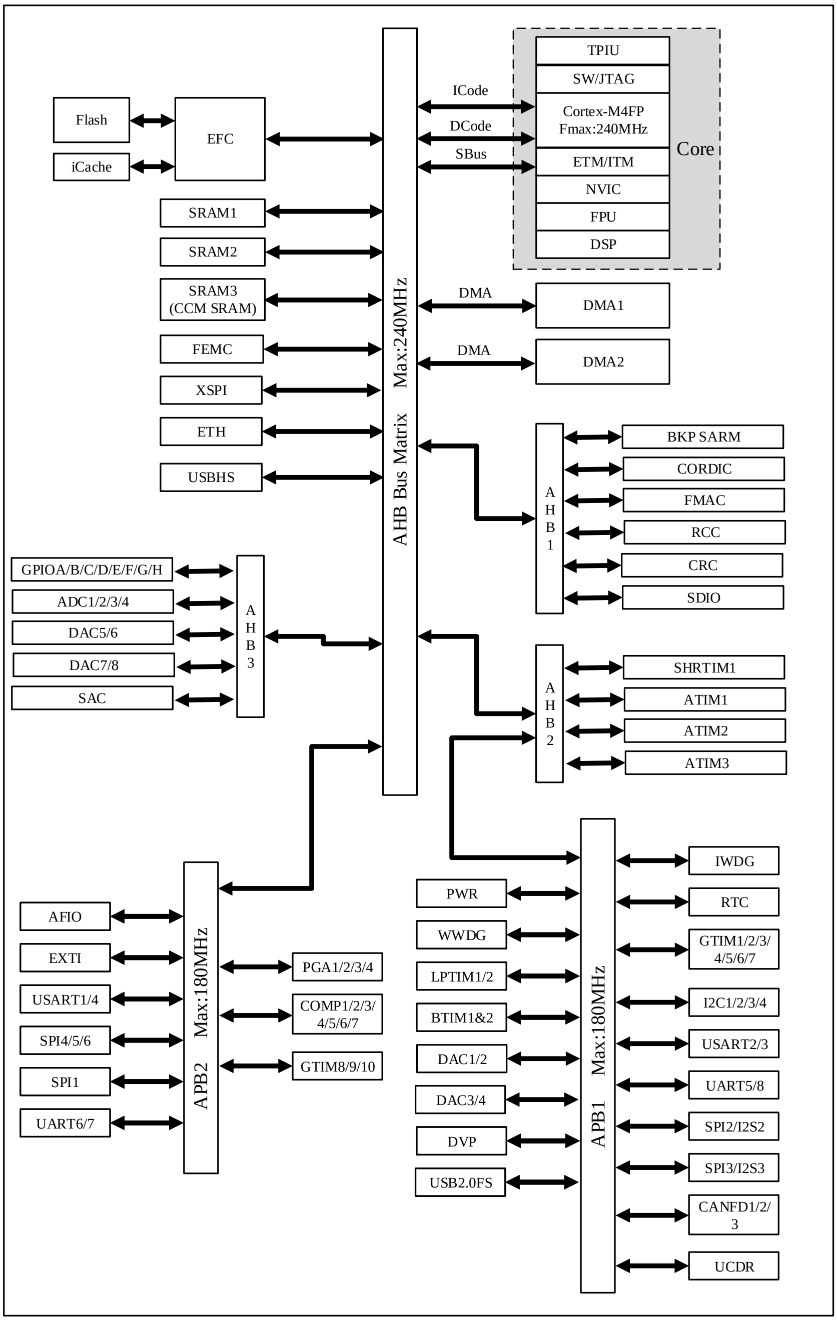
For all models of N32H47x, N32H48x microcontroller series, the existence and number of a peripheral, please refer to the data sheet of the corresponding model.

2 Memory And Bus Architecture

2.1 System Architecture

2.1.1 Bus Architecture

Figure 2-1 Bus Architecture



Note: The highest frequency of N32H480 series is 180MHz, while the highest frequency of other series is 240MHz. Different series and models support different peripherals and quantities. Please refer to the respective datasheets for *specific information.*

- ICode bus: This bus connects the ICode bus of Cortex^{TM}-M4FP core with the Flash instruction interface. Instruction prefetching is completed on this bus.
- DCode bus: This bus connects the DCode bus of Cortex^{TM}-M4FP core with the data interface of Flash memory (constant loading and debugging access).
- SBus: This bus connects the SBus bus (peripheral bus) of Cortex^{TM}-M4FP core to the bus matrix, which coordinates the access between the core and DMA.

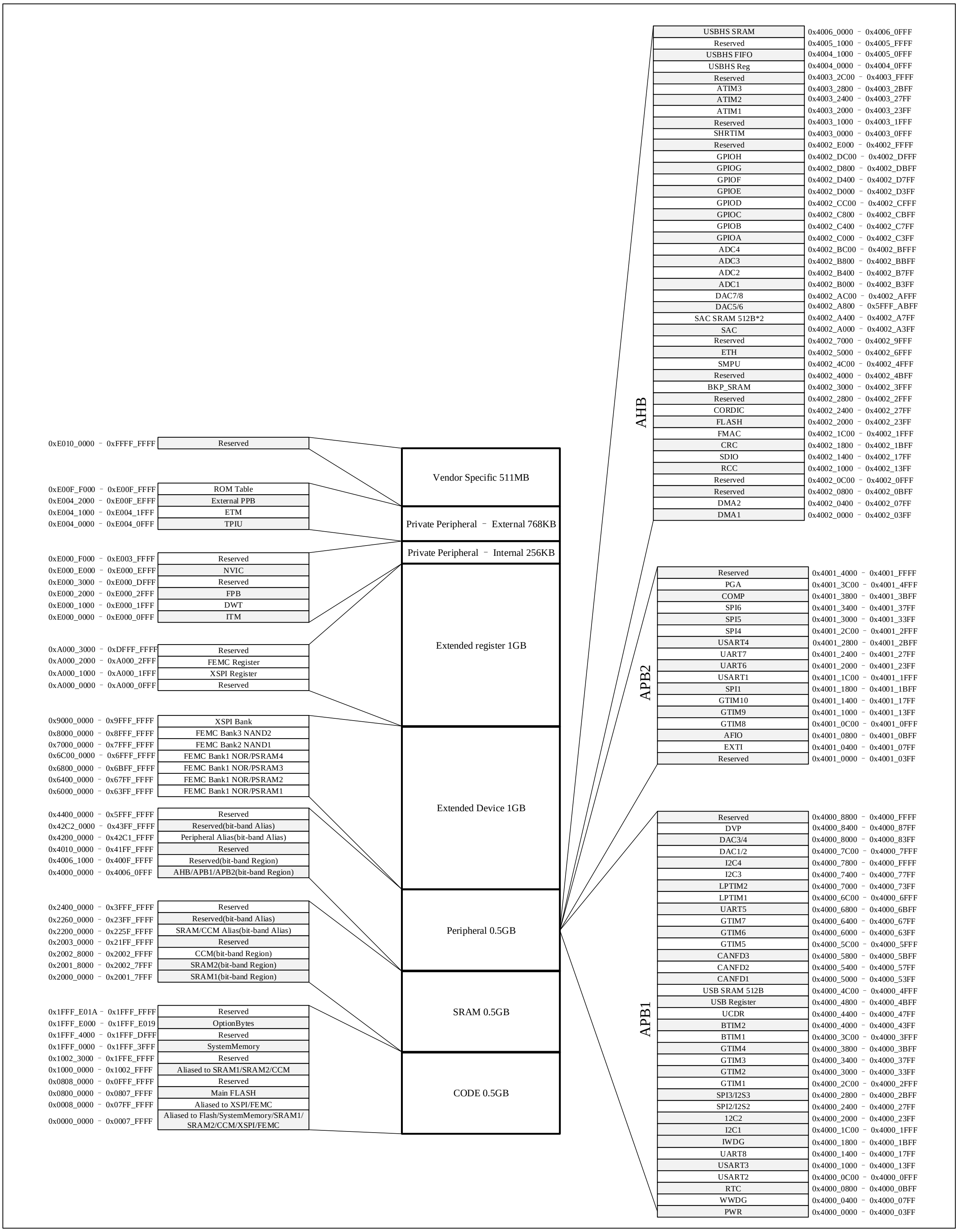
The bus system adopts an architecture interconnected with BusMatrix and AHB/APB buses. The bus matrix uses a round-robin scheduling mechanism to coordinate the core's ICode/DCode/SBus and DMA1/DMA2/ETH/USB HS/CANFD/DVP access to FLASH/SRAM/peripherals. When accessing FLASH, Dcode has higher priority than ICode.

- SAC/CRC has designed matrix interconnection, which supports DMA transmission by software triggering.
- The bus matrix includes 8 AHB masters: ICode/DCode/SBus/DMA1/DMA2/ETH/USB HS/CANFD; and 8 AHB slaves: SRAM1/2, CCM, FEMC, XSPI, AHB1/2/3; with a maximum speed of 240MHz for HCLK.
- The system consists of two AHB2APB Bridges, namely1 AHB2APB1 and AHB2APB2. APB1 contains 33 low-speed APB peripherals, the maximum speed of PCLK1 is 180 MHz; APB2 contains 24 high-speed APB peripherals, the maximum speed of PCLK2 is 180 MHz.

2.1.2 Bus Address Mapping

The address mapping includes all AHB and APB peripherals: AHB peripherals, APB1 peripherals, APB2 peripherals, Flash, SRAM, System Memory, etc. And the address space of SRAM/CCM is located in the bit-band region of SRAM, and atomic accesses can be made through the bit-band Alias to perform read-modify-write operations on the target bits of the bit-band region. The address spaces of all APB and AHB peripherals are located in the bit-band region of the peripherals. Atomic accesses can be made through the bit-band Alias to perform read-modify-write operations on the target bits of the bit-band region. The specific mapping is as follows:

Figure 2-2 Bus Address Map



0xE010_0000 – 0xFFFF_FFFF	Reserved

0xE00F_F000 – 0xE00F_FFFF	ROM Table
0xE004_2000 – 0xE00F_EFFF	External PPB
0xE004_1000 – 0xE004_1FFF	ETM
0xE004_0000 – 0xE004_0FFF	TPIU

0xE000_F000 – 0xE003_FFFF	Reserved
0xE000_E000 – 0xE000_EFFF	NVIC
0xE000_3000 – 0xE000_DFFF	Reserved
0xE000_2000 – 0xE000_2FFF	FPB
0xE000_1000 – 0xE000_1FFF	DWT
0xE000_0000 – 0xE000_0FFF	ITM

0xA000_3000 – 0xDFFF_FFFF	Reserved
0xA000_2000 – 0xA000_2FFF	FEMC Register
0xA000_1000 – 0xA000_1FFF	XSPI Register
0xA000_0000 – 0xA000_0FFF	Reserved

0x9000_0000 – 0x9FFF_FFFF	XSPI Bank
0x8000_0000 – 0x8FFF_FFFF	FEMC Bank3 NAND2
0x7000_0000 – 0x7FFF_FFFF	FEMC Bank2 NAND1
0x6C00_0000 – 0x6FFF_FFFF	FEMC Bank1 NOR/PSRAM4
0x6800_0000 – 0x6BFF_FFFF	FEMC Bank1 NOR/PSRAM3
0x6400_0000 – 0x67FF_FFFF	FEMC Bank1 NOR/PSRAM2
0x6000_0000 – 0x63FF_FFFF	FEMC Bank1 NOR/PSRAM1

0x4400_0000 – 0x5FFF_FFFF	Reserved
0x42C2_0000 – 0x43FF_FFFF	Reserved(bit-band Alias)
0x4200_0000 – 0x42C1_FFFF	Peripheral Alias(bit-band Alias)
0x4010_0000 – 0x41FF_FFFF	Reserved
0x4006_1000 – 0x400F_FFFF	Reserved(bit-band Region)
0x4000_0000 – 0x4006_0FFF	AHB/APB1/APB2(bit-band Region)

0x2400_0000 – 0x3FFF_FFFF	Reserved
0x2260_0000 – 0x23FF_FFFF	Reserved(bit-band Alias)
0x2200_0000 – 0x225F_FFFF	SRAM/CCM Alias(bit-band Alias)
0x2003_0000 – 0x21FF_FFFF	Reserved
0x2002_8000 – 0x2002_FFFF	CCM(bit-band Region)
0x2001_8000 – 0x2002_7FFF	SRAM2(bit-band Region)
0x2000_0000 – 0x2001_7FFF	SRAM1(bit-band Region)

0x1FFF_E01A – 0x1FFF_FFFF	Reserved
0x1FFF_E000 – 0x1FFF_E019	OptionBytes
0x1FFF_4000 – 0x1FFF_DFFF	Reserved
0x1FFF_0000 – 0x1FFF_3FFF	SystemMemory
0x1002_3000 – 0x1FFE_FFFF	Reserved
0x1000_0000 – 0x1002_FFFF	Aliased to SRAM1/SRAM2/CCM
0x0808_0000 – 0x0FFF_FFFF	Reserved
0x0800_0000 – 0x0807_FFFF	Main FLASH
0x0008_0000 – 0x07FF_FFFF	Aliased to XSPI/FEMC
0x0000_0000 – 0x0007_FFFF	Aliased to Flash/SystemMemory/SRAM1/ SRAM2/CCM/XSPI/FEMC

AHB

USBHS SRAM	0x4006_0000 – 0x4006_0FFF
Reserved	0x4005_1000 – 0x4005_FFFF
USBHS FIFO	0x4004_1000 – 0x4005_0FFF
USBHS Reg	0x4004_0000 – 0x4004_0FFF
Reserved	0x4003_2C00 – 0x4003_FFFF
ATIM3	0x4003_2800 – 0x4003_2BFF
ATIM2	0x4003_2400 – 0x4003_27FF
ATIM1	0x4003_2000 – 0x4003_23FF
Reserved	0x4003_1000 – 0x4003_1FFF
SHRTIM	0x4003_0000 – 0x4003_0FFF
Reserved	0x4002_E000 – 0x4002_FFFF
GPIOH	0x4002_DC00 – 0x4002_DFFF
GPIOG	0x4002_D800 – 0x4002_DBFF
GPIOF	0x4002_D400 – 0x4002_D7FF
GPIOE	0x4002_D000 – 0x4002_D3FF
GPIOD	0x4002_CC00 – 0x4002_CFFF
GPIOC	0x4002_C800 – 0x4002_CBFF
GPIOB	0x4002_C400 – 0x4002_C7FF
GPIOA	0x4002_C000 – 0x4002_C3FF
ADC4	0x4002_BC00 – 0x4002_BFFF
ADC3	0x4002_B800 – 0x4002_BBFF
ADC2	0x4002_B400 – 0x4002_B7FF
ADC1	0x4002_B000 – 0x4002_B3FF
DAC7/8	0x4002_AC00 – 0x4002_AFFF
DAC5/6	0x4002_A800 – 0x5FFF_ABFF
SAC SRAM 512B*2	0x4002_A400 – 0x4002_A7FF
SAC	0x4002_A000 – 0x4002_A3FF
Reserved	0x4002_7000 – 0x4002_9FFF
ETH	0x4002_5000 – 0x4002_6FFF
SMPU	0x4002_4C00 – 0x4002_4FFF
Reserved	0x4002_4000 – 0x4002_4BFF
BKP_SRAM	0x4002_3000 – 0x4002_3FFF
Reserved	0x4002_2800 – 0x4002_2FFF
CORDIC	0x4002_2400 – 0x4002_27FF
FLASH	0x4002_2000 – 0x4002_23FF
FMAC	0x4002_1C00 – 0x4002_1FFF
CRC	0x4002_1800 – 0x4002_1BFF
SDIO	0x4002_1400 – 0x4002_17FF
RCC	0x4002_1000 – 0x4002_13FF
Reserved	0x4002_0C00 – 0x4002_0FFF
Reserved	0x4002_0800 – 0x4002_0BFF
DMA2	0x4002_0400 – 0x4002_07FF
DMA1	0x4002_0000 – 0x4002_03FF

APB2

Reserved	0x4001_4000 – 0x4001_FFFF
PGA	0x4001_3C00 – 0x4001_4FFF
COMP	0x4001_3800 – 0x4001_3BFF
SPI6	0x4001_3400 – 0x4001_37FF
SPI5	0x4001_3000 – 0x4001_33FF
SPI4	0x4001_2C00 – 0x4001_2FFF
USART4	0x4001_2800 – 0x4001_2BFF
UART7	0x4001_2400 – 0x4001_27FF
UART6	0x4001_2000 – 0x4001_23FF
USART1	0x4001_1C00 – 0x4001_1FFF
SPI1	0x4001_1800 – 0x4001_1BFF
GTIM10	0x4001_1400 – 0x4001_17FF
GTIM9	0x4001_1000 – 0x4001_13FF
GTIM8	0x4001_0C00 – 0x4001_0FFF
AFIO	0x4001_0800 – 0x4001_0BFF
EXTI	0x4001_0400 – 0x4001_07FF
Reserved	0x4001_0000 – 0x4001_03FF

APB1

Reserved	0x4000_8800 – 0x4000_FFFF
DVP	0x4000_8400 – 0x4000_87FF
DAC3/4	0x4000_8000 – 0x4000_83FF
DAC1/2	0x4000_7C00 – 0x4000_7FFF
I2C4	0x4000_7800 – 0x4000_FFFF
I2C3	0x4000_7400 – 0x4000_77FF
LPTIM2	0x4000_7000 – 0x4000_73FF
LPTIM1	0x4000_6C00 – 0x4000_6FFF
UART5	0x4000_6800 – 0x4000_6BFF
GTIM7	0x4000_6400 – 0x4000_67FF
GTIM6	0x4000_6000 – 0x4000_63FF
GTIM5	0x4000_5C00 – 0x4000_5FFF
CANFD3	0x4000_5800 – 0x4000_5BFF
CANFD2	0x4000_5400 – 0x4000_57FF
CANFD1	0x4000_5000 – 0x4000_53FF
USB SRAM 512B	0x4000_4C00 – 0x4000_4FFF
USB Register	0x4000_4800 – 0x4000_4BFF
UCDR	0x4000_4400 – 0x4000_47FF
BTIM2	0x4000_4000 – 0x4000_43FF
BTIM1	0x4000_3C00 – 0x4000_3FFF
GTIM4	0x4000_3800 – 0x4000_3BFF
GTIM3	0x4000_3400 – 0x4000_37FF
GTIM2	0x4000_3000 – 0x4000_33FF
GTIM1	0x4000_2C00 – 0x4000_2FFF
SPI3/I2S3	0x4000_2800 – 0x4000_2BFF
SPI2/I2S2	0x4000_2400 – 0x4000_27FF
12C2	0x4000_2000 – 0x4000_23FF
I2C1	0x4000_1C00 – 0x4000_1FFF
IWDG	0x4000_1800 – 0x4000_1BFF
UART8	0x4000_1400 – 0x4000_17FF
USART3	0x4000_1000 – 0x4000_13FF
USART2	0x4000_0C00 – 0x4000_0FFF
RTC	0x4000_0800 – 0x4000_0BFF
WWDG	0x4000_0400 – 0x4000_07FF
PWR	0x4000_0000 – 0x4000_03FF

2.1.2.1 Bit banding

Cortex^{TM}-M4FP memory map includes two bit-band regions. These two bit-band regions map each word in the alias memory region to a bit in the bit-band memory region. When writing a word in the alias region, it is equivalent to performing a read-modify-write operation on the target bits of the bit-band region.

Both the peripheral registers and SRAM are mapped into a bit-band region, which allows a single bit-band region write and read operation to be performed.

The following mapping formula shows how each byte in the alias region corresponds to the corresponding bit in the bit band region:

bitband_byte_addr = bitband _base + (byte_offset×32) + (bit_number×4)

In which:

bitband_byte_addr is the address of the byte in the alias memory region, which is mapped to a certain target bit;

bitband_base is the starting address of the alias region;

byte_offset is the serial number of the byte containing the target bit in the bit-band region;

bit_number is the position of the target bit (0-7).

For example:

The following example shows how to map bit 4 of the byte located at SRAM address 0x20000400 to the alias region:

0x22008010 = 0x22000000 + (0x400×32) + (4×4).

Writing to address 0x22008010 has the same effect as read-modify-write operation on bit 4 of the byte at SRAM address 0x20000400.

Reading 0x22008010 address returns the value of bit 4 (0x01 or 0x00) of address 0x20000400 bytes in SRAM. Please refer to "Cortex^{TM}-M4F Technical Reference Manual" for more information about bit-banding.

2.1.3 Boot Management

2.1.3.1 Boot address

During system startup, you can select the BOOT mode after the reset through the BOOT0 pin and the user OptionByte BOOT configuration. After a system reset or exit from Standby mode, the value of the BOOT pin will be re-sampled and the OptionByte boot configuration will be re-read. After a startup delay has elapsed, the CPU fetches the top-of-stack value from address 0x0000_0000 and executes the code from the reset vector address indicated by address 0x0000_0004. Because of the Cortex^{TM}-M4F always gets the top-of-stack valueand reset vector from addresses 0x0000_0000 and 0x0000_0004 via ICode bus, so boot is only suitable for booting from the CODE area, and address remapping is designed for boot space. There are three boot modes to choose from:

- Boot from main Flash, including booting from the front bank (0x0800_0000) of the main Flash and the rear bank (0x0804_0000) of the main Flash.:
    - ◆ Main Flash memory is mapped to the boot space (0x0000_0000);
    - ◆ Main flash memory is accessible in two address areas, 0x0000_0000 or 0x0800_0000/0x0804_0000 (either one) (ICode/DCode/DMA1/DMA2);

- Boot from System Memory:
  - System Memory is mapped to boot space (0x0000_0000);
  - System Memory can be accessed in two address areas, 0x0000_0000 or 0x1FFF_0000 (ICode/DCode/DMA1/DMA2);
- Boot from the embedded SRAM:
  - The embedded SRAM is mapped to boot space (0x0000_0000);
  - The embedded SRAM is accessible in two address areas, 0x0000_0000 or 0x2000_0000 (ICode/DCode/DMA1/DMA2);

2.1.3.2 Boot from External Memory

In addition to the three self-boot methods mentioned above, physical address remapping of address 0 can also be achieved by configuring RCC_BOOTREMAP.REMAPSEL[2:0]. For specific details, refer to Section 4.3.29.

- Boot from external XSPI memory:
  - The external XSPI memory is mapped to boot space (0x0000_0000);
  - The external XSPI memory is accessible in two address areas, 0x0000_0000 or 0x9000_0000 (ICode/DCode/DMA1/DMA2);
- Boot from external FEMC memory:
  - The external FEMC memory is mapped to boot space (0x0000_0000);
  - The external XSPI memory is accessible in two address areas, 0x0000_0000 or 0x6000_0000 (ICode/DCode/DMA1/DMA2);

When code/data is stored in an external memory (XSPI/FEMC), it can be encrypted to prevent theft. Encrypted downloads can be achieved using the download tool provided by the Nsing Technologies.

When the code/data is encrypted and downloaded to the external memory, the MCU needs to decrypt it when using the external memory.

The decryption process for XSPI is as follows:

- Write the decryption start address of the XSPI external memory to XSPI_DSTRADD.ADD.
- Write the decryption end address of the XSPI external memory to XSPI_DENDADD.ADD.
- If the RTPD_KEY.KEY value does not match the KEY used for encryption, the RTPD_KEY.KEY value needs to be updated.
- Set XSPI_FEMC_DEN.EN to 1.

The decryption process for FEMC is as follows:

- Write the decryption start address of the FEMC external memory to FEMC_DSTRADD.ADD.
- Write the decryption end address of the FEMC external memory to FEMC_DENDADD.ADD.
- If the RTPD_KEY.KEY value does not match the KEY used for encryption, the RTPD_KEY.KEY value needs to be updated.

- Set XSPI_FEMC_DEN.EN to 1.

Note: 1) Setting XSPI_FEMC_DEN.EN to 1 should be configured after writing the address and KEY.

2) The key for software encryption and the key for hardware encryption are in reverse word order, e.g., the key for software encryption is 0x01234567, 0x89abcdef, 0xfedcba98, 0x76543210; then the key order for hardware configuration decryption is to be configured as:

FLASH->RTPD_KEY = 0x76543210;

FLASH->RTPD_KEY = 0xfedcba98;

FLASH->RTPD_KEY = 0x89abcdef;

FLASH->RTPD_KEY = 0x01234567;

3) If decryption is enabled, swd debug cannot be used to prevent reading user data in debug mode;

2.1.3.3 Boot configuration

In addition, SRAM can also be accessed through virtual address segment 0x1000_0000, which makes the CPU jump to SRAM to execute programs through ICode/DCode after booting from Main Flash or System Memory (note that this is not booting from SRAM, and it is not part of the boot mode). In addition to configuring boot program with the BOOT pin , there are two ways to run the program in SRAM:

- Jump directly to the physical address segment 0x2000_0000 of SRAM to run the program. At this time, the program will be run through SBus.
- Jump to the virtual address segment 0x1000_0000 of SRAM, and internally remap to the physical address segment 0x2000_0000 to run the program. At this time, the program will run efficiently through ICode/DCode.

Table 2-1 List Of Boot Mode

Option Byte				**Pin**	**Boot mode**	**The starting address of the memory space to be accessed in the corresponding startup mode**			
FLASH_BOOT	nBOOT1	nBOOT0	nSWBOOT0	BOOT0		Main Flash Ex-BANK	Main Flash Post-BANK	System Memory	SRAM
1	x	x	1	0	Main Flash Ex-bank boot	0x0000_0000 0x0800_0000	0x0804_0000	0x1FFF_0000	0x1000_0000 0x2000_0000
1	x	1	0	x					
0	x	x	1	0	Main Flash Post-bank boot	0x0800_0000	0x0000_0000 0x0804_0000	0x1FFF_0000	0x1000_0000 0x2000_0000
0	x	1	0	x					
x	1	x	1	1	SystemMemory boot	0x08000000	0x0804_0000	0x0000_0000 0x1FFF_0000	0x1000_0000 0x2000_0000
x	1	0	0	x					
x	0	x	1	1	SRAM boot	0x08000000	0x0804_0000	0x1FFF_0000	0x0000_0000 0x1000_0000 0x2000_0000
x	0	0	0	x					

2.1.3.4 Embedded bootloader

The embedded bootloader is stored in the System Memory, for further details please refer to the bootloader manual.

2.2 Memory System

The program memory, data memory, registers and I/O ports are organized in the same 4GB linear address space. Data bytes are stored in the memory in Little Endian format. The lowest numbered byte in a word is regarded as the least significant byte of the word, while the highest numbered byte is the most significant byte. The specifications of program memory and data memory are as follows.

2.2.1 FLASH Specification

The Flash consists of a main Flash memory block and an information block, which are described below: (The capacity values in the following description do not include ECC)

- The maximum main memory block is 512 KB, also known as main Flash memory, which contains 64 pages for storing and running user programs and storing data.
- The information block is 32 KB, including 4 pages, and consists of SystemMemory area (16 KB), system configuration area (8 KB) and OptionByte area (8 KB).
  - The system memory area is 16 KB, which contains 2 pages, also known as SystemMemory, and is used for storing and running the bootloader (BOOT).
  - The system configuration area is 8 KB, including 1 page.
  - The OptionByte area is 8 KB, containing 1 page, also known as OptionByte, with an effective space of 26B. Both the BOOT programs and user programs can read, write, and erase this area.

2.2.1.1 Flash memory organization

Both the main memory block and the information block are allocated to bus address space.

Table 2-2 Flash Bus Address List

Memory Area	Page Name	Address Range	Size
Main memory block	Page 0	0x0800_0000 – 0x0800_1FFF	8KB
	Page 1	0x0800_2000 – 0x0800_3FFF	8KB
	Page 2	0x0800_4000 – 0x0800_5FFF	8KB
	⋮	⋮	⋮
	Page 63	0x0807_E000 – 0x0807_FFFF	8KB
Information block	SystemMemory area	0x1FFF_0000 – 0x1FFF_3FFF	16KB
	System configuration area	0x1FFF_C000 – 0x1FFF_DFFF	8KB
	OptionByte area	0x1FFF_E000 – 0x1FFF_E019	26B
Flash memory interface registers	FLASH_AC	0x4002_2000 – 0x4002_2003	4B
	FLASH_CTRL	0x4002_2004 – 0x4002_2007	4B
	FLASH_STS	0x4002_2008 – 0x4002_200B	4B
	FLASH_ADD	0x4002_200C – 0x4002_200F	4B
	FLASH_KEY	0x4002_2010 – 0x4002_2013	4B
	FLASH_OPTKEY	0x4002_2014 – 0x4002_2017	4B

	FLASH_OB	0x4002_2018 – 0x4002_201B	4B
	FLASH_WRP	0x4002_201C – 0x4002_201F	4B
	FLASH_ECC	0x4002_2020 – 0x4002_2023	4B
	Reserved	0x4002_2024 – 0x4002_202B	4B
	FLASH_RDN	0x4002_202C – 0x4002_202F	8B
	FLASH_CAHR	0x4002_2030 – 0x4002_2033	4B
	FLASH_ROWP	0x4002_2034 – 0x4002_2037	4B
	CCM_WRP1	0x4002_2038 – 0x4002_203B	4B
	CCM_KEY	0x4002_203C – 0x4002_203F	4B
	CCM_ERASE	0x4002_2040 – 0x4002_2043	4B
	CCM_WRP2	0x4002_2044 – 0x4002_2047	4B
	CCM_MODE	0x4002_2048 – 0x4002_204B	4B
	XSPI_DSTRADD	0x4002_204C – 0x4002_204F	4B
	XSPI_DENDADD	0x4002_2050 – 0x4002_2053	4B
	FEMC_DSTRADD	0x4002_2054 – 0x4002_2057	4B
	FEMC_DENDADD	0x4002_2058 – 0x4002_205B	4B
	RTPD_KEY	0x4002_205C – 0x4002_205F	4B
	JTAG_SEAL	0x4002_2060 – 0x4002_2063	4B
	RTPD_KEY_WCNT	0x4002_2064 – 0x4002_2067	4B
	Reserved	0x4002_2068 – 0x4002_206C	4B
	XSPI_FEMC_DEN	0x4002_206C – 0x4002_206F	4B
	XUID	0x4002_2070 – 0x4002_2073	4B

The flash memory is organized as 128-bit wide memory units, which can store codes and data constants.

Information block is divided into three parts:

- The system memory area is used to store the bootloader in the system memory.
- System configuration area contains basic information about the chip.
- The OptionByte area, writing to main memory block and information block is managed by embedded flash programming/erasing controller.

There are two ways to protect Flash memory from illegal access (read, write and erase):

- Page write protection (WRP)
- Read protection (RDP)

When the Flash memory write operation is executed, any read operation to the Flash memory will stall the bus, and the read operation can only be performed correctly after the write operation is completed. This means that code or data fetches cannot be made while a program/erase operation is ongoing.

When performing Flash programming operations(write or erase), the internal RC oscillator (HSI) must be turned on

Note: In the low power consumption mode, all flash memory operations are suspended or halted.

2.2.1.2 Read and write operation

The Flash operation only supports 64-bit operation, and the Flash should be erased before the write operation, and the minimum block size for erasing is one page, which is 8 KB. The write operation consists of the erase and program phases.

When reading Flash, the number of waiting cycles for reading can be configured by the register. When using, it needs to be calculated in combination with the clock frequency of AHB interface. For example, when HCLK<=40 MHz, the minimum number of waiting periods is 0; When 40 MHz<HCLK<=80 MHz, the minimum number of waiting periods is 1; When 80MHz<HCLK<=120MHz, the minimum number of waiting periods is 2; When 120MHz<HCLK<=160MHz, the minimum number of waiting periods is 3; When 160MHz<HCLK<=200MHz, the minimum number of waiting periods is 4; When 200MHz<HCLK<=240MHz, the minimum number of waiting periods is 5.

Note: When the number of wait cycles is not zero, enable prefetch buffer can improve overall efficiency..

2.2.1.3 Unlock Flash

After reset, the Flash module is protected and cannot be written into the FLASH_CTRL register to prevent accidental operation of Flash memory due to electrical disturbances and other reasons. By writing a specific sequence of key values into the FLASH_KEY register, you can unlock the FLASH_CTRL register. The specific sequence is: first, writing KEY1 = 0x45670123 to the FLASH_KEY register, and then writing KEY2 = 0xCDEF89AB to the FLASH_KEY register.

If there is an error in sequence or key value, a bus error will occur and the FLASH_CTRL register will be locked until the next reset. Software can check the FLASH_CTRL.LOCK bit to confirm whether the Flash is unlocked. If normal locking is required, software can set the FLASH_CTRL.LOCK bit to 1. After that, the Flash can be unlocked by writing the correct key sequence to the FLASH_KEY register.

2.2.1.4 Erase and program

Note: When programming or erasing Flash, it is not allowed to be interrupted, otherwise it may result in data loss

2.2.1.4.1 Erase operations of main Flash area

The main memory block can be erased by page or by chip(Mass Erase).

Page Erase

Page Erase process:

- Check the FLASH_STS.BUSY bit to confirm that there are no other flash operations in progress;
- Set the FLASH_CTRL.PER bit to '1';
- Select the page to be erased with the FLASH_ADD register;
- Set the FLASH_CTRL.START bit to '1';
- Wait for the FLASH_STS.BUSY bit to change to '0';
- Read out the content of the erased page and verify them.

Mass Erase

Mass Erase process:

- Check the FLASH_STS.BUSY bit to confirm that there are no other flash operations in progress;

- Set the FLASH_CTRL.MER bit to '1';
- Set the FLASH_CTRL.START bit to '1';
- Wait for the FLASH_STS.BUSY bit to change to '0';
- Read out all pages and verify them.

Note: When erasing or writing to Flash, access to Flash is held until the Flash erase or write operation is completed. If program execution is required during Flash write, it is recommended to move the program to SRAM for execution. If interrupts need to be responded to, Flash operations, interrupt handling, and interrupt vector tables should all run in SRAM.

2.2.1.4.2 Main Flash area programming

The main Flash area can be programmed 64-bit at a time. When the FLASH_CTRL.PG bit is' 1', writing two words into a flash address will start programming once; Writing any half word of data will result in a bus error. During the programming process (the FLASH_STS.BUSY bit is' 1'), any operation of reading or writing the flash memory will cause the CPU to pause until the end of the flash programming.

Main memory programming process:

- Check the FLASH_STS.BUSY bit to confirm that there are no other flash operations in progress;
- Set the FLASH_CTRL.PG bit to '1';
- Write the double word to be programmed at the specified address;
- Wait for the FLASH_STS.BUSY bit to change to '0';
- Read the data of written address and verify the data.

Note: When the FLASH_STS.BUSY bit is '1', you cannot write to any register.

Note: After writing data to two addresses in programming, it is necessary to add 9 __NOP() to prevent the bus from reading Flash data and causing programming failure

2.2.1.4.3 OptionByte erase and programming

The OptionByte area is programmed differently from the main Flash block. The number of OptionBytes is limited to 12 bytes (4 bytes for write protection, 2 bytes for read protection, 3 bytes for configuration, 2 bytes for storing user data and 1 byte for CCMSRAM_RST). After unlocking the Flash, you must write KEY1 and KEY2 respectively (refer to 2.2.1.3) to the FLASH_OPTKEY register, and then set the FLASH_CTRL.OPTWE bit to '1'. At this time, the OptionByte area can be programmed, set the FLASH_CTRL.OPTPG bit to' 1' and then write a word to the specified address.

When programming the word in the OptionByte area, use the low byte in the half-word and automatically calculate the high byte (the high byte is the complement of the low byte) before starting the programming operation. This ensures that the OptionByte and its complement are always correct.

OptionByte erase process:

- Check the FLASH_STS.BUSY bit to confirm that there are no other flash operations in progress;
- Unlock the FLASH_CTRL.OPTWE bit;

- Set the FLASH_CTRL.OPTER bit to '1';
- Set the FLASH_CTRL.START bit to '1';
- Wait for the FLASH_STS.BUSY bit to change to '0';
- Read the erased OptionByte and verify them.

OptionByte area programming process:

- Check the FLASH_STS.BUSY bit to confirm that there are no other flash operations in progress;
- Unlock the FLASH_CTRL.OPTWE bit;
- Set the FLASH_CTRL.OPTPG bit to '1';
- Writing the word to be programmed to the specified address;
- Wait for the FLASH_STS.BUSY bit to change to '0';
- Read the written address and verify the data.

2.2.1.4.4 ROW program

The operations for row programming and double-word programming are the same. Before starting programming, the following registers need to be configured:

- Set FLASH_ROWP.RPEN to '1' to select row programming;
- Set FLASH_ROWP.RPNUM bit to select the word length for row programming;
- Set FLASH_ROWP.RPADD bit to select the starting address for row programming;
- Set FLASH_ROWP.RPAREA bit to select the area for row programming.

Note: 1.When writing data to the specified address for programming, the programming address must match the starting address and word length for row programming.

2. ROW programming cannot span 512 bytes

2.2.1.5 ECC function

The Flash module supports the ECC functionality, enabling 2-bit error detection and 1-bit error correction. ECC encoding and decoding (error correction, error detection) are automatically performed by hardware. If an error is detected, the error bit is set and an interrupt is generated.

2.2.1.6 Instruction prefetching

The Flash module supports instruction prefetch function with the prefetch buffer size of 32B. Through instruction prefetching, the instruction execution efficiency of CPU can be improved. The instruction prefetch function can be configured to be enabled or disabled through the register, and it is enabled by default.

2.2.1.7 OptionByte

OptionByte block is mainly used to configure read/write protection, software/hardware watchdog configuration, boot management, BOR threshold levels selection and reset options when the system is in STANDBY/STOP mode. They consist of 12 options bytes : 4 bytes for write protection, 2 bytes for read protection, 3 bytes for configuration option, 2 bytes defined by user, and 1 byte for CCMSRAM_RST. These 12 bytes need to be written through the bus. The

OptionByte block also contains the complement codes corresponding to these 12 OptionBytes. These complement codes need to be automatically calculated by hardware when the OptionBytes are written, and used for verification when the OptionBytes are read.

By default, the OptionByte block is always read-accessible and write-protected. To write (program/erase) the OptionByte block, first unlock the Flash, then unlock the OptionByte: write the correct key value sequence (KEY1 = 0x45670123, KEY2 = 0xCDEF89AB) into the FLASH_OPTKEY, and then write operation to the OptionByte block will be allowed. If the sequence is wrong or the key value is wrong, a bus error will be returned and the OptionByte will be locked until the next reset. To lock the OptionByte nomally, write '0' to the FLASH_CTRL.OPTWE bit by software. The OptionByte can be unlocked by writing the correct key value sequence in the FLASH_OPTKEY.

After each system reset, the OptionByte data is read out from the OptionByte block and stored it in the OptionByte register (FLASH_OB/FLASH_WRP) with read-only property. At the same time, the OptionByte complement data read out together will be used to verify whether the OptionByte data is correct. If it does not match, an OptionByte error flag (FLASH_OB.OBERR) will be generated. When an OptionByte error occurs, the corresponding OptionByte is forced to 0xFF. When the OptionByte and its complement are both 0xFF (the state after erasing), the above verification steps are skipped and verification is not required.

Table 2-3 Option Byte List

Address	[31:24] Corresponding complement code	[23:16] Option byte	[15:8] Corresponding complement code	[7:0] Option byte
0x1FFF_E000	nUSER	USER	nRDP1	RDP1
0x1FFF_E004	nData1	Data1	nData0	Data0
0x1FFF_E008	nWRP1	WRP1	nWRP0	WRP0
0x1FFF_E00C	nWRP3	WRP3	nWRP2	WRP2
0x1FFF_E010	nUSER2	USER2	nRDP2	RDP2
0x1FFF_E014	-	-	nUSER3	USER3
0x1FFF_E018	-	-	nCCMSRAM_RST	CCMSRAM_RST

- Read protection L1 level OptionByte: RDP1
  - ◆ Protect the code stored in the Flash memory;
  - ◆ When the correct value is written, it is not allowed to read the Flash memory;
  - ◆ The result of whether RDP1 is turned on or not can be inquired through FLASH_OB[1];
- User configuration options: USER
  - ◆ USER[7]: Reserved
  - ◆ USER[6]: IWDGSLEEPFRZ, can be queried through FLASH_OB[7];

    0: IWDG freeze in SLEEP mode

    1: IWDG unfreeze in SLEEP mode
  - ◆ USER[5]: IWDGSTDBYFRZ, can be queried through FLASH_OB[6];

    0: IWDG freeze in STANDBY mode

1: IWDG unfreeze in STANDBY mode

- ◆ USER[4]: Reserved;
- ◆ USER[3]: IWDGSTOPFRZ, can be queried through FLASH_OB[5];

  0: IWDG freeze in STOP mode

  1: IWDG unfreeze in STOP mode

- ◆ USER[2]: nRST_STDBY configuration options, read through FLASH_OB[4]

  0: Reset occurs when entering STANDBY mode

  1: No reset occurs when entering STANDBY mode

- ◆ USER[1]: nRST_STOP, read through FLASH_OB[3]

  0: Reset occurs when entering STOP mode

  1: No reset occurs when entering STOP mode

- ◆ USER[0]: WDG_SW configuration options, read through FLASH_OB[2]

  0: Hardware watchdog

  1: Software watchdog

- ● 2 bytes of user data: Datax
  - ◆ Data1 (FLASH_OB[27:20]);
  - ◆ Data0 (FLASH_OB[19:12]);
- ● Write protection OptionByte: WRP0 ~ 3, which can be inquired through the register FLASH _WRP[31:0]
  - ◆ WRP0: write protection of pages 0-15, bit [0] corresponds to Page0/1, bit [7] corresponds to page14/ 15;
  - ◆ WRP1: write protection of pages 16-31, bit [0] corresponds to Page16/17, bit [7] corresponds to Page30/31;
  - ◆ WRP2: write protection of pages 32-47, bit [0] corresponds to Page32/33, bit [7] corresponds to Page46/47;
  - ◆ WRP3: write protection of pages 48-63, bit [0] corresponds to Page48/49, bit [7] corresponds to Page62/63;
- ● Read protection L2 level OptionByte: RDP2
  - ◆ Add protection function on the basis of L1, refer to the detailed description of read protection in section 2.2.1.9;
  - ◆ The result of whether RDP2 is turned on or not can be inquired through FLASH_OB[31];
- ● User Configuration: USER2
  - ◆ USER2[7:5]: BOR_LVL, BOR reset voltage control:

    000: Reset level threshold 1.6V (rising 1.66V/falling 1.62V), default level

    001: Reset level threshold 2.0V (rising 2.1V/falling 2.0V)

    010: Reset level threshold 2.2V (rising 2.3V/falling 2.2V)

    011: Reset level threshold 2.5V (rising 2.6V/falling 2.5V)

100: Reset level threshold 2.8V (rising 2.9V/falling 2.8V)

- USER2[4]: Reserved
- USER2[3]: FLASH_BOOT control, valid when the boot partition is main Flash:

  0: Boot from the rear bank of the main flash

  1: Boot from the front bank of the main flash, default value
- USER2[2] : nSWBOOT0, default value is 1
- USER2[1]: nBOOT1, default value is 1
- USER2[0]: nBOOT0, default value is 1

- User Configuration: USER3
  - USER3[7:2]: Reserved
  - USER3[1:0]: NRST_SEL [1:0] control bit, PH6 function selection:

    00: NRST input and output

    01: NRST input

    10: GPIO function

    11: NRST input and output
- CCMSRAM_RST: Configure whether CCMSRAM is erased after reset release
  - Value is 0xAB, CCMSRAM_RSTEN is 0, perform erase
  - Value is not 0xAB, CCMSRAM_RSTEN is 1, do not perform erase

2.2.1.8 Write protect

Write protection can be configured for all pages of the Flash main memory area (maximum 512 KB), to prevent accidental write operations caused by program crashes or electrical disturbances. The basic unit of write protection is as follows: every 2 pages is a basic protection unit. Write protection can be configured by setting WRP0 to WRP3 in the OptionByte block; After each configuration, a system reset is required for the configured value to be reloaded to take effect. If a protected page is programmed or erased, a protection error flag will be returned in the FLASH_STS.

The system information area contains the following blocks:

- The system memory block (16KB) in the system information area stores the boot program and cannot be changed.
- The system configuration block (8KB) in the system information area stores the basic information of the chip and cannot be changed.
- The OptionByte block (8KB) in the system information area stores the user-configurable OptionByte information. The write protection of the OptionByte block is achieved by writing 0 to the FLASH_CTRL.OPTWE bit by software, and after that, you can write the correct key value sequence to FLASH_OPTKEY to release the write protection of the OptionByte.

2.2.1.9 Read protection

The user code in flash can be protected against unauthorized reading by setting read protection. Read protection is mainly targeted at protecting access operations to main Flash block and OptionByte block after chip sealing operation is completed. Read protection is set by configuring RDP bytes in the OptionByte block. Three different read protection levels can be configured, as shown in the following Table

Table 2-4 Read Protection Configuration List

Read Protection Level	RDP1	nRDP1	nRDP2	RDP2
L0 level	0xA5	0x5A	RDP2! = 0xCC || nRDP2! = 0x33	
L2 level	0xXX	0xXX	0x33	0xCC
L1 level	Not the above two configurations			

- L0 level:
  - In unprotected state, (RDP1 == 0xA5 & nRDP1 == 0x5A) && (RDP2!= 0xCC | nRDP2!= 0x33);
  - The main Flash area and OptionBytes can be read arbitrarily;
  - The main Flash area and OptionBytes can be programmed and erase, with configurable read/write protection.
- L1 level:
  - The corresponding ~ (((RDP1 == 0xA5 & nRDP1 == 0x5A) && (RDP2!= 0xCC | nRDP2!= 0x33)) | (RDP2 == 0xCC & nRDP2 == 0x33));
  - Only the read operation of the main Flash area from the user code is allowed, that is, the read operation of the main Flash area is permitted only when the program is started from the main Flash memory in non debugging mode,
  - All pages can be programmed through the code executed in the main Flash memory (using for IAP or data storage and other functions);
  - All pages are not allowed to write or erase (except for mass erase) in debug mode or after booting from embedded SRAM;
  - The function of loading code into the embedded SRAM through JTAG/SWD remains effective, and it can be started from the embedded SRAM through JTAG/SWD, which can be used to remove read protection;
  - When the read-protected OptionByte is reprogrammed to the value 0xA5 to move back to the unprotected L0 level, all the main Flash areas will be automatically erased, and the process is as follows: (Erasing the OptionByte block will not trigger Mass Erase operation, because the result of erasing is 0xFF, which is equivalent to remaining in the protection state of L1 level)
    - Write the correct key value sequence into FLASH_OPTKEY to unlock the OptionByte block;
    - The bus initiates a command to erase the entire OptionByte area (Page erase);
    - Bus write 0xA5 to read protection OptionByte;
    - Automatically erase all main memory area by hardware;
    - Automatically write 0xA5 to read protection OptionByte by hardware;
    - When the system is reset (such as software reset, etc.), the OptionByte block (including the new RDP value 0xA5) will be reloaded into the system, and the read protection will be released;
  - The following access operations to the Flash memory will be prohibited:

- ➢ Boot from embedded SRAM to execute code to access main Flash memory (including using DMA);
- ➢ Access the main flash memory by JTAG, SWV (serial wire viewr), SWD (serial wire debug) and boundary scanning;

- L2 level: Except that SRAM boot disabled, debug mode disabled, OptionByte write/page erase disabled and the protection level cannot be modified (irreversible), other features are the same as L1 level. The L2 level is realized by configuring another OptionByte, RDP2. No matter what the value of RDP1 is, as long as it satisfies (RDP2==0xCC & nRDP2==0x33), it is L2 level.

Table 2-5 Flash Read-Write-Erase(1) Permission Control Table

Boot Mode: System Memory										
RDP Level	User Access Area		Region Size	System Memory	Main Flash	FEMC/XSPI Memory		CCM SRAM(2)(4)	SRAM (5)	JTAG&SWD
						Encrypt/Decrypt Segment (2)	Non-Encrypt/Non-Decrypt Segment			
L0	System memory		16KB	RWE	NA	NA	NA	RWE(2)/ NA(2)	NA	NA
	Option Byte		8KB	RWE	RWE	RWE	RWE	RWE	RWE	RWE
	Main Flash	NOT WRP	512KB	RWE/ME	RWE/ME	RWE/ME	RWE/ME	RWE/ME	RWE/ME	RWE/ME
		WRP		RO	RO	RO	RO	RO	RO	RO
	FEMC/ xSPI memory(2)(6)	Encrypt/Decrypt Segment	User config	RW(2)/ NA(2)	RW(2)/ NA(2)	RW(2)/ NA(2)	NA	RW(2)/ NA(2)	RW(2)/ NA(2)	NA
		Non-Encrypt/Non-Decrypt Segment		RW	RW	RW	RW	RW	RW	RW
	CCM SRAM(2)	NOT WRP	32KB	RW(2)/ NA(2)	RW(2)/ NA(2)	RW(2)/ NA(2)	RW(2)/ NA(2)	RW(2)/ NA(2)	RW(2)/ NA(2)	NA
		WRP		RO2)/ NA(2)	RO(2)/ NA(2)	RO(2)/ NA(2)	RO(2)/ NA(2)	RO(2)/ NA(2)	RO(2)/ NA(2)	NA
	SRAM(4)	Enable debug access	160KB/ 192KB	RW	RW	RW	RW	RW	RW	RW
		Disable debug access		RW	RW	RW	RW	RW	RW	NA
	Backup SRAM		4KB	RW	RW	RW	RW	RW	RW	RW
	Backup Register		20*32bits	RW	RW	RW	RW	RW	RW	RW

L1	System Memory		16KB	RWE	NA	NA	NA	RWE[2]/ NA[2]	NA	NA
	Option Byte		8KB	RWE	RWE	RWE	RWE	RWE	RWE	RWE
	Main Flash	NOT WRP	512KB	RWE/ME	RWE/ME	RWE/ME	NA	RWE/ME	NA	NA
		WRP		RO	RO	RO	NA	RO	NA	NA
	FEMC/ xSPI memory[2][6]	Encrypt/Decrypt Segment	User config	RW[2]/ NA[2]	RW[2]/ NA[2]	RW[2]/ NA[2]	NA	RW[2]/ NA[2]	RW[2]/ NA[2]	NA
		Non-Encrypt/Non-Decrypt Segment		RW	RW	RW	RW	RW	RW	RW
	CCM SRAM[2]	NOT WRP	32KB	RW[2]/ NA[2]	RW[2]/ NA[2]	RW[2]/ NA[2]	RW[2]/ NA[2]	RW[2]/ NA[2]	RW[2]/ NA[2]	NA
		WRP		RO[2]/NA[2]	RO[2]/NA[2]	RO[2]/NA[2]	RO[2]/NA[2]	RO[2]/NA[2]	RO[2]/NA[2]	NA
	SRAM[4]	Enable debug access	160KB/ 192KB	RW	RW	RW	RW	RW	RW	RW
		Disable debug access		RW	RW	RW	RW	RW	RW	NA
	Backup SRAM		4KB	NA	RW[3]	RW[2]/ NA[2]	NA	RW[2]/ NA[2]	NA	NA
	Backup Register		20*32bits	NA	RW[3]	RW[2]/ NA[2]	NA	RW[2]/ NA[2]	NA	NA
L2	System Memory		16KB	RWE	NA	NA	NA	RWE[2]/ NA[2]	NA	JTAG/ SWD disabled
	Option Byte		8KB	RO	RO	RO	RO	RO	RO	
	Main Flash	NOT WRP	512KB	RWE/ME	RWE/ME	RWE/ME	NA	RWE/ME	NA	
		WRP		RO	RO	RO	NA	RO	NA	
	FEMC/xSPI memory	Encrypt/Decrypt Segment	User config	RW[2]/ NA[2]	RW[2]/ NA[2]	RW[2]/ NA[2]	NA	RW[2]/ NA[2]	RW[2]/ NA[2]	
		Non-Encrypt/Non-Decrypt Segment		RW	RW	RW	RW	RW	RW	
	CCM SRAM	NOT WRP	32KB	RW[2]/ NA[2]	RW[2]/ NA[2]	RW[2]/ NA[2]	RW[2]/ NA[2]	RW[2]/ NA[2]	RW[2]/ NA[2]	

		WRP		RO[2]/ NA[2]	RO[2]/ NA[2]	RO[2]/ NA[2]	RO[2]/ NA[2]	RO[2]/ NA[2]	RO[2]/ NA[2]	
	SRAM[4]	Enable debug access	160KB/ 192KB	RW	RW	RW	RW	RW	RW	
		Disable debug access		RW	RW	RW	RW	RW	RW	
	Backup SRAM		4KB	NA	RW[3]	RW[2]/ NA[2]	NA	RW[2]/ NA[2]	NA	
	Backup Register		20*32bits	NA	RW[3]	RW[2]/ NA[2]	NA	RW[2]/ NA[2]	NA	
Boot Mode: Main Flash										
RDP Level	User Access Area		Region Size	System Memory	Main Flash	FEMC/XSPI Memory		CCM SRAM[2][4]	SRAM[5]	JTAG& SWD
						Encrypt/Decrypt Segment (2)	Non-Encrypt/Non-Decrypt Segment			
L0	System Memory		16KB	RWE	NA	NA	NA	RWE[2]/ NA[2]	NA	NA
	Option Byte		8KB	RWE	RWE	RWE	RWE	RWE	RWE	RWE
	Main Flash	NOT WRP	512KB	RWE/ME	RWE/ME	RWE/ME	RWE/ME	RWE/ME	RWE/ME	RWE/ME
		WRP		RO	RO	RO	RO	RO	RO	RO
	FEMC/ xSPI memory[2][6]	Encrypt/Decrypt Segment	User config	RW[2]/ NA[2]	RW[2]/ NA[2]	RW[2]/ NA[2]	NA	RW[2]/ NA[2]	RW[2]/ NA[2]	NA
		Non-Encrypt/Non-Decrypt Segment		RW	RW	RW	RW	RW	RW	RW
	CCM SRAM[2]	NOT WRP	32KB	RW[2]/ NA[2]	RW[2]/ NA[2]	RW[2]/ NA[2]	RW[2]/ NA[2]	RW[2]/ NA[2]	RW[2]/ NA[2]	NA
		WRP		RO2)/ NA[2]	RO[2]/ NA[2]	RO[2]/ NA[2]	RO[2]/ NA[2]	RO[2]/ NA[2]	RO[2]/ NA[2]	NA
	SRAM[4]	Enable debug access	160KB/ 192KB	RW	RW	RW	RW	RW	RW	RW
		Disable debug access		RW	RW	RW	RW	RW	RW	NA

	Backup SRAM		4KB	RW	RW	RW	RW	RW	RW	RW
	Backup Register		20*32bits	RW	RW	RW	RW	RW	RW	RW
L1	System Memory		16KB	RWE	NA	NA	NA	RWE[2]/ NA[2]	NA	NA
	Option Byte		8KB	RWE	RWE	RWE	RWE	RWE	RWE	RWE
	Main Flash	NOT WRP	512KB	RWE/ME	RWE/ME	RWE/ME	NA	RWE/ME	NA	NA
		WRP		RO	RO	RO	NA	RO	NA	NA
	FEMC/ xSPI memory[2][6]	Encrypt/Decrypt Segment	User config	RW[2]/ NA[2]	RW[2]/ NA[2]	RW[2]/ NA[2]	NA	RW[2]/ NA[2]	RW[2]/ NA[2]	NA
		Non-Encrypt/Non-Decrypt Segment		RW	RW	RW	RW	RW	RW	RW
	CCM SRAM[2]	NOT WRP	32KB	RW[2]/ NA[2]	RW[2]/ NA[2]	RW[2]/ NA[2]	RW[2]/ NA[2]	RW[2]/ NA[2]	RW[2]/ NA[2]	NA
		WRP		RO[2]/ NA[2]	RO[2]/ NA[2]	RO[2]/ NA[2]	RO[2]/ NA[2]	RO[2]/ NA[2]	RO[2]/ NA[2]	NA
	SRAM[4]	Enable debug access	160KB/ 192KB	RW	RW	RW	RW	RW	RW	RW
		Disable debug access		RW	RW	RW	RW	RW	RW	NA
	Backup SRAM		4KB	NA	RW[3]	RW[2]/ NA[2]	NA	RW[2]/ NA[2]	NA	NA
	Backup Register		20*32bits	NA	RW[3]	RW[2]/ NA[2]	NA	RW[2]/ NA[2]	NA	NA
L2	System Memory		16KB	RWE	NA	NA	NA	RWE[2]/ NA[2]	NA	JTAG/ SWD disabled
	Option Byte		8KB	RO	RO	RO	RO	RO	RO	
	Main Flash	NOT WRP	512KB	RWE/ME	RWE/ME	RWE/ME	NA	RWE/ME	NA	
		WRP		RO	RO	RO	NA	RO	NA	
	FEMC/ xSPI memory	Encrypt/Decrypt Segment	User config	RW[2]/ NA[2]	RW[2]/ NA[2]	RW[2]/ NA[2]	NA	RW[2]/ NA[2]	RW[2]/ NA[2]	
		Non-Encrypt/Non-Decrypt Segment		RW	RW	RW	RW	RW	RW	

	CCM SRAM	NOT WRP	32KB	RW(2)/ NA(2)	RW(2)/ NA(2)	RW(2)/ NA(2)	RW(2)/ NA(2)	RW(2)/ NA(2)	RW(2)/ NA(2)	
		WRP		RO(2)/ NA(2)	RO(2)/ NA(2)	RO(2)/ NA(2)	RO(2)/ NA(2)	RO(2)/ NA(2)	RO(2)/ NA(2)	
	SRAM(4)	Enable debug access	160KB/ 192KB	RW	RW	RW	RW	RW	RW	
		Disable debug access		RW	RW	RW	RW	RW	RW	
	Backup SRAM		4KB	NA	RW(3)	RW(2)/ NA(2)	NA	RW(2)/ NA(2)	NA	
	Backup Register		20*32bits	NA	RW(3)	RW(2)/ NA(2)	NA	RW(2)/ NA(2)	NA	
Boot Mode: SRAM										
RDP Level	User Access Area		Region Size	System Memory	Main Flash	FEMC/XSPI Memory		CCM SRAM(2)(4)	SRAM(5)	JTAG& SWD
						Encrypt/Decrypt Segment (2)	Non-Encrypt/Non-Decrypt Segment			
L0	System Memory		16KB	RWE	NA	NA	NA	RWE(2)/NA(2)	NA	NA
	Option Byte		8KB	RWE	RWE	RWE	RWE	RWE	RWE	RWE
	Main Flash	NOT WRP	512KB	RWE/ME	RWE/ME	RWE/ME	RWE/ME	RWE/ME	RWE/ME	RWE/ME
		WRP		RO	RO	RO	RO	RO	RO	RO
	FEMC/ xSPI memory(2)(6)	Encrypt/Decrypt Segment	User config	RW(2)/ NA(2)	RW(2)/ NA(2)	RW(2)/ NA(2)	NA	RW(2)/ NA(2)	RW(2)/ NA(2)	NA
		Non-Encrypt/Non-Decrypt Segment		RW	RW	RW	RW	RW	RW	RW
	CCM SRAM(2)	NOT WRP	32KB	RW(2)/ NA(2)	RW(2)/ NA(2)	RW(2)/ NA(2)	RW(2)/ NA(2)	RW(2)/ NA(2)	RW(2)/ NA(2)	NA
		WRP		RO(2)/ NA(2)	RO(2)/ NA(2)	RO(2)/ NA(2)	RO(2)/ NA(2)	RO(2)/ NA(2)	RO(2)/ NA(2)	NA
	SRAM(4)	Enable debug access	160KB/ 192KB	RW	RW	RW	RW	RW	RW	RW
		Disable debug access		RW	RW	RW	RW	RW	RW	NA

	Backup SRAM		4KB	RW	RW	RW	RW	RW	RW	RW
	Backup Register		20*32bits	RW	RW	RW	RW	RW	RW	RW
L1	System Memory		16KB	RWE	NA	NA	NA	RWE[2]/NA[2]	NA	NA
	Option Byte		8KB	RWE	RWE	RWE	RWE	RWE	RWE	RWE
	Main Flash	NOT WRP	512KB	RWE/ME	RWE/ME	RWE/ME	NA	RWE/ME	NA	NA
		WRP		RO	RO	RO	NA	RO	NA	NA
	FEMC/ xSPI memory[2][6]	Encrypt/Decrypt Segment	User config	RW[2]/NA[2]	RW[2]/NA[2]	RW[2]/NA[2]	NA	RW[2]/NA[2]	RW[2]/NA[2]	NA
		Non-Encrypt/Non-Decrypt Segment		RW	RW	RW	RW	RW	RW	RW
	CCM SRAM[2]	NOT WRP	32KB	RW[2]/NA[2]	RW[2]/NA[2]	RW[2]/NA[2]	RW[2]/NA[2]	RW[2]/NA[2]	RW[2]/NA[2]	NA
		WRP		RO[2]/NA[2]	RO[2]/NA[2]	RO[2]/NA[2]	RO[2]/NA[2]	RO[2]/NA[2]	RO[2]/NA[2]	NA
	SRAM[4]	Enable debug access	160KB/ 192KB	RW	RW	RW	RW	RW	RW	RW
		Disable debug access		RW	RW	RW	RW	RW	RW	NA
	Backup SRAM		4KB	NA	RW[3]	RW[2]/NA[2]	NA	RW[2]/NA[2]	NA	NA
	Backup Register		20*32bits	NA	RW[3]	RW[2]/NA[2]	NA	RW[2]/NA[2]	NA	NA
L2	System Memory		16KB	L2 protection level, unable to boot from SRAM						JTAG/ SWD disabled
	Option Byte		8KB							
	Main flash	NOT WRP	512KB							
		WRP								
	FEMC/ xSPI memory	Encrypt/Decrypt Segment	User config							
		Non-Encrypt/Non-Decrypt Segment								
	CCM SRAM	NOT WRP	32KB							
		WRP								

	SRAM(4)	Enable debug access	160KB/ 192KB		
		Disable debug access			
	Backup SRAM		4KB		
	Backup Register		20*32bits		

Notes:

(1) R=Read, W=Write, E=Erase, RO=Read only, PE=Page Erase, ME=Mass Erase, NA=NOT Access, stands for Not Applicable.

(2) The access permissions (User) and the accessed permissions for the encrypt/decrypt segments of FEMC/XSPI memory and CCM SRAM are the same as those set by the user.

(3) Access granted only to USER1.

(4) CCM SRAM, logical address range after CCM is enabled

(5) The maximum capacity of SRAM includes two scenarios: 160KB and 192KB. When CCM is enabled, the maximum capacity of SRAM is 160KB, and when CCM is disabled, the maximum capacity is 192KB.

(6) Only applicable for XIP mode read and write, accessed as an external communication interface, no control.

(7) DMA access to System memory, Main flash, FEMC/XSPI memory, SRAM is the same as the access permissions for the corresponding storage when the DMA is started by the user.

(8) When sending read/write/erase commands to FEMC/XSPI storage peripherals via the interface, no permission management will be performed, and permission management will only be carried out when accessing in XIP mode.

2.2.2 ICache

To achieve higher system performance, an instruction cache needs to be added between the high-speed CPU and the low-speed Flash to improve the instruction execution efficiency. Because of the existence of the instruction cache, the CPU will be able to work at a higher frequency. When the instruction requested by the CPU is in the instruction cache, the CPU can obtain the instruction without delay and enabling zero waiting for execution. When the current instruction sequence, instruction prefetch sequence and instruction cache all miss, Flash will be re-read and the Cache will be updated accordingly. Consequently, the cache mainly stores the program's entry points..

The main features of the instruction cache are as follows:

- 8 KB ICache.
- The degree of cache associativity: 4 Way

Note: ICache corresponds to main Flash and System memory area, once ICache is turned on (default off), then the program can not jump between these two areas during running, unless ICache is turned off and ICahce is cleared first.

2.2.2.1 Software interface

- Enable
  - Provide software configuration to enable/disable ICache. The switch can be toggle without any restrictions(see the FLASH_AC.ICAHEN bit).
- Reset
  - Provide software interface to clear the ICache, it must be initiated when ICache is disabled. Reset and Enable cannot be effective at the same time. First, turn off FLASH_AC.ICAHEN, then write 1 to FLASH_AC.ICAHRST, and then turn on FLASH_AC.ICAHEN.
- Lock
  - Cache locking mechanism is supported, and the software configuration puts the program into its designated way. When all the ways are locked, the new data will not be written into the cache. After the software resets the cache, the lock state is automatically cleared.
- Additional remarks
  - Selection of Cache replacement algorithm is not supported.
  - When using ICache, there is no Write-Back(WB) or Write-Through(WT) selection when the CPU writes operation.

2.2.2.2 Register description

FLASH_AC.ICAHEN and FLASH_AC.ICAHRST are the ICache enable and ICache data clear respectively.

FLASH_CAHR.LOCKSTRT and FLASH_CAHR.LOCKSTOP respectively control the enbale and disable of the lock mechanism for ICache. After ICache is reset, the FLASH_CAHR register automatically returns to the reset value. See for detailed usage method of 2.2.2.3.3 ICache locking.

2.2.2.3 Operating process

2.2.2.3.1 ICache enable and disable

Users can turn on and switch off ICache at any time. If the user program needs to jump between the main memory area and other memory areas, the ICache must be closed and the data of the ICache must be cleared, otherwise, the instruction acquisition error will occur.

2.2.2.3.2 ICache data refresh

The ICache is designed as instruction cache. When the instruction is updated by application software or the instruction jumps between the main memory area and other memory areas, the software must set the FLASH_AC.ICAHRST bit to 1 to clear the data in the instruction cache.

Note: FLASH_AC.ICAHRST bit is a write-only bit, and it returns to 0 when read.

2.2.2.3.3 ICache locking

The software controls the FLASH_CAHR register to lock some frequently used codes in ICache to improve the efficiency of code execution. ICache module has four latch channels, and the size of each channel is 1/4 of the whole cache. When using a single channel, you must ensure that the amount of code to latch is less than the size of each channel. Otherwise need to use more channels to latch the code. The latch function can be used according to the following control flow:

1. Set FLASH_CAHR.LOCKSTRT[0] to 1;
2. Execute function 1 that needs to be locked in channel 0 (the code amount of function 1 should be less than the size of a single channel);
3. Set FLASH_CAHR.LOCKSTOP[0] to 1 after the function 1 is executed;
4. Then set FLASH_CAHR.LOCKSTRT[1] to 1;
5. Execute function 2 that needs to be locked in channel 1 (the code amount of function 2 should be less than the size of a single channel);
6. After the function 2 is executed, set FLASH_CAHR.LOCKSTOP[1] to 1;

Notes:

1. when the channel is latched, the register operation must follow a fixed process - First set FLASH_CAHR.LOCKSTRT then set FLASH_CAHR.LOCKSTOP;

2. The order of channel latch must be 0~3, otherwise it will reduce the execution efficiency.

2.2.3 SRAM

SRAM is mainly used for code operation to store variables and data or stacks during program execution. The maximum capacity is 160KB.

SRAM supports read-write access of byte, half-word and word.

SRAM supports code execution (supports access of SBus, ICode and DCode), and can run programs at full speed in SRAM. SRAM is divided into SRAM1 and SRAM2. SRAM1 has a maximum capacity of 96K, with a maximum address range of 0x2000_0000 to 0x2001_7FFF; SRAM2 has a maximum capacity of 64K, with a maximum address range of 0x2001_8000 to 0x2002_7FFF.

The address ranges for different capacity SRAM are as follows:

Table 2-6 The Access Address Range of N32H47xxE/H48xxE Series

SRAM Region	**SBUS Access Address Region**	**ICode/DCode Access Address Region**	**Size**
SRAM1	0x2000_0000~0x2001_7FFF	0x1000_0000~0x1001_7FFF	96KB
SRAM2	0x2001_8000~0x2002_7FFF	0x1001_8000~0x1002_7FFF	64KB
SRAM3(CCM SRAM)	0x2002_8000~0x2002_FFFF	0x1002_8000~0x1002_FFFF	32KB

Table 2-7 The Access Address Range of N32H47xxC Series

SRAM Region	**SBUS Access Address Region**	**ICode/DCode Access Address Region**	**Size**
SRAM1	0x2000_0000~0x2001_7FFF	0x1000_0000~0x1001_7FFF	96KB
SRAM2	0x2001_8000~0x2001_BFFF	0x1001_8000~0x1001_BFFF	16KB
SRAM3(CCM SRAM)	0x2001_C000~0x2002_3FFF	0x1001_C000~0x1002_3FFF	32KB

SRAM data cannot be retained in VBAT and Standby modes; data can be retained normally in other operating modes (Run/Sleep/Stop).

The main features are as follows:

- The maximum capacity is 160KB in total.
- Support byte/half-word/word reading and writing.
- I/D/S/DMA1/DMA2 can all access SRAM.
- The I/D bus can be remapped to access the SRAM, allowing programs to execute directly from the SRAM at the full speed of the CPU.
- Support parity check.

At the same time, SRAM also supports one click initialization, which can modify data within the set range to the same initial value. The process is as follows:

1. Configure RCC_SRAMCFG3.INISTART and RCC_SRAMCFG4.INIEND, confirm the starting and ending addresses of the one click initialization range
2. Configure RCC_SRAMCFG2.INIDAT and confirm the initial value
3. Enable RCC_SRAMCFG1.CSRINEN to start one click initialization
4. Wait for RCC_SRAMCFG1.CSRINIF to be set to 1, initialization is complete, and all SRAM data within the configuration range is written to the initial value

Note: Before executing step 1, it is necessary to write an initial value of one word to the starting address.

Note: The ending address cannot be configured to be the same as the starting address.

2.2.4 CCM SRAM

CCM SRAM is also mainly used for code execution, storing variables and data or stack during program execution, with a total capacity of 32KB, supporting ECC check. The bus address of CCM SRAM is contiguous with SRAM, and in application, SRAM and CCM SRAM can be treated as a single block of regular SRAM. In the maximum case, the physical address 0 of CCM SRAM corresponds to the bus starting address of 0x2002 8000, with the corresponding bus address range being 0x2002 8000~0x2002 FFFF. CCM SRAM supports byte, half-word, and word read/write access, and supports access by SBus, DMA1, and DMA2.

When CCM is used as regular SRAM, the read/write protection function is bypassed.

Because the bus address of CCM SRAM is contiguous with SRAM, and the effective capacity of SRAM available for different product models varies, the bus starting address of CCM SRAM is different for different product models.

The main features of CCM SRAM are as follows:

- The total capacity is 32KB (supports ECC check, initialization required before use)
- Support ECC check, detect 2-bit errors, correct 1-bit error
  - ➢ The ECC enable/disable of CCM SRAM is configured through the RCC register, enabled by default, refer to RCC_SRAMCFG1.SRAMPEN for details.
  - ➢ Before enabling ECC check, CCM SRAM should be initialized first, and then ECC check can be started; the initialization configuration of CCM SRAM can be found in registers RCC_SRAMCFG1/2/3/4, the

initialization configuration of CCM should be done before enabling CCM SRAM.

- Byte/half-word/word read/write
- I/D/SBus/DMA1/DMA2 can access
- The bus starting address is contiguous with SRAM
- The bus starting address changes with the capacity of SRAM
- Support write protection
  - Each page size is 1KB, with a total of 32 pages;
  - Write protection is configured through the CCM_WRP1 register, each bit can only be set from 0 to 1 by software, writing to CCM after write protection will result in an error;
  - After setting the write protection, software can erase;
  - Write protection of CCM SRAM can only be removed by system reset.
- Erase
  - Software mass erase: Unlock the write protection of CCM_ERASE.EN bit by sequentially writing 0xCA, 0x53 to CCM_KEY.KEY bit, then set CCM_ERASE.EN bit to 1 to start mass erase of CCM.
  - Automatic mass erase when system reset: Enable/disable full erase of CCM SRAM when system reset through Option Byte CCMSRAM_RST:

    0: enable mass erase

    1: disable mass erase
  - When downgrading from RDP level 1 to level 0, a forced mass erase will be performed, and the erased value is determined by the RCC register
- Read protection
  - Support, see section 2.2.1.9 for details.
  - It is set to L1 by default, but users can change it to L0.

2.2.5 BKP SRAM

BKP SRAM is also mainly used for data storage, with a total capacity of 4KB, supporting ECC check (correct 1-bit error, detect 2-bits errors). The bus address of BKP SRAM is in the peripheral address area, and in application, BKP SRAM can be used as a peripheral register. The physical address 0 of BKP SRAM corresponds to the bus starting address of 0x4002 3000, with the corresponding bus address range being 0x4002 3000 ~ 0x4002 3FFF. BKP SRAM supports byte, half-word, and word read/write access, and supports access by SBus, DMA1, DMA2.

SRAM data can be optionally retained in VBAT and Standby modes; data can be retained normally in other operating modes (Run/Sleep/Stop).

Main features are as follows:

- The maximum total capacity is 4KB

- Byte/half-word/word read/write
- S/DMA1/DMA2 can access
- Support ECC check

2.2.6 FLASH Registers

All register operations must be performed in words (32 bits).

2.2.6.1 FLASH register overview

Table 2-8 FLASH Register Overview

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	FLASH_AC	Reserved																								PRFTBFE	PRFTBFS	ICAHRST	ICAHEN	Reserved	LATENCY		
	Reset value																									1	1	0	0		0	0	0
0x04	FLASH_CTRL	Reserved																RPADDERRITE	DECCERRITE	EC2ERRITE	JSERRITE	EC1ERRITE	FERRITE	ERRITE	EOPITE	OPTWE	OPTER	OPTPG	START	MER	PER	PG	LOCK
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
0x08	FLASH_STS	Reserved															RPADDERR	FWORDF	DECCERR	DECCRDF	ECC2ERR	RTPDKEYERR	JSERR	NRDKEYEN	RDXKEYERR	RDKEYERR	ECC1ERR	EVERR	PVERR	WRPERR	PGERR	BUSY	EOP
	Reset value																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0c	FLASH_ADD	FADD																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x10	FLASH_KEY	FKEY																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x14	FLASH_OPTKEY	OPTKEY																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x18	FLASH_OB	OBERR	nSWBOOT0	nBOOT1	nBOOT0	Data1[7:0]								Data0[7:0]								Reserved		Reserved	IWDGSLEEPFRZ	IWDGSTDBYFR	Reserved	IWDGSTOPFRZ	nRST_STDBY	nRST_STOP	WDG_SW	RDPRT2	RDPRT1
	Reset value	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1			1	1	1	1	1	1	1	1	0	0
0x1c	FLASH_WRP	WRPT																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x20	FLASH_ECC	Reserved																		ECCHW						Reserved		ECCLW					
	Reset value																			0	0	0	0	0	0			0	0	0	0	0	0
0x2c	FLASH_RDN	Reserved							FLASH_RDN1									Reserved							FLASH_RDN0								
	Reset value								0	0	0	0	0	0	0	0	0								0	0	0	0	0	0	0	0	0
0x30	FLASH_CAHR	Reserved																								LOCKSTOP[3:0]				LOCKSTRT[3:0]			

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reset value																									0	0	0	0	0	0	0	0
0x34	FLASH_ROWP	Reserved									RPAREA	RPADD																	RPNUM				RPEN
	Reset value										0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x38	CCM_WRP1	WRPTx																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x3C	CCM_KEY	Reserved																								KEY							
	Reset value																									0	0	0	0	0	0	0	0
0x40	CCM_ERASE	Reserved																														BUSY	EN
	Reset value																															0	0
0x44	CCM_WRP2	WRPTx																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x48	CCM_MODE	Reserved																															EN
	Reset value																																0
0x4C	XSPI_DSTRADD	ADD																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x50	XSPI_DENDADD	ADD																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x54	FEMC_DSTRADD	ADD																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x58	FEMC_DENDADD	ADD																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x5C	RTPD_KEY	KEY																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x60	JTAG_SEAL	Reserved																								DATA							
	Reset value																									0	0	0	0	0	0	0	0
0x64	RTPD_KEY_WCNT	Reserved																													CNT		
	Reset value																														0	0	0
0x6C	XSPI_FEMC_DEN	Reserved																															EN
	Reset value																																0
0x70	XUID	Reserved																						CCMUID					XFUID				
	Reset value																							0	0	0	0	0	0	0	0	0	0

2.2.6.2 FLASH control and status register

For abbreviations related to register descriptions, please refer to section 1.1.

2.2.6.2.1 FLASH access control register (FLASH_AC)

Address offset: 0x00

Reset value: 0x0000 00C0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								PRFTBFE	PRFTBFS	ICAHRST	ICAHEN	Reserved	LATENCY		
								rw	rw	rw	rw			rw	

Bit Field	Name	Description
31:8	Reserved	Reserved, the reset value must be maintained.
7	PRFTBFE	Prefetch buffer enable 0: Disable the prefetch buffer; 1: Enable prefetch buffer.
6	PRFTBFS	Prefetch buffer status This bit indicates the status of the prefetch buffer 0: The prefetch buffer is disable; 1: The prefetch buffer is enable.
5	ICAHRST	ICache reset 0: Writing '0' is invalid; 1: Write '1' to reset.
4	ICAHEN	ICache enable 0: Disable iCache; 1: Enable iCache.
3	Reserved	Reserved, the reset value must be maintained.
2:0	LATENCY	time delay These bits represent the ratio of SYSCLK (system clock) period to flash memory access time. 000: Zero period delay, when 0 < SYSCLK <= 40 MHz 001: One cycle delay, when 40 MHz < SYSCLK <= 80 MHz 010: Two cycle delay, when 80 MHz < SYSCLK <= 120 MHz 011: Three cycle delay, when 120 MHz < SYSCLK <= 160 MHz 100: Four cycle delay, when 160 MHz < SYSCLK <= 200 MHz 101: Five cycle delay, when 200 MHz < SYSCLK <= 240 MHz Other values: reserved

2.2.6.2.2 The FLASH control register (FLASH_CTRL)

Address offset: 0x04

Reset value: 0x0000 0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RPADD ERRITE	DECC ERRITE	EC2ERR ITE	JSERRITE	EC1ERR ITE	FERRITE	ERRITE	EOPITE	OPTWE	OPTER	OPTPG	START	MER	PER	PG	LOCK
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit Field	**Name**	**Description**
31:16	Reserved	Reserved, the reset value must be maintained.
15	RPADDERRITE	Row Programming Address Error Interrupt This bit allows interrupt generation when FLASH_STS.RPADDERR bit becomes '1'. 0: Disable interrupt generation 1: Enable interrupt generation
14	DECCERRITE	XSPI/FEMC Decryption Configuration Error Interrupt This bit allows interrupt generation when FLASH_STS.DECCERR bit goes to '1'. 0: Disable interrupt generation 1: Enable interrupt generation
13	EC2ERRITE	2 bit ECC error interrupt This bit allows an interrupt to be generated when the FLASH_STS.ECC2ERR bit goes to '1'. 0: Disable interrupt generation; 1: Enable interrupt generation.
12	JSERRITE	JTAG seal error interrupt This bit allows interrupt generation when FLASH_STS.JSERR bit becomes '1'. 0: Disable interrupt generation 1: Enable interrupt generation
11	EC1ERRITE	1 bit ECC error interrupt This bit allows an interrupt to be generated when the FLASH_STS.ECC1ERR bit goes to '1'. 0: Disable interrupt generation; 1: Enable interrupt generation.
10	FERRITE	Erase/Program Verify Error Interrupt This bit allows an interrupt to be generated when the FLASH_STS.EVERR/PVERR bit goes to '1'. 0: Interrupt generation is prohibited; 1: Enable interrupt generation.

Bit Field	Name	Description
9	ERRITE	Error status interrupt allowed This bit allows an interrupt to be generated when a Flash error occurs (when FLASH_STS.PGERR/ FLASH_STS.WRPERR is set to '1'). 0: interrupt generation is prohibited; 1: interrupt generation is allowed.
8	EOPITE	Allow operation completion interrupt. This bit allows an interrupt to be generated when the FLASH_STS.EOP bit becomes '1'. 0: Interrupt generation is prohibited; 1: Interrupt generation is allowed.
7	OPTWE	Allow write OptionByte When this bit is '1', the OptionByte is allowed to be programmed. When the correct key sequence is written in the FLASH_OPTKEY register, this bit is set to '1'. Software can clear this bit.
6	OPTER	Erase OptionBytes 0: Disable OptionBytes erase mode; 1: Enable OptionBytes erase mode.
5	OPTPG	Program OptionBytes 0: Disable OptionBytes program mode; 1: Enable OptionBytes program mode.
4	START	Start When this bit is '1', an erase operation will be triggered. This bit can only be set to '1' by software and cleared to '0' when FLASH_STS.BUSY becomes '1'.
3	MER	Mass erase 0: Disable mass erase mode; 1: Enable mass erase mode.
2	PER	Page erase 0: Disable page erase mode; 1: Enable page erase mode
1	PG	Program *Note: Do not set breakpoints during Row/Double Word programming in debug mode, as it may cause programming errors.* 0: Disable program mode; 1: Enable program mode.
0	LOCK	Lock Can only be written as '1'. When this bit is '1', Flash and FLASH_CTRL are locked. After detecting the correct unlocking sequence, hardware clears this bit to '0'. After an unsuccessful unlocking operation, this bit cannot be changed until the next system reset.

Note: Please refer to section 2.2.1.4 for programming and erasing.

2.2.6.2.3 The FLASH status register (FLASH_STS)

Address offset: 0x08

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															RPADD ERR
															rc_w1

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FWORDF	DECCER R	DECCRD F	ECC2ERR	RTPDKEY ERR	JSERR	NRDKEY EN	RDXKEY ERR	RDKEY ERR	ECC1ERR	EVERR	PVERR	WRPERR	PGERR	BUSY	EOP
r	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	r	r	r	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	r	rc_w1

Bit Field	**Name**	**Description**
31:17	Reserved	Reserved, the reset value must be maintained.
16	RPADDERR	Row programming address error (the start address is not the configured start address, or the programming address exceeds the range address of 512 bytes) *Note: After this error occurs, FLASH_ROWP.RPEN needs to be set to 0, and then restart ROW programming.*
15	FWORDF	First-word write flag In double-word programming, this signal is set to 1 after the first word is sent over the bus.
14	DECCERR	XSPI/FEMC decryption configuration sequence error 1) Inconsistent UID of configuring KEY+start address+end address 2) Configuring the above three addresses after decryption enablement
13	DECCRDF	XSPI/FEMC decryption configuration completion flag Indicates whether one or all (XSPI and FEMC) STR address, end address, RTPD_KEY have been configured.
12	ECC2ERR	2 bit ECC error Read FLASH error, hardware set this bit to '1', write '1' to clear this state.
11	RTPDKEYERR	Indicates whether the value of the RTPD_KEY register matches the value of the hardware backup. After configuring the RTPD_KEY, it will be automatically written to the hardware backup. If the values of these two registers do not match, an alert will be issued, indicating that the configuration is in an error state. This can be used to detect unauthorized changes at the hardware level.
10	JSERR	JTAG or SRAM access JTAG_SEAL register error Error reported when JTAG or SRAM access JTAG_SEAL register, hardware sets this bit to '1', writing '1' can clear this status.
9	NRDKEYEN	Disable Read RTPD_KEY Enable 0: Allow reading RTPD_KEY value through SystemMemory API 1: Disallow reading RTPD_KEY value through SystemMemory API *Note: The set to 1 operation is only allowed once, cannot be set back to 0 after*

Bit Field	Name	Description
		setting to 1, decryption key cannot be read again in this chip.
8	RDXKEYERR	xFlash Key Error 0: FEMC/XSPI memory decryption key matches its backup value 1: FEMC/XSPI memory decryption key does not match its backup value
7	RDKEYERR	Read KEY Error Error reported during KEY reading, hardware sets this bit to '1'.
6	ECC1ERR	1 bit ECC error Read FLASH error, hardware set this bit to '1', write '1' to clear this state.
5	EVERR	Erase verify error When the page is erased and the check reports an error, the hardware sets this bit to '1', and writing '1' can clear this state.
4	PVERR	Programming verification error When an error is reported during verification after programming, the hardware sets this bit to '1', and writing '1' can clear this state.
3	WRPERR	Write protection error When trying to program a write-protected flash address, the hardware sets this bit to '1', and writing '1' can clear this bit.
2	PGERR	Programming error When trying to program an address whose content is not '0xFFFF_FFFF', the hardware sets this bit to '1', and writing '1' can clear this state. *Note: Before programming, the FLASH_CTRL.START bit must be cleared.*
1	BUSY	Busy This bit indicates that a flash operation is in progress. At the beginning of flash operation, this bit is set to '1'; This bit is cleared to '0' when the operation ends or an error occurs.
0	EOP	End of operation When the flash operation (programming/erasing) is completed, the hardware sets this bit to '1', and writing '1' can clear this bit status. *Note: Every successful programming or erasing will set the EOP state.*

2.2.6.2.4 The FLASH address register (FLASH_ADD)

Address offset: 0x0C

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FADD[31:16]															
w															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FADD[15:0]															
w															

Bit Field	**Name**	**Description**
31:0	FADD	Flash address Select the address to be programmed when programming, and select the page to be erased when page erasing. *Note: When the FLASH_STS.BUSY bit is '1', this register cannot be written.*

2.2.6.2.5 The FLASH key register (FLASH_KEY)

Address offset: 0x10

Reset value: 0xXXXX XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FKEY[31:16]															
w															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FKEY[15:0]															
w															

Bit Field	**Name**	**Description**
31:0	FKEY	Used to unlock the FLASH_CTRL.LOCK bit.

2.2.6.2.6 The FLASH OPTKEY register (FLASH_OPTKEY)

Address offset: 0x14

Reset value: 0xXXXX XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OPTKEY[31:16]															
w															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OPTKEY[15:0]															
w															

Bit Field	**Name**	**Describe**
31:0	OPTKEY	Used to unlock the FLASH_CTRL.OPTWE bit.

2.2.6.2.7 Option byte register (FLASH_OB)

Address offset: 0x18

Reset value: 0x07FF FFFC

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

OBERR	nSW BOOT0	nBOOT1	nBOOT0	Data1								Data0			
r	r	r	r	r								r			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Data0				Reserved		Reserved	IWDGSLE EPFRZ	IWDGST DBYFRZ	Reserved	IWDG STOPFRZ	nRST_ STOP	nRST_ STDBY	IWDG_ SW	RDPRT2	RDPRT1
r						r	r	r	r	r	r	r	r	r	r

Bit Field	Name	Description
31	OBERR	Option byte error When this bit is '1', it indicates that the option byte and its complement do not match. *Note: Read-only bit.*
30	nSWBOOT0	Refer to Section 2.1.3.3 for usage rules.
29	nBOOT1	Refer to Section 2.1.3.3 for usage rules.
28	nBOOT0	Refer to Section 2.1.3.3 for usage rules.
27:20	Data1[7:0]	Data1 *Note: This bit is read-only.*
19:12	Data0[7:0]	Data0 *Note: This bit is read-only.*
11:10	Reserved	Reserved, the reset value must be maintained.
9	Reserved	Not used (if the corresponding option byte bit is written, it can be read from this register, no effect on the chip)
8	IWDGSLEEPFRZ	IWDG freeze enable in SLEEP mode 0: IWDG frozen in SLEEP mode 1: IWDG not frozen in SLEEP mode *Note: Read-only bit*
7	IWDGSTDBYFRZ	IWDG freeze enable in STANDBY mode 0: IWDG frozen in STANDBY mode 1: IWDG not frozen in STANDBY mode *Note: Read-only bit*
6	Reserved	Not used (if the corresponding option byte bit is written, it can be read from this register, no effect on the chip)
5	IWDGSTOPFRZ	IWDG freeze enable in STOP mode 0: IWDG frozen in STOP mode 1: IWDG not frozen in STOP mode *Note: Read-only bit*
4	nRST_STDBY	Enter STANDBY mode reset configuration. 0: Reset immediately after entering STANDBY mode; 1: No reset occurs after entering STANDBY mode. *Note: This bit is read-only.*
3	nRST_STOP	Enter STOP mode reset configuration. 0: Reset occurs immediately after entering STOP mode;

Bit Field	Name	Description
		1: No reset occurs after entering the STOP mode. *Note: This bit is read-only.*
2	IWDG_SW	Set independent watchdog 0: Hardware watchdog; 1: Software watchdog. *Note: This bit is read-only.*
1	RDPRT2	Read protection L2 level protection 0: Read protection L2 level is disabled; 1: Read protection L2 level is enabled. *Note: This bit is read-only.*
0	RDPRT1	Read protection L1 level protection 0: Read protection L1 level is not enabled; 1: Read protection L1 level is enabled. *Note: This bit is read-only.*

2.2.6.2.8 Write protection register (FLASH_WRP)

Address offset: 0x1C

Reset value: 0xFFFF FFFF

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="16">WRPT[31:16]</td></tr>
<tr><td colspan="16">w</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="16">WRPT[15:0]</td></tr>
<tr><td colspan="16">w</td></tr>
</table>

Bit Field	Name	Description
31:0	WRPT	Write protect This register contains the write protection OptionByte loaded by OptionByte area. 0: Write protection is valid; 1: Write protection is invalid. *Note: These bits are read-only.*

2.2.6.2.9 ECC register（FLASH_ECC）

Address offset: 0x20

Reset value: 0x0000 0000

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="16">Reserved</td></tr>
</table>

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		ECCHW						Reserved		ECCLW					
		r								r					

Bit Field	Name	Description
31:16	Reserved	Reserved, the reset value must be maintained.
15:8	ECCHW	After writing a word to a 32-bit Flash address, the corresponding lower 6-bit ECC value.
7:0	ECCLW	After writing a word to a 32-bit Flash address, the corresponding lower 6-bit ECC value.

2.2.6.2.10 FLASH RDN register (FLASH_RDN)

Address offset: 0x2C

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved							FLASH_RDN1								
							r								

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved							FLASH_RDN0								
							r								

Bit Field	Name	Description
31:25	Reserved	Reserved, the reset value must be maintained.
24:16	FLASH_RDN1	Address of Flash redundant block page 1
15:9	Reserved	Reserved, the reset value must be maintained.
8:0	FLASH_RDN0	Address of Flash redundant block page 0

2.2.6.2.11 FLASH CAHR register (FLASH_CAHR)

Address offset: 0x30

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								LOCKSTOP[3:0]				LOCKSTRT[3:0]			
								rw				rw			

Bit Field	Name	Description
31:8	Reserved	Reserved, the reset value must be maintained.
7:4	LOCKSTOP[3:0]	ICache lock stop (see for detailed operation instructions 2.2.2.3.3 ICache locking Chapter). 0: Disable 1: Enable
3:0	LOCKSTRT[3:0]	ICache lock start. 0: Disable 1: Enable

2.2.6.2.12 FLASH ROW program register (FLASH_ROWP)

Address offset: 0x34

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved									RPAREA	RPADD					
									rw	rw					

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RPADD											RPNUM				RPEN
rw											rw				rw

Bit Field	Name	Description
31:23	Reserved	Reserved, the reset value must be maintained.
22	RPAREA	ROW programming area 1: system memory area 0: main flash area
21:5	RPADD	ROW programming address AHB bus [18:2]
4:1	RPNUM	Number of words programmed in a ROW: 0/1: 2-word programming 2/3: 4-word programming ... 14/15: 16-word programming *Note: Single-word programming is not supported*
0	RPEN	ROW programming enable bit: 1: ROW programming 0: Double-word programming

2.2.6.2.13 CCM write protection register 1 (CCM_WRP1)

Address offset: 0x38

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
WRPT[31:16]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WRPT[15:0]															
rw															

Bit Field	Name	Description
31:0	WRPTx	WRPTx (x = 0 to 31): CCM SRAM Page x Write Protection These bits are set by software and are cleared only on system reset. 0: Disable write protection for Page x of CCM SRAM. 1: Enable write protection for Page x of CCM SRAM.

2.2.6.2.14 CCM key register (CCM_KEY)

Address offset: 0x3C

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								KEY							
								w							

Bit Field	Name	Description
31:8	Reserved	Reserved, the reset value must be maintained.
7:0	KEY	Used for software to remove write protection from the CCM_ERASE.EN bit: To remove the write protection from the CCM_ERASE.EN bit, the following steps need to be taken: 1. Write '0xCA' to the KEY 2. Write '0x53' to the KEY Writing an incorrect KEY will reactivate the write protection.

2.2.6.2.15 CCM erase register (CCM_ERASE)

Address offset: 0x40

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

Reserved

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved														BUSY	EN
														r	rw

Bit Field	Name	Description
31:2	Reserved	Reserved, the reset value must be maintained.
1	BUSY	CCM SRAM busy with erase operation 0: No CCM SRAM erase operation. 1: CCM SRAM is currently performing an erase operation.
0	EN	CCM SRAM erase Setting this bit will initiate the hardware CCM SRAM erase operation. After the CCM SRAM erase operation is complete, this bit will be automatically cleared. *Note: This bit is write-protected: It can only be set after writing the correct KEY in sequence in CCM_KEY.KEY.*

2.2.6.2.16 CCM mode register (CCM_MODE)

Address offset: 0x48

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															EN
															rw

Bit Field	Name	Description
31:1	Reserved	Reserved, the reset value must be maintained.
0	EN	Enable CCM SRAM mode. This register can only be configured when the UID matches the UID configured last time. If another UID wants to configure this value, the value of this register needs to be 0. 0: No CCM mode, CCM used as regular SRAM 1: CCM mode, CCM used as CCM SRAM

2.2.6.2.17 XSPI decryption start address register (XSPI_DSTRADD)

Address offset: 0x4C

Reset value: 0x0000 0000

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="16">ADD[31:16]</td></tr>
<tr><td colspan="16">rw</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="16">ADD[15:0]</td></tr>
<tr><td colspan="16">rw</td></tr>
</table>

Bit Field	Name	Description
31:0	ADD	XSPI decryption start address *Note: If XSPI_FEMC_DEN.EN = 0, then this bit is invalid.*

2.2.6.2.18 XSPI decryption end address register (XSPI_DENDADD)

Address offset: 0x50

Reset value: 0x0000 0000

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="16">ADD[31:16]</td></tr>
<tr><td colspan="16">rw</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="16">ADD[15:0]</td></tr>
<tr><td colspan="16">rw</td></tr>
</table>

Bit Field	Name	Description
31:0	ADD	XSPI decryption end address *Note: If XSPI_FEMC_DEN.EN = 0, then this bit is invalid.*

2.2.6.2.19 FEMC decryption start address register (FEMC_DSTRADD)

Address offset: 0x54

Reset value: 0x0000 0000

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="16">ADD[31:16]</td></tr>
<tr><td colspan="16">rw</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="16">ADD[15:0]</td></tr>
<tr><td colspan="16">rw</td></tr>
</table>

Bit Field	Name	Description
31:0	ADD	FEMC decryption start address *Note: If XSPI_FEMC_DEN.EN = 0, then this bit is invalid.*

2.2.6.2.20 FEMC decryption end address register (FEMC_DENDADD)

Address offset: 0x58

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADD[31:16]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADD[15:0]															
rw															

Bit Field	Name	Description
31:0	ADD	FEMC decryption end address *Note: If XSPI_FEMC_DEN.EN = 0, then this bit is invalid.*

2.2.6.2.21 RTPD key register (RTPD_KEY)

Address offset: 0x5C

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEY[31:16]															
w															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY[15:0]															
w															

Bit Field	Name	Description
31:0	KEY	The KEY is 128 bits in total, and needs to be written to this register in four separate writes. *Note: Configuration needs to be done under the same UID.*

2.2.6.2.22 JTAG seal register (JTAG_SEAL)

Address offset: 0x60

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								DATA							
								rw							

Bit Field	Name	Description
31:8	Reserved	Reserved, the reset value must be maintained.
7:0	DATA	DATA = 0x5A, JTAG SEAL enabled DATA ≠ 0x5A, JTAG SEAL disabled *Note: JTAG and SRAM users cannot configure this register*

2.2.6.2.23 RTPD_KEY write count register (RTPD_KEY_WCNT)

Address offset: 0x64

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													CNT		
													r		

Bit Field	Name	Description
31:3	Reserved	Reserved, the reset value must be maintained.
2:0	CNT	RTPD_KEY write counter. When writing to the RTPD_KEY register, the counter will increment by 1. When the counter reaches 3'd7, continue writing to these two registers, the counter will reset to 1, and increment by 1 after the next write.

2.2.6.2.24 XSPI_FEMC decryption enable register (XSPI_FEMC_DEN)

Address offset: 0x6C

Reset value: 0x0000 00000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															EN
															rw

Bit Field	Name	Description
31:1	Reserved	Reserved, the reset value must be maintained.
0	EN	Enable XSPI and FEMC decryption. It must be configured after the (XSPI and FEMC) start address, end address, and RTPD_KEY. If EN is 1, the (XSPI and FEMC) start address, end address, and RTPD_KEY cannot be configured. *Note: Setting from 1 to 0 will clear the (XSPI and FEMC) start address, end address, and RTPD_KEY.*

2.2.6.2.25 XUID register (XUID)

Address offset: 0x70

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						CCMUID					XFUID				
						r					r				

Bit Field	Name	Description
31:10	Reserved	Reserved, the reset value must be maintained.
9:5	CCMUID	Configure the user UID of CCM Bit0: system memory Bit1: user1 Bit2: user2 Bit3: user3 Bit4: SRAM
4:0	XFUID	Configure the user UID for the (XSPI and FEMC) start address, end address, and RTPD_KEY Bit0: system memory Bit1: user1 Bit2: user2 Bit3: user3 Bit4: SRAM *Note: When the user configures xspi_femc_dec_en from 1 to 0, XFUID will be set*

Bit Field	Name	Description
		to 5'h0.

3 Power Control (PWR)

3.1 General Description

The PWR is power management unit to control status of different modules in different power modes. Its major function is to control MCU to enter different power modes and wakeup when events or interrupts happen. The MCU supports the following modes: RUN、SLEEP、STOP0、STANDBY and VBAT .

3.1.1 Power Supply

The operating voltage of the MCU is 1.8V~3.6V, and the power control system is described as follows. For details, please refer to Figure 3-1 power supply block diagram.

- 4 external power supplies: VDD, VDDA, VREF, VBAT. Among them, VDD is the chip power supply, mainly providing power to the power supply system and clock system; VDDA is the analog peripheral power supply, mainly providing power to the analog peripherals; VREF provides a reference power supply to the analog peripherals to provide higher accuracy. VBAT is connected to the battery, providing power to the backup domain.
- 5 power domains, powered by external power supplies:
  - V_{DD} domain: The voltage input range is 1.8V~3.6V, mainly for MR, most GPIOs, HSE, HSI, PLL, POR/PDR, BOR, PVD, USB PHY.
  - V_{DDA} domain: The voltage input range is 1.8V~3.6V, which mainly supplies power for ADC, DAC, COMP, PGA, VREFBUF, TS, etc.
  - V_{DDBK} domain: The voltage input range is 1.8V~3.6V, which mainly supplies power for WKUP , NRST、PC13/14/15、LSE、LSI, etc.
  - V_{DDD} domain: The voltage input range is 1.1V or 0.9V, which mainly supplies power for CPU、AHB、APB、SRAM、FLASH、RCC、TRNG and most most peripherals.
  - V_{DDDBK} domain: The voltage input range is 0.9V or 0.8V, which mainly supplies power for PWR, backup SRAM (4KB), RTC, LPTIM, WKUP pins, NRST, PC13/14/15, backup IOM, IWDG, and RCC_BDCTRL register.

When VDD is turned off, the VDDBK domain is powered by VBAT selected by SW1; when VDD is turned on, the VDDBK domain is powered by VDD selected by SW1. If an external battery is not needed in the application, it is recommended to connect the VBAT pin to VDD, and also add a 100nF ceramic capacitor.

- 2 voltage regulators, providing low-voltage power supply for digital modules:
  - MR

    For the internal main power domain controller, mainly used in RUN mode, SLEEP mode, and STOP0 mode. The MR has two modes, normal mode and low power mode, with the low power mode used in STOP0 to further reduce power consumption.

    When the MR enters low power mode, the CPU will enter deep sleep mode. At this time, the PWR_CTRL.PDS bit should be set to 0, and the PWR_CTRL.LPS bit should be set to 1. When the MR enters normal mode, the PWR_CTRL.PDS bit should be set to 0, and the PWR_CTRL.LPS bit should also be set to 0.

- BKR

  For the internal backup power domain controller, mainly used in STANDBY and VBAT modes, configure the BKR output voltage to 0.9V or 0.8V through PWR_CTRL3.BKRSEL.

The VDDDBK domain remains powered in all modes. In RUN, SLEEP, and STOP0 modes, the VDDDBK domain selects the MR output as the supply voltage through SW2. In STANDBY and VBAT modes, the VDDDBK domain selects the BKR output as the supply voltage through SW2.

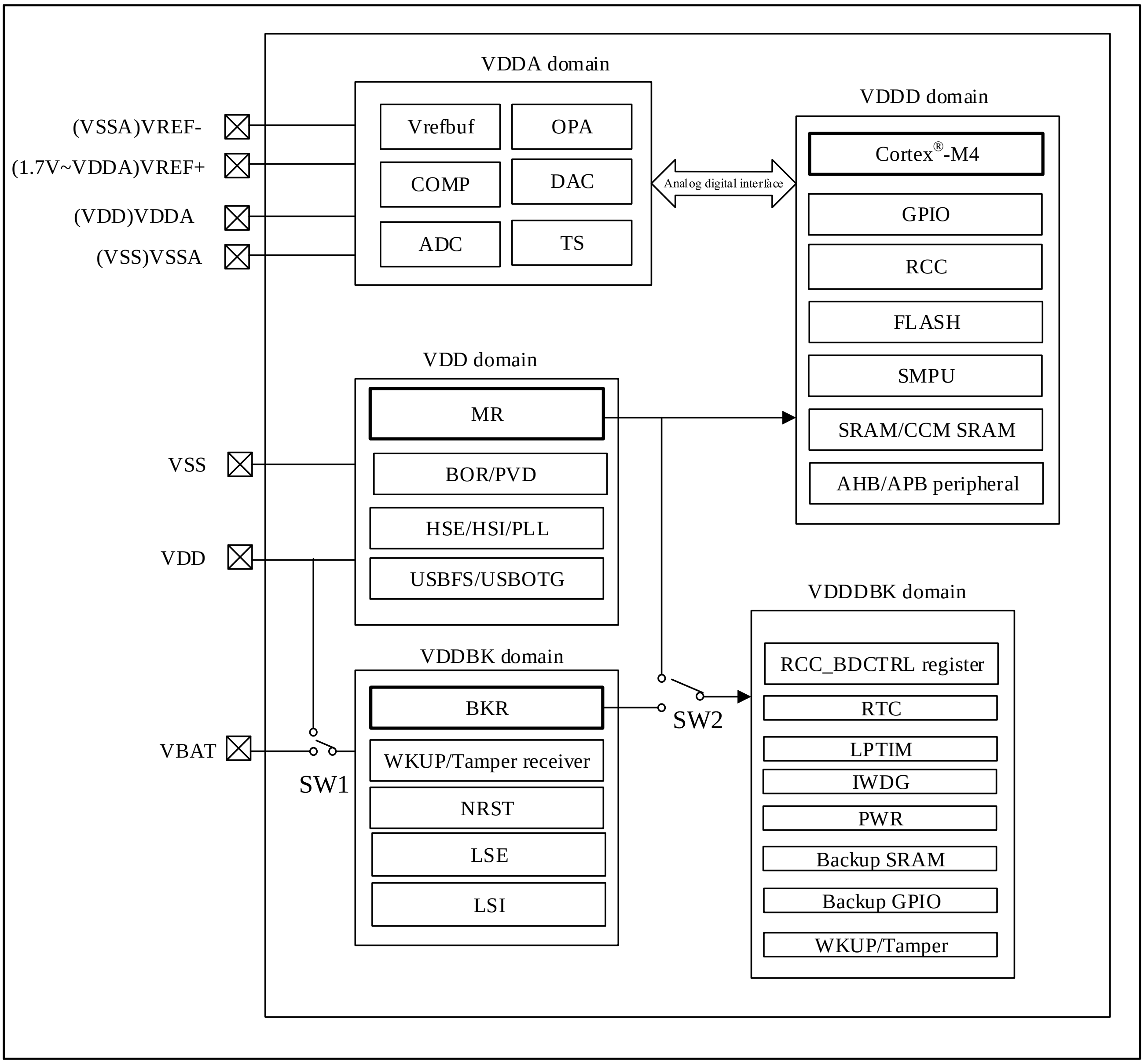
Notes:

1. During the VDD rising phase or when PDR is detected, the switch between VBAT and VDD remains connected to the VBAT domain.

2. During the startup phase, if VDD rises quickly and VDD > VBAT, current can be injected into VBAT through an internal diode connection. If the power supply or battery connected to the VBAT pin cannot withstand this injected current, it is strongly recommended to add a low-voltage diode between the power supply and the VBAT pin.

3. In low-voltage (1.8V) applications to achieve lower power consumption, it is recommended to turn on the peripheral modules in batches (configure the corresponding bits of RCC AHBPCLKEN registers) with each batch of modules not exceeding 4, and a 1ms clock interval between every two batches of modules being turned on.

Figure 3-1 Power Supply Block Diagram



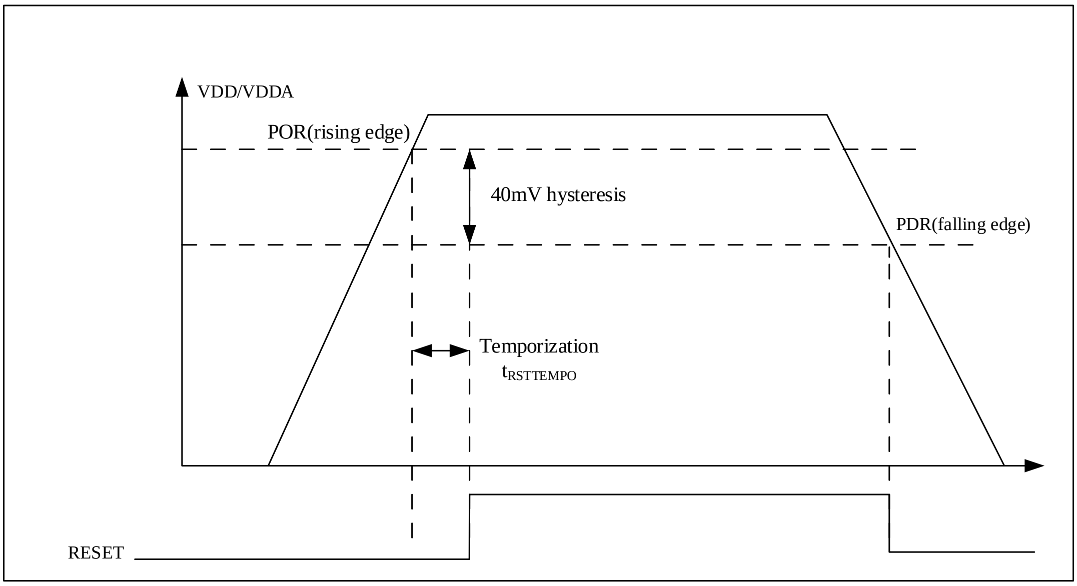
3.1.2 Power Supply Supervisor

3.1.2.1 Power on reset (POR) and power down reset (PDR)

The power-on reset (POR) and power-down reset (PDR) circuits are integrated internally in the chip. It can operate at a minimum voltage of 1.8V. There is no need for an external reset circuit; the chip will remain in a reset state when VDD or VDDA falls below the specified threshold (VPOR/PDR).

For detailed information on the switch power reset threshold, please refer to the Electrical Characteristics section of the relevant datasheet.

Figure 3-2 Power On Reset and Power Down Reset Waveform



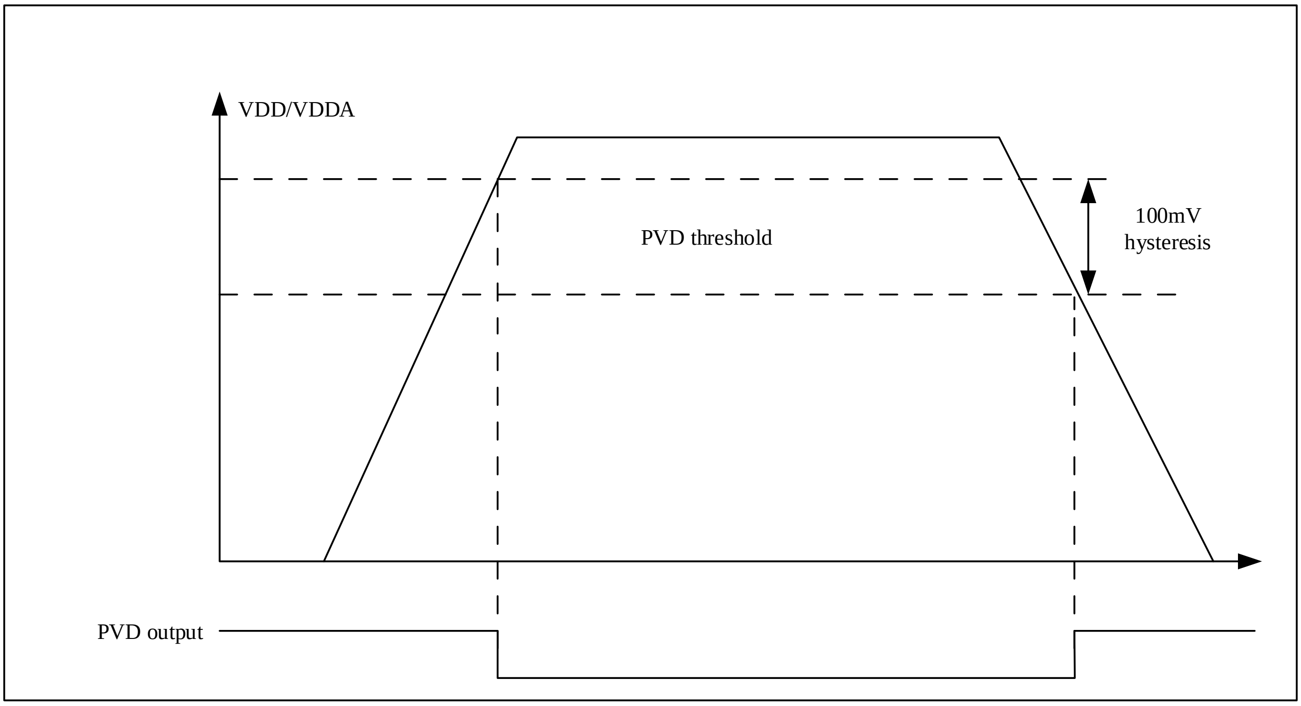
3.1.2.2 Programmable voltage detector (PVD)

The PVD can be used to monitor the VDD/VDDA power supply by comparing it with the threshold set by the PWR_CTRL.PRS[2:0] bits in the power control register. Enable the PVD by setting PWR_CTRL.PVDEN.

The PWR_CTRLSTS.PVDO flag is used to indicate whether VDD/VDDA is above/below the PVD voltage threshold. This event is internally connected to external interrupt line 16, and if the interrupt is enabled in the external interrupt register, an interrupt will be generated. Depending on the rising/falling edge trigger setting of external interrupt line 16, a PVD interrupt will occur when VDD/VDDA falls below the PVD threshold or rises above it. For example, this feature can be used to perform emergency shutdown tasks.

Note: The PVD threshold needs to be used in conjunction with the PWR_CTRL.MSB register, as described in the PWR_CTRL.PRS[2:0] description.

Figure 3-3 PVD Threshold Waveform



3.1.2.3 Brown out reset（BOR）

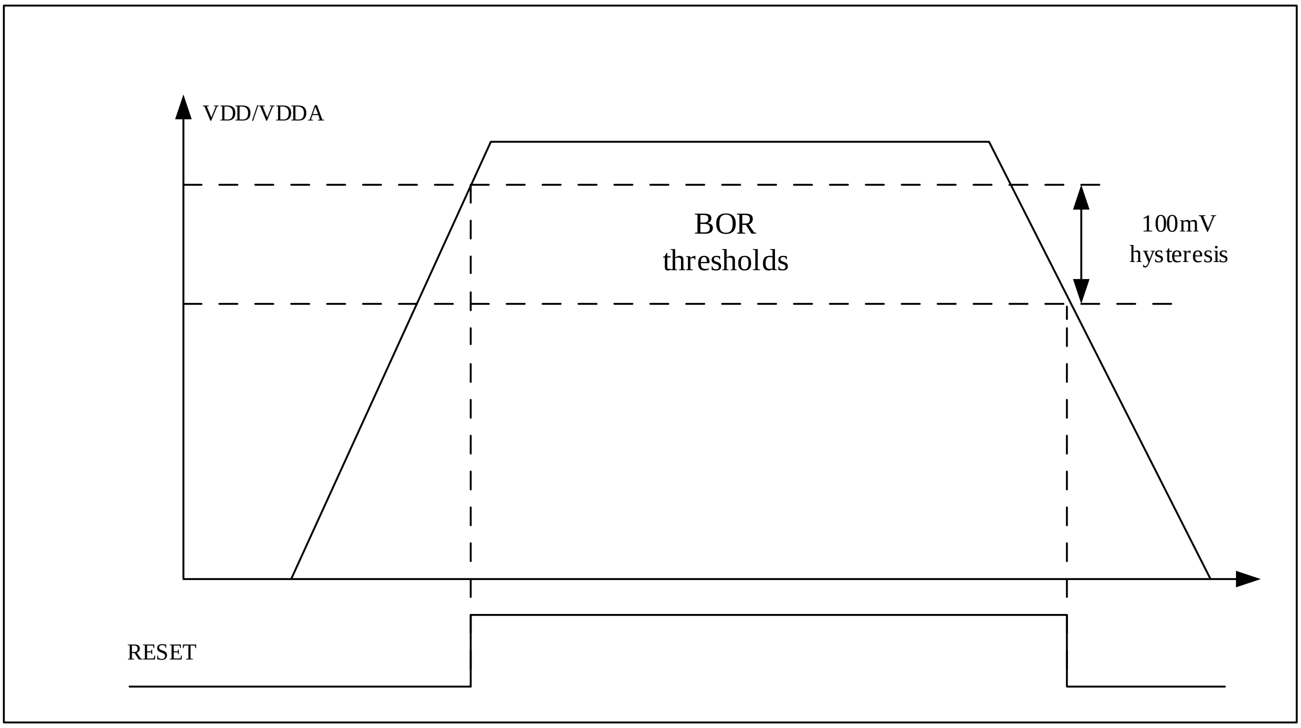
During power-up, the Brown-Out Reset (BOR) will keep the device in a reset state until the power voltage reaches the specified VBOR threshold.

V_{BOR} is configured through option bytes and offers a choice of 5 V_{BOR} thresholds.

- BOR level 0 - reset level threshold 1.62 V (release level threshold 1.66V)
- BOR level 1 - reset level threshold 2.0 V (release level threshold 2.1 V)
- BOR level 2 - reset level threshold 2.2 V (release level threshold 2.3 V)
- BOR level 3 - reset level threshold 2.5 V (release level threshold 2.6 V)
- BOR level 4 - reset level threshold 2.8 V (release level threshold 2.9 V)

When the power supply voltage (VDD) drops below the selected V_{BOR} threshold, the device will be reset.

Figure 3-4 BOR Threshold Waveform



3.2 Power Modes

The MCU has 5 power modes: RUN、SLEEP、STOP0、STANDBY and VBAT. Different mode has different performance and power consumption. A summary of MCU power modes is shown below.

Table 3-1 Power Modes

Mode	Conditions	Enter	Exit
RUN	CPU startup peripheral configuration	Power on, system reset, low power wake-up	Enter SLEEP, STOP0, STANDBY, and VBAT modes
SLEEP	The CPU enters sleep mode, with the core stopped. All peripheral configurations remain active, and the voltage regulators are still running. Any interrupt or event can wake up the CPU.	Immediate sleep mode: 1) SCB_SCR.SLEEPDEEP = 0, SCB_SCR.SLEEPONEXIT = 0 2) WFI/WFE Sleep mode on exit: 1) SCB_SCR.SLEEPDEEP = 0, SCB_SCR.SLEEPONEXIT = 1 2) WFI has no interrupt pending, CPU returns from ISR	If entered through WFI, any interrupt from the NVIC table can wake up the device; if entered through WFE, with SCB_SCR.SEVONPEND=1, any peripheral interrupt (without enabling NVIC) can wake up the device, and with SCB_SCR.SEVONPEND=0, any EXTI line wake-up event.
STOP0[1]	The CPU enters deep sleep mode. Peripheral clocks, all digital modules, and voltage regulators are still running.	FI/WFE： 1) SCB_SCR.SLEEPDEEP = 1，PWR_CTRL.PDS＝0，	If entered through WFI, any interrupt from the EXTI line (requires NVIC enable) can wake

Mode	Conditions	Enter	Exit
	HSE/HSI/PLL are turned off. LSE/LSI can be configured. All SRAM retains data. Upon wake-up, HSI is enabled, and the code resumes from the suspended point.	2) PWR_CTRL.LPS=0 or 1 selects the main voltage regulator operating mode	up; If entered through WFE, any EXTI line wake-up event
STANDBY	The CPU enters deep sleep mode. The main voltage regulator is turned off, HSE/HSI/PLL are turned off. LSE/LSI can be configured. The backup domain is functional, the main domain is off, backup SRAM retention is configured through PWR_CTRL2.BSRSTBRET. Data in other SRAM and registers is lost. All IO port states can be configured to remain output.	WFI/WFE: 1) SCB_SCR.SLEEPDEEP = 1 2) PWR_CTRL.PDS = 1	WKUP pins (PA0, PE6, PC13, PA2, PC5) high or low level, RTC alarm event, RTC wake-up event, RTC timestamp event, RTC tamper event, LPTIM1/2 wake-up event, NRST reset, IWDG reset
VBAT	CPU turned off. All peripherals off, main voltage regulator off. LSE/LSI configurable, HSE/HSI/PLL off. Backup SRAM retention configured through PWR_CTRL2.BSRVBRET. Except for NRST/PC13-TAMPER/ PC14-OSC32_IN/PC15-OSC32_OUT/ WKUP(PA0, PE6, PC13, PA2, PC5), most IO ports are in high-impedance state.	VDD off	VDD on

Note:

1. *In STOP2 mode, after wake-up, the code can continue running from the stop position.*

The operating enabled status of different modules in different power consumption modes are shown in the following table:

Table 3-2 Modules Running State[(1)]

Peripheral	Run	Sleep	Stop0		Standby		VBAT
			-	Wakeup capability	-	Wakeup capability	
CPU	Y	-	-	-	-	-	-
Flash memory	O[(2)]	O[(2)]	-	-	-	-	-

SRAM (192 KB) (include CCM SRAM)	Y	Y	Y	-	-	-	-
Backup SRAM (4 KB)	Y	Y	Y	-	O(3)	-	O(3)
MR	Y	Y	Y	-	-	-	-
POR(VDDD)	Y	Y	Y	Y	-	-	-
POR(VDDA)	Y	Y	Y	Y	Y	Y	-
POR(VDDBK)	Y	Y	Y	Y	Y	Y	Y
BOR	Y	Y	Y	Y	-	-	-
PVD	O	O	O	O	-	-	-
HSI	O	O	-(4)	-	-	-	-
HSE	O	O	-	-	-	-	-
LSI	O	O	O	-	O	-	O
LSE	O	O	O	-	O	-	O
HSE CSS	O	O	-	-	-	-	-
LSE CSS	O	O	O	Y	O	Y	Y
OSC300M(UCDR)	O	O	-	-	-	-	-
PLL	O	O	-	-	-	-	-
SHRTPLL	O	O	-	-	-	-	-
RTC / Auto wakeup	O	O	O	O	O	O	O
Backup Registers	Y	Y	Y	-	Y	-	Y
Number of RTC Tamper pins	3	3	3	O	3	O	3
ATIMx (x=1,2,3)	O	O	-	-	-	-	-
GTIMx (x=1~10)	O	O	-	-	-	-	-
BTIMx (x=1,2)	O	O	-	-	-	-	-
SHRTIMx (x=1)	O	O	-	-	-	-	-
LPTIMx(x=1,2)	O	O	O	O	O	O	O
IWDG	O	O	O	O	O	O	O
WWDG	O	O	-	-	-	-	-
SysTick timer	O	O	-	-	-	-	-
U(S)ARTx(x=1~8)	O	O	-	-	-	-	-
I2Cx (x=1,2,3,4)	O	O	-	-	-	-	-
SPIx (x=1~6)	O	O	-	-	-	-	-
I2Sx (x=2,3)	O	O	-	-	-	-	-
FDCANx(x=1,2,3)	O	O	-	-	-	-	-
USB HS Dualrole	O	O	O(6)	O	-	-	-
USB FS DEVICE	O	O	O(6)	O	-	-	-
xSPI	O	O	-	-	-	-	-
SD&MMC	O	O	-	-	-	-	-
FEMC	O	O	-	-	-	-	-
DVP	O	O	-	-	-	-	-
ETH MAC	O	O	O(7)	O	-	-	-
VREFBUF	O	O	O	O	-	-	-

ADCx (x=1,2,3,4)	O	O	-	-	-	-	-
DACx (x=1~8)	O	O	-	-	-	-	-
PGAx(x=1,2,3,4)	O	O	-	-	-	-	-
COMPx(x=1~7)	O	O	O	O	-	-	-
Temperature sensor	O	O	-	-	-	-	-
SAC	O	O	-	-	-	-	-
RNG	O	O	-	-	-	-	-
CRC	O	O	-	-	-	-	-
Cordic	O	O	-	-	-	-	-
FMAC	O	O	-	-	-	-	-
GPIOs	O	O	O	O	5pin(5)	5 pin(5)	-

Notes:

1. *Y: Yes (Enable), O: Option, -: Not available.*
2. *Flash can be configured in power-down mode, it is not in power-down mode by default.*
3. *The content of Backup SRAM can be retained by controlling the relevant bits of the power control register.*
4. *Some peripherals that can wake up from STOP mode may require HSI to be enabled. In this case, HSI is enabled by the peripheral and is only supplied to that peripheral. When the peripheral is no longer needed, HSI is automatically turned off.*
5. *The pins with the capability to wake up from STANDBY are PE6 (WKUP0), PA0 (WKUP1), PC13 (WKUP2), PA2 (WKUP3), PC5 (WKUP4), and NRST.*
6. *The USB module only works with wake-up-related logic, waking up through EXTI.*
7. *The ETH MAC module only works with wake-up-related logic, waking up through EXTI.*

3.2.1 SLEEP Mode

The CPU is halted, all peripherals, including peripherals around the Cortex™-M4F core (such as NVIC, SysTick, etc.), can operate and wake up the CPU when an interrupt or event occurs. In SLEEP mode, all I/O pins maintain the same state/function as in the run mode.

3.2.1.1 Entering SLEEP mode

SLEEP mode can be entered by executing the WFI (Wait For Interrupt) or WFE (Wait For Event) instruction and setting SCB_SCR.SLEEPDEEP = 0. Depending on SCB_SCR.SLEEPONEXIT, there are two ways to enter SLEEP mode:

- SLEEP-NOW: If SCB_SCR.SLEEPONEXIT = 0, executing the WFI or WFE instruction immediately puts the system into SLEEP mode.
- SLEEP-ON-EXIT: If SCB_SCR.SLEEPONEXIT = 1, the system enters SLEEP mode immediately upon exiting the lowest priority ISR.

In SLEEP mode, all I/O pins maintain the same state/function as in the run mode.

3.2.1.2 Exiting SLEEP mode

If entering SLEEP mode using the WFI instruction, any NVIC interrupt can wake up the device from SLEEP mode.

If entering SLEEP mode using the WFE instruction, the MCU will immediately exit the SLEEP mode when an event occurs. Wake-up events can be generated in the following ways:

- Enable interrupts in the peripheral control register instead of NVIC, and enable SCB_SCR.SEVONPEND. When the MCU is woken up by WFE, the peripheral interrupt pending bit and the peripheral NVIC interrupt channel pending bit (in the NVIC interrupt clear-pending register) must be cleared.
- Configure external or internal EXTI event mode. When the MCU wakes up, there is no need to clear the peripheral interrupt pending bit and the peripheral NVIC interrupt channel pending bit (in the NVIC interrupt clear-pending register) because there are no pending bits set for the event line. This mode provides the shortest wake-up time, as there is no time spent on interrupt entry or exit.

3.2.2 STOP0 Mode

STOP0 mode is based on the Cortex^{TM}-M4F deep sleep mode and incorporates peripheral clock control mechanisms. The voltage regulator can be configured in normal or low-power mode. In STOP0 mode, most clock sources in the core domain are disabled, such as PLL, HSI, and HSE. However, the contents of SRAM, CCM SRAM, backup SRAM, and all registers are preserved.

In STOP0 mode, all I/O pins maintain the same state as in the run mode.

3.2.2.1 Entering STOP0 mode

When entering STOP0 mode, set SCB_SCR.SLEEPDEEP=1 and PWR_CTRL.PDS=0. Additionally, the MR can operate in normal mode or low-power mode. When configured with PWR_CTRL.LPS = 1, the MR operates in low-power mode; when PWR_CTRL.LPS = 0, the MR operates in normal mode.

In STOP0 mode, all I/O pins maintain the same state and functionality as in the run mode.

If a FLASH operation is in progress, the entry into STOP0 mode will be delayed until the memory access is complete.

If an access to the APB region is ongoing, the entry into STOP0 mode will be delayed until the APB access is complete.

In STOP0 mode, the following features can be selected by programming various control bits:

- Independent Watchdog (IWDG): The independent watchdog will be started when its related registers are written to by software or through hardware operation. Once started, the independent watchdog will continue to operate until it generates a reset.
- RTC: Enabled by setting the RCC_BDCTRL.RTCEN bit in the register.
- Internal RC oscillator (LSI RC): Enabled by setting the RCC_CTRLSTS.LSIEN bit in the register.
- External 32.768kHz crystal oscillator (LSE OSC): Enabled by setting the RCC_BDCTRL.LSEEN bit in the register.

ADC or DAC can also consume power in STOP0 mode, and can be disabled before entering STOP0 mode.

Note: If the application needs to disable external clocks before entering STOP mode, the RCC_CTRL.HSEEN bit must be disabled first, and then the system clock switched to HSI. Otherwise, if the RCC_CTRL.HSEEN bit remains enabled when entering STOP mode, and the external clock (external oscillator) is removed, the Clock Security System (HSECSS) feature must be enabled to detect any external oscillator failures and avoid faulty behavior when entering STOP mode.

3.2.2.2 Exiting STOP0 mode

When exiting STOP0 mode via interrupt or wake-up event, the HSI RC oscillator is automatically selected as the system clock. Additional startup delay is generated when waking up from STOP0 mode while the voltage regulator is operating in low-power mode. In STOP0 mode, having the internal regulator in normal mode reduces the start-up time but increases power consumption correspondingly.

3.2.3 STANDBY Mode

STANDBY mode is based on the Cortex^{TM}-M4F Deep-Sleep mode, achieving lower power consumption. Powered by the BKR output, the backup domain is enabled.

3.2.3.1 Entering STANDBY mode

When entering STANDBY mode. The main difference is setting SCB_SCR.SLEEPDEEP=1, PWR_CTRL.PDS=1.

In STANDBY mode, the state of all IO pins can be configured to remain output mode.

If a FLASH operation is in progress, the entry into STANDBY mode will be delayed until the memory access is complete.

If an access to the APB region is ongoing, the entry into STANDBY mode will be delayed until the APB access is complete.

In STANDBY mode, the following features can be selected by programming various control bits:

- Independent Watchdog (IWDG) optional: The independent watchdog will be started when its related registers are written to by software or through hardware operation. Once started, the independent watchdog will continue to operate until it generates a reset.
- RTC optional: Can be enabled by setting RCC_BDCTRL.RTCEN.
- Internal RC oscillator (LSI RC): Always on, and can perform automatic recovery once stopped oscillating.
- External 32.768kHz crystal oscillator (LSE OSC) optional: Can be enabled by setting RCC_BDCTRL.LSEEN.
- Backup SRAM data retention can be enabled by setting the register PWR_CTRL2.BSRSTBRET bit.

3.2.3.2 Exiting STANDBY mode

When an external reset (NRST pin), IWDG reset, edge of WKUP pin, RTC alarm, wakeup, tamper, or timestamp event occurs, the MCU exits STANDBY mode. All registers, except for the PWR_CTRLSTS register and RTC backup registers, are reset upon waking up from STANDBY mode.

After waking up from STANDBY mode, the code execution is the same as after a reset (checking the BOOT pin, obtaining reset vectors, etc.). The PWR_CTRLSTS.SBF status flag indicates that the MCU has exited STANDBY mode, and it needs to be cleared by writing a 1 to the PWR_CTRL.CSBVBF bit.

3.2.4 VBAT Mode

In VBAT mode, the CPU is turned off, all peripherals are turned off, the main voltage regulator is turned off, LSE/LSI can be configured, and HSE/HSI/PLL are turned off. Except for NRST, WKUP (PA0, PE6, PC13, PA2, PC5), PC13_TAMPER, PC14, PC15, most IO ports are in a high-impedance state.

In VBAT mode, based on the configuration before VDD power-off, the following features can be used:

- RTC optional: Can be enabled by setting RCC_BDCTRL.RTCEN.

- Internal RC oscillator (LSI RC): Can be enabled by setting the register RCC_CTRLSTS.LSIEN bit.
- External 32.768kHz crystal oscillator (LSE OSC) optional: Can be enabled by setting RCC_BDCTRL.LSEEN.
- Backup SRAM data retention can be enabled by setting the register PWR_CTRL2.BSRSTBRET bit.

3.2.4.1 Entering VBAT mode

When VDD power is lost, the device will enter VBAT mode at any time.

3.2.4.2 Exiting VBAT mode

When VDD returns to the power-on reset threshold, the MCU exits VBAT mode. After VDD recovery, the MCU's main power domain will fully execute according to the power-on sequence. Upon waking up from VBAT mode, the code execution is equivalent to that after a reset. The PWR_CTRLSTS.VBATF status flag indicates that the MCU has exited VBAT mode, and it needs to be cleared by writing a 1 to the PWR_CTRL.CSBVBF bit.

3.3 Low-Power Auto-Wakeup (AWU) Mode

In automatic wake-up mode, the RTC can be used to wake up the MCU from different low-power modes without relying on external interrupts. The RTC provides a programmable time base for waking up from STOP0 and STANDBY modes at regular intervals. For it, two of the three optional RTC clock sources can be selected by software programming RCC_BDCTRL.RTCSEL[1:0] as follows:

- 32.768 kHz external crystal oscillator (LSE OSC)

  This clock source provides an accurate time base with very low power consumption.
- Internal RC crystal oscillator (LSI RC)

  This clock source has the advantage of saving the cost of the 32.768 kHz crystal, but the clock accuracy is worse than the LSE.

To wake up from STOP0 mode using the RTC alarm event, you need:

- Configure EXTI 17 rising edge trigger.
- Configure RTC to enable RTC alarm event.

To wake up from STANDBY mode using RTC alarm event, EXTI 17 does not need to be configured.

VBAT mode cannot be woken up by the RTC.

3.4 Low-Speed Clock Output (LCO)

The chip supports outputting the low-speed clock through the PA2 pin in STOP0 or STANDBY mode, selecting the output clock as LSI or LSE via PWR_CTRL6.LCOSEL, and enabling the low-speed clock output function with PWR_CTRL6.LCOEN.

3.5 PWR Registers

3.5.1 PWR Register Overview

Table 3-3 PWR Register Overview

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
00h	PWR_CTRL	Reserved															CWKUP5F	CWKUP4F	CWKUP3F	CWKUPPF	CWKUP2F	CWKUP1F	CWKUP0F	MSB	DBKP	PLS[2:0]			PVDEN	CSBVBF	Reserved	PDS	LPS
	Reset Value																0	0	0	0	0	0	0	0	0	0	0	0	0	0		0	0
04h	PWR_CTRLSTS	Reserved								WKUP5EN	WKUP5F	WKUP4F	WKUP3F	WKUP4POL	WKUP3POL	WKUP4EN	WKUP3EN	VBATF	WKUP2POL	WKUP1POL	WKUP0POL	WKUPP1EN	WKUP2EN	WKUP1EN	WKUP0EN	WKUPP0EN	WKUPPF	WKUP2F	WKUP1F	WKUP0F	PVDO	SBF	Reserved
	Reset Value									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
08h	PWR_CTRL2	Reserved																					IWDGRSTEN	Reserved							BSRSTBRET	BSRVBRET	Reserved
	Reset Value																						1								1	0	
0Ch	PWR_CTRL3	Reserved																	BKRSEL	Reserved													
	Reset Value																		1														
10h	PWR_CTRL4	Reserved																					LSENIMEN	Reserved									
	Reset Value																						0										
28h	PWR_CTRL5	LSECSSCRYIEN	Reserved	LSECSSCRYF	Reserved																												
	Reset Value	0		0																													
2Ch	PWR_CTRL6	LSECSSSW	LPWKUPSEL	NRSTSEL		LSEFHF	LSEFLF	LSECSSF	Reserved																								
	Reset Value	0	0	0	0	0	0	0																									
C0h	PWR_EMCCTRL	Reserved																BSRECCF[1:0]		BSRECCRSTEN[1:0]		BSRECCIEN[1:0]		EMCRST	BKPGBRST	BKPGBNRST	BKPCLPRST	BKPGBDET	BKPGBNDET	BKPCLPDET	Reserved		
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0			
C4h	PWR_EMCCTRL2	Reserved																BKPGBF	BKPGBNF	BKPCLPF	GB3F	GB2F	GB1F	GB0F	GBN3F	GBN2F	GBN1F	GBN0F	CLP3F	CLP2F	CLP1F	CLP0F	EMCFCLR
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

3.5.2 Power Control Register (PWR_CTRL)

Address offset: 0x00

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															CWKUP5F
															w

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CWKUP4F	CWKUP3F	CWKUPPF	CWKUP2F	CWKUP1F	CWKUP0F	MSB	DBKP	PLS[2:0]			PVDEN	CSBVBF	Reserved	PDS	LPS
w	w	w	w	w	w	rw	rw	rw			rw	w		rw	rw

Bit Field	Name	Description
31:17	Reserved	Reserved, the reset value must be maintained.
16	CWKUP5F	Clear the PH6 wakeup bit. Always read as 0. 0: Invalid 1: Clear the PWR_CTRLSTS.WKUP5F wakeup bit after 2 system clock cycles.
15	CWKUP4F	Clear the PC5 wakeup bit. Always read as 0. 0: Invalid 1: Clear the PWR_CTRLSTS.WKUP4F wakeup bit after 2 system clock cycles.
14	CWKUP3F	Clear the PA2 wakeup bit. Always read as 0. 0: Invalid 1: Clear the PWR_CTRLSTS.WKUP3F wakeup bit after 2 system clock cycles.
13	CWKUPPF	Clear the peripheral wakeup bit. Always read as 0. 0: Invalid 1: Clear the PWR_CTRLSTS.WKUPPF wakeup bit after 2 system clock cycles.
12	CWKUP2F	Clear the PC13 wakeup bit. Always read as 0. 0: Invalid 1: Clear the PWR_CTRLSTS.WKUP2F wakeup bit after 2 system clock cycles.
11	CWKUP1F	Clear the PA0 wakeup bit. Always read as 0. 0: Invalid 1: Clear the PWR_CTRLSTS.WKUP1F wakeup bit after 2 system clock cycles.
10	CWKUP0F	Clear the PE6 wakeup bit. Always read as 0. 0: Invalid 1: Clear the PWR_CTRLSTS.WKUP0F wakeup bit after 2 system clock cycles.
9	MSB	PVD monitoring voltage threshold level selection. When the MSB bit is enabled or cleared, the PVD detection voltage threshold levels

<table>
<tr><th>Bit Field</th><th>Name</th><th>Description</th></tr>
<tr><td></td><td></td><td>corresponding to PWR_CTRL.PRS[2:0] are different. Refer to the specific levels in the description of the PWR_CTRL.PRS[2:0] register.
0: PVD monitoring voltage threshold range 2.18V~2.88V
1: PVD monitoring voltage threshold range 1.78V~3.60V
<i>Note: This bit can only be operated when PWR_CTRL3.EXMODE = 1.</i></td></tr>
<tr><td>8</td><td>DBKP</td><td>Disable write protection for the backup domain.
In the reset state, the RTC and backup domain registers are protected to prevent unauthorized writes. This bit must be set to enable write access to these registers.
0: Disable access to RTC and backup registers
1: Enable access to RTC and backup registers
<i>Note: If the RTC clock is HSE/128, this bit must remain set to 1.</i></td></tr>
<tr><td>7:5</td><td>PLS[2:0]</td><td>PVD voltage threshold selection.
Different combinations of the PWR_CTRL.MSB bits represent different voltage threshold values for the voltage detector.
When the MSB bit is 0, the thresholds are as follows:
<table>
<tr><td>PRS[2:0]</td><td>Voltage</td></tr>
<tr><td>000</td><td>2.18v</td></tr>
<tr><td>001</td><td>2.28v</td></tr>
<tr><td>010</td><td>2.38v</td></tr>
<tr><td>011</td><td>2.48v</td></tr>
<tr><td>100</td><td>2.58v</td></tr>
<tr><td>101</td><td>2.68v</td></tr>
<tr><td>110</td><td>2.78v</td></tr>
<tr><td>111</td><td>2.88v</td></tr>
</table>
When the MSB bit is 1, the thresholds are as follows:
<table>
<tr><td>PRS[2:0]</td><td>Voltage</td></tr>
<tr><td>000</td><td>1.78v</td></tr>
<tr><td>001</td><td>1.88v</td></tr>
<tr><td>010</td><td>1.98v</td></tr>
<tr><td>011</td><td>2.08v</td></tr>
<tr><td>100</td><td>3.28v</td></tr>
<tr><td>101</td><td>3.38v</td></tr>
<tr><td>110</td><td>3.48v</td></tr>
<tr><td>111</td><td>3.58v</td></tr>
</table>
<i>Note: For detailed explanations, refer to the Electrical Characteristics section in the datasheet.</i></td></tr>
<tr><td>4</td><td>PVDEN</td><td>Power Voltage Detector (PVD) Enable.
0: Disable PVD
1: Enable PVD</td></tr>
<tr><td>3</td><td>CSBVBF</td><td>Clear STANDBY and VBAT flags.
Always read as 0.
0: Invalid</td></tr>
</table>

Bit Field	Name	Description
		1: Clear PWR_CTRLSTS.SBF and PWR_CTRLSTS.VBATF flags.
2	Reserved	Reserved, the reset value must be maintained.
1	PDS	Power down deep sleep bit. Cooperates with the LPS bit. 0: When the CPU outputs DEEPSLEEP as '1', the chip enters standby mode, and the voltage regulator status is controlled by the LPS bit. 1: When the CPU outputs DEEPSLEEP as '1', the chip enters stop mode.
0	LPS	Low power consumption in deep sleep mode. When PDS=0, it regulates the operation mode of the voltage regulator. 0: In standby mode, the voltage regulator is in normal mode. 1: In standby mode, the voltage regulator is in low power mode.

3.5.3 Power Control Status Register (PWR_CTRLSTS)

Address offset: 0x04

Reset value: 0x0000 0000 (reset by wakeup from STANDBY mode or system reset)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								WKUP5EN	WKUP5F	WKUP4F	WKUP3F	WKUP4POL	WKUP3POL	WKUP4EN	WKUP3EN
								rw	r	r	r	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VBATF	WKUP2POL	WKUP1POL	WKUP0POL	WKUPP1EN	WKUP2EN	WKUP1EN	WKUP0EN	WKUPP0EN	WKUPPF	WKUP2F	WKUP1F	WKUP0F	PVDO	SBF	Reserved
r	rw	rw	rw	rw	rw	rw	rw	rw	r		r	r	r	r	

Bit Field	Name	Description
31:24	Reserved	Reserved, the reset value must be maintained.
23	WKUP5EN	WKUP5 pin PH6 wakeup enable 0: WKUP pin is a general I/O, events on the WKUP pin cannot wake up the chip from standby mode 1: WKUP pin is used to wake up the chip from standby mode, configured as input with pull-down (rising edge on WKUP pin will wake up from standby mode) *Note: This bit is cleared during system reset.*
22	WKUP5F	WKUP5 pin PH6 wakeup flag. This bit is set to 1 by hardware and cleared by system reset or setting the PWR_CTRL.CWKUP5F bit. 0: No wakeup event occurred 1: Wakeup event occurred on the WKUP pin *Note: When the WKUP pin is already at a high level, enabling the WKUP pin for wakeup will detect an additional wakeup event.*
21	WKUP4F	WKUP4 pin PC5 wakeup flag.

Bit Field	Name	Description
		This bit is set to 1 by hardware and cleared by system reset or setting the PWR_CTRL.CWKUP4F bit. 0: No wakeup event occurred 1: Wakeup event occurred on the WKUP pin *Note: When the WKUP pin is already at a high level, enabling the WKUP pin for wakeup will detect an additional wakeup event.*
20	WKUP3F	WKUP3 pin PA2 wakeup flag. This bit is set to 1 by hardware and cleared by system reset or setting the PWR_CTRL.CWKUP3F bit. 0: No wakeup event occurred 1: Wakeup event occurred on the WKUP pin *Note: When the WKUP pin is already at a high level, enabling the WKUP pin for wakeup will detect an additional wakeup event.*
19	WKUP4POL	Wake-up polarity for the WKUP4 pin PC5. Use a high level or low level to wake up from standby mode, make sure to disable wake-up enable when changing the wake-up method. 0: High level wake-up 1: Low level wake-up
18	WKUP3POL	Wake-up polarity for the WKUP3 pin PA2. Use a high level or low level to wake up from standby mode, make sure to disable wake-up enable when changing the wake-up method. 0: High level wake-up 1: Low level wake-up
17	WKUP4EN	WKUP4 pin PC5 wakeup enable 0: WKUP pin is a general I/O, events on the WKUP pin cannot wake up the chip from standby mode 1: WKUP pin is used to wake up the chip from standby mode *Note: This bit is cleared during system reset.*
16	WKUP3EN	WKUP3 pin PA2 wakeup enable 0: WKUP pin is a general I/O, events on the WKUP pin cannot wake up the chip from standby mode 1: WKUP pin is used to wake up the chip from standby mode *Note: This bit is cleared during system reset.*
15	VBATF	VBAT flag. This bit is set by hardware and cleared by POR/PDR (power-on/power-down reset) or setting the PWR_CTRL.CSBVBF bit. 0: Device is not in VBAT mode 1: Device enters VBAT mode
14	WKUP2POL	Wake-up polarity for the WKUP2 pin PC13. Use a high level or low level to wake up from standby mode, make sure to disable wake-up enable when changing the wake-up method. 0: High level wake-up

Bit Field	Name	Description
		1: Low level wake-up
13	WKUP1POL	Wake-up polarity for the WKUP1 pin PA0. Use a high level or low level to wake up from standby mode, make sure to disable wake-up enable when changing the wake-up method. 0: High level wake-up 1: Low level wake-up
12	WKUP0POL	Wake-up polarity for the WKUP0 pin PE6. Use a high level or low level to wake up from standby mode, make sure to disable wake-up enable when changing the wake-up method. 0: High level wake-up 1: Low level wake-up
11	WKUPP1EN	RTC/LSECSS peripheral wakeup enable 0: RTC/LSECSS peripheral cannot wake up the chip from standby mode 1: RTC/LSECSS peripheral can wake up the chip from standby mode
10	WKUP2EN	WKUP2 pin PC13 wakeup enable 0: WKUP pin is a general I/O, events on the WKUP pin cannot wake up the chip from STANDBY mode 1: WKUP pin is used to wake up the chip from STANDBY mode *Note: This bit is cleared during system reset.*
9	WKUP1EN	WKUP1 pin PA0 wakeup enable 0: WKUP pin is a general I/O, events on the WKUP pin cannot wake up the chip from standby mode 1: WKUP pin is used to wake up the chip from standby mode *Note: This bit is cleared during system reset.*
8	WKUP0EN	WKUP0 pin PE6 wakeup enable 0: WKUP pin is a general I/O, events on the WKUP pin cannot wake up the chip from standby mode 1: WKUP pin is used to wake up the chip from standby mode *Note: This bit is cleared during system reset.*
7	WKUPP0EN	LPTIM1/LPTIM2 peripheral wakeup enable 0: LPTIM1/LPTIM2 peripheral cannot wake up the chip from standby mode 1: LPTIM1/LPTIM2 peripheral can wake up the chip from standby mode
6	WKUPPF	Peripheral wakeup flag (RTC/LSECSS/LPTIM1/LPTIM2) This bit is set to 1 by hardware and cleared by system reset or setting the PWR_CTRL.CWKUPPF bit. 0: No peripheral wakeup event occurred 1: Peripheral wakeup event occurred
5	WKUP2F	WKUP2 pin PC13 wakeup flag. This bit is set to 1 by hardware and cleared by system reset or setting the PWR_CTRL.CWKUP2F bit. 0: No wakeup event occurred 1: Wakeup event occurred on the WKUP pin

Bit Field	Name	Description
		Note: When the WKUP pin is already at a high level, enabling the WKUP pin for wakeup will detect an additional wakeup event.
4	WKUP1F	WKUP1 pin PA0 wakeup flag. This bit is set to 1 by hardware and cleared by system reset or setting the PWR_CTRL.CWKUP1F bit. 0: No wakeup event occurred 1: Wakeup event occurred on the WKUP pin *Note: When the WKUP pin is already at a high level, enabling the WKUP pin for wakeup will detect an additional wakeup event.*
3	WKUP0F	WKUP0 pin PE6 wakeup flag. This bit is set to 1 by hardware and cleared by system reset or setting the PWR_CTRL.CWKUP0F bit. 0: No wakeup event occurred 1: Wakeup event occurred on the WKUP pin *Note: When the WKUP pin is already at a high level, enabling the WKUP pin for wakeup will detect an additional wakeup event.*
2	PVDO	PVD output. This bit is only valid when PVD is enabled by the PWR_CTRL.PVDEN bit. When PWR_CTRL6.PVDSRC=0: 0: VDD/VDDA is higher than the PVD threshold selected by PWR_CTRL.PRS[2:0] 1: VDD/VDDA is lower than the PVD threshold selected by PWR_CTRL.PRS[2:0] When PWR_CTRL6.PVDSRC=1: 0: Voltage at the PVD_IN pin is higher than Vrefint 1: Voltage at the PVD_IN pin is lower than Vrefint *Note: PVD is stopped in standby mode. Therefore, after standby mode or reset, this bit remains 0 until the PWR_CTRL.PVDEN bit is set.* *Note: After enabling the PVD function, a delay of 2000 cycles is required before determining the PVDO bit*
1	SBF	STANDBY flag. This bit is set by hardware and cleared by POR/PDR (power-on/power-down reset) or by setting the PWR_CTRL.CSBVBF bit. 0: System is not in standby mode 1: System enters standby mode
0	Reserved	Reserved, the reset value must be maintained.

3.5.4 Power Control Register 2 (PWR_CTRL2)

Address offset: 0x08

Reset value: 0x0000 2704

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved					IWDGRSTEN	Reserved							BSRSTBRET	BSRVBRET	Reserved
					rw								rw	rw	

Bit Field	Name	Description
31:11	Reserved	Reserved, the reset value must be maintained.
10	IWDGRSTEN	Independent watchdog reset enable. 0: Independent watchdog cannot generate a reset to RCC 1: Independent watchdog can generate a reset to RCC
9:3	Reserved	Reserved, the reset value must be maintained.
2	BSRSTBRET	Backup SRAM retention enable in STANDBY mode. 0: Backup SRAM does not retain in STANDBY mode 1: Backup SRAM retains in STANDBY mode
1	BSRVBRET	Backup SRAM retention enable in VBAT mode. 0: Backup SRAM does not retain in VBAT mode 1: Backup SRAM retains in VBAT mode
0	Reserved	Reserved, the reset value must be maintained.

3.5.5 Power Control Register 3 (PWR_CTRL3)

Address offset: 0x0C

Reset value: 0x0009DB2F

This register is write-protected and must be unlocked by writing the key 0x55AA before each software write operation to this register.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	BKRSEL	Reserved													
	rw														

Bit Field	Name	Description
31:15	Reserved	Reserved, the reset value must be maintained.
14	BKRSEL	BKR voltage regulator output level selection 0: 0.8V

Bit Field	Name	Description
		1: 0.9V.
13:0	Reserved	Reserved, the reset value must be maintained.

3.5.6 Power Control Register 4 (PWR_CTRL4)

Address offset: 0x10

Reset value: 0x00000145

This register is write-protected and must be unlocked by writing the key 0x55AA before each software write operation to this register.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved					LSENIMEN	Reserved								LSEDRVRIM[1:0]	
					rw									rw	

Bit Field	Name	Description
31:11	Reserved	Reserved, the reset value must be maintained.
10	LSENIMEN	LSE noise cancellation enable. Set or cleared by software to eliminate noise and optimize the 50% duty cycle. 0: Disable noise cancellation 1: Enable noise cancellation (requires enabling after setting the RCC_BDCTRL.LSERD bit)
9:2	Reserved	Reserved, the reset value must be maintained.
1:0	LSEDRVRIM[1:0]	LSE Transconductance gm Selection 00：13(μ A/V) 01：20(μ A/V) 11：40(μ A/V)

3.5.7 NRST Control Register (PWR_NRSTCTRL)

Address offset: 0x20

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Reserved	NRSTDFW[7:0]	NRSTDFEN	NRSTAFEN	Reserved
	rw	rw	rw	

Bit Field	Name	Description
31:12	Reserved	Reserved, the reset value must be maintained.
11:4	NRSTDFW[7:0]	NRST digital filter width Filter width=sysclk * NRSTDFW Note: When sysclk=240M, the recommended configuration for NRSTDFW is 0x9, which needs to be adjusted according to the usage scenario. When the environment is harsh, the value should be appropriately increased
3	NRSTDFEN	NRST Digital Filter Enable 0: Enable 1: Disable
2	NRSTAFEN	NRST Analog Filter Enable 0: Enable 1: Disable
1:0	Reserved	Reserved, the reset value must be maintained.

3.5.8 Power Control Register 5 (PWR_CTRL5)

Address offset: 0x28

Reset value: 0x00A91099

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LSECSSCRYIEN	Reserved	LSECSSCRYF	Reserved												
rw		r													

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															

Bit Field	Name	Description
31	LSECSSCRYIEN	LSE crystal mode clock security system interrupt enable. Set and cleared by software to enable and disable the clock security system interrupt. 0: Disable LSE clock crystal mode security system interrupt 1: Enable LSE clock crystal mode security system interrupt
30	Reserved	Reserved, the reset value must be maintained.
29	LSECSSCRYF	LSE crystal mode clock security fault flag. Set by hardware when a fault is detected in the external LSE oscillator, cleared by writing a 1 to the PWR_CTRL6.LSECSSF bit. 0: No clock security system fault caused by LSE clock failure in crystal mode

Bit Field	Name	Description
		1: Clock security system fault caused by LSE clock failure in crystal mode
28:0	Reserved	Reserved, the reset value must be maintained.

3.5.9 Power Control Register 6 (PWR_CTRL6)

Address offset: 0x2C

Reset value: 0x016681C4

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LSECSSSW	LPWKUPSEL	NRSTSEL		LSEFHF	LSEFLF	LSECSSF	Reserved								
rw	rw	rw		r	r	rc_wl									

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			LCOSEL	LCOEN	LSEFHTHR	Reserved				PVDSRC	Reserved				
			rw	rw	rw					rw					

Bit Field	Name	Description
31	LSECSSSW	LSE clock security system switch. Set and cleared by software to choose whether to switch the RTC clock source from LSE to LSI when the LSE clock fails. 0: Do not switch RTC clock source when LSE fails 1: Switch RTC clock source to LSI clock when LSE fails
30	LPWKUPSEL	Low power mode wakeup clock selection. Set and cleared by software, ensure both LSI clock and LSE clock are enabled when switching. 0: Wakeup using LSI 1: Wakeup using LSE
29:28	NRSTSEL	NRST (PH6) pin function configuration. Set and cleared by software, can configure the NRST pin for general GPIO use. 00: Reset input/output 01: Reset input 10: General GPIO function 11: Reset input/output
27	LSEFHF	LSE frequency exceeding high limit flag. Set to 1 by hardware when LSE frequency exceeds the high limit value, cleared by writing a 1 to the PWR_CTRL6.LSECSSF bit. The high limit value in crystal mode is fixed at 160KHz. In bypass mode, the high limit value can be configured as 160KHz or 1MHz using PWR_CTRL6.LSEFHTHR. 0: No frequency exceeding high limit value occurred 1: Frequency exceeding high limit value occurred
26	LSEFLF	LSE frequency below low limit flag. Set to 1 by hardware when LSE frequency is below 20KHz, cleared by writing a

Bit Field	Name	Description
		1 to the PWR_CTRL6.LSECSSF bit. 0: No frequency below low limit value occurred 1: Frequency below low limit value occurred
25	LSECSSF	LSE clock security system flag. Set by hardware when a fault is detected in the external LSE oscillator, cleared by writing a 1 to the PWR_CTRL6.LSECSSF bit. 0: No clock security system fault caused by LSE clock failure 1: Clock security system fault caused by LSE clock failure
24:13	Reserved	Reserved, the reset value must be maintained.
12	LCOSEL	LCO low-speed clock output selection 0: PA2 outputs LSI clock 1: PA2 outputs LSE clock
11	LCOEN	LCO low-speed clock output enable Supports clock output in STOP0 and STANDBY modes. 0: Disable 1: Enable
10	LSEFHTHR	Bypass mode LSE frequency high limit value configuration 0: Bypass mode high limit value is 1M 1: Bypass mode high limit value is 160K
9:6	Reserved	Reserved, the reset value must be maintained.
5	PVDSRC	PVD input source selection 0: PVD input source is VDD 1: PVD input source is the PVD_IN pin *Note: When PVDSRC is configured as 1, the input voltage at the PVD_IN pin is compared with Vrefint (1.2V reference voltage). PVD_IN can be reused as PB7 or PA10 (default is PA10).*
4:0	Reserved	Reserved, the reset value must be maintained.

3.5.10 EMC Control Register (PWR_EMCCTRL)

Address offset: 0xC0

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BSRECCF[1:0]		BSRECCRSTEN[1:0]		BSRECCIEN[1:0]		EMCRST	BKPGBRST	BKPGBNRST	BKPCLPRST	BKPGBDET	BKPGBNDET	BKPCLPDET	Reserved		
rc_w1		rw		rw		rw	rw	rw	rw	rw	rw	rw			

Bit Field	Name	Description
31:16	Reserved	Reserved, the reset value must be maintained.

Bit Field	Name	Description
15:14	BSRECCF[1:0]	BKP SRAM ECC error flag 00: No ECC error occurred 01: 1-bit ECC error occurred 10: 2-bit ECC error occurred 11: 1-bit and 2-bit ECC errors occurred
13:12	BSRECCRSTEN[1:0]	BKP SRAM ECC error reset enable 00: No reset generated when ECC error is detected 01: Reset generated when 1-bit ECC error is detected 10: Reset generated when 2-bit ECC error is detected 11: Reset generated when 1-bit and 2-bit ECC errors are detected
11:10	BSRECCIEN[1:0]	BKP SRAM ECC error interrupt enable 00: No interrupt triggered when ECC error is detected 01: Interrupt triggered when 1-bit ECC error is detected 10: Interrupt triggered when 2-bit ECC error is detected 11: Interrupt triggered when 1-bit and 2-bit ECC errors are detected
9	EMCRST	Main power domain and backup domain EMC reset enable This bit is set and cleared by software. When set to 1, a reset will be generated for the main power domain or backup domain during GB, GBN, and clamp. 0: Disable reset request 1: Enable reset request
8	BKPGBRST	Backup domain EMC GB reset enable This bit is set and cleared by software. 0: Disable reset request 1: Enable reset request
7	BKPGBNRST	Backup domain EMC GBN reset enable This bit is set and cleared by software. 0: Disable reset request 1: Enable reset request
6	BKPCLPRST	Backup domain EMC Clamp reset enable This bit is set and cleared by software. 0: Disable reset request 1: Enable reset request
5	BKPGBDET	Backup domain EMC GB detection enable This bit is set and cleared by software. 0: Disable detection 1: Enable detection
4	BKPGBNDET	Backup domain EMC GBN detection enable This bit is set and cleared by software. 0: Disable detection 1: Enable detection
3	BKPCLPDET	Backup domain EMC Clamp detection enable This bit is set and cleared by software.

Bit Field	Name	Description
		0: Disable detection 1: Enable detection
2:0	Reserved	Reserved, the reset value must be maintained.

3.5.11 EMC Control Register 2 (PWR_EMCCTRL2)

Address offset: 0xC4

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BKPGBF	BKPGBNF	BKPCLPF	GB3F	GB2F	GB1F	GB0F	GBN3F	GBN2F	GBN1F	GBN0F	CLP3F	CLP2F	CLP1F	CLP0F	EMCFCLR
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	w

Bit Field	Name	Description
31:16	Reserved	Reserved, the reset value must be maintained.
15	BKPGBF	Backup Domain EMC GB Flag: When the backup domain detects an EMC GB event, this bit is set by the hardware. Writing 1 to the EMCFCLR bit clears this bit. 0: No EMC GB event 1: EMC GB event occurred
14	BKPGBNF	Backup Domain EMC GBN Flag: When the backup domain detects an EMC GBN event, this bit is set by the hardware. Writing 1 to the EMCFCLR bit clears this bit. 0: No EMC GBN event 1: EMC GBN event occurred
13	BKPCLPF	Backup Domain EMC Clamp Flag: When the backup domain detects an EMC Clamp event, this bit is set by the hardware. Writing 1 to the EMCFCLR bit clears this bit. 0: No EMC Clamp event 1: EMC Clamp event occurred
12	GB3F	Main Domain EMC GB3 Flag: When the main domain detects an EMC GB3 event, this bit is set by the hardware. Writing 1 to the EMCFCLR bit clears this bit. 0: No EMC GB3 event 1: EMC GB3 event occurred
11	GB2F	Main Domain EMC GB2 Flag: When the main domain detects an EMC GB2 event, this bit is set by the

Bit Field	Name	Description
		hardware. Writing 1 to the EMCFCLR bit clears this bit. 0: No EMC GB2 event 1: EMC GB2 event occurred
10	GB1F	Main Domain EMC GB1 Flag: When the main domain detects an EMC GB1 event, this bit is set by the hardware. Writing 1 to the EMCFCLR bit clears this bit. 0: No EMC GB1 event 1: EMC GB1 event occurred
9	GB0F	Main Domain EMC GB0 Flag: When the main domain detects an EMC GB0 event, this bit is set by the hardware. Writing 1 to the EMCFCLR bit clears this bit. 0: No EMC GB0 event 1: EMC GB0 event occurred
8	GBN3F	Main Domain EMC GBN3 Flag: When the main domain detects an EMC GBN3 event, this bit is set by the hardware. Writing 1 to the EMCFCLR bit clears this bit. 0: No EMC GBN3 event 1: EMC GBN3 event occurred
7	GBN2F	Main Domain EMC GBN2 Flag: When the main domain detects an EMC GBN2 event, this bit is set by the hardware. Writing 1 to the EMCFCLR bit clears this bit. 0: No EMC GBN2 event 1: EMC GBN2 event occurred
6	GBN1F	Main Domain EMC GBN1 Flag: When the main domain detects an EMC GBN1 event, this bit is set by the hardware. Writing 1 to the EMCFCLR bit clears this bit. 0: No EMC GBN1 event 1: EMC GBN1 event occurred
5	GBN0F	Main Domain EMC GBN0 Flag: When the main domain detects an EMC GBN0 event, this bit is set by the hardware. Writing 1 to the EMCFCLR bit clears this bit. 0: No EMC GBN0 event 1: EMC GBN0 event occurred
4	CLP3F	Main Domain EMC Clamp3 Flag: When the main domain detects an EMC Clamp3 event, this bit is set by the hardware. Writing 1 to the EMCFCLR bit clears this bit. 0: No EMC Clamp3 event 1: EMC Clamp3 event occurred
3	CLP2F	Main Domain EMC Clamp2 Flag: When the main domain detects an EMC Clamp2 event, this bit is set by the hardware. Writing 1 to the EMCFCLR bit clears this bit. 0: No EMC Clamp2 event

Bit Field	Name	Description
		1: EMC Clamp2 event occurred
2	CLP1F	Main Domain EMC Clamp1 Flag: When the main domain detects an EMC Clamp1 event, this bit is set by the hardware. Writing 1 to the EMCFCLR bit clears this bit. 0: No EMC Clamp1 event 1: EMC Clamp1 event occurred
1	CLP0F	Main Domain EMC Clamp0 Flag: When the main domain detects an EMC Clamp0 event, this bit is set by the hardware. Writing 1 to the EMCFCLR bit clears this bit. 0: No EMC Clamp0 event 1: EMC Clamp0 event occurred
0	EMCFCLR	Clear EMC Flag: 0: No effect 1: Clear EMC flag

4 Reset And Clock Control (RCC)

The highest frequency of N32H480 series is 180MHz, while the highest frequency of other series is 240MHz. Different series and models support different peripherals and quantities. For specific details, please refer to the corresponding model's datasheet.

4.1 Reset Control Unit

Supports the following three types of reset:

- Power Reset
- System Reset
- Low power domain Reset

4.1.1 Power Reset

A power reset occurs in the following circumstances:

- Power-on reset (POR reset).
- Power-down reset(PDR reset).
- When exiting STANDBY mode.

A power reset will set all registers to their reset values except the ones in the Backup domain (see Figure 3-1).

The reset source in the diagram will ultimately act on the NRST pin and maintain a low level during the reset process. The reset entry vector is fixed at address 0x0000_0004. For more details, refer to Table 6-1 Vector Table.

4.1.2 System Reset

Except the reset flags in the Control/Status Register (RCC_CTRLSTS) and the registers in the low power domain (see Figure 3-1), a system reset sets all registers to their reset values.

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

- A low level on the NRST pin (external reset)
- Window watchdog event (WWDG reset)
- Independent watchdog eevent (IWDG reset)
- Software reset (SW reset)
- Low power management reset
- SMPU protection reset
- RAM ECC or parity error reset
- FLASH ECC reset
- Backup domain EMC reset

- Main domain EMC reset
- BOR reset

The reset source can be identified by checking the reset flags in the Control/Status register (RCC_CTRLSTS).

4.1.2.1 Software reset

A software reset can be generated by setting the SYSRESETREQ bit in Cortex^{TM}-M4F Application Interrupt and Reset Control Register. Refer to Cortex^{TM}-M4 technical reference manual for further information.

4.1.2.2 Low-power management reset

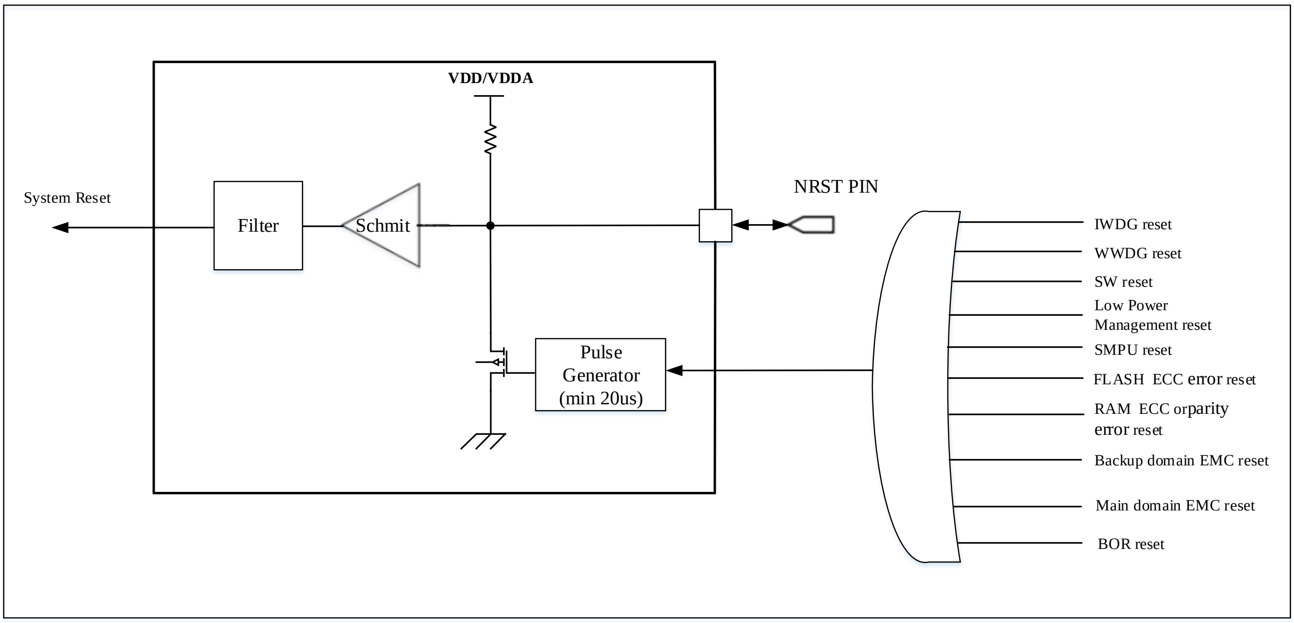
Low-power management reset can be generated by using the following methods:

- Generate low power management reset when entering STANDBY mode: This reset is enabled by setting the nRST_STDBY bit in the user OptionByte. At this time, even if the procedure to enter STANDBY mode is performed, the system will be reset instead of entering STANDBY mode.
- Generate low power management reset when entering STOP0 mode: This reset is enabled by setting the nRST_STOP bit in the user OptionByte. At this time, even if the process to enter STOP0 mode is performed, the system will be reset instead of entering STOP0 mode.

The system reset signal provided to the chip is output on the NRST pin. The pulse generator guarantees a minimum reset pulse duration of 20μs for each reset source (external or internal). For external reset, the reset pulse is generated while the NRST pin is asserted low.

The Figure below shows the system reset generation circuit.

Figure 4-1 System Reset Generation



4.1.3 Low Power Domain Reset

The backup area has two dedicated resets, which only affect the backup area (refer to Figure 3-1 Power Supply

Diagram):

- Software reset: The low power domain reset can be generated by setting the RCC_BDCTRL.BDSFTRST bit.
- When both V_{DD} and V_{BAT} are powered off, a reset of the backup area will only be triggered when V_{DD} or V_{BAT} is powered on.

4.2 Clock Control Unit

Five different clock sources can be used to drive the system clock (SYSCLK):

- HSI oscillator clock;
- HSE oscillator clock;
- PLL clock;
- SHRTPLL clock;
- USBHS240M clock.

This selects the clock source as the internal clock, PLL clock, or HSE clock through RCC_CFG.SCLKSW; when the clock source is selected as the PLL clock, further select PLL, SHRTPLL, or USBHS240M through RCC_PLLCTRL.SCLKPLLSEL.

The devices have the following two secondary clock sources:

- LSI: 32 KHz internal low-speed RC can be used to drive the independent watchdog and can be selected to drive the RTC through the program. The RTC is used to automatically wake up the system from STOP0/STANDBY mode.
- LSE: 32.768 kHz external low-speed crystal can also be selected by software to drive RTC/LPTIMER/ LPUART.

Each clock source can be turned on or off independently when it is not used to optimize power consumption.
Several prescalers can be used to configure the frequencies of the AHB, the high-speed APB (APB2), and the low-speed APB (APB1) domains. The maximum frequencies of the AHB, APB2, and APB1 domains are 240 MHz, 180MHz, and 180 MHz respectively.

Note: Avoid configuring the APB2 frequency to be the same as the AHB frequency.

Note: The highest frequency of N32H480 series is 180MHz, while the highest frequency of other series is 240MHz.

The clock frequency of the SDIO, USBHS, and ETH interfaces can be configured as HCLK or HCLK/2.

RCC provides the Cortex System Timer (SysTick) external clock with the AHB clock (HCLK) divided by 8. This clock or Cortex clock(HCLK) can be selected to drive the SysTick by programming the SysTick Control and Status Register.

The ADC operating clock is generated by dividing the AHB clock, PLL clock or USBHS240M. The ADC1M clock is generated by dividing by HSE or HSI.

The FDCAN clock source can be selected as SHRTPLL divided clock, PLL divided clock, or HSE, and the frequency must be configured as a multiple of 20M.

The clock source for I2S2 and I2S3 can be selected as SYSCLK, SHRTPLL, HSI, or the I2S_CLKIN pin.

The clock source for LPTIM1 and LPTIM2 can be selected as LSI, LSE, HSI, HSE, or PCLK1.

The clock frequencies of GTIM1/2/3/4/5/6/7 and BTIM1/2/3 are automatically set by hardware. There are two scenarios:

- If the APB prescaler is 1, the timer clock frequencies are set to the same frequency as that of the APB domain to which the timers are connected.
- Otherwise, they are set to twice the frequency of the APB domain to which the timers are connected.

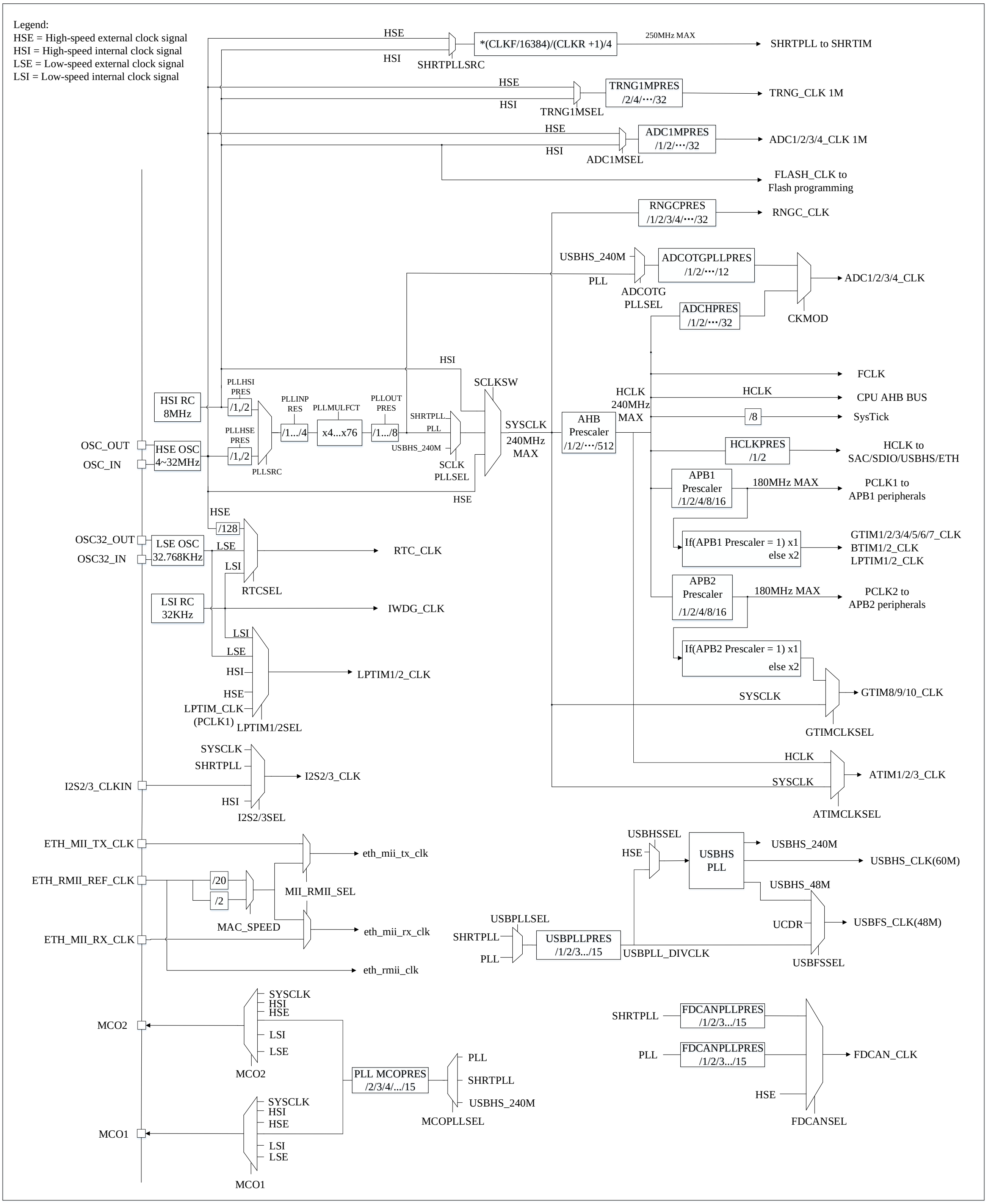
For timers GTIM8/9/10, the clock source can be selected as APB or SYSCLK.

For timers ATIM1/2/3, the clock source can be selected as AHB or SYSCLK.

FCLK is the free-running clock of Cortex^{TM}-M4F. For more details, refer to the ARM Cortex^{TM}-M4 technical reference manual.

4.2.1 Clock Tree Diagram

Figure 4-2 Clock Tree



Notes:

1. *For more details about the internal and external clock source characteristics, please refer to the "Electrical Characteristics" section in the product datasheet.*
2. *Avoid configuring the APB2 frequency to be the same as the AHB frequency.*
3. *The highest frequency of N32H480 series is 180MHz, while the highest frequency of other series is 240MHz.*
4. *For ATIM1-3/GTIM8-10, the maximum operating clock is the corresponding series' maximum main frequency. For GTIM1-7/BTIM1-2 /LPTIM1-2, the maximum operating clock is 180MHz. Therefore, when AHB is greater than 180MHz and GTIM1-7/BTIM1-2 /LPTIM1-2 are needed, APB1 cannot be divided by 1 or 2. For example, when the AHB frequency is 240MHz and GTIM1 needs to be used, APB1 can only be divided by 4 to achieve 60MHz, resulting in GTIM1 operating at 120MHz.*
5. *When the input clock is less than 4 MHz, this clock cannot be used as the PLL clock source*
6. *The FDCAN module clock source (FDCAN_CLK) shall be configured to the following frequencies: 20M, 40M, 80M and cannot be higher than the APB1 clock frequency (PCLK1).*

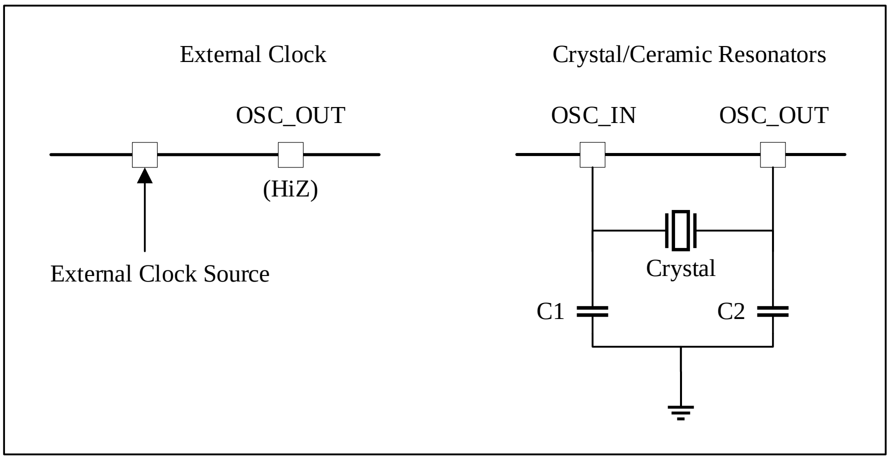
4.2.2 HSE Clock

The high-speed external clock signal (HSE) can be generated from the following two clock sources:

- HSE external crystal/ceramic resonator
- HSE user external clock

To reduce distortion of the clock output and shorten the start-up stablize time, the crystal/ceramic resonator and load capacitor must be placed as close as possible to the oscillator pins of the chip. The loading capacitance value must be adjusted according to the chosen oscillator.

Figure 4-3 HSE/LSE Clock Source



4.2.2.1 External clock source (HSE bypass mode)

In this mode, an external clock source must be provided. Its frequency range is 1-50 MHz. Users can select this mode by setting the RCC_CTRL.HSEBP and RCC_CTRL.HSEEN bits. The external clock signal (50% duty cycle square, sine or triangle wave) must be connected to the OSC_IN pin while the OSC_OUT pin must be left floating (Hi-Z). See Figure 4-3.

4.2.2.2 External crystal/ceramic resonator (HSE crystal mode)

The 4 to 32 MHz external oscillator has the advantage of producing a more accurate main clock for the system. The associated hardware configuration is shown in See Figure 4-3. For more details, please refer to the electrical characteristics section of the datasheet.

The RCC_CTRL.HSERDF bit indicates whether the high-speed external oscillator is stable or not. At startup, the clock is not released until this bit is set by hardware. An interrupt can be generated if enabled in the Clock Interrupt Register (RCC_CLKINT).

HSE clock can be switched on and off by setting the RCC_CTRL.HSEEN bit.

4.2.3 HSI Clock

The HSI (High Speed Internal) clock signal is generated by an internal 8 MHz RC oscillator and can be used directly as system clock or as PLL input after dividing by 2. The HSI RC oscillator can provide a clock source without any external devices. It also has a shorter startup time than the HSE crystal oscillator. However, even with calibration the frequency is less accurate.

The HSI clock frequency of each chip has been calibrated to 1% (25°C) before leaving the factory. After the system reset, the factory calibration value is loaded into the RCC_CTRL.HSICAL[11:0] bits.

If the user application is subject to voltage or temperature variations, this may affect the accuracy of the RC oscillator. The HSI frequency can be trimmed by using the RCC_CTRL.HSITRIM[6:0] bits.

The RCC_CTRL.HSIRDF bit flag indicates if the HSI RC oscillator is stable. At startup, the HSI RC output clock is not released until this bit is set by hardware. HSI clock can be switched on and off using the RCC_CTRL.HSIEN bit.

If the HSE crystal oscillator fails, the HSI clock will be used as the backup clock source. Refer to section 4.2.10 for the Clock Security System.

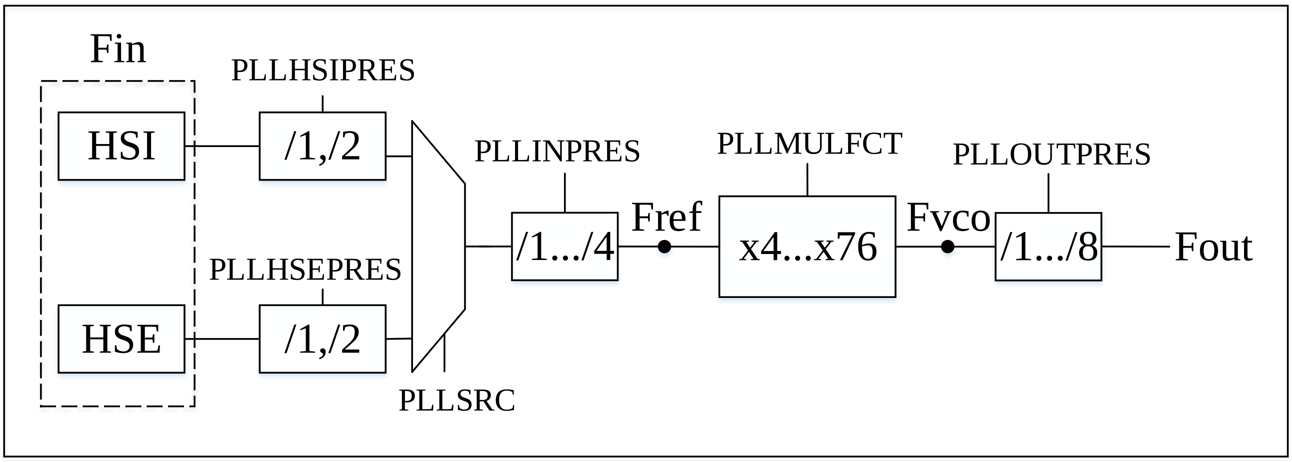
4.2.4 PLL Clock

The internal PLL can be used to multiply the HSI or the HSE clock frequency. Refer to Figure 4-2 Clock Tree, the PLL configuration (selection of PLL input clock (HSI/HSE and divider) and multiplication factor) must be done before enabling PLL. Once the PLL is enabled, these parameters cannot be changed. The PLL can be configured using control bits in RCC_CTRL, RCC_CFG and RCC_PLLCTRL registers.

If the PLL interrupt is enabled in the clock interrupt register, an interrupt request can be generated when the PLL is ready.

As shown in the following figure, the Fin frequency is required to be in the range of 4MHz to 50MHz, the Fref frequency is required to be in the range of 4MHz to 25MHz, the Fvco frequency is required to be in the range of 64MHz to 500MHz, and the PLLMULFCT multiplication factor should not be less than 8x.

Figure 4-4 PLL Clock Source Selection



4.2.5 SHRTPLL Clock

SHRTPLL can be used to multiply the frequency of the HSI or HSE clock as the working clock source for SHRTIM. Please refer to Figure 4-2 for the clock tree.

The process of configuring SHRTPLL as the clock source for the high-precision timer is as follows:

1. Configure RCC_AHBPRST.SHRTPLLPHARST as 1 to enable phase reset
2. Configure RCC_SHRTPLLCTRL3.SHRTPLLEN as 1 to enable the power supply of the SHRTPLL module
3. Configure RCC_SHRTPLLCTRL3.SHRTPLLSRC to select the clock source as HSI or HSE
4. Configure RCC_SHRTPLLCTRL1.SHRTPLLMODE as 0 to select the operating mode as normal mode
5. Configure RCC_SHRTPLLCTRL1.SATEN and RCC_SHRTPLLCTRL1.FRACEN as 1 to enable the saturation and fractional accumulation functions
6. Configure RCC_SHRTPLLCTRL1.CLKF to select the multiplication factor, RCC_SHRTPLLCTRL2.CLKR to select the division factor, and RCC_SHRTPLLCTRL3.BWTRIM to select the bandwidth
7. Configure RCC_AHBPRST.SHRTPLLRST as 1, then after 5us, configure RCC_AHBPRST.SHRTPLLRST as 0 to reset SHRTPLL
8. Wait for RCC_SHRTPLLCTRL3.SHRTPLLRDF to be set to 1, indicating that the clock is ready
9. Configure RCC_AHBPRST.SHRTPLLPHARST as 0 to disable the phase reset
10. Configure RCC_AHBPCLKEN.SHRTIMEN as 1 to enable the high-precision timer

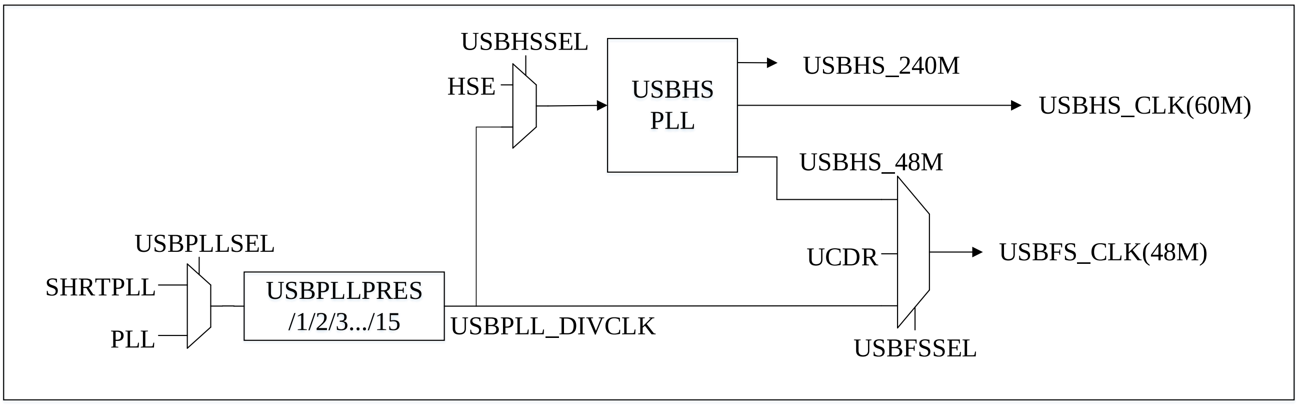
The clock source frequency range is from 1MHz to 50MHz, and the output clock frequency range is from 75MHz to 250MHz. The calculation formula for SHRTPLL is as follows, with the maximum allowable output frequency configured as 250MHz:

$$\text{SHRTPLL frequency} = \text{Clock source frequency} * (\text{CLKF}[25:0]/16384) / (\text{CLKR}[5:0] + 1) / 4$$

When SHRTIM is not operating, it can also be used as the clock source for I2S, FDCAN, USB, or the system clock.

4.2.6 USB Clock

There is a dedicated PLL internally for generating the USB clock, refer to Figure 4-2 for the clock tree. Configure the USB PLL clock source and division factor using RCC_CFG3.USBPLLSEL and RCC_CFG3.USBPLLPRES[3:0]. The resulting USBPLL divided clock can be used as the working clock source for the USBHS or USBFS modules.



4.2.6.1 USBHS Clock

By selecting the clock source as HSE or PLL through RCC_USBHSCTRL1.USBHSSEL, ensure that the actual frequency of the clock source is 16M, 19.2M, 20M, 24M, 26M, or 32M. Confirm the frequency selection through RCC_USBHSCTRL1.USBHSFSEL and the bandwidth selection through RCC_USBHSCTRL1.PLLBW. The USBHS PLL module automatically multiplies to obtain three clocks: USBHS240M, USBHS48M, and USBHSCLK. USBHS240M can be used as the system clock source or ADC working clock source; USBHS48M can be used as the USBFS clock source; USBHSCLK is the working clock source of the USBHS module.

4.2.6.2 USBFS Clock

By selecting the clock source as USBHS48M, UCDR, or the USBPLL divided clock through RCC_CFG3.USBFSSEL, when choosing the USBPLL divided clock source, the input frequency must be greater than or equal to 96M, the output frequency must be configured as 48M.

4.2.7 LSE Clock

The LSE crystal is a 32.768KHz low speed external crystal or ceramic resonator. It provides a low-power and accurate clock source for the real-time clock or other timing functions.

The LSE clock is switched on and switched off by the RCC_BDCTRL.LSEEN bit.

The RCC_BDCTRL.LSERD bit indicates whether the LSE clock is stable. During the startup phase, the LSE clock signal is not released until this bit is set by hardware. If enabled in the clock interrupt register, an interrupt request can be generated.

4.2.7.1 LSE external clock source(LSE bypass)

In this mode, an external clock source with a frequency of up to 1 MHz can be provided. Users can select this mode by setting the RCC_BDCTRL.LSEBP and RCC_BDCTRL.LSEEN bits. The external clock signal(square, sinus or triangle wave) with 50% duty cycle must be connected to the OSC32_IN pin while the OSC32_OUT pin must be left floating (Hi-Z).

4.2.8 LSI Clock

The LSI RC can clock the IWDG and AWU in STOP0 and STANDBY modes. The LSI clock frequency is about 32kHz. For further information please refer to the Electrical Characteristics section of the data sheet.

The LSI clock can be turned on or off using the RCC_CTRLSTS.LSIEN bit.

The RCC_CTRLSTS.LSIRD bit flag indicates if the LSI clock is stable. At startup, the clock is not released until this bit is set by hardware. An interrupt can be generated if enabled in the Clock Interrupt Register (RCC_CLKINT).

An internal embedded LSI clock detection system is in place, which automatically recovers when a low LSI frequency is detected.

4.2.8.1 LSI calibration

You can calibrate and compensate for the internal low-speed oscillator LSI frequency error to obtain a higher accuracy RTC timebase and IWDG timeout (when these peripherals are clocked by LSI).

Calibration can be achieved by using the capture measurement of the LSI clock frequency with GTIM1's channel 3, GTIM5's channel 1, or GTIM7's channel 2. Software can adjust the 22-bit prescaler of the RTC to obtain an accurate RTC clock reference and calculate an accurate IWDG timeout.

4.2.9 System Clock (SYSCLK) Selection

After a system reset, the HSI oscillator is selected as the system clock. When the clock source is used directly or indirectly through the PLL as the system clock, it cannot be stopped.

Switching from one clock source to another will only occur when the target clock source is ready (either after a delay to start the stabilization phase or PLL stabilization). When the selected clock source is not ready, the switching of the system clock will not occur until the target clock source is ready.

The status bits in the Clock Control Register (RCC_CTRL) indicate which clock is ready and which clock is currently being used as the system clock.

4.2.10 Clock Security System (CLKSS)

HSE, HSI, and LSE each have their own clock security systems to detect clock failures.

4.2.10.1 HSE clock security system

By setting the RCC_CTRL.HSECSSEN bit, the HSE clock security system is activated. Once activated, the clock detector is enabled after the start-up delay of the HSE oscillator and disabled when the HSE clock is turned off.

If the HSE oscillator is directly or indirectly used as the system clock (indirectly means: it is used as the PLL input clock, and the PLL clock is used as the system clock), the clock failure will cause a switch of the system clock to the MSI oscillator and the disabling of the external HSE oscillator. If HSE clock (divided or not) is selected as PLL input clock then upon HSE clock failure, the PLL will be turned off.

If a clock failure occurs, the clock failure event will be sent to the brake input of the advanced timer and generate a clock security system interrupt. The interrupt is connected to the Cortex™-M4F Non-Maskable Interrupt (NMI), allowing software to take rescue measures in the interrupt, set and clear the interrupt.

The HSE clock security system can enable different error condition detections, including phase offset out of range, frequency exceeding the upper limit, and frequency falling below the lower limit. The thresholds for these conditions are flexible and configurable. For specific details, refer to the relevant register section 4.3.16.

4.2.10.2 HSI clock security system

By setting the RCC_CSSCTRL.HSICSSEN bit, the HSI clock security system is activated. Once activated, the clock detector is enabled after the start-up delay of the HSI oscillator and disabled when the HSI clock is turned off. If RCC_CLKINT.HSICSSIEN is enabled, an interrupt will be generated when the HSI CSS event occurs.

4.2.10.3 LSE clock security system

By setting the RCC_BDCTRL.LSECSSEN bit, the LSE clock security system is activated. Once activated, the clock detector is enabled after the start-up delay of the LSE oscillator and disabled when the LSE clock is turned off. If RCC_CLKINT.LSECSSIEN is enabled, an interrupt will be generated when the LSE CSS event occurs.

If the LSE oscillator serves as the RTC clock source, the hardware can automatically switch the RTC clock source to the LSI oscillator when the clock fails, by configuring it through PWR_CTRL6.LSECSSW.

4.2.11 RTC Clock

By programming RCC_BDCTRL.RTCSEL [1:0] bits, the RTCCLK clock source can be either the HSE/128, LSE, or LSI clocks. This selection cannot be changed unless the low power domain is reset.

The LSE and LSI clocks are in the low power domain, but the HSE clocks are not. therefore:

- If LSE or LSI is selected as RTC clock:
  - As long as VBAT remains powered, the RTC continues to operate even if VDD power is cut off.
- If the HSE clock divided by 128 is used as the RTC clock:
  - If VDD power is cut off or the voltage regulator is turned off, the RTC state becomes uncertain.
  - The power control register PWR_CTRL.DBKP bit must be set to 1 (disable write protection for the backup domain).

4.2.12 Watchdog Clock

If the IWDG is started by either hardware option or software access, the LSI oscillator will be forced ON and cannot be disabled. After the LSI oscillator is stabilized, the clock is provided to the IWDG.

4.2.13 Clock Output (MCO)

In RUN mode, the Microcontroller Clock Output (MCO) function allows the clock signal to be output to the external MCO pin, and the corresponding GPIO port register must be configured for the corresponding function.

By setting the RCC_MCOCFG.MCO1[3:0] bits, select the MCO1 clock, and by setting the RCC_MCOCFG.MCO2[3:0] bits, select the MCO2 clock. The following clock sources are available for output:

- SYSCLK
- HSI

- HSE
- PLL Clock Division
- SHRTPLL Clock Division
- USBHS240M Clock Division
- LSI
- LSE

When selecting the clock signal for PLL clock division, SHRTPLL clock division, or USBHS240M clock division, further selection of the clock source and division factor is required through RCC_CFG.MCOPLLSEL and RCC_CFG.MCOPRES[3:0].

4.2.14 Voltage Reference Buffer (VREFBUF)

4.2.14.1 Introduction to VREFBUF

The chip features an internal voltage reference buffer that can be used as the voltage reference for the ADC, 12-bit DAC, and internal 6-bit DAC of the COMP. It can also be used as an external component's voltage reference through the VREF+ pin. When the VREF+ pin is connected to the VDDA pin within the package (refer to the package pin description in the datasheet for details), the voltage reference buffer is not available, must remain disabled, and the RCC_VREFCTRL.HIM bit must be set to 1.

Note: VREFBUF is strictly prohibited to turn on the external input at the same time as the internal output mode, and the output mode and the external input mode can only be one or the other.

4.2.14.2 VREFBUF Functional Description

The built-in voltage reference buffer supports three voltage levels: 2.048V, 2.5V, and 2.9V. The voltage level can be selected by configuring the RCC_VREFCTRL.VLSEL[1:0] register bits.

Table 4-1 Voltage Reference Buffer Level Selection

RCC_VREFCTRL.VLSEL[1:0]	Voltage Reference Buffer Level Selection
00	2.048V
01	2.5V
10	2.9V

The internal voltage reference can be set in four different modes based on the configuration of the EN and HIM bits. These modes are listed in the table below:

RCC_VREFCTRL.EN	RCC_VREFCTRL.HIM	Voltage Reference Buffer Level Selection
0	0	VREFBUF output disabled, V_{REF+} pin pulled down to VSSA
0	1	VREFBUF output in high-impedance state, V_{REF+} pin in external input

		mode
1	0	VREFBUF output enabled, V_{REF+} pin connected to VREFBUF output
1	1	VREFBUF output in high-impedance state, V_{REF+} pin voltage maintained by an external capacitor

Users can enable the operation of VREFBUF by first setting the RCC_VREFCTRL.HIM bit to 0 and then setting the RCC_VREFCTRL.EN bit to 1. At this point, users must wait for the RCC_VREFCTRL.RDY bit to be set, indicating that the voltage reference output has reached 90% of the expected value.

Notes:

1. When VREFBUF is operational, VREF- is connected to VSSA by default;

2. The input voltage of VREF+ must not exceed VDDA;

3. After enabling VREFBUF, the VREF+ pin should not be externally connected to a voltage;

4. When using the VREFBUF output, configure RCC_VREFCTRL.HIM = 0 first;

5. The 100nF filtering capacitor is crucial and should be placed as close as possible to the VREF+ pin in PCB design.

4.2.14.3 VREFBUF trim

The output voltage of VREFBUF is calibrated by the chip during manufacturing. When restarting VREFBUF or each time the voltage level of VREFBUF is changed (RCC_VREFCTRL.VLSEL[1:0]), the calibration data is automatically loaded into the RCC_VREFCTRL.TRIMDAT[5:0] register. Users can choose to fine-tune the output voltage by directly changing the bits of the RCC_VREFCTRL.TRIMDATA[5:0] register, and then enable the new trimming data by setting the RCC_VREFCTRL.TRIMEN. In this case, the voltage level setting of VREFBUF no longer affects the TRIM register unless the device is reset.

4.3 RCC Registers

The RCC registers are accessible through AHB bus. The register descriptions are as follows.

4.3.1 RCC Register Overview

Table 4-2 RCC Register Overview

Off set	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
00h	RCC_CTRL	Reserved					HSITRIM[6:0]							HSICAL[11:0]												HSECSSEN	HSEBP	HSERDF	HSEEN	HSIRDF	HSIEN	PLLRDF	PLLEN
	Reset Value						1	0	0	0	0	0	0	0	1	0	1	0	1	1	1	1	1	0	0	0	0	0	0	1	1	0	0
04h	RCC_CFG	MCOPLLSEL[1:0]		Reserved		SCLKSW[1:0]		SCLKSTS[1:0]		MCOPRES[3:0]				AHBPRES[3:0]				Reserved	APB2PRES[2:0]			APB1PRES[2:0]			PLLMULFCT[5:0]						PLLHSIPRES	PLLHSEPRES	PLLSRC
	Reset Value	0	0			0	0	0	0	0	0	1	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
08h	RCC_CFG2	Reserved			ATIMCLKSEL	GTIMCLKSEL	HCLKPRES	Reserved	RNGCPRES[4:0]					ADC1MSEL	Reserved		ADC1MPRES[4:0]					Reserved			ADCOTGPLLEN	ADCOTGPLLSEL	ADCOTGPLLPRES[2:0]			ADCHPRES[3:0]			
	Reset Value				0	0	0		0	0	0	0	0	0			0	0	1	1	1				0	0	0	0	0	0	0	0	0
0Ch	RCC_CFG3	FDCANPLLPRES[3:0]				I2S3SEL[1:0]		I2S2SEL[1:0]		FDCANSEL[1:0]		TRACEPRES[1:0]		USBPLLPRES[3:0]				USBPLLSEL	RPDMAEN	RPINTEN	RPINTF	USBFSSEL[1:0]		USBFSTM	TRNG1MPRES[3:0]				Reserved		TRNG1MEN	TRNG1MSEL	BORRSTEN
	Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1			0	0	1
10h	RCC_CLKINT	Reserved		HSICSSIF	HSICSSF	HSECSSF	LSECSSF	HSICSSIEN	HSICSSICLR	LSECSSICLR	BORICLR	LSERDICLR	LSIRDICLR	HSECSSICLR	HSERDICLR	HSIRDICLR	PLLRDICLR	LSECSSIEN	BORIEN	LSERDIEN	LSIRDIEN	HSECSSIEN	HSERDIEN	HSIRDIEN	PLLRDIEN	LSECSSIF	BORIF	LSERDIF	LSIRDIF	HSECSSIF	HSERDIF	HSIRDIF	PLLRDIF
	Reset Value			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14h	RCC_CTRLSTS	LPWRRSTF	WWDGRSTF	IWDGRSTF	SFTRSTF	PORRSTF	PINRSTF	SMPURSTF	Reserved	RAMERSTF	Reserved	BKPEMCF	MAINEMCF	BORRSTF	Reserved															RMRSTF	Reserved	LSIRD	LSIEN
	Reset Value	0	0	0	0	1	1	0		0		0	0	0																0		1	1
18h	RCC_AHBPCLKEN	Reserved													SHRTIMEN	ATIM3EN	ATIM2EN	ATIM1EN	DVPEN	CORDICEN	FMACEN	USBHSEN	ETHEN	CRCEN	DMA2EN	DMA1EN	SDIOEN	XSPIEN	RNGCEN	BKPSRAMEN	FEMCEN	FLITFEN	SRAMEN
	Reset Value														0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1Ch	RCC_APB1PCLKEN	DAC34EN	DAC12EN	WWDGEN	I2C4EN	I2C3EN	FDCAN3OFFEN	FDCAN2OFFEN	FDCAN1OFFEN	BTIM2EN	BTIM1EN	FDCAN3EN	FDCAN2EN	FDCAN1EN	USBFSEN	UCDREN	GTIM7EN	GTIM6EN	GTIM5EN	GTIM4EN	GTIM3EN	GTIM2EN	GTIM1EN	SPI3EN	SPI2EN	I2C2EN	I2C1EN	UART5EN	UART8EN	USART3EN	USART2EN	BKPEN	PWREN
	Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20h	RCC_APB2PCLKEN	Reserved																	USART4EN	COMPFILTEN	PGAEN	COMPEN	GTIM10EN	GTIM9EN	GTIM8EN	UART7EN	UART6EN	USART1EN	SPI1EN	SPI4EN	SPI5EN	SPI6EN	AFIOEN
	Reset Value																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
24h	RCC_AHBPRST	ETHRST	USBHSRST	FMACRST	CORDICRST	ATIM3RST	ATIM2RST	ATIM1RST	DVPRST	DAC78RST	DAC56RST	ADC4RST	ADC3RST	ADC2RST	ADC1RST	IOPHRST	IOPGRST	IOPFRST	IOPERST	IOPDRST	IOPCRST	IOPBRST	IOPARST	Reserved	FEMCRST	SDIORST	USBHSPHYRST	XSPIRST	RNGCRST	SACRST	SHRTPLLPHARST	SHRTPLLRST	SHRTIMRST
	Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		0	0	1	0	0	0	0	0	0
28h	RCC_APB1PRST	DAC34RST	DAC12RST	WWDGRST	I2C4RST	I2C3RST	Reserved			BTIM2RST	BTIM1RST	FDCAN3RST	FDCAN2RST	FDCAN1RST	USBFSRST	UCDRRST	GTIM7RST	GTIM6RST	GTIM5RST	GTIM4RST	GTIM3RST	GTIM2RST	GTIM1RST	SPI3RST	SPI2RST	I2C2RST	I2C1RST	UART5RST	UART8RST	USART3RST	USART2RST	BKPRST	PWRRST
	Reset Value	0	0	0	0	0				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2Ch	RCC_APB2PRST	Reserved																	USART4RST	Reserved	PGARST	COMPRST	GTIM10RST	GTIM9RST	GTIM8RST	UART7RST	UART6RST	USART1RST	SPI1RST	SPI4RST	SPI5RST	SPI6RST	AFIORST
	Reset Value																		0		0	0	0	0	0	0	0	0	0	0	0	0	0

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
30h	RCC_MCOCFG	Reserved																								MCO2[3:0]				MCO1[3:0]			
	Reset Value																									0	0	0	0	0	0	0	0
34h	RCC_BDCTRL	Reserved		LPTIM2RST	LPTIM1RST	LPTIM2EN	LPTIM2SEL[2:0]			LPTIM1EN	LPTIM1SEL[2:0]			Reserved			BDSFTRST	RTCEN	Reserved					RTCSEL[1:0]		Reserved				LSECSSEN	LSEBP	LSERD	LSEEN
	Reset Value			0	0	0	0	0	0	0	0	0	0				0	0						0	0					0	0	0	0
38h	RCC_CSSCTRL	Reserved									HSICSSEN	Reserved	WKUPCLKSEL	Reserved	HSEPOF	HSEFHF	HSEFLF	HSEPOEN	HSEFHEN	HSEFLEN	HSEDLSEL	HSEPOTHR[3:0]				HSEFHTHR[3:0]				HSEFLTHR[3:0]			
	Reset Value										0		0		0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	1	0	0	0
3Ch	RCC_AHB1PCLKEN	Reserved																SACEN	DAC78EN	DAC56EN	ADC4EN	ADC3EN	ADC2EN	ADC1EN	IOFILTSEL	IOPHEN	IOPGEN	IOPFEN	IOPEEN	IOPDEN	IOPCEN	IOPBEN	IOPAEN
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
40h	RCC_PLLCTRL	Reserved										SCLKPLLSEL[1:0]		PLLOUTPRES[1:0]		PLLINPRES[1:0]		DEADZEN	LPFC2[2:0]			LPFR[3:0]				LPFC1[3:0]				Reserved			
	Reset Value											0	0	0	1	0	0	0	1	0	1	0	1	1	0	1	0	0	0				
44h	RCC_VREFCTRL	Reserved																	TRIMEN	Reserved	RDY	Reserved		HIM	EN	VLSEL[1:0]		TRIMDATA[5:0]					
	Reset Value																		0		0			0	0	0	0	0	1	1	1	1	1
48h	RCC_SHRTPLLCTRL1	Reserved	SATEN	Reserved	SHRTPLLMODE	FRACEN	Reserved	CLKF[25:0]																									
	Reset Value		0		1	0		0	0	0	0	0	1	1	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4Ch	RCC_SHRTPLLCTRL2	CLKR[5:0]						Reserved																									
	Reset Value	0	0	0	0	0	0																										
50h	RCC_SHRTPLLCTRL3	Reserved			SHRTPLLRDF	Reserved	SHRTPLLSRC	SHRTPLLEN	Reserved	BWTRIM[11:0]												Reserved											
	Reset Value				0		0	0		0	0	0	0	0	0	1	1	1	1	0	1												
70h	RCC_SRAMCFG1	Reserved		CSRECCF[1:0]		Reserved	SR2PEF	Reserved	SR1PEF	FECCRSTEN	CSRECCCLR	SR2PECLR	SR1PECLR	CSRECCRSTEN[1:0		CSRECCIEN[1:0]		Reserved	SR2PERSTEN	Reserved	SR2PEIEN	Reserved	SR1PERSTEN	Reserved	SR1PEIEN	Reserved	SRAMFIN	SRAMWEF	SRAMREF	CSRINIF	CSRINIEN	Reserved	SRAMPEN
	Reset Value			0	0		0		0	0	0	0	0	0	0	0	0		0		0		0		0		0	0	0	0	0		1

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
74h	RCC_SRAMCFG2	INIDAT[31:0]																															
	Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
78h	RCC_SRAMCFG3	INISTART[31:0]																															
	Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7Ch	RCC_SRAMCFG4	INIEND[31:0]																															
	Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
84h	RCC_USBHSCTRL1	Reserved	RXTXPATH	BSTHEN	BSTLEN	Reserved				USBHSSEL	SEBP	Reserved		RESSEL	Reserved			PDSEL	Reserved			PLLBW	USBHSFSEL			Reserved			WKUPSEL	Reserved			
	Reset Value		0	0	0					0	0			1				1				0	0	0	0				0				
88h	RCC_USBHSCTRL2	Reserved								RESTRIM				IDDQEN	Reserved			IMPTRIM		RFTRIM		CROSTRIM		DATAVTRIM		HDCVTRIM				FLIMPTRIM			
	Reset Value									1	0	0	0	0				0	1	0	1	1	1	0	1	0	0	1	1	0	0	1	1
8Ch	RCC_BOOTREMAP	Reserved																REMAPKEY[7:0]								Reserved						REMAPSEL[2:0]	
	Reset Value																	0	0	0	0	0	0	0	0							0	0
90h	RCC_FDCANBADDR	BADDR[31:0]																															
	Reset Value	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

4.3.2 Clock Control Register (RCC_CTRL)

Address offset: 0x00

Reset value: 0x84057C0C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved					HSITRIM[6:0]							HSICAL[11:0]			
					rw							r			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HSICAL[11:0]								HSECSSEN	HSEBP	HSERDF	HSEEN	HSIRDF	HSIEN	PLLRDF	PLLEN
r								rw	rw	r	rw	r	rw	r	rw

Bit Field	Name	Description
31:27	Reserved	Reserved, the reset value must be maintained.
26:20	HSITRIM[6:0]	Internal High-Speed Clock Correction Value: Written by software. The value of these

Bit Field	Name	Description
		bits will be added to the HSICAL[11:0] bits to form the final value for calibrating the internal HSI RC oscillator frequency. The default value is 64, with an adjustment step of 5KHz.
19:8	HSICAL[11:0]	Internal High-Speed Clock Calibration Value These bits are automatically initialized at power-up.
7	HSECSSEN	HSE clock security system enable Set and cleared by software. 0: Disable the clock detector 1: Enable the clock detector if the HSE oscillator is ready
6	HSEBP	External high-speed clock bypass enable Set and cleared by software. This bit can only be written when the HSE oscillator is disabled. 0: Disable the bypass function of HSE oscillator 1: Enable the bypass function of HSE oscillator
5	HSERDF	External high-speed clock ready flag Set by hardware once HSE is ready. This bit takes 6 HSE clock cycles to clear after the HSEEN bit is cleared. 0: HSE is not ready 1: HSE is ready
4	HSEEN	External high-speed clock enable Set and cleared by software. When entering the stop2 or standby mode, it is cleared by hardware. This bit cannot be cleared when HSE is used directly or indirectly as the system clock. 0: Disable HSE oscillator 1: Enable HSE oscillator
3	HSIRDF	Internal High-Speed Clock Ready Flag The flag is set by hardware after HSI is ready. After clearing the HSIEN bit, it takes 6 internal 8MHz oscillator clock cycles to clear this flag. 0: HSI not ready 1: HSI ready
2	HSIEN	Internal High-Speed Clock Enable The internal high-speed clock enable bit is set and cleared by software. This bit cannot be cleared when HSI is used as the system clock. When returning from STOP0 or STANDBY mode or in case of HSE failure, the hardware sets this bit to enable the HSI oscillator. If HSI is used directly or indirectly as the system clock, this bit cannot be reset. 0: Disable HSI oscillator 1: Enable HSI oscillator
1	PLLRDF	PLL Clock Ready Flag The PLL clock ready flag is set by hardware after the PLL clock is ready. 0: PLL not ready 1: PLL ready

Bit Field	Name	Description
0	PLLEN	PLL enable Set and cleared by software. It is cleared by hardware when entering STOP0/STANDBY mode. When the PLL is used as the system clock, this bit cannot be cleared. The PLL will not be enabled when HSI/HSE is used as the clock source for the PLL until the HSI/HSE clock is ready. 0: Disable PLL 1: Enable PLL

4.3.3 Clock Configuration Register (RCC_CFG)

Address offset: 0x04

Reset value: 0x00200004

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MCOPLLSEL[1:0]		Reserved		SCLKSW[1:0]		SCLKSTS[1:0]		MCOPRES[3:0]				AHBPRES[3:0]			
rw				rw		r		rw				rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	APB2PRES[2:0]			APB1PRES[2:0]			PLLMULFCT[5:0]						PLLHSIPRES	PLLHSEPRES	PLLSRC
	rw			rw			rw						rw	rw	rw

Bit Field	Name	Description
31:30	MCOPLLSEL[1:0]	MCO Output PLL Selection x0: Select PLL clock as MCO output source 01: Select SHRTPLL clock as MCO output source 11: Select USBHS 240M clock as MCO output source
29:28	Reserved	Reserved, the reset value must be maintained.
27:26	SCLKSW[1:0]	System Clock Switching: Set and cleared by software to select the system clock source. When exiting STOP0 or STANDBY mode, or in case of HSE oscillator failure (RCC_CTRL.HSECSSEN enabled), hardware sets it to force the selection of HSI. 00: Select HSI as system clock 01: Select HSE as system clock 10: Select PLL as system clock (PLL, SHRTPLL, USBHS240M can be selected) 11: Not available *Note: When selecting the PLL clock as the system clock source, further selection of PLL, SHRTPLL, or USBHS240M as the system clock can be made through RCC_PLLCTRL.SCLKPLLSEL (default is PLL).*
25:24	SCLKSTS[1:0]	System Clock Switching Status: Set and cleared by hardware to indicate which clock source is being used as the system clock. 00: System clock from HSI

		01: System clock from HSE 10: System clock from PLL, SHRTPLL, or USBHS240M 11: Not available
23:20	MCOPRES[3:0]	MCO Prescaler. Software sets or clears. 0010: PLL clock divided by 2 as MCO clock 0011: PLL clock divided by 3 as MCO clock 0100: PLL clock divided by 4 as MCO clock 0101: PLL clock divided by 5 as MCO clock 0110: PLL clock divided by 6 as MCO clock 0111: PLL clock divided by 7 as MCO clock 1000: PLL clock divided by 8 as MCO clock 1001: PLL clock divided by 9 as MCO clock 1010: PLL clock divided by 10 as MCO clock 1011: PLL clock divided by 11 as MCO clock 1100: PLL clock divided by 12 as MCO clock 1101: PLL clock divided by 13 as MCO clock 1110: PLL clock divided by 14 as MCO clock 1111: PLL clock divided by 15 as MCO clock Other values: Not allowed to be set
19:16	AHBPRES[3:0]	AHB Prescaler: Set and cleared by software to configure the division factor of the AHB clock (HCLK). 0xxx: SYSCLK not divided 1000: SYSCLK divided by 2 1001: SYSCLK divided by 4 1010: SYSCLK divided by 8 1011: SYSCLK divided by 16 1100: SYSCLK divided by 64 1101: SYSCLK divided by 128 1110: SYSCLK divided by 256 1111: SYSCLK divided by 512
15	Reserved	Reserved, the reset value must be maintained.
14:12	APB2PRES[2:0]	APB high-speed (APB2) prescaler Set and cleared by software to configure the division factor of APB2 clock (PCLK2). Make sure that PCLK2 does not exceed 180 MHz, and the frequency is not the same as AHB. 0xx: HCLK not divided 100: HCLK divided by 2 101: HCLK divided by 4 110: HCLK divided by 8 111: HCLK divided by 16
11:9	APB1PRES[2:0]	APB low-speed (APB1) prescaler

		Set and cleared by software to configure the division factor of APB1 clock (PCLK1). Make sure that PCLK1 does not exceed 180MHz. 0xx: HCLK not divided 100: HCLK divided by 2 101: HCLK divided by 4 110: HCLK divided by 8 111: HCLK divided by 16
8:3	PLLMULFCT[5:0]	PLL multiplication factor Written by software to define PLL multiplication factor. These bits can only be written when the PLL is disabled. Note: It is recommended that the octave factor not be less than 8 000000:PLL input clock×4 000001:PLL input clock×6 000010:PLL input clock×8 000011:PLL input clock×10 000100:PLL input clock×12 000101:PLL input clock×14 000110:PLL input clock×16 000111:PLL input clock×18 001000:PLL input clock×20 001001:PLL input clock×22 001010:PLL input clock×24 001011:PLL input clock×26 001100:PLL input clock×28 001101:PLL input clock×30 001110:PLL input clock×32 001111:PLL input clock×34 010000:PLL input clock×36 010001:PLL input clock×38 010010:PLL input clock×40 010011:PLL input clock×42 010100:PLL input clock×44 010101:PLL input clock×46 010110:PLL input clock×48 010111:PLL input clock×50 011000:PLL input clock×52 011001:PLL input clock×54 011010:PLL input clock×56 011011:PLL input clock×58 011100:PLL input clock×60 011101:PLL input clock×62 011110:PLL input clock×64 011111:PLL input clock×66

		100000:PLL input clock×68 100001:PLL input clock×70 100010:PLL input clock×72 100011:PLL input clock×74 100100:PLL input clock×76 Other value：disabled
2	PLLHSIPRES	PLL Input HSI Prescaler: Set and cleared by software to configure the HSI division before entering the PLL. This bit can only be written when the PLL is disabled. 0: HSI clock not divided 1: HSI clock divided by 2
1	PLLHSEPRES	PLL Input HSE Prescaler: Set and cleared by software to configure the HSE division before entering the PLL. This bit can only be written when the PLL is disabled. 0: HSE clock not divided 1: HSE clock divided by 2
0	PLLSRC	PLL Clock Source: Set and cleared by software to configure the selection of the PLL clock source. This bit can only be written when the PLL is disabled. 0: Select HSI (or divided by 2) clock as the PLL input clock 1: Select HSE (or divided by 2) clock as the PLL input clock

4.3.4 Clock Configuration Register2 (RCC_CFG2)

Address offset: 0x08

Reset value: 0x00007000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved			ATIMCLKSEL	GTIMCLKSEL	HCLKPRES	Reserved	RNGCPRES[4:0]					ADC1MSEL	Reserved		ADC1MPRES[4:0]
			rw	rw	rw		rw					rw			rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADC1MPRES[4:0]				Reserved			ADCOTGPLLEN	ADCOTGPLLSEL	ADCOTGPLLPRES[2:0]			ADCHPRES[3:0]			
rw							rw	rw	rw			rw			

Bit Field	Name	Description
31:29	Reserved	Reserved, the reset value must be maintained.
28	ATIMCLKSEL	ATIM1/2/3 Clock Source Selection Set and cleared by software. 0: Select SYSCLK as the ATIM1/2/3 clock source 1: Select HCLK as the ATIM1/2/3 clock source
27	GTIMCLKSEL	GTIM8/9/10 Clock Source Selection

Bit Field	Name	Description
		Set and cleared by software. 0: If APB2 prescaler is 1, select PCLK2 as the GTIM8/9/10 clock source. Otherwise, select PCLK2 x 2 1: Select SYSCLK as the GTIM8/9/10 clock source
26	HCLKPRES	SAC/SDIO/USBHS/ETH Access Clock HCLK Prescaler Set and cleared by software to configure the prescaler factor for SAC_HCLK/SDIO_HCLK/USBHS_HCLK/ETH_HCLK. 0: No division of HCLK 1: HCLK divided by 2 *Note: Must be configured to 0 when using SAC; When using SDIO/USBHS/ETH, set it to 1 when HCLK is greater than 180M, otherwise set it to 0*
25	Reserved	Reserved, the reset value must be maintained.
24:20	RNGCPRES[4:0]	RNGC Prescaler. These bits are set or cleared by software to configure the prescaler factor for the RNGC clock. 00000: No division of SYSCLK 00001: SYSCLK divided by 2 00010: SYSCLK divided by 3 ... 11110: SYSCLK divided by 31 11111: SYSCLK divided by 32
19	ADC1MSEL	ADC 1M Clock Source Selection. Set or cleared by software. 0: Select HSI oscillator clock as the input clock for ADC 1M 1: Select HSE oscillator clock as the input clock for ADC 1M
18:17	Reserved	Reserved, the reset value must be maintained.
16:12	ADC1MPRES[4:0]	ADC 1M Clock Divider. Set or clear these bits by software to configure the prescaler factor for the ADC1M clock source. 00000: No division of the ADC 1M clock source 00001: ADC 1M clock source divided by 2 00010: ADC 1M clock source divided by 3 ... 11110: ADC 1M clock source divided by 31 11111: ADC 1M clock source divided by 32 *Note: The clock for the ADC must be configured as 1M*
11:9	Reserved	Reserved, the reset value must be maintained.
8	ADCOTGPLLEN	ADC USBHS 240M or PLL Enable: When selecting the ADC working clock source as USBHS 240M or PLL clock through the RCC_CFG.ADCOTGPLLSEL bit, use this bit to enable or disable it. 0: ADC USBHS 240M or PLL clock disabled 1: ADC USBHS 240M or PLL clock enabled

Bit Field	Name	Description
7	ADCOTGPLLSEL	ADC Clock Source USBHS 240M or PLL Selection: When selecting the ADC working clock source as PLL through the ADC_CTRL3.CKMOD bit, use this bit to further specify the specific PLL clock source. 0: Select PLL clock as the source for division 1: Select USBHS 240M clock as the source for division
6:4	ADCOTGPLLPRES[2:0]	ADC USBHS 240M or PLL Prescaler: When selecting the ADC working clock source as USBHS 240M or PLL clock through the RCC_CFG.ADCOTGPLLSEL bit, further configure the specific division factor using this bit. 000: No division of USBHS 240M or PLL clock 001: USBHS 240M or PLL clock divided by 2 010: USBHS 240M or PLL clock divided by 3 011: USBHS 240M or PLL clock divided by 4 100: USBHS 240M or PLL clock divided by 6 101: USBHS 240M or PLL clock divided by 8 110: USBHS 240M or PLL clock divided by 10 111: USBHS 240M or PLL clock divided by 12
3:0	ADCHPRES[3:0]	ADC HCLK Prescaler: These bits are set and cleared by software to configure the division factor of the HCLK clock to the ADC. When the ADC working clock source is selected as the HCLK clock through the ADC_CTRL.CKMOD bit, further configure the division factor using this bit. 0000: No division of the HCLK clock 0001: HCLK clock divided by 2 0010: HCLK clock divided by 3 0011: HCLK clock divided by 4 0100: HCLK clock divided by 6 0101: HCLK clock divided by 8 0110: HCLK clock divided by 10 0111: HCLK clock divided by 12 1000: HCLK clock divided by 16 Others: HCLK clock divided by 32

4.3.5 Clock Configuration Register3 (RCC_CFG3)

Address offset: 0x0C

Reset value: 0xC00C0071

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FDCANPLLPRES[3:0]				I2S3SEL[1:0]		I2S2SEL[1:0]		FDCANSEL[1:0]		Reserved		USBPLLPRES[3:0]			
rw				rw		rw		rw				rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USBPLLSEL	RPDMAEN	RPINTEN	RPINTF	USBFSSEL[1:0]		USBFSTM	TRNG1MPRES[3:0]				Reserved		TRNG1MEN	TRNG1MSEL	BORRSTEN
rw	rw	rw	r	rw		rw	rw						rw	rw	rw

Bit Field	Name	Description
31:28	FDCANPLLPRES[3:0]	FDCAN1/2/3 PLL Clock Source Prescaler 0000: Not used 0001: PLL or SHRTPLL clock divided by 1 0010: PLL or SHRTPLL clock divided by 2 0011: PLL or SHRTPLL clock divided by 3 ... 1110: PLL or SHRTPLL clock divided by 14 1111: PLL or SHRTPLL clock divided by 15
27:26	I2S3SEL[1:0]	I2S3 Clock Source Selection The I2S3 clock source selection must not exceed 180MHz 00: Select SYSCLK as the input clock 01: Select SHRTPLL clock as the input clock 10: Select I2S_CLKIN pin clock as the input clock 11: Select HSI clock as the input clock
25:24	I2S2SEL[1:0]	I2S2 Clock Source Selection The I2S2 clock source selection must not exceed 180M. 00: Select SYSCLK as the input clock 01: Select SHRTPLL clock as the input clock 0: Select I2S_CLKIN pin clock as the input clock 11: Select HSI clock as the input clock
23:22	FDCANSEL[1:0]	FDCAN1/2/3 Clock Source Selection 00: Select PLL divided clock as the input clock 01: Select SHRTPLL divided clock as the input clock 1x: Select HSE clock as the input clock
21:20	Reserved	Reserved, the reset value must be maintained.
19:16	USBPLLPRES[3:0]	USB PLL clock source predivider 0000: Not used 0001: PLL or SHRTPLL clock divided by 1 0010: PLL or SHRTPLL clock divided by 2 0011: PLL or SHRTPLL clock divided by 3 ... 1110: PLL or SHRTPLL clock divided by 14 1111: PLL or SHRTPLL clock divided by 15
15	USBPLLSEL	USB PLL divider clock source selection When the USBHS or USBFS clock source is selected as the PLL divider clock, this bit can further select the PLL clock source as the PLL clock or SHRTPLL clock. 0: Select PLL clock 1: Select SHRTPLL clock
14	RPDMAEN	DMA access permission to Flash during Row programming 0: DMA can access Flash space

Bit Field	Name	Description
		1: DMA is prohibited from accessing Flash space
13	RPINTEN	Row programming interrupt flag enable 0: When a row programming interrupt event occurs, the corresponding flag bit RCC_CFG3.RPINTF is not set to 1 1: When a row programming interrupt event occurs, the corresponding flag bit RCC_CFG3.RPINTF is set to 1
12	RPINTF	Row programming interrupt flag. This bit is set to 1 by hardware when RCC_CFG3.RPINTEN = 1 and a row programming interrupt event occurs, and is cleared by setting RCC_CFG3.RPINTEN to 0. 0: No row programming interrupt event has occurred 1: A row programming interrupt event has occurred
11:10	USBFSSEL[1:0]	USBFS clock source selection The USBFS clock source selection must be configured as 48M. 00: Select UCDR clock 01: Select PLL divider clock 1x: Select USBHS 48M clock *Note: When selecting the PLL divider clock, the PLL clock source can be further selected as the PLL clock or SHRTPLL clock through RCC_CFG3.USBPLLSEL.*
9	USBFSTM	USBFS 48M timing mode selection 0: Optimize timing when the PCLK clock frequency of USBFS is less than 120M 1: Optimize timing when the PCLK clock frequency of USBFS is greater than or equal to 120M
8:5	TRNG1MPRES[3:0]	TRNG 1M clock predivider. These bits are set or cleared by software to generate the TRNG 1M clock. 0000: TRNG 1M clock source divided by 2 0001: TRNG 1M clock source divided by 4 0010: TRNG 1M clock source divided by 6 0011: TRNG 1M clock source divided by 8 0100: TRNG 1M clock source divided by 10 ... 1101: TRNG 1M clock source divided by 28 1110: TRNG 1M clock source divided by 30 1111: TRNG 1M clock source divided by 32
4:3	Reserved	Reserved, the reset value must be maintained.
2	TRNG1MEN	TRNG analog interface clock enable. Set or clear by software. 0: Disable TRNG analog interface clock 1: Enable TRNG analog interface clock
1	TRNG1MSEL	TRNG 1M clock selection. Set or clear by software. 0: Select HSI oscillator as the TRNG 1M input clock

Bit Field	Name	Description
		1: Select HSE oscillator as the TRNG 1M input clock
0	BORRSTEN	BOR reset enable. Set or clear by software. 0: BOR reset disabled 1: BOR reset enabled

4.3.6 Clock Interrupt Register (RCC_CLKINT)

Address offset: 0x10

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		HSICSSIF	HSICSSF	HSECSSF	LSECSSF	HSICSSIEN	HSICSSICLR	LSECSSICLR	BORICLR	LSERDICLR	LSIRDICLR	HSECSSICLR	HSERDICLR	HSIRDICLR	PLLRDICLR
		r	r	r	r	rw	w	w	w	w	w	w	w	w	w

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LSECSSIEN	BORIEN	LSERDIEN	LSIRDIEN	HSECSSIEN	HSERDIEN	HSIRDIEN	PLLRDIEN	LSECSSIF	BORIF	LSERDIF	LSIRDIF	HSECSSIF	HSERDIF	HSIRDIF	PLLRDIF
rw	rw	rw	rw	rw	rw	rw	rw	r	r	r	r	r	r	r	r

Bit Field	Name	Description
31:30	Reserved	Reserved, the reset value must be maintained.
29	HSICSSIF	HSI clock security system interrupt flag This bit is set by hardware when HSICSSIEN is set and a fault is detected in the external HSI oscillator. Software clears this bit by setting the HSICSSICLR bit. 0: No clock security system interrupt caused by HSI clock failure 1: Clock security system interrupt caused by HSI clock failure
28	HSICSSF	HSI clock security system flag This bit is set by hardware when a fault is detected in the external HSI oscillator. 0: No clock security system fault caused by HSI clock failure 1: Clock security system fault caused by HSI clock failure
27	HSECSSF	HSE clock security system flag This bit is set by hardware when a fault is detected in the external HSE oscillator. 0: No clock security system fault caused by HSE clock failure 1: Clock security system fault caused by HSE clock failure
26	LSECSSF	LSE clock security system flag This bit is set by hardware when a fault is detected in the external LSE oscillator. 0: No clock security system fault caused by LSE clock failure 1: Clock security system fault caused by LSE clock failure
25	HSICSSIEN	HSI clock security system interrupt enable Set and clear by software to enable or disable the clock security system interrupt.

		0: Disable HSI clock security system interrupt 1: Enable HSI clock security system interrupt
24	HSICSSICLR	HSI clock security system interrupt clear Set by software to clear the HSICSSIF flag. 0: No effect 1: Clear the HSICSSIF flag
23	LSECSSICLR	LSE clock security system interrupt clear Set by software to clear the LSECSSIF flag. 0: No effect 1: Clear the LSECSSIF flag
22	BORICLR	BOR interrupt clear Set by software to clear the BORIF flag. 0: No effect 1: Clear the BORIF flag
21	LSERDICLR	LSE ready interrupt clear Set by software to clear the LSERDIF flag. 0: No effect 1: Clear the LSERDIF flag
20	LSIRDICLR	LSI ready interrupt clear Set by software to clear the LSIRDIF flag. 0: No effect 1: Clear the LSIRDIF flag
19	HSECSSICLR	HSE clock security system interrupt clear Set by software to clear the HSECSSIF flag. 0: No effect 1: Clear the HSECSSIF flag
18	HSERDICLR	HSE ready interrupt clear Set by software to clear the HSERDIF flag. 0: No effect 1: Clear the HSERDIF flag
17	HSIRDICLR	HSI ready interrupt clear Set by software to clear the HSIRDIF flag. 0: No effect 1: Clear the HSIRDIF flag
16	PLLRDICLR	PLL ready interrupt clear Set by software to clear the PLLRDIF flag. 0: No effect 1: Clear the PLLRDIF flag
15	LSECSSIEN	LSE clock security system interrupt enable Set and clear by software to enable or disable the clock security system interrupt. 0: Disable LSE clock security system interrupt 1: Enable LSE clock security system interrupt

14	BORIEN	BOR interrupt enable Set and clear by software to enable or disable the BOR interrupt. 0: Disable BOR interrupt 1: Enable BOR interrupt
13	LSERDIEN	LSE ready interrupt enable Set and clear by software to enable or disable the LSE ready interrupt. 0: Disable LSE ready interrupt 1: Enable LSE ready interrupt
12	LSIRDIEN	LSI ready interrupt enable Set and clear by software to enable or disable the LSI ready interrupt. 0: Disable LSI ready interrupt 1: Enable LSI ready interrupt
11	HSECSSIEN	HSE clock security system interrupt enable Set and clear by software to enable or disable the clock security system interrupt. 0: Disable HSE clock security system interrupt 1: Enable HSE clock security system interrupt
10	HSERDIEN	HSE ready interrupt enable Set and clear by software to enable or disable the HSE ready interrupt. 0: Disable HSE ready interrupt 1: Enable HSE ready interrupt
9	HSIRDIEN	HSI ready interrupt enable Set and clear by software to enable or disable the HSI ready interrupt. 0: Disable HSI ready interrupt 1: Enable HSI ready interrupt
8	PLLRDIEN	PLL ready interrupt enable Set and clear by software to enable or disable the PLL ready interrupt 0: Disable PLL ready interrupt 1: Enable PLL ready interrupt
7	LSECSSIF	LSE clock security system interrupt flag This bit is set by hardware when LSECSSIEN is set and a fault is detected in the external LSE oscillator. Software clears this bit by setting the LSECSSICLR bit. 0: No clock security system interrupt caused by LSE clock failure 1: Clock security system interrupt caused by LSE clock failure
6	BORIF	BOR interrupt flag This bit is set by hardware when BORIEN is set and a BOR occurs. Software clears this bit by setting the BORICLR bit. 0: No BOR reset interrupt occurred 1: BOR reset interrupt occurred
5	LSERDIF	LSE ready interrupt flag This bit is set by hardware when LSERDIEN is set and the LSE clock is ready. Software clears this bit by setting the LSERDICLR bit. 0: No clock ready interrupt caused by the LSE oscillator 1: Clock ready interrupt caused by the LSE oscillator

4	LSIRDIF	LSI ready interrupt flag This bit is set by hardware when LSIRDIEN is set and the LSI clock is ready. Software clears this bit by setting the LSIRDICLR bit. 0: No clock ready interrupt caused by the LSI oscillator 1: Clock ready interrupt caused by the LSI oscillator
3	HSECSSIF	HSE clock security system interrupt flag This bit is set by hardware when HSECSSIEN is set and a fault is detected in the external HSE oscillator. Software clears this bit by setting the HSECSSICLR bit. 0: No clock security system interrupt caused by HSE clock failure 1: Clock security system interrupt caused by HSE clock failure
2	HSERDIF	HSE ready interrupt flag This bit is set by hardware when HSERDIEN is set and the HSE clock is ready. Software clears this bit by setting the HSERDICLR bit. 0: No clock ready interrupt caused by the HSE oscillator 1: Clock ready interrupt caused by the HSE oscillator
1	HSIRDIF	HSI ready interrupt flag This bit is set by hardware when HSIRDIEN is set and the HSI clock is ready. Software clears this bit by setting the HSIRDICLR bit. 0: No clock ready interrupt caused by the HSI oscillator 1: Clock ready interrupt caused by the HSI oscillator
0	PLLRDIF	PLL ready interrupt flag This bit is set by hardware when PLLRDIEN is set and the PLL clock is ready. Software clears this bit by setting the PLLRDICLR bit. 0: No clock ready interrupt caused by PLL lock 1: Clock ready interrupt caused by PLL lock

4.3.7 Control/Status Register (RCC_ CTRLSTS)

Address offset: 0x14

Reset value: 0x0C000003

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LPWRRSTF	WWDGRSTF	IWDGRSTF	SFTRSTF	PORRSTF	PINRSTF	SMPURSTF	Reserved	RAMERSTF	Reserved	BKPEMCF	MAINEMCF	BORRSTF	Reserved		
r	r	r	r	r	r	r		r		r	r	r			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												RMRSTF	Reserved	LSIRD	LSIEN
												rw		r	rw

Bit Field	Name	Description
31	LPWRRSTF	Low power reset flag This bit is set by hardware when a low power management reset occurs.

Bit Field	Name	Description
		Software clears this bit by writing to the RMRSTF bit. 0: No low power management reset occurred 1: Low power management reset occurred
30	WWDGRSTF	Window watchdog reset flag This bit is set by hardware when a window watchdog reset occurs. Software clears this bit by writing to the RMRSTF bit. 0: No window watchdog reset occurred 1: Window watchdog reset occurred
29	IWDGRSTF	Independent watchdog reset flag This bit is set by hardware when an independent watchdog reset occurs. Software clears this bit by writing to the RMRSTF bit. 0: No independent watchdog reset occurred 1: Independent watchdog reset occurred
28	SFTRSTF	Software reset flag This bit is set by hardware when a software reset occurs. Software clears this bit by writing to the RMRSTF bit. 0: No software reset occurred 1: Software reset occurred
27	PORRSTF	Power-on/Power-down reset flag This bit is set by hardware when a power-on/power-off reset occurs. Software clears this bit by writing to the RMRSTF bit. 0: No power-on/power-off reset occurred 1: Power-on/power-off reset occurred
26	PINRSTF	External pin reset flag This bit is set by hardware when a reset occurs at the NRST pin. Software clears this bit by writing to the RMRSTF bit. 0: No reset occurred at the NRST pin 1: Reset occurred at the NRST pin
25	SMPURSTF	SMPU reset flag This bit is set by hardware when an SMPU reset occurs. Software clears this bit by writing to the RMRSTF bit. 0: No SMPU reset occurred 1: SMPU reset occurred
24	Reserved	Reserved, the reset value must be maintained.
23	RAMERSTF	RAM ECC or parity check error reset flag. This bit is set to 1 by hardware when a RAM ECC or parity check error reset occurs, and software clears it by writing to the RMRSTF bit. 0: No RAM check reset occurred 1: RAM check reset occurred
22	Reserved	Reserved, the reset value must be maintained.
21	BKPEMCF	Backup domain EMC reset flag. This bit is set to 1 by hardware when a backup domain EMC reset occurs, and

Bit Field	Name	Description
		software clears it by writing to the RMRSTF bit. 0: No backup domain EMC reset occurred 1: Backup domain EMC reset occurred
20	MAINEMCF	Main domain EMC reset flag. This bit is set to 1 by hardware when a main domain EMC reset occurs, and software clears it by writing to the RMRSTF bit. 0: No main domain EMC reset occurred 1: Main domain EMC reset occurred
19	BORRSTF	BOR reset flag. This bit is set to 1 by hardware when a BOR reset occurs, and software clears it by writing to the RMRSTF bit. 0: No BOR reset occurred 1: BOR reset occurred
18:4	Reserved	Reserved, the reset value must be maintained.
3	RMRSTF	Clear reset flags Software sets this bit to 1 to clear all reset flags. 0: No effect 1: Clear reset flags
2	Reserved	Reserved, the reset value must be maintained.
1	LSIRD	Internal low-speed oscillator ready Set and cleared by hardware to indicate whether the internal RC 32kHz oscillator is ready. After clearing LSIEN, LSIRD is cleared 3 internal RC 32kHz oscillator clock cycles later. 0: Internal 32kHz RC oscillator clock not ready 1: Internal 32kHz RC oscillator clock ready
0	LSIEN	Internal low-speed oscillator enable Set and cleared by software. 0: Disable internal RC 32kHz oscillator 1: Enable internal RC 32kHz oscillator

4.3.8 AHB Peripheral Clock Enable Register (RCC_AHBPCLKEN)

Address offset: 0x18

Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved													SHRTIMEN	ATIM3EN	ATIM2EN
													rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ATIM1EN	DVPEN	CORDICEN	FMACEN	USBHSEN	ETHEN	CRCEN	DMA2EN	DMA1EN	SDIOEN	XSPIEN	RNGCEN	BKPSRAMEN	FEMCEN	FLITFEN	SRAMEN
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit Field	Name	Description
31:19	Reserved	Reserved, the reset value must be maintained.
18	SHRTIMEN	SHRTIM clock enable Set and cleared by software. 0: Disable SHRTIM clock 1: Enable SHRTIM clock
17	ATIM3EN	ATIM3 clock enable Set and cleared by software. 0: Disable ATIM3 clock 1: Enable ATIM3 clock
16	ATIM2EN	ATIM2 clock enable Set and cleared by software. 0: Disable ATIM2 clock 1: Enable ATIM2 clock
15	ATIM1EN	ATIM1 clock enable Set and cleared by software. 0: Disable ATIM1 clock 1: Enable ATIM1 clock
14	DVPEN	DVP clock enable Set and cleared by software. 0: Disable DVP clock 1: Enable DVP clock
13	CORDICEN	CORDIC clock enable Set and cleared by software. 0: Disable CORDIC clock 1: Enable CORDIC clock
12	FMACEN	FMAC clock enable Set and cleared by software. 0: Disable FMAC clock 1: Enable FMAC clock
11	USBHSEN	USBHS clock enable Set and cleared by software. 0: Disable USBHS clock 1: Enable USBHS clock
10	ETHEN	ETH clock enable Set and cleared by software. 0: Disable ETH clock 1: Enable ETH clock
9	CRCEN	CRC clock enable Set and cleared by software. 0: Disable CRC clock 1: Enable CRC clock

Bit Field	Name	Description
8	DMA2EN	DMA2 clock enable Set and cleared by software. 0: Disable DMA2 clock 1: Enable DMA2 clock
7	DMA1EN	DMA1 clock enable Set and cleared by software. 0: Disable DMA1 clock 1: Enable DMA1 clock
6	SDIOEN	SDIO clock enable Set and cleared by software. 0: Disable SDIO clock 1: Enable SDIO clock
5	XSPIEN	XSPI clock enable Set and cleared by software. 0: Disable XSPI clock 1: Enable XSPI clock
4	RNGCEN	RNGC clock enable Set and cleared by software. 0: Disable RNGC clock 1: Enable RNGC clock
3	BKPSRAMEN	Backup SRAM clock enable Set and cleared by software. 0: Disable Backup SRAM clock 1: Enable Backup SRAM clock
2	FEMCEN	FEMC clock enable Set and cleared by software. 0: Disable FEMC clock 1: Enable FEMC clock
1	FLITFEN	Flash interface circuit clock enable. Set or cleared by software. 0: Disable flash interface circuit clock 1: Enable flash interface circuit clock
0	SRAMEN	SRAM clock enable In SLEEP mode, set and clear by software to enable/disable the SRAM clock. 0: SRAM clock disabled in SLEEP mode 1: SRAM clock enabled in SLEEP mode

4.3.9 APB1 Peripheral Clock Enable Register (RCC_ APB1PCLKEN)

Address offset: 0x1C

Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DAC34EN	DAC12EN	WWDGEN	I2C4EN	I2C3EN	FDCAN3OFFEN	FDCAN2OFFEN	FDCAN1OFFEN	BTIM2EN	BTIM1EN	FDCAN3EN	FDCAN2EN	FDCAN1EN	USBFSEN	UCDREN	GTIM7EN
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GTIM6EN	GTIM5EN	GTIM4EN	GTIM3EN	GTIM2EN	GTIM1EN	SPI3EN	SPI2EN	I2C2EN	I2C1EN	UART5EN	UART8EN	USART3EN	USART2EN	BKPEN	PWREN
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit Field	Name	Description
31	DAC34EN	DAC3/4 clock enable Set and cleared by software. 0: Disable DAC3/4 clock 1: Enable DAC3/4 clock
30	DAC12EN	DAC1/2 clock enable Set and cleared by software. 0: Disable DAC1/2 clock 1: Enable DAC1/2 clock
29	WWDGEN	WWDG clock enable Set and cleared by software. 0: Disable WWDG clock 1: Enable WWDG clock
28	I2C4EN	I2C4 clock enable Set and cleared by software. 0: Disable I2C4 clock 1: Enable I2C4 clock
27	I2C3EN	I2C3 clock enable Set and cleared by software. 0: Disable I2C3 clock 1: Enable I2C3 clock
26	FDCAN3OFFEN	FDCAN3 shutdown request signal enable When this bit is enabled, a shutdown request is sent to the FDCAN module. The CPU clears this bit upon receiving a shutdown acknowledgment from the FDCAN module. 0: Do not send shutdown request 1: Send shutdown request
25	FDCAN2OFFEN	FDCAN2 shutdown request signal enable When this bit is enabled, a shutdown request is sent to the FDCAN module. The CPU clears this bit upon receiving a shutdown acknowledgment from the FDCAN module. 0: Do not send shutdown request 1: Send shutdown request
24	FDCAN1OFFEN	FDCAN1 shutdown request signal enable

Bit Field	Name	Description
		When this bit is enabled, a shutdown request is sent to the FDCAN module. The CPU clears this bit upon receiving a shutdown acknowledgment from the FDCAN module. 0: Do not send shutdown request 1: Send shutdown request
23	BTIM2EN	BTIM2 clock enable Set and cleared by software. 0: Disable BTIM2 clock 1: Enable BTIM2 clock
22	BTIM1EN	BTIM1 clock enable Set and cleared by software. 0: Disable BTIM1 clock 1: Enable BTIM1 clock
21	FDCAN3EN	FDCAN3 clock enable Set and cleared by software. 0: Disable FDCAN3 clock 1: Enable FDCAN3 clock
20	FDCAN2EN	FDCAN2 clock enable Set and cleared by software. 0: Disable FDCAN2 clock 1: Enable FDCAN2 clock
19	FDCAN1EN	FDCAN1 clock enable Set and cleared by software. 0: Disable FDCAN1 clock 1: Enable FDCAN1 clock
18	USBFSEN	USBFS clock enable Set and cleared by software. 0: Disable USBFS clock 1: Enable USBFS clock
17	UCDREN	UCDR clock enable Set and cleared by software. 0: Disable UCDR clock 1: Enable UCDR clock
16	GTIM7EN	GTIM7 clock enable Set and cleared by software. 0: Disable GTIM7 clock 1: Enable GTIM7 clock
15	GTIM6EN	GTIM6 clock enable Set and cleared by software. 0: Disable GTIM6 clock 1: Enable GTIM6 clock
14	GTIM5EN	GTIM5 clock enable

Bit Field	Name	Description
		Set and cleared by software. 0: Disable GTIM5 clock 1: Enable GTIM5 clock
13	GTIM4EN	GTIM4 clock enable Set and cleared by software. 0: Disable GTIM4 clock 1: Enable GTIM4 clock
12	GTIM3EN	GTIM3 clock enable Set and cleared by software. 0: Disable GTIM3 clock 1: Enable GTIM3 clock
11	GTIM2EN	GTIM2 clock enable Set and cleared by software. 0: Disable GTIM2 clock 1: Enable GTIM2 clock
10	GTIM1EN	GTIM1 clock enable Set and cleared by software. 0: Disable GTIM1 clock 1: Enable GTIM1 clock
9	SPI3EN	SPI3 clock enable Set and cleared by software. 0: Disable SPI3 clock 1: Enable SPI3 clock
8	SPI2EN	SPI2 clock enable Set and cleared by software. 0: Disable SPI2 clock 1: Enable SPI2 clock
7	I2C2EN	I2C2 clock enable Set and cleared by software. 0: Disable I2C2 clock 1: Enable I2C2 clock
6	I2C1EN	I2C1 clock enable Set and cleared by software. 0: Disable I2C1 clock 1: Enable I2C1 clock
5	UART5EN	UART5 clock enable Set and cleared by software. 0: Disable UART5 clock 1: Enable UART5 clock
4	UART8EN	UART8 clock enable Set and cleared by software. 0: Disable UART8 clock

Bit Field	Name	Description
		1: Enable UART8 clock
3	USART3EN	USART3 clock enable Set and cleared by software. 0: Disable USART3 clock 1: Enable USART3 clock
2	USART2EN	USART2 clock enable Set and cleared by software. 0: Disable USART2 clock 1: Enable USART2 clock
1	BKPEN	Backup interface enable Software set and clear. 0: Backup interface disabled 1: Backup interface enabled
0	PWREN	Power interface clock enable Software set and clear 0: Power interface clock disabled 1: Power interface clock enabled

4.3.10 APB2 Peripheral Clock Enable Register (RCC_APB2PCLKEN)

Address offset: 0x20

Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	USART4EN	COMPFILTEN	PGAEN	COMPEN	GTIM10EN	GTIM9EN	GTIM8EN	UART7EN	UART6EN	USART1EN	SPI1EN	SPI4EN	SPI5EN	SPI6EN	AFIOEN
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit Field	Name	Description
31:15	Reserved	Reserved, the reset value must be maintained.
14	USART4EN	USART4 clock enable Set and cleared by software. 0: USART4 clock disabled 1: USART4 clock enabled
13	COMPFILTEN	COMP filter clock enable Software set and reset. 0: COMP filter clock disabled 1: COMP filter clock enabled

Bit Field	Name	Description
12	PGAEN	PGA clock enable Software set and reset. 0: PGA clock disabled 1: PGA clock enabled
11	COMPEN	COMP clock enable Software set and reset. 0: COMP clock disabled 1: COMP clock enabled
10	GTIM10EN	GTIM10 Timer clock enable Set and cleared by software. 0: GTIM10 timer clock disabled 1: GTIM10 timer clock enabled
9	GTIM9EN	GTIM9 Timer clock enable Set and cleared by software. 0: GTIM9 timer clock disabled 1: GTIM9 timer clock enabled
8	GTIM8EN	GTIM8 Timer clock enable Set and cleared by software. 0: GTIM8 timer clock disabled 1: GTIM8 timer clock enabled
7	UART7EN	UART7 Timer clock enable Set and cleared by software. 0: UART7 timer clock disabled 1: UART7 timer clock enabled
6	UART6EN	UART6 Timer clock enable Set and cleared by software. 0: UART6 timer clock disabled 1: UART6 timer clock enabled
5	USART1EN	USART1 Timer clock enable Set and cleared by software. 0: USART1 timer clock disabled 1: USART1 timer clock enabled
4	SPI1EN	SPI1 clock enable Software set and reset. 0: SPI1 clock disabled 1: SPI1 clock enabled
3	SPI4EN	SPI4 clock enable Software set and reset. 0: SPI4 clock disabled 1: SPI4 clock enabled
2	SPI5EN	SPI5 clock enable Software set and reset.

Bit Field	Name	Description
		0: SPI5 clock disabled 1: SPI5 clock enabled
1	SPI6EN	SPI6 clock enable Software set and reset. 0: SPI6 clock disabled 1: SPI6 clock enabled
0	AFIOEN	Alternate function IO clock enable Set and cleared by software. 0: Alternate Function IO clock disabled 1: Alternate Function IO clock enabled

4.3.11 AHB Peripheral Reset Register (RCC_AHBPRST)

Address offset: 0x24

Reset value: 0x0000 0100

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ETHRST	USBHSRST	FMACRST	CORDICRST	ATIM3RST	ATIM2RST	ATIM1RST	DVPRST	DAC78RST	DAC56RST	ADC4RST	ADC3RST	ADC2RST	ADC1RST	IOPHRST	IOPGRST
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IOPFRST	IOPERST	IOPDRST	IOPCRST	IOPBRST	IOPARST	Reserved	FEMCRST	SDIORST	USBHSPHYRST	XSPIRST	RNGCRST	SACRST	SHRTPLLPHARST	SHRTPLLRST	SHRTIMRST
rw	rw	rw	rw	rw	rw		rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit Field	Name	Description
31	ETHRST	ETH reset Software set or clear. 0: Clear reset 1: Reset ETH
30	USBHSRST	USBHS reset Software set or clear. 0: Clear reset 1: Reset USBHS
29	FMACRST	FMAC reset Software set or clear. 0: Clear reset 1: Reset FMAC
28	CORDICRST	CORDIC reset Software set or clear. 0: Clear reset 1: Reset CORDIC

Bit Field	Name	Description
27	ATIM3RST	ATIM3 reset Software set or clear. 0: Clear reset 1: Reset ATIM3
26	ATIM2RST	ATIM2 reset Software set or clear. 0: Clear reset 1: Reset ATIM2
25	ATIM1RST	ATIM1 reset Software set or clear. 0: Clear reset 1: Reset ATIM1
24	DVPRST	DVP reset Software set or clear. 0: Clear reset 1: Reset DVP
23	DAC78RST	DAC7/8 reset Software set or clear. 0: Clear reset 1: Reset DAC7/8
22	DAC56RST	DAC5/6 reset Software set or clear. 0: Clear reset 1: Reset DAC5/6
21	ADC4RST	ADC4 reset Software set or clear. 0: Clear reset 1: Reset ADC4
20	ADC3RST	ADC3 reset Software set or clear. 0: Clear reset 1: Reset ADC3
19	ADC2RST	ADC2 reset Software set or clear. 0: Clear reset 1: Reset ADC2
18	ADC1RST	ADC1 reset Software set or clear. 0: Clear reset 1: Reset ADC1
17	IOPHRST	GPIOH reset Software set or clear.

Bit Field	Name	Description
		0: Clear reset 1: Reset GPIOH
16	IOPGRST	GPIOG reset Software set or clear. 0: Clear reset 1: Reset GPIOG
15	IOPFRST	GPIOF reset Software set or clear. 0: Clear reset 1: Reset GPIOF
14	IOPERST	GPIOE reset Software set or clear. 0: Clear reset 1: Reset GPIOE
13	IOPDRST	GPIOD reset Software set or clear. 0: Clear reset 1: Reset GPIOD
12	IOPCRST	GPIOC reset Software set or clear. 0: Clear reset 1: Reset GPIOC
11	IOPBRST	GPIOB reset Software set or clear. 0: Clear reset 1: Reset GPIOB
10	IOPARST	GPIOA reset Software set or clear. 0: Clear reset 1: Reset GPIOA
9	Reserved	Reserved, the reset value must be maintained.
8	FEMCRST	FEMC reset Software set or clear. 0: Clear reset 1: Reset FEMC
7	SDIORST	SDIO reset Software set or clear. 0: Clear reset 1: Reset SDIO
6	USBHSPHYRST	USBHS PHY POR reset Software set or clear. 0: Clear reset

Bit Field	Name	Description
		1: Reset USBHS PHY POR *Note: Before using the USBHS module, this bit should be cleared first.*
5	XSPIRST	XSPI reset Software set or clear. 0: Clear reset 1: Reset XSPI
4	RNGCRST	RNGC reset Software set or clear. 0: Clear reset 1: Reset RNGC
3	SACRST	SAC reset Software set or clear. 0: Clear reset 1: Reset SAC
2	SHRTPLLPHARST	SHRTPLL phase reset Software set or clear. 0: Clear reset 1: Reset SHRTPLL phase.
1	SHRTPLLRST	SHRTPLL reset Software set or clear. 0: Clear reset 1: Reset SHRTPLL. *Note: Reset must be maintained for 5us before clearing the reset.*
0	SHRTIMRST	SHRTIM reset Software set or clear. 0: Clear reset 1: Reset SHRTIM

4.3.12 APB1 Peripheral Reset Register (RCC_APB1PRST)

Address offset: 0x28

Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DAC34RST	DAC12RST	WWDGRST	I2C4RST	I2C3RST	Reserved			BTIM2RST	BTIM1RST	FDCAN3RST	FDCAN2RST	FDCAN1RST	USBFSRST	UCDRRST	GTIM7RST
rw	rw	rw	rw	rw				rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GTIM6RST	GTIM5RST	GTIM4RST	GTIM3RST	GTIM2RST	GTIM1RST	SPI3RST	SPI2RST	I2C2RST	I2C1RST	UART5RST	UART8RST	USART3RST	USART2RST	BKPRST	PWRRST
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit Field	Name	Description
31	DAC34RST	DAC3/4 reset Software set or clear. 0: Clear reset 1: Reset DAC3/4.
30	DAC12RST	DAC1/2 reset Software set or clear. 0: Clear reset 1: Reset DAC1/2
29	WWDGRST	WWDG reset Software set or clear. 0: Clear reset 1: Reset WWDG
28	I2C4RST	I2C4 reset Software set or clear. 0: Clear reset 1: Reset I2C4
27	I2C3RST	I2C3 reset Software set or clear. 0: Clear reset 1: Reset I2C3
26:24	Reserved	Reserved, the reset value must be maintained.
23	BTIM2RST	BTIM2 reset Software set or clear. 0: Clear reset 1: Reset BTIM2
22	BTIM1RST	BTIM1 reset Software set or clear. 0: Clear reset 1: Reset BTIM1
21	FDCAN3RST	FDCAN3 reset Software set or clear. 0: Clear reset 1: Reset FDCAN3
20	FDCAN2RST	FDCAN2 reset Software set or clear. 0: Clear reset 1: Reset FDCAN2
19	FDCAN1RST	FDCAN1 reset Software set or clear. 0: Clear reset 1: Reset FDCAN1
18	USBFSRST	USBFS reset

Bit Field	Name	Description
		Software set or clear. 0: Clear reset 1: Reset USBFS
17	UCDRRST	UCDR reset Software set or clear. 0: Clear reset 1: Reset UCDR
16	GTIM7RST	GTIM7 reset Software set or clear. 0: Clear reset 1: Reset GTIM7
15	GTIM6RST	GTIM6 reset Software set or clear. 0: Clear reset 1: Reset GTIM6
14	GTIM5RST	GTIM5 reset Software set or clear. 0: Clear reset 1: Reset GTIM5
13	GTIM4RST	GTIM4 reset Software set or clear. 0: Clear reset 1: Reset GTIM4
12	GTIM3RST	GTIM3 reset Software set or clear. 0: Clear reset 1: Reset GTIM3
11	GTIM2RST	GTIM2 reset Software set or clear. 0: Clear reset 1: Reset GTIM2
10	GTIM1RST	GTIM1 reset Software set or clear. 0: Clear reset 1: Reset GTIM1
9	SPI3RST	SPI3 reset Software set or clear. 0: Clear reset 1: Reset SPI3
8	SPI2RST	SPI2 reset Software set or clear. 0: Clear reset

Bit Field	Name	Description
		1: Reset SPI2
7	I2C2RST	I2C2 reset Software set or clear. 0: Clear reset 1: Reset I2C2
6	I2C1RST	I2C1 reset Software set or clear. 0: Clear reset 1: Reset I2C1
5	UART5RST	UART5 reset Software set or clear. 0: Clear reset 1: Reset UART5
4	UART8RST	UART8 reset Software set or clear. 0: Clear reset 1: Reset UART8
3	USART3RST	USART3 reset Software set or clear. 0: Clear reset 1: Reset USART3
2	USART2RST	USART2 reset Software set or clear. 0: Clear reset 1: Reset USART2
1	BKPRST	Backup interface reset Software set or clear. 0: Clear reset 1: Reset backup interface
0	PWRRST	Power interface reset Software set or clear. 0: Clear reset 1: Reset power interface

4.3.13 APB2 Peripheral Reset Register (RCC_APB2PRST)

Address offset: 0x2C

Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	USART4RST	Reserved	PGARST	COMPRST	GTIM10RST	GTIM9RST	GTIM8RST	UART7RST	UART6RST	USART1RST	SPI1RST	SPI4RST	SPI5RST	SPI6RST	AFIORST
	rw		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit Field	Name	Description
31:15	Reserved	Reserved, the reset value must be maintained.
14	USART4RST	USART4 reset Software set or clear. 0: Clear reset 1: Reset USART4
13	Reserved	Reserved, the reset value must be maintained.
12	PGARST	PGA reset Software set or clear. 0: Clear reset 1: Reset PGA
11	COMPRST	COMP reset Software set or clear. 0: Clear reset 1: Reset COMP
10	GTIM10RST	GTIM10 reset Software set or clear. 0: Clear reset 1: Reset GTIM10
9	GTIM9RST	GTIM9 reset Software set or clear. 0: Clear reset 1: Reset GTIM9
8	GTIM8RST	GTIM8 reset Software set or clear. 0: Clear reset 1: Reset GTIM8
7	UART7RST	UART7 reset Software set or clear. 0: Clear reset 1: Reset UART7
6	UART6RST	UART6 reset Software set or clear. 0: Clear reset 1: Reset UART6
5	USART1RST	USART1 reset Software set or clear.

Bit Field	Name	Description
		0: Clear reset 1: Reset USART1
4	SPI1RST	SPI1 reset Software set or clear. 0: Clear reset 1: Reset SPI1
3	SPI4RST	SPI4 reset Software set or clear. 0: Clear reset 1: Reset SPI4
2	SPI5RST	SPI5 reset Software set or clear. 0: Clear reset 1: Reset SPI5
1	SPI6RST	SPI6 reset Software set or clear. 0: Clear reset 1: Reset SPI6
0	AFIORST	AFIO reset Software set or clear. 0: Clear reset 1: Reset AFIO

4.3.14 MCO Configuration Register (RCC_MCOCFG)

Address offset: 0x30

Reset value: 0x00000000

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="16">Reserved</td></tr>
</table>

<table>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="8">Reserved</td><td colspan="4">MCO2[3:0]</td><td colspan="4">MCO1[3:0]</td></tr>
<tr><td colspan="8"></td><td colspan="4">rw</td><td colspan="4">rw</td></tr>
</table>

Bit Field	Name	Description
31:8	Reserved	Reserved, the reset value must be maintained.
7:4	MCO2[3:0]	Microcontroller Clock Output 2 It is set and cleared by software. 0xxx: No clock

Bit Field	Name	Description
		1000: Select system clock 1001: Select HSI clock 1010: Select HSE clock 1011: Select PLL divided clock 1100: Select LSI clock 1101: Select LSE clock Other values: Select PLL divided clock *Note: The clock output may be truncated during startup and when switching the MCO clock source. When the system clock is output to the MCO pin, ensure that the output clock frequency does not exceed the maximum I/O port frequency of 50MHz.*
3:0	MCO1[3:0]	Microcontroller Clock Output 1 It is set and cleared by software. 0xxx: No clock 1000: Select system clock 1001: Select HSI clock 1010: Select HSE clock 1011: Select PLL divided clock 1100: Select LSI clock 1101: Select LSE clock Other values: Select PLL divided clock *Note: The clock output may be truncated during startup and when switching the MCO clock source. When the system clock is output to the MCO pin, ensure that the output clock frequency does not exceed the maximum I/O port frequency of 50MHz.*

4.3.15 Backup Domain Control Register (RCC_BDCTRL)

Address offset: 0x34

Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		LPTIM2RST	LPTIM1RST	LPTIM2EN	LPTIM2SEL[2:0]			LPTIM1EN	LPTIM1SEL[2:0]			Reserved			BDSFTRST
		rw	rw	rw	rw			rw	rw						rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTCEN	Reserved					RTCSEL[1:0]		Reserved				LSECSSEN	LSEBP	LSERD	LSEEN
rw						rw						rw	rw	r	rw

Bit Field	Name	Description
31:30	Reserved	Reserved, the reset value must be maintained.

Bit Field	Name	Description
29	LPTIM2RST	LPTIM2 reset Software set or clear. 0: Clear reset 1: Reset LPTIM2
28	LPTIM1RST	LPTIM1 reset Software set or clear. 0: Clear reset 1: Reset LPTIM1
27	LPTIM2EN	LPTIM2 clock enable Software set and clear. 0: Disable LPTIM2 clock 1: Enable LPTIM2 clock
26:24	LPTIM2SEL[2:0]	LPTIM2 clock source selection. Software set or clear. 010: Select HSI as input clock 011: Select HSE as input clock 100: If APB1 prescaler is 1, select PCLK1 as LPTIM clock source. Otherwise, select PCLK1×2 110: Select LSI as input clock 111: Select LSE as input clock Other values: Not allowed to configure *Note: For A version chips, select LSI by configuring as 0b000, and select LSE by configuring as 0b001 as the input clock.*
23	LPTIM1EN	LPTIM1 clock enable Software set and clear. 0: Disable LPTIM1 clock 1: Enable LPTIM1 clock
22:20	LPTIM1SEL[2:0]	LPTIM1 clock source selection. Software set or clear. 010: Select HSI as input clock 011: Select HSE as input clock 100: If APB1 prescaler is 1, select PCLK1 as LPTIM clock source. Otherwise, select PCLK1×2 110: Select LSI as input clock 111: Select LSE as input clock Other values: Not allowed to configure *Note: For A version chips, select LSI by configuring as 0b000, and select LSE by configuring as 0b001 as the input clock.*
19:17	Reserved	Reserved, the reset value must be maintained.
16	BDSFTRST	Backup domain software reset. Software set to 1 or cleared. 0: No effect

Bit Field	Name	Description
		1: Reset the entire backup domain
15	RTCEN	RTC clock enable Software set and clear. 0: Disable RTC clock 1: Enable RTC clock
14:10	Reserved	Reserved, the reset value must be maintained.
9:8	RTCSEL[1:0]	RTC clock source selection It is set by software to choose the RTC clock source. Once the RTC clock source is selected, it cannot be changed until the next backup domain reset. These bits can be reset by setting the BDSFTRST bit. 00: No clock 01: Select LSE oscillator as RTC clock 10: Select LSI oscillator as RTC clock 11: Select HSE oscillator divided by 128 as RTC clock
3	LSECSSEN	LSE clock security system enable Software set or clear. 0: Disable LSE clock detector 1: Enable LSE clock detector if LSE is ready
2	LSEBP	External Low-Speed Oscillator Bypass. In debug mode, software sets and clears the oscillator bypass. This bit can only be written when the external low-speed oscillator is disabled. 0: LSE oscillator not bypassed 1: LSE oscillator bypassed
1	LSERD	External Low-Speed Clock Oscillator Ready. Hardware sets and clears to indicate whether the LSE oscillator is ready. After clearing the LSEEN bit, LSERD is cleared after 6 cycles of the LSE clock. 0: External low-speed oscillator not ready 1: External low-speed oscillator ready
0	LSEEN	External Low-Speed Clock Oscillator Enable Software set and clear. 0: Disable external low-speed oscillator 1: Enable external low-speed oscillator.

Note: This register is in the backup domain, therefore, these bits are write-protected after reset and can only be modified after setting the PWR_CTRL.DBKP bit. These bits can only be cleared by a backup domain reset. Any internal or external reset will not affect these bits.

4.3.16 Clock Security System Control Register (RCC_CSSCTRL)

Address offset: 0x38

Reset value: 0x00000848

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

Reserved									HSICSSEN	Reserved	WKUPCLKSEL	Reserved	HSEPOF	HSEFHF	HSEFLF
									rw		rw		r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HSEPOEN	HSEFHEN	HSEFLEN	HSEDLSEL	HSEPOTHR[3:0]				HSEFHTHR[3:0]				HSEFLTHR[3:0]			
rw	rw	rw	rw	rw				rw				rw			

Bit Field	Name	Description
31:23	Reserved	Reserved, the reset value must be maintained.
22	HSICSSEN	HSI Clock Security System Enable bit Set and cleared by software. 0: Disable clock detector 1: Enable clock detector if HSI oscillator is ready
21	Reserved	Reserved, the reset value must be maintained.
20	WKUPCLKSEL	STOP0 Mode Wake-up Clock Selection Set and cleared by software. 0: Wake-up using HSI 1: Wake-up using HSE
19	Reserved	Reserved, the reset value must be maintained.
18	HSEPOF	HSE frequency change rate Out of Range Flag 0: No frequency change rate out of range occurred 1: Frequency change rate out of range occurred
17	HSEFHF	HSE Frequency Exceeded High Limit Flag 0: No frequency exceeded high limit occurred 1: Frequency exceeded high limit occurred
16	HSEFLF	HSE Frequency Exceeded Low Limit Flag 0: No frequency exceeded low limit occurred 1: Frequency exceeded low limit occurred
15	HSEPOEN	HSE frequency change rate Out of Range Detection Enable 0: Detection disabled 1: Detection enabled
14	HSEFHEN	HSE Frequency Exceeded High Limit Detection Enable 0: Detection disabled 1: Detection enabled
13	HSEFLEN	HSE Frequency Exceeded Low Limit Detection Enable 0: Detection disabled 1: Detection enabled
12	HSEDLSEL	HSE Detection Delay Selection 0: Delay of 128 cycles based on HSE clock 1: Delay of 32 cycles based on LSI clock (LSI needs to be enabled)
11:8	HSEPOTHR[3:0]	HSE frequency change rate Threshold Configuration Error occurs when the difference between the cycle values of adjacent HSE cycles

Bit Field	Name	Description
		exceeds (t_{HSI} * (HSEPOTHR-2)/32) microseconds.
7:4	HSEFHTHR[3:0]	HSE High Frequency Limit Configuration Error reported when the frequency is greater than (2*32/HSEFLTHR)MHz
3:0	HSEFLTHR[3:0]	HSE Low Frequency Limit Configuration Error reported when the frequency is less than (32/ (HSEFLTHR+1))MHz

4.3.17 AHB1 Peripheral Clock Enable Register (RCC_AHB1PCLKEN)

Address offset: 0x3C

Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SACEN	DAC78EN	DAC56EN	ADC4EN	ADC3EN	ADC2EN	ADC1EN	IOFILTSEL	IOPHEN	IOPGEN	IOPFEN	IOPEEN	IOPDEN	IOPCEN	IOPBEN	IOPAEN
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit Field	Name	Description
31:16	Reserved	Reserved, the reset value must be maintained.
15	SACEN	SAC clock enable Software set and clear. 0: Disable SAC clock 1: Enable SAC clock
14	DAC78EN	DAC7/8 clock enable Software set and clear. 0: Disable DAC7/8 clock 1: Enable DAC7/8 clock
13	DAC56EN	DAC5/6 clock enable Software set and clear. 0: Disable DAC5/6 clock 1: Enable DAC5/6 clock
12	ADC4EN	ADC4 clock enable Software set and clear. 0: Disable ADC4 clock 1: Enable ADC4 clock
11	ADC3EN	ADC3 clock enable Software set and clear. 0: Disable ADC3 clock 1: Enable ADC3 clock

Bit Field	Name	Description
10	ADC2EN	ADC2 clock enable Software set and clear. 0: Disable ADC2 clock 1: Enable ADC2 clock
9	ADC1EN	ADC1 clock enable Software set and clear. 0: Disable ADC1 clock 1: Enable ADC1 clock
8	IOFILTSEL	GPIO filter clock selection 0: Select HCLK clock 1: Select HCLK/2 clock
7	IOPHEN	GPIOH clock enable Software set and clear. 0: Disable GPIOH clock 1: Enable GPIOH clock
6	IOPGEN	GPIOG clock enable Software set and clear. 0: Disable GPIOG clock 1: Enable GPIOG clock
5	IOPFEN	GPIOF clock enable Software set and clear. 0: Disable GPIOF clock 1: Enable GPIOF clock
4	IOPEEN	GPIOE clock enable Software set and clear. 0: Disable GPIOE clock 1: Enable GPIOE clock
3	IOPDEN	GPIOD clock enable Software set and clear. 0: Disable GPIOD clock 1: Enable GPIOD clock
2	IOPCEN	GPIOC clock enable Software set and clear. 0: Disable GPIOC clock 1: Enable GPIOC clock
1	IOPBEN	GPIOB clock enable Software set and clear. 0: Disable GPIOB clock 1: Enable GPIOB clock
0	IOPAEN	GPIOA clock enable Software set and clear. 0: Disable GPIOA clock

Bit Field	Name	Description
		1: Enable GPIOA clock

4.3.18 PLL Clock Control Register (RCC_PLLCTRL)

Address offset: 0x40

Reset value: 0x00045686

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved										SCLKPLLSEL[1:0]		PLLOUTPRES[1:0]		PLLINPRES[1:0]	
										rw		rw		rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DEADZEN	LPFC2[2:0]			LPFR[3:0]				LPFC1[3:0]				Reserved			
rw	rw			rw				rw							

Bit Field	Name	Description
31:22	Reserved	Reserved, the reset value must be maintained.
21:20	SCLKPLLSEL[1:0]	System Clock PLL Clock Source Selection Software set or clear. If RCC_CFG.SCLKSW bit is configured as 10, a different PLL source can be selected as the system clock. x0: Select PLL as system clock (default) 01: Select SHRTPLL as system clock 11: Select USBHS240M as system clock
19:18	PLLOUTPRES[1:0]	PLL Output Division Ratio: 00: No division 01: Division by 2 10: Division by 4 11: Division by 8 *Note: When configured for no division, the duty cycle deviation is 50%+-5%, with lower power consumption; when configured for division, the duty cycle deviation is 50%+-1%, but with higher power consumption*
17:16	PLLINPRES[1:0]	P LL Input Division 00: No division 01: Division by 2 10: Division by 3 11: Division by 4
15	DEADZEN	PLL Monitoring Phase Detector Dead Zone Adjustment
14:12	LPFC2[2:0]	PLL Loop Filter Integral Capacitor Control Select different configurations based on the multiplication factor corresponding to RCC_CFG.PLLMULFCT[5:0]. Amplification frequency range 4~24 times: Configuration 6

Bit Field	Name	Description
		24~40 times: Configuration 6 40~76 times: Configuration 6
11:8	LPFR[3:0]	PLL Loop Filter Zero Resistance Control Select different configurations based on the multiplication factor corresponding to RCC_CFG.PLLMULFCT[5:0]. Amplification frequency range 4~24 times: Configuration 3 24~40 times: Configuration 7 40~76 times: Configuration 12
7:4	LPFC1[3:0]	PLL Loop Filter Bypass Capacitor Control Select different configurations based on the multiplication factor corresponding to RCC_CFG.PLLMULFCT[5:0]. Amplification frequency range 4~24 times: Configuration 4 24~40 times: Configuration 4 40~76 times: Configuration 4
3:0	Reserved	Reserved, the reset value must be maintained.

4.3.19 VREFBUF Control Register (RCC_VREFCTRL)

Address offset: 0x44

Reset value: 0x0000001F

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	TRIMEN	Reserved	RDY	Reserved		HIM	EN	VLSEL[1:0]		TRIMDATA[5:0]					
	rw		r			rw	rw	rw		rw					

Bit Field	Name	Description
31:15	Reserved	Reserved, the reset value must be maintained.
14	TRIMEN	VREFBUF Fine-Tuning Selection 0: VREFBUF fine-tuning configured by hardware 1: VREFBUF fine-tuning configured by the user
13	Reserved	Reserved, the reset value must be maintained.
12	RDY	VREFBUF Ready Flag 0: VREFBUF output not completed 1: VREFBUF output ready
11:10	Reserved	Reserved, the reset value must be maintained.
9	HIM	High-Impedance Mode Enable 0: REF+ pin internally connected to VREFBUF output

Bit Field	Name	Description
		1: REF+ pin in high-impedance state.
8	EN	VREFBUF Enable 0: Disable 1: Enable
7:6	VLSEL[1:0]	Voltage Range Selection 00: VREFBUF output 2.048V 01: VREFBUF output 2.5V 10: VREFBUF output 2.9V Other values: Not allowed to set
5:0	TRIMDATA[5:0]	VREFBUF Calibration VLSEL[1:0] = 00 (VREFBUF output 2.048V): Adjustment step size is 1.7mV VLSEL[1:0] = 01 (VREFBUF output 2.5V): Adjustment step size is 2.1mV VLSEL[1:0] = 10 (VREFBUF output 2.9V): Adjustment step size is 2.5mV

4.3.20 SHRTPLL Clock Control Register1 (RCC_SHRTPLLCTRL1)

Address offset: 0x48

Reset value: 0x101F4000

Note: This register can only be cleared by POR reset

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved	SATEN	Reserved	SHRTPLL MODE	FRACEN	Reserved	CLKF[25:0]									
	rw		rw	rw		rw									

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CLKF[25:0]															
rw															

Bit Field	Name	Description
31	Reserved	Reserved, the reset value must be maintained.
30	SATEN	SHRTPLL Saturation Function Enable 0: Disable 1: Enable
29	Reserved	Reserved, the reset value must be maintained.
28	SHRTPLLMODE	SHRTPLL Module Operating Mode 0: Normal Mode 1: Power-Down Mode
27	FRACEN	SHRTPLL Fractional Accumulation Function Enable 0: Disable 1: Enable
26	Reserved	Reserved, the reset value must be maintained.

Bit Field	Name	Description
25:0	CLKF[25:0]	SHRTPLL Multiplication Factor SHRTPLL = Clock Source * (CLKF[25:0]/16384) / (CLKR[5:0] + 1) / 4

4.3.21 SHRTPLL Clock Control Register2 (RCC_SHRTPLLCTRL2)

Address offset: 0x4C

Reset value: 0x00000000

Note: This register can only be cleared by POR reset

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CLKR[5:0]						Reserved									
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															

Bit Field	Name	Description
31:26	CLKR[5:0]	SHRTPLL Division Factor SHRTPLL = Clock Source * (CLKF[25:0]/16384) / (CLKR[5:0] + 1) / 4
25:0	Reserved	Reserved, the reset value must be maintained.

4.3.22 SHRTPLL Clock Control Register3 (RCC_SHRTPLLCTRL3)

Address offset: 0x50

Reset value: 0x0003D000

Note: This register can only be cleared by POR reset

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved			SHRTPLL RDF	SHRTPLL LCK	SHRTPLL SRC	SHRTPLL EN	Reserved	BWTRIM[11:0]							
			r	rw	rw	rw		rw							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BWTRIM[11:0]				Reserved											
rw															

Bit Field	Name	Description
31:29	Reserved	Reserved, the reset value must be maintained.
28	SHRTPLLRDF	SHRTPLL Clock Ready Flag

Bit Field	Name	Description
		SHRTPLL clock is set by hardware after being ready. 0: SHRTPLL not ready 1: SHRTPLL ready
27	SHRTPLLLCK	SHRTPLL Clock Lose Lock Control 0: Pause SHRTIM counter when SHRTPLL loses lock 1: Do not pause SHRTIM counter when SHRTPLL loses lock
26	SHRTPLLSRC	SHRTPLL Clock Source Set and clear by software, configure to select SHRTPLL clock source. 0: Select HSI clock as SHRTPLL input clock 1: Select HSE clock as SHRTPLL input clock
25	SHRTPLLEN	SHRTPLL Power Enable Bit Set and clear by software. 0: Disable SHRTPLL power 1: Enable SHRTPLL power
24	Reserved	Reserved, the reset value must be maintained.
23:12	BWTRIM[11:0]	SHRTPLL Bandwidth Adjustment Bandwidth = BWTRIM[11:0] + 1 = CLKF[25:0]/32768
11:0	Reserved	Reserved, the reset value must be maintained.

4.3.23 SRAM Configuration Register 1 (RCC_SRAMCFG1)

Address offset: 0x70

Reset value: 0x00000001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		CSRECCF[1:0]		Reserved	SR2PEF	Reserved	SR1PEF	FECCRSTEN	CSRECCCLR	SR2PECLR	SR1PECLR	CSRECCRSTEN[1:0]		CSRECCIEN[1:0]	
		r			r		r	rw	w	w	w	rw		rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	SR2PERSTEN	Reserved	SR2PEIEN	Reserved	SR1PERSTEN	Reserved	SR1PEIEN	Reserved	SRAMFIN	SRAMWEF	SRAMREF	CSRINIF	CSRINIEN	Reserved	SRAMPEN
	rw		rw		rw		rw		rw	r	r	r	rw		rw

Bit Field	Name	Description
31:30	Reserved	Reserved, the reset value must be maintained.
29:28	CSRECCF[1:0]	CCM SRAM ECC Error Flag 00: No ECC error occurred 01: 1-bit ECC error occurred 10: 2-bit ECC error occurred 11: 1-bit and 2-bit ECC errors occurred
27	Reserved	Reserved, the reset value must be maintained.
26	SR2PEF	SRAM2 Parity Check Error Flag

Bit Field	Name	Description
		0: No parity check error occurred 1: Parity check error occurred
25	Reserved	Reserved, the reset value must be maintained.
24	SR1PEF	SRAM1 Parity Check Error Flag 0: No parity check error occurred 1: Parity check error occurred
23	FECCRSTEN	FLASH ECC Error Reset Enable 0: Reset generated when ECC error detected 1: No reset generated when ECC error detected
22	CSRECCCLR	Clearing CCM SRAM ECC Error Flag Set by software to clear the CSRECCF flag. 0: No effect 1: Clear CSRECCF flag
21	SR2PECLR	Clearing SRAM2 Parity Check Error Flag Set by software to clear the SRAM2PEF flag. 0: No effect 1: Clear SRAM2PEF flag
20	SR1PECLR	Clearing SRAM1 Parity Check Error Flag Set by software to clear the SRAM1PEF flag. 0: No effect 1: Clear SRAM2PEF flag
19:18	CSRECCRSTEN[1:0]	CCM SRAM ECC Error Reset Enable 00: No reset generated when ECC error detected 01: Reset generated when 1-bit ECC error detected 10: Reset generated when 2-bit ECC error detected 11: Reset generated when 1-bit and 2-bit ECC errors detected
17:16	CSRECCIEN[1:0]	CCM SRAM ECC Error Interrupt Enable 00: No interrupt triggered when ECC error detected 01: Interrupt triggered when 1-bit ECC error detected 10: Interrupt triggered when 2-bit ECC error detected 11: Interrupt triggered when 1-bit and 2-bit ECC errors detected
15	Reserved	Reserved, the reset value must be maintained.
14	SR2PERSTEN	SRAM2 Parity Check Error Reset Enable 0: No reset generated when parity check error detected 1: Reset generated when parity check error detected
13	Reserved	Reserved, the reset value must be maintained.
12	SR2PEIEN	SRAM2 Parity Check Error Interrupt Enable 0: No interrupt triggered when parity check error detected 1: Interrupt triggered when parity check error detected
11	Reserved	Reserved, the reset value must be maintained.
10	SR1PERSTEN	SRAM1 Parity Check Error Reset Enable 0: No reset generated when parity check error detected

Bit Field	Name	Description
		1: Reset generated when parity check error detected
9	Reserved	Reserved, the reset value must be maintained.
8	SR1PEIEN	SRAM1 Parity Check Error Interrupt Enable 0: No interrupt triggered when parity check error detected 1: Interrupt triggered when parity check error detected
7	Reserved	Reserved, the reset value must be maintained.
6	SRAMFIN	SRAM Injection Error Enable 0: Disable ECC or parity check error injection 1: Enable ECC or parity check error injection
5	SRAMWEF	SRAM Write Error Flag 0: No write error occurred 1: Write error occurred
4	SRAMREF	SRAM Read Error Flag 0: No read error occurred 1: Read error occurred
3	CSRINIF	CCM SRAM Initialization Completion Flag 0: CCM SRAM initialization not completed 1: CCM SRAM initialization completed
2	CSRINIEN	CCM SRAM Initialization Enable After setting by software, write to the RCC_SRAMCFG3.INISTART~RCC_SRAMCFG4.INIEND address range with RCC_SRAMCFG2.INIDAT 0: Disable CCM SRAM initialization 1: Enable CCM SRAM initialization
1	Reserved	Reserved, the reset value must be maintained.
0	SRAMPEN	CCM SRAM and BKP SRAM ECC enable 0: Disable 1: Enable

4.3.24 SRAM Configuration Register 2 (RCC_SRAMCFG2)

Address offset: 0x74

Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
INIDAT[31:0]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INIDAT[31:0]															
rw															

Bit Field	Name	Description
31:0	INIDAT[31:0]	SRAM initial data

4.3.25 SRAM Configuration Register 3 (RCC_SRAMCFG3)

Address offset: 0x78

Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
INISTART[31:0]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INISTART[31:0]															
rw															

Bit Field	Name	Description
31:0	INISTART[31:0]	SRAM initial start address

4.3.26 SRAM Configuration Register 4 (RCC_SRAMCFG4)

Address offset: 0x7C

Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
INIEND[31:0]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INIEND[31:0]															
rw															

Bit Field	Name	Description
31:0	INISTART[31:0]	SRAM initial end address

4.3.27 USBHS Clock Control Register 1 (RCC_USBHSCTRL1)

Address offset: 0x84

Reset value: 0x080B900C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved	RXTXPATH	BSTHEN	BSTLEN	Reserved				USBHSSEL	SEBP	Reserved		RESSEL	Reserved		
	rw	rw	rw					rw	rw			rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PDSEL	Reserved			PLLBW	USBHSFSEL			Reserved			WKUPSEL	Reserved			
rw				rw		rw					rw				

Bit Field	Name	Description
31	Reserved	Reserved, the reset value must be maintained.
30	RXTXPATH	Transceiver Asynchronous Control 0: Select FS transmit/receive path 1: Select HS transmit/receive path
29	BSTHEN	High-Byte Transmission Bit-Stuffing Enable 0: Disable Bit-Stuffing 1: Enable Bit-Stuffing
28	BSTLEN	Low-Byte Transmission Bit-Stuffing Enable 0: Disable Bit-Stuffing 1: Enable Bit-Stuffing
27:24	Reserved	Reserved, the reset value must be maintained.
23	USBHSSEL	USBHS Clock Source Selection The clock source frequency must be 16M, 19.2M, 20M, 24M, 26M, or 32M. 0: Select HSE clock as USBHS input clock 1: Select PLL divided clock as USBHS input clock *Note: When selecting the PLL divided clock, the RCC_CFG3.USBPLLSEL can be used to choose the PLL divided clock source as either PLL clock or SHRTPLL clock.*
22	SEBP	Noise Equalizer Bypass 0: Equalizer enabled 1: Equalizer bypassed
21:20	Reserved	Reserved, the reset value must be maintained.
19	RESSEL	Calibration Resistor Selection Used to calibrate the high-speed source impedance of the USB 2.0 femto PHY 0: TXRTUNE requires external resistor connection (REXT) 1: Internal resistor calibration used
18:16	Reserved	Reserved, the reset value must be maintained.
15	PDSEL	Power-Down Control Signal Selection for the Common Module 0: In suspend or sleep mode, REFCLK_LOGIC, Bias, and PLL modules remain powered (input clock remains open and valid) 1: In suspend mode, REFCLK_LOGIC, Bias, and PLL modules power down; in sleep mode, Bias and PLL modules power down
14:12	Reserved	Reserved, the reset value must be maintained.
11	PLLBW	USBHS PLL Bandwidth Adjustment

Bit Field	Name	Description
		0: Select this bit when the input clock frequency is 16M 1: Select this bit when the input clock frequency is 19.2M, 20M, 24M, 26M, or 32M
10:8	USBHSFSEL	USBHS Clock Source Frequency Selection 000: 19.2M 001: 20M 010: 24M 011: 16M or 32M 110: 26M Other values: Not allowed to set
7:5	Reserved	Reserved, the reset value must be maintained.
4	WKUPSEL	Wake-up Mode Selection 0: Select suspend mode wake-up 1: Select sleep mode wake-up
3:0	Reserved	Reserved, the reset value must be maintained.

4.3.28 USBHS Clock Control Register 2 (RCC_USBHSCTRL2)

Address offset: 0x88

Reset value: 0x00805D33

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved												IDDQEN	Reserved		
												rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IMPTRIM		RFTRIM		CROSTRIM		DATAVTRIM		HDCVTRIM				FLIMPTRIM			
rw		rw		rw		rw		rw				rw			

Bit Field	Name	Description
31:20	Reserved	Reserved, the reset value must be maintained.
19	IDDQEN	IDDQ Test Enable 0: IDDQ analog module powered on 1: IDDQ analog module powered off
18:16	Reserved	Reserved, the reset value must be maintained.
15:14	IMPTRIM	USB Signal Source Impedance Adjustment 00: Source impedance increased by approximately 1.5 ohms 01: 0 (default) 10: Source impedance reduced by approximately 2 ohms 11: Source impedance reduced by approximately 4 ohms
13:12	RFTRIM	HS Transmitter Rise/Fall Time Adjustment 00: +5%

Bit Field	Name	Description
		01: 0 (default) 10: -3% 11: -7%
11:10	CROSTRIM	Transmitter High-Speed Divider Adjustment 00: Reserved 01: -15 mV 10: +15 mV 11: 0 (default)
9:8	DATAVTRIM	Data Detection Voltage Adjustment 00: +10% 01: 0 (default) 10: -10% 11: -20%
7:4	HDCVTRIM	High-Speed DC Voltage Level Adjustment 0000：-9% 0001：-6% 0010：-3% 0011：0 (default) 0100：+3% 0101：+6% 0110：+9% 0111：+12% 1000：+15% 1001：+18% 1010：+21% 1011：+24% 1100：+27% 1101：+30% 1110：+33% 1111：+36%
3:0	FLIMPTRIM	FS/LS Signal Source Impedance Adjustment 0000：+5% 0001：+2.5% 0011：0 (default) 0111：-2.5% 1111：-5%

4.3.29 BOOT Remap Register （RCC_BOOTREMAP)

Address offset: 0x8C

Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REMAPKEY[7:0]								Reserved					REMAPSEL[2:0]		
rw													rw		

Bit Field	Name	Description
31:16	Reserved	Reserved, the reset value must be maintained.
15:8	REMAPKEY[7:0]	System Remap Enable Writing 0xCC to REMAPKEY[7:0] will trigger boot mode remapping and generate a CPU soft reset.
7:3	Reserved	Reserved, the reset value must be maintained.
2:0	REMAPSEL[2:0]	Remap Boot Mode Selection 000: Boot from system memory; 001: Boot from flash main front bank; 010: Boot from flash main rear bank; 011: Boot from SRAM; 100: Boot from external NORFLASH connected to FEMC; 101: Boot from external memory connected to XSPI; Other values: Disabled

4.3.30 FDCAN Base Address Register (RCC_FDCANBADDR)

Address offset: 0x90

Reset value: 0x20000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BADDR[31:0]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BADDR[31:0]															
rw															

Bit Field	Name	Description
31:0	BADDR[31:0]	Base address of the SRAM where FDCAN module stores data

5 GPIO And AFIO

5.1 Introduction

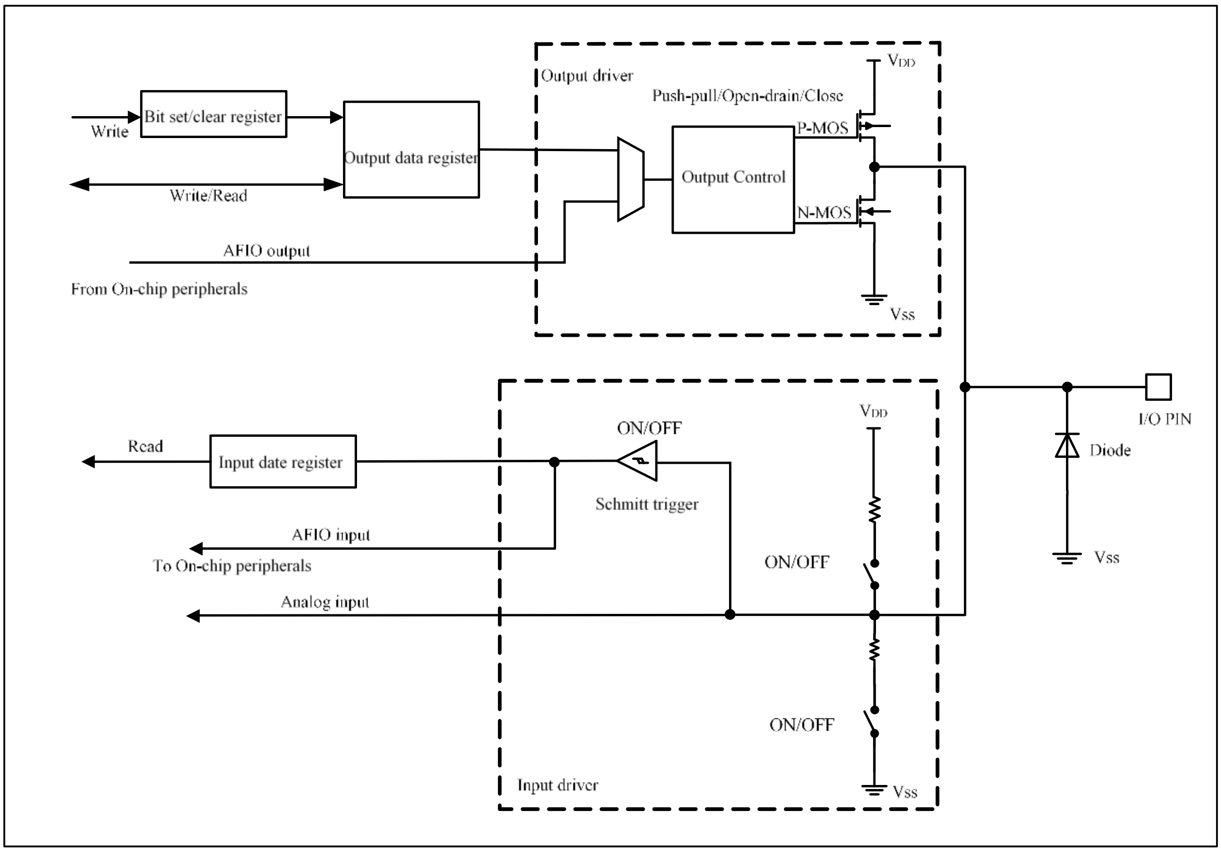
GPIO refers to general-purpose I/O, and AFIO refers to alternate function I/O. The chip supports up to 119 GPIOs, which are divided into 8 groups (GPIOA/GPIOB/GPIOC/GPIOD/GPIOE/GPIOF/GPIOG/GPIOH). Each group has 16 ports (Group H has a total of 7 ports). GPIO ports share pins with other multiplexed peripherals, allowing users to configure them flexibly according to requirements. Each GPIO pin can be independently configured as an output (push-pull or open-drain), input (floating, pull-up, or pull-down) or alternate peripheral function port. Except for the analog pins, other GPIO pins have high current capacity.

GPIO ports have the following characteristics:

- Each bit of the GPIO port can be configured separately by the software into multiple modes:
  - Input floating
  - Input pull up
  - Input pull down
  - Analog function
  - Alternate input, pull-up and pull-down configurable
  - Open drain output, pull-up and pull-down configurable
  - Push-pull output, pull-up and pull-down configurable
  - Push-pull alternate function, pull-up and pull-down configurable
  - Open drain alternate function, pull-up and pull-down configurable
- Separate bit setting or bit clearing functions
- All IOs have external interrupt capability
- All I/O support low-power mode wake-up, with configurable rising or falling edge
  - 16 EXTI can be used for STOP0 mode wake-up, all I/O can be alternated as EXTI
  - PA0/PA2/PC5/PC13/PE6 can be used for STANDBY mode wake-up
- Support software remapping of IO alternate function
- Support GPIO lock mechanism, which can only be cleared by reset after being locked

Each I/O port bit can be programmed arbitrarily, but the I/O port registers must be accessed as 32-bit words (16-bit half-word or 8-bit byte access is not allowed). The figure below shows the basic structure of an I/O port.

Figure 5-1 Basic Structure of I/O Port



Most I/O ports can be configured to be compatible with 5V.

5.2 I/O Function Description

5.2.1 I/O Mode Configuration

The IO port mode can be configured through the registers GPIOx_PMODE, GPIOx_POTYPE and GPIOx_PUPD (x= A,B,C,D,E,F,G,H). The I/O configurations in different operation modes are shown in the following table:

Table 5-1 I/O Port Configuration Table

PMODE[1:0]	POTYPE	PUPD[1:0]		I/O configuration
01	0	0	0	General-purpose output push-pull
	0	0	1	General-purpose output push-pull + pull-up
	0	1	0	General-purpose output push-pull + pull-down
	0	1	1	Reserved
	1	0	0	General-purpose output open-drain
	1	0	1	General-purpose output open-drain + pull-up
	1	1	0	General-purpose output open-drain + pull-down
	1	1	1	Reserved
10	0	0	0	Alternate function push-pull

PMODE[1:0]	POTYPE	PUPD[1:0]		I/O configuration
	0	0	1	Alternate function push-pull + pull-up
	0	1	0	Alternate function push-pull + pull-down
	0	1	1	Reserved
	1	0	0	Alternate function open-drain
	1	0	1	Alternate function open-drain + pull-up
	1	1	0	Alternate function open-drain + pull-down
	1	1	1	Reserved
00	x	0	0	Input floating
	x	0	1	Input pull-up
	x	1	0	Input pull-down
	x	1	1	Reserved
11	x	0	0	Analog
	x	0	1	Reserved
	x	1	0	
	x	1	1	

The input and output characteristics of I/O under different configurations are shown in the following table：

Table 5-2 Input And Output Characteristics Of Different Configurations

Feature	GPIO Input	GPIO Output	Analog	Alternate function
Output buffer	Disabled	Enabled	Disabled	Configure according to peripheral functions
Schmitt trigger	Enabled	Enabled	Disabled Output is forced to 0	Enabled
Pull-up/pull-down/float	Configurable	Configurable	Disabled	Configurable
Open drain	Disabled	Configurable, GPIO outputs 0 when the output data is "0", and GPIO high impedance when "1"	Disabled	Configurable, GPIO outputs 0 when the output data is "0", and GPIO high impedance when "1"
Push pull	Disabled	Configurable, when the output data is "0", the GPIO outputs 0, and when the output data is "1", the GPIO outputs 1	Disabled	Configurable, GPIO outputs 0 when the output data is "0", and GPIO high impedance when "1"
Input data register (I/O status)	Readable	Readable	Read out 0	Readable
Output data register(Output value)	Invalid	Readable and writable	Invalid	Readable

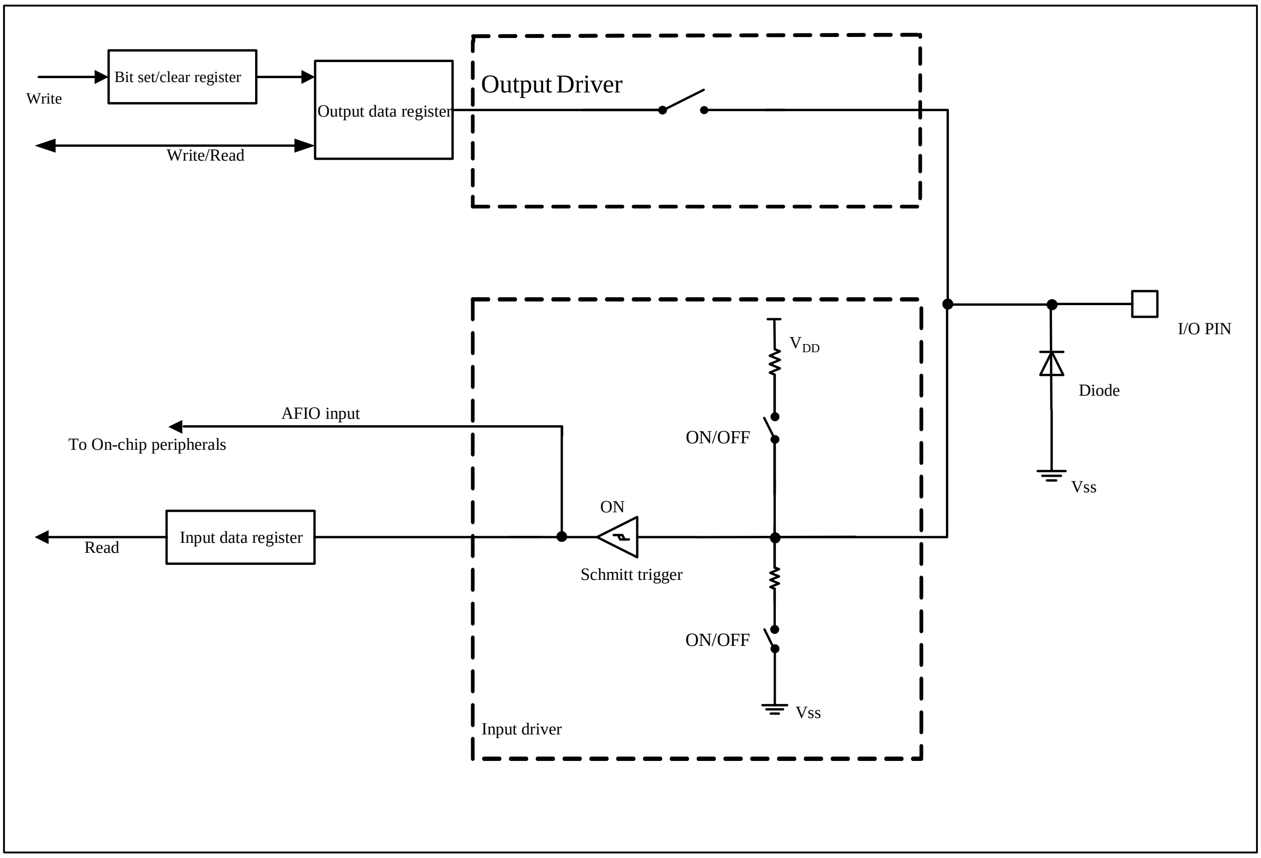
5.2.1.1 Input mode

When I/O port is configured in input mode:

- Output buffer is disabled
- Schmitt trigger input is activated

- Whether pull-up and pull-down resistors are enabled depends on the configuration of the GPIO_PUPD register
- The data appearing on the I/O pin is sampled into the input data register
- Read access to the input data register provides the I/O status
- When used as alternate input, the GPIOx_AFSELy register needs to be configured

Figure 5-2 Input Floating/Pull-up/Pull-Down/Alternate Configuration



5.2.1.2 Output mode

When I/O port is configured as output mode:

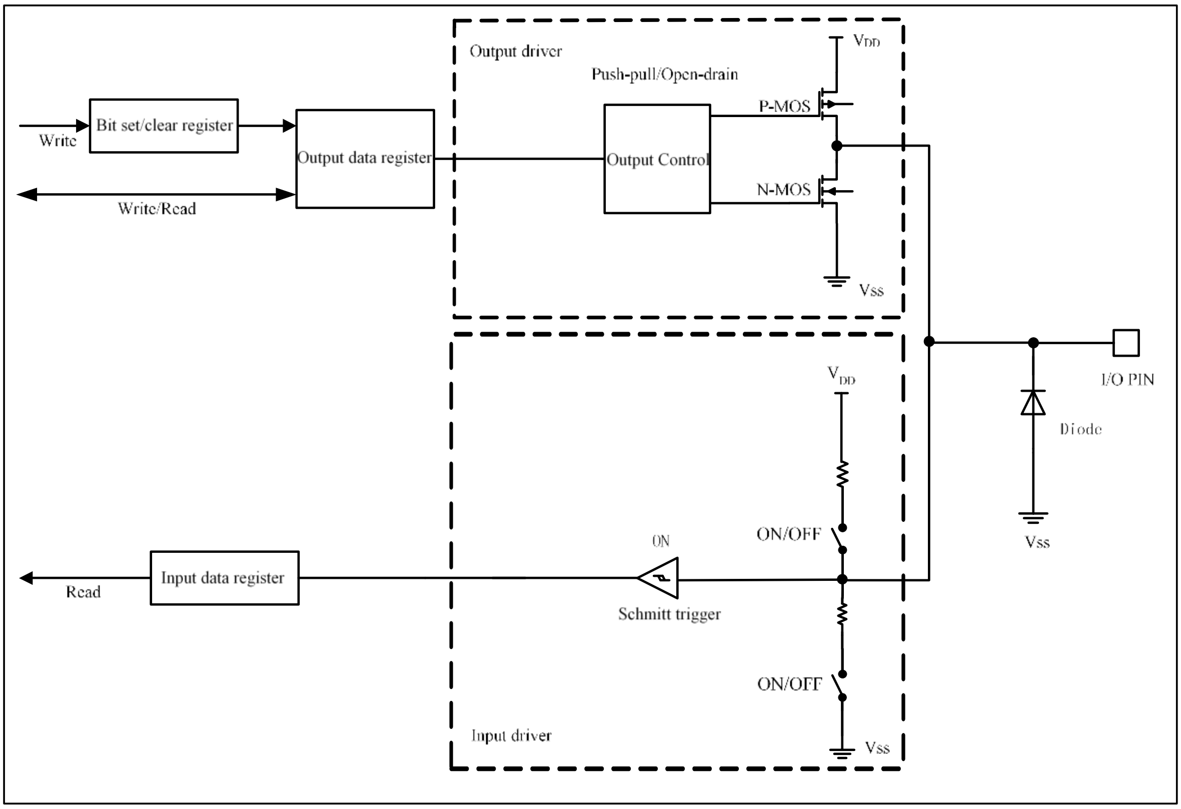
- Output buffer is activated
  - Open-drain mode: '0' on the output data register activates N-MOS, and the pin outputs low level.

    The '1' port on the output data register is placed in a high impedance state (P-MOS is never activated)
  - Push-pull mode: '0' on the output data register activates N-MOS, and the pin outputs low level.

    '1' on the output data register activates P-MOS, and the pin outputs high level.
- Schmidt trigger input is activated
- Whether the weak pull-up and pull-down resistors are enabled depends on the configuration of the GPIO_PUPD register
- The data appearing on the I/O pin is sampled into the input data register

- Read access to input data register for I/O status
- Read access to the output data register gets the last written value

Figure 5-3 Output Mode Configuration

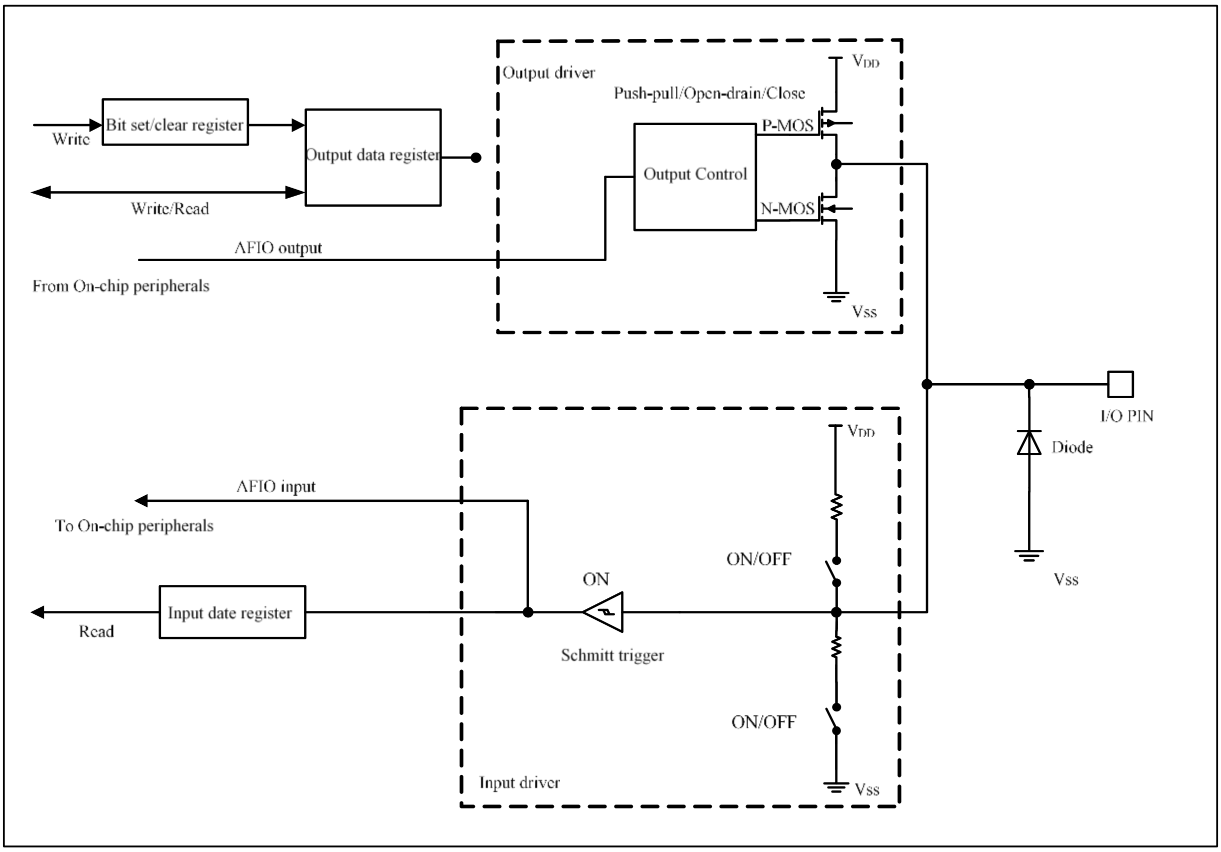


5.2.1.3 Alternate function mode

When the I/O port is configured as alternate function mode:

- In open-drain or push-pull configuration, the output buffer is controlled by the peripheral
- Schmidt trigger input is activated
- Whether the weak pull-up and pull-down resistors are connected depends on the configuration of the GPIOx_PUPD register
- Signal-driven output buffer with built-in peripherals
- The data appearing on the I/O pin is sampled into the input data register
- Read access to input data register for I/O status
- The GPIOx_AFSELy register must be configured

Figure 5-4 Alternate Function Configuration

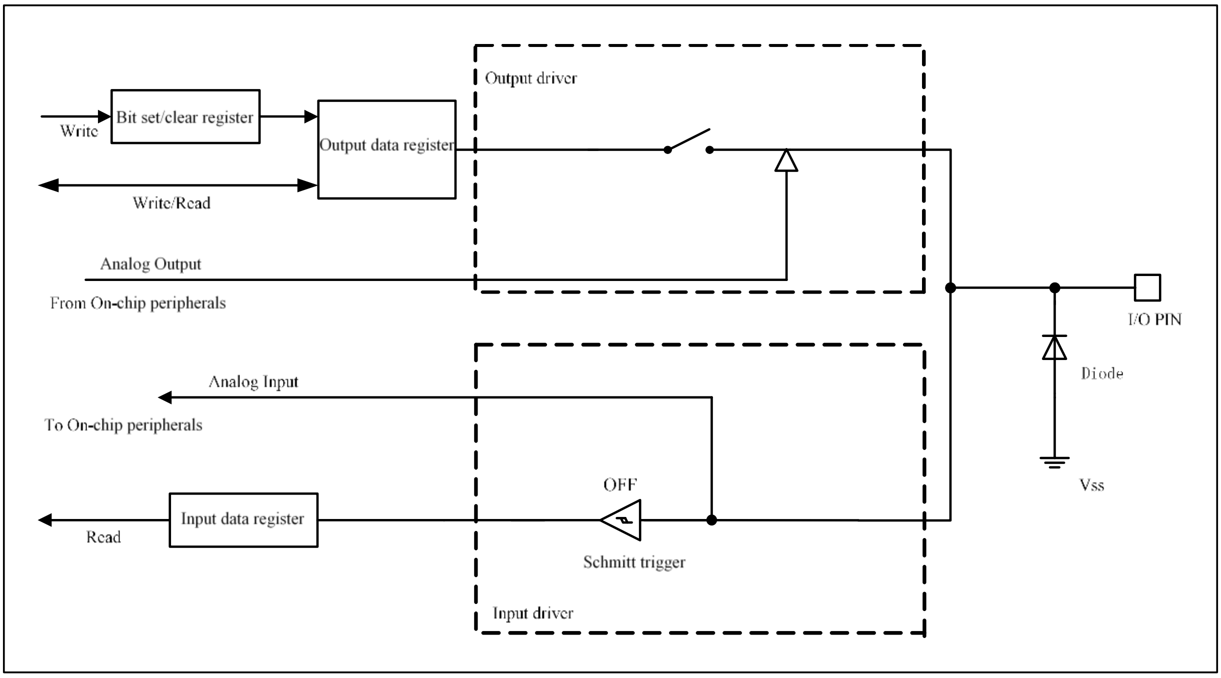


5.2.1.4 Analog mode

When the I/O Port is programmed as analog mode：

- Output buffer is disabled
- Schmitt trigger input is disabled and output value is forced to '0' (achieving zero consumption on each analog I/O pin)
- Weak pull-up and pull-down resistors are disabled
- When reading the input data register, the value is '0'

Figure 5-5 High Impedance Analog Mode Configuration



5.2.2 Status After Reset

During and after reset, the alternate function is not turned on, and the I/O port is configured to analog function mode (GPIOx_PMODE.PMODEx[1:0]=11b). But there are several exceptional signals:

- PH4-BOOT0 (or PB8-BOOT0)、PH6-NRST、PH0-OSC_IN、PH1-OSC_OUT has no GPIO function by default:
  - PH4-BOOT0 (or PB8-BOOT0) is default as the BOOT0 pin, placed in pull-down input mode
  - PH6-NRST is default as the analog pin NRST
  - PH0-OSC_IN, PH1-OSC_OUT pins are default as OSC_IN, OSC_OUT pins, placed in analog mode

  Note: In the N32H47x series, the BOOT0 pin is PB8, and in the N32H48x series, the BOOT0 pin is PH4

- After reset, the SWD_JTAG pins related to the debugging system are enabled by default, and put JTAG pin into input pull-up or pull-down mode
  - PA15: JTDI in input pull-up mode
  - PA14: JTCK in input pull-down mode
  - PA13: JTMS in input pull-up mode
  - PB4: NJTRST in input pull-up mode
  - PB3: JTD0 is placed in input floating mode
- PC13、PC14、PC15：
  - PC13~15 are t in the Backup domain, and the default in analog mode when powered on

5.2.3 Individual Bit Setting And Bit Clearing

By writing '1' to the bit in the "set register (GPIOx_PBSC) and reset register (GPIOx_PBC)", the individual bit operation of the data register (GPIOx_POD) can be realized, and one or more bits can be set. The bit written with '1' is set or cleared accordingly, and the bit not written with '1' will not be changed. The software does not need interrupt disable, and is completed in a single AHB write operation.

5.2.4 External Interrupt/Wake-Up Line

All ports have external interrupt capability, which can be configured in the EXTI module:

- The port must be configured in input mode
- All ports can be configured for SLEEP/STOP0 mode wake-up, supporting configurable rising or falling edges
- PA0/PA2/PC5/PC13/PE6/PH6 can be used for waking up from STANDBY mode
- All I/O ports can be connected to any of the 16 external interrupt/event lines, configured by the AFIO_EXTI_CFGx register

5.2.5 Alternate Function

When the I/O port is configured in alternate function mode. The port mode register (GPIOx_PMODE), output type register (GPIOx_POTYPE configuration push-pull or open-drain) and alternate function configuration register(GPIOx_AFSEL0/1/2/3) must be programmed before using, specifically as follows:

- Alternate input function: The port must be configured in input mode (floating, pull-up, or pull-down) and the input pin must be driven externally.
- Alternate output function: The port must be configured as a multiplexed output mode (push-pull or open-drain).
- Bidirectional alternate function: The port must be configured as a multiplexed output mode (push-pull or open-drain). In this case, the input driver is automatically configured as a floating input mode.
- Alternate function configuration register must be programmed to configure the desired multiplexed function.

In alternate output mode, the port is disconnected from the pin and output data register, and is connected to the output signal of the on-chip peripheral. If software configures a GPIO pin as a alternate output function but the peripheral is not activated, its output will be undefined.

5.2.5.1 Clock output MCO

The microcontroller allows the output of clock signals to external MCO pins, with two MCO outputs (MCO1/MCO2), which can be output through PA3/PA7/PA8/PC9/PF13/PF14/PH6 pins. The MCO pins must be configured as alternate push-pull output mode. The selection of the clock signal for MCO clock can be made from the following 6 clock signals, controlled by the clock configuration register RCC_MCOCFG:

- SYSCLK
- HSI

- HSE
- PLL
- LSI
- LSE

The alternate relationship of MCO1 and MCO2 signals is shown in the table below:

Table 5-3 MCO Alternate Function Remap

Alternate function	GPIO port	Remap
MCO1	PA7	AF13
	PA8	AF10
	PF13	AF6
	PH6	AF1
MCO2	PA3	AF15
	PC9	AF10
	PF14	AF6

5.2.5.2 Backup domain PC13/PC14/PC15 function remapping

The mode of PC14/PC15 is determined according to the following priority order:

- When LSE is enabled (RCC_LDCTRL.LSEEN is set), PC14/PC15 pins will be forced to analog mode. If LSE is configured in external clock mode (RCC_LDCTRL.LSEBP is set), PC14 is forced to analog mode, OSC32_OUT (PC15) can also be used for other purposes
- If LSE is not enabled and the backup domain is powered by VDD, PC14/PC15 can be used as GPIO.

The mode of PC13 is determined according to the following priority order:

- When RTC-TAMP1 is enabled, PC13 is used as the intrusion detection input pin.
- If RTC-TAMP1 is not enabled, but RTC-OUT is enabled (alarm output, calibration clock output, automatic wake-up output, or intrusion event output enabled), PC13 is used as the RTC output pin.
- If none of the above functions are enabled, and the backup domain is powered by VDD, PC13 can be used as GPIO.

The configuration conditions for different modes of PC13 to PC15 are shown in the table below:

PC14 and PC15	Conditions	PAD Mode Configuration	Priority
LSE	LSE enable	Analog mode	high
GPIO	LSE is turned off, the backup domain is powered by VDD, and does not enter low power mode (STANDBY, STOP0).	The mode of GPIO is determined by the application.	low
PC13	**Conditions**	**PAD Mode Configuration**	**Priority**
RTC-TAMP1	RTC-TAMP1 enable	Input floating	high
RTC-OUT	RTC-TAMP1 disabled, alarm output, calibration clock output, automatic	Alternate push-pull	mediu

	wake-up output, or intrusion event output enabled.	output	m
GPIO	RTC-TAMP1 disabled, RTC-OUT disabled, backup domain powered by VDD, and not entering low power mode (STANDBY, STOP0)	The mode of GPIO is determined by the application.	low

When PC13~PC15 are used as GPIO, they can be configured as timestamp trigger input pins, as detailed in the table below:

Table 5-4 MCO Alternate Function Remap

Alternate function	GPIO port	Remap
Timestamp trigger input	PC13	AF9
	PC14	AF9
	PC15	AF9

5.2.5.3 NRST pin used as GPIO port

NRST is the chip's reset pin, which can also be mapped to PH6, controlled by USER3[1:0] in the Option Byte (default is NRST pin). If NRST is configured as a GPIO, the state during reset remains as NRST.

5.2.5.4 HSE/LSE pins used as GPIO ports

OSC_IN and OSC_OUT of HSE are mapped to PH0 and PH1 respectively, and OSC32_IN and OSC32_OUT of LSE are mapped to PC14 and PC15 respectively. If HSE or LSE is off, the corresponding pin can be used as GPIO. If HSE or LSE is on, the corresponding pin goes into analog mode and bypasses the GPIO configuration.

When HSE or LSE is configured in bypass mode, the pins remain as external clock inputs, and OSC_OUT or OSC32_OUT can still be used as GPIO.

5.2.5.5 JTAG/SWD alternate function remapping

The SWD-JTAG debug interface is enabled by default when the chip is powered on, and the debug interface is mapped to the GPIO port, as shown in the following table.

Alternate function	GPIO port	Remap
JTMS/SWDIO	PA13	AF0
JTCK/SWCLK	PA14	AF0
JTDI	PA15	AF0
JTDO	PB3	AF0
	PG15	AF0
NJTRST	PB4	AF0

If you need to use its GPIO function during debugging, you can change the above remapping configuration by setting the register AFIO_RMP_CFG. SWJ_CFG[2:0]. See the table below.

Table 5-5 Debug Port Image

SWJ_CFG [2:0]	Possible Debug Ports	SWD_JTAG I/O Pin Allocation				
		PA13/ JTMS/ SWDIO	PA14/ JTCK/ SWCLK	PA15/ JTDI	PB3/ JTDO	PB4/ NJTRST

000	Complete SWD_JTAG (JTAG-DP+SW-DP) (reset state)	I/O is not available	I/O is not available	I/O is not available	I/O is not available	I/O is not available
001	Complete SWD_JTAG (JTAG-DP+SW-DP) But there is no NJTRST.	I/O is not available	I/O is not available	I/O is not available	I/O is not available	I/O available
010	Turn off JTAG-DP and enable SW-DP.	I/O is not available	I/O is not available	I/O available	I/O available	I/O available
100	Turn off JTAG-DP and SW-DP.	I/O available	I/O available	I/O available	I/O available	I/O available
Others	Disabled					

Note: When the debug port is multiplexed with other peripheral ports, the related debug functions need to be disabled. For example, when PA15 is used as I2C1_SCL, SWJ_CFG needs to be configured as 010 or 100 to release PA15.

5.2.5.5.1 SWJ_CFG configuration matters

When the write buffer of the APB bridge is full, an additional APB cycle is required when writing to the AFIO_RMP_CFG register. This is because the release of the SWD_JTAG pin requires two APB cycles to ensure that the input signals of NJTRST and JTCK are at a clean level.

5.2.5.5.2 Pull-Up/Pull-Down configuration

Since the JTAG pin is directly connected to the internal debug register (JTCK/SWCLK is directly connected to the clock terminal), it must be ensured that the JTAG input pin cannot be in a floating state. In order to avoid any uncontrolled IO levels, the input pins of JTAG are fixed with internal pull-up/pull-down:

- NJTRST: internal pull-up
- JTDI: internal pull-up
- JTMS/SWDIO: internal pull-up
- JTCK/SWCLK: internal pull-down

Note: Once the JTAG interface is released by user software, the GPIO controller regains control, and it is necessary to ensure that the reset state of GPIO registers is consistent with the above.

5.2.5.6 SHRTIM alternate function remapping

The external event input SHRTIM1_EXEVx (x=1~10) can be mapped to all pins, configured in the AFIO_SHRT_EXEV_CFGx (x=1, 2, 3) registers. For other signal mapping relationships, refer to the table below.

Table 5-6 SHRTIM Alternate Function Remapping

Alternate function	GPIO port	Remap
SHRTIM1_CHA1	PA8	AF6
SHRTIM1_CHA2	PA9	AF6
SHRTIM1_CHB1	PA10	AF12
SHRTIM1_CHB2	PA11	AF12
SHRTIM1_CHC1	PB12	AF10
SHRTIM1_CHC2	PB13	AF10
SHRTIM1_CHD1	PB14	AF10

Alternate function	GPIO port	Remap
SHRTIM1_CHD2	PB15	AF10
SHRTIM1_CHE1	PC8	AF11
SHRTIM1_CHE2	PC9	AF11
SHRTIM1_CHF1	PC6	AF11
SHRTIM1_CHF2	PC7	AF11
SHRTIM1_SCIN	PB2	AF5
	PB6	AF9
SHRTIM1_SCOUT	PB1	AF5
	PB3	AF5
SHRTIM1_FALT1	PA12	AF12
SHRTIM1_FALT2	PA15	AF12
SHRTIM1_FALT3	PB10	AF10
SHRTIM1_FALT4	PB11	AF10
SHRTIM1_FALT5	PB0	AF5
	PC7	AF12
SHRTIM1_FALT6	PC10	AF11

5.2.5.7 ATIMx alternate function remapping

5.2.5.7.1 ATIM1 alternate function remapping

Table 5-7 ATIM1 Alternate Function Remapping

Alternate function	GPIO port	Remap
ATIM1_ETR	PA12	AF3
	PC4	AF6
	PE7	AF2
ATIM1_CH1	PA8	AF3
	PC0	AF3
	PE9	AF3
ATIM1_CH2	PA9	AF3
	PB13	AF3
	PC1	AF3
	PE11	AF3
ATIM1_CH3	PA10	AF3
	PC2	AF3
	PD8	AF5
	PE13	AF3
ATIM1_CH4	PA11	AF3
	PC3	AF3
	PD2	AF10

Alternate function	GPIO port	Remap
	PD10	AF5
	PE14	AF3
ATIM1_BKIN	PA6	AF5
	PA14	AF7
	PA15	AF8
	PB8	AF8
	PB10	AF5
	PB12	AF5
	PC13	AF6
	PE15	AF2
ATIM1_BKIN2	PA11	AF7
	PC3	AF6
	PE14	AF2
ATIM1_CH1N	PA7	AF5
	PA11	AF6
	PB13	AF5
	PC13	AF3
	PE8	AF3
	PE10	AF5
ATIM1_CH2N	PA12	AF6
	PB0	AF7
	PB14	AF5
	PE10	AF3
ATIM1_CH3N	PB1	AF7
	PB9	AF8
	PB15	AF5
	PD9	AF5
	PE12	AF3
	PH0	AF3
ATIM1_CH4N	PC5	AF3
	PD5	AF3
	PE15	AF3

5.2.5.7.2 ATIM2 alternate function remapping

Table 5-8 ATIM2 Alternate Function Remapping

Alternate function	GPIO port	Remap
ATIM2_ETR	PA0	AF8
	PB4	AF5
	PB6	AF8

Alternate function	GPIO port	Remap
ATIM2_CH1	PA15	AF6
	PB6	AF7
	PC6	AF3
	PD14	AF6
ATIM2_CH2	PA14	AF6
	PB8	AF5
	PC6	AF15
	PC7	AF3
	PD15	AF6
ATIM2_CH3	PB9	AF5
	PC8	AF3
ATIM2_CH4	PB15	AF3
	PC9	AF3
	PD1	AF3
	PG14	AF11
ATIM2_BKIN	PA0	AF6
	PA6	AF6
	PA10	AF7
	PB3	AF11
	PB7	AF6
	PD2	AF5
ATIM2_BKIN2	PB6	AF5
	PC9	AF5
	PD1	AF5
ATIM2_CH1N	PA5	AF6
	PA7	AF6
	PA15	AF7
	PB3	AF6
	PC10	AF5
	PD15	AF7
ATIM2_CH2N	PB0	AF6
	PB4	AF7
	PB14	AF6
	PC7	AF15
	PC11	AF5
	PC12	AF10
ATIM2_CH3N	PA11	AF9
	PB1	AF6
	PB5	AF7
	PB15	AF6

Alternate function	GPIO port	Remap
	PC9	AF12
	PC12	AF5
	PD2	AF6
ATIM2_CH4N	PC13	AF5
	PD0	AF3
	PD2	AF7
	PG15	AF2
	PH3	AF6

5.2.5.7.3 ATIM3 alternate function remapping

Table 5-9 ATIM3 Alternate Function Remapping

Alternate function	GPIO port	Remap
ATIM3_ETR	PA15	AF13
	PD10	AF3
	PE0	AF6
	PF11	AF3
	PG5	AF2
ATIM3_CH1	PB2	AF6
	PE2	AF3
	PF0	AF3
	PF12	AF3
ATIM3_CH2	PC2	AF6
	PC11	AF12
	PE3	AF3
	PF1	AF3
	PF13	AF3
ATIM3_CH3	PC8	AF1
	PC12	AF12
	PF2	AF3
	PF14	AF3
ATIM3_CH4	PD0	AF5
	PD2	AF14
	PE1	AF3
	PF3	AF3
	PF15	AF3
ATIM3_BKIN	PC13	AF13
	PF7	AF3
	PF9	AF3
	PG3	AF2

Alternate function	GPIO port	Remap
	PG6	AF2
	PH3	AF3
ATIM3_BKIN2	PF8	AF3
	PF10	AF3
	PG4	AF2
ATIM3_CH1N	PE4	AF3
	PF4	AF3
	PG0	AF3
	PH0	AF6
ATIM3_CH2N	PE5	AF3
	PF5	AF3
	PG1	AF3
	PH1	AF6
ATIM3_CH3N	PA0	AF7
	PC4	AF3
	PE6	AF3
	PG2	AF3
ATIM3_CH4N	PA1	AF6
	PE0	AF2
	PG3	AF3

5.2.5.8 GTIMx alternate function remapping

5.2.5.8.1 GTIM1 alternate function remapping

Table 5-10 GTIM1 Alternate Function Remapping

Alternate function	GPIO port	Remap
GTIM1_ETR	PA0	AF10
	PA5	AF2
	PA15	AF15
	PD3	AF2
GTIM1_CH1	PA0	AF3
	PA5	AF3
	PA15	AF3
	PD3	AF3
GTIM1_CH2	PA1	AF3
	PB3	AF3
	PD4	AF5
GTIM1_CH3	PA2	AF3
	PA9	AF2
	PB10	AF3

Alternate function	GPIO port	Remap
	PD7	AF5
GTIM1_CH4	PA3	AF3
	PA10	AF2
	PB2	AF8
	PB11	AF3
	PD6	AF5

5.2.5.8.2 GTIM2 alternate function remapping

Table 5-11 GTIM2 Alternate Function Remapping

Alternate function	GPIO port	Remap
GTIM2_ETR	PB3	AF8
	PD2	AF3
GTIM2_CH1	PA6	AF3
	PB4	AF3
	PC6	AF6
	PE2	AF2
	PE10	AF6
GTIM2_CH2	PA4	AF3
	PA7	AF3
	PB5	AF3
	PC7	AF6
	PE3	AF2
GTIM2_CH3	PB0	AF3
	PC8	AF6
	PE4	AF2
GTIM2_CH4	PB1	AF3
	PB7	AF7
	PC9	AF6
	PE5	AF2

5.2.5.8.3 GTIM3 alternate function remapping

Table 5-12 GTIM3 Alternate Function Remapping

Alternate function	GPIO port	Remap
GTIM3_ETR	PA8	AF2
	PB3	AF7
	PE0	AF3
GTIM3_CH1	PA11	AF2
	PB6	AF3

Alternate function	GPIO port	Remap
	PD12	AF5
GTIM3_CH2	PA12	AF2
	PB7	AF3
	PD13	AF5
GTIM3_CH3	PA13	AF2
	PB8	AF3
	PD14	AF5
GTIM3_CH4	PB9	AF3
	PD15	AF5
	PF6	AF3
	PH2	AF3

5.2.5.8.4 GTIM4 alternate function remapping

Table 5-13 GTIM4 Alternate Function Remapping

Alternate function	GPIO port	Remap
GTIM4_ETR	PB12	AF3
	PD11	AF2
	PF6	AF5
GTIM4_CH1	PA0	AF2
	PB2	AF3
	PF6	AF6
	PH4	AF5
GTIM4_CH2	PA1	AF2
	PC12	AF3
	PE7	AF5
	PF7	AF6
GTIM4_CH3	PA2	AF2
	PE8	AF5
	PF8	AF6
GTIM4_CH4	PA3	AF2
	PE9	AF5
	PF9	AF6

5.2.5.8.5 GTIM5 alternate function remapping

Table 5-14 GTIM5 Alternate Function Remapping

Alternate function	GPIO port	Remap
GTIM5_ETR	PC5	AF8
	PG2	AF5

Alternate function	GPIO port	Remap
GTIM5_CH1	PA2	AF7
	PE5	AF5
	PF4	AF5
	PH3	AF5
GTIM5_CH2	PA3	AF7
	PD2	AF15
	PE6	AF5
	PF5	AF5
GTIM5_CH3	PH0	AF5
GTIM5_CH4	PH1	AF5

5.2.5.8.6 GTIM6 alternate function remapping

Table 5-15 GTIM6 Alternate Function Remapping

Alternate function	GPIO port	Remap
GTIM6_ETR	PB2	AF13
	PG4	AF5
GTIM6_CH1	PB8	AF6
	PD5	AF5
	PD12	AF2
	PF6	AF2
GTIM6_CH2	PD13	AF2
	PG9	AF5
GTIM6_CH3	PD14	AF2
	PG14	AF3
GTIM6_CH4	PD15	AF2
	PG15	AF3

5.2.5.8.7 GTIM7 alternate function remapping

Table 5-16 GTIM7 Alternate Function Remapping

Alternate function	GPIO port	Remap
GTIM7_ETR	PD9	AF2
GTIM7_CH1	PA4	AF12
	PA7	AF12
	PB9	AF7
	PF7	AF2
	PG10	AF3
GTIM7_CH2	PA5	AF7
	PG0	AF5

Alternate function	GPIO port	Remap
	PG11	AF3
GTIM7_CH3	PC14	AF7
	PG1	AF5
	PG12	AF3
GTIM7_CH4	PC15	AF7
	PE12	AF8
	PG13	AF3

5.2.5.8.8 GTIM8 alternate function remapping

Table 5-17 GTIM8 Alternate Function Remapping

Alternate function	GPIO port	Remap
GTIM8_ETR	PA2	AF14
	PF7	AF5
GTIM8_CH1	PA2	AF12
	PB14	AF3
	PD15	AF3
	PF9	AF5
GTIM8_CH2	PA3	AF12
	PB15	AF7
	PC7	AF0
	PF10	AF5
GTIM8_CH3	PA13	AF13
	PC9	AF0
GTIM8_CH4	PA14	AF13
GTIM8_BKIN	PA9	AF7
	PC5	AF6
GTIM8_CH1N	PA1	AF8
	PB15	AF2
	PG9	AF3

5.2.5.8.9 GTIM9 alternate function remapping

Table 5-18 GTIM9 Alternate Function Remapping

Alternate function	GPIO port	Remap
GTIM9_ETR	PB4	AF11
	PD6	AF7
GTIM9_CH1	PA6	AF2
	PA12	AF7
	PB4	AF13

Alternate function	GPIO port	Remap
	PB8	AF7
	PE0	AF1
	PE10	AF8
	PF8	AF2
GTIM9_CH2	PA7	AF7
	PB14	AF7
GTIM9_CH3	PB12	AF6
	PD9	AF3
	PH2	AF5
GTIM9_CH4	PB8	AF9
	PB13	AF7
	PC10	AF6
	PE7	AF6
GTIM9_BKIN	PB5	AF6
GTIM9_CH1N	PA13	AF6
	PB6	AF6

5.2.5.8.10 GTIM10 alternate function remapping

Table 5-19 GTIM10 Alternate Function Remapping

Alternate function	GPIO port	Remap
GTIM10_ETR	PC11	AF11
GTIM10_CH1	PA7	AF2
	PB5	AF5
	PB9	AF6
	PC0	AF6
	PD8	AF3
	PE1	AF1
	PE15	AF1
	PF9	AF2
GTIM10_CH2	PB13	AF6
	PC1	AF6
	PD9	AF6
GTIM10_CH3	PC2	AF7
	PD11	AF3
GTIM10_CH4	PC3	AF7
	PD14	AF3
GTIM10_BKIN	PA10	AF0
	PB4	AF14
GTIM10_CH1N	PB7	AF5

Alternate function	GPIO port	Remap
	PH4	AF6

5.2.5.9 LPTIMx alternate function remapping

5.2.5.9.1 LPTIM1 alternate function remapping

Table 5-20 LPTIM1 Alternate Function Remapping

Alternate function	GPIO port	Remap
LPTIM1_ETR	PB6	AF14
	PC3	AF5
LPTIM1_IN1	PB5	AF14
	PC0	AF5
LPTIM1_IN2	PB7	AF14
	PC2	AF5
LPTIM1_OUT	PA14	AF10
	PB2	AF9
	PC1	AF5

5.2.5.9.2 LPTIM2 alternate function remapping

Table 5-21 LPTIM2 Alternate Function Remapping

Alternate function	GPIO port	Remap
LPTIM2_ETR	PC13	AF7
LPTIM2_IN1	PB4	AF10
	PE6	AF6
	PH3	AF1
LPTIM2_IN2	PA4	AF13
LPTIM2_OUT	PC4	AF15

5.2.5.10 FDCAN alternate function remapping

FDCAN supports full pin mapping, meaning FDCANx_RX and FDCANx_TX can be mapped to any GPIO port.

5.2.5.10.1 FDCAN1 alternate function remapping

Table 5-22 FDCAN1 Alternate Function Remapping

Alternate function	GPIO port	Remap
FDCAN1_TX	PAx(x=0~15)	AF25
	PBx(x=0~15)	AF25
	PCx(x=0~15)	AF25
	PDx(x=0~15)	AF25
	PEx(x=0~15)	AF25

Alternate function	GPIO port	Remap
	PFx(x=0~15)	AF25
	PGx(x=0~15)	AF25
	PHx(x=0~6)	AF25
FDCAN1_RX	PAx(x=0~15)	AF26
	PBx(x=0~15)	AF26
	PCx(x=0~15)	AF26
	PDx(x=0~15)	AF26
	PEx(x=0~15)	AF26
	PFx(x=0~15)	AF26
	PGx(x=0~15)	AF26
	PHx(x=0~6)	AF26

5.2.5.10.2 FDCAN2 alternate function remapping

Table 5-23 FDCAN2 Alternate Function Remapping

Alternate function	GPIO port	Remap
FDCAN2_TX	PAx(x=0~15)	AF27
	PBx(x=0~15)	AF27
	PCx(x=0~15)	AF27
	PDx(x=0~15)	AF27
	PEx(x=0~15)	AF27
	PFx(x=0~15)	AF27
	PGx(x=0~15)	AF27
	PHx(x=0~6)	AF27
FDCAN2_RX	PAx(x=0~15)	AF28
	PBx(x=0~15)	AF28
	PCx(x=0~15)	AF28
	PDx(x=0~15)	AF28
	PEx(x=0~15)	AF28
	PFx(x=0~15)	AF28
	PGx(x=0~15)	AF28
	PHx(x=0~6)	AF28

5.2.5.10.3 FDCAN3 alternate function remapping

Table 5-24 FDCAN3 Alternate Function Remapping

Alternate function	GPIO port	Remap
FDCAN3_TX	PAx(x=0~15)	AF29
	PBx(x=0~15)	AF29
	PCx(x=0~15)	AF29

Alternate function	GPIO port	Remap
	PDx(x=0~15)	AF29
	PEx(x=0~15)	AF29
	PFx(x=0~15)	AF29
	PGx(x=0~15)	AF29
	PHx(x=0~6)	AF29
FDCAN3_RX	PAx(x=0~15)	AF30
	PBx(x=0~15)	AF30
	PCx(x=0~15)	AF30
	PDx(x=0~15)	AF30
	PEx(x=0~15)	AF30
	PFx(x=0~15)	AF30
	PGx(x=0~15)	AF30
	PHx(x=0~6)	AF30

5.2.5.11 DVP alternate function remapping

Table 5-25 DVP Alternate Function Remapping

Alternate function	GPIO port	Remap
DVP_PIXCLK	PA3	AF10
	PA6	AF10
	PE4	AF6
DVP_HSYNC	PA1	AF9
	PA4	AF10
	PE2	AF6
DVP_VSYNC	PA2	AF10
	PB7	AF12
	PE3	AF6
	PG9	AF9
DVP_D0	PA4	AF6
	PA9	AF10
	PC6	AF8
	PD6	AF8
	PE5	AF7
DVP_D1	PA5	AF5
	PA10	AF10
	PC7	AF8
	PE6	AF7
DVP_D2	PA6	AF14
	PC0	AF9

Alternate function	GPIO port	Remap
	PC8	AF8
	PE0	AF8
	PG10	AF9
DVP_D3	PA7	AF14
	PB2	AF11
	PC9	AF8
	PE1	AF8
	PG11	AF9
DVP_D4	PB10	AF12
	PC4	AF1
	PC11	AF8
	PE4	AF8
	PF12	AF8
DVP_D5	PB6	AF12
	PB11	AF12
	PC5	AF1
	PD3	AF8
	PF13	AF8
DVP_D6	PB0	AF11
	PB8	AF12
	PE5	AF8
	PF14	AF8
DVP_D7	PB1	AF11
	PB9	AF12
	PE6	AF8
	PF15	AF8
DVP_D8	PC10	AF8
DVP_D9	PC12	AF8
DVP_D10	PB5	AF0
DVP_D11	PD2	AF8
	PF10	AF8
DVP_D12	PF11	AF8
	PG6	AF9
	PH3	AF7
DVP_D13	PB14	AF11
	PG7	AF9
	PG15	AF8
	PH2	AF7
DVP_D14	PB15	AF9
	PF6	AF13

Alternate function	GPIO port	Remap
DVP_D15	PC0	AF12
	PF9	AF8

5.2.5.12 FEMC alternate function remapping

Table 5-26 FEMC Alternate Function Remapping

Alternate function	GPIO port	Remap
FEMC_A0	PF0	AF10
	PF10	AF10
FEMC_A1	PF1	AF10
	PF7	AF10
FEMC_A2	PF2	AF10
FEMC_A3	PF3	AF10
FEMC_A4	PF4	AF10
FEMC_A5	PF5	AF10
FEMC_A6	PF12	AF10
FEMC_A7	PF13	AF10
FEMC_A8	PF14	AF10
FEMC_A9	PF15	AF10
FEMC_A10	PG0	AF10
FEMC_A11	PG1	AF10
FEMC_A12	PG2	AF10
FEMC_A13	PG3	AF10
FEMC_A14	PG4	AF10
FEMC_A15	PG5	AF10
FEMC_A16	PC6	AF14
	PD11	AF10
FEMC_A17	PC7	AF14
	PD12	AF10
FEMC_A18	PD13	AF10
FEMC_A19	PE3	AF10
FEMC_A20	PE4	AF10
FEMC_A21	PE5	AF10
FEMC_A22	PE6	AF10
FEMC_A23	PE2	AF10
FEMC_A24	PF8	AF10
	PG13	AF10
FEMC_A25	PF9	AF10
	PG14	AF10
FEMC_D0	PD14	AF10

Alternate function	GPIO port	Remap
FEMC_D1	PD15	AF10
FEMC_D2	PD0	AF10
FEMC_D3	PD1	AF10
FEMC_D4	PE7	AF10
FEMC_D5	PE8	AF10
FEMC_D6	PE9	AF10
FEMC_D7	PE10	AF10
FEMC_D8	PE11	AF10
FEMC_D9	PE12	AF10
FEMC_D10	PE13	AF10
FEMC_D11	PE14	AF10
	PB10	AF14
FEMC_D12	PE15	AF10
	PB11	AF14
FEMC_D13	PD8	AF10
FEMC_D14	PD9	AF10
FEMC_D15	PD10	AF10
FEMC_NE1	PB9	AF0
	PD7	AF11
FEMC_NE2	PB6	AF15
	PG9	AF10
FEMC_NE3	PG8	AF10
	PG10	AF10
FEMC_NE4	PF11	AF10
	PG12	AF10
FEMC_NCE2	PB9	AF9
	PD7	AF10
FEMC_NCE3	PB6	AF10
	PG9	AF11
FEMC_NOE	PC9	AF14
	PD4	AF10
FEMC_NWE	PA10	AF15
	PD5	AF10
FEMC_NADV	PB7	AF13
FEMC_NWAIT	PB8	AF15
	PD6	AF10
FEMC_NBL0	PE0	AF10
FDMC_NBL1	PE1	AF10
FEMC_ALE	PD12	AF11
FEMC_CLE	PD11	AF11

Alternate function	GPIO port	Remap
FEMC_CLK	PD3	AF10
FEMC_INT2	PG6	AF10
FEMC_INT3	PG7	AF10

5.2.5.13 USARTx alternate function remapping

5.2.5.13.1 USART1 alternate function remapping

Table 5-27 USART1 Alternate Function Remapping

Alternate function	GPIO port	Remap
USART1_TX	PA4	AF15
	PA9	AF5
	PA15	AF10
	PB6	AF1
	PC4	AF5
	PE0	AF7
	PG9	AF8
	PH4	AF1
USART1_RX	PA5	AF15
	PA10	AF5
	PB3	AF13
	PB7	AF1
	PC5	AF5
	PE1	AF7
	PG15	AF5
	PH2	AF1
USART1_CK	PA8	AF5
USART1_CTS	PA11	AF5
USART1_RTS_DE	PA12	AF5

5.2.5.13.2 USART2 alternate function remapping

Table 5-28 USART2 Alternate Function Remapping

Alternate function	GPIO port	Remap
USART2_TX	PA2	AF5
	PA14	AF5
	PB3	AF9
	PB4	AF8
	PC8	AF15
	PD5	AF6
	PH1	AF2
USART2_RX	PA3	AF5
	PA15	AF5

Alternate function	GPIO port	Remap
	PB4	AF9
	PB5	AF15
	PC9	AF15
	PD6	AF6
	PH0	AF2
USART2_CK	PA4	AF5
	PB5	AF9
	PD7	AF6
USART2_CTS	PA0	AF5
	PA15	AF14
	PC6	AF10
	PD3	AF6
USART2_RTS_DE	PA1	AF5
	PB3	AF10
	PC7	AF10
	PD4	AF6

5.2.5.13.3 USART3 alternate function remapping

Some signals of USART3 support full pin mapping, as detailed in the table below.

Table 5-29 USART3 Alternate Function Remapping

Alternate function	GPIO port	Remap
USART3_TX	PAx(x=0~15)	AF17
	PBx(x=0~15)	AF17
	PCx(x=0~15)	AF17
	PDx(x=0~15)	AF17
	PEx(x=0~15)	AF17
	PFx(x=0~15)	AF17
	PGx(x=0~15)	AF17
	PHx(x=0~6)	AF17
USART3_RX	PAx(x=0~15)	AF18
	PBx(x=0~15)	AF18
	PCx(x=0~15)	AF18
	PDx(x=0~15)	AF18
	PEx(x=0~15)	AF18
	PFx(x=0~15)	AF18
	PGx(x=0~15)	AF18
	PHx(x=0~6)	AF18
USART3_CK	PB12	AF9
	PC12	AF6
	PD10	AF6
USART3_CTS	PA13	AF5

Alternate function	GPIO port	Remap
	PB13	AF9
	PD11	AF6
USART3_RTS_DE	PAx(x=0~15)	AF16
	PBx(x=0~15)	AF16
	PCx(x=0~15)	AF16
	PDx(x=0~15)	AF16
	PEx(x=0~15)	AF16
	PFx(x=0~15)	AF16
	PGx(x=0~15)	AF16
	PHx(x=0~6)	AF16

5.2.5.13.4 USART4 alternate function remapping

Table 5-30 USART4 Alternate Function Remapping

Alternate function	GPIO port	Remap
USART4_TX	PA11	AF14
	PB0	AF13
	PC0	AF10
	PC6	AF5
	PD11	AF12
	PE2	AF7
	PE10	AF9
	PG14	AF6
USART4_RX	PA12	AF14
	PB1	AF13
	PC1	AF10
	PC7	AF5
	PE3	AF7
	PE11	AF9
	PE15	AF5
	PG9	AF6
USART4_CK	PB14	AF9
	PC8	AF5
	PG7	AF6
USART4_CTS	PG13	AF6
	PG15	AF6
USART4_RTS_DE	PG8	AF6
	PG12	AF6
	PH2	AF2

5.2.5.14 UARTx alternate function remapping

5.2.5.14.1 UART5 alternate function remapping

Some signals of UART5 support full pin mapping, as detailed in the table below.

Table 5-31 UART5 Alternate Function Remapping

Alternate function	GPIO port	Remap
UART5_TX	PAx(x=0~15)	AF20
	PBx(x=0~15)	AF20
	PCx(x=0~15)	AF20
	PDx(x=0~15)	AF20
	PEx(x=0~15)	AF20
	PFx(x=0~15)	AF20
	PGx(x=0~15)	AF20
	PHx(x=0~6)	AF20
UART5_RX	PAx(x=0~15)	AF21
	PBx(x=0~15)	AF21
	PCx(x=0~15)	AF21
	PDx(x=0~15)	AF21
	PEx(x=0~15)	AF21
	PFx(x=0~15)	AF21
	PGx(x=0~15)	AF21
	PHx(x=0~6)	AF21
UART5_CTS	PB5	AF8
UART5_RTS_DE	PAx(x=0~15)	AF19
	PBx(x=0~15)	AF19
	PCx(x=0~15)	AF19
	PDx(x=0~15)	AF19
	PEx(x=0~15)	AF19
	PFx(x=0~15)	AF19
	PGx(x=0~15)	AF19
	PHx(x=0~6)	AF19

5.2.5.14.2 UART6 alternate function remapping

Table 5-32 UART6 Alternate Function Remapping

Alternate function	GPIO port	Remap
UART6_TX	PA0	AF11
	PA13	AF12
	PB2	AF12
	PC10	AF7
	PD0	AF6
UART6_RX	PA1	AF11
	PA14	AF12

Alternate function	GPIO port	Remap
	PC11	AF7
	PD1	AF6
	PE7	AF8
UART6_CTS	PB7	AF8
UART6_RTS_DE	PA15	AF11

5.2.5.14.3 UART7 alternate function remapping

Table 5-33 UART6 Alternate Function Remapping

Alternate function	GPIO port	Remap
UART7_TX	PA2	AF11
	PB11	AF8
	PC1	AF7
	PC2	AF13
	PC4	AF13
	PE8	AF9
	PF7	AF12
	PG0	AF8
	PG7	AF8
UART7_RX	PA3	AF11
	PB10	AF8
	PC0	AF7
	PC3	AF13
	PC5	AF13
	PE7	AF9
	PF6	AF12
	PG1	AF8
	PG8	AF8
UART7_CTS	PA6	AF11
	PB13	AF8
	PG5	AF8
UART7_RTS_DE	PB1	AF9
	PB12	AF8
	PG6	AF8

5.2.5.14.4 UART8 alternate function remapping

Some signals of UART8 support full pin mapping, as detailed in the table below.

Table 5-34 UART8 Alternate Function Remapping

Alternate function	GPIO port	Remap
UART8_TX	PAx(x=0~15)	AF23
	PBx(x=0~15)	AF23
	PCx(x=0~15)	AF23

Alternate function	GPIO port	Remap
	PDx(x=0~15)	AF23
	PEx(x=0~15)	AF23
	PFx(x=0~15)	AF23
	PGx(x=0~15)	AF23
	PHx(x=0~6)	AF23
UART8_RX	PAx(x=0~15)	AF24
	PBx(x=0~15)	AF24
	PCx(x=0~15)	AF24
	PDx(x=0~15)	AF24
	PEx(x=0~15)	AF24
	PFx(x=0~15)	AF24
	PGx(x=0~15)	AF24
	PHx(x=0~6)	AF24
UART8_CTS	PB15	AF8
	PF0	AF6
	PH3	AF2
UART8_RTS_DE	PAx(x=0~15)	AF22
	PBx(x=0~15)	AF22
	PCx(x=0~15)	AF22
	PDx(x=0~15)	AF22
	PEx(x=0~15)	AF22
	PFx(x=0~15)	AF22
	PGx(x=0~15)	AF22
	PHx(x=0~6)	AF22

5.2.5.15 I2C alternate function remapping

5.2.5.15.1 I2C1 alternate function remapping

Table 5-35 I2C1 Alternate Function Remapping

Alternate function	GPIO port	Remap
I2C1_SCL	PA9	AF15
	PA13	AF8
	PA15	AF9
	PB6	AF2
	PB8	AF2
	PD11	AF5
	PG14	AF7
I2C1_SDA	PA14	AF11
	PB7	AF2
	PB9	AF13
	PD1	AF7
	PE15	AF9

Alternate function	GPIO port	Remap
	PG15	AF7
I2C1_SMBA	PB5	AF12

5.2.5.15.2 I2C2 alternate function remapping

Table 5-36 I2C2 Alternate Function Remapping

Alternate function	GPIO port	Remap
I2C2_SCL	PA4	AF11
	PA9	AF11
	PB10	AF7
	PC4	AF7
	PF1	AF7
	PF6	AF7
	PG2	AF6
	PH1	AF7
I2C2_SDA	PA5	AF11
	PA8	AF11
	PA10	AF11
	PB3	AF12
	PB11	AF7
	PC12	AF9
	PF0	AF7
	PG3	AF6
	PH0	AF7
I2C2_SMBA	PA8	AF8
	PA10	AF8
	PB12	AF7
	PF2	AF7

5.2.5.15.3 I2C3 alternate function remapping

Table 5-37 I2C3 Alternate Function Remapping

Alternate function	GPIO port	Remap
I2C3_SCL	PA8	AF7
	PC0	AF8
	PC4	AF14
	PC8	AF7
	PF3	AF7
	PF4	AF6
	PG7	AF7

Alternate function	GPIO port	Remap
I2C3_SDA	PB4	AF12
	PB5	AF11
	PC1	AF9
	PC5	AF14
	PC9	AF7
	PC11	AF9
	PF4	AF7
	PF5	AF7
	PG8	AF7
I2C3_SMBA	PA9	AF8
	PB2	AF7
	PG6	AF7

5.2.5.15.4 I2C4 alternate function remapping

Table 5-38 I2C4 Alternate Function Remapping

Alternate function	GPIO port	Remap
I2C4_SCL	PA9	AF13
	PA13	AF7
	PC6	AF7
	PD14	AF9
	PF14	AF7
	PG3	AF7
I2C4_SDA	PA10	AF6
	PB7	AF9
	PC7	AF7
	PD15	AF9
	PF15	AF7
	PG4	AF7
I2C4_SMBA	PA14	AF8
	PD11	AF9
	PF13	AF7

5.2.5.16 SPI/I2S alternate function remapping

5.2.5.16.1 SPI1 alternate function remapping

Table 5-39 SPI1 Alternate Function Remapping

Alternate function	GPIO port	Remap
SPI1_NSS	PA4	AF1
	PA15	AF2
	PB2	AF14
	PG5	AF1
SPI1_SCK	PA5	AF1

Alternate function	GPIO port	Remap
	PB3	AF2
	PE7	AF1
	PG2	AF1
SPI1_MISO	PA6	AF1
	PB4	AF2
	PE8	AF1
	PG3	AF1
SPI1_MOSI	PA7	AF1
	PB5	AF2
	PE9	AF1
	PG4	AF1

5.2.5.16.2 SPI2/I2S2 alternate function remapping

Table 5-40 SPI2/I2S2 Alternate Function Remapping

Alternate function	GPIO port	Remap
SPI2_NSS	PA12	AF9
	PD15	AF1
SPI2_I2S2_NSS_WS	PB9	AF2
	PB12	AF1
	PC6	AF13
	PD1	AF1
	PE10	AF1
	PH0	AF1
SPI2_SCK	PF9	AF1
	PF10	AF1
SPI2_I2S2_SCK_CK	PA9	AF12
	PB10	AF2
	PB13	AF1
	PC7	AF13
	PD3	AF1
	PE11	AF6
	PH1	AF1
SPI2_MISO	PA10	AF1
	PB14	AF1
	PC2	AF1
	PC8	AF13
	PE12	AF6
	PG9	AF7
	PG14	AF9
SPI2_MOSI	PD2	AF11
	PG9	AF2

Alternate function	GPIO port	Remap
SPI2_I2S2_MOSI_SD	PA11	AF1
	PB15	AF1
	PC1	AF2
	PC3	AF1
	PC9	AF13
	PE13	AF6
I2S2_MCK	PA3	AF9
	PA6	AF12
	PA8	AF1
	PC6	AF1
I2S2_AUX_SD	PA10	AF13
	PB14	AF2
	PC2	AF2

5.2.5.16.3 SPI3/I2S3 alternate function remapping

Table 5-41 SPI3/I2S3 Alternate Function Remapping

Alternate function	GPIO port	Remap
SPI3_I2S3_NSS_WS	PA4	AF2
	PA15	AF1
	PC2	AF12
	PD2	AF1
	PD8	AF1
SPI3_SCK	PG9	AF1
SPI3_I2S3_SCK_CK	PB3	AF1
	PC3	AF2
	PC10	AF1
	PD9	AF1
SPI3_MISO	PA0	AF12
	PB4	AF1
	PC11	AF1
	PD11	AF1
SPI3_MOSI	PD0	AF2
SPI3_I2S3_MOSI_SD	PA1	AF12
	PB0	AF9
	PB2	AF1
	PB5	AF1
	PC1	AF1
	PC12	AF1
	PD6	AF1
	PD12	AF1
I2S3_MCK	PA9	AF1

Alternate function	GPIO port	Remap
	PC7	AF1
I2S3_AUX_SD	PB4	AF15
	PC11	AF3

5.2.5.16.4 I2S alternate function remapping

I2S_CKIN is the I2S external reference clock input pin, applicable to both I2S2 and I2S3.

Table 5-42 I2S Alternate Function Remapping

Alternate function	GPIO port	Remap
I2S_CKIN	PA2	AF15
	PA12	AF1
	PB11	AF5
	PC9	AF1

5.2.5.16.5 SPI4 alternate function remapping

Table 5-43 SPI4 Alternate Function Remapping

Alternate function	GPIO port	Remap
SPI4_ NSS	PB12	AF12
	PE3	AF1
	PE4	AF1
	PE11	AF1
	PG14	AF8
SPI4 _SCK	PB13	AF12
	PE2	AF1
	PE12	AF1
	PG11	AF8
SPI4_MISO	PA11	AF13
	PD0	AF1
	PE5	AF1
	PE13	AF1
	PG12	AF8
SPI4_ MOSI	PA1	AF13
	PA12	AF13
	PE6	AF1
	PE14	AF1
	PG13	AF8

5.2.5.16.6 SPI5 alternate function remapping

Table 5-44 SPI5 Alternate Function Remapping

Alternate function	GPIO port	Remap
SPI5_ NSS	PB1	AF14

Alternate function	GPIO port	Remap
	PE11	AF5
	PF6	AF11
SPI5 _SCK	PB0	AF8
	PE12	AF5
	PF7	AF11
SPI5_MISO	PE13	AF5
	PF8	AF11
SPI5_ MOSI	PA10	AF14
	PB8	AF0
	PE14	AF5
	PF9	AF11
	PF11	AF11

5.2.5.16.7 SPI6 alternate function remapping

Table 5-45 SPI5 Alternate Function Remapping

Alternate function	GPIO port	Remap
SPI6_ NSS	PA2	AF6
	PG15	AF1
SPI6 _SCK	PA1	AF14
SPI6_MISO	PA4	AF14
SPI6_ MOSI	PA5	AF14

5.2.5.17 SDIO alternate function remapping

Table 5-46 SDIO Alternate Function Remapping

Alternate function	GPIO port	Remap
SDIO_D0	PB4	AF6
	PC8	AF2
	PE8	AF2
SDIO_D1	PA8	AF14
	PB0	AF10
	PC9	AF2
	PE9	AF2
SDIO_D2	PA9	AF14
	PB1	AF10
	PC10	AF2
	PE10	AF2
SDIO_D3	PC11	AF2
	PE11	AF2
SDIO_D4	PB8	AF1
SDIO_D5	PB9	AF1

Alternate function	GPIO port	Remap
SDIO_D6	PC6	AF2
SDIO_D7	PB10	AF1
	PC7	AF2
SDIO_CLK	PB2	AF10
	PC12	AF2
	PE12	AF2
SDIO_CMD	PA6	AF13
	PD2	AF2
	PE13	AF2

5.2.5.18 XSPI alternate function remapping

Table 5-47 XSPI Alternate Function Remapping

Alternate function	GPIO port	Remap
XSPI_NSS0	PA2	AF8
	PB11	AF6
	PD3	AF9
	PE11	AF7
XSPI_NSS1	PA4	AF9
	PC10	AF13
	PF0	AF9
	PG8	AF5
XSPI_CLK	PA3	AF8
	PA5	AF9
	PB10	AF6
	PC11	AF13
	PE10	AF7
	PF1	AF9
	PF10	AF9
	PG13	AF5
XSPI_IO0	PA5	AF8
	PA6	AF15
	PB1	AF2
	PC12	AF13
	PE12	AF7
	PF2	AF9
	PF8	AF9
	PG14	AF5
XSPI_IO1	PA7	AF15
	PB0	AF2
	PD0	AF9

Alternate function	GPIO port	Remap
	PE13	AF7
	PF3	AF9
	PF9	AF9
	PG12	AF5
XSPI_IO2	PA7	AF8
	PC4	AF9
	PD1	AF9
	PE14	AF7
	PF4	AF9
	PF7	AF9
	PG10	AF5
XSPI_IO3	PA6	AF8
	PC5	AF9
	PD2	AF9
	PE15	AF7
	PF5	AF9
	PF6	AF9
	PG11	AF5
XSPI_IO4	PC1	AF8
	PD4	AF9
XSPI_IO5	PB2	AF2
	PC2	AF8
	PD5	AF9
XSPI_IO6	PC3	AF8
	PD6	AF9
XSPI_IO7	PC4	AF8
	PD7	AF9
XSPI_RXDS	PC0	AF11
	PC13	AF8
	PD13	AF8

5.2.5.19 ETH alternate function remapping

Table 5-48 ETH Alternate Function Remapping

Alternate function	GPIO port	Remap
ETH_ MDC	PC1	AF11
ETH_MDIO	PA2	AF1
ETH_PPS_OUT	PB5	AF13
	PB6	AF13
	PG8	AF1
ETH_MII_TXD3	PB7	AF15

Alternate function	GPIO port	Remap
	PB8	AF13
	PE2	AF8
ETH_MII_TXD2	PC2	AF11
ETH_MII_TXD1	PB13	AF13
	PG14	AF1
ETH_MII_TXD0	PB12	AF13
	PG13	AF1
ETH_MII_TX_CLK	PC3	AF11
ETH_MII_TX_EN	PB11	AF1
	PG11	AF1
ETH_MII_RXD3	PB1	AF1
	PD12	AF8
ETH_MII_RXD2	PB0	AF1
	PD11	AF8
ETH_MII_RXD1	PC5	AF11
	PD10	AF8
ETH_MII_RXD0	PC4	AF11
	PD9	AF8
ETH_MII_RX_CLK	PA1	AF1
ETH_MII_RX_DV	PA7	AF10
	PD8	AF8
ETH_MII_RX_ER	PB10	AF13
ETH_MII_COL	PA3	AF1
ETH_MII_CRS	PA0	AF1
ETH_RMII_TXD1	PB13	AF2
	PG14	AF2
ETH_RMII_TXD0	PB12	AF2
	PG13	AF2
ETH_RMII_TX_EN	PB11	AF2
	PG11	AF2
ETH_RMII_RXD1	PC5	AF10
	PD10	AF9
ETH_RMII_RXD0	PC4	AF10
	PD9	AF9
ETH_RMII_REF_CLK	PA1	AF7
ETH_RMII_CRS_DV	PA7	AF11
	PD8	AF9

5.2.5.20 RTC alternate function remapping

Table 5-49 RTC Alternate Function Remapping

Alternate function	GPIO port	Remap
RTC_REFIN	PA1	AF10
	PB15	AF12

5.2.5.21 USB_HS alternate function remapping

Table 5-50 USB_HS Alternate Function Remapping

Alternate function	GPIO port	Remap
USB_HS_ID	PB12	AF14
USB_HS_SOF	PA4	AF8

5.2.5.22 EVENT alternate function remapping

EVENT supports full pin mapping, as detailed in the table below.

Table 5-51 EVENT Alternate Function Remapping

Alternate function	GPIO port	Remap
EVENT_OUT	PAx(x=0~15)	AF4
	PBx(x=0~15)	AF4
	PCx(x=0~15)	AF4
	PDx(x=0~15)	AF4
	PEx(x=0~15)	AF4
	PFx(x=0~15)	AF4
	PGx(x=0~15)	AF4
	PHx(x=0~6)	AF4

5.2.5.23 COMP alternate function remapping

5.2.5.23.1 COMP1 alternate function remapping

Table 5-52 COMP1 Alternate Function Remapping

Alternate function	GPIO port	Remap
COMP1_OUT	PA0	AF9
	PA6	AF9
	PA11	AF8
	PB1	AF12
	PB8	AF11
	PF4	AF8

5.2.5.23.2 COMP2 alternate function remapping

Table 5-53 COMP2 Alternate Function Remapping

Alternate function	GPIO port	Remap
COMP2_OUT	PA2	AF9

Alternate function	GPIO port	Remap
	PA6	AF7
	PA7	AF9
	PA12	AF8
	PB9	AF11

5.2.5.23.3 COMP3 alternate function remapping

Table 5-54 COMP3 Alternate Function Remapping

Alternate function	GPIO port	Remap
COMP3_OUT	PA2	AF13
	PA8	AF9
	PB7	AF11
	PB10	AF11
	PB15	AF11
	PC2	AF9
	PC8	AF9
	PC10	AF9

5.2.5.23.4 COMP4 alternate function remapping

Table 5-55 COMP4 Alternate Function Remapping

Alternate function	GPIO port	Remap
COMP4_OUT	PB1	AF8
	PB6	AF11
	PB12	AF11
	PB14	AF12
	PC5	AF15
	PC11	AF10

5.2.5.23.5 COMP5 alternate function remapping

Table 5-56 COMP5 Alternate Function Remapping

Alternate function	GPIO port	Remap
COMP5_OUT	PA9	AF9
	PA11	AF15
	PB0	AF12
	PB6	AF0
	PB11	AF11
	PC7	AF9

5.2.5.23.6 COMP6 alternate function remapping

Table 5-57 COMP6 Alternate Function Remapping

Alternate function	GPIO port	Remap
COMP6_OUT	PA10	AF9

Alternate function	GPIO port	Remap
	PA12	AF15
	PB7	AF0
	PC6	AF9
	PC9	AF9

5.2.5.23.7 COMP7 alternate function remapping

Table 5-58 COMP7 Alternate Function Remapping

Alternate function	GPIO port	Remap
COMP7_OUT	PA8	AF12
	PC2	AF10
	PC8	AF10
	PD12	AF9

5.2.6 IO Configuration Of Peripherals

Table 5-59 ADC/DAC

ADC/DAC Pin	GPIO Configuration
ADC	Analog mode
DAC	Analog mode

Table5-60 SHRTIM

SHRTIM Pin	Configuration	GPIO Configuration
SHRTIM1_CHx1(x=A~F)	Output compare channel x	Push-pull alternate output
SHRTIM1_CHx2(x=A~F)	Complementary output channel x	Push-pull alternate output
SHRTIM1_FALTx(x=1~6)	Fault input x	Alternate input mode
SHRTIM1_SCIN	Synchronous signal input	Alternate input mode
SHRTIM1_SCOUT	Synchronous signal output	Push-pull alternate output
SHRTIM1_EXEVx(x=1~10)	External event input x	Alternate input mode

Table 5-61 ATIM1/2/3

ATIM Pin	Configuration	GPIO Configuration
ATIM1/2/3_CHx(x=1~4)	Input capture channel x	Alternate input mode
	Output compare channel x	Push-pull alternate output
ATIM1/2/3_CHxN(x=1~4)	Complementary output channel x	Push-pull alternate output
ATIM1/2/3_BKIN	Brake input	Alternate input mode
ATIM1/2/3_BKIN2	Bidirectional brake	Open-drain alternate output
ATIM1/2/3_ETR	External trigger clock input	Alternate input mode

Table 5-62 GTIM1~7

GTIM pin	Configuration	GPIO Configuration
GTIM1~7_CHx(x=1~4)	Input capture channel x	Alternate input mode
	Output compare channel x	Push-pull alternate output
GTIM1~7_ETR	External trigger clock input	Alternate input mode

Table 5-63 GTIM8/9/10

GTIM pin	Configuration	GPIO Configuration
GTIM8/9/10_CHx(x=1~4)	Input capture channel x	Alternate input mode
	Output compare channel x	Push-pull alternate output
GTIM8/9/10_CH1N	Complementary output channel 1	Push-pull alternate output
GTIM8/9/10_BKIN	Brake input	Alternate input mode
	Bidirectional brake	Open-drain alternate output
GTIM8/9/10_ETR	External trigger clock input	Alternate input mode

Table 5-64 LPTIM1/2

LPTIM Pin	Configuration	GPIO Configuration
LPTIM1/2_INx(x=1，2)	Input channel x	Alternate input mode
LPTIM1/2_OUT	PWM output	Push-pull alternate output
LPTIM1/2_ETR	External trigger input	Alternate input mode

Table 5-65 FDCAN

FDCAN Pin	GPIO Configuration
FDCAN_TX	Push-pull alternate output
FDCAN_RX	Alternate input mode

Table 5-66 DVP

DVP Pin	GPIO Configuration
DVP_HSYNC	Alternate input mode
DVP_VSYNC	Alternate input mode
DVP_PCLK	Alternate input mode
DVP_Dx(x=0~7)	Alternate input mode

Table 5-67 FEMC

FEMC Pin	GPIO Configuration
FEMC_Ax(x=0~25)	Push-pull alternate output
FEMC_Dx(x=0~15)	Push-pull alternate output
FEMC_CLK	Push-pull alternate output
FEMC_NOE	Push-pull alternate output
FEMC_NWE	Push-pull alternate output
FEMC_NE(x=1~4)	Push-pull alternate output
FEMC_NCE(x=2~3)	Push-pull alternate output
FEMC_ALE	Push-pull alternate output

FEMC Pin	GPIO Configuration
FEMC_CLE	Push-pull alternate output
FEMC_NBLx(x=0~1)	Push-pull alternate output
FEMC_NADV	Push-pull alternate output
FEMC_NWAIT	Alternate input mode
FEMC_INTx(x=2~3)	Alternate input mode

Table 5-68 U(S)ART

U(S)ART Pin	Configuration	GPIO Configuration
U(S)ARTx_TX	Full duplex transmissions	Push-pull alternate output
	Half duplex synchronous mode	Push-pull alternate output + Pull-up
	Single-wire mode	Alternate input mode
U(S)ARTx_RX	Full duplex transmissions	Alternate input mode
	Half duplex synchronous mode	Unused, can be used as general I/O.
USARTx_CK	Synchronous mode	Push-pull alternate output
U(S)ARTx_RTS_DE	Hardware flow control	Push-pull alternate output
U(S)ARTx_CTS	Hardware flow control	Alternate input mode

Table 5-69 I2C

I2C Pin	Configuration	GPIO Configuration
I2Cx_SCL	I2C clock	Open-drain alternate output
I2Cx_SDA	I2C data	Open-drain alternate output
I2Cx_SMBA	SMBA data	Push-pull alternate output

Table 5-70 SPI

SPI Pin	Configuration	GPIO Configuration
SPIx_SCK	Master mode	Push-pull alternate output
	Slave mode	Alternate input mode
SPIx_MOSI	Full-duplex mode / Master mode	Push-pull alternate output
	Full-duplex mode / Slave mode	Alternate input mode
	Simplex bidirectional data line / Master mode	Push-pull alternate output
	Simplex bidirectional data line / Slave mode	Unused, can be used as general I/O.
SPIx_MISO	Full-duplex mode / Master mode	Push-pull alternate output
	Full-duplex mode / Slave mode	Alternate input mode
	Simplex bidirectional data line / Master mode	Push-pull alternate output
	Simplex bidirectional data line / Slave mode	Unused, can be used as general I/O.
SPIx_NSS	Hardware slave mode	Alternate input mode
	Hardware master mode / NSS output enable	Push-pull alternate output
	Software mode	Unused, can be used as general I/O.

Table 5-71 I2S

I2S Pin	Configuration	GPIO Configuration
I2Sx_WS	Master mode	Push-pull alternate output
	Slave mode	Alternate input mode
I2Sx_CK	Master mode	Push-pull alternate output
	Slave mode	Alternate input mode
I2Sx_SD	Transmitter	Push-pull alternate output
	Receiver	Alternate input mode
I2Sx_AUX_SD	Transmitter	Alternate input mode
	Receiver	Push-pull alternate output
I2Sx_MCK	Master mode	Push-pull alternate output
	Slave mode	Unused, can be used as general I/O.
I2S_CKIN	-	Alternate input mode

Table 5-72 SDIO

SDIO Pin	GPIO Configuration
SDIO_CLK	Push-pull alternate output
SDIO_CMD	Push-pull alternate output
SDIO_DATx(x=0~7)	Push-pull alternate output

Table 5-73 XSPI

XSPI Pin	Configuration	GPIO Configuration
XSPI_IOx(x=0~7)	Master /Slave mode	Push-pull alternate output
XSPI_CLK	Master mode	Push-pull alternate output
	Slave mode	Alternate input mode
XSPI_NSS	Master mode	Push-pull alternate output
	Slave mode	Alternate input mode
XSPI_RXDS	Master mode	Alternate input mode

Table 5-74 ETH

ETH Pin	GPIO Configuration
ETH_MDC	Push-pull alternate output + fast toggle
ETH_MDIO	Push-pull alternate output + fast toggle
ETH_PPS_OUT	Push-pull alternate output + fast toggle
ETH_MII_CRS	Alternate input mode
ETH_MII_COL	Alternate input mode
ETH_MII_RX_ER	Alternate input mode
ETH_MII_RX_DV ETH_RMII_CRS_DV	Alternate input mode
ETH_MII_RXD3	Alternate input mode
ETH_MII_RXD2	Alternate input mode
ETH_MII_RXD1 ETH_RMII_RXD1	Alternate input mode
ETH_MII_RXD0 ETH_RMII_RXD0	Alternate input mode
ETH_MII_RX_CLK ETH_RMII_REF_CLK	Alternate input mode
ETH_MII_TX_EN ETH_RMII_TX_EN	Push-pull alternate output + fast toggle
ETH_MII_TX_CLK	Alternate input mode
ETH_MII_TXD3	Push-pull alternate output + fast toggle
ETH_MII_TXD2	Push-pull alternate output + fast toggle
ETH_MII_TXD1 ETH_RMII_TXD1	Push-pull alternate output + fast toggle
ETH_MII_TXD0 ETH_RMII_TXD0	Push-pull alternate output + fast toggle

Table 5-75 USB FS Device

USB Pin	GPIO Configuration
USB_DM USB_DP	Once the USB module is enabled, these pins are automatically connected to the internal USB transceiver

Table 5-76 USB HS Dualrole

USB Pin	GPIO Configuration
USB_HS_DM USB_HS_DP	Analog mode
USB_HS_ID	Alternate input mode
USB_HS_SOF	Push-pull alternate output

Table 5-77 Other

pin	Alternate function	GPIO configuration
RTC_REFIN	RTC reference clock input	Alternate input mode

pin	Alternate function	GPIO configuration
Timestamp	Timestamp trigger input	Alternate input mode
MCO	clock output	Push-pull alternate output
EXTI Input Line	External interrupt input	Input mode
COMPx_OUT	COMP output	Push-pull alternate output
EVENT_OUT	Event output	Push-pull alternate output

5.2.7 GPIO locking mechanism

The locking mechanism is used to freeze the IO configuration to prevent accidental changes. When a lock (LOCK) procedure is performed on a port bit, the configuration of the port cannot be changed until the next reset, refer to the port configuration lock register GPIOx_PLOCK.

- PLOCKK, that is, GPIOx_PLOCK [16], becomes 1 only after the correct sequence w1-> w0-> w1-> r0 (r0 here is also a must). After that, it becomes 0 only if the system reset is performed.
- GPIOx_PLOCK.PLOCK[15:0] can only be modified at GPIOx_PLOCK.PLOCKK=0.
- The lock sequence to set GPIOx_PLOCK.PLOCKK bit, w1-> w0-> w1-> r0 will be valid only if the value (1 or 0) in GPIOx_PLOCK.PLOCK [15:0] does not change during this sequence.The GPIOx_PLOCK.PLOCKK bit will not be set if the value in GPIOx_PLOCK.PLOCK [15:0] changes during this sequence.
- As long as GPIOx_PLOCK.PLOCKK=0 and GPIOx_PLOCK.PLOCKx=0 or 1, all configuration and alternate function bits can be modified. When GPIOx_PLOCK.PLOCKK=1 but GPIOx_PLOCK.PLOCK[x]=0, the corresponding configuration and alternate function bits corresponding to GPIOx_PLOCK.PLOCK[x]=0 can be modified.
- Only when GPIOx_PLOCK.PLOCKK=1 and GPIOx_PLOCK.PLOCK[x]=1, the configurations corresponding to GPIOx_PLOCK.PLOCK[x]=1 are locked and can not be modified.
- If the lock sequence operation is wrong, then it must be redone (w1-> w0-> w1-> r0) to initiate the lock operation again.

5.3 GPIO Registers

These peripheral registers must be operated as 32-bit words.

5.3.1 GPIO Registers Overview

GPIOA base address：0x4002C000

GPIOB base address：0x4002C400

GPIOC base address：0x4002C800

GPIOD base address：0x4002CC00

GPIOE base address：0x4002D000

GPIOF base address：0x4002D400

GPIOG base address：0x4002D800

GPIOH base address：0x4002DC00

Table 5-78 GPIO Registers Overview

Offset	Register		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
000h	GPIOx_PMODE		PMODE15[1:0]		PMODE14[1:0]		PMODE13[1:0]		PMODE12[1:0]		PMODE11[1:0]		PMODE10[1:0]		PMODE9[1:0]		PMODE8[1:0]		PMODE7[1:0]		PMODE6[1:0]		PMODE5[1:0]		PMODE4[1:0]		PMODE3[1:0]		PMODE2[1:0]		PMODE1[1:0]		PMODE0[1:0]	
	Reset Value	x=A	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
		x=B	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	1	1	1	1	1	1
		x=C,D,E,F,G	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
		x=H	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
004h	GPIOx_PUPD		PUPD15[1:0]		PUPD14[1:0]		PUPD13[1:0]		PUPD12[1:0]		PUPD11[1:0]		PUPD10[1:0]		PUPD9[1:0]		PUPD8[1:0]		PUPD7[1:0]		PUPD6[1:0]		PUPD5[1:0]		PUPD4[1:0]		PUPD3[1:0]		PUPD2[1:0]		PUPD1[1:0]		PUPD0[1:0]	
	Reset Value	x=A	0	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		x=B	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
		x=C,D,E,F,G,H	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
008h	GPIOx_POTYPE		Reserved																POT15	POT14	POT13	POT12	POT11	POT10	POT9	POT8	POT7	POT6	POT5	POT4	POT3	POT2	POT1	POT0
	Reset Value																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
00Ch	GPIOx_AFSEL0		Reserved			AFSEL3[4:0]					Reserved			AFSEL2[4:0]					Reserved			AFSEL1[4:0]					Reserved			AFSEL0[4:0]				
	Reset Value					0	0	0	0	0				0	0	0	0	0				0	0	0	0	0				0	0	0	0	0
010h	GPIOx_AFSEL1		Reserved			AFSEL74:0]					Reserved			AFSEL6[4:0]					Reserved			AFSEL5[4:0]					Reserved			AFSEL4[4:0]				
	Reset Value					0	0	0	0	0				0	0	0	0	0				0	0	0	0	0				0	0	0	0	0
014h	GPIOx_AFSEL2		Reserved			AFSEL11[4:0]					Reserved			AFSEL10[4:0]					Reserved			AFSEL9[4:0]					Reserved			AFSEL8[4:0]				
	Reset Value					0	0	0	0	0				0	0	0	0	0				0	0	0	0	0				0	0	0	0	0
018h	GPIOx_AFSEL3		Reserved			AFSEL15[4:0]					Reserved			AFSEL14[4:0]					Reserved			AFSEL13[4:0]					Reserved			AFSEL12[4:0]				
	Reset Value					0	0	0	0	0				0	0	0	0	0				0	0	0	0	0				0	0	0	0	0
01Ch	GPIOx_DS		DS15[1:0]		DS14[1:0]		DS13[1:0]		DS12[1:0]		DS11[1:0]		DS10[1:0]		DS9[1:0]		DS8[1:0]		DS7[1:0]		DS6[1:0]		DS5[1:0]		DS4[1:0]		DS3[1:0]		DS2[1:0]		DS1[1:0]		DS0[1:0]	
	Reset Value		0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
020h	GPIOx_SR		Reserved																SR15	SR14	SR13	SR12	SR11	SR10	SR9	SR8	SR7	SR6	SR5	SR4	SR3	SR2	SR1	SR0
	Reset Value																		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
024h	GPIOx_PBSC		PBC15	PBC14	PBC13	PBC12	PBC11	PBC10	PBC9	PBC8	PBC7	PBC6	PBC5	PBC4	PBC3	PBC2	PBC1	PBC0	PBS15	PBS14	PBS13	PBS12	PBS11	PBS10	PBS9	PBS8	PBS7	PBS6	PBS5	PBS4	PBS3	PBS2	PBS1	PBS0
	Reset Value		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
028h	GPIOx_PBC		Reserved																PBC15	PBC14	PBC13	PBC12	PBC11	PBC10	PBC9	PBC8	PBC7	PBC6	PBC5	PBC4	PBC3	PBC2	PBC1	PBC0
	Reset Value																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
02Ch	GPIOx_PID		Reserved																PID15	PID14	PID13	PID12	PID11	PID10	PID9	PID8	PID7	PID6	PID5	PID4	PID3	PID2	PID1	PID0
	Reset Value																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
030h	GPIOx_POD		Reserved																POD15	POD14	POD13	POD12	POD11	POD10	POD9	POD8	POD7	POD6	POD5	POD4	POD3	POD2	POD1	POD0
	Reset Value	x=A																	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
		x=B																	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
		x=C,D,E,F,G,H																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
034h	GPIOx_PLOCK		Reserved															PLOCKK	PLOCK15	PLOCK14	PLOCK13	PLOCK12	PLOCK11	PLOCK10	PLOCK9	PLOCK8	PLOCK7	PLOCK6	PLOCK5	PLOCK4	PLOCK3	PLOCK2	PLOCK1	PLOCK0
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

5.3.2 GPIO Mode Description Register (GPIOx_PMODE)

Address offset: 0x00

Reset value: 0x03FF FFFF（x=A）；0xFFFF FC3F（x=B）；0xFFFFFFFF（x=C,D,E,F,G），0x0000 3FFF（x=H）

31 30	29 28	27 26	25 24	23 22	21 20	19 18	17 16
PMODE15[1:0]	PMODE14[1:0]	PMODE13[1:0]	PMODE12[1:0]	PMODE11[1:0]	PMODE10[1:0]	PMODE9[1:0]	PMODE8[1:0]
rw	rw	rw	rw	rw	rw	rw	rw

15 14	13 12	11 10	9 8	7 6	5 4	3 2	1 0
PMODE7[1:0]	PMODE6[1:0]	PMODE5[1:0]	PMODE4[1:0]	PMODE3[1:0]	PMODE2[1:0]	PMODE1[1:0]	PMODE0[1:0]
rw	rw	rw	rw	rw	rw	rw	rw

Bit Field	Name	Description
31:30	PMODEy[1:0]	Mode bits for port y (y = 0…15)
29:28		00: Input mode
27:26		01: General output mode
25:24		10: Alternate function mode
23:22		11: Analog function mode (state after reset)
21:20		
19:18		
17:16		
15:14		
13:12		
11:10		
9:8		
7:6		
5:4		
3:2		
1:0		

5.3.3 GPIO Pull-Up/Pull-Down Description Register (GPIOx_PUPD)

Address offset: 0x04

Reset value: 0x6400 0000（x=A）；0x0000 0100（x=B）；0x0000 0000（x=C,D,E,F,G,H）

31 30	29 28	27 26	25 24	23 22	21 20	19 18	17 16
PUPD15[1:0]	PUPD14[1:0]	PUPD13[1:0]	PUPD12[1:0]	PUPD11[1:0]	PUPD10[1:0]	PUPD9[1:0]	PUPD8[1:0]
rw	rw	rw	rw	rw	rw	rw	rw

15 14	13 12	11 10	9 8	7 6	5 4	3 2	1 0
PUPD7[1:0]	PUPD6[1:0]	PUPD5[1:0]	PUPD4[1:0]	PUPD3[1:0]	PUPD2[1:0]	PUPD1[1:0]	PUPD0[1:0]
rw	rw	rw	rw	rw	rw	rw	rw

Bit Field	Name	Description
31:30	PUPDy[1:0]	Mode bits for port y (y = 0…15)
29:28		00: No pull-up, pull-down
27:26		01: Pull-up
25:24		10: Pull-down
23:22		11: Reserved

Bit Field	Name	Description
21:20 19:18 17:16 15:14 13:12 11:10 9:8 7:6 5:4 3:2 1:0		

5.3.4 GPIO Output Type Definition (GPIOx_POTYPE)

Address offset: 0x08

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
POT15	POT14	POT13	POT12	POT11	POT10	POT9	POT8	POT7	POT6	POT5	POT4	POT3	POT2	POT1	POT0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit Field	Name	Description
31:16	Reserved	Reserved, the reset value must be maintained.
15:0	POTy	Output mode bits for port y (y = 0…15) 0: Output push-pull mode (state after reset) 1: Output open-drain mode

5.3.5 GPIO Alternate Function Register 0 (GPIOx_AFSEL0)

Address offset: 0x0C

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved			AFSEL3[4:0]					Reserved			AFSEL2[4:0]				
			rw								rw				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			AFSEL1[4:0]					Reserved			AFSEL0[4:0]				
			rw								rw				

Bit Field	Name	Description
31:29 23:21 15:13 7:5	Reserved	Reserved, the reset value must be maintained.
28:24 20:16 12:8 4:0	AFSELy[4:0]	Alternate function configuration for port y (y = 0…3) 00000：AF0 00001：AF1 00010：AF2 00011：AF3 00100：AF4 00101：AF5 00110：AF6 00111：AF7 01000：AF8 01001：AF9 01010：AF10 01011：AF11 01100：AF12 01101：AF13 01110：AF14 01111：AF15 10000：AF16 10001：AF17 10010：AF18 10011：AF19 10100：AF20 10101：AF21 10110：AF22 10111：AF23 11000：AF24 11001：AF25 11010：AF26 11011：AF27 11100：AF28 11101：AF29 11110：AF30 11111：AF31

5.3.6 GPIO Alternate Function Register 1 (GPIOx_AFSEL1)

Address offset: 0x10

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved			AFSEL7[4:0]					Reserved			AFSEL6[4:0]				
			rw								rw				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			AFSEL5[4:0]					Reserved			AFSEL4[4:0]				
			rw								rw				

Bit Field	Name	Description
31:29 23:21 15:13 7:5	Reserved	Reserved, the reset value must be maintained.
28:24 20:16 12:8 4:0	AFSELy[4:0]	Alternate function configuration for port y (y = 4…7) 00000：AF0 00001：AF1 00010：AF2 00011：AF3 00100：AF4 00101：AF5 00110：AF6 00111：AF7 01000：AF8 01001：AF9 01010：AF10 01011：AF11 01100：AF12 01101：AF13 01110：AF14 01111：AF15 10000：AF16 10001：AF17 10010：AF18 10011：AF19 10100：AF20 10101：AF21 10110：AF22 10111：AF23 11000：AF24 11001：AF25 11010：AF26 11011：AF27 11100：AF28

Bit Field	Name	Description
		11101：AF29 11110：AF30 11111：AF31

5.3.7 GPIO Alternate Function Register 2 (GPIOx_AFSEL2)

Address offset: 0x14

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved			AFSEL11[4:0]					Reserved			AFSEL10[4:0]				
			rw								rw				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			AFSEL9[4:0]					Reserved			AFSEL8[4:0]				
			rw								rw				

Bit Field	Name	Description
31:29 23:21 15:13 7:5	Reserved	Reserved, the reset value must be maintained.
28:24 20:16 12:8 4:0	AFSELy[4:0]	Alternate function configuration bits y for port y (y = 8…11) 00000：AF0 00001：AF1 00010：AF2 00011：AF3 00100：AF4 00101：AF5 00110：AF6 00111：AF7 01000：AF8 01001：AF9 01010：AF10 01011：AF11 01100：AF12 01101：AF13 01110：AF14 01111：AF15 10000：AF16 10001：AF17 10010：AF18 10011：AF19

Bit Field	Name	Description
		10100：AF20 10101：AF21 10110：AF22 10111：AF23 11000：AF24 11001：AF25 11010：AF26 11011：AF27 11100：AF28 11101：AF29 11110：AF30 11111：AF31

5.3.8 GPIO Alternate Function Register 3 (GPIOx_AFSEL3)

Address offset: 0x18

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved			AFSEL15[4:0]					Reserved			AFSEL14[4:0]				
			rw								rw				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			AFSEL13[4:0]					Reserved			AFSEL12[4:0]				
			rw								rw				

Bit Field	Name	Description
31:29 23:21 15:13 7:5	Reserved	Reserved, the reset value must be maintained.
28:24 20:16 12:8 4:0	AFSELy[4:0]	Alternate function configuration for port y (y = 8…11) 00000：AF0 00001：AF1 00010：AF2 00011：AF3 00100：AF4 00101：AF5 00110：AF6 00111：AF7 01000：AF8 01001：AF9 01010：AF10

Bit Field	Name	Description
		01011：AF11 01100：AF12 01101：AF13 01110：AF14 01111：AF15 10000：AF16 10001：AF17 10010：AF18 10011：AF19 10100：AF20 10101：AF21 10110：AF22 10111：AF23 11000：AF24 11001：AF25 11010：AF26 11011：AF27 11100：AF28 11101：AF29 11110：AF30 11111：AF31

5.3.9 GPIO Driver Strength Configuration Register (GPIOx_DS)

Address offset: 0x1C

Reset value: 0x5555 5555（x = A,B,C,D,E,F,G）；0x0000 1555（x=H）

31 30	29 28	27 26	25 24	23 22	21 20	19 18	17 16
DS15[1:0]	DS14[1:0]	DS13[1:0]	DS12[1:0]	DS11[1:0]	DS10[1:0]	DS9[1:0]	DS8[1:0]
rw	rw	rw	rw	rw	rw	rw	rw
15 14	13 12	11 10	9 8	7 6	5 4	3 2	1 0
DS7[1:0]	DS6[1:0]	DS5[1:0]	DS4[1:0]	DS3[1:0]	DS2[1:0]	DS1[1:0]	DS0[1:0]
rw	rw	rw	rw	rw	rw	rw	rw

Bit Field	Name	Description
31:30 29:28 27:26 25:24 23:22 21:20 19:18	DSy[1:0]	Port y drive capability configuration (y = 0…15) 00：2mA 01：8mA 10：4mA 11：12mA

Bit Field	Name	Description
17:16 15:14 13:12 11:10 9:8 7:6 5:4 3:2 1:0		

5.3.10 GPIO Port Slew Rate Configuration Register (GPIOx_SR)

Address offset: 0x20

Reset value: 0x0000 FFFF（x= A,B,C,D,E,F,G）; 0x0000 007F（x=H）

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SR15	SR14	SR13	SR12	SR11	SR10	SR9	SR8	SR7	SR6	SR5	SR4	SR3	SR2	SR1	SR0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit Field	Name	Description
31:16	Reserved	Reserved, the reset value must be maintained.
15:0	SRy	Toggle rate configuration for port y (y = 0…15) These bits can only be read or written as 16-bit words. 0：Fast slew rate 1：Slow slew rate

5.3.11 GPIO Bit Set/Clear Register (GPIOx_PBSC)

Address offset: 0x24

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PBC15	PBC14	PBC13	PBC12	PBC11	PBC10	PBC9	PBC8	PBC7	PBC6	PBC5	PBC4	PBC3	PBC2	PBC1	PBC0
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PBS15	PBS14	PBS13	PBS12	PBS11	PBS10	PBS9	PBS8	PBS7	PBS6	PBS5	PBS4	PBS3	PBS2	PBS1	PBS0
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

Bit Field	Name	Description
31:16	PBCy	Clear bit y of port GPIOx (y = 0…15)

Bit Field	Name	Description
		These bits can only be written and operated as words (16 bits). 0: Does not affect the corresponding PODy bit 1: Clear the corresponding PODy bit to 0 *Note: if the corresponding bits of PBSy and PBCy are set at the same time, the PBSy bit works.*
15:0	PBSy	Set bit y of port GPIOx (y = 0…15) These bits can only be written and operated as words (16 bits). 0: Does not affect the corresponding PODy bit 1: Set the corresponding PODy bit to 1

5.3.12 GPIO Bit Clear Register (GPIOx_PBC)

Address offset: 0x28

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PBC15	PBC14	PBC13	PBC12	PBC11	PBC10	PBC9	PBC8	PBC7	PBC6	PBC5	PBC4	PBC3	PBC2	PBC1	PBC0
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

Bit Field	Name	Description
31:16	Reserved	Reserved, the reset value must be maintained.
15:0	PBCy	Clear bit y of port GPIOx (y = 0...15) These bits can only be written and operated as words (16 bits). 0: Does not affect the corresponding PODy bit 1: Clear the corresponding PODy bit to 0

5.3.13 GPIO Input Data Register (GPIOx_PID)

Address offset: 0x2C

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PID15	PID14	PID13	PID12	PID11	PID10	PID9	PID8	PID7	PID6	PID5	PID4	PID3	PID2	PID1	PID0
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

Bit Field	Name	Description
31:16	Reserved	Reserved, the reset value must be maintained.

Bit Field	Name	Description
15:0	PIDy	Port input data (y = 0…15) These bits are read-only and can only be read in the form of 16-bit words, and the read value is the state of the corresponding I/O port.

5.3.14 GPIO Output Data Register (GPIOx_POD)

Address offset: 0x30

Reset value: 0x0000 A000（x=A）; 0x0000 0010（x=B）; 0x0000 0000（x=C,D,E,F,G,H）

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
POD15	POD14	POD13	POD12	POD11	POD10	POD9	POD8	POD7	POD6	POD5	POD4	POD3	POD2	POD1	POD0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit Field	Name	Description
31:16	Reserved	Reserved, the reset value must be maintained.
15:0	PODy	Port output data (y = 0…15) These bits can only be read or written as 16-bit words. For GPIOx_PBSC (x = A…D), the corresponding POD bits can be independently set/cleared.

5.3.15 GPIO Configuration Lock Register (GPIOx_PLOCK)

Address offset: 0x34

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															PLOCKK
															rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PLOCK15	PLOCK14	PLOCK13	PLOCK12	PLOCK11	PLOCK10	PLOCK9	PLOCK8	PLOCK7	PLOCK6	PLOCK5	PLOCK4	PLOCK3	PLOCK2	PLOCK1	PLOCK0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit Field	Name	Description
31:17	Reserved	Reserved, the reset value must be maintained.
16	PLOCKK	Lock key. This bit can be read at any time, and it can only be modified by the key lock write sequence. 0: Port configuration lock key is activated 1: The port configuration lock key is activated, and the GPIOx_PLOCK register is locked before the next system reset. The write sequence of the lock key:

Bit Field	Name	Description
		Write 1 -> write 0 -> write 1 -> read 0 -> read 1 The last reading can be omitted, but it can be used to confirm that the lock key has been activated. *Note: the value of PLock [15:0] cannot be changed when the writing sequence of lock key is operated. Any error in the operation key writing sequence will not activate the key.*
15:0	PLOCKy	Configuration lock bit y of port GPIOx (y = 0…15) These bits are readable and writable but can only be written when the PLOCKK bit is 0. 0: Do not lock the configuration of the port 1: Lock the configuration of the port

5.4 AFIO Registers

5.4.1 AFIO Register Overview

AFIO base address: 0x40010800

Table 5-79 AFIO Register Overview

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	AFIO_RMP_CFG	XSPI_FLASH2	XSPI_NSS_SEL	XSPI_NSS_IN	ETH_MODE	XSPI_DUAL_QUAD	XSPI_HALF_DUPLEX	XSPI_XIP_WR_BE	EXTI_FILTER	FEMC_NADV	IO_FILTER[5:0]						SWJ_CFG[2:0]			SPI6_NSS	SPI5_NSS	SPI4_NSS	SPI3_NSS	SPI2_NSS	SPI1_NSS	Reserved							
	Reset value	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0								
0x04	AFIO_EXTI_CFG1	Reserved	EXTI3[6:0]							Reserved	EXTI2[6:0]							Reserved	EXTI1[6:0]							Reserved	EXTI0[6:0]						
	Reset value		0	0	0	0	0	0	0		0	0	0	0	0	0	0		0	0	0	0	0	0	0		0	0	0	0	0	0	0
0x08	AFIO_EXTI_CFG2	Reserved	EXTI7[6:0]							Reserved	EXTI6[6:0]							Reserved	EXTI5[6:0]							Reserved	EXTI4[6:0]						
	Reset value		0	0	0	0	0	0	0		0	0	0	0	0	0	0		0	0	0	0	0	0	0		0	0	0	0	0	0	0
0x0C	AFIO_EXTI_CFG3	Reserved	EXTI11[6:0]							Reserved	EXTI10[6:0]							Reserved	EXTI9[6:0]							Reserved	EXTI8[6:0]						
	Reset value		0	0	0	0	0	0	0		0	0	0	0	0	0	0		0	0	0	0	0	0	0		0	0	0	0	0	0	0
0x10	AFIO_EXTI_CFG4	Reserved	EXTI15[6:0]							Reserved	EXTI14[6:0]							Reserved	EXTI13[6:0]							Reserved	EXTI12[6:0]						
	Reset value		0	0	0	0	0	0	0		0	0	0	0	0	0	0		0	0	0	0	0	0	0		0	0	0	0	0	0	0
0x14	AFIO_ANAEN_CFG1	PB15ANAEN	PB14ANAEN	PB13ANAEN	PB12ANAEN	PB11ANAEN	PB10ANAEN	Reserved							PB2ANAEN	PB1ANAEN	PB0ANAEN	Reserved					PA10ANAEN	PA9ANAEN	PA8ANAEN	PA7ANAEN	PA6ANAEN	PA5ANAEN	PA4ANAEN	PA3ANAEN	PA2ANAEN	PA1ANAEN	PA0ANAEN
	Reset value	1	1	1	1	1	1								1	1	1						1	1	1	1	1	1	1	1	1	1	1
0x18	AFIO_ANAEN_CFG2	PD15ANAEN	PD14ANAEN	PD13ANAEN	PD12ANAEN	PD11ANAEN	PD10ANAEN	PD9ANAEN	PD8ANAEN	Reserved														PC9ANAEN	Reserved			PC5ANAEN	PC4ANAEN	PC3ANAEN	PC2ANAEN	PC1ANAEN	PC0ANAEN
	Reset value	1	1	1	1	1	1	1	1															1				1	1	1	1	1	1
0x1C	AFIO_ANAEN_CFG3	Reserved					PF10ANAEN	PF9ANAEN	PF8ANAEN	PF7ANAEN	PF6ANAEN	PF5ANAEN	PF4ANAEN	PF3ANAEN	PF2ANAEN	Reserved		PE15ANAEN	PE14ANAEN	PE13ANAEN	PE12ANAEN	PE11ANAEN	PE10ANAEN	PE9ANAEN	PE8ANAEN	PE7ANAEN	PE6ANAEN	Reserved					
	Reset value						1	1	1	1	1	1	1	1	1			1	1	1	1	1	1	1	1	1	1						
0x20	AFIO_ANAEN_CFG4	Reserved									PH6ANAEN	Reserved				PH1ANAEN	PH0ANAEN	Reserved															
	Reset value										1					1	1																
0x24	AFIO_FILTER_CFG1	PB_FILTER[15:0]																PA_FILTER[15:0]															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x28	AFIO_ FILTER _CFG2	PD_FILTER[15:0]																PC_FILTER[15:0]															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x2C	AFIO_ FILTER _CFG3	PF_FILTER[15:0]																PE_FILTER[15:0]															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x30	AFIO_ FILTER _CFG4	Reserved									PH_FILTER[6:0]							PG_FILTER[15:0]															
	Reset value										0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x34	AFIO_EMC_CFG	Reserved		XSPI_DLY[1:0]		XSPI_RD_BE	XSPI_WR_BE	XSPI_RXDS[1:0]		GB4_DET	GB3_DET	GB2_DET	GB1_DET	GBN4_DET	GBN3_DET	GBN2_DET	GBN1_DET	CLAMP4_DET	CLAMP3_DET	CLAMP2_DET	CLAMP1_DET	GB4_RST	GB3_RST	GB2_RST	GB1_RST	GBN4_RST	GBN3_RST	GBN2_RST	GBN1_RST	CLAMP4_RST	CLAMP3_RST	CLAMP2_RST	CLAMP1_RST
	Reset value			0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x38	AFIO_EMC_CNT	Reserved																	PC_RD_DELAY_EN	XSPI_NSS_HIGH [3:0]				EMC_CNT									
	Reset value																		0	0	0	0	0	1	0	1	1	0	1	0	1	0	1
0x3C	AFIO_SHRT_EXEV_CFG1	Reserved	SHRT1_EXEV4[6:0]							Reserved	SHRT1_EXEV3[6:0]							Reserved	SHRT1_EXEV2[6:0]							Reserved	SHRT1_EXEV1[6:0]						
	Reset value		1	1	1	1	1	1	1		1	1	1	1	1	1	1		1	1	1	1	1	1	1		1	1	1	1	1	1	1
0x40	AFIO_SHRT_EXEV_CFG2	Reserved	SHRT1_EXEV8[6:0]							Reserved	SHRT1_EXEV7[6:0]							Reserved	SHRT1_EXEV6[6:0]							Reserved	SHRT1_EXEV5[6:0]						
	Reset value		1	1	1	1	1	1	1		1	1	1	1	1	1	1		1	1	1	1	1	1	1		1	1	1	1	1	1	1
0x44	AFIO_SHRT_EXEV_CFG3	Reserved																	SHRT1_EXEV10[6:0]							Reserved	SHRT1_EXEV9[6:0]						
	Reset value																		1	1	1	1	1	1	1		1	1	1	1	1	1	1

5.4.2 AFIO Mapping Configuration Control Register (AFIO_RMP_CFG)

Address offset: 0x00

Reset value: 0x2000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
XSPI_FLASH2	XSPI_NSS_SEL	XSPI_NSS_IN	ETH_MODE	XSPI_DUAL_QUAD	XSPI_HALF_DUPLEX	XSPI_XIP_WR_BE	EXTI_FILTER	FEMC_NADV	IO_FILTER[5:0]						SWJ_CFG[2]
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	w

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SWJ_CFG[1:0]		SPI6_NSS	SPI5_NSS	SPI4_NSS	SPI3_NSS	SPI2_NSS	SPI1_NSS	Reserved							
w		rw	rw	rw	rw	rw	rw								

Bit Field	Name	Description
31	XSPI_FLASH2	Dual Quad-SPI Flash 2 configuration enable bit. 0: Disable Flash 2 configuration 1: Enable Flash 2 configuration
30	XSPI_NSS_SEL	When the host mode CS input is enabled, the effective CS input pin selection bit. 0: NSS0 is effective 1: NSS1 is effective
29	XSPI_NSS_IN	In host mode, CS input enable bit, used for multi-master conflict test. 0: CS input disabled 1: CS input enabled
28	ETH_MODE	Ethernet PHY interface mode configuration bit. 0: MII 1: RMII
27	XSPI_DUAL_QUAD	XSPI dual Quad-SPI mode enable bit 0: Disable dual Quad-SPI mode 1: Enable dual Quad-SPI mode

Bit Field	Name	Description
26	XSPI_HALF_DUPLEX	XSPI half-duplex mode enable bit 0: Disable half-duplex mode 1: Enable half-duplex mode
25	XSPI_XIP_WR_BE	XSPI XIP mode write operation endianness configuration bit. 0: Little-endian mode 1: Big-endian mode
24	EXTI_FILTER	EXTI analog filtering bypass control bit 0: Enable EXTI analog filtering 1: Disable EXTI analog filtering
23	FEMC_NADV	FEMC NADV signal control bit 0: NADV signal connected to external pin 1: NADV disconnected from external pin
22:17	IO_FILTER[5:0]	IO digital filtering control bit 000000: Disable IO digital filtering, minimum effective pulse width is 1HCLK 000001: IO filtering enabled, minimum effective pulse width is 2HCLK 000010: IO filtering enabled, minimum effective pulse width is 3HCLK ... 111111: IO filtering enabled, minimum effective pulse width is 64HCLK

16:14	SWJ_CFG[2:0]	Serial wire JTAG configuration
		These bits are write-only by software (reading these bits will return undefined values) and are used to configure the I/O port for SWD_JTAG multiplexing. SWD_JTAG (Serial Wire JTAG) supports JTAG or SWD access to the Cortex debug port. The default state after system reset is to enable SWD_JTAG, in this state, JTAG or SW (Serial Wire) mode can be selected through specific signals on the JTMS/JTCK pins.
		000: Full SWD_JTAG (JTAG-DP + SW-DP): Reset state;
		001: Full SWD_JTAG (JTAG-DP + SW-DP) but without NJTRST;
		010: Disable JTAG-DP, enable SW-DP;
		100: Disable JTAG-DP, disable SW-DP;
		Other values: No effect.

13	SPI6_NSS	SPI6 NSS idle state configuration bit 0: NSS is in high-impedance state when idle 1: NSS is at high level when idle
12	SPI5_NSS	SPI5 NSS idle state configuration bit 0: NSS is in high-impedance state when idle 1: NSS is at high level when idle
11	SPI4_NSS	SPI4 NSS idle state configuration bit 0: NSS is in high-impedance state when idle 1: NSS is at high level when idle
10	SPI3_NSS	SPI3 NSS idle state configuration bit 0: NSS is in high-impedance state when idle 1: NSS is at high level when idle

Bit Field	Name	Description
9	SPI2_NSS	SPI2 NSS idle state configuration bit 0: NSS is in high-impedance state when idle 1: NSS is at high level when idle
8	SPI1_NSS	SPI1 NSS idle state configuration bit 0: NSS is in high-impedance state when idle 1: NSS is at high level when idle
7:0	Reserved	Reserved, the reset value must be maintained.

5.4.3 AFIO External Interrupt Configuration Register 1(AFIO_EXTI_CFG1)

Address offset: 0x04

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved	EXTI3[6:0]							Reserved	EXTI2[6:0]						
	rw								rw						

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	EXTI1[6:0]							Reserved	EXTI0[6:0]						
	rw								rw						

Bit Field	Name	Description
31 23 25 7	Reserved	Reserved, the reset value must be maintained.
30:24 22:16 14:8 6:0	EXTIx[6:0]	EXTIx configuration（x = 0 … 3） These bits can be read and written by software to select the input source for the EXTIx external interrupt. 0000000:PA[0] 0000001:PB[0] 0000010:PC[0] 0000011:PD[0] 0000100:PE[0] 0000101:PF[0] 0000110:PG[0] 0000111:PH[0] 0001000:PA[1] 0001001:PB[1] 0001010:PC[1] 0001011:PD[1] 0001100:PE[1] 0001101:PF[1] 0001110:PG[1] 0001111:PH[1] 0010000:PA[2] 0010001:PB[2] 0010010:PC[2] 0010011:PD[2] 0010100:PE[2] 0010101:PF[2] 0010110:PG[2] 0010111:PH[2] 0011000:PA[3] 0011001:PB[3] 0011010:PC[3] 0011011:PD[3] 0011100:PE[3] 0011101:PF[3] 0011110:PG[3] 0011111:PH[3] 0100000:PA[4] 0100001:PB[4] 0100010:PC[4] 0100011:PD[4] 0100100:PE[4] 0100101:PF[4] 0100110:PG[4] 0100111:PH[4] 0101000:PA[5] 0101001:PB[5] 0101010:PC[5] 0101011:PD[5] 0101100:PE[5] 0101101:PF[5] 0101110:PG[5] 0101111:PH[5] 0110000:PA[6] 0110001:PB[6] 0110010:PC[6] 0110011:PD[6] 0110100:PE[6] 0110101:PF[6] 0110110:PG[6] 0110111:PH[6]

Bit Field	Name	Description
		0111000:PA[7] 0111001:PB[7] 0111010:PC[7] 0111011:PD[7] 0111100:PE[7] 0111101:PF[7] 0111110:PG[7] 1000000:PA[8] 1000001:PB[8] 1000010:PC[8] 1000011:PD[8] 1000100:PE[8] 1000101:PF[8] 1000110:PG[8] 1001000:PA[9] 1001001:PB[9] 1001010:PC[9] 1001011:PD[9] 1001100:PE[9] 1001101:PF[9] 1001110:PG[9] 1010000:PA[10] 1010001:PB[10] 1010010:PC[10] 1010011:PD[10] 1010100:PE[10] 1010101:PF[10] 1010110:PG[10] 1011000:PA[11] 1011001:PB[11] 1011010:PC[11] 1011011:PD[11] 1011100:PE[11] 1011101:PF[11] 1011110:PG[11] 1100000:PA[12] 1100001:PB[12] 1100010:PC[12] 1100011:PD[12] 1100100:PE[12] 1100101:PF[12] 1100110:PG[12] 1101000:PA[13] 1101001:PB[13] 1101010:PC[13] 1101011:PD[13] 1101100:PE[13] 1101101:PF[13] 1101110:PG[13] 1110000:PA[14] 1110001:PB[14] 1110010:PC[14] 1110011:PD[14] 1110100:PE[14] 1110101:PF[14] 1110110:PG[14] 1111000:PA[15] 1111001:PB[15] 1111010:PC[15] 1111011:PD[15] 1111100:PE[15] 1111101:PF[15] 1111110:PG[15] Others: reserved

5.4.4 AFIO External Interrupt Configuration Register 2(AFIO_EXTI_CFG2)

Address offset: 0x08

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved	EXTI7[6:0]							Reserved	EXTI6[6:0]						
	rw								rw						

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	EXTI5[6:0]							Reserved	EXTI4[6:0]						
	rw								rw						

Bit Field	Name	Description
31 23 25 7	Reserved	Reserved, the reset value must be maintained.
30:24 22:16 14:8 6:0	EXTIx[6:0]	EXTIx configuration（x = 4 … 7） These bits can be read and written by software to select the input source for the EXTIx external interrupt. 0000000:PA[0] 0000001:PB[0] 0000010:PC[0] 0000011:PD[0] 0000100:PE[0] 0000101:PF[0] 0000110:PG[0] 0000111:PH[0]

		0001000:PA[1] 0001001:PB[1] 0001010:PC[1] 0001011:PD[1] 0001100:PE[1] 0001101:PF[1] 0001110:PG[1] 0001111:PH[1] 0010000:PA[2] 0010001:PB[2] 0010010:PC[2] 0010011:PD[2] 0010100:PE[2] 0010101:PF[2] 0010110:PG[2] 0010111:PH[2] 0011000:PA[3] 0011001:PB[3] 0011010:PC[3] 0011011:PD[3] 0011100:PE[3] 0011101:PF[3] 0011110:PG[3] 0011111:PH[3] 0100000:PA[4] 0100001:PB[4] 0100010:PC[4] 0100011:PD[4] 0100100:PE[4] 0100101:PF[4] 0100110:PG[4] 0100111:PH[4] 0101000:PA[5] 0101001:PB[5] 0101010:PC[5] 0101011:PD[5] 0101100:PE[5] 0101101:PF[5] 0101110:PG[5] 0101111:PH[5] 0110000:PA[6] 0110001:PB[6] 0110010:PC[6] 0110011:PD[6] 0110100:PE[6] 0110101:PF[6] 0110110:PG[6] 0110111:PH[6] 0111000:PA[7] 0111001:PB[7] 0111010:PC[7] 0111011:PD[7] 0111100:PE[7] 0111101:PF[7] 0111110:PG[7] 1000000:PA[8] 1000001:PB[8] 1000010:PC[8] 1000011:PD[8] 1000100:PE[8] 1000101:PF[8] 1000110:PG[8] 1001000:PA[9] 1001001:PB[9] 1001010:PC[9] 1001011:PD[9] 1001100:PE[9] 1001101:PF[9] 1001110:PG[9] 1010000:PA[10] 1010001:PB[10] 1010010:PC[10] 1010011:PD[10] 1010100:PE[10] 1010101:PF[10] 1010110:PG[10] 1011000:PA[11] 1011001:PB[11] 1011010:PC[11] 1011011:PD[11] 1011100:PE[11] 1011101:PF[11] 1011110:PG[11] 1100000:PA[12] 1100001:PB[12] 1100010:PC[12] 1100011:PD[12] 1100100:PE[12] 1100101:PF[12] 1100110:PG[12] 1101000:PA[13] 1101001:PB[13] 1101010:PC[13] 1101011:PD[13] 1101100:PE[13] 1101101:PF[13] 1101110:PG[13] 1110000:PA[14] 1110001:PB[14] 1110010:PC[14] 1110011:PD[14] 1110100:PE[14] 1110101:PF[14] 1110110:PG[14] 1111000:PA[15] 1111001:PB[15] 1111010:PC[15] 1111011:PD[15] 1111100:PE[15] 1111101:PF[15] 1111110:PG[15] Others: reserved

5.4.5 AFIO External Interrupt Configuration Register 3(AFIO_EXTI_CFG3)

Address offset: 0x0C

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved	EXTI11[6:0]							Reserved	EXTI10[6:0]						
	rw								rw						

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	EXTI9[6:0]							Reserved	EXTI8[6:0]						
	rw								rw						

Bit Field	Name	Description
31 23 25 7	Reserved	Reserved, the reset value must be maintained.
30:24 22:16 14:8 6:0	EXTIx[6:0]	EXTIx configuration（x = 8 … 11） These bits can be read and written by software to select the input source for the EXTIx external interrupt. 0000000:PA[0] 0000001:PB[0] 0000010:PC[0] 0000011:PD[0] 0000100:PE[0] 0000101:PF[0] 0000110:PG[0] 0000111:PH[0] 0001000:PA[1] 0001001:PB[1] 0001010:PC[1] 0001011:PD[1] 0001100:PE[1] 0001101:PF[1] 0001110:PG[1] 0001111:PH[1] 0010000:PA[2] 0010001:PB[2] 0010010:PC[2] 0010011:PD[2] 0010100:PE[2] 0010101:PF[2] 0010110:PG[2] 0010111:PH[2] 0011000:PA[3] 0011001:PB[3] 0011010:PC[3] 0011011:PD[3] 0011100:PE[3] 0011101:PF[3] 0011110:PG[3] 0011111:PH[3] 0100000:PA[4] 0100001:PB[4] 0100010:PC[4] 0100011:PD[4] 0100100:PE[4] 0100101:PF[4] 0100110:PG[4] 0100111:PH[4] 0101000:PA[5] 0101001:PB[5] 0101010:PC[5] 0101011:PD[5] 0101100:PE[5] 0101101:PF[5] 0101110:PG[5] 0101111:PH[5] 0110000:PA[6] 0110001:PB[6] 0110010:PC[6] 0110011:PD[6] 0110100:PE[6] 0110101:PF[6] 0110110:PG[6] 0110111:PH[6] 0111000:PA[7] 0111001:PB[7] 0111010:PC[7] 0111011:PD[7] 0111100:PE[7] 0111101:PF[7] 0111110:PG[7] 1000000:PA[8] 1000001:PB[8] 1000010:PC[8] 1000011:PD[8] 1000100:PE[8] 1000101:PF[8] 1000110:PG[8] 1001000:PA[9] 1001001:PB[9] 1001010:PC[9] 1001011:PD[9] 1001100:PE[9] 1001101:PF[9] 1001110:PG[9] 1010000:PA[10] 1010001:PB[10] 1010010:PC[10] 1010011:PD[10] 1010100:PE[10] 1010101:PF[10] 1010110:PG[10] 1011000:PA[11] 1011001:PB[11] 1011010:PC[11] 1011011:PD[11] 1011100:PE[11] 1011101:PF[11] 1011110:PG[11] 1100000:PA[12] 1100001:PB[12] 1100010:PC[12] 1100011:PD[12] 1100100:PE[12] 1100101:PF[12] 1100110:PG[12] 1101000:PA[13] 1101001:PB[13] 1101010:PC[13] 1101011:PD[13] 1101100:PE[13] 1101101:PF[13] 1101110:PG[13] 1110000:PA[14] 1110001:PB[14] 1110010:PC[14] 1110011:PD[14] 1110100:PE[14] 1110101:PF[14] 1110110:PG[14] 1111000:PA[15] 1111001:PB[15] 1111010:PC[15] 1111011:PD[15] 1111100:PE[15] 1111101:PF[15] 1111110:PG[15] Others: reserved

5.4.6 AFIO External Interrupt Configuration Register 4(AFIO_EXTI_CFG4)

Address offset: 0x10

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved	EXTI15[6:0]							Reserved	EXTI14[6:0]						
	rw								rw						

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	EXTI13[6:0]							Reserved	EXTI12[6:0]						
	rw								rw						

Bit Field	Name	Description
31 23 25 7	Reserved	Reserved, the reset value must be maintained.
30:24 22:16 14:8 6:0	EXTIx[6:0]	EXTIx configuration（x = 12 … 15） These bits can be read and written by software to select the input source for the EXTIx external interrupt. 0000000:PA[0] 0000001:PB[0] 0000010:PC[0] 0000011:PD[0] 0000100:PE[0] 0000101:PF[0] 0000110:PG[0] 0000111:PH[0] 0001000:PA[1] 0001001:PB[1] 0001010:PC[1] 0001011:PD[1] 0001100:PE[1] 0001101:PF[1] 0001110:PG[1] 0001111:PH[1] 0010000:PA[2] 0010001:PB[2] 0010010:PC[2] 0010011:PD[2] 0010100:PE[2] 0010101:PF[2] 0010110:PG[2] 0010111:PH[2] 0011000:PA[3] 0011001:PB[3] 0011010:PC[3] 0011011:PD[3] 0011100:PE[3] 0011101:PF[3] 0011110:PG[3] 0011111:PH[3] 0100000:PA[4] 0100001:PB[4] 0100010:PC[4] 0100011:PD[4] 0100100:PE[4] 0100101:PF[4] 0100110:PG[4] 0100111:PH[4] 0101000:PA[5] 0101001:PB[5] 0101010:PC[5] 0101011:PD[5] 0101100:PE[5] 0101101:PF[5] 0101110:PG[5] 0101111:PH[5] 0110000:PA[6] 0110001:PB[6] 0110010:PC[6] 0110011:PD[6] 0110100:PE[6] 0110101:PF[6] 0110110:PG[6] 0110111:PH[6] 0111000:PA[7] 0111001:PB[7] 0111010:PC[7] 0111011:PD[7] 0111100:PE[7] 0111101:PF[7] 0111110:PG[7] 1000000:PA[8] 1000001:PB[8] 1000010:PC[8] 1000011:PD[8] 1000100:PE[8] 1000101:PF[8] 1000110:PG[8] 1001000:PA[9] 1001001:PB[9] 1001010:PC[9] 1001011:PD[9] 1001100:PE[9] 1001101:PF[9] 1001110:PG[9] 1010000:PA[10] 1010001:PB[10] 1010010:PC[10] 1010011:PD[10] 1010100:PE[10] 1010101:PF[10] 1010110:PG[10] 1011000:PA[11] 1011001:PB[11] 1011010:PC[11] 1011011:PD[11]

		1011100:PE[11] 1011101:PF[11] 1011110:PG[11] 1100000:PA[12] 1100001:PB[12] 1100010:PC[12] 1100011:PD[12] 1100100:PE[12] 1100101:PF[12] 1100110:PG[12] 1101000:PA[13] 1101001:PB[13] 1101010:PC[13] 1101011:PD[13] 1101100:PE[13] 1101101:PF[13] 1101110:PG[13] 1110000:PA[14] 1110001:PB[14] 1110010:PC[14] 1110011:PD[14] 1110100:PE[14] 1110101:PF[14] 1110110:PG[14] 1111000:PA[15] 1111001:PB[15] 1111010:PC[15] 1111011:PD[15] 1111100:PE[15] 1111101:PF[15] 1111110:PG[15] Others: reserved

5.4.7 AFIO IO Port Analog Signal Channel Configuration Register 1 （AFIO_ANAEN_CFG1）

Address offset: 0x14

Reset value: 0xFFFF FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PB15A NAEN	PB14A NAEN	PB13A NAEN	PB12A NAEN	PB11A NAEN	PB10A NAEN	Reserved							PB2A NAEN	PB1A NAEN	PB0A NAEN
rw	rw	rw	rw	rw	rw								rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved					PA10A NAEN	PA9A NAEN	PA8A NAEN	PA7A NAEN	PA6A NAEN	PA5A NAEN	PA4A NAEN	PA3A NAEN	PA2A NAEN	PA1A NAEN	PA0A NAEN
					rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit Field	Name	Description
31:26	PBxANAEN	PBx port analog signal channel enable bit (x = 10 … 15) 0: Disable analog signal channel for port PBx 1: Enable analog signal channel for port PBx
25:19	Reserved	Reserved, the reset value must be maintained.
18:16	PBxANAEN	PBx port analog signal channel enable bit (x = 0 … 2) 0: Disable analog signal channel for port PBx 1: Enable analog signal channel for port PBx
15:11	Reserved	Reserved, the reset value must be maintained.
10:0	PAxANAEN	PAx port analog signal channel enable bit (x = 0 … 10) 0: Disable analog signal channel for port PAx 1: Enable analog signal channel for port PAx

5.4.8 AFIO IO Port Analog Signal Channel Configuration Register 2 （AFIO_ANAEN_CFG2）

Address offset: 0x18

Reset value: 0xFFFF FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PD15A NAEN	PD14A NAEN	PD13A NAEN	PD12A NAEN	PD11A NAEN	PD10A NAEN	PD9A NAEN	PD8A NAEN	Reserved							
rw	rw	rw	rw	rw	rw	rw	rw								

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						PC9A NAEN	Reserved			PC5A NAEN	PC4A NAEN	PC3A NAEN	PC2A NAEN	PC1A NAEN	PC0A NAEN
						rw				rw	rw	rw	rw	rw	rw

Bit Field	Name	Description
31:24	PDxANAEN	PDx port analog signal channel enable bit (x = 8 … 15) 0: Disable analog signal channel for port PDx 1: Enable analog signal channel for port PDx
23:10	Reserved	Reserved, the reset value must be maintained.
9	PC9ANAEN	PC9 port analog signal channel enable bit 0: Disable analog signal channel for port PC9 1: Enable analog signal channel for port PC9
8:6	Reserved	Reserved, the reset value must be maintained.
5:0	PCxANAEN	PCx port analog signal channel enable bit (x = 0 … 5) 0: Disable analog signal channel for port PCx 1: Enable analog signal channel for port PCx

5.4.9 AFIO IO Port Analog Signal Channel Configuration Register 3 （AFIO_ANAEN_CFG3）

Address offset: 0x1C

Reset value: 0xFFFF FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved					PF10A NAEN	PF9A NAEN	PF8A NAEN	PF7A NAEN	PF6A NAEN	PF5A NAEN	PF4A NAEN	PF3A NAEN	PF2A NAEN	Reserved	
					rw	rw	rw	rw	rw	rw	rw	rw	rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PE15A NAEN	PE14A NAEN	PE13A NAEN	PE12A NAEN	PE11A NAEN	PE10A NAEN	PE9A NAEN	PE8A NAEN	PE7A NAEN	PE6A NAEN	Reserved					
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw						

Bit Field	Name	Description
31:27	Reserved	Reserved, the reset value must be maintained.
26:18	PFxANAEN	PFx port analog signal channel enable bit (x = 2 … 10) 0: Disable analog signal channel for port PFx 1: Enable analog signal channel for port PFx
17:16	Reserved	Reserved, the reset value must be maintained.
15:6	PExANAEN	PEx port analog signal channel enable bit (x = 6 … 15)

Bit Field	Name	Description
		0: Disable analog signal channel for port PEx 1: Enable analog signal channel for port PEx
5:0	Reserved	Reserved, the reset value must be maintained.

5.4.10 AFIO IO Port Analog Signal Channel Configuration Register 4 (AFIO_ANAEN_CFG4)

Address offset: 0x20

Reset value: 0xFFFF FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved									PH6ANAEN	Reserved				PH1ANAEN	PH0ANAEN
									rw					rw	rw
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved															

Bit Field	Name	Description
31:23	Reserved	Reserved, the reset value must be maintained.
22	PHxANAEN	PH6 port analog signal channel enable bit 0: Disable analog signal channel for port PH6 1: Enable analog signal channel for port PH6
21:18	Reserved	Reserved, the reset value must be maintained.
17:16	PHxANAEN	PHx port analog signal channel enable bit (x = 0,1) 0: Disable analog signal channel for port PHx 1: Enable analog signal channel for port PHx
15:0	Reserved	Reserved, the reset value must be maintained.

5.4.11 AFIO IO Port Digital Filtering Configuration Register 1 （AFIO_FILTER_CFG1）

Address offset: 0x24

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PB_FILTER[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PA_FILTER[15:0]															

rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit Field	Name	Description
31:16	PB_FILTER[15:0]	PBx port digital filtering enable bit (x = 0 … 15) 0: Disable digital filtering for port PBx 1: Enable digital filtering for port PBx
15:0	PA_FILTER[15:0]	PAx port digital filtering enable bit (x = 0 … 15) 0: Disable digital filtering for port PAx 1: Enable digital filtering for port PAx

5.4.12 AFIO IO Port Digital Filtering Configuration Register 2 （AFIO_FILTER_CFG2）

Address offset: 0x28

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PD_FILTER[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PC_FILTER[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit Field	Name	Description
31:16	PD_FILTER[15:0]	PDx port digital filtering enable bit (x = 0 … 15) 0: Disable digital filtering for port PDx 1: Enable digital filtering for port PDx
15:0	PC_FILTER[15:0]	PCx port digital filtering enable bit (x = 0 … 15) 0: Disable digital filtering for port PCx 1: Enable digital filtering for port PCx

5.4.13 AFIO IO Port Digital Filtering Configuration Register 3 （AFIO_FILTER_CFG3）

Address offset: 0x2C

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PF_FILTER[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PE_FILTER[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit Field	Name	Description
31:16	PF_FILTER[15:0]	PFx port digital filtering enable bit (x = 0 … 15) 0: Disable digital filtering for port PFx 1: Enable digital filtering for port PFx
15:0	PE_FILTER[15:0]	PEx port digital filtering enable bit (x = 0 … 15) 0: Disable digital filtering for port PEx 1: Enable digital filtering for port PEx

5.4.14 AFIO IO Port Digital Filtering Configuration Register 4 （AFIO_FILTER_CFG4）

Address offset: 0x30

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved									PH_FILTER[6:0]						
									rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PG_FILTER[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit Field	Name	Description
31:23	Reserved	Reserved, the reset value must be maintained.
22:16	PH_FILTER[6:0]	PHx port digital filtering enable bit (x = 0 … 6) 0: Disable digital filtering for port PHx 1: Enable digital filtering for port PHx
15:0	PG_FILTER[15:0]	PGx port digital filtering enable bit (x = 0 … 15) 0: Disable digital filtering for port PGx 1: Enable digital filtering for port PGx

5.4.15 AFIO EMC Configuration Register（AFIO_EMC_CFG）

Address offset: 0x34

Reset value: 0x0C00 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

Reserved		XSPI_EXTEND[1:0]		XSPI_RD_BE	XSPI_WR_BE	XSPI_RXDS [1:0]		GB4_DET	GB3_DET	GB2_DET	GB1_DET	GBN4_DET	GBN3_DET	GBN2_DET	GBN1_DET
								rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CLAMP4_DET	CLAMP3_DET	CLAMP2_DET	CLAMP1_DET	GB4_RST	GB3_RST	GB2_RST	GB1_RST	GBN4_RST	GBN3_RST	GBN2_RST	GBN1_RST	CLAMP4_RST	CLAMP3_RST	CLAMP2_RST	CLAMP1_RST
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit Field	Name	Description
31:30	Reserved	Reserved, the reset value must be maintained.
29:28	XSPI_EXTEND[1:0]	XSPI CS setup and hold time extension configuration bit. 00: Disable CS setup and hold time extension function 01: Only enable CS setup time extension function 10: Only enable CS hold time extension function 11: Enable both CS setup and hold time extension functions
27	XSPI_ RD_BE	XSPI normal mode (non-XIP) read operation endianness configuration bit. 0: Little-endian mode 1: Big-endian mode
26	XSPI_ WR_BE	XSPI normal mode (non-XIP) write operation endianness configuration bit. 0: Little-endian mode 1: Big-endian mode
25：24	XSPI_RXDS[1:0]	XSPI RXDS sampling delay time configuration bit. 01: 1HCLK 10: 2HCLK other: Bypass
23	GB4_DET	EMC GB4 detection enable bit (ground bounce detection). 0: Disable 1: Enable
22	GB3_DET	EMC GB3 detection enable bit 0: Disable 1: Enable
21	GB2_DET	EMC GB2 detection enable bit 0: Disable 1: Enable
20	GB1_DET	EMC GB1 detection enable bit 0: Disable 1: Enable
19	GBN4_DET	EMC GBN4 detection enable bit 0: Disable 1: Enable
18	GBN3_DET	EMC GBN3 detection enable bit 0: Disable 1: Enable

Bit Field	Name	Description
17	GBN2_DET	EMC GBN2 detection enable bit 0: Disable 1: Enable
16	GBN1_DET	EMC GBN1 detection enable bit 0: Disable 1: Enable
15	CLAMP4_DET	EMC_CLAMP4 detection enable bit for VDD_4 0: Disable 1: Enable
14	CLAMP3_DET	EMC_CLAMP3 detection enable bit for VDD_3 0: Disable 1: Enable
13	CLAMP2_DET	EMC_CLAMP2 detection enable bit for VDD_2 0: Disable 1: Enable
12	CLAMP1_DET	EMC_CLAMP1 detection enable bit for VDD_1 0: Disable 1: Enable
11	GB4_RST	System reset enable bit when EMC GB4 is detected. 0: Disable 1: Enable
10	GB3_RST	System reset enable bit when EMC GB3 is detected. 0: Disable 1: Enable
9	GB2_RST	System reset enable bit when EMC GB2 is detected. 0: Disable 1: Enable
8	GB1_RST	System reset enable bit when EMC GB1 is detected. 0: Disable 1: Enable
7	GBN4_RST	System reset enable bit when EMC GBN4 is detected. 0: Disable 1: Enable
6	GBN3_RST	System reset enable bit when EMC GBN3 is detected. 0: Disable 1: Enable
5	GBN2_RST	System reset enable bit when EMC GBN2 is detected. 0: Disable 1: Enable
4	GBN1_RST	System reset enable bit when EMC GBN1 is detected. 0: Disable 1: Enable

Bit Field	Name	Description
3	CLAMP4_RST	System reset enable bit when EMC CLAMP4 is detected. 0: Disable 1: Enable
2	CLAMP3_RST	System reset enable bit when EMC CLAMP3 is detected. 0: Disable 1: Enable
1	CLAMP2_RST	System reset enable bit when EMC CLAMP2 is detected. 0: Disable 1: Enable
0	CLAMP1_RST	System reset enable bit when EMC CLAMP1 is detected. 0: Disable 1: Enable

5.4.16 AFIO EMC Counter Register（AFIO_EMC_CNT）

Address offset: 0x38

Reset value: 0x000 02D5

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	PC_RD_DELAY_EN	XSPI_NSS_HIGH[3:0]				EMC_CNT[9:0]									
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit Field	Name	Description
31:15	Reserved	Reserved, the reset value must be maintained.
14	PC_RD_DELAY_EN	GPIOC register back-to-back read delay enable bit. 0: Disable back-to-back read delay for GPIOC register 1: Enable back-to-back read delay for GPIOC register
13:10	XSPI_NSS_HIGH[3:0]	NSS idle time (high level) configuration bit. In XSPI host single-line mode communication, if the effective level of NSS is low, it supports setting NSS to a high level between every two data frames (4~32 bits) as idle time by enabling bit[14] SSTE in the XSPI_CTRL0 register. When SSTE=0, NSS outputs a valid low level at the beginning of communication, and then maintains it until all data frames are sent, and then outputs an idle high level. When SSTE=1, NSS outputs a valid low level before sending each data frame, and outputs an idle high level after each data frame is sent. This configuration bit is used to configure the duration of the idle high level for NSS when SSTE=1. 0000: Idle high level between data frames lasts for 1xSPI_CLK 0001: Idle high level between data frames lasts for 2xSPI_CLK

Bit Field	Name	Description
		… 1111: Idle high level between data frames lasts for 16*xSPI_CLK
9:0	EMC_CNT[9:0]	EMC error counter

5.4.17 AFIO SHRTIM1 External Event Configuration Register 1 (AFIO_SHRT_EXEV_CFG1)

Address offset: 0x3C

Reset value: 0x7F7F 7F7F

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved	SHRT1_EXEV4[6:0]							Reserved	SHRT1_EXEV3[6:0]						
	rw								rw						

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	SHRT1_EXEV2[6:0]							Reserved	SHRT1_EXEV1[6:0]						
	rw								rw						

Bit Field	Name	Description
31 23 25 7	Reserved	Reserved, the reset value must be maintained.
30:24 22:16 14:8 6:0	SHRT1_EXEVx[6:0]	SHRTIM_EXEVx external event input channel configuration bit (x = 1 … 4) These bits can be read and written by software to select the external input source for SHRTIM_EXEVx. 0000000:PA[0] 0000001:PB[0] 0000010:PC[0] 0000011:PD[0] 0000100:PE[0] 0000101:PF[0] 0000110:PG[0] 0000111:PH[0] 0001000:PA[1] 0001001:PB[1] 0001010:PC[1] 0001011:PD[1] 0001100:PE[1] 0001101:PF[1] 0001110:PG[1] 0001111:PH[1] 0010000:PA[2] 0010001:PB[2] 0010010:PC[2] 0010011:PD[2] 0010100:PE[2] 0010101:PF[2] 0010110:PG[2] 0010111:PH[2] 0011000:PA[3] 0011001:PB[3] 0011010:PC[3] 0011011:PD[3] 0011100:PE[3] 0011101:PF[3] 0011110:PG[3] 0011111:PH[3] 0100000:PA[4] 0100001:PB[4] 0100010:PC[4] 0100011:PD[4] 0100100:PE[4] 0100101:PF[4] 0100110:PG[4] 0100111:PH[4] 0101000:PA[5] 0101001:PB[5] 0101010:PC[5] 0101011:PD[5] 0101100:PE[5] 0101101:PF[5] 0101110:PG[5] 0101111:PH[5] 0110000:PA[6] 0110001:PB[6] 0110010:PC[6] 0110011:PD[6] 0110100:PE[6] 0110101:PF[6] 0110110:PG[6] 0110111:PH[6] 0111000:PA[7] 0111001:PB[7] 0111010:PC[7] 0111011:PD[7] 0111100:PE[7] 0111101:PF[7] 0111110:PG[7]

Bit Field	Name	Description
		1000000:PA[8] 1000001:PB[8] 1000010:PC[8] 1000011:PD[8] 1000100:PE[8] 1000101:PF[8] 1000110:PG[8] 1001000:PA[9] 1001001:PB[9] 1001010:PC[9] 1001011:PD[9] 1001100:PE[9] 1001101:PF[9] 1001110:PG[9] 1010000:PA[10] 1010001:PB[10] 1010010:PC[10] 1010011:PD[10] 1010100:PE[10] 1010101:PF[10] 1010110:PG[10] 1011000:PA[11] 1011001:PB[11] 1011010:PC[11] 1011011:PD[11] 1011100:PE[11] 1011101:PF[11] 1011110:PG[11] 1100000:PA[12] 1100001:PB[12] 1100010:PC[12] 1100011:PD[12] 1100100:PE[12] 1100101:PF[12] 1100110:PG[12] 1101000:PA[13] 1101001:PB[13] 1101010:PC[13] 1101011:PD[13] 1101100:PE[13] 1101101:PF[13] 1101110:PG[13] 1110000:PA[14] 1110001:PB[14] 1110010:PC[14] 1110011:PD[14] 1110100:PE[14] 1110101:PF[14] 1110110:PG[14] 1111000:PA[15] 1111001:PB[15] 1111010:PC[15] 1111011:PD[15] 1111100:PE[15] 1111101:PF[15] 1111110:PG[15] Others: reserved

5.4.18 AFIO SHRTIM1 External Event Configuration Register 2 (AFIO_SHRT_EXEV_CFG2)

Address offset: 0x40

Reset value: 0x0000 7F7F

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved	SHRT1_EXEV8[6:0]							Reserved	SHRT1_EXEV7[6:0]						
	rw								rw						
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved	SHRT1_EXEV6[6:0]							Reserved	SHRT1_EXEV5[6:0]						
	rw								rw						

Bit Field	Name	Description
31 23 25 7	Reserved	Reserved, the reset value must be maintained.
30:24 22:16 14:8 6:0	SHRT1_EXEVx[6:0]	SHRTIM_EXEVx external event input channel configuration bit (x = 5 … 8) These bits can be read and written by software to select the external input source for SHRTIM_EXEVx. 0000000:PA[0] 0000001:PB[0] 0000010:PC[0] 0000011:PD[0] 0000100:PE[0] 0000101:PF[0] 0000110:PG[0] 0000111:PH[0]

Bit Field	Name	Description
		0001000:PA[1] 0001001:PB[1] 0001010:PC[1] 0001011:PD[1] 0001100:PE[1] 0001101:PF[1] 0001110:PG[1] 0001111:PH[1] 0010000:PA[2] 0010001:PB[2] 0010010:PC[2] 0010011:PD[2] 0010100:PE[2] 0010101:PF[2] 0010110:PG[2] 0010111:PH[2] 0011000:PA[3] 0011001:PB[3] 0011010:PC[3] 0011011:PD[3] 0011100:PE[3] 0011101:PF[3] 0011110:PG[3] 0011111:PH[3] 0100000:PA[4] 0100001:PB[4] 0100010:PC[4] 0100011:PD[4] 0100100:PE[4] 0100101:PF[4] 0100110:PG[4] 0100111:PH[4] 0101000:PA[5] 0101001:PB[5] 0101010:PC[5] 0101011:PD[5] 0101100:PE[5] 0101101:PF[5] 0101110:PG[5] 0101111:PH[5] 0110000:PA[6] 0110001:PB[6] 0110010:PC[6] 0110011:PD[6] 0110100:PE[6] 0110101:PF[6] 0110110:PG[6] 0110111:PH[6] 0111000:PA[7] 0111001:PB[7] 0111010:PC[7] 0111011:PD[7] 0111100:PE[7] 0111101:PF[7] 0111110:PG[7] 1000000:PA[8] 1000001:PB[8] 1000010:PC[8] 1000011:PD[8] 1000100:PE[8] 1000101:PF[8] 1000110:PG[8] 1001000:PA[9] 1001001:PB[9] 1001010:PC[9] 1001011:PD[9] 1001100:PE[9] 1001101:PF[9] 1001110:PG[9] 1010000:PA[10] 1010001:PB[10] 1010010:PC[10] 1010011:PD[10] 1010100:PE[10] 1010101:PF[10] 1010110:PG[10] 1011000:PA[11] 1011001:PB[11] 1011010:PC[11] 1011011:PD[11] 1011100:PE[11] 1011101:PF[11] 1011110:PG[11] 1100000:PA[12] 1100001:PB[12] 1100010:PC[12] 1100011:PD[12] 1100100:PE[12] 1100101:PF[12] 1100110:PG[12] 1101000:PA[13] 1101001:PB[13] 1101010:PC[13] 1101011:PD[13] 1101100:PE[13] 1101101:PF[13] 1101110:PG[13] 1110000:PA[14] 1110001:PB[14] 1110010:PC[14] 1110011:PD[14] 1110100:PE[14] 1110101:PF[14] 1110110:PG[14] 1111000:PA[15] 1111001:PB[15] 1111010:PC[15] 1111011:PD[15] 1111100:PE[15] 1111101:PF[15] 1111110:PG[15] Others: reserved

5.4.19 AFIO SHRTIM1 External Event Configuration Register 3 (AFIO_SHRT_EXEV_CFG3)

Address offset: 0x44

Reset value: 0x7F7F 7F7F

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
				rw								rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Reserved	SHRT1_EXEV10[6:0]	Reserved	SHRT1_EXEV9[6:0]
	rw		rw

Bit Field	Name	Description
31 23 25 7	Reserved	Reserved, the reset value must be maintained.
30:24 22:16 14:8 6:0	SHRT1_EXEVx[6:0]	SHRTIM_EXEVx external event input channel configuration bit (x = 9 … 10) These bits can be read and written by software to select the external input source for SHRTIM_EXEVx. 0000000:PA[0] 0000001:PB[0] 0000010:PC[0] 0000011:PD[0] 0000100:PE[0] 0000101:PF[0] 0000110:PG[0] 0000111:PH[0] 0001000:PA[1] 0001001:PB[1] 0001010:PC[1] 0001011:PD[1] 0001100:PE[1] 0001101:PF[1] 0001110:PG[1] 0001111:PH[1] 0010000:PA[2] 0010001:PB[2] 0010010:PC[2] 0010011:PD[2] 0010100:PE[2] 0010101:PF[2] 0010110:PG[2] 0010111:PH[2] 0011000:PA[3] 0011001:PB[3] 0011010:PC[3] 0011011:PD[3] 0011100:PE[3] 0011101:PF[3] 0011110:PG[3] 0011111:PH[3] 0100000:PA[4] 0100001:PB[4] 0100010:PC[4] 0100011:PD[4] 0100100:PE[4] 0100101:PF[4] 0100110:PG[4] 0100111:PH[4] 0101000:PA[5] 0101001:PB[5] 0101010:PC[5] 0101011:PD[5] 0101100:PE[5] 0101101:PF[5] 0101110:PG[5] 0101111:PH[5] 0110000:PA[6] 0110001:PB[6] 0110010:PC[6] 0110011:PD[6] 0110100:PE[6] 0110101:PF[6] 0110110:PG[6] 0110111:PH[6] 0111000:PA[7] 0111001:PB[7] 0111010:PC[7] 0111011:PD[7] 0111100:PE[7] 0111101:PF[7] 0111110:PG[7] 1000000:PA[8] 1000001:PB[8] 1000010:PC[8] 1000011:PD[8] 1000100:PE[8] 1000101:PF[8] 1000110:PG[8] 1001000:PA[9] 1001001:PB[9] 1001010:PC[9] 1001011:PD[9] 1001100:PE[9] 1001101:PF[9] 1001110:PG[9] 1010000:PA[10] 1010001:PB[10] 1010010:PC[10] 1010011:PD[10] 1010100:PE[10] 1010101:PF[10] 1010110:PG[10] 1011000:PA[11] 1011001:PB[11] 1011010:PC[11] 1011011:PD[11] 1011100:PE[11] 1011101:PF[11] 1011110:PG[11] 1100000:PA[12] 1100001:PB[12] 1100010:PC[12] 1100011:PD[12] 1100100:PE[12] 1100101:PF[12] 1100110:PG[12] 1101000:PA[13] 1101001:PB[13] 1101010:PC[13] 1101011:PD[13] 1101100:PE[13] 1101101:PF[13] 1101110:PG[13] 1110000:PA[14] 1110001:PB[14] 1110010:PC[14] 1110011:PD[14] 1110100:PE[14] 1110101:PF[14] 1110110:PG[14] 1111000:PA[15] 1111001:PB[15] 1111010:PC[15] 1111011:PD[15] 1111100:PE[15] 1111101:PF[15] 1111110:PG[15]

Bit Field	Name	Description
		Others: reserved

6 Interrupts And Events

6.1 Nested Vectored Interrupt Register

Features

- 118 maskable interrupt channels (excluding 16 Cortex^{TM}-M4F interrupt lines).
- 16 programmable priority levels (using 4-bit interrupt priority);
- Low-latency exception and interrupt handling;
- Power management control;
- Implementation of system control registers;

The nested vectored interrupt controller (NVIC) is closely linked to the processor core, enabling low latency interrupt processing and efficient processing of late interrupts. The nested vectored interrupt controller manages interrupts including core exceptions.

6.1.1 SysTick Calibration Value Register

The system tick calibration value is fixed at 9000. When the system tick clock is set to 9MHz, 1ms time base is generated. If you need to configure to another time base, configure the corresponding SysTick.LOAD register based on the clock source period.

6.1.2 Interrupt And Exception Vectors

Table 6-1 Vector Table

Position	Priority	Priority type	Name	Description	Address
-	-	-	-	Reserved	0x0000_0000
-	-3	Fixed	Reset	Reset	0x0000_0004
-	-2	Fixed	NMI	Non-maskable interrupt. RCC clock security system (CSS) is connected to NMI vector.	0x0000_0008
-	-1	Fixed	HardFault	All types of errors (fault)	0x0000_000C
-	0	Settable	MemManage	Memory management	0x0000_0010
-	1	Settable	BusFault	Prefetch means failure. Memory access failed	0x0000_0014
-	2	Settable	UsageFault	Undefined instruction or illegal status	0x0000_0018
-	-	-	-	Reserved	0x0000_001C ~0x0000_002B
-	3	Settable	SVCall	System services invoked by SWI directives	0x0000_002C
-	4	Settable	DebugMonitor	Debug monitor	0x0000_0030

Position	Priority	Priority type	Name	Description	Address
-	-	-	-	Reserved	0x0000_0034
-	5	Settable	PendSV	System services that can be suspended	0x0000_0038
-	6	Settable	SysTick	System tick timer	0x0000_003C
0	7	Settable	WWDG	Window timer interrupt	0x0000_0040
1	8	Settable	PVD	Power supply voltage detection (PVD) interrupt connected to EXTI line 16	0x0000_0044
2	9	Settable	RTC_TAMPER	RTC timestamp interrupt connected to EXTI line 19	0x0000_0048
3	10	Settable	RTC_WKUP	Real time clock (RTC) wake up interrupt connected to EXTI line 20	0x0000_004C
4	11	Settable	FLASH	Flash global interrupt	0x0000_0050
5	12	Settable	RCC	Reset and clock control (RCC) interruption	0x0000_0054
6	13	Settable	EXTI0	EXTI line 0 interrupt	0x0000_0058
7	14	Settable	EXTI1	EXTI line 1 interrupt	0x0000_005C
8	15	Settable	EXTI2	EXTI line 2 interrupt	0x0000_0060
9	16	Settable	EXTI3	EXTI line 3 interrupt	0x0000_0064
10	17	Settable	EXTI4	EXTI line 4 interrupt	0x0000_0068
11	18	Settable	The DMA1 channel 1	DMA1 channel 1 global interrupt	0x0000_006C
12	19	Settable	The DMA1 channel 2	DMA1 channel 2 global interrupt	0x0000_0070
13	20	Settable	The DMA1 channel 3	DMA1 channel 3 global interrupt	0x0000_0074
14	21	Settable	The DMA1 channel 4	DMA1 channel 4 global interrupt	0x0000_0078
15	22	Settable	The DMA1 channel 5	DMA1 channel 5 global interrupt	0x0000_007C
16	23	Settable	The DMA1 channel 6	DMA1 channel 6 global interrupt	0x0000_0080
17	24	Settable	The DMA1 channel 7	DMA1 channel 7 global interrupt	0x0000_0084
18	25	Settable	ADC1_2	ADC1 and ADC2 global interrupt	0x0000_0088
19	26	Settable	USB_FS_HP	USB FS Device high priority interrupt	0x0000_008C
20	27	Settable	USB_FS_LP	USB FS Device low priority interrupt	0x0000_0090
21	28	Settable	FDCAN1_INT0	FDCAN1 interrupt 0	0x0000_0094
22	29	Settable	FDCAN1_INT1	FDCAN1 interrupt 1	0x0000_0098
23	30	Settable	EXTI9_5	EXTI[9:5] interrupt lines	0x0000_009C
24	31	Settable	ATIM1_BRK	ATIM1 brake interrupt	0x0000_00A0
25	32	Settable	ATIM1_UP	ATIM1 update interrupt	0x0000_00A4
26	33	Settable	ATIM1_TRG_COM	ATIM1 triggers and communication interrupt	0x0000_00A8
27	34	Settable	ATIM1_CC	ATIM1 capture comparison interrupt	0x0000_00AC
28	35	Settable	GTIM1	GTIM1 global interrupt	0x0000_00B0
29	36	Settable	GTIM2	GTIM2 global interrupt	0x0000_00B4
30	37	Settable	GTIM3	GTIM3 global interrupt	0x0000_00B8
31	38	Settable	I2C1_EV	I2C1 event interrupt	0x0000_00BC

Position	Priority	Priority type	Name	Description	Address
32	39	Settable	I2C1_ER	I2C1 error interrupt	0x0000_00C0
33	40	Settable	I2C2_EV	I2C2 event interrupt	0x0000_00C4
34	41	Settable	I2C2_ER	I2C2 error interrupt	0x0000_00C8
35	42	Settable	SPI1	SPI1 global interrupt	0x0000_00CC
36	43	Settable	SPI2_I2S2	SPI2/I2S2 global interrupt	0x0000_00D0
37	44	Settable	USART1	USART1 global interrupt	0x0000_00D4
38	45	Settable	USART2	USART2 global interrupt	0x0000_00D8
39	46	Settable	USART3	USART3 global interrupt	0x0000_00DC
40	47	Settable	EXTI15_10	The EXTI line [15:10] interrupt	0x0000_00E0
41	48	Settable	RTCAlarm	The RTC alarm interrupt connected to EXTI line 17	0x0000_00E4
42	49	Settable	USB_FS_WKUP	USB FS Device wake up failure interrupt connected to EXTI line 18	0x0000_00E8
43	50	Settable	ATIM2_BRK	ATIM2 brake failure interrupt	0x0000_00EC
44	51	Settable	ATIM2_UP	ATIM2 update interrupt	0x0000_00F0
45	52	Settable	ATIM2_TRG_COM	ATIM2 triggers and communication interrupt	0x0000_00F4
46	53	Settable	ATIM2_CC	ATIM2 capture comparison interrupt	0x0000_00F8
47	54	Settable	ADC3_ADC4	ADC3 and ADC4 global interrupt	0x0000_00FC
48	55	Settable	UCDR	UCDR global interrupt	0x0000_0100
49	56	Settable	SDIO	SDIO global interrupt	0x0000_0104
50	57	Settable	GTIM4	GTIM4 global interrupt	0x0000_0108
51	58	Settable	SPI3_I2S3	SPI3/I2S3 global interrupt	0x0000_010C
52	59	Settable	USART4	USART4 global interrupt	0x0000_0110
53	60	Settable	UART5	UART5 send interrupt	0x0000_0114
54	61	Settable	BTIM1	BTIM1 global interrupt	0x0000_0118
55	62	Settable	BTIM2	BTIM2 global interrupt	0x0000_011C
56	63	Settable	DMA2_Channel1	DMA2 channel 1 global interrupt	0x0000_0120
57	64	Settable	DMA2_Channel2	DMA2 channel 2 global interrupt	0x0000_0124
58	65	Settable	DMA2_Channel3	DMA2 channel 3 global interrupt	0x0000_0128
59	66	Settable	DMA2_Channel4	DMA2 channel 4 global interrupt	0x0000_012C
60	67	Settable	DMA2_Channel5	DMA2 channel 5 global interrupt	0x0000_0130
61	68	Settable	SRAM_ERR	SRAM ECC error interrupt	0x0000_0134
62	69	Settable	COMP4/5/6	Interrupts connected to EXTI line 22/30/31 from COMP4/5/6	0x0000_0138
63	70	Settable	FDCAN2_INT0	FDCAN2 interrupt 0	0x0000_013C
64	71	Settable	FDCAN2_INT1	FDCAN2 interrupt 1	0x0000_0140
65	72	Settable	FDCAN3_INT0	FDCAN3 interrupt 0	0x0000_0144
66	73	Settable	FDCAN3_INT1	FDCAN3 interrupt 1	0x0000_0148
67	74	Settable	xSPI	xSPI global interrupt	0x0000_014C

Position	Priority	Priority type	Name	Description	Address
68	75	Settable	DMA2_Channel6	DMA2 channel 6 global interrupt	0x0000_0150
69	76	Settable	DMA2_Channel7	DMA2 channel 7 global interrupt	0x0000_0154
70	77	Settable	I2C3_EV	I2C3 event interrupt	0x0000_0158
71	78	Settable	I2C3_ER	I2C4 error interrupt	0x0000_015C
72	79	Settable	I2C4_EV	I2C3 event interrupt	0x0000_0160
73	80	Settable	I2C4_ER	I2C4 error interrupt	0x0000_0164
74	81	Settable	UART6	UART6 global interrupt	0x0000_0168
75	82	Settable	UART7	UART7 global interrupt	0x0000_016C
76	83	Settable	DMA1_Channel8	DMA1 channel 8 global interrupt	0x0000_0170
77	84	Settable	DMA2_Channel8	DMA2 channel 8 global interrupt	0x0000_0174
78	85	Settable	DVP	DVP global interrupt	0x0000_0178
79	86	Settable	SAC	SAC global interrupt	0x0000_017C
80	87	Settable	SMPU	SMPU global interrupt	0x0000_0180
81	88	Settable	COMP1/2/3	Interrupts connected to EXTI line 21/28/29 from COMP1/2/3	0x0000_0184
82	89	Settable	USB_HS_EPx_OUT	USB HS endpoint output global interrupt	0x0000_0188
83	90	Settable	USB_HS_EPx_IN	USB HS endpoint input global interrupt	0x0000_018C
84	91	Settable	USB_HS_WKUP	USB HS wake-up interrupt connected to EXTI line 23	0x0000_0190
85	92	Settable	UART8	UART8 global interrupt	0x0000_0194
86	93	Settable	SPI4	SPI4 global interrupt	0x0000_0198
87	94	Settable	ETH	ETH global interrupt	0x0000_019C
88	95	Settable	ETH_WKUP	Interrupts connected to EXTI line 27 from ETH wakeup	0x0000_01A0
89	96	Settable	GTIM5	GTIM5 global interrupt	0x0000_01A4
90	97	Settable	COMP7	Interrupts connected to EXTI line 24 from COMP7	0x0000_01A8
91	98	Settable	SHRT1_MST	SHRTIM1 main timer interrupt	0x0000_01AC
92	99	Settable	SHRT1_TIMA	SHRTIM1 timer unit A interrupt	0x0000_01B0
93	100	Settable	SHRT1_TIMB	SHRTIM1 timer unit B interrupt	0x0000_01B4
94	101	Settable	SHRT1_TIMC	SHRTIM1 timer unit C interrupt	0x0000_01B8
95	102	Settable	SHRT1_TIMD	SHRTIM1 timer unit D interrupt	0x0000_01BC
96	103	Settable	SHRT1_TIME	SHRTIM1 timer unit E interrupt	0x0000_01C0
97	104	Settable	SHRT1_TIMF	SHRTIM1 timer unit F interrupt	0x0000_01C4
98	105	Settable	SHRT1_FLT	SHRTIM1 error interrupt	0x0000_01C8
99	106	Settable	ATIM3_BRK	ATIM3 brake interrupt	0x0000_01CC
100	107	Settable	ATIM3_UP	ATIM3 update interrupt	0x0000_01D0
101	108	Settable	ATIM3_TRG_COM	ATIM3 triggers and communication interrupt	0x0000_01D4
102	109	Settable	ATIM3_CC	ATIM3 capture comparison interrupt	0x0000_01D8

Position	Priority	Priority type	Name	Description	Address
103	110	Settable	GTIM6	GTIM6 global interrupt	0x0000_01DC
104	111	Settable	LPTIM1_WKUP	Interrupts connected to EXTI line 25 from LPTIM1 wakeup	0x0000_01E0
105	112	Settable	LPTIM2_WKUP	Interrupts connected to EXTI line 26 from LPTIM2 wakeup	0x0000_01E4
106	113	Settable	FMAC_IRQ	FMAC global interrupt	0x0000_01E8
107	114	Settable	Cordic_IRQ	Cordic global interrupt	0x0000_01EC
108	115	Settable	SPI5	SPI5 global interrupt	0x0000_01F0
109	116	Settable	SPI6	SPI6 global interrupt	0x0000_01F4
110	117	Settable	GTIM7	GTIM7 global interrupt	0x0000_01F8
111	118	Settable	GTIM8	GTIM8 global interrupt	0x0000_01FC
112	119	Settable	GTIM9	GTIM9 global interrupt	0x0000_0200
113	120	Settable	GTIM10	GTIM10 global interrupt	0x0000_0204
114	121	Settable	DAC1/2/3/4 Underflow	DAC1/2/3/4 underflow interrupt	0x0000_0208
115	122	Settable	DAC5/6/7/8 Underflow	DAC5/6/7/8 underflow interrupt	0x0000_020C
116	123	Settable	USB_HS	USB HS global interrupt	0x0000_0210
117	124	Settable	FEMC_INT	FEMC NAND interrupt	0x0000_0214

6.2 Extended Interrupt/Event Controller (EXTI)

6.2.1 Introduction

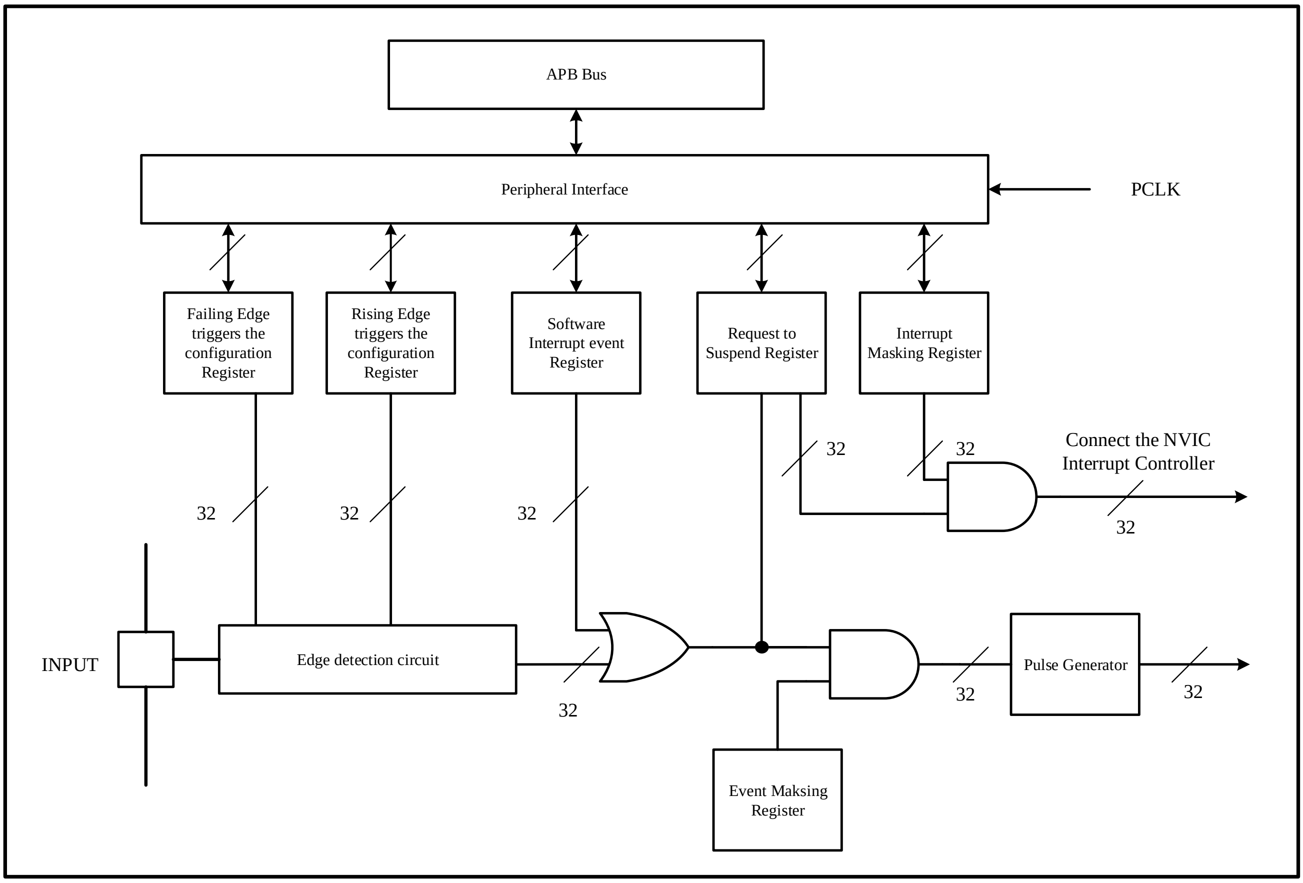
The extended interrupt/event controller contains 32 edge detection circuits that generate interrupt/event triggers. Each input line can be independently configured with pulse or pending input types, and 3 trigger event types including rising edge, falling edge or double edges, which can also be independently shielded. Interrupt requests that hold the state line in the pending register can be cleared by writing '1' in the corresponding bit of the pending register.

6.2.2 Main Features

The main features of EXTI controller are as follows:

- Support 32 software interrupt/event requests.
- Interrupts/events corresponding to each input line can be configured to trigger or mask independently.
- Each interrupt line has an independent state bit.
- Support for pulse or pending input types.
- 3 trigger events are supported: rising edge, falling edge, and double edge.
- Can wake up MCU to exit low power mode.

Figure 6-1 External Interrupt/Event Controller Block Diagram


APB Bus
Peripheral Interface
PCLK
Failing Edge triggers the configuration Register
Rising Edge triggers the configuration Register
Software Interrupt event Register
Request to Suspend Register
Interrupt Masking Register
32
32
Connect the NVIC Interrupt Controller
32
32
32
32
INPUT
Edge detection circuit
32
Pulse Generator
32
32
Event Maksing Register

6.2.3 Functional Description

The EXTI contains 32 interrupt lines, 16 lines from I/O pins and 16 lines from internal modules. To generate interrupts, the NVIC interrupt channel of the extended interrupt controller must be configured to enable the appropriate interrupt lines. Select rising edge, falling edge, or double edges trigger event types by edge trigger configuration registers EXTI_RT_CFG and EXTI_FT_CFG, and write '1' to the corresponding bit of interrupt masking register EXTI_IMASK to allow interrupt requests. When a preset edge trigger polarity is detected on the external interrupt line, an interrupt request is generated and the corresponding pending bit is set to '1'. Writing '1' to the corresponding bit of the pending register clears the interrupt request.

To generate events, the corresponding event line must be configured and enabled. According to the desired edge detection polarity, set up the rise/fall edge trigger configuration register, while writing '1' in the corresponding bit of the event masking register to allow interrupt requests. When a preset edge occurs on an event line, an event request pulse is generated and the corresponding pending bit is not set to '1'.

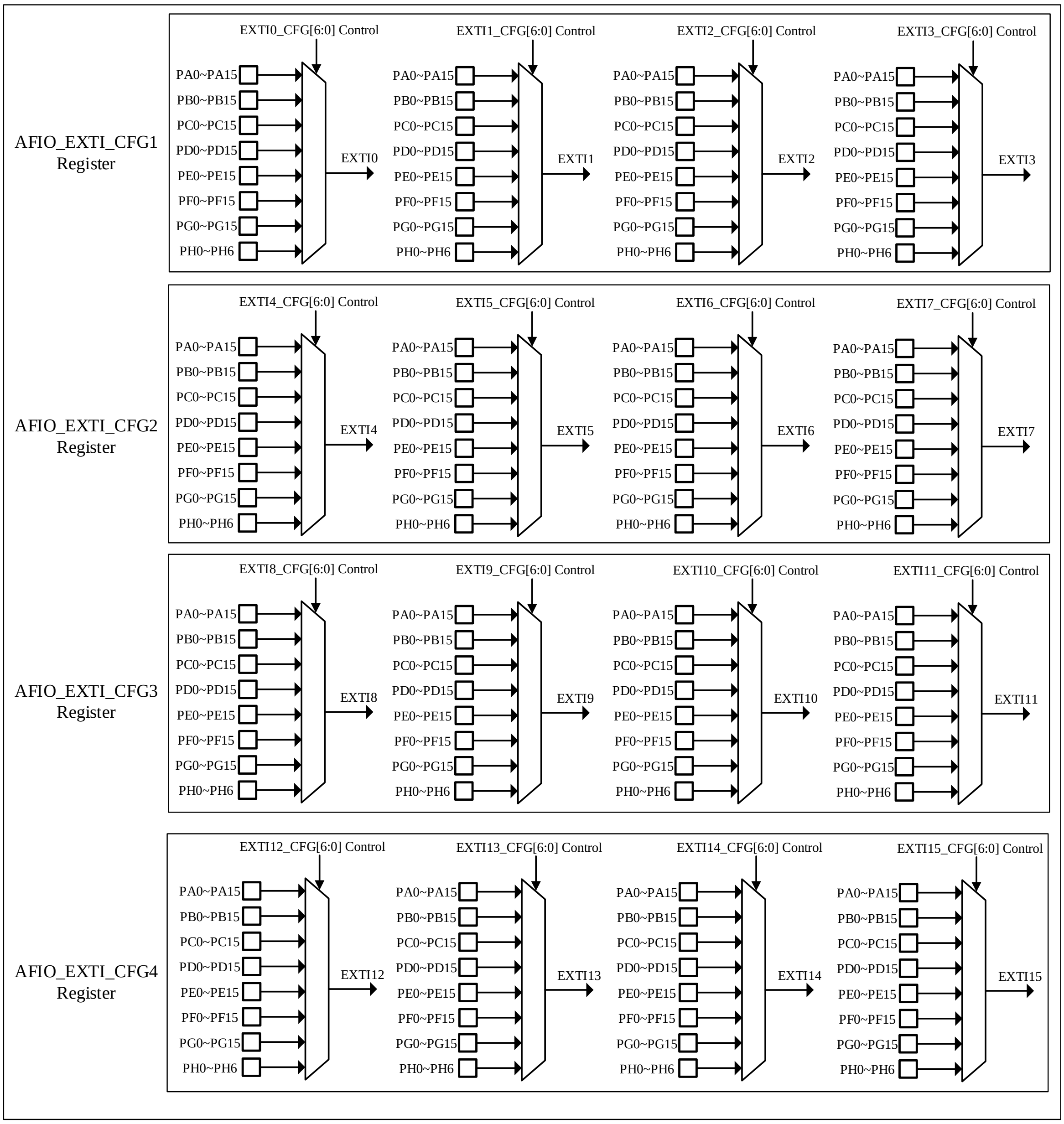
In addition, interrupt/event requests can also be generated by software by writing a '1' in the software interrupt/event register.

- Hardware interrupt configuration, select and configure 32 lines as interrupt sources as required:
  - Configure the mask bit (EXTI_IMASK) for 32 interrupt lines.

  - Configure the Trigger Selection bits of the selected interrupt line (EXTI_RT_CFG and EXTI_FT_CFG);
  - Configure the enable and mask bits of the NVIC interrupt channel corresponding to the external interrupt controller so that the requests in the 32 interrupt lines can be correctly responded to.
- Hardware event configuration: Select 32 lines as event sources as required:
  - Configure the mask bit (EXTI_EMASK) for 32 event lines.
  - Configure the Trigger Selection bits for the selected event line (EXTI_RT_CFG and EXTI_FT_CFG).
- Software interrupt/event configuration, select 32 lines as software interrupt/event lines as required:
  - Configure 32 interrupt/event line mask bits (EXTI_IMASK and EXTI_EMASK).
  - Configure the request bit of the software interrupt event register (EXTI_SWIE).

6.2.4 EXTI Line Mapping

Figure 6-2 External Interrupt Generic I/O Mapping



To configure external interrupts/events on the GPIO line using AFIO_EXTI_CFGy, the AFIO clock must be enabled first. General I/O ports are connected to 16 external interrupt/event lines as shown above. The connection mode of the other 16 EXTI lines is as follows:

- EXTI line 16 is connected to the PVD output
- EXTI line 17 is connected to the RTC alarm event
- EXTI line 18 is connected to the USB FS Device alarm
- EXTI line 19 is connected to the RTC tamper or LSE-CSS event

- EXTI line 20 is connected to the RTC Wake up event
- EXTI line 21 is connected to the COMP1 output
- EXTI line 22 is connected to the COMP4 output
- EXTI line 23 is connected to the USB HS Dualrole wake up interrupt
- EXTI line 24 is connected to the COMP7 interrupt
- EXTI line 25 is connected to the LPTIM1 wake up interrupt
- EXTI line 26 is connected to the LPTIM2 wake up interrupt
- EXTI line 27 is connected to the ETH wake up interrupt
- EXTI line 28 is connected to the COMP2 output
- EXTI line 29 is connected to the COMP3 output
- EXTI line 30 is connected to the COMP5 output
- EXTI line 31 is connected to the COMP6 output

6.3 EXTI Registers

EXTI base address: 0x40010400

6.3.1 EXTI Register Overview

Table 6-2 EXTI Register Overview

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
000h	EXTI_EMASK	EMASK[31:0]																															
	Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
004h	EXTI_IMASK	IMASK[31:0]																															
	Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
008h	EXTI_FT_CFG	FT_CFG[31:0]																															
	Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
00Ch	EXTI_RT_CFG	RT_CFG[31:0]																															
	Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
010h	EXTI_PEND	PEND31	PEND30	PEND29	PEND28	PEND27	PEND26	PEND25	PEND24	PEND23	PEND22	PEND21	PEND20	PEND19	PEND18	PEND17	PEND16	PEND15	PEND14	PEND13	PEND12	PEND11	PEND10	PEND9	PEND8	PEND7	PEND6	PEND5	PEND4	PEND3	PEND2	PEND1	PEND0
	Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
014h	EXTI_SWIE	SWIE[31:0]																															
	Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
018h	EXTI_TS_SEL	Reserved																												TSSEL[3:0]			
	Reset Value																													0	0	0	0

6.3.2 EXTI Event Mask Register (EXTI_EMASK)

Address offset: 0x00

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EMASK[31:16]															

rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EMASK[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit Field	Name	Description
31:0	EMASKx	Event mask on line x 0: Masking the event requests from line x. 1: Not masking the event requests from line x.

6.3.3 EXTI Interrupt Mask Register (EXTI_IMASK)

Address offset: 0x04

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IMASK[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IMASK[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit Field	Name	Description
31:0	IMASKx	Interrupt mask on line x 0: Masking the interrupt requests from line x. 1: Not masking the interrupt requests from line x.

6.3.4 EXTI Falling Edge Trigger Configuration Register (EXTI_FT_CFG)

Address offset: 0x08

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FT_CFG[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FT_CFG[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit Field	Name	Description
31:0	FT_CFGx	The falling edge on line x triggers the configuration bit 0: Disable falling edge triggering (interrupts and events) on input line x 1: Enable falling edge triggering (interrupts and events) on input line x

6.3.5 EXTI Rising Edge Trigger Configuration Register (EXTI_RT_CFG)

Address offset: 0x0C

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RT_CFG[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RT_CFG[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit Field	Name	Description
31:0	RT_CFGx	The rising edge on line x triggers the configuration bit 0: Disable rising edge triggering (interrupts and events) on input line x 1: Enable rising edge triggering (interrupts and events) on input line x

6.3.6 EXTI Pending Register (EXTI_PEND)

Address offset: 0x10

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PEND[31:16]															
rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PEND[15:0]															
rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1

Bit Field	Name	Description
31:0	PENDx	Pending bit on line x 0: No pending request has occurred 1: A pending trigger request occurred This bit is set to '1' when a selected edge trigger event occurs on the external interrupt line. It can be cleared by writing '1' to the bit, or by changing the polarity of the edge detection.

6.3.7 EXTI Software Interrupt Event Register (EXTI_SWIE)

Address offset: 0x14

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SWIE[31:16]															

rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SWIE[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit Field	Name	Description
31:0	SWIEx	Software interrupt on line X When the bit is' 0 ', writing '1' sets the corresponding pending bit in EXTI_PEND. If this interrupt is allowed in EXTI_IMASK and EXTI_EMASK, an interrupt will be generated. *Note: This bit can be cleared to '0' by writing '1' to clear the corresponding bit of EXTI_PEND.*

6.3.8 EXTI Timestamp Trigger Source Selection Register (EXTI_TS_SEL)

Address offset: 0x18

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												TSSEL[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit Field	Name	Description
31:4	Reserved	Reserved, the reset value must be maintained.
3:0	TSSEL[3:0]	Select the external interrupt input as the trigger source for the timestamp event 0000: Select EXTI0 as the trigger source of the timestamp event; 0001: Select EXTI1 as the trigger source of the timestamp event. ... 1111: Select EXTI15 as the trigger source for the timestamp event.

7 DMA Controller

7.1 Introduction

The DMA controller can access slaves: Flash, Sram1, Sram2, CCM Sram3, FEMC, XSPI, CRC, SDIO, FMAC, CORDIC, APB1, APB2, ATIM, SHRTIM, ADC, DAC.

DMA Controller is controlled by CPU to perform fast data transfer from source to destination. After configuration, data can be transferred without CPU intervention. Thus, CPU can be released for other computation/control tasks or save overall system power consumption.

The chip has two DMA controllers (DMA1, DMA2), each with 8 logical channels. Each logical channel is used to serve memory access requests from one or multiple peripherals. An internal arbiter controls the priority of different DMA channels.

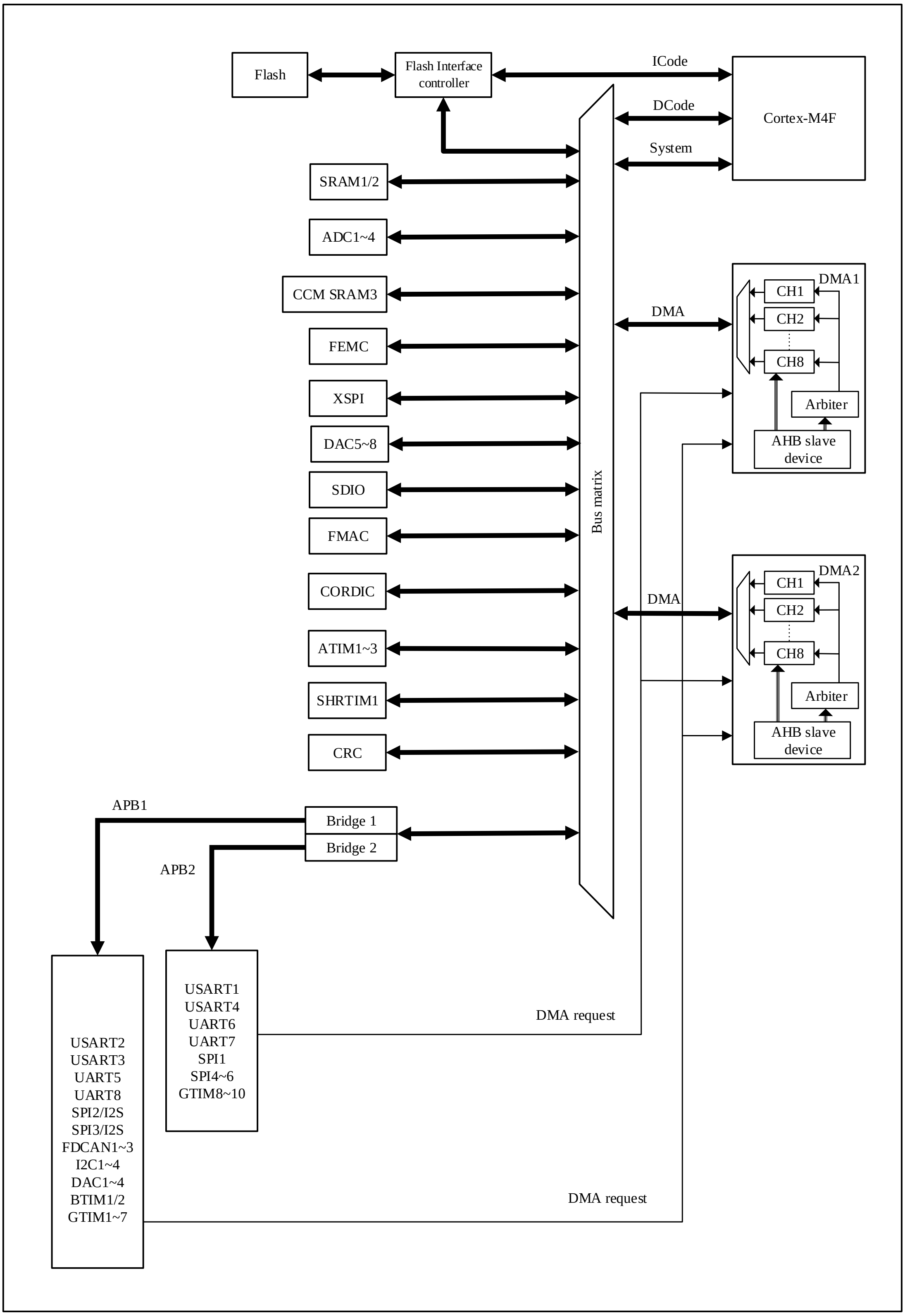
7.2 Main Features

DMA main features:

- 16 DMA channels which can be configured independently: DMA1 and DMA2 each have 8 channels.
- Supports three types of transfers: memory-to-memory, memory-to-peripheral, and peripheral-to-memory.
- Each DMA channel supports hardware requests and software triggers to initiate transfer, and is configured by software.
- Each DMA channel has dedicated software priority level (DMA_CHCFGx.PRIOLVL [1:0] bits, corresponding to 4 levels of priority) which can be configured individually. Channels with the same software priority level will further compare hardware index (channel number) to decide final priority (lower index number channel will has higher priority).
- Configurable source and destination size. Address setting should correspond to data size.
- Configurable circular transfer mode for each channel.
- Each channel has 3 independent event flags and interrupts (Transfer complete, Half transfer, Transfer error), and 1 global interrupt flag (set by logical or of 3 events).
- Access slaves: Flash, Sram1,Sram2, CCM Sram3, FEMC, XSPI, CRC, SDIO,FMAC, CORDIC, APB1, APB2, ATIM, SHRTIM, ADC, DAC.
- Configurable data transmit number (0~65535).
- Supports burst transfers, burst length is configurable, can be set to 1/2/3/4/5/6/7/8 units.

7.3 Block Diagram

Figure 7-1 DMA Block Diagram



7.4 Function Description

DMA controller and Cortex^{TM}-M4F core share the same system data bus. When CPU and DMA access the same destination (RAM or peripheral) at the same time, DMA request will suspend CPU from accessing the system bus for several cycles, and the bus arbiter will perform round-robin scheduling. This allows the CPU to get at least half of the system bus (memory or peripheral) bandwidth.

7.4.1 DMA Operation

A DMA request can be triggered by hardware peripherals or software, and the DMA controller processes the request according to the priority level of the channel. Data is read from the source address according to the configured transfer address and bit width, and then the read data is stored in the destination address space. After one operation, the controller calculates the number of remaining transfers and updates the source address and the destination address of the next transfer.

Each DMA data transfer consists of three operations:

- Data access: determine the source address (DMA_PADDRx or DMA_MADDRx) according to the transfer direction and read data from the source address.
- Data storage: determine the destination address (DMA_PADDRx or DMA_MADDRx) according to the transfer direction and store the read data into the destination address space.
- Calculate the number of outstanding operations, perform a decrement operation of the DMA_TXNUMx register, and update the source and destination addresses of the next operation.

7.4.2 Channel Priority And Arbitration

The DMA uses an arbitration strategy to handle multiple requests from different channels. The priority of each channel is programmable in the channel control register (DMA_CHCFGx).

4 levels of priority:

- Very high priority
- High priority
- Medium priority
- Low priority

By default, channel with lower index has higher priority if the programmed priority is the same.

7.4.3 DMA Channels And Number Of Transfers

Each channel can perform DMA transfer between the peripheral register at the specified address and the memory address. The number of data transferred by DMA is programmable, and the maximum supported value is 65535. The DMA_TXNUM register is decremented after each transfer.

7.4.4 Programmable Data Width

Peripheral and memory transfer data bit width supports byte, half-word and word, which can be programmed through DMA_CHCFGx.PSIZE and DMA_CHCFGx.MSIZE.

When DMA_CHCFGx.PSIZE and DMA_CHCFGx.MSIZE are different, the DMA module aligns the data according to the Table 7-1 below.

Table 7-1 Programmable Data Width And Endian Operation (when PINC = MINC = 1)

Source width (bit)	Destina-tion width (bit)	Number of transfer (Byte)	Source: Address / data	Transfer operations (R: Read, W: Write)	Destination: Address / data
8	8	4	0x0 / B0	1: R B0 [7:0] @0x0, W B0 [7:0] @0x0	0x0 / B0
			0x1 / B1	2: R B1 [7:0] @0x1, W B1 [7:0] @0x1	0x1 / B1
			0x2 / B2	3: R B2 [7:0] @0x2, W B2 [7:0] @0x2	0x2 / B2
			0x3 / B3	4: R B3 [7:0] @0x3, W B3 [7:0] @0x3	0x3 / B3
8	16	4	0x0 / B0	1: R B0 [7:0] @0x0, W 00B0 [15:0] @0x0	0x0 / 00B0
			0x1 / B1	2: R B1 [7:0] @0x1, W 00B1 [15:0] @0x2	0x2 / 00B1
			0x2 / B2	3: R B2 [7:0] @0x2, W 00B2 [15:0] @0x4	0x4 / 00B2
			0x3 / B3	4: R B3 [7:0] @0x3, W 00B3 [15:0] @0x6	0x6 / 00B3
8	32	4	0x0 / B0	1: R B0 [7:0] @0x0, W 000000B0 [31:0] @0x0	0x0 / 000000B0
			0x1 / B1	2: R B1 [7:0] @0x1, W 000000B1 [31:0] @0x4	0x4 / 000000B1
			0x2 / B2	3: R B2 [7:0] @0x2, W 000000B2 [31:0] @0x8	0x8 / 000000B2
			0x3 / B3	4: R B3 [7:0] @0x3, W 000000B3 [31:0] @0xC	0xC / 000000B3
16	8	4	0x0 / B1B0	1: R B1B0 [15:0] @0x0, W B0 [7:0] @0x0	0x0 / B0
			0x2 / B3B2	2: R B3B2 [15:0] @0x2, W B2 [7:0] @0x1	0x1 / B2
			0x4 / B5B4	3: R B5B4 [15:0] @0x4, W B4 [7:0] @0x2	0x2 / B4
			0x6 / B7B6	4: R B7B6 [15:0] @0x6, W B6 [7:0] @0x3	0x3 / B6
16	16	4	0x0 / B1B0	1: R B1B0 [15:0] @0x0, W B1B0 [15:0] @0x0	0x0 / B1B0
			0x2 / B3B2	2: R B3B2 [15:0] @0x2, W B3B2 [15:0] @0x2	0x2 / B3B2
			0x4 / B5B4	3: R B5B4 [15:0] @0x4, W B5B4 [15:0] @0x4	0x4 / B5B4
			0x6 / B7B6	4: R B7B6 [15:0] @0x6, W B7B6 [15:0] @0x6	0x6 / B7B6
16	32	4	0x0 / B1B0	1: R B1B0 [15:0] @0x0, W 0000B1B0 [31:0] @0x0	0x0 / 0000B1B0
			0x2 / B3B2	2: R B3B2 [15:0] @0x2, W 0000B3B2 [31:0] @0x4	0x4 / 0000B3B2
			0x4 / B5B4	3: R B5B4 [15:0] @0x4, W 0000B5B4 [31:0] @0x8	0x8 / 0000B5B4
			0x6 / B7B6	4: R B7B6 [15:0] @0x6, W 0000B7B6 [31:0] @0xC	0xC / 0000B7B6
32	8	4	0x0 / B3B2B1B0	1: R B3B2B1B0 [31:0] @0x0, W B0 [7:0] @0x0	0x0 / B0
			0x4 / B7B6B5B4	2: R B7B6B5B4 [31:0] @0x4, W B4 [7:0] @0x1	0x1 / B4
			0x8 / BBBAB9B8	3: R BBBAB9B8 [31:0] @0x8, W B8 [7:0] @0x2	0x2 / B8
			0xC / BFBEBDBC	4: R BFBEBDBC [31:0] @0xC, W BC [7:0] @0x3	0x3 / BC

Source width (bit)	Destina-tion width (bit)	Number of transfer (Byte)	Source: Address / data	Transfer operations (R: Read, W: Write)	Destination: Address / data
32	16	4	0x0 / B3B2B1B0 0x4 / B7B6B5B4 0x8 / BBBAB9B8 0xC / BFBEBDBC	1: R B3B2B1B0 [31:0] @0x0, W B1B0 [15:0] @0x0 2: R B7B6B5B4 [31:0] @0x4, W B5B4 [15:0] @0x2 3: R BBBAB9B8 [31:0] @0x8, W B9B8 [15:0] @0x4 4: R BFBEBDBC [31:0] @0xC, W BDBC [15:0] @0x6	0x0 / B1B0 0x2 / B5B4 0x4 / B9B8 0x6 / BDBC
32	32	4	0x0 / B3B2B1B0 0x4 / B7B6B5B4 0x8 / BBBAB9B8 0xC / BFBEBDBC	1: R B3B2B1B0 [31:0] @0x0, W B3B2B1B0 [31:0] @0x0 2: R B7B6B5B4 [31:0] @0x4, W B7B6B5B4 [31:0] @0x4 3: R BBBAB9B8 [31:0] @0x8, W BBBAB9B8 [31:0] @0x8 4: R BFBEBDBC [31:0] @0xC, W BFBEBDBC [31:0] @0xC	0x0 / B3B2B1B0 0x4 / B7B6B5B4 0x8 / BBBAB9B8 0xC / BFBEBDBC

Notice:

DMA always provide full 32-bits data to HWDATA[31:0] no matter what destination size it is (HSIZE still follows destination size setting for device supports byte/half-word operation). The HWDATA[31:0] follows the following rules:

- When source size is smaller than destination size, the DMA pads the MSB with 0 until their sizes match and duplicates it to be 32 bits. E.g., source is 8 bits data 0x55 and destination size is 16 bits. DMA fills the source data with 0 to make it 16 bits and become 0x0055, then duplicate it to 32-bit data 0x0055_0055 and provide to HWDATA[31:0]; (if destination size is 32-bit then DMA will only pad source data with 0).
- When source size is larger or equal to destination size and smaller than 32 bits, DMA duplicates source data to 32 bits data. E.g., source data is 8 bits data 0x1F, HWDATA[31:0] = 0x1F1F_1F1F. If source data is 16 bits data 0x2345, then HWDATA[31:0] = 0x2345_2345.

This ensures peripherals that only support word operation won't generate bus error and the desired data can still move to the place we want with extra bits i.e. 0 padding. If user wants to configure an 8-bit register but is aligned to a 32-bit address boundary, the source size should be set to 8 bits and destination to 32 bits so extra bits will be padded with 0.

7.4.5 Peripheral/Memory Address Incrementation

DMA_CHCFGx.PINC and DMA_CHCFGx.MINC respectively control whether the peripheral address and memory address are enabled in auto-increment mode. The software cannot (can read) write the address register during transfer.

- In auto-increment mode, the next address to be transferred is automatically increased according to the data bit width (1, 2 or 4) after each transfer. The address of the first transfer is stored in DMA_PADDRx or DMA_MADDRx register.
- In fixed mode, the address is always fixed to the initial address.

At the end of transfer (i.e. the transfer count changes to 0), different processes will be carried out according to whether the current work is under circular mode or not.

- In non-circular mode, DMA stops after the transfer is completed. To start a new DMA transfer, need to rewrite the transfer number in the DMA_TXNUMx register with the DMA channel disabled.
- In circular mode, at the end of a transfer, the content of the DMA_TXNUMx register will be automatically reloaded to its initial value, and the current internal peripheral or memory address register will also be reloaded to the initial base address set by the DMA_PADDRx or DMA_MADDRx register.

7.4.6 Single Transfer And Burst Transfer

The DMA controller can generate single transfers or burst transfers of 2 to 8 units. DMA_BURSTEN. BURST_BYPASS controls whether burst mode is enabled, and burst mode is not enabled by default.

The burst length is configurable and can be set to 1/2/3/4/5/6/7/8 units, with 1 unit representing a single transfer. The unit depends on the data width setting (word/half-word/byte).

7.4.7 Channel Configuration Process

The detail configuration process is as below:

1. In the DMA_CHCFGx register:
   a) Set the channel transfer direction (DIR bit)
   b) Set the circular mode (CIRC bit)
   c) Set the peripheral and memory address increment (PINC bit and MINC bit)
   d) Set the data width of the peripheral and memory (PSIZE[1:0] bits and MSIZE[1:0] bits)
   e) Set the channel priority (PRIOLVL[1:0] bits)
   f) If it is memory-to-memory, set the MEM2MEM bit (Note: To configure the DMA to operate in M2M mode, the user needs to set the corresponding channel selection value to a reserved value)
2. In the DMA_TXNUMx register, set the channel transfer count.
3. Configure the DMA_MADDRx register to set the channel peripheral address.
4. Configure the DMA_PADDRx register to set the memory address.
5. Configure DMA_BURSTEN. BURST_BYPASS = 1, this bit is default to 1, where 1 indicates that burst function is not enabled.
6. Configure the interrupt bits (transfer half-complete, transfer complete, and transfer error) in the DMA_CHCFGx register, 1: enable interrupt, 0: disable interrupt.
7. Repeat steps 1 to 6 on channels 1 to 8.
8. Finally, enable the corresponding channel.

The detailed configuration process for enabling DMA burst mode is as follows:

1. In the DMA_CHCFGx register:
   a) Set the channel transfer direction (DIR bit)

b) Set the circular mode (CIRC bit)

c) Set the peripheral and memory address increment (PINC bit and MINC bit)

d) Set the data width of the peripheral and memory (PSIZE[1:0] bits and MSIZE[1:0] bits)

e) Set the channel priority (PRIOLVL[1:0] bits)

f) If it is memory-to-memory, set the MEM2MEM bit (Note: To configure the DMA to operate in M2M mode, the user needs to set the corresponding channel selection value to a reserved value)

g) Set the burst mode (BURST_MODE bit)

h) Set the burst length (BLEN[3:0] bits)

2. In the DMA_TXNUMx register, set the channel transfer count.
3. Configure the DMA_MADDRx register to set the channel peripheral address.
4. Configure the DMA_PADDRx register to set the memory address.
5. Configure DMA_BURSTEN. BURST_BYPASS = 0, this bit is default to 1, where 1 indicates that burst function is not enabled.
6. Configure the interrupt bits (transfer half-complete, transfer complete, and transfer error) in the DMA_CHCFGx register, 1: enable interrupt, 0: disable interrupt.
7. Repeat steps 1 to 6 on channels 1 to 8.
8. Finally, enable the corresponding channel.

Notes:

(1) Before enabling a channel, the registers of that channel must be pre-configured by the current user. If other users configure the registers of that channel, it may result in permission error events.

(2) If software is used to serve interrupt, the software must enquire interrupt status register to check which interrupt occurred (software needs to write 1 to interrupt flag clear bit to clear the corresponding interrupt). Before enable channel, all interrupts corresponding to the channel should be cleared.

(3)If the interrupt is transfer complete interrupt, software can configure the next transfer, or report to user this channel transformation is done.

(4) The configuration sequence must be executed in the order specified in the user manual.

7.4.8 Transfer Types

Three major transfer types are supported:

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

Flow control is controlled by two register bits in each DMA channel configuration register. Flow control is used to control source/destination and direction of DMA channel.

Table 7-2 Flow Control Table

DMA_CHCFGx.MEM2MEM	DMA_CHCFGx.DIR	Source	Destination	Transfer[1]
1	x	Memory	Memory	AHB read to AHB write, can do back2back transfer
0	1	Memory	AHB Peripheral	AHB read to AHB write, single transfer
			APB Peripheral	AHB read to APB write, single transfer
0	0	AHB Peripheral	Memory	AHB read to AHB write, single transfer
		APB Peripheral		APB read to AHB write, single transfer

Note: [1] Support back-to-back transfers.

7.4.9 Circular Mode

The circular mode is used to process circular buffers and continuous data transmission (such as ADC scan mode). The DMA_CHCFGx.CIRC is used to enable this function. When the circular mode is activated, if the number of data to be transferred becomes 0, it will automatically be restored to the initial value when configuring the channel, and the DMA operation will continue.

If the user wants to turn off the circular mode, the user needs to write 0 to DMA_CHCFGx.CHEN to disable the DMA channel, and then write 0 to DMA_CHCFGx.CIRC (when DMA_CHCFGx.CHEN is 1, other bits in the DMA_CHCFGx register cannot be rewritten).

7.4.10 Error Management

DMA access to a reserved address space will cause DMA transmission errors. When an error occurs, the transfer error flag is set, and the hardware automatically clears the current DMA channel enable bit (DMA_CHCFGx.CHEN), and the channel operation is stopped. If the transfer error interrupt enable bit is set in the DMA_CHCFGx register, an interrupt will be generated.

7.4.11 Interrupt

- Transfer complete interrupt:

  An interrupt is generated when channel data transfer is complete. Interrupt is a level signal. Each channel has its dedicated interrupt, interrupt mask control and interrupt status bit. Interrupt status bit is cleared when interrupt flag clear bit is set.

- Half transfer interrupt:

  An interrupt is generated when half of the channel data is transferred. Interrupt is a level signal. Each channel has its dedicated interrupt, interrupt mask control and interrupt status bit. Interrupt status bit is cleared when interrupt flag clear bit is set.

- Transfer error interrupt:

  An interrupt is generated when bus returned error. Interrupt is a level signal. Each channel has its dedicated interrupt, interrupt mask control and interrupt status bit. Interrupt status bit is cleared when interrupt flag clear bit is set.

Table 7-3 DMA Interrupt Request

Interrupt event	Event flag bit	Enable control bit
Half transfer	HTXF	HTXIE
Transfer complete	TXCF	TXCIE
Transfer error	ERRF	ERRIE

7.4.12 DMA Request Mapping

Totally there are 150 DMA requests from all the peripherals. To have better support with full flexibility, register bits can be used to select which DMA request is mapped to which DMA channel. The table blow show the mapping scheme of peripherals' DMA request to DMA controller's DMA channels.

Table 7-4 DMA Request Mapping

DMA request source select	Peripheral DMA request	DMA request source select	Peripheral DMA request
Set = 0	ADC1	Set = 76	GTIM3_CH4
Set = 1	ADC2	Set = 77	GTIM3_UP
Set = 2	ADC3	Set = 78	GTIM3_TRIG
Set = 3	ADC4	Set = 79	GTIM4_CH1
Set = 4	USART1_TX	Set = 80	GTIM4_CH2
Set = 5	USART1_RX	Set = 81	GTIM4_CH3
Set = 6	USART2_TX	Set = 82	GTIM4_CH4
Set = 7	USART2_RX	Set = 83	GTIM4_UP
Set = 8	USART3_TX	Set = 84	GTIM4_TRIG
Set = 9	USART3_RX	Set = 85	GTIM5_CH1
Set = 10	USART4_TX	Set = 86	GTIM5_CH2
Set = 11	USART4_RX	Set = 87	GTIM5_CH3
Set = 12	UART5_TX	Set = 88	GTIM5_CH4
Set = 13	UART5_RX	Set = 89	GTIM5_UP
Set = 14	UART6_TX	Set = 90	GTIM5_TRIG
Set = 15	UART6_RX	Set = 91	GTIM6_CH1
Set = 16	UART7_TX	Set = 92	GTIM6_CH2
Set = 17	UART7_RX	Set = 93	GTIM6_CH3

Set = 18	UART8_TX	Set = 94	GTIM6_CH4
Set = 19	UART8_RX	Set = 95	GTIM6_UP
Set = 20	SPI1_TX	Set = 96	GTIM6_TRIG
Set = 21	SPI1_RX	Set = 97	GTIM7_CH1
Set = 22	SPI2_I2S2_TX	Set = 98	GTIM7_CH2
Set = 23	SPI2_I2S2_RX	Set = 99	GTIM7_CH3
Set = 24	-	Set = 100	GTIM7_CH4
Set = 25	-	Set = 101	GTIM7_UP
Set = 26	SPI3_I2S3_TX	Set = 102	GTIM7_TRIG
Set = 27	SPI3_I2S3_RX	Set = 103	GTIM8_CH1
Set = 28	-	Set = 104	GTIM8_CH2
Set = 29	-	Set = 105	GTIM8_CH3
Set = 30	SPI4_TX	Set = 106	GTIM8_CH4
Set = 31	SPI4_RX	Set = 107	GTIM8_COM
Set = 32	SPI5_TX	Set = 108	GTIM8_UP
Set = 33	SPI5_RX	Set = 109	GTIM8_TRIG
Set = 34	SPI6_TX	Set = 110	GTIM9_CH1
Set = 35	SPI6_RX	Set = 111	GTIM9_CH2
Set = 36	XSPI_TX	Set = 112	GTIM9_CH3
Set = 37	XSPI_RX	Set = 113	GTIM9_CH4
Set = 38	I2C1_TX	Set = 114	GTIM9_COM
Set = 39	I2C1_RX	Set = 115	GTIM9_UP
Set = 40	I2C2_TX	Set = 116	GTIM9_TRIG
Set = 41	I2C2_RX	Set = 117	GTIM10_CH1
Set = 42	I2C3_TX	Set = 118	GTIM10_CH2
Set = 43	I2C3_RX	Set = 119	GTIM10_CH3
Set = 44	I2C4_TX	Set = 120	GTIM10_CH4
Set = 45	I2C4_RX	Set = 121	GTIM10_COM
Set = 46	SDIO_DMA	Set = 122	GTIM10_UP
Set = 47	DAC1_DMA	Set = 123	GTIM10_TRIG
Set = 48	DAC2_DMA	Set = 124	ATIM1_CH1
Set = 49	DAC3_DMA	Set = 125	ATIM1_CH2
Set = 50	DAC4_DMA	Set = 126	ATIM1_CH3
Set = 51	DAC5_DMA	Set = 127	ATIM1_CH4
Set = 52	DAC6_DMA	Set = 128	ATIM1_COM
Set = 53	DAC7_DMA	Set = 129	ATIM1_UP
Set = 54	DAC8_DMA	Set = 130	ATIM1_TRIG
Set = 55	FMAC_WR	Set = 131	ATIM2_CH1
Set = 56	FMAC_RD	Set = 132	ATIM2_CH2
Set = 57	CORDIC_WR	Set = 133	ATIM2_CH3
Set = 58	CORDIC_RD	Set = 134	ATIM2_CH4
Set = 59	BTIM1_UP	Set = 135	ATIM2_COM

Set = 60	BTIM2_UP	Set = 136	ATIM2_UP
Set = 61	GTIM1_CH1	Set = 137	ATIM2_TRIG
Set = 62	GTIM1_CH2	Set = 138	ATIM3_CH1
Set = 63	GTIM1_CH3	Set = 139	ATIM3_CH2
Set = 64	GTIM1_CH4	Set = 140	ATIM3_CH3
Set = 65	GTIM1_UP	Set = 141	ATIM3_CH4
Set = 66	GTIM1_TRIG	Set = 142	ATIM3_COM
Set = 67	GTIM2_CH1	Set = 143	ATIM3_UP
Set = 68	GTIM2_CH2	Set = 144	ATIM3_TRIG
Set = 69	GTIM2_CH3	Set = 145	SHRTIM1_CH1
Set = 70	GTIM2_CH4	Set = 146	SHRTIM1_CH2
Set = 71	GTIM2_UP	Set = 147	SHRTIM1_CH3
Set = 72	GTIM2_TRIG	Set = 148	SHRTIM1_CH4
Set = 73	GTIM3_CH1	Set = 149	SHRTIM1_CH5
Set = 74	GTIM3_CH2	Set = 150	SHRTIM1_CH6
Set = 75	GTIM3_CH3	Set = 151	SHRTIM1_CH7

Note: 1.Different DMA channels cannot use the same request source, otherwise only the channel with higher priority will be triggered.

2. When the DMA is set to memory-to-memory mode, the DMA request mapping needs to be set to (Set = 24).

7.5 DMA Registers

7.5.1 DMA Register Overview

Table 7-5 DMA Register Overview

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	DMA_CHCFG1	Reserved												BLEN[3:0]				BURST_MODE	MEM2MEM	PRIOLVL[1:0]		MSIZE[1:0]		PSIZE[1:0]		MINC	PINC	CIRC	DIR	ERRIE	HTXIE	TXCIE	CHEN
	Reset value													0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x04	DMA_MADDR1	ADDR[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x08	DMA_PADDR1	ADDR[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0C	DMA_CHSEL1	Reserved																								CH_SEL[7:0]							
	Reset value																									0	0	0	0	0	0	0	0
0x10	DMA_TXNUM1	Reserved																NDTX[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x14	DMA_CHCFG2	Reserved												BLEN[3:0]				BURST_MODE	MEM2MEM	PRIOLVL[1:0]		MSIZE[1:0]		PSIZE[1:0]		MINC	PINC	CIRC	DIR	ERRIE	HTXIE	TXCIE	CHEN
	Reset value													0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x18	DMA_MADDR2	ADDR[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x1C	DMA_PADDR2	ADDR[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x20	DMA_CHSEL2	Reserved																								CH_SEL[7:0]							
	Reset value																									0	0	0	0	0	0	0	0
0x24	DMA_TXNUM2	Reserved																NDTX[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0		0
0x28	DMA_CHCFG3	Reserved												BLEN[3:0]				BURST_MODE	MEM2MEM	PRIOLVL[1:0]		MSIZE[1:0]		PSIZE[1:0]		MINC	PINC	CIRC	DIR	ERRIE	HTXIE	TXCIE	CHEN
	Reset value													0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x2C	DMA_MADDR3	ADDR[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x30	DMA_PADDR3	ADDR[31:0]																															

	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x34	DMA_CHSEL3	Reserved																								CH_SEL[7:0]							
	Reset value																									0	0	0	0	0	0	0	0
0x38	DMA_ TXNUM3	Reserved																NDTX[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0		0
0x3C	DMA_CHCFG4	Reserved												BLEN[3:0]				BURST_MODE	MEM2MEM	PRIOLVL[1:0]		MSIZE[1:0]		PSIZE[1:0]		MINC	PINC	CIRC	DIR	ERRIE	HTXIE	TXCIE	CHEN
	Reset value													0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x40	DMA_ MADDR4	ADDR[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x44	DMA_ PADDR4	ADDR[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x48	DMA_CHSEL4	Reserved																								CH_SEL[7:0]							
	Reset value																									0	0	0	0	0	0	0	0
0x4C	DMA_ TXNUM4	Reserved																NDTX[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0		0
0x50	DMA_CHCFG5	Reserved												BLEN[3:0]				BURST_MODE	MEM2MEM	PRIOLVL[1:0]		MSIZE[1:0]		PSIZE[1:0]		MINC	PINC	CIRC	DIR	ERRIE	HTXIE	TXCIE	CHEN
	Reset value													0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x54	DMA_ MADDR5	ADDR[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x58	DMA_ PADDR5	ADDR[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x5C	DMA_CHSEL5	Reserved																								CH_SEL[7:0]							
	Reset value																									0	0	0	0	0	0	0	0
0x60	DMA_ TXNUM5	Reserved																NDTX[15:0]															

	Reset value	Reserved																0	0	0	0	0	0	0	0	0	0	0	0	0	0		0
0x64	DMA_CHCFG6	Reserved												BLEN[3:0]				BURST_MODE	MEM2MEM	PRIOLVL[1:0]		MSIZE[1:0]		PSIZE[1:0]		MINC	PINC	CIRC	DIR	ERRIE	HTXIE	TXCIE	CHEN
	Reset value													0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x68	DMA_MADDR6	ADDR[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x6C	DMA_PADDR6	ADDR[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x70	DMA_CHSEL6	Reserved																								CH_SEL[7:0]							
	Reset value																									0	0	0	0	0	0	0	0
0x74	DMA_TXNUM6	Reserved																NDTX[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0		0
0x78	DMA_CHCFG7	Reserved												BLEN[3:0]				BURST_MODE	MEM2MEM	PRIOLVL[1:0]		MSIZE[1:0]		PSIZE[1:0]		MINC	PINC	CIRC	DIR	ERRIE	HTXIE	TXCIE	CHEN
	Reset value													0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x7C	DMA_MADDR7	ADDR[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x80	DMA_PADDR7	ADDR[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x84	DMA_CHSEL7	Reserved																								CH_SEL[7:0]							
	Reset value																									0	0	0	0	0	0	0	0
0x88	DMA_TXNUM7	Reserved																NDTX[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x8C	DMA_CHCFG8	Reserved												BLEN[3:0]				BURST_MODE	MEM2MEM	PRIOLVL[1:0]		MSIZE[1:0]		PSIZE[1:0]		MINC	PINC	CIRC	DIR	ERRIE	HTXIE	TXCIE	CHEN
	Reset value													0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x90	DMA_MADDR8	ADDR[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

0x94	DMA_ PADDR8	ADDR[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x98	DMA_CHSEL8	Reserved																								CH_SEL[7:0]							
	Reset value																									0	0	0	0	0	0	0	0
0x9C	DMA_ TXNUM8	Reserved																NDTX[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0xA0	DMA_INTCLR	CERRF8	CHTXF8	CTXCF8	CGLBF8	CERRF7	CHTXF7	CTXCF7	CGLBF7	CERRF6	CHTXF6	CTXCF6	CGLBF6	CERRF5	CHTXF5	CTXCF5	CGLBF5	CERRF4	CHTXF4	CTXCF4	CGLBF4	CERRF3	CHTXF3	CTXCF3	CGLBF3	CERRF2	CHTXF2	CTXCF2	CGLBF2	CERRF1	CHTXF1	CTXCF1	CGLBF1
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0xA4	DMA_INTSTS	ERRF8	HTXF8	TXCF8	GLBF8	ERRF7	HTXF7	TXCF7	GLBF7	ERRF6	HTXF6	TXCF6	GLBF6	ERRF5	HTXF5	TXCF5	GLBF5	ERRF4	HTXF4	TXCF4	GLBF4	ERRF3	HTXF3	TXCF3	GLBF3	ERRF2	HTXF2	TXCF2	GLBF2	ERRF1	HTXF1	TXCF1	GLBF1
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0xA8	DMA_ BURSTEN	Reserved																															BURST_BYPASS
	Reset value																																1

7.5.2 DMA Channel x Configuration Register (DMA_CHCFGx)

Note: The x is channel number, x = 1…8

Address offset: 0x00+20 * (x–1)

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved												BLEN[3:0]			
												rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BURST_MODE	MEM2MEM	PRIOLVL[1:0]		MSIZE[1:0]		PSIZE[1:0]		MINC	PINC	CIRC	DIR	ERRIE	HTXIE	TXCIE	CHEN
rw	rw	rw		rw		rw		rw	rw	rw	rw	rw	rw	rw	rw

Bit Field	Name	Description
31:20	Reserved	Reserved, the reset value must be maintained.
19:16	BLEN[3:0]	Burst length 0~1: len = 1 (DMA is single transfer at this point) Greater than 8: len = 8
15	BURST_MODE	Burst mode 0: In burst mode, the DMA will try to transfer data in burst mode as much as possible, and only use single transfer when the remaining data at the end is less

Bit Field	Name	Description
		than the burst length; 1: In burst mode, the DMA will dynamically transfer data in single and burst mode based on the requests from the peripheral and burst requests.
14	MEM2MEM	Memory to memory mode. Software can configure this channel to memory to memory transfer when it is not yet enabled. 0: Channel transfer between memory and peripheral. 1: Channel set to memory to memory transfer.
13:12	PRIOLVL[1:0]	Channel priority. Software can program channel priority when channel is not enable. 00: Low 01: Medium 10: High 11: Very high
11:10	MSIZE[1:0]	Memory data size. Software can configure data size read/write from/to memory address. 00: 8-bits 01: 16-bits 10: 32-bits 11: Reserved
9:8	PSIZE[1:0]	Peripheral data size. Software can configure data size read/write from/to peripheral address. 00: 8-bits 01: 16-bits 10: 32-bits 11: Reserved
7	MINC	Memory increment mode. Software can enable/disable memory address increment mode. 0: Memory address won't increase with each transfer. 1: Memory address increase with each transfer.
6	PINC	Peripheral increment mode. Software can enable/disable peripheral address increment mode. 0: Peripheral address won't increase with each transfer. 1: Peripheral address increase with each transfer.
5	CIRC	Circular mode. Software can set/clear this bit. 0: Channel will stop after one round of transfer. 1: Channel configure as circular mode.
4	DIR	Data transfer direction Software can set/clear this bit. 0: Data transfer from Peripheral to Memory 1: Data transfer from Memory to Peripheral.

Bit Field	Name	Description
3	ERRIE	Transfer error interrupt enable. Software can enable/disable transfer error interrupt. 0: Disable transfer error interrupt of channel x. 1: Enable transfer error interrupt of channel x.
2	HTXIE	Half transfer interrupt enable. Software can enable/disable half transfer interrupt. 0: Disable half transfer interrupt of channel x. 1: Enable half transfer interrupt of channel x.
1	TXCIE	Transfer complete interrupt enable. Software can enable/disable transfer complete interrupt. 0: Disable transfer complete interrupt of channel x. 1: Enable transfer complete interrupt of channel x.
0	CHEN	Channel enable. Software can set/reset this bit. 0: Disable channel. 1: Enable channel.

7.5.3 DMA Channel x Memory Address Register (DMA_MADDRx)

Note: The x is channel number, x = 1…8

Address offset: 0x04+20 * (x–1)

Reset value: 0x0000 0000

This register can only be written if the channel is disabled (DMA_CHCFGx.CHEN = 0).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADDR[31:0]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR[31:0]															
rw															

Bit Field	Name	Description
31:0	ADDR[31:0]	ADDR Memory address. Memory starting address for DMA to read/write from/to. Increment of address will be decided by DMA_CHCFGx.MSIZE. With DMA_CHCFGx.MSIZE equal to 01, DMA ignores bit 0 of MADDR and if DMA_CHCFGx.MSIZE equal to 10 DMA will ignore bit [1:0] of MADDR.

7.5.4 DMA Channel x Peripheral Address Register (DMA_PADDRx)

Note: The x is channel number, x = 1…8

Address offset: 0x08+20 * (x–1)

Reset value: 0x0000 0000

This register can only be written if the channel is disabled (DMA_CHCFGx.CHEN = 0).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADDR[31:0]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR[31:0]															
rw															

Bit Field	Name	Description
31:0	ADDR[31:0]	Peripheral address. Peripheral starting address for DMA to read/write from/to. Increment of address will be decided by DMA_CHCFGx.PSIZE. With DMA_CHCFGx.PSIZE equal to 01, DMA ignores bit 0 of PADDR and if DMA_CHCFGx.PSIZE equal to 10 DMA will ignore bit [1:0] of PADDR.

7.5.5 DMA Channel x Request Select Register (DMA_CHSELx)

Note: The x is channel number, x = 1…8

Address offset: 0x0C+20 * (x–1)

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								CH_SEL[7:0]							
								rw							

Bit Field	Name	Description
31:8	Reserved	Reserved, the reset value must be maintained.
7:0	CH_SEL[7:0]	DMA channel request selection 0: ADC1 1: ADC2 2: ADC3 3: ADC4 4: USART1_TX 5: USART1_RX 6: USART2_TX 7: USART2_RX

Bit Field	Name	Description
		8: USART3_TX 9: USART3_RX 10: USART4_TX 11: USART4_RX 12: UART5_TX 13: UART5_RX 14: UART6_TX 15: UART6_RX 16: UART7_TX 17: UART7_RX 18: UART8_TX 19: UART8_RX 20: SPI1_TX 21: SPI1_RX 22: SPI2_I2S2_TX 23: SPI2_I2S2_RX 24: - 25: - 26: SPI3_I2S3_TX 27: SPI3_I2S3_RX 28: - 29: - 30: SPI4_TX 31: SPI4_RX 32: SPI5_TX 33: SPI5_RX 34: SPI6_TX 35: SPI6_RX 36: XSPI_TX 37: XSPI_RX 38: I2C1_TX 39: I2C1_RX 40: I2C2_TX 41: I2C2_RX 42: I2C3_TX 43: I2C3_RX 44: I2C4_TX 45: I2C4_RX 46: SDIO_DMA 47: DAC1_DMA 48: DAC2_DMA 49: DAC3_DMA

Bit Field	Name	Description
		50: DAC4_DMA 51: DAC5_DMA 52: DAC6_DMA 53: DAC7_DMA 54: DAC8_DMA 55: FMAC_WR 56: FMAC_RD 57: CORDIC_WR 58: CORDIC_RD 59: BTIM1_UP 60: BTIM2_UP 61: GTIM1_CH1 62: GTIM1_CH2 63: GTIM1_CH3 64: GTIM1_CH4 65: GTIM1_UP 66: GTIM1_TRIG 67: GTIM2_CH1 68: GTIM2_CH2 69: GTIM2_CH3 70: GTIM2_CH4 71: GTIM2_UP 72: GTIM2_TRIG 73: GTIM3_CH1 74: GTIM3_CH2 75: GTIM3_CH3 76: GTIM3_CH4 77: GTIM3_UP 78: GTIM3_TRIG 79: GTIM4_CH1 80: GTIM4_CH2 81: GTIM4_CH3 82: GTIM4_CH4 83: GTIM4_UP 84: GTIM4_TRIG 85: GTIM5_CH1 86: GTIM5_CH2 87: GTIM5_CH3 88: GTIM5_CH4 89: GTIM5_UP 90: GTIM5_TRIG 91: GTIM6_CH1

Bit Field	Name	Description
		92: GTIM6_CH2 93: GTIM6_CH3 94: GTIM6_CH4 95: GTIM6_UP 96: GTIM6_TRIG 97: GTIM7_CH1 98: GTIM7_CH2 99: GTIM7_CH3 100: GTIM7_CH4 101: GTIM7_UP 102: GTIM7_TRIG 103: GTIM8_CH1 104: GTIM8_CH2 105: GTIM8_CH3 106: GTIM8_CH4 107: GTIM8_COM 108: GTIM8_UP 109: GTIM8_TRIG 110: GTIM9_CH1 111: GTIM9_CH2 112: GTIM9_CH3 113: GTIM9_CH4 114: GTIM9_COM 115: GTIM9_UP 116: GTIM9_TRIG 117: GTIM10_CH1 118: GTIM10_CH2 119: GTIM10_CH3 120: GTIM10_CH4 121: GTIM10_COM 122: GTIM10_UP 123: GTIM10_TRIG 124: ATIM1_CH1 125: ATIM1_CH2 126: ATIM1_CH3 127: ATIM1_CH4 128: ATIM1_COM 129: ATIM1_UP 130: ATIM1_TRIG 131: ATIM2_CH1 132: ATIM2_CH2 133: ATIM2_CH3

Bit Field	Name	Description
		134: ATIM2_CH4 135: ATIM2_COM 136: ATIM2_UP 137: ATIM2_TRIG 138: ATIM3_CH1 139: ATIM3_CH2 140: ATIM3_CH3 141: ATIM3_CH4 142: ATIM3_COM 143: ATIM3_UP 144: ATIM3_TRIG 145: SHRTIM1_CH1 146: SHRTIM1_CH2 147: SHRTIM1_CH3 148: SHRTIM1_CH4 149: SHRTIM1_CH5 150: SHRTIM1_CH6 151: SHRTIM1_CH7

7.5.6 DMA Channel x Transfer Number Register (DMA_TXNUMx)

Note: The x is channel number, x = 1…8

Address offset: 0x0c+20 * (x–1)

Reset value: 0x0000 0000

This register can only be written if the channel is disabled (DMA_CHCFGx.CHEN = 0).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NDTX[15:0]															
rw															

Bit Field	Name	Description
31:16	Reserved	Reserved, the reset value must be maintained.
15:0	NDTX[15:0]	Number of data to transfer. Number of data to be transferred (0～65535). Software can read/write the number of transfers when channel is disable and it will be read only after channel enable. Every successful transfer of corresponding DMA channel will decrease this register by 1. If circular mode is enable, it will automatically reload pre-set value when it reach zero. Otherwise it will keep at zero and reset channel enable.

7.5.7 DMA Interrupt Flag Clear Register (DMA_INTCLR)

Address offset: 0xA0

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CERR F8	CHTX F8	CTXC F8	CGLB F8	CERR F7	CHTX F7	CTXC F7	CGLB F7	CERR F6	CHTX F6	CTXC F6	CGLB F6	CERR F5	CHTX F5	CTXC F5	CGLB F5
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CERR F4	CHTX F4	CTXC F4	CGLB F4	CERR F3	CHTX F3	CTXC F3	CGLB F3	CERR F2	CHTX F2	CTXC F2	CGLB F2	CERR F1	CHTX F1	CTXC F1	CGLB F1
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

Bit Field	Name	Description
31/27/23/19/15/11/7/3	CERRFx	Clear transfer error flag for channel x (x=1…8). Software can set this bit to clear ERRF of corresponding channel. 0: No action. 1: Reset DMA_INTSTS.ERRF bit of corresponding channel.
30/26/22/18/14/10/6/2	CHTXFx	Clear half transfer flag for channel x (x=1…8). Software can set this bit to clear HTXF of corresponding channel. 0: No action. 1: Reset DMA_INTSTS.HTXF bit of corresponding channel.
29/25/21/17/13/9/5/1	CTXCFx	Clear transfer complete flag for channel x (x=1…8). Software can set this bit to clear TXCF of corresponding channel. 0: No action. 1: Reset DMA_INTSTS.TXCF bit of corresponding channel.
28/24/20/16/12/8/4/0	CGLBFx	Clear global event flag for channel x (x=1…8). Software can set this bit to clear GLBF of corresponding channel. 0: No action. 1: Reset DMA_INTSTS.GLBF bit of corresponding channel.

7.5.8 DMA Interrupt Status Register (DMA_INTSTS)

Address offset: 0xA4

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ERRF 8	HTXF 8	TXCF 8	GLBF 8	ERRF 7	HTXF 7	TXCF 7	GLBF 7	ERRF 6	HTXF 6	TXCF 6	GLBF 6	ERRF 5	HTXF 5	TXCF 5	GLBF 5
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ERRF 4	HTXF 4	TXCF 4	GLBF 4	ERRF 3	HTXF 3	TXCF 3	GLBF 3	ERRF 2	HTXF 2	TXCF 2	GLBF 2	ERRF 1	HTXF 1	TXCF 1	GLBF 1
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

Bit Field	Name	Description
31/27/23/19/15/11/7/3	ERRFx	Transfer error flag for channel x (x=1…8). Hardware sets this bit when transfer error happen. This bit is cleared by software by writing ‘1’ to DMA_INTCLR.CERRFx bit. 0: Transfer error no happened on channel x. 1: Transfer error happened on channel x.
30/26/22/18/14/10/6/2	HTXFx	Half transfer flag for channel x (x=1…8). Hardware sets this bit when half transfer is done. This bit is cleared by software by writing ‘1’ to DMA_INTCLR.CHTXFx bit. 0: Half transfer not yet done on channel x. 1: Half transfer was done on channel x.
29/25/21/17/13/9/5/1	TXCFx	Transfer complete flag for channel x (x=1…8). Hardware sets this bit when transfer is done. This bit is cleared by software by writing ‘1’ to DMA_INTCLR.CTXCFx bit. 0: Transfer not yet done on channel x. 1: Transfer was done on channel x.
28/24/20/16/12/8/4/0	GLBFx	Global flag for channel x (x=1…8). Hardware sets this bit when any interrupt events happen in this channel. This bit is cleared by software by writing ‘1’ to DMA_INTCLR.CGLBFx bit. 0: No transfer error, half transfer or transfer done event happen on channel x. 1: One of transfer error, half transfer or transfer done event happen on channel x.

7.5.9 DMA Burst Transfer Enable Register (DMA_BURSTEN)

Address offset:0xA8

Reset value:0x0000 0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															BURST_BYPASS
															rw

Bit Field	Name	Description
31:1	Reserved	Reserved, the reset value must be maintained.
0	BURST_BYPASS	DMA burst function 1: Disable DMA burst function 0: Enable DMA burst function

8 CRC Calculation Unit

8.1 CRC Introduction

This module integrates the functions of CRC32 and CRC16, and the cyclic redundancy check (CRC) calculation unit obtains any CRC calculation result according to a fixed generator polynomial. In other applications, CRC technology is mainly used to verify the correctness and integrity of data transmission or data storage. EN/IEC 60335-1 provides a method to verify the integrity of Flash memory. CRC calculation unit can calculate the signature of the software when the program is running, then compare it with the reference signature generated during connection, and then store it in the specified memory space.

8.2 CRC Main Features

8.2.1 CRC32 Module

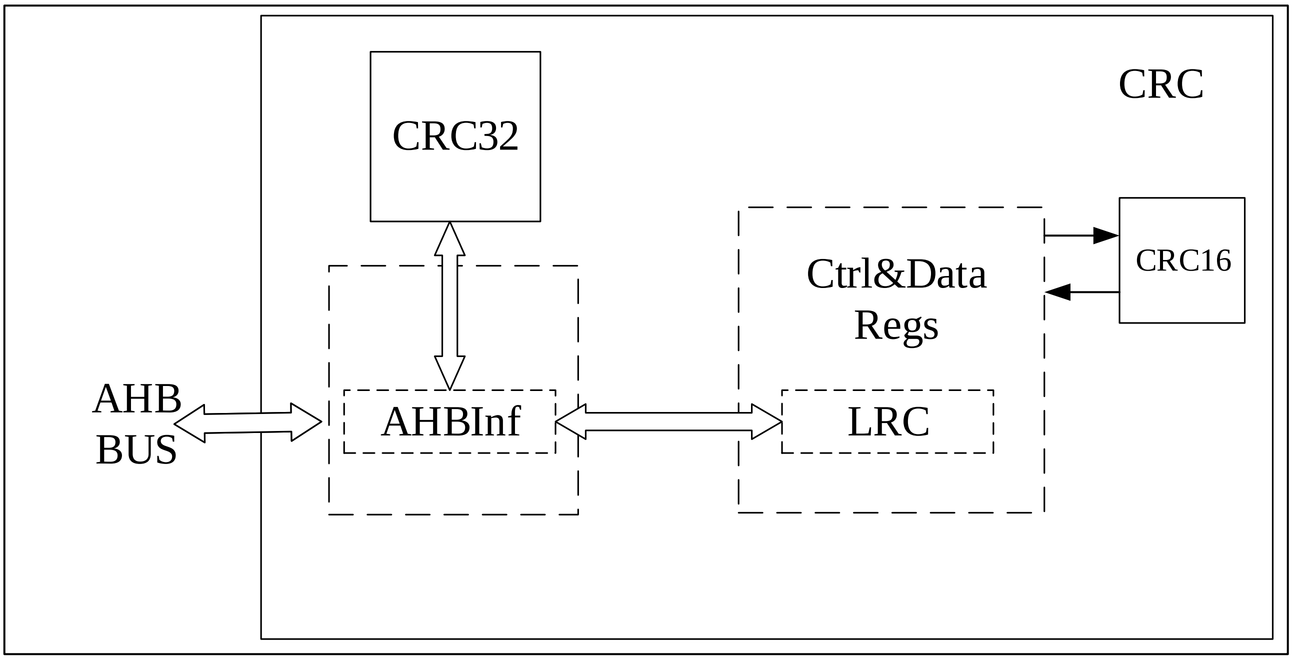
- CRC32 ($X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^{8} + X^{7} + X^{5} + X^{4} + X^{2} + X + 1$)
- 32 bits of data to be checked and 32 bits of output check code.
- CRC calculation time: 1 AHB clock cycles (HCLK)
- General-purpose 8-bit register (can be used to store temporary data)
- Initial value for cyclic redundancy check (CRC) calculation is configurable

8.2.2 CRC16 Module

- CRC16 ($X^{16} + X^{15} + X^{2} + 1$)
- There are 8 bits of data to be checked and 16 bits of output check code.
- CRC calculation time: 1 AHB clock cycle (HCLK)
- The verification initial value can be configured, and the size end of the data to be verified can be configured.
- Support 8bit LRC check value generation
- Initial value for cyclic redundancy check (CRC) calculation is configurable

The following figure is the block diagram of CRC unit.

Figure 8-1 CRC Calculation Unit Block Diagram



8.3 CRC Function Description

8.3.1 CRC32 Usage Process

1) Configure the CRCEN bit in the RCC_AHBPCLKEN register to enable the CRC module clock;
2) Software writes a 1 to the RESET bit in the CRC_CRC32CTRL register to reset the CRC calculation unit, otherwise the initial value is the result of the previous operation;
3) Software writes to CRC_CRC32D to set the initial value for calculation;
4) Software writes to the CRC_CRC32DAT register to configure the data to be verified;
5) If further verification is needed, go back to step 4
6) Software reads from CRC_CRC32D to obtain the CRC calculation result;

8.3.2 CRC16 Usage Process

1) Configure the CRCEN bit in the RCC_AHBPCLKEN register to enable the CRC module clock;
2) Configure the CRC_CRC16CTRL register to determine the endianness of the data;
3) If it is necessary to clear the result of the previous CRC calculation, it can be done by setting CRC_CRC16CTRL.CLEAR to 1 or configuring CRC_CRC16D to 16'h00;
4) If a specific initial value is required for the CRC verification, configure the CRC_CRC16D register first, otherwise the initial value will be the result of the previous operation;
5) Software writes to the CRC_CRC16DAT register to configure the data to be verified;

6) If further verification is needed, go back to step 5;

7) Software reads from the CRC_CRC16D register to obtain the CRC calculation result.

8.3.3 LRC Usage Process

The LRC calculation process is similar to CRC, and it should be noted that CRC and LRC calculations are performed simultaneously. Users can choose to read either CRC or LRC based on their actual needs.

1) Configure the CRCEN bit in the RCC_AHBPCLKEN register to enable the CRC module clock;

2) If a specific initial value is required for LRC verification, configure the CRC_LRC register before the operation;

3) Software writes to the CRC_CRC16DAT register to configure the data to be verified;

4) If further verification is needed, go back to step 3;

5) Software reads from the CRC_LRC register to obtain the LRC calculation result.

8.4 CRC Registers

8.4.1 CRC Register Overview

Table 8-1 CRC Register Overview

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
000h	CRC_CRC32CTRL	Reserved																															RESET
	Reset Value																																0
004h	CRC_CRC32DAT	CRC32DAT[31:0]																															
	Reset Value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
008h	CRC_CRC32IDAT	Reserved																								CRC32IDAT[7:0]							
	Reset Value																									0	0	0	0	0	0	0	0
00Ch	CRC_CRC32D	CRC32D[31:0]																															
	Reset Value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
010h	CRC_CRC16CTRL	Reserved																													CLEAR	ENDIAN	Reserved
	Reset Value																														0	0	
014h	CRC_CRC16DAT	Reserved																								CRC16DAT [7:0]							
	Reset Value																									0	0	0	0	0	0	0	0
018h	CRC_CRC16D	Reserved																CRC16D[15:0]															
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
01Ch	CRC_LRC	Reserved																								LRCDAT [7:0]							

	Reset Value		0	0	0	0	0	0	0	0

8.4.2 CRC32 Control Register (CRC_CRC32CTRL)

Address offset: 0x00

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															RESET
															wo

Bit Field	Name	Description
31:1	Reserved	Reserved, the reset value must be maintained.
0	RESET	CRC32 reset control register It can reset CRC32 module and set data register to be 0xFFFF_FFFF. This reset can only write 1, and hardware will clear to 0 automatically.

8.4.3 CRC32 Data Register (CRC_CRC32DAT)

Address offset: 0x04

Reset value: 0xFFFF FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CRC32DAT[31:0]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CRC32DAT[31:0]															
rw															

Bit Field	Name	Description
31:0	CRC32DAT[31:0]	CRC32 Data register. The written data is the CRC value to be checked. The read data is the CRC calculation result. Only 32-bit operations are supported.

8.4.4 CRC32 Independent Data Register (CRC_CRC32IDAT)

Address offset: 0x08

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								CRC32IDAT[7:0]							
								rw							

Bit Field	Name	Description
31:8	Reserved	Reserved, the reset value must be maintained.
7:0	CRC32IDAT[7:0]	General 8 bits data register. It is for temporary stored 1-byte data. CRC_ CRC32CTRL.RESET reset signal will not impact this register.

Note: This register is not a part of CRC calculation and can be used to store any data.

8.4.5 CRC32 Check Data Register（CRC_CRC32D）

Address offset: 0x0C

Reset value: 0xFFFF FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CRC32D[31:0]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CRC32D[31:0]															
rw															

Bit Field	Name	Description
31:0	CRC32D[31:0]	CRC32 verification result 32-bit register.

8.4.6 CRC16 Control Register (CRC_CRC16CTRL)

Address offset: 0x10

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													CLEAR	ENDIAN	Reserved
													rw	rw	

Bit Field	Name	Description
31:3	Reserved	Reserved, the reset value must be maintained.
2	CLR	Clear CRC16 results. 0: Not clear. 1: Clear to default value 0x0000. Set this bit to 1 will only maintain 1 clock cycle, hardware will clear automatically. (Software read always 0).
1	ENDHL	Data to be verified start to calculate from MSB or LSB(configured endian). 0: From MSB to LSB 1: From LSB to MSB This bit is only for data to be verified.
0	Reserved	Reserved, the reset value must be maintained.

Note: 8-bits, 16-bits and 32-bits operations are supported.

8.4.7 CRC16 Data Register (CRC_CRC16DAT)

Address offset: 0x14

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								CRC16DAT[7:0]							
								rw							

Bit Field	Name	Description
31:8	Reserved	Reserved, the reset value must be maintained.
7:0	CRC16DAT[7:0]	Data to be verified.

Note: 8-bits, 16-bits and 32-bits operations are supported.

8.4.8 CRC16 Check Data Register (CRC_CRC16D)

Address offset: 0x18

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

Reserved

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CRC16D[15:0]															
rw															

Bit Field	Name	Description
31:16	Reserved	Reserved, the reset value must be maintained.
15:0	CRC16D[15:0]	16-bit value of cyclic redundancy result data.

Note: 8-bits, 16-bits and 32-bits operations are supported

8.4.9 LRC Result Register (CRC_LRC)

Address offset: 0x1C

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								LRCDAT[7:0]							
								rw							

Bit Field	Name	Description
31:8	Reserved	Reserved, the reset value must be maintained.
7:0	LRCDAT[7:0]	LRC check value register. 1) This register can be directly read and written to. Before use, software can configure the initial value. 2) Each time a value is written to the CRCDR register, it will be XORed with the LRC register, and the result will be fed back to the LRC. Software reads the LRC result, and before the next operation, the software should clear it to zero.

9 Super High-resolution Timer (SHRTIM)

9.1 Introduction

The supper high-resolution timer can generate up to 12 channels of highly precise digital signals, mainly used for driving power conversion systems such as switch-mode power supplies or lighting systems, but can also be used for applications with extremely high time resolution requirements.

This timer adopts a modular architecture that can generate independent waveforms or coupled waveforms. The waveform is determined by independent timing signals (using counters and comparator units) as well as various external events (such as analog or digital feedback and synchronous signals), thus enabling the generation of a wide range of control signals (PWM, phase shift, constant Ton, etc.), meeting the requirements of most conversion topologies.

For control and monitoring purposes, the timer also has timing measurement functions and is connected to built-in ADC and DAC converters. Additionally, the timer features a light-load management mode and can handle various fault mechanisms to achieve safe shut-down.

9.2 Main Features

- Multiple super high-resolution timing units
  - 125ps resolution, all outputs support full resolution, possibility to adjust duty cycle, frequency, and pulse width in triggered one-pulse mode
  - 6 16-bit timing units (each timing unit contains 1 independent counter and 5 compare units, compare unit 5 dedicated to ADC triggering)
  - 12 channels of output can be controlled by any timing unit, with up to 32 set/reset sources for each channel
  - Modular structure can meet the requirements of multiple independent converters equipped with 1 or 2 switches, as well as the requirements of a few large multi-switch topologies
- Up to 10 external events can be used for any timing unit
  - Programmable polarity and edge effectiveness
  - 10 events with a fast asynchronous mode
  - 10 events with a programmable digital filters
  - Spurious events filtering with blanking and windowing modes
  - All 10 external events fully mapped to any GPIO or any analog comparator
- Multiple channels connect to built-in analog peripherals
  - 10 trigger signals for ADC converters, ADC trigger signals can be fully mapped to any compare unit
  - 3 trigger signals for DAC converters
  - 7 comparators for analog signal conditioning

- Rich protection mechanisms
  - 6 fault inputs can be combined and associated with any timing unit
  - 6 fault inputs can be fully mapped to any analog comparator
  - Programmable polarity and edge effectiveness, digital filters
  - Resonant converters equipped with dedicated delay protection
- Multiple SHRTIM instances can be synchronized with external synchronization input/output
- Multi-functional output stage
  - Full resolution deadtime time insertion
  - Programmable output polarity
  - Chopping mode
- Burst mode controller, capable of handling light-load operations on multiple converters simultaneously, supporting 32-bit burst mode counting
- 8 interrupt vectors, each vector with up to 14 sources
- 7 DMA requests, with up to 14 sources, with a burst mode for multiple registers update

9.3 Function Description

9.3.1 Overview

SHRTIM can be divided into the following sub-entities:

- The master timer
- The timing units (timer A to timer F)
- The output stage
- The burst mode controller
- The External event and fault signal conditioning logic, shared by all timers
- The System interface

The master timer is based on a 16-bit up counter. It can set/reset any of the 12 outputs through 4 compare units and provide synchronous signals to 6 timer units. Its main purpose is to control the timer units with a unique clock source. A typical application example is interleaved buck converters, where the master timer manages the phase shift between multiple units. The compare unit 5 is dedicated to ADC triggering.

The timer units can work independently or coupled with other timers (including the master timer). Each timer can control two outputs. The output set/reset events can be triggered by the timer unit's compare register, events from the main timer, other timers, or external events.

The output stage serves various purposes:

- Adding deadtime when configuring 2 outputs in complementary PWM mode

- Adding carrier frequency to the modulation signal
- Managing fault events by asynchronously asserting the output to a predefined safe level

In light-load operation, the burst mode controller can control one or multiple timers. The burst length and period, as well as the output idle state, can be programmed.

The external event and fault signal conditioning logic include:

- Input selection MUX (e.g., selecting a digital input or an on-chip clock source for a specific external event channel)
- Programming of polarity and edge effectiveness
- Digital filtering

The system interface allows SHRTIM to interact with the rest of the MCU:

- Sending interrupt requests to the CPU
- DMA controller for automatic accesses to/from the memories, including a SHRTIM specific burst mode
- Triggering ADC and DAC converters

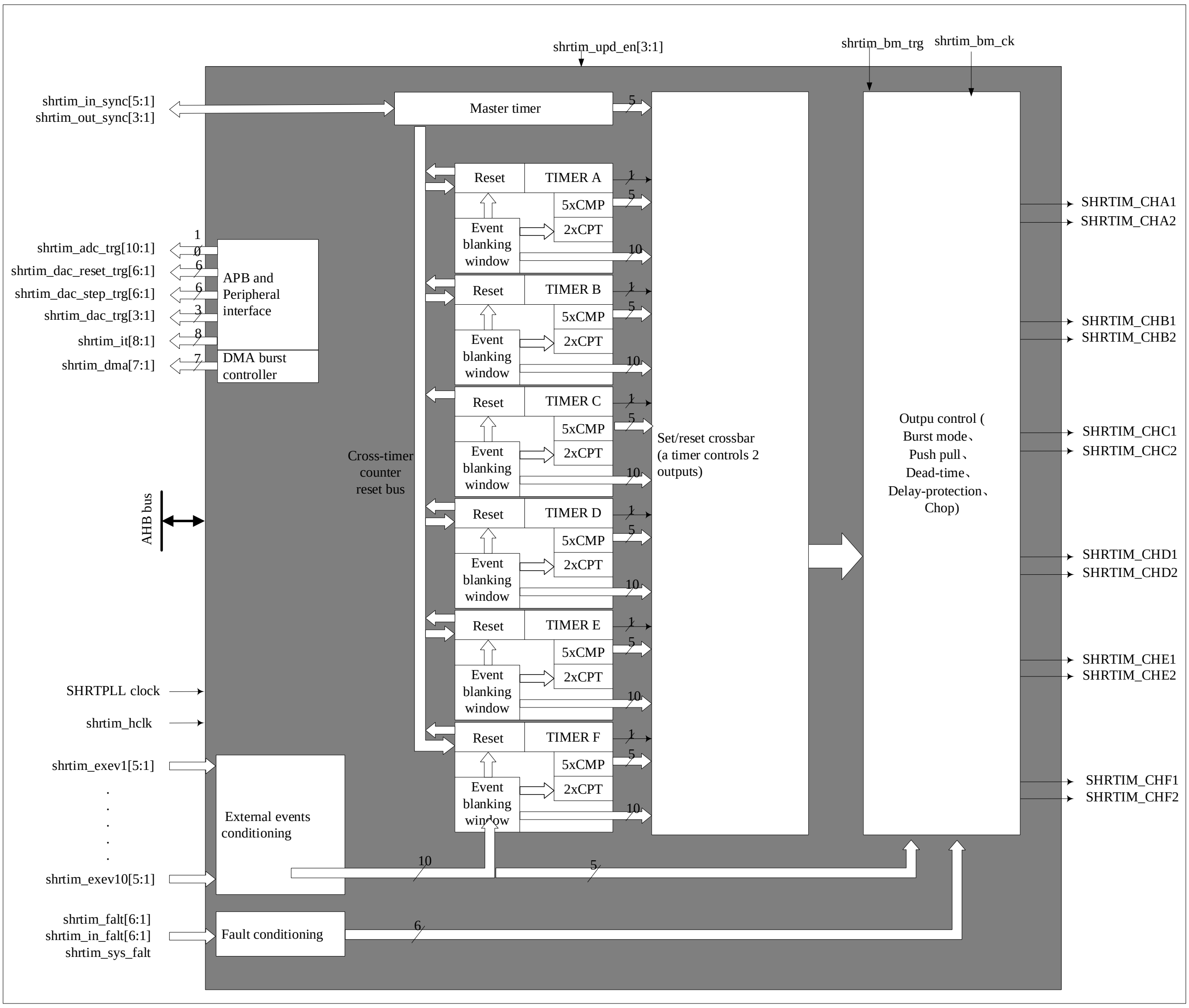
The SHRTIM registers are divided into 7 groups:

- Master timer registers
- Timer A to Timer F registers
- Common registers for functions shared by all timer units

Note: According to the document writing convention, references to the 6 timing units are uniformly represented by the letter "x" in text and registers (where "x" can be any value from A to F).

The block diagram of the timer is shown in the following figure.

Figure 9-1 SHRTIM Overview



9.3.2 SHRTIM Pins and Internal Signals

The following table summarizes the on-chip and off-chip SHRTIM inputs and outputs.

Table 9-1 SHRTIM Inputs/Outputs Summary

Signal Name	Signal Type	Description
SHRTIM_CHA1, SHRTIM_CHA2, SHRTIM_CHB1, SHRTIM_CHB2, SHRTIM_CHC1, SHRTIM_CHC2, SHRTIM_CHD1, SHRTIM_CHD2,	Outputs	Main SHRTIM timer outputs. They can insert deadtime when these outputs work in pairs (SHRTIM_CHx1 and SHRTIM_CHx2), or they can work independently.

SHRTIM_CHE1, SHRTIM_CHE2, SHRTIM_CHF1, SHRTIM_CHF2		
shrtim_in_falt1[4:1] shrtim_in_falt2[4:1] shrtim_in_falt3[4:1] shrtim_in_falt4[4:1] shrtim_in_falt5[4:1] shrtim_in_falt6[4:1]	Digital input	Fault inputs: When set to active, immediately disable SHRTIM outputs (12 on-chip inputs and 6 off-chip SHRTIM_FALTx inputs).
shrtim_sys_flt	Digital input	System faults gathering internal MCU fault events (Clock Security System, SRAM ECC error, SRAM parity error, Cortex^{TM}-M4F LOCKUP (HardFault), PVD output, FLASH ECC dual check error).
shrtim_in_sync[5:1]	Digital input	Synchronization inputs for synchronizing the entire SHRTIM with other internal or external timer resources: SHRTIM_in_sync1: atim1_trgo SHRTIM_in_sync2: atim2_trgo SHRTIM_in_sync3: atim3_trgo SHRTIM_in_sync4: SHRTIM_SCIN(IOM) SHRTIM_in_sync5: Reserved
shrtim_out_sync[2:1]	Digital output	This output is used for cascading or synchronizing multiple on-chip or off-chip SHRTIM instances: shrtim_out_sync1: the destination is off-chip SHRTIM or external peripherals (via SHRTIM_SCOUT output pin) shrtim_out_sync2: the destination is on-chip peripherals
shrtim_exev1[5:1]	Digital input	External events: all 10 events can be selected from 5 groups of sources, including on-chip sources (from other built-in peripherals: comparators, ADC analog watchdog, TIMx timer trigger output, CAN output) or off-chip sources (SHRTIM_EXEVx input pin)
shrtim_exev2[5:1]		
shrtim_exev3[5:1]		
shrtim_exev4[5:1]		
shrtim_exev5[5:1]		
shrtim_exev6[5:1]		
shrtim_exev7[5:1]		
shrtim_exev8[5:1]		
shrtim_exev9[5:1]		
shrtim_exev10[5:1]		
shrtim_upd_en[3:1]	Digital input	The SHRTIM register update enable input (on-chip interconnect) triggers the transfer operation from shadow registers to active registers.
shrtim_bm_trg	Digital input	Burst mode trigger event (on-chip interconnect)
shrtim_bm_ck[4:1]	Digital input	Burst mode clock (on-chip interconnect)
shrtim_adc_trg[10:1]	Digital output	ADC start of conversion trigger signal
shrtim_dac_trg[3:1]	Digital output	DAC conversion update trigger signal
shrtim_dac_reset_trg[6:1] shrtim_dac_step_trg[6:1]	Digital output	Dual channel DAC triggers

shrtim_it[8:1]	Digital output	Interrupt requests
shrtim_dma[7:1]	Digital output	DMA requests
shrtim_hclk	-	AHB clock
SHRTPLL clock	-	SHRTIM kernel clock（referred to as f_{SHRTIM}）

Table 9-2 External Event Mapping and Associated Features

SHRTIM external event input signal	Fast mode	Digital filter	Balanced idle A,B,C	Balanced idle D,E,F	EXEVxSRC[2:0]=0 (from GPIO pin)(1)	EXEVxSRC[2:0]=1(2)	EXEVxSRC[2:0]=2	EXEVxSRC[2:0]=3	EXEVxSRC[2:0]=4
shrtim_exev1[5:1]	Yes	Yes	-	-	SHRTIM_EXEV1	compx_out(1~7)	atim1_trgo	adc1_AWD1	N/A
shrtim_exev2[5:1]	Yes	Yes	-	-	SHRTIM_EXEV2	compx_out(1~7)	atim2_trgo	adc1_AWD2	N/A
shrtim_exev3[5:1]	Yes	Yes	-	-	SHRTIM_EXEV3	compx_out(1~7)	atim3_trgo	adc1_AWD3	N/A
shrtim_exev4[5:1]	Yes	Yes	-	-	SHRTIM_EXEV4	compx_out(1~7)	reserved	adc2_AWD1	N/A
shrtim_exev5[5:1]	Yes	Yes	-	-	SHRTIM_EXEV5	compx_out(1~7)	reserved	adc2_AWD2	N/A
shrtim_exev6[5:1]	Yes	Yes	Yes	-	SHRTIM_EXEV6	compx_out(1~7)	reserved	adc2_AWD3	N/A
shrtim_exev7[5:1]	Yes	Yes	Yes	-	SHRTIM_EXEV7	compx_out(1~7)	btim2_trgo	adc3_AWD2	N/A
shrtim_exev8[5:1]	Yes	Yes	-	Yes	SHRTIM_EXEV8	compx_out(1~7)	reserved	adc3_AWD3	N/A
shrtim_exev9[5:1]	Yes	Yes	-	Yes	SHRTIM_EXEV9	compx_out(1~7)	gtim8_trgo	adc4_AWD2	N/A
shrtim_exev10[5:1]	Yes	Yes	-	-	SHRTIM_EXEV10	compx_out(1~7)	gtim9_trgo	adc4_AWD3	N/A

1. Each EXEV can be mapped to any IO

2. compx_out can be mapped to any comparator

Table 9-3 Update Enable Inputs and Sources

SHRTIM update enable signal	SHRTIM update enable assignment
SHRTIM_upd_en1	gtim8_oc1
SHRTIM_upd_en2	gtim9_oc1
SHRTIM_upd_en3	gtim10_oc1

Table 9-4 Burst Mode Clock Sources

SHRTIM Burst mode trigger event/ clock signal	SHRTIM Burst mode trigger event/ clock signal ssignment
SHRTIM_bm_trg	btim1_trgo
SHRTIM_bm_ck1	gtim8_oc1
SHRTIM_bm_ck2	gtim9_oc1
SHRTIM_bm_ck3	gtim10_oc1

SHRTIM_bm_ck4	btim1_trgo

Table 9-5 Fault Inputs

SHRTIM Fault	SHRTIM External Input	On-chip source	External Input	On-chip source
channel	FALTxSRC[1:0] = 00	FALTxSRC[1:0] = 01(1)	FALTxSRC[1:0] =10	FALTxSRC[1:0] = 11
shrtim_falt1[5:1]	SHRTIM_FALT1	compx_out(1~7)	EXEV1_muxout	N/A
shrtim_falt2[4:1]	SHRTIM_FALT2	compx_out(1~7)	EXEV2_muxout	N/A
shrtim_falt3[4:1]	SHRTIM_FALT3	compx_out(1~7)	EXEV3_muxout	N/A
shrtim_falt4[4:1]	SHRTIM_FALT4	compx_out(1~7)	EXEV4_muxout	N/A
shrtim_falt5[4:1]	SHRTIM_FALT5	compx_out(1~7)	EXEV5_muxout	N/A
shrtim_falt6[4:1]	SHRTIM_FALT6	compx_out(1~7)	EXEV6_muxout	N/A

1. compx_out can be mapped to any comparator.

Table 9-6 SHRTIM DAC Triggers Connections

SHRTIM DAC triggers	DAC1/DAC2	DAC3/DAC4	DAC5/DAC6	DAC7/DAC8
shrtim_dac_reset_trg1, shrtim_dac_step_trg1	Yes	Yes	Yes	Yes
shrtim_dac_reset_trg2, shrtim_dac_step_trg2	Yes	Yes	Yes	Yes
shrtim_dac_reset_trg3, shrtim_dac_step_trg3	Yes	Yes	Yes	Yes
shrtim_dac_reset_trg4, shrtim_dac_step_trg4	Yes	Yes	Yes	Yes
shrtim_dac_reset_trg5, shrtim_dac_step_trg5	Yes	Yes	Yes	Yes
shrtim_dac_reset_trg6, shrtim_dac_step_trg6	Yes	Yes	Yes	Yes
sshrtim_dac_trg1	Yes	-	-	Yes
sshrtim_dac_trg2	-	Yes	-	-
sshrtim_dac_trg3	-	-	Yes	-

9.3.3 Clocks

SHRTIM must be clocked by SHRTPLL to achieve full resolution. All clocks in SHRTIM are generated by this reference clock.

9.3.3.1 Definition of Terms

f_{SHRTIM}: SHRTIM main clock (equivalent to SHRTPLL clock), all subsequent clocks are generated by this clock source and synchronized with this clock source.

f_{HRCK}: SHRTIM equivalent clock. Given that the f_{SHRTIM} clock period is divided by 32, its equivalent frequency is

250MHz x 32 = 8 GHz.

f_{DTG}: dead-time generator clock. For convenience, only the t_{DTG} period (t_{DTG} = 1/f_{DTG}) is used in this document.

f_{CHPFRQ}: chopper stage clock source.

f_{1STPW}: clock source defining the initial pulse length in chopper mode. For convenience, only the t_{1STPW} period (t_{1STPW} = 1/f_{1STPW}) is used in this document.

f_{BRST}: burst mode controller counter clock.

$f_{SAMPLING}$: clock for sampling fault or external event inputs.

f_{FALTS}: clock derived from f_{SHRTIM}, used as the source for $f_{SAMPLING}$ to filter fault events.

f_{EXEVS}: clock derived from f_{SHRTIM}, used as the source for $f_{SAMPLING}$ to filter external events.

F_{hclk} (shrtim_hclk): AHB bus clock for register read/write accesses.

9.3.3.2 Timer Clock and Prescaler

Each timer in SHRTIM has an independent clock prescaler for users to adjust the timer resolution. See the table below.

Table 9-7 Timer Resolution and Minimum PWM Frequency at f_{SHRTIM} = 250MHz

CKPSC[2:0]	Prescaling ratio	f_{HRCK} equivalent frequency	Resolution	Min PWM frequency
0	1	250 x 32MHz = 8 GHz	125 ps	122.1 kHz
1	2	250 x 16MHz = 4 GHz	250 ps	61.0 kHz
10	4	250 x 8MHz = 2 GHz	500 ps	30.5 kHz
11	8	250 x 4MHz = 1 GHz	1 ns	15.3 kHz
100	16	250 x 2MHz = 500 MHz	2 ns	7.63 kHz
101	32	250 MHz	4 ns	3.81 kHz
110	64	250/2MHz = 125 MHz	8 ns	1.91 kHz
111	128	250 /4MHz = 62.5 MHz	16 ns	0.95 kHz

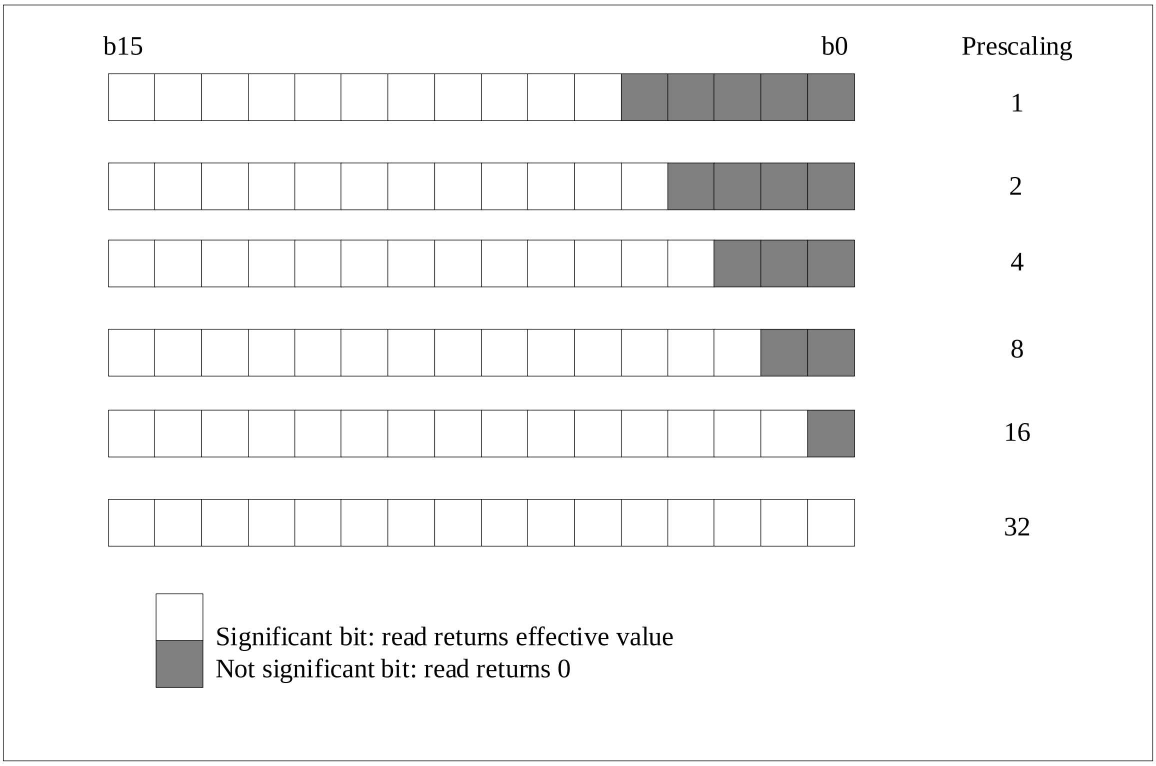
The high -resolution can be used for edge positioning, PWM period adjustment, and pulse duration for external triggers. The following functions are not supported with high-resolution:

- Timer counter read/write accesses
- Capture unit
- CMP5 triggering ADC

For clock prescaler ratios less than 32 (CKPSC[2:0] < 5), the least significant bits of the counter and capture registers are not significant. These least significant bits cannot be written (only for counter registers) and return 0 when read.

For example, if CKPSC[2:0] = 2 (prescaling by 4), writing 0xFFFF to the counter register results in an actual effective value of 0xFFF8. Conversely, any counter value between 0xFFFF and 0xFFF8 will be displayed as 0xFFF8 when read.

Figure9-2 Counter and Capture Register Format vs Clock Prescaling Factor


b15
b0
Prescaling
1
2
4
8
16
32
Significant bit: read returns effective value
Not significant bit: read returns 0

9.3.3.3 Initialization

During start-up, it is essential to initialize the prescaler fields before writing to the compare and period registers. Once the timer is enabled (MCNTEN or TxCNTEN bit in the SHRTIM_MCTRL register is set to 1), the prescaler cannot be modified.

If multiple timers are enabled, the prescaler will synchronize with the prescaler of the timer first started.

Warning: The master timer and TIMA..F timers can only use different prescaler ratios if the counter and output behavior does not depend on the information and signals from other timers. If an event is transmitted from one timer unit (or the main timer) to another timer unit, it is essential to configure the same prescaler ratio in these timers for the following events: output set/reset events, counter reset events, update events, external event filtering, or capture triggers. Having unequal prescaler ratios can lead to unpredictable results.

9.3.3.4 Deadtime Generator Clock

The deadtime prescaler is provided by $(f_{SHRTIM} \text{ x } 8) / 2^{(DTPSC[2:0])}$, programmed through the DTPSC[2:0] bits in the SHRTIM_TxDT register.

When f_{SHRTIM} = 250 MHz, the range of t_{DTG} is from 500 ps to 64 ns.

9.3.3.5 Chopper Stage Clock

The chopper stage clock source f_{CHPFRQ} is generated by f_{SHRTIM}, using a prescaler factor from 16 to 256, resulting in 976.56 KHz <= f_{CHPFRQ} <= 15.625 MHz (when f_{SHRTIM} = 250MHz).

t_{1STPW} is the length of the initial pulse in chopper mode, programmed through the STARTPW[3:0] bits in the SHRTIM_TxCHOP register, with the calculation formula as follows:

t_{1STPW} = (STARTPW[3:0]+1) x 16 x $t_{SHRTIM.}$

It uses f_{SHRTIM} / 16 as the clock source for calculations (15.625 MHz for f_{SHRTIM} = 250 MHz).

9.3.3.6 Burst Mode Prescaler

The burst mode controller counter clock f_{BRST} can be provided by multiple clock sources, with one clock source generated by f_{SHRTIM}. In this case, the range of f_{BRST} is from f_{SHRTIM} to f_{SHRTIM} / 32768 (7.629KHz for f_{SHRTIM} = 250 MHz).

9.3.3.7 Fault Input Sampling Clock

The time constant of the fault input noise rejection filter is defined by $f_{SAMPLING}$ and can be either f_{SHRTIM} or f_{FALTS}. f_{FALTS} is generated by f_{SHRTIM}, with a range of 250 MHz to 31.25 MHz (for f_{SHRTIM} = 250 MHz).

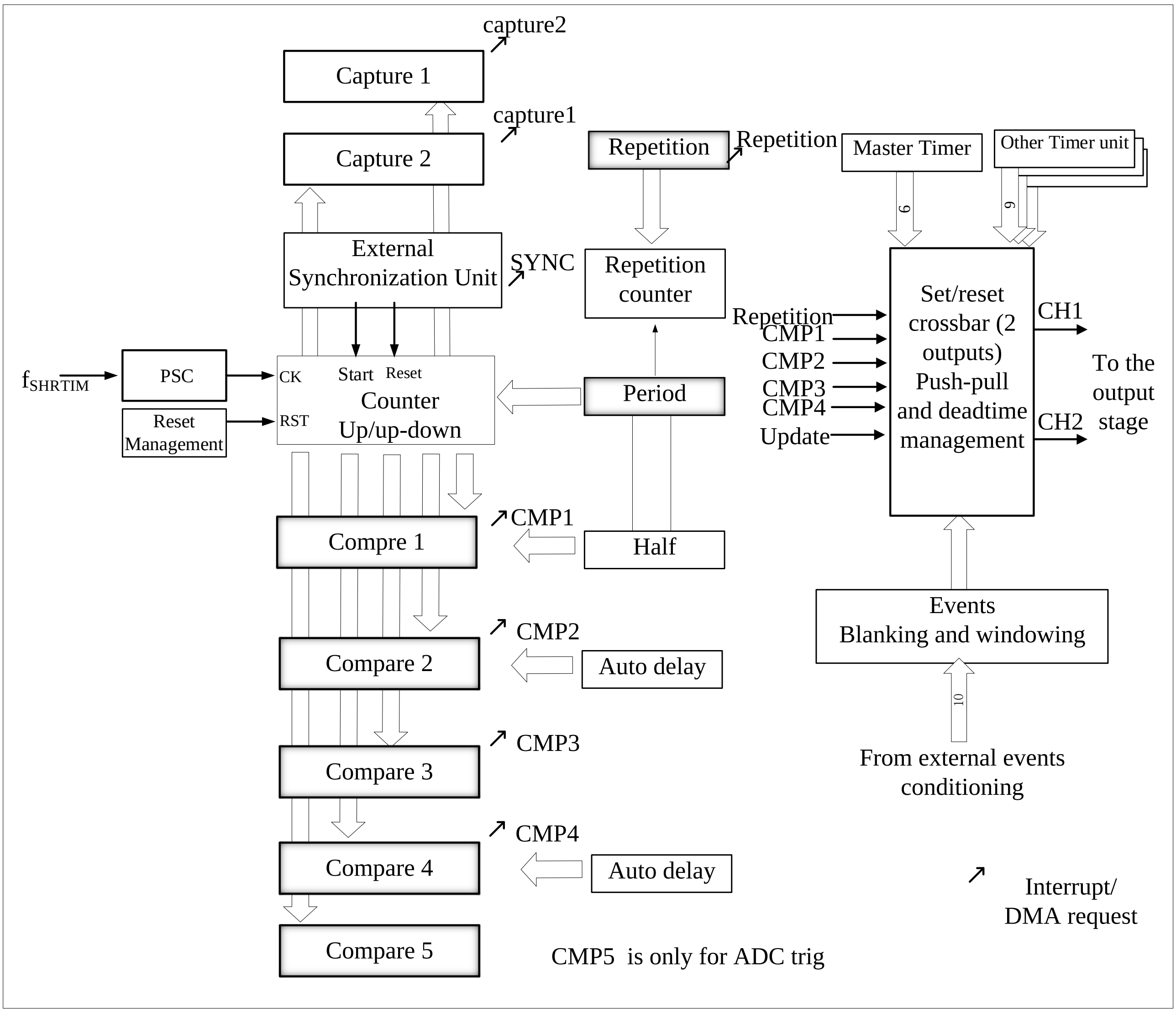
9.3.3.8 External Event Input Sampling Clock

The time constant of the external event input noise rejection filter is defined by $f_{SAMPLING}$ and can be either f_{SHRTIM} or f_{EXEVS}. f_{EXEVS} is generated by f_{SHRTIM}, with a range of 250 MHz to 31.25MHz (for f_{SHRTIM} = 250 MHz).

9.3.4 TIM A..F Timer Units

The SHRTIM embeds 6 identical timer units, each consisting of a 16-bit up counter with automatic reload mechanism to define the counting period, 5 comparison units, and 2 capture units, as shown in Figure9-3. Each unit contains all control functions for 2 outputs, allowing it to operate as an independent timer.

Figure9-3 Timer A…F Overview



The period and compare values must be within the upper and lower limits associated with high resolution, with specific values listed in Table9-8:

- The minimum value must be greater than or equal to 3 cycles of the fSHRTIM clock. The value 0x0000 can only be written to CMP1 and CMP3 registers to skip PWM pulses. For more information, refer to Section 9.3.4.8 Null duty cycle exception case .
- The maximum value must be less than or equal to 0xFFFF minus 1 cycle of the fSHRTIM clock.

Table9-8 Period and Compare Registers Min and Max Values

CKPSC[2:0] value	Min[1]	Max
0	0x0060	0xFFDF
1	0x0030	0xFFEF
2	0x0018	0xFFF7
3	0x000C	0xFFFB

4	0x0006	0xFFFD
≥ 5	0x0003	0xFFFD

Note: If the compare value is greater than the period register value, a compare match event will not be generated.

Unlike compare units 1 to 4, compare unit 5 is dedicated to ADC triggering and does not have the functionality of CMP1 to CMP4. It only supports the following features:

1. The CMP5 has regular resolution and does not have high-resolution features.
2. When CMP5 matches, it will generate the status flag of CMP5, which can also clear the status flag of CMP5. However, the CMP5 event will not be linked to NVIC, so no interrupt will be generated.
3. The CMP5 event can be used to trigger the ADC.

Counter operating mode

Timers A..F can operate in continuous (free-run) mode or single-shot mode, where counting starts from a reset event trigger. The operating mode is set by the CONT bit in the SHRTIM_TxCTRL control register. The additional RTG bit can be used to select whether the single-shot operation is retriggerable or non-retriggerable. Further details on the operating modes are provided and summarized as follows.

Table9-9 Timer Operating Modes

CONT	RTG	Operating Mode	Start/Stop Conditions Clocking and Event Generation
0	0	Single-shot Non-retriggerable	Setting TxCNTEN to 1 will enable the timer but will not start the counter. The first reset event will trigger the counter to start counting, but the counter will ignore any subsequent reset events until it reaches the period value. Subsequently, a period event will occur, and the counter will stop counting. After a reset event occurs, the counter will restart counting from 0x0000.
0	1	Single-shot Retriggerable	Setting TxCNTEN to 1 will enable the timer but will not start the counter. If the counter is not counting, a reset event will start the counter, otherwise, it will reset the counter. When the counter reaches the period value, a period event will occur, and the counter will stop counting. After a reset event occurs, the counter will restart counting from 0x0000.
1	X	Continuous mode	Setting TxCNTEN to 1 will enable the timer and start the counter. When the counter reaches the period value, it will roll-over to 0x0000 and restart counting. The counter can be reset at any time.

The TxCNTEN bit can be cleared at any time to disable the timer and stop the counting.

Figure9-4 Continuous Timer Operation Mode

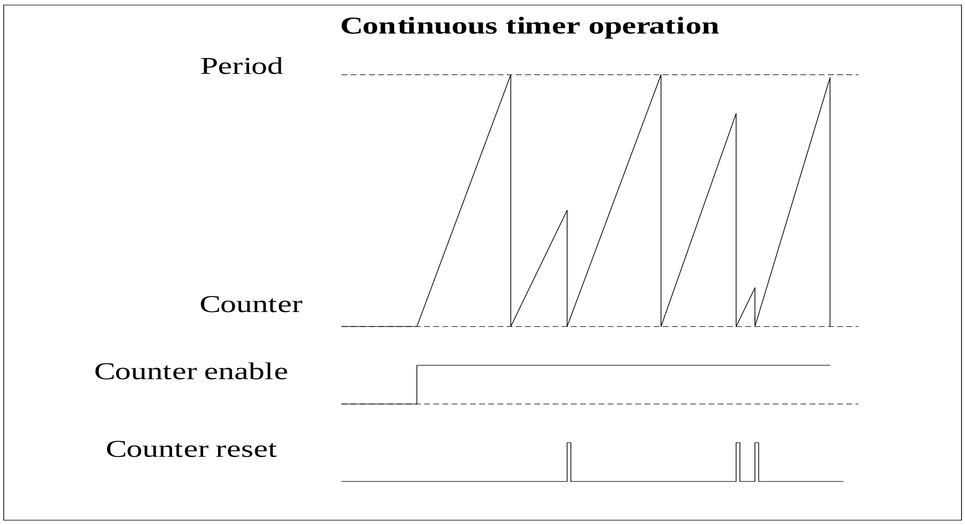
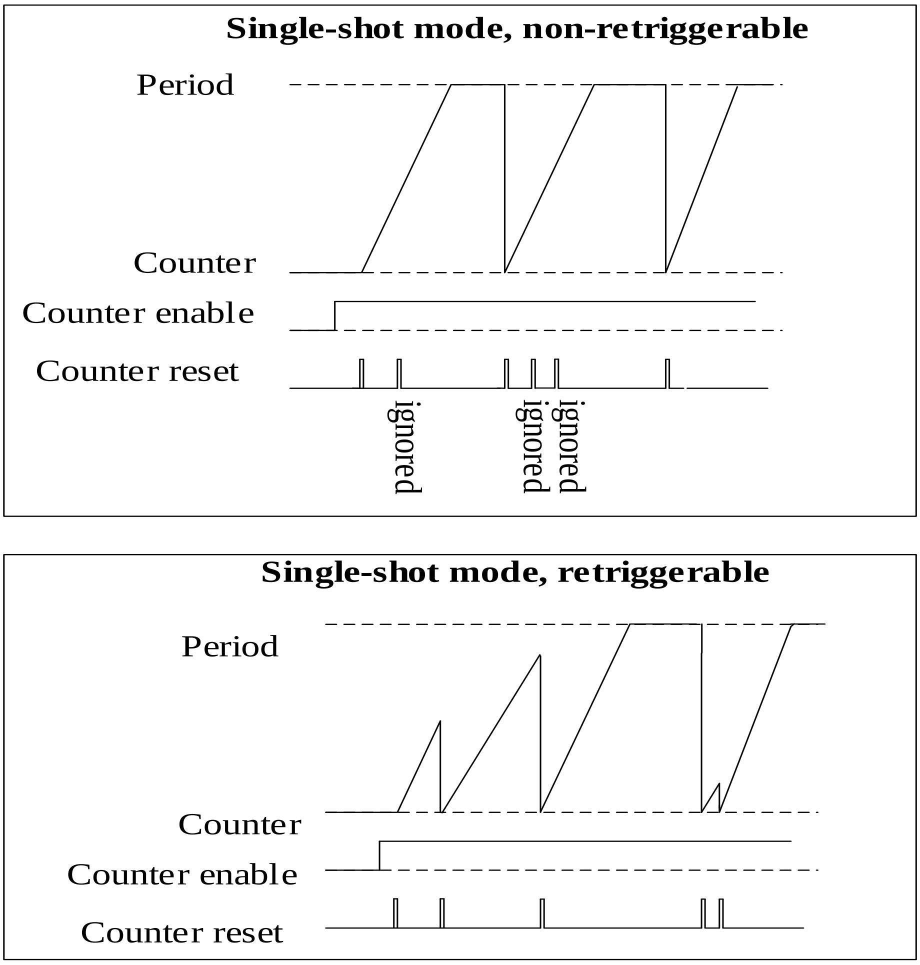


Figure9-5 Single-Shot Timer Operation Mode



9.3.4.1 Roll-over event

In continuous mode, if the counter resets to 0 after reaching the period value set in the SHRTIM_TxPRD register, a counter roll-over event will be generated. In single-shot mode, the rollover event is generated when a reset event happens following the counter counts to the period value.

This event is used for various purposes in the SHRTIM:

- To set/reset the outputs
- To trigger the register content update (transfer from preload to active)
- To trigger an IRQ or a DMA request
- To serve as a burst mode clock source or burst start trigger signal
- To serve as an ADC trigger signal
- To decrement a repetition counter

If the initial counter value is greater than the period value at timer start, or a new period is set while the counter has already exceeded that value, the counter will not reset: it will overflow when reaching the maximum period value, and the repetition counter will not decrement.

9.3.4.2 Timer reset

The reset of the timer unit counter can be triggered by up to 30 events, which can be selected simultaneously in the SHRTIM_TACNTRST register. The specific reset sources include the following:

- Timer unit: compare 2, compare 4, and update (3 events)
- Master timer: reset and compare 1..4 (5 events)
- External events EXEV1..10（10 events）
- All other timer units (e.g. for Timer A, it would be Timer B..F): compare 1, 2, and 4 (12 events)

Multiple events can be selected simultaneously to handle multiple reset sources. In this case, an OR operation will be performed on multiple reset requests. If two counter reset events are generated within the same f_{SHRTIM} clock cycle, the last timer reset event will be considered.

Additionally, the TxRSTRO bit in the SHRTIM_CTRL2 register can be used to perform a software reset on the counter. These control bits are grouped into one register, allowing multiple counters to be reset simultaneously. The reset requests will only be considered when the relevant counter has been enabled (TxCNTEN set to 1).

Note: When multiple reset event happened, the phase of the last reset event will be taken into account to adjust crossbar set/reset event.

Note: When using update event to reset timer unit itself, the update event is low resolution (no phase).

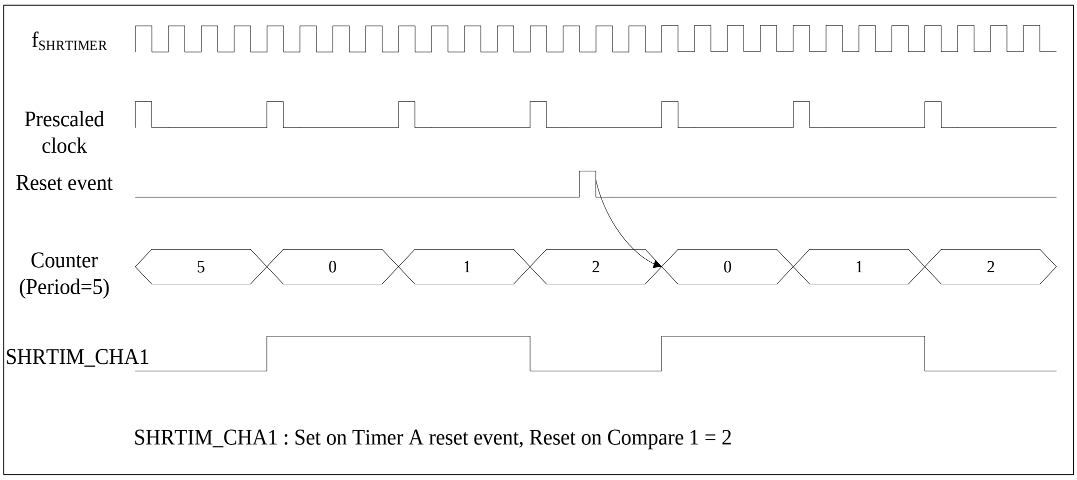
Note: If the reset event occurs within the same f_{SHRTIM} clock cycle as the period event, the period is postponed to a programmed reset event (since both events are causing a counter roll-over) whatever reset event phase is smaller or bigger than period phase. This applies only when the high resoulution is active (CKPSC[2:0] < 5).

Note: The reset event mentioned here are the valid reset event after adjustment with PRD phase or reset event phase, it is not the original register value configed by software.

If the clock prescaler ratio of the f_{SHRTIM} clock is greater than 32, the counter reset event will be delayed until the next valid edge of the prescaler clock. This ensures that the waveform generated at the output changes synchronized to the reset event (usually a constant Ton time converter) without jitter.

The following diagram illustrates the reset handling with a clock prescaler ratio of 4 (f_{SHRTIM} divided by 4).

Figure9-6 Timer Reset Resynchronization (Prescaling Ratio above 32)



9.3.4.3 Repetition counter

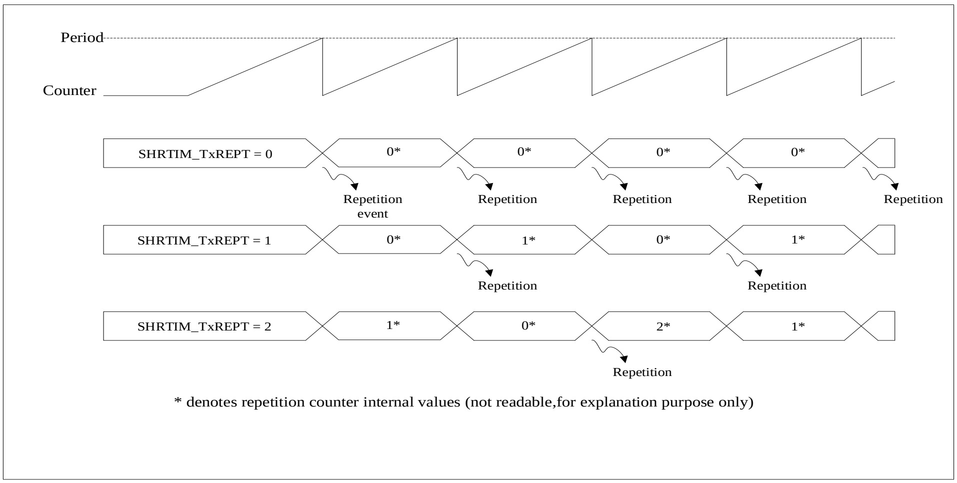
Software typically generates an interrupt when the period value is reached, allowing the longest possible time for processing before the next cycle begins. The primary purpose of the repetition counter is to adjust the periodic interrupt rate and off-load the CPU by separating the switching frequency from the interrupt frequency.

The timer unit includes a repetition counter. This counter cannot be read, but can only be programmed using the automatic reload value in the SHRTIM_TxREPT register.

After the timer is enabled (TxCNTEN set to 1), the repetition counter is initialized with the content of the SHRTIM_TxREPT register. Once the timer is enabled, the repetition counter is decreased each time the counter is cleared due to a reset event or counter roll-over. When the repetition counter reaches 0, a REPT interrupt or DMA request is generated (if enabled, using the REPTIEN and REPTDEN bits in the SHRTIM_TxIDEN register).

If the SHRTIM_TxREPT register is set to 0, an interrupt will occur every period. If the value is greater than 0, an REPTITF interrupt will occur after (SHRTIM_TxREPT + 1) periods. Figure9-7 illustrates the operation of the repetitive counter with different values in continuous mode.

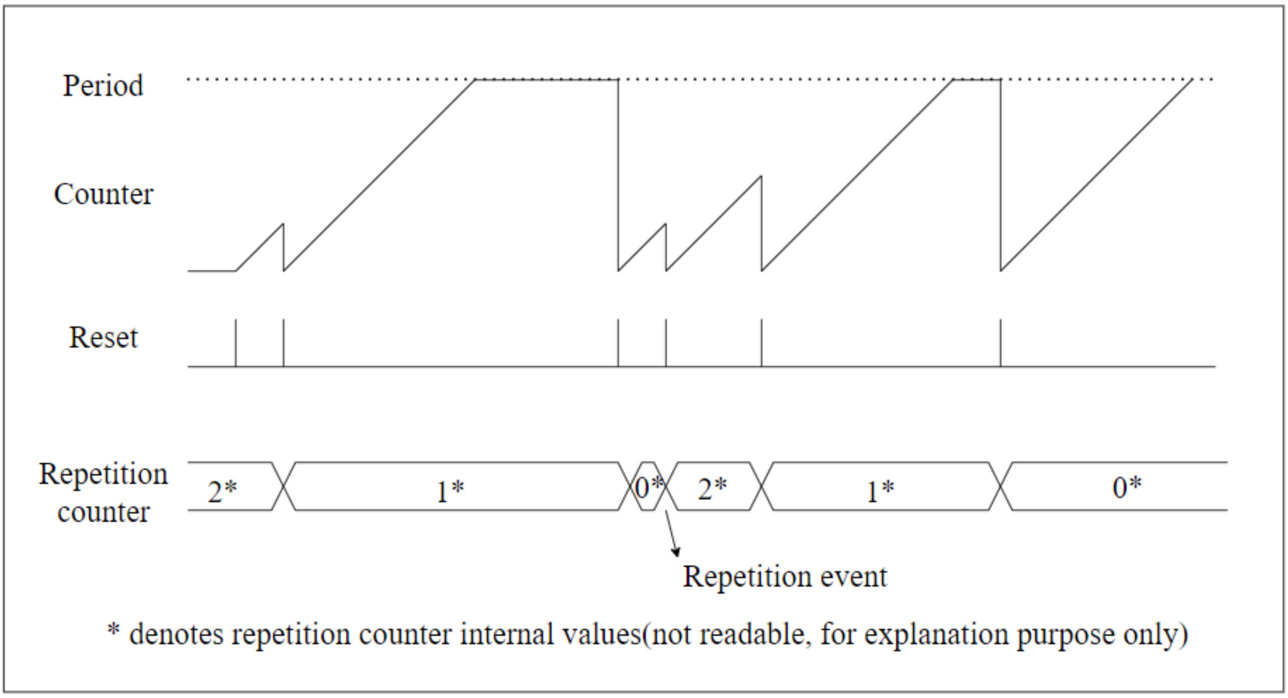
Figure9-7 Repetition Rate versus SHRTIM_TxREPT Content in Continuous Mode



* denotes repetition counter internal values (not readable,for explanation purpose only)

Whether in continuous or single-shot mode, if the counter is reset before reaching the period value (variable frequency operation), the repetition counter can also be used (as shown in Figure9-8). The reset will cause the repetition counter to decrement on the first start after the counter is enabled (TxCNTEN set to 1).

Figure9-8 Repetition Counter Behavior in Single-Shot Mode



* denotes repetition counter internal values(not readable, for explanation purpose only)

A reset or start event from the shrtim_in_sync[3:1] source will cause the repetition counter to be decremented as any other reset event. However, in single-shot mode started by SYNCIN (SYNCSTRTx set to 1 in the SHRTIM_TxCTRL register), the repetition counter will only decrement on the first reset event after the period. Any subsequent reset events will not change the value of the repetition counter until the counter is restarted by a new request from the shrtim_in_sync[3:1] input.

9.3.4.4 Set/reset crossbar

A "set" event corresponds to a transition to the active state of the output, while a "reset" event corresponds to a transition to the inactive state of the output.

The waveform polarity is defined at the output stage to accommodate external components with positive or negative logic: for positive polarity (POLx = 0), the active level corresponds to logic level 1; for negative polarity (POLx = 1), the active level corresponds to logic level 0.

Each timer unit controls the set/reset crossbar for two outputs. These two outputs can be set, reset, or toggled by up to 32 events, which can be selected from the following sources:

- Timer unit: period, Compare 1..4, register update (6 events)
- Master timer: period, compare 1..4, SHRTIM synchronization（6 events）
- All other timer units (e.g. for Timer A, it would be Timer B..F): TIMEV1..9 (9 events as described in Table9-10)
- External events EXEV1..10（10 events）
- A software forced (1 event)

Note: In up/down mode (UPDOWNM bit set to 1), the counter period event is defined based on the setting of the OUTPUTROM[1:0] bits.

The event sources are ORed together, allowing multiple events to be selected simultaneously.

Each output is controlled by two 32-bit registers, one for the set code (SHRTIM_TxSETy) and the other for the reset code (SHRTIM_TxRSTy), where x represents timer units A..F and y represents output 1 or 2 (e.g. SHRTIM_TASET1, SHRTIM_TCRST2...).

If the same event is selected for both set and reset, the output state will toggle. The number of state transitions for each t_{SHRTIM} period cannot exceed one: if there are two consecutive toggle events within the same period, please refer to Section “Set/Reset Event Priority and Narrow Pulse Management” for set/reset priority.

The set and reset requests will only be considered when the counter is enabled (TxCNTEN set to 1), except for cases where software forces a preset output at timer start-up.

Table9-10 summarizes the events from other timing units can be used to set and reset outputs. The numbers correspond to the timer events (such as TIMEVNTx) listed in the register, empty locations indicate unavailable events.

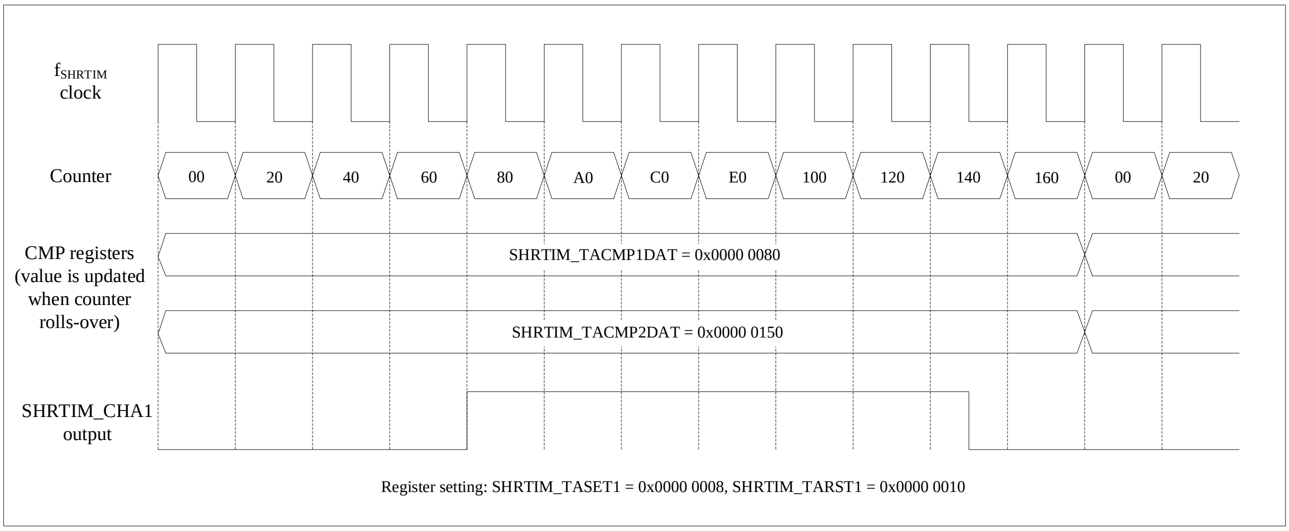
For example, Timer A output can be set or reset by the following events: Timer B compare 1, 2, and 4, Timer C compare 2 and 3... and Timer E compare 3 will be listed as TIMEV8 in SHRTIM_TASET1.

Table9-10 Events Mapping across Timer A to F

Source		TIMA				TIMB				TIMC				TIMD				TIME				TIMF			
		CMP1	CMP2	CMP3	CMP4	CMP1	CMP2	CMP3	CMP4	CMP1	CMP2	CMP3	CMP4	CMP1	CMP2	CMP3	CMP4	CMP1	CMP2	CMP3	CMP4	CMP1	CMP2	CMP3	CMP4
Destination	TA	-	-	-	-	1	2	-	-	-	3	4	-	5	6	-	-	-	-	7	8	-	-	-	9
	TB	1	2	-	-	-	-	-	-	-	-	3	4	-	-	5	6	7	8	-	-	-	-	9	-
	TC	-	1	2	-	-	3	4	-	-	-	-	-	-	5	-	6	-	-	7	8	-	9	-	-
	TD	1	-	-	2	-	3	-	4	-	-	-	5	-	-	-	-	6	-	-	7	8	-	9	-
	TE	-	-	-	1	-	-	2	3	4	5	-	-	6	7	-	-	-	-	-	-	-	-	8	9
	TF	-	-	1	-	2	-	-	3	4	-	-	5	-	-	6	7	-	8	9	-	-	-	-	-

The following diagram shows how PWM signals are generated through two compare events.

Figure9-9 Compare Events Action on outputs: Set on Compare 1, Reset on Compare 2


fSHRTIM clock
Counter
00
20
40
60
80
A0
C0
E0
100
120
140
160
00
20
CMP registers (value is updated when counter rolls-over)
SHRTIM_TACMP1DAT = 0x0000 0080
SHRTIM_TACMP2DAT = 0x0000 0150
SHRTIM_CHA1 output
Register setting: SHRTIM_TASET1 = 0x0000 0008, SHRTIM_TARST1 = 0x0000 0010

9.3.4.5 Set/reset on update events

The set/reset events that occur on update are executed in low resolution. When CKPSC[2:0] is less than 5, the high-resolution is set to high-resolution part of the period event. If there is counter reset event, the high-resolution is adjusted by counter reset event, otherwise , it is still adjusted by period event.

9.3.4.6 Half mode

This mode is used to generate a square wave signal with a fixed 50% duty cycle and variable frequency (commonly used in converters using resonant topologies), allowing the duty cycle to be automatically forced to half the period value when setting a new period.

To enable this mode, write 1 to the HLF bit in the SHRTIM_TxCTRL register. After writing a value to the SHRTIM_TxPRD register, the compare 1 value is automatically updated to SHRTIM_TxPRD/2.

The output for generating the square wave must be programmed to have one transition on CMP1 event and one transition on period event as follows:

- SHRTIM_TxSETy = 0x0000 0008，SHRTIM_TxRSTy = 0x0000 0004, or
- SHRTIM_TxSETy = 0x0000 0004，SHRTIM_TxRSTy = 0x0000 0008

The HALF mode will override the content of the SHRTIM_TxCMP1DAT register. Accessing the SHRTIM_TxPRD register will only update the compare 1 internal register. The accessible SHRTIM_TxCMP1DAT register will not be updated to the value of SHRTIM_TxPRD/2.

When preload is enabled (PLEN = 1, MUPDDIS, TxUPDDIS), the compare 1 active register will be refreshed on an update event. If preload is disabled (PLEN = 0), the compare 1 active register will update immediately after writing a value to SHRTIM_TxPRD.

When the HALF mode is enabled, the period must be greater than or equal to 6 cycles of the f_{SHRTIM} clock (0xC0 if CKPSC[2:0]=0, 0x60 if CKPSC[2:0]=1, 0x30 if CKPSC[2:0]=2, and so on).

9.3.4.7 Interleaved mode

This mode complements the HALF mode and helps in implementing interleaved topologies.

It allows for automatic recalculation of the contents of the compare registers when updating the SHRTIM_TxPRD value.

The selection is done using the HLF bit and ILV[1:0] bits in the SHRTIM_MCTRL and SHRTIM_TxCTRL registers, as shown in the table below.

Table9-11 Interleaved Mode Selection

HLF	ILV [1:0] Bits	Mode
0	00	Disabled
0	01	Triple interleaved(120°)
0	10	Quad interleaved(90°)
0	11	Reserved
1	xx	Dual interleaved(180°)

The table below provides compare values for the three available modes. The contents of the compare registers will be overridden. The corresponding compare events can be used to trigger output set/reset or reset a slave timer.

Table9-12 Compare 1..3 Values in Interleaved Mode

Mode	Dual Interleaved	Triple Interleaved	Quad Interleaved
CMP1 value	TxPRD/2	TxPRD/3	TxPRD/4
CMP2 value	Not affected	2 x (TxPRD/3)	TxPRD/2
CMP3 value	Not affected	Not affected	3 x (TxPRD/4)

Note: In the HALF mode and INTERLEAVED mode, the compare registers are controlled by hardware, and writing to them has no effect. However, the written values are stored in the preload registers and take effect on the update event after exiting these modes.

Note: Triple and quadruple interleaved modes cannot be used simultaneously with other modes that use CMP2 (such as dual DAC trigger and trigger half mode, auto-delayed mode).

9.3.4.8 Null duty cycle exception case

For pulses narrower than 3 tSHRTIM periods, high-resolution behavior is not supported (refer to Section 9.3.7 Set/Reset Event Priority and Narrow Pulse Management), and any value strictly less than 3 periods of the f_{SHRTIM} clock in the SHRTIM_TxCMPyDAT register is prohibited (i.e. 0x60 if CKPSC[2:0]=0, 0x30 if CKPSC[2:0]=1, 0x18 if CKPSC[2:0]=2,…) (refer to Section 9.4.2.8 SHRTIM Timer x Compare 1 Register（SHRTIM_TxCMP1DAT).

However, it is possible to skip the output pulse and achieve a 0% duty cycle by simply writing zero values in the following two registers: SHRTIM_TxCMP1DAT and SHRTIM_TxCMP3DAT, only under the following conditions:

- Output set event generated by the period event
- Output reset event generated by the compare 1 (or compare 3) event
- The compare 1 (or compare 3) event is active within the timer unit itself, and not used for other timing units

For any other use case, set and reset events can be achieved by programming the same compare values with more than 3 periods of the f_{SHRTIM} clock. In this case, the output is forced to reset (following Section 9.3.7 Set/Reset Event

Priority and Narrow Pulse Management)

9.3.4.9 Swap mode

This mode allows to swap the two outputs with a single bit access: the output 1 signal is connected to the output 2 pin and the output 2 signal is connected to output 1 pin. The output swap is triggered with the SWAPx bits in the SHRTIM_CTRL2 register and is effective on the next update event.

The outputs are swapped prior to the set/reset crossbar unit, as following:

- if SWAPx = 0, SHRTIM_TxSET1 and SHRTIM_TxRST1 are coding for the output 1, SHRTIM_TxSET2 and SHRTIM_TxRST2 are coding for the output 2.
- if SWAPx = 1, SHRTIM_TxSET1 and SHRTIM_TxRST1 are coding for the output 2, SHRTIM_TxSET2 and SHRTIM_TxRST2 are coding for the output 1.

The swap mode is only affecting the preload register, and not the active registers.

Note: The preload mode must be enabled when using the swap mode.

Consequently, it does not modify the auxiliary outputs in parallel with the regular outputs going to the output stage. They provide the following internal status, events and signals:

- O1BCKUP, O2BCKUP, SETyITF and RSTyITF status flags, together with the corresponding interrupts and DMA requests
- Capture triggers upon output set/reset (TA2, TB2, TC2, TD2, TE2, TF2)
- External event filters generated with a Tx2 output copy

For instance the SET1ITF flag is related to the output 1 when SWAP = 0 and is related to the output 2 when SWAPx = 1.

Similarly, the swap mode does not change the attribution of control bits in the SHRTIM_TxOUT register (DIDLx, CHPx, FALTx[1:0], IDLESx, POLx bits). For instance, the POL1 bit controls the output 1 polarity whatever the SWAP bit value.

Note: The SWAPx bits are ignored in push-pull mode (PP = 1 in the SHRTIM_TxCTRL register).

9.3.4.10 Capture

The timing unit is able to capture the counter value triggered by internal and external events. The purposes of capturing are:

- Measure the arrival time of events or occurrence intervals.
- Update the compare 2 and compare 4 values in auto-delay mode (refer to Section 9.3.4.11 Auto-delayed mode).

The capturing is done with f_{SHRTIM} resolution: for clock prescaler ratios less than 32 (CKPSC[2:0] < 5), the least significant bits of the register are not significant (read as 0).

The timer contains 2 capture registers: SHRTIM_TxCPT1 and SHRTIM_TxCPT2. The capture trigger events are programmed in the SHRTIM_TxCPT1CTRL and SHRTIM_TxCPT2CTRL registers.

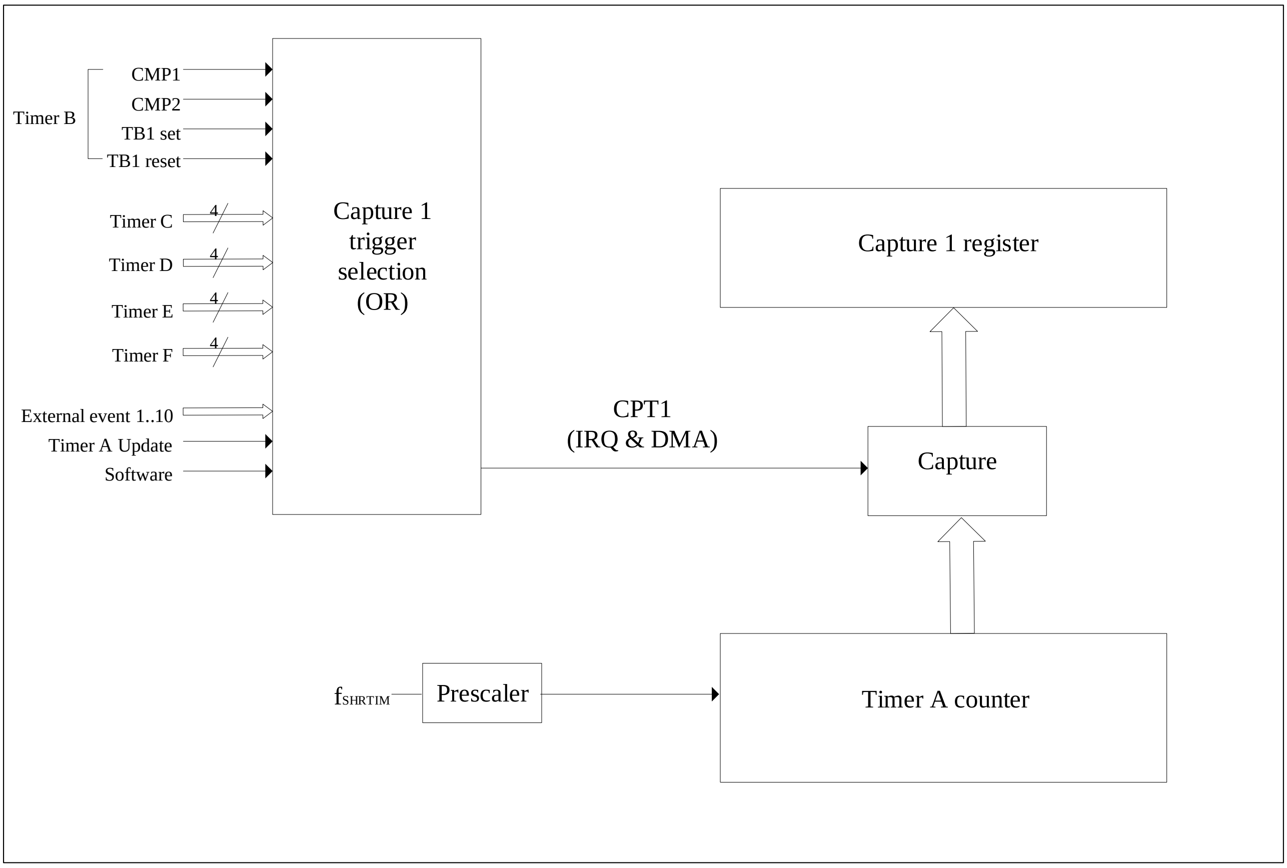
The capture of the timing unit counter can be triggered by up to 32 events, which can be selected simultaneously in the SHRTIM_TxCPT1CTRL and SHRTIM_TxCPT2CTRL registers, including the following trigger sources:

- External events, EXEV1..10（10 events）
- All other timing units (for example, for Timer A, it would be Timer B..F): compare 1, 2, and output 1 set/reset events (16 events)
- Timing unit: update event (1 event)
- A software capture event (1 event)

Multiple events can be selected simultaneously to handle multiple capture trigger signals. In this case, an OR operation is performed on multiple concurrent trigger requests. If the CPTxIEN and CPTxDEN bits in the SHRTIM_TxIDEN register are set to 1, the capture can generate an interrupt or DMA request.

The circuit does not employ a mechanism to avoid over-capture: a new capture will be triggered even if the previous value has not been read or the capture flag has not been cleared.

Figure9-10 Timer Unit Capture Circuitry



9.3.4.11 Auto-delayed mode

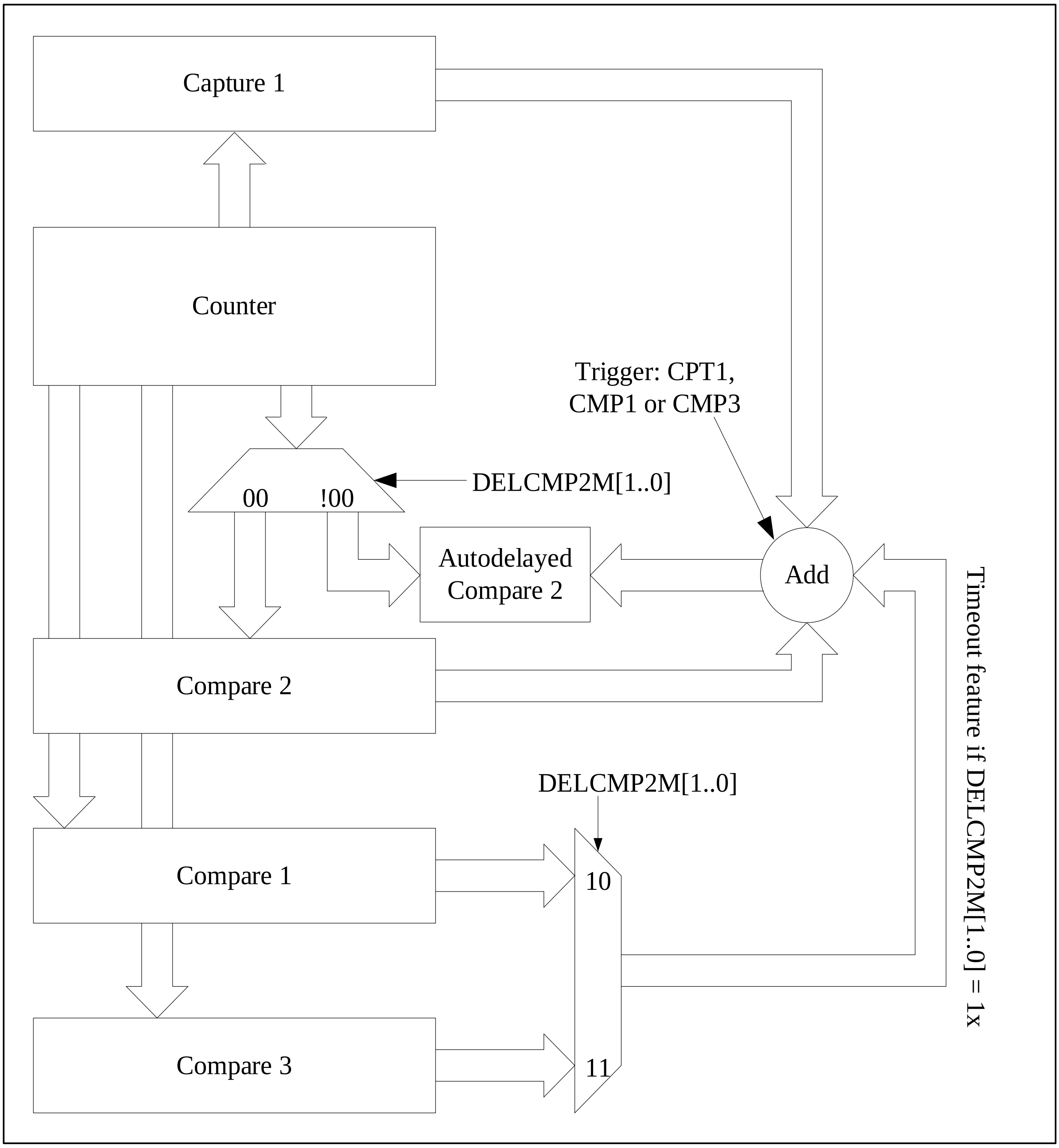
This mode allows compare events to be generated relative to capture events, enabling operations such as output changes to be performed at a set time after capture. In this case, the compare match will occur independently of the timer counter value. This allows for the generation of waveforms synchronized with external events without the need for software calculations and interrupt servicing.

As long as no capture event is triggered, the contents of the SHRTIM_TxCMPyDAT register will be ignored (no compare event will be generated when the counter value matches the compare value). Once a capture event is triggered, the compare value programmed in SHRTIM_TxCMPyDAT is summed with the captured counter value in

SHRTIM_TxCPTy, and the result is used to update the internal auto-delay compare counter, as shown in Figure9-11. The auto-delay compare register is an internal register of the timing unit and cannot be read. The content of the SHRTIM_TxCMPyDAT preload register will not be modified after the calculation.

This feature only applies to the compare 2 and compare 4 registers. Compare 2 register is associated with capture 1, while compare 4 register is associated with capture 2. Like in the regular mode, the SHRTIM_TxCMP2DAT and SHRTIM_TxCMP4DAT compare registers cannot be programmed with values less than 3 f_{SHRTIM} clock cycles.

Figure9-11 Auto-Delayed Overview (Compare 2 Only)



The validity of the auto-delay compare register starts from the capture and lasts until the period event: after the counter reaches the period value, the system will re-armed, and the compare registers will be disabled before the capture occurs.

The DELCMP2M[1:0] and DELCMP4M[1:0] bits in the SHRTIM_TxCTRL register can configure the auto-delay mode as follows:

- 00

  Regular compare mode: directly compares the contents of the SHRTIM_TxCMP2DAT and SHRTIM_TxCMP4DAT registers with the counter value.

- 01

  Auto-delay mode: recalculates the values of compare 2 and compare 4 registers and compares the calculated value with the counter value after capture 1/2 event.

- 1X

  Auto-delay mode with timeout: recalculates the values of compare 2 and compare 4 registers and compares the calculated value with the counter value after capture 1/2 events. If capture 1/2 events is missing, the comparison is done with the counter value after compare 1 match (DELCMPxM[1:0]=10) or compare 3 match (DELCMPxM[1:0]=11) to implement the timeout function.

When the capture occurs, it is compared with the value (SHRTIM_CMP2/4xR + SHRTIM_CPT1/2xR). If no capture is triggered within the period, the behavior will depend on the DELCMPxM[1:0] value:

- DELCMPxM[1:0] = 01: no compare event is generated
- DELCMPxM[1:0] = 10 or 11: It is compared with the sum of the values in the 2 compare registers (for example, SHRTIM_TxCMP2DAT + SHRTIM_TxCMP1DAT). If the capture is triggered after CMP1 (or CMP3), the capture will not be considered. The capture will be re-enabled at the beginning of the next PWM period.

If the result of the auto-delay summation exceeds 0xFFFF (overflow), the value will be ignored, and no compare event will be generated before the start of the new period.

Note: If DELCMPxM[1:0] is reprogrammed from one value to another to correctly reinitialize the automatic delay mechanism, the DELCMPxM[1:0] bit field must be reset, for example:

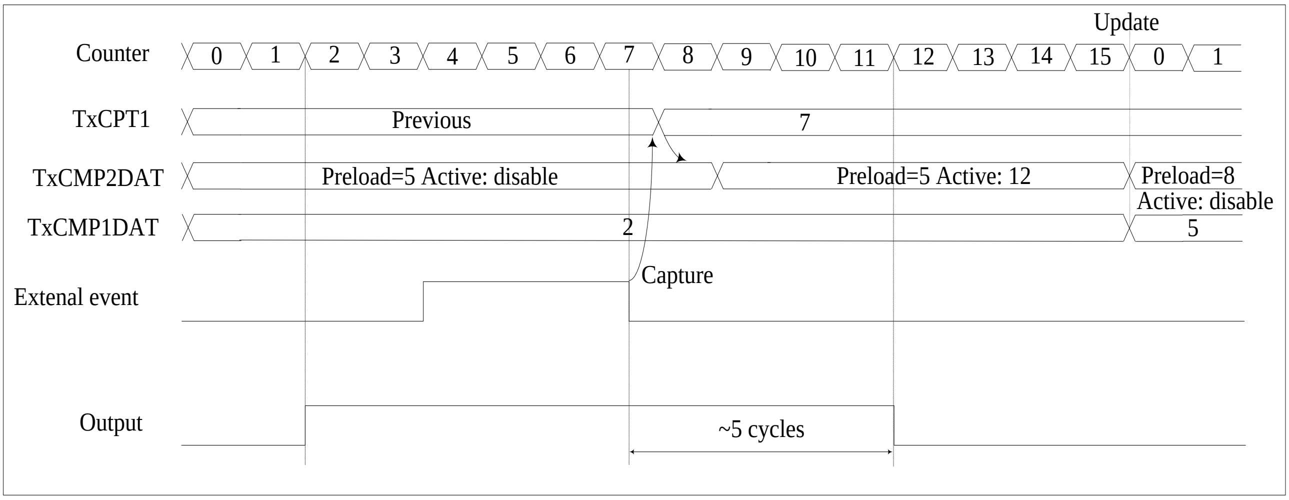
  - DELCMPxM[1:0] = 10
  - DELCMPxM[1:0] = 00
  - DELCMPxM[1:0] = 11

The following example illustrates how the following signals are generated:

- Output set when the counter equals the compare 1 value
- Output reset 5 cycle after the falling edge of a given external event

Note: to simplify the diagram, high-resolution is not used in this example (CKPSC[2:0] = 101), so the counter increases at the f_{SHRTIM} rate. Similarly, the external event signal shows no resynchronization delay: in reality, there is a delay of 1 to 2 f_{SHRTIM} clock cycles between the falling edge and the capture event due to internal resynchronization stages (which are necessary for processing external input signals).

Figure9-12 Auto-Delayed Compare



Using the regular compare channel (such as compare 1) for output set: the output will immediately become active when the counter matches the content of the compare register.

Using delayed compare for output reset: a compare event will only be generated when a capture event occurs. If the counter matches the delayed compare value (counter value = 5), no event will be generated. After a capture event triggered by an external event, the content of the capture register is immediately added to the delayed compare value to have a new compare value. In the example, the auto-delay value of 5 is added to the capture value of 7, resulting in a value of 12 in the auto-delay compare register. From this point on, a compare event can be generated and will occur when the counter equals 12, causing the output to be reset.

Overcapture management in auto-delayed mode

When the auto-delay mode is enabled (DELCMPxM[1:0] = 01, 10, 11), overcapture is prevented.

If multiple capture requests occur within the same counting period, only the first capture request will be considered to calculate the auto-delay compare value. A new capture can only occur under the following conditions:

- Auto-delay compare has matched counter value (compare event)
- The counter has rolled over (period)
- The timer has reset

Change the auto-delay compare value

If the auto-delay compare value has been preloaded (PLEN set to 1), the new compare value (compare value written by software) will be considered at the next update event (such as a period event) regardless of when the compare register was written, or whether a capture event occurred (refer to Figure9-12, where the delay is changed on counter rolls over).

If preload is disabled (PLEN bit reset), the new compare value will be considered immediately even if the compare value was modified after a capture event, as shown in the following example:

1. At t_1, DELCMP2M = 1
2. At t_2, CMP2_act = 0x40 => comparison disabled
3. At t_3, a capture event occurs, capturing the value CPTR1 = 0x20. => Comparison enabled, new compare value

= 0x60

4. At t4, CMP2_act = 0x100 (before the counter value reaches CPTR1 + 0x40) => comparison still enabled, new compare value = 0x120

5. At t5, the counter value reaches the period value => comparison disabled, cmp2_act = 0x100

Similarly, if the value of CMP1(CMP3) changes when DELCMPxM = 10 or 11, and preload is disabled:

1. At t1, DELCMP2M = 2

2. At t2, CMP2_act = 0x40 => comparison disabled

3. At t3, CMP3 event occurs - before capture 1 event, CMP3_act = 0x50 => comparison enabled, new compare value = 0x90

4. At t4, CMP3_act = 0x100 (before the counter reaches 0x90) => Comparison still enabled, compare 2 event will occur when CMP3_act = 0x140.

9.3.4.12 Triggered-half mode

The purpose of this mode is to allow synchronization of two interleaved converters with variable frequency operation and requiring a 180° phase-shift. The basic principle is:

There is a master-slave system. The synchronization of the slave converter is continuously adjusted based on the previous switch period of the master converter.

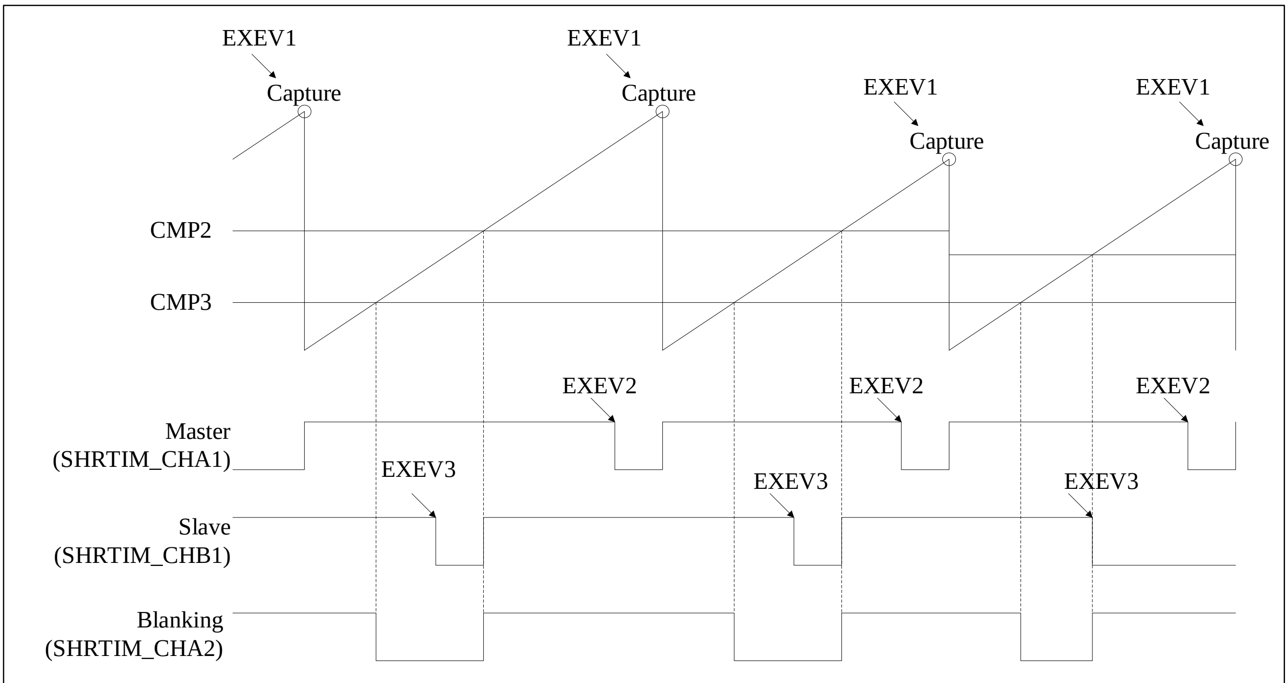
This is achieved through the capture unit. The switch period of the master converter is captured, divided by 2, and then stored in the compare 2 register by the hardware. The compare 2 register contains the value equal to half of the captured period, which is the switch period of the master converter. The compare 2 event can then be used to trigger the second timing unit of the slave converter management.

Enable this mode by setting the TRGHLF bit in the SHRTIM_TxCTRL2 register. Once the timer is operating (TxCNTEN bit set), this bit cannot be changed.

The trigger-half mode must not be used simultaneously with other modes that use CMP2 (such as dual DAC trigger, interleaved, and balanced idle modes, auto-delayed mode).

The user can write an initial value to CMP2, but once the first capture is triggered, this value will be ignored. When the TRGHLF bit is set, the preload mechanism for CMP2 is disabled.

Figure9-13 Triggered-Half Mode Example



SHRTIM_CHA1
Set on EXEV1(EXEV1 triggers capture 1)
Reset on EXEV2
EXEV1 Blanking on TIMA_CMP1 (avoid frequency run-away)
SHRTIM_CHB1
Set on CMP2 (triggered-half) or EXEV1 blanking by TIMA_TA2
(TB1 is set if EXEV1 occurs between CMP1 and CMP2,else on CMP2)
Reset on EXEV3
SHRTIM_CHA2
Set on TA_CMP1
Reset on TA_CMP2

Note: In the trigger-half mode, the compare 2 register is controlled by hardware, and writing to it has no effect. However, the written values are stored in the preload register and become active on the update event after exiting this mode.

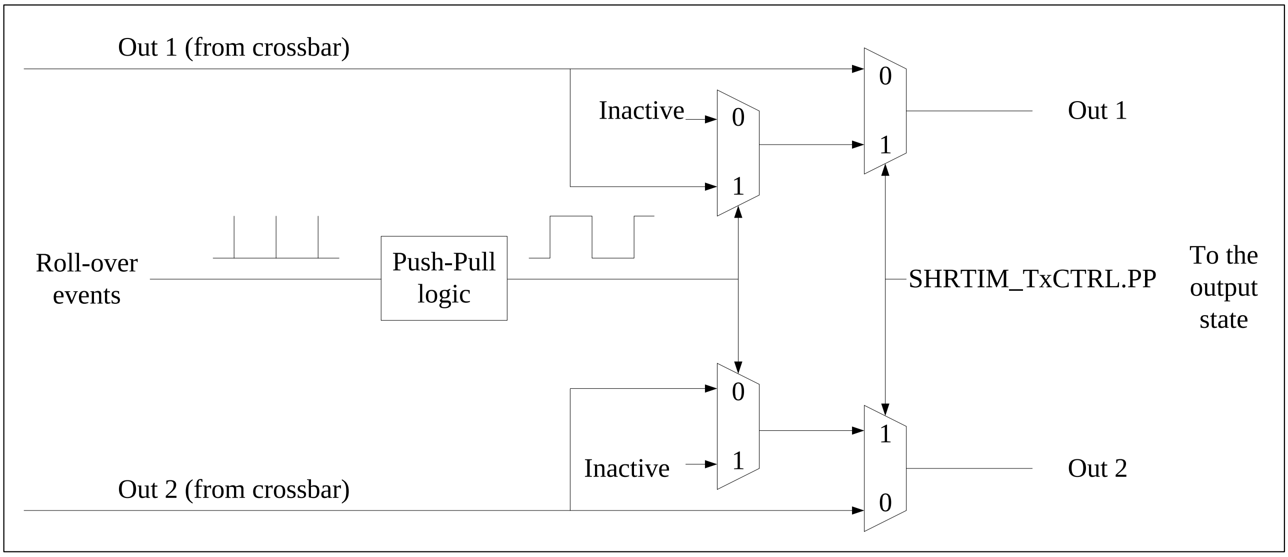
9.3.4.13 Push-pull mode

The main purpose of this mode is to drive converters using a push-pull topology. If delayed idle protection is required (commonly used for resonant converters), this mode also needs to be enabled (refer to Section 9.3.10 Delayed Protection).

The push-puull mode is enabled by setting the PP bit in the SHRTIM_TxCTRL register.

In this mode, the signals generated by the crossbar are alternately applied to output 1 and output 2, on the period basis (if the signal is applied to output 1, output 2 remains inactive, and vice versa). The redirection rate (push-pull frequency) is defined by the timer's period event, as shown in Figure9-14. The push-pull period is twice the timer counting period.

Figure9-14 Push-Pull Mode Block Diagram



The push-pull mode is available when the timer operates in continuous and single-shot modes. The timer must be disabled to stop the push-pull operation, and the counter must be reset before re-enabling it.

To get a correct behavior, the event selected as the counter reset source must also be selected to set (or reset) the output. If the output is set on the period, it must be set, otherwise it must be reset. If it is not done, incorrect behavior may occur when the output switches from the inactive period to the active period (it may unexpectedly rise or may unexpectedly stay low).

The signal waveforms for the two outputs are defined by SHRTIM_TxSETy and SHRTIM_TxRSTy. To ensure both output waveforms are identical and achieve balanced operation, it is necessary that SHRTIM_TxSET2 = SHRTIM_TxSET1 and SHRTIM_TxRST2 = SHRTIM_TxRST1. However, it is possible to have different programming on both outputs for other purposes.

The CPPSTS status bit in SHRTIM_TxINTSTS indicates which output signal is currently active. CPPSTS is reset when the push-pull mode is disabled.

In the example provided in the diagram below, the internal waveform of the timer is defined as follows:

- Output set on period event
- Output reset on compare 1 match event

Figure9-15 Push-Pull Mode Example

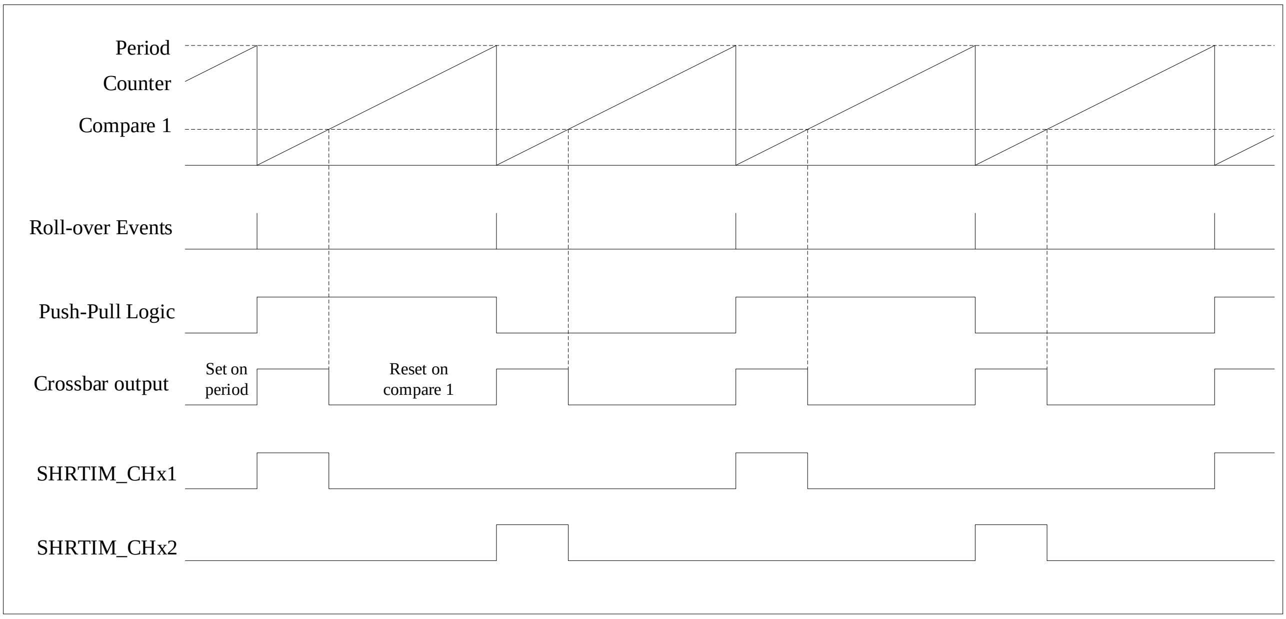
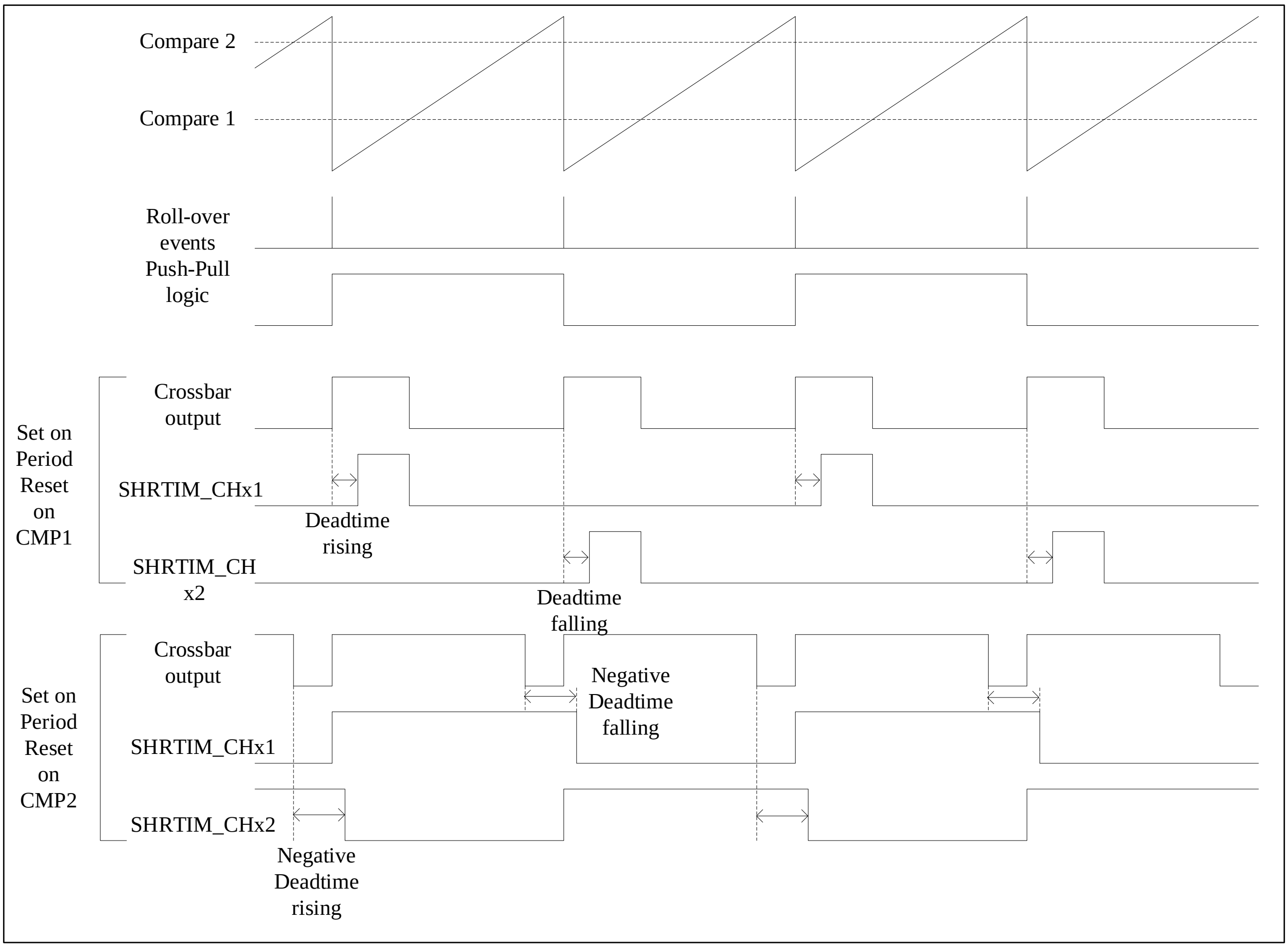


Figure9-16 shows how to insert positive and negative dead times in push-pull mode. In this case, the outputs are no longer complementary, but the dead time is applied individually to each output (outputs 1 and 2 of the crossbar are used).

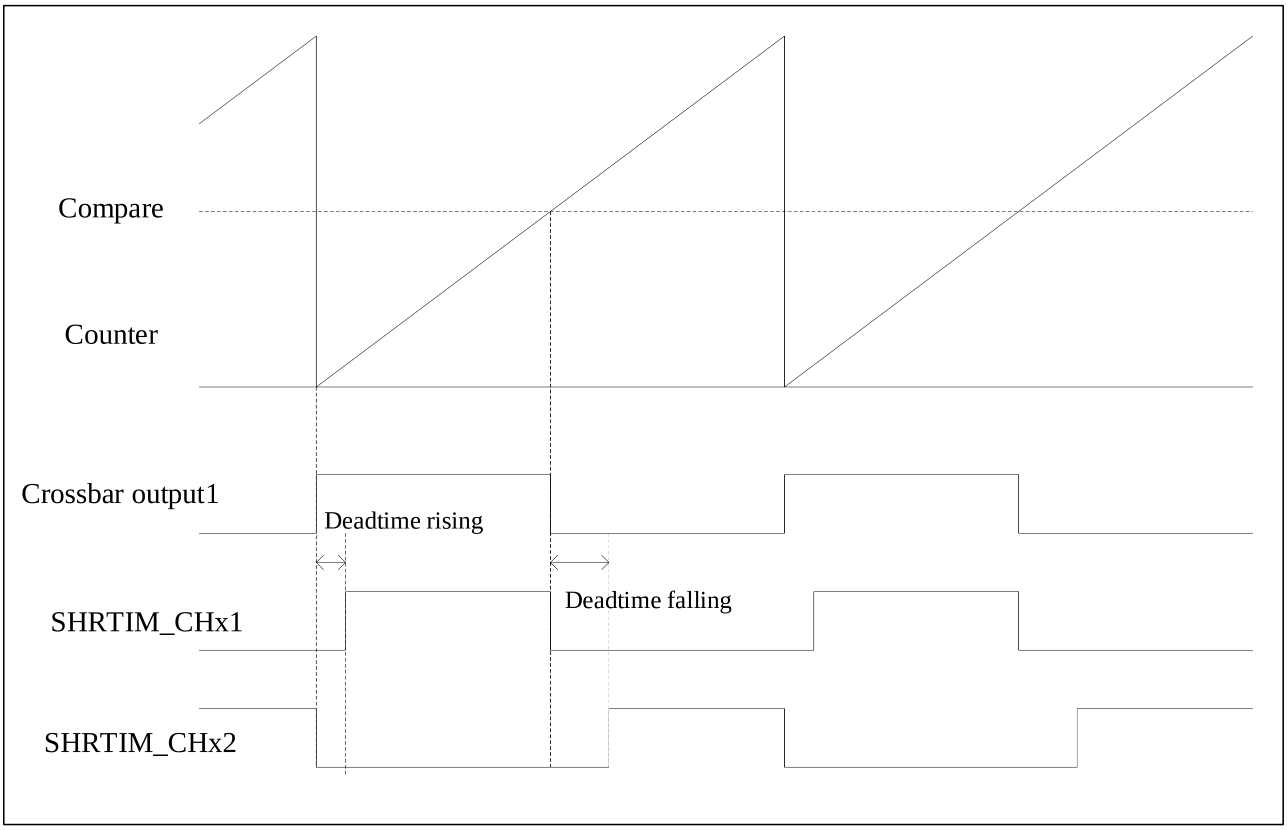
Figure9-16 Push-Pull with Deadtime



9.3.4.14 Deadtime

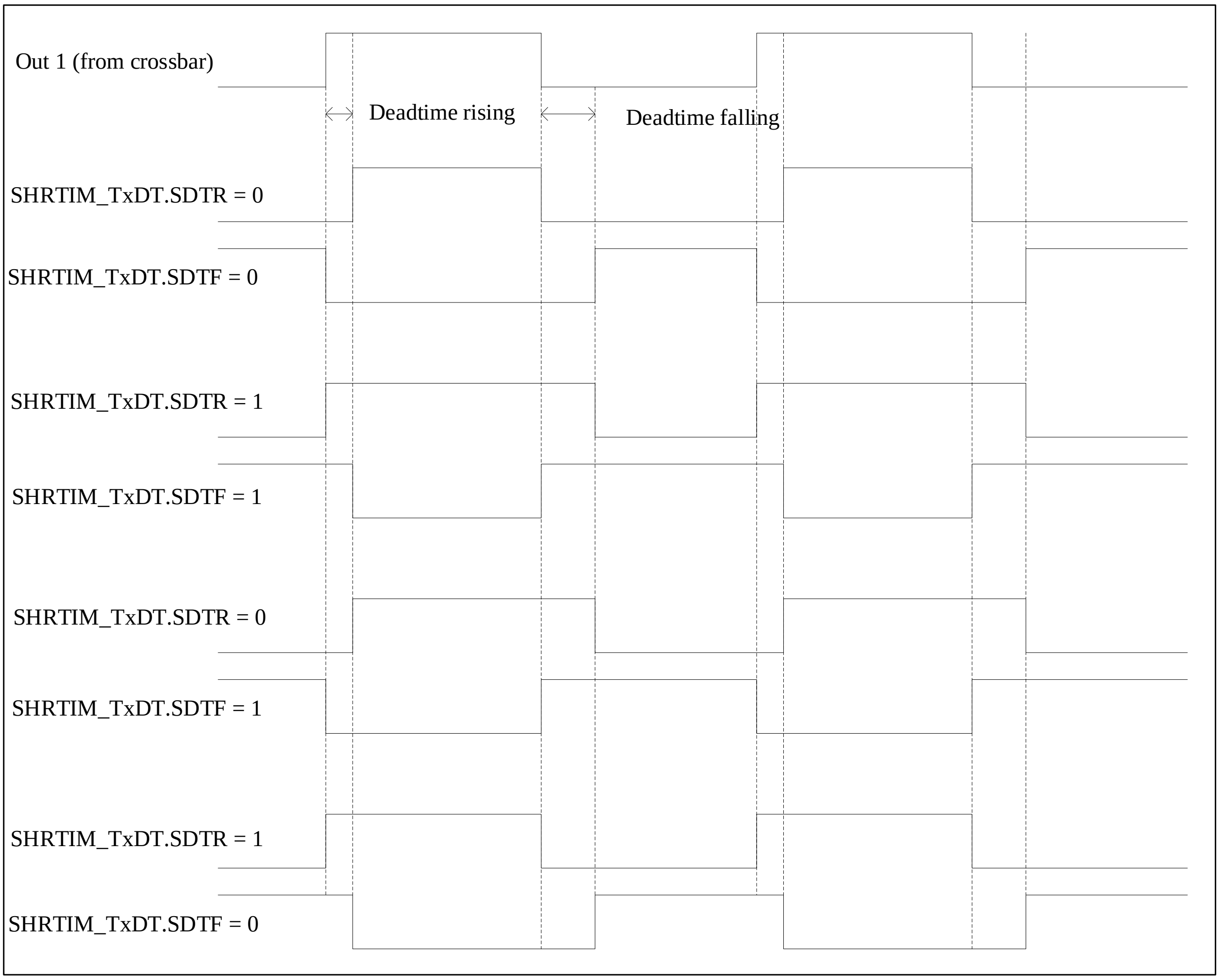
A deadtime insertion unit can generate a pair of complementary signals from a single reference waveform, and the delay between the active state transitions can be programmed. This is commonly used in topologies using half-bridge or full-bridge configurations. It simplifies the software operation flow: only one waveform needs to be programmed and controlled to drive two outputs. The deadtime insertion is enabled by setting the DTEN bit in the SHRTIM_TxOUT register. The complementary signals are built based on the reference waveform defined for output 1, using the SHRTIM_TxSET1 and SHRTIM_TxRST1 registers: if DTEN is set to 1, the SHRTIM_TxSET2 and SHRTIM_TxRST2 registers are disabled.

Figure9-17 Complementary Outputs with Deadtime Insertion



If some control overlap is needed, a negative dead-time value can be defined, in which case the dead-time sign bit (SDTF and SDTR bits in the SHRTIM_TxDT register) should be used. Figure9-18 shows the complementary signal waveforms corresponding to each sign.

Figure9-18 Deadtime Insertion versus Deadtime Sign (1 Indicates Negative Deadtime)



The deadtime value is defined using the DTF[8:0] and DTR[8:0] bit fields, based on a specific clock prescaled by the DTPSC[2:0] bits, as follows:

t_{DTx} = +/- DTx[8:0] x t_{DTG}

where x is R or F, and $t_{DTG} = (2^{(DTPSC[2:0])})$ x (t_{SHRTIM} / 8).

The table below provides the resolution and maximum absolute value based on the prescaling value.

Table9-13 Deadtime Resolution and Max Absolute Values

DTPSC[2:0]	t_{DTG}	t_{DTx} max	f_{SHRTIM} = 250 MHz	
			t_{DTG}(ns)	$\lvert t_{DTx}\rvert$ max (us)
000	t_{SHRTIM} / 8	511 * t_{DTG}	0.5	0.26
001	t_{SHRTIM} / 4		1.0	0.51
010	t_{SHRTIM} / 2		2.0	1.02
011	t_{SHRTIM}		4	2.04
100	2 * t_{SHRTIM}		8	4.09
101	4 * t_{SHRTIM}		16	8.18

110	8 * t_{SHRTIM}		32	16.35
111	16 * t_{SHRTIM}		64	32.70

The following diagram illustrates how the deadtime generator handles the reference waveform with pulse widths smaller than the deadtime value in all deadtime configurations.

Figure9-19 Complementary Outputs for Low Pulsewidth (SDTR = SDTF = 0)

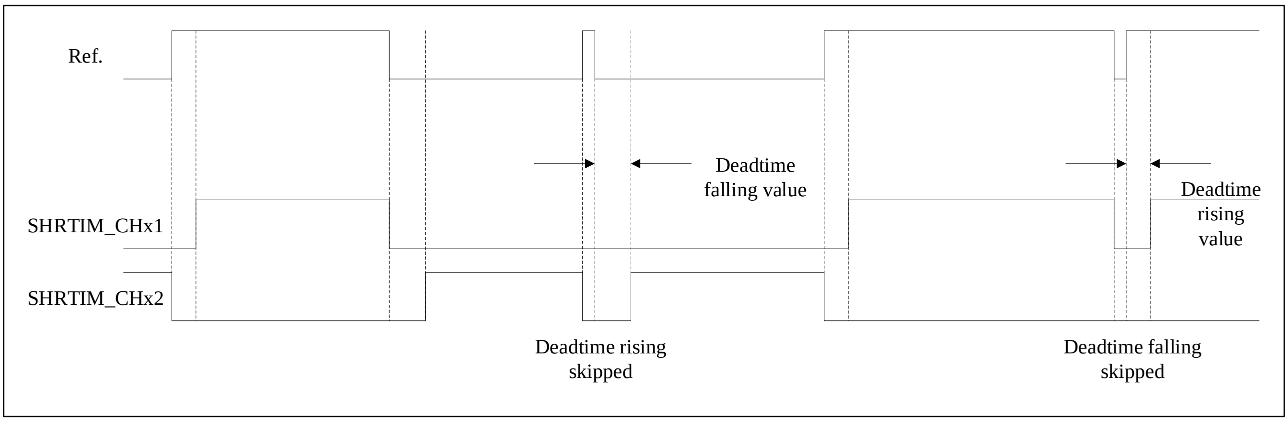


Figure9-20 Complementary Outputs for Low Pulsewidth (SDTR = SDTF = 1)

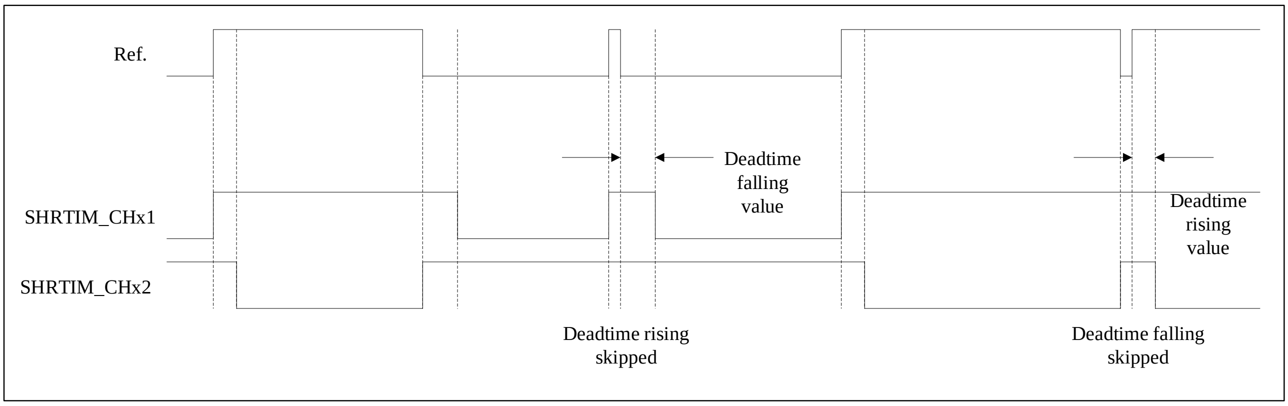


Figure9-21 Complementary Outputs for Low Pulsewidth（SDTR = 0，SDTF = 1）

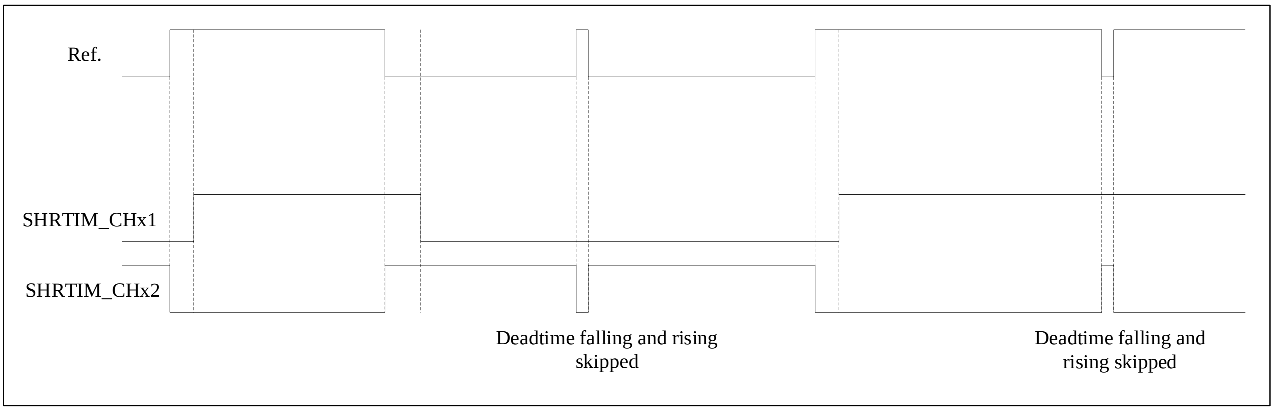
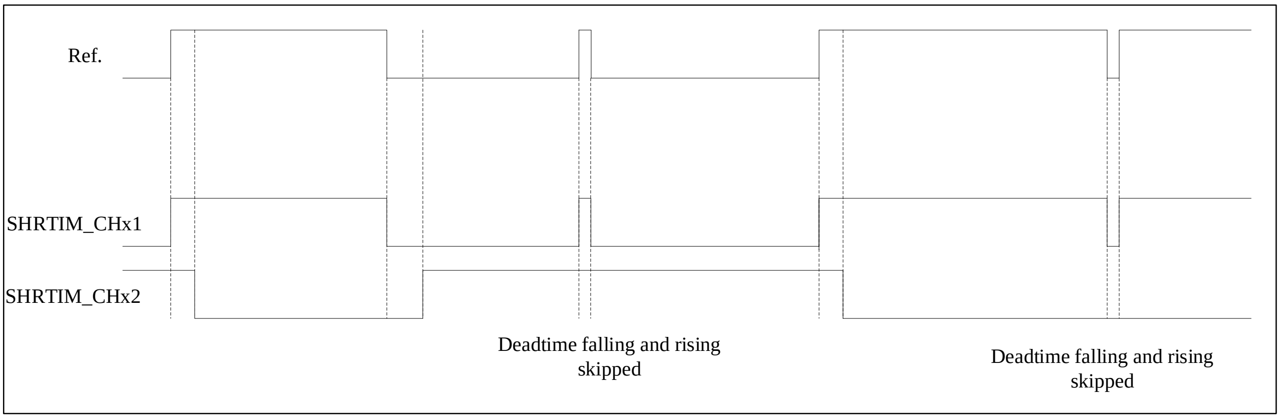

Ref.
SHRTIM_CHx1
SHRTIM_CHx2
Deadtime falling and rising skipped
Deadtime falling and rising skipped

Figure9-22 Complementary Outputs for Low Pulsewidth（SDTR = 1，SDTF = 0）


Ref.
SHRTIM_CHx1
SHRTIM_CHx2
Deadtime falling and rising skipped
Deadtime falling and rising skipped

For safety considerations, the DTFLCK, DTRLCK, DTFSLCK, and DTRSLCK can be used to lock the sign and/or value of the deadtime to prevent accidental writes to the deadtime registers. Once these bits are set to 1, the associated bits and bit fields will become read-only until the next system reset.

Note: The DTEN bit must not be changed in the following cases:

- *When the timer is enabled (TxCNTEN bit set to 1)*
- *When the timer output is set/reset by another timer (TxCNTEN reset)*

Otherwise, unpredictable behavior may occur.

Therefore, it is necessary to disable the timer (TxCNTEN bit reset) and disable the corresponding output.

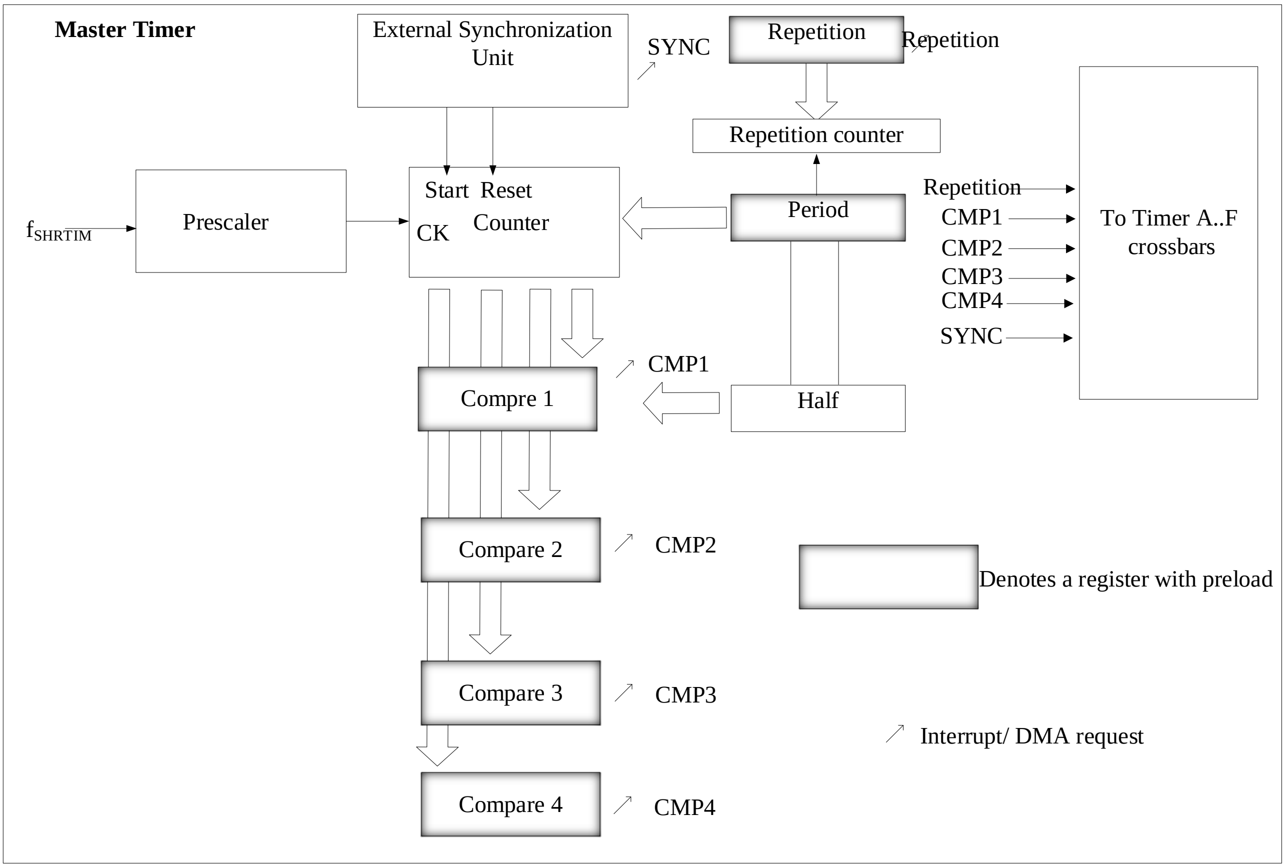
For the particular case where DTEN must be set to 1 while the burst mode is enabled with a deadtime upon entry (BMEN = 1, DIDL = 1, IDLEM = 1). Before setting DTEN to 1, it is necessary to use a software command to force the two outputs to their IDLES state (SWT, RSTROITF bits). This is to avoid any side effect resulting from a burst mode entry that would happen immediately before a deadtime enable.

9.3.5 Master Timer

The main purpose of the master timer is to provide a common signal to 6 timing units for synchronization or to set/reset outputs. The master timer does not directly control any outputs, but it can still be indirectly used for by the set/reset crossbar.

The overview of the master timer is shown in the diagram below.

Figure9-23 The Master Timer Overview.



The architecture of the master timer is very similar to the timing units, with the following differences:

- The master timer is not associated with outputs and does not have output-related controls.
- The master timer does not have its own crossbar unit, push-pull, or deadtime modes.
- The master timer can only be reset through an external synchronization circuit.
- The master timer does not have capture units or auto-delay modes.
- The master timer does not include external event bankking and window circuits.
- The number of interrupt/DMA requests for the master timer is limited: compare 1..4, repetition, register update, and external synchronization events.

The control register of the master timer includes all timer enable bits for the master timer and timing units A..F. This allows for simultaneous start of all timers with a single write access.

The master timer will also utilize internal and external (input/output) resources of the MCU to handle the external synchronization of the entire SHRTIM timer (refer to Section 9.3.19 Synchronize the SHRTIM with other timers or SHRTIM instances).

The mapping offset of the master timer control register is the same as the offset of the timing unit registers.

9.3.6 Up-down Counting Mode

The SHRTIM itself is designed as an up-counter. However, it also provides an operational mode that supports both up-down counting, known as center-aligned mode. This mode is enabled by using the UPDOWNM bit in the SHRTIM_TxCTRL2 register. Once the timer starts running (TxCNTEN bit set), this bit should not be changed. It is only applicable to the TIMA..F timers. The master timer can only operate in an up-counting mode. Not all SHRTIM functions support up-down counting. This section details the functional differences compared to the up-counting mode. In the up-down mode, the period in SHRTIM_TxPRD must be preloaded (or static). It can only be updated on a period event or counter reset.

Note: When the counter is disabled during a down-counting period (TxCNTEN reset), the counter will continue to count up.

The differences in programming the set/reset crossbar are as follows:

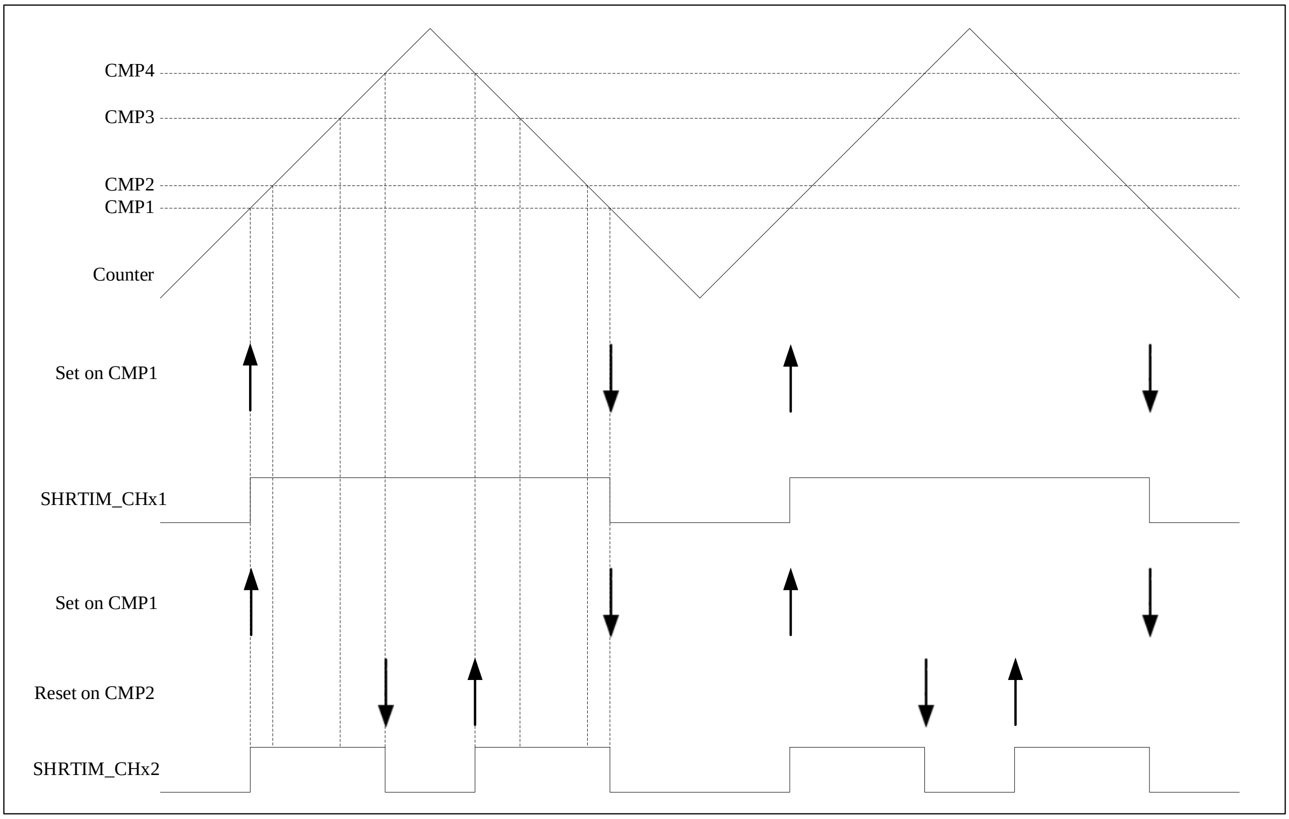
The events from the timing units will set/reset outputs based on the up/down counting direction of the counter:

- If the event is enabled in the SHRTIM_TxSETy register, the output will be set during up counting and reset during down counting.
- If the event is enabled in the SHRTIM_TxRSTy register, the output will be reset during up counting and set during down counting.
- If the event is enabled in both the SHRTIM_TxSETy and SHRTIM_TxRSTy registers simultaneously, the output will toggle.

This applies to:

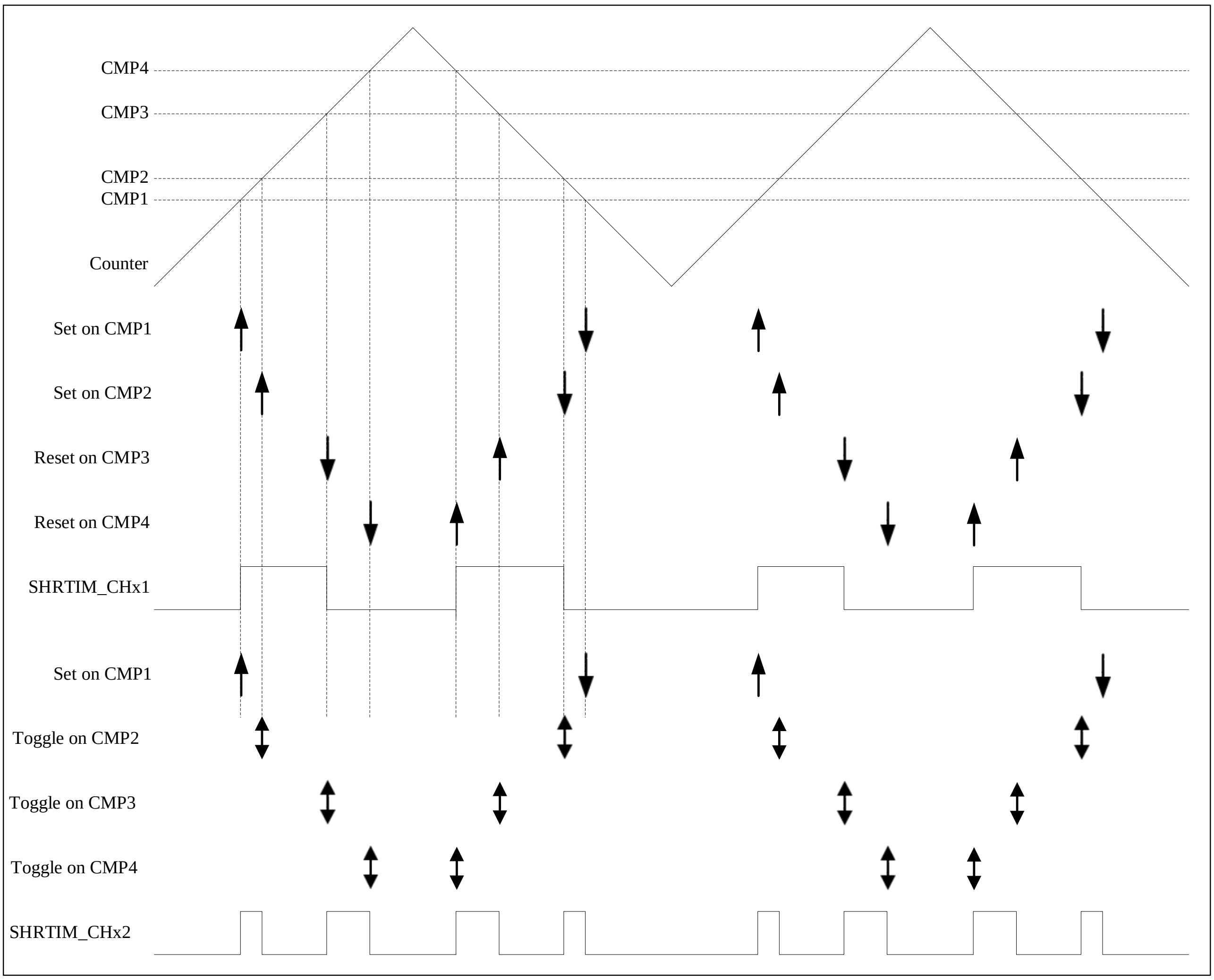
- Timing units: period, compare 1..4, register update (6 events)
- Master timer: period, compare 1..4, SHRTIM synchronization (6 events)
- All other timing units (e.g. timer B..F for timer A): TIMEV1..9 (9 events described) The following content shows how to generate basic waveforms.

Figure9-24 Basic Symmetric Waveform in Up-Down Counting Mode



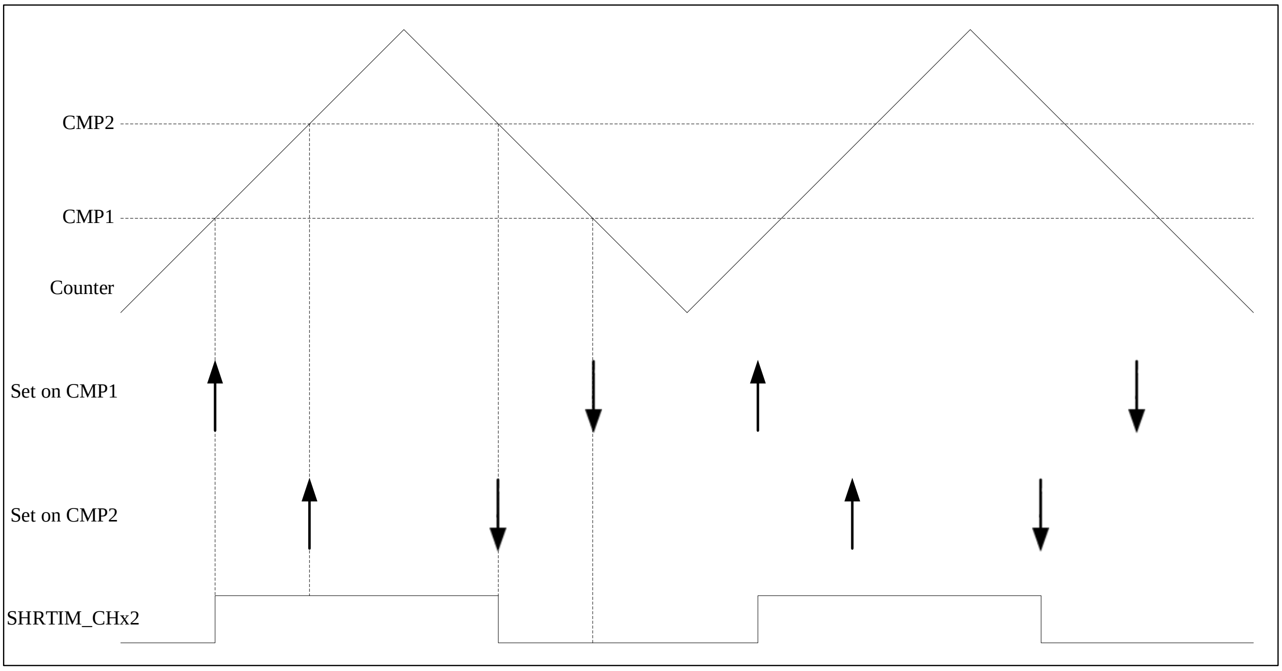
The following content demonstrates how to use the 4 available compare units and the toggle modes to generate some more complex waveforms.

Figure9-25 Complex Symmetric Waveform in Up-Down Counting Mode



The following content demonstrates how to generate asymmetric waveforms. In this case, it is important to note that for waveform asymmetry, the value of compare 2 must be greater than the value of compare 1.

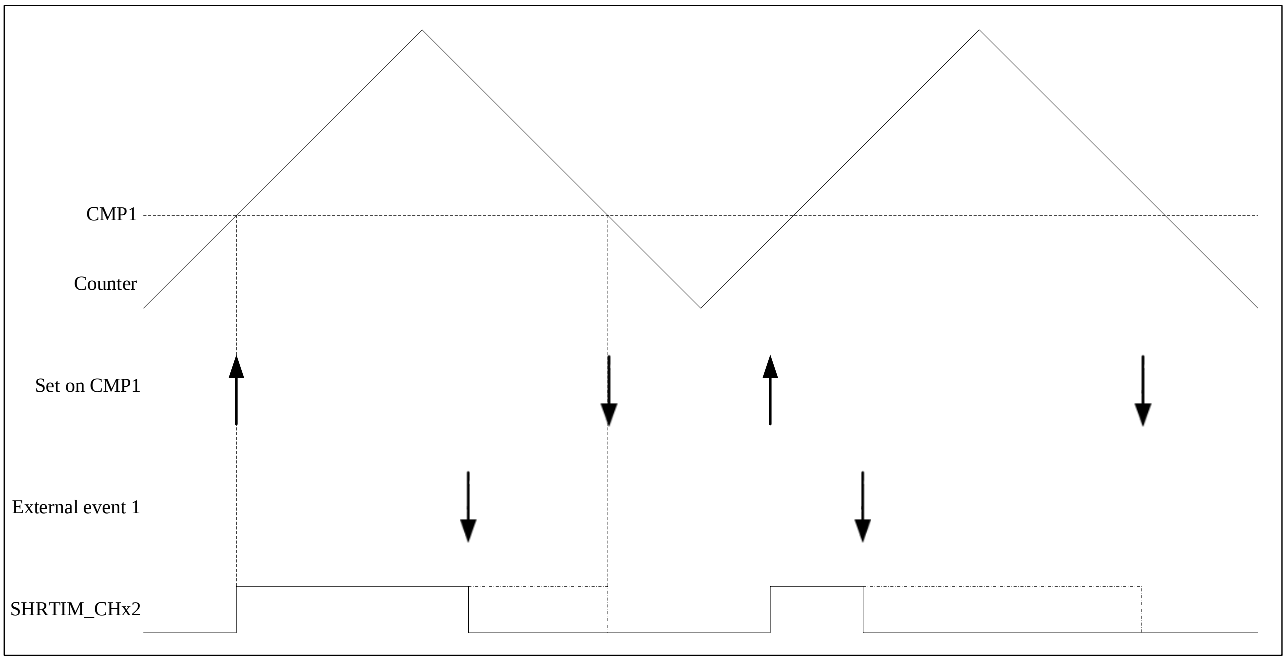
Figure9-26 Asymmetric Waveform in Up-Down Counting Mode



Note: For asymmetric operation, it is required that CMP2 > CMP1.

The behavior of software force bits and external events EXEV1..10 is the same in both up-counting and up-down counting modes. The following content demonstrates how to respond to external events to shorten pulses.

Figure9-27 External Event Management in Up-Down Counting Mode



The up-down counting mode is applicable to continuous and one-shot (retriggerable and non-retriggerable) operation modes. A reset will cause the counter to re-start from 0. The following content demonstrates the behavior of the counter in timer B in the one-shot retriggerable mode.

Figure9-28 Interleaved Up-Down Counter Operation

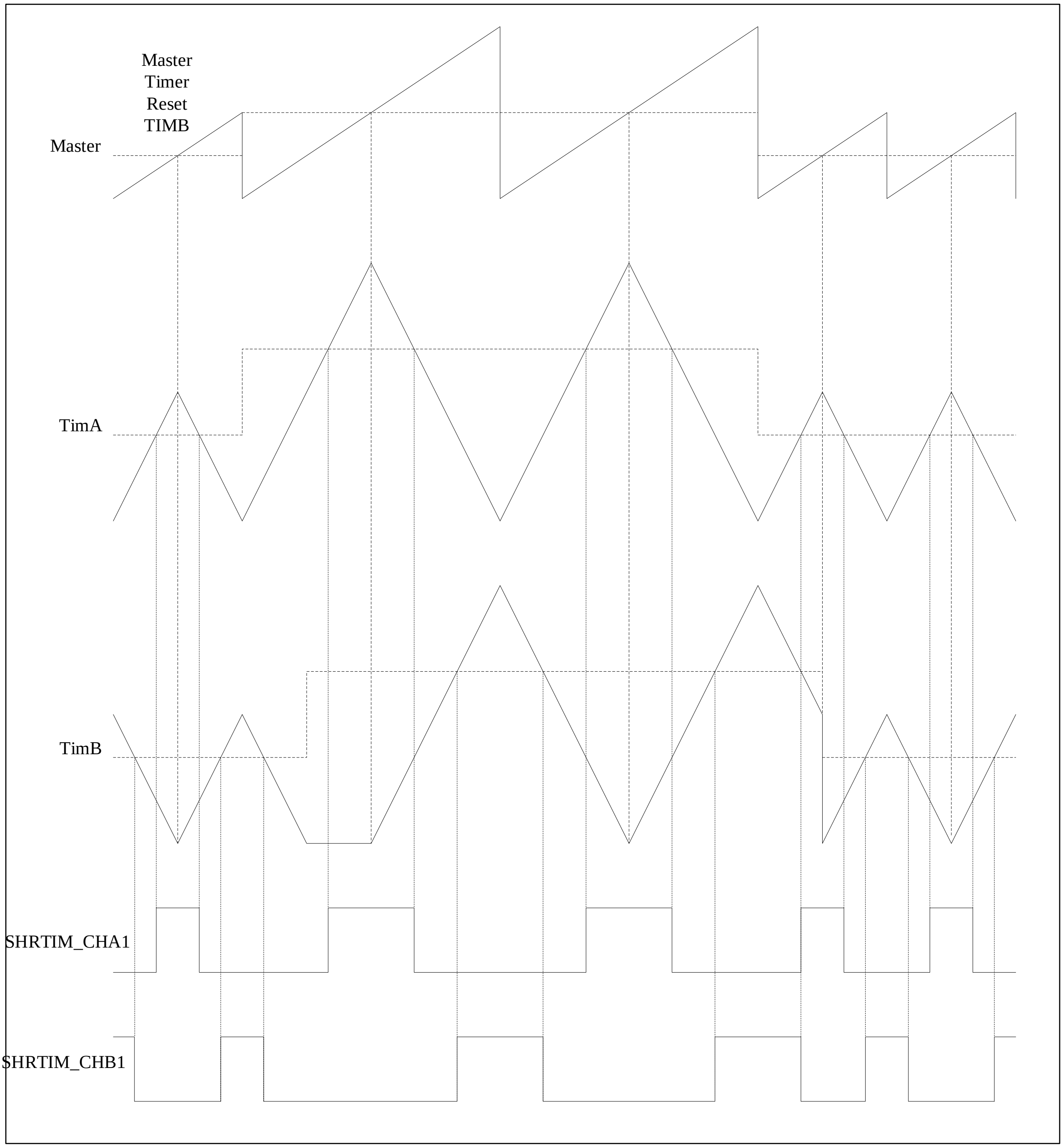
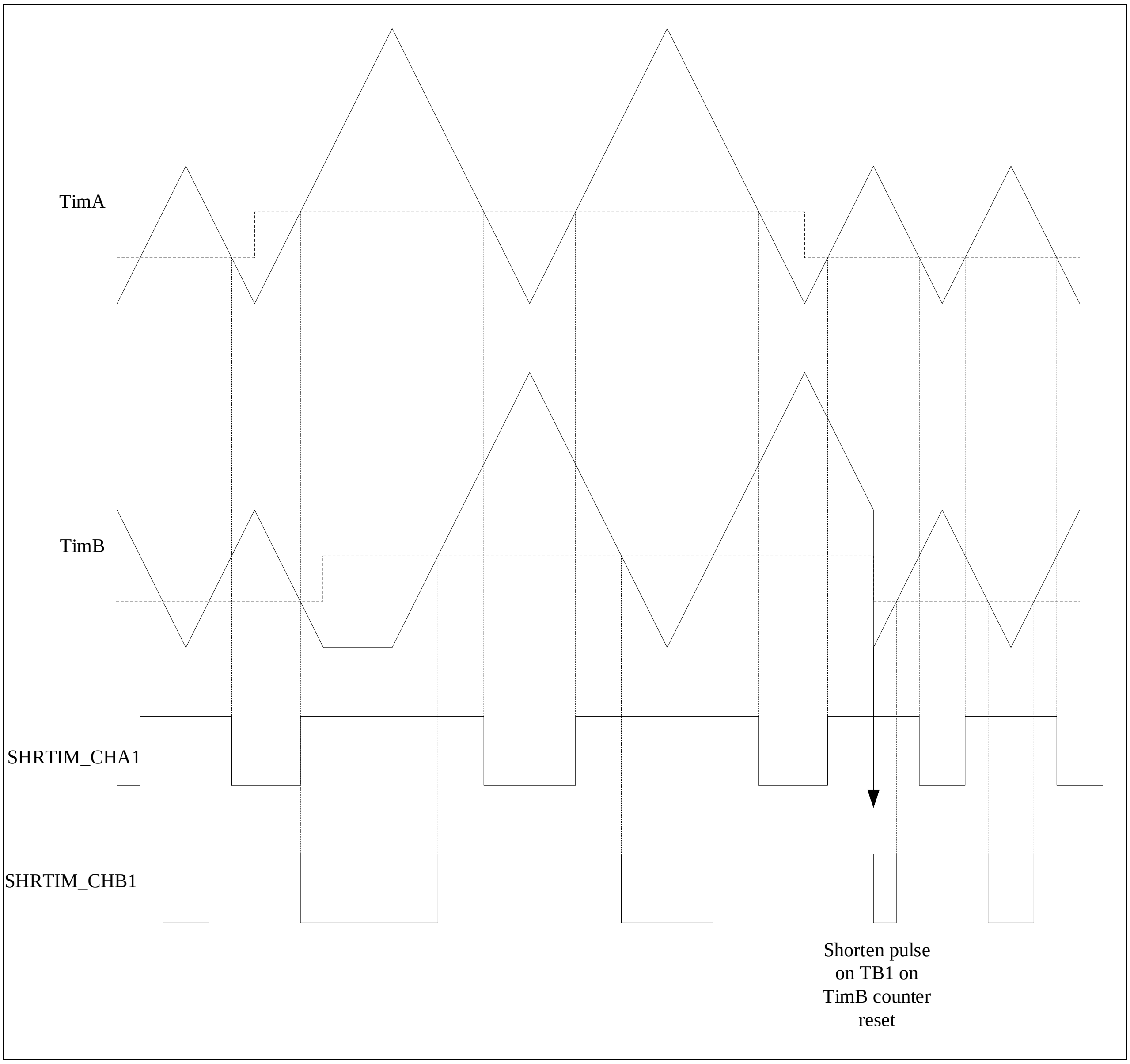


Figure9-29 Interleaved Up-Down Counter Operation



Note: In the up-down counting mode, the compare value must be lower than the period value by 3 f_{SHRTIM} clock cycles (if CKPSC[2:0] = 0, then TxPRD - 0xC0; if CKPSC[2:0] = 1, then TxPRD - 0x60; if CKPSC[2:0] = 2, then TxPRD - 0x30,...). This applies for compare events generated inside the timing unit. For compare events generated in other timing units, events must be avoided within the f_{SHRTIM} clock cycle of the counter direction change (counter reaches 0, period event, or counter reset).

The following features are supported in the up-down counting mode:

- Half mode
- Insertion of dead time
- Push-pull mode, alternance push-pull done on when the counter reaches 0 (refer to Figure9-30).
- Delayed idle mode

- Burst mode
- PWM mode with “greater than” comparison (refer to Figure9-31).

Figure9-30 Push-Pull Up-Down Mode Example

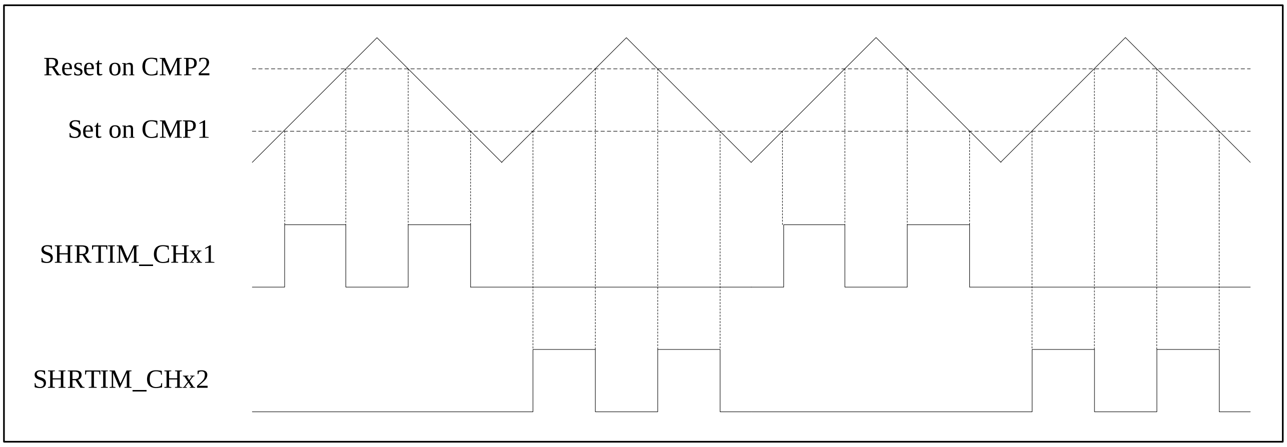
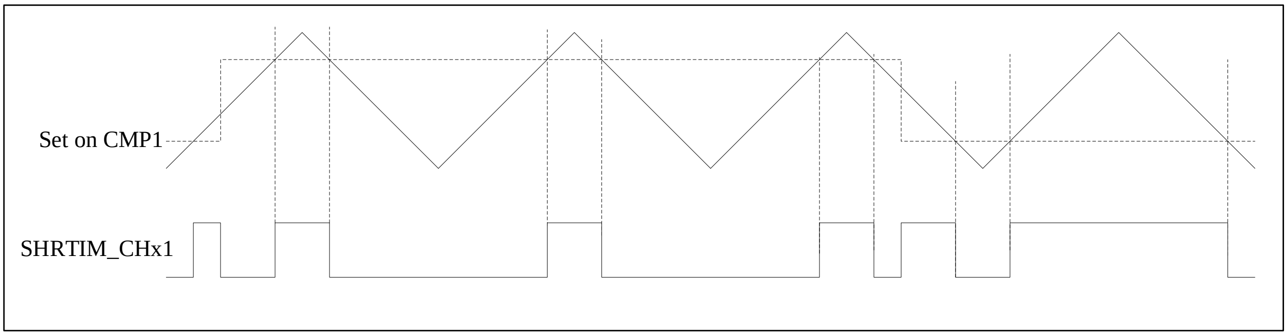


Figure9-31 Up-Down Mode with “Greater Than” Comparison



Note: The following features are not supported in the up-down counting mode:

- *Auto-delay mode*
- *Balanced idle*
- *Triggered-half mode*

The capture function is supported with the following differences:

- The bit 16 of the capture register retains the counter direction state

The counter roll-over event definition in the up-down counting mode is different to support various operating conditions. It can be generated in the following cases:

- When the counter reaches 0 ("valley" mode).
- When the counter reaches the period value set in SHRTIM_TxPRD ("crest" mode).
- When both conditions are met (0 or SHRTIM_TxPRD value).

This event in SHRTIM is used for various purposes. The roll-over (valley, peak, or both) can be programmed individually depending on the destination. The table below summarizes the use cases and the corresponding roll-over modes (xxROM[1:0]) programming bits in the SHRTIM_TxCTRL2 register.

Table9-14 Roll-Over Event Destination and Mode Programming

Roll-over event use	Programming bits
Output set/reset	OUTROM[1:0]
Register content update trigger (transfer from preload to active)	ROM[1:0]
IRQ and/or DMA request trigger	ROM[1:0]
Burst mode clock source and /or burst start trigger	BMROM[1:0]
ADC trigger (see ADC post-scaler for details)	ADCROM[1:0]
External event filtering	ROM[1:0]
Repetition counter decrement	ROM[1:0]
Fault and event counter	FEROM[1:0]

Note: For events where both reset and roll-over are considered (TxCTRL.RSTROUEN, IRQ/DMA, burst clock source and/or burst trigger, external event filtering, repeat counter decrement, fault and event counters, set/reset output), the reset event is taken into account.

The definition of roll-over event generation is as follows with the xxROM[1:0] bit setting:

- xxROM[1:0] = 00: event is generated when both conditions are met (0 or SHRTIM_TxPRD value).
- xxROM[1:0] = 01: event is generated when the counter reaches 0 ("valley" mode).
- xxROM[1:0] = 10: event is generated when the counter reaches the period value set in SHRTIM_TxPRD ("crest" mode).

Notes:The definition of the rollover event in single-shot mode is slightly different from continuous mode, as follows:

- *xxROM[1:0] = 00: event is generated when the counter is at 0 and the reset event occurs.*
- *xxROM[1:0] = 01: no event is generated.*
- *xxROM[1:0] = 10: event is generated when the counter is at 0 and the reset event occurs*

Figure9-32 Updown + single-shot，the generation of roll-over event

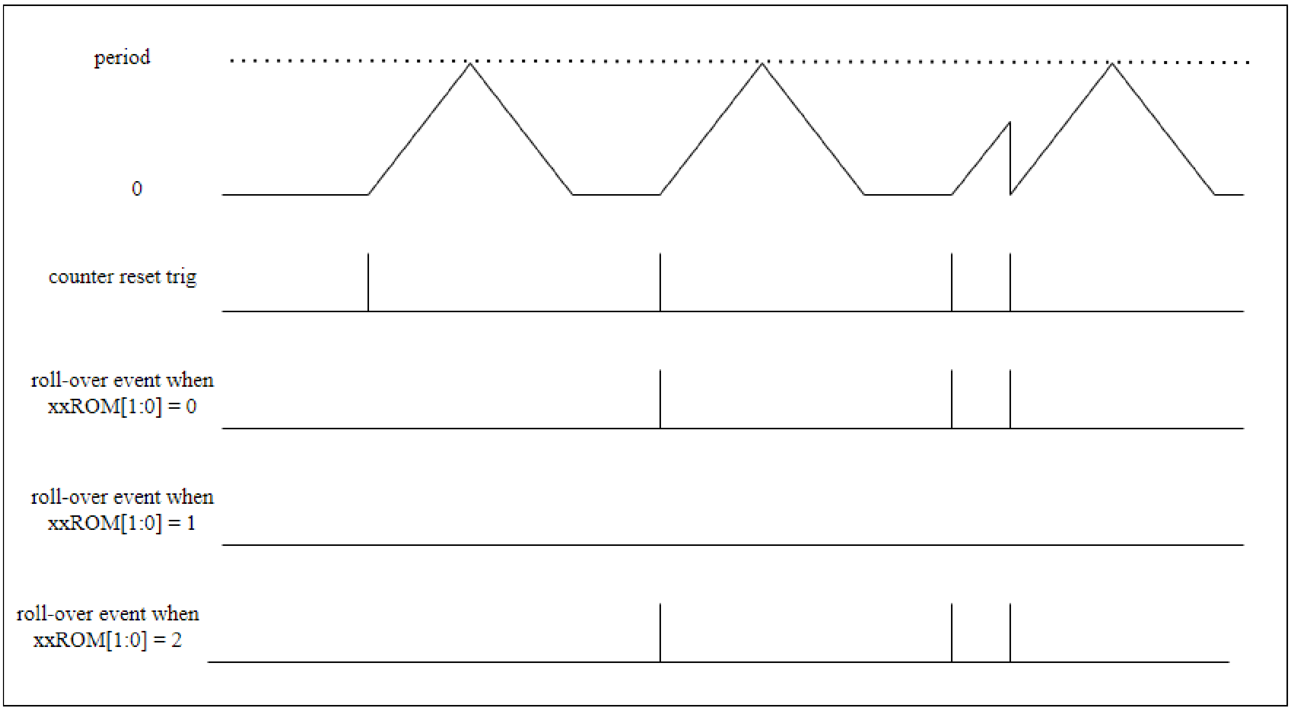
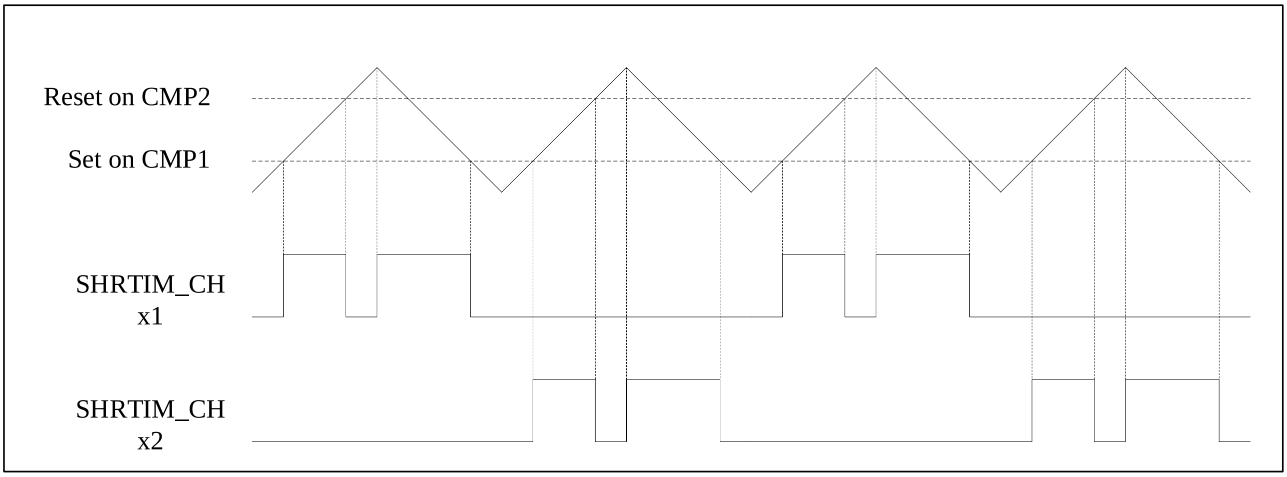
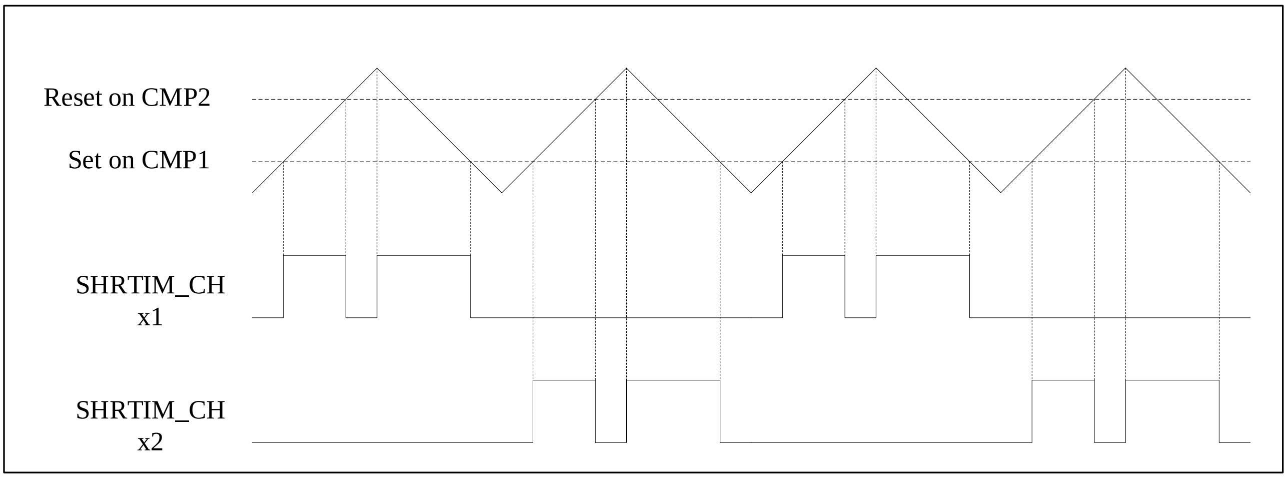

period
0
counter reset trig
roll-over event when xxROM[1:0] = 0
roll-over event when xxROM[1:0] = 1
roll-over event when xxROM[1:0] = 2

Figure9-33 Up-Down Mode With Output Set on Period Event, OUTROM[1:0]=10



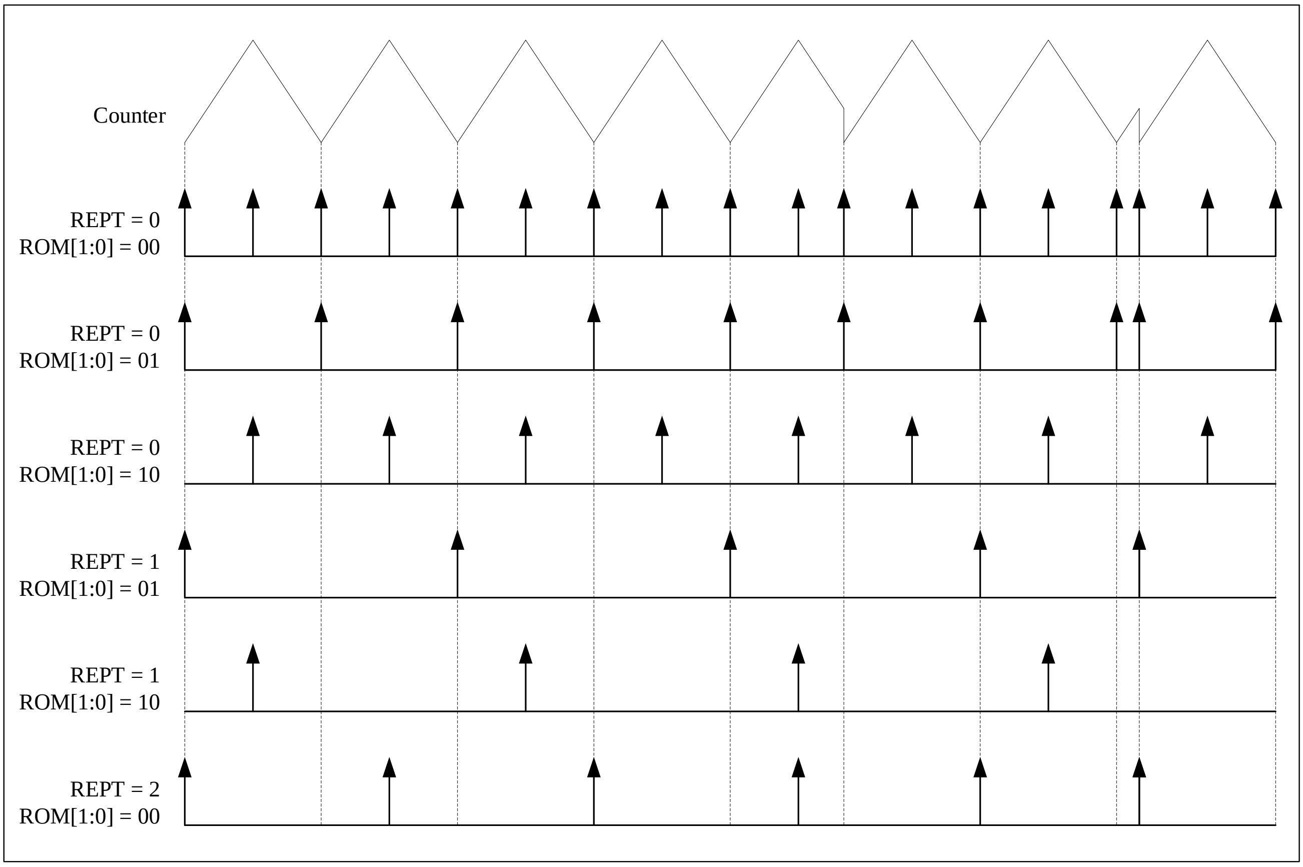
The following figure shows the push-pull up-down mode with setting the output on the period event, OUTROM[1:0] = 10.

Figure9-34 Up-Down Mode With Output Set on Period Event, OUTROM[1:0]=10



The following figure shows how the repeat counter decrements in the up-down counting mode.

Figure9-35 Repetition Counter Behavior in Up-Down Counting Mode



The operation of the dual DAC trigger is consistent with the up-counting mode.

The handling of event blanking and windowing is different, aiming to blank or window within a programmable time during the output pulse. The EXEVxFLT[3:0] code depends on the UPDOWNM bit setting, as described in the table below. Whenever the roll-over event is used for blanking or windowing, the programming of ROM[1:0] applies to define when it is generated.

Table9-15 EXEVxFLT[3:0] Codes Depending on UPDOWNM Bit Setting

EXEVxFLT [3:0]	Up-counting Mode (UPDOWNM = 0)	Up/down-counting Mode (UPDOWNM = 1)
10	Blanking from counter reset/roll-over to compare 2	Blanking from compare 1 to compare 2, only during the up-counting phase
100	Blanking from counter reset/roll-over to compare 4	Blanking from compare 3 to compare 4, only during the up-counting phase
1101	Windowing from counter reset/roll-over to compare 2	Windowing from compare 2 to compare 3, only during the up-counting phase
1110	Windowing from counter reset/roll-over to compare 3	Windowing from compare 2 to compare 3, only during the down-counting phase
1111	Windowing from another timing unit: TIMWIN source (refer to**Table** 9-19 for details)	Windowing from compare 2 during the up-counting phase to compare 3 during the down-counting phase

9.3.7 Set/Reset Event Priority and Narrow Pulse Management

This section describes how the output waveform is generated when multiple set and/or reset requests occur within 3 consecutive t_{SHRTIM} cycles.

Scenario 1: Clock prescaler CKPSC[2:0] < 5

In each t_{SHRTIM} cycle, a four-step arbitration process is executed:

1. For each active event, determine the required output transition (set, reset, or toggle).
2. For multiple set event (or reset event) occur in the same t_{SHRTIM} cycle, the low-resolution events (software, external events and synchronization events, update event) have the higher priority:

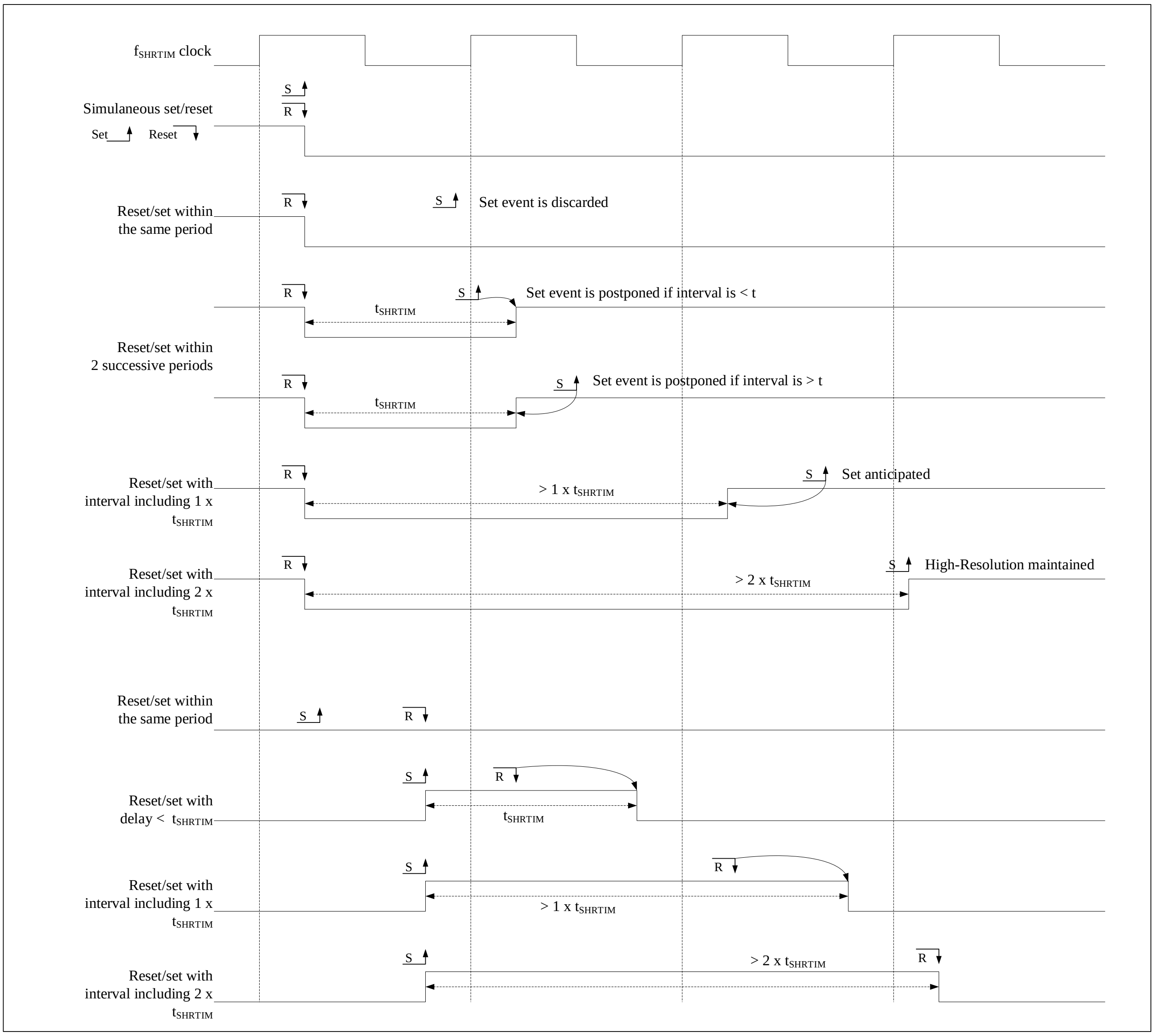
   Software → External events and synchronization events → Update event → Master Timer events/Self-Timer events/Other Timer events. (Priority from high to low)
3. For multiple set event (or reset event) occur in the same t_{SHRTIM} cycle, the high-resolution events (Master Timer events/Self-Timer events/Other Timer events) have the lower priority than the low-resolution events. Predefined arbitration is performed between the high-resolution active events:
   a) If events come from different timer units, the lower high-resolution portion between CMPx and PRD has a higher priority.
   b) If events come from the same timer unit, the priority from high to low is CMP4 → CMP3 → CMP2 → CMP1 → PRD.
4. In low-resolution events and winning predefined arbitration events, arbitration based on high-resolution delay is executed with reset having the highest priority.

When set and reset requests occur simultaneously from two different sources, the reset action has the highest priority. If the interval between set and reset requests is less than 2 t_{SHRTIM} period, their behavior depends on the time interval and alignment with the f_{SHRTIM} clock, as shown in the figure below.

Note: Reset events have the highest priority.

Note: The CMPx/PRD value in this section refer to the valid CMPx/PRD value after adjustment with the period phase or timer reset phase (excluding those with special instructions), not the original CMPx/PRD value in the register written by software.

Figure9-36 Short Distance Set/Reset Management for Narrow Pulse Generation


fSHRTIM clock
Simulaneous set/reset
Set
Reset
S
R
Reset/set within the same period
Set event is discarded
Set event is postponed if interval is < t
tSHRTIM
Reset/set within 2 successive periods
Set event is postponed if interval is > t
Reset/set with interval including 1 x tSHRTIM
> 1 x tSHRTIM
Set anticipated
Reset/set with interval including 2 x tSHRTIM
> 2 x tSHRTIM
High-Resolution maintained
Reset/set within the same period
Reset/set with delay < tSHRTIM
Reset/set with interval including 1 x tSHRTIM
Reset/set with interval including 2 x tSHRTIM

If set and reset events are generated within the same t_{SHRTIM} cycle, the reset event has the highest priority, and the set event is ignored.

If set and reset events are generated with an interval below t_{SHRTIM}, across 2 periods, a pulse of 1 t_{SHRTIM} period is generated.

If set and reset events are generated with an interval below 2 t_{SHRTIM} periods, a pulse of 2 t_{SHRTIM} periods is generated.

If set and reset events are generated with an interval between 2 and 3 t_{SHRTIM} periods, the high resolution is available if the interval exceeds 2 complete t_{SHRTIM} periods.

If set and reset events are generated with an interval above 3 t_{SHRTIM} periods, high resolution is always available.

Simultaneous set requests/simultaneous reset requests

When multiple sources are selected as set events, arbitration is performed if set requests occur within the same f_{SHRTIM} clock cycle. For multiple requests from adjacent timers (TIMEVNT1..9), the first request is considered. Arbitration is done in two steps, depending on:

1. The source (CMP4 → CMP3 → CMP2 → CMP1),
2. The delay.

If multiple requests from the master timer occur within the same f_{SHRTIM} clock period, a predefined arbitration will be applied (from highest to lowest priority):

MCMPDAT4 → MCMPDAT3 → MCMPDAT2 → MCMPDAT1 → MPRD

Note: It is recommended to avoid generating multiple set (reset) requests from the master timer to a given timer within intervals less than 3x t_{SHRTIM} to maintain high resolution.

When multiple requests internal to the timer occur within the same f_{SHRTIM} clock period, a predefined arbitration will be applied, and requests will be processed based on the following priorities, and only a single request will be considered, regardless of the actual high-resolution settings (from highest to lowest):

CMP4 → CMP3 → CMP2 → CMP1 → PRD

Note: In fact, this is very important when multiple compare events can be generated simultaneously or when auto-delayedcompare 2 and compare 4 are used simultaneously (i.e. because it is related to external events, effective set/reset cannot be determined in advance). In this case, the highest priority signal must be assigned to the CMP4 event.

Note: For the priority judgment of multiple requests internal to the timer occur within the same f_{SHRTIM} clock period, the CMPx/PRD value refer to the original CMPx/PRD value in the register written by software.

Finally, the highest priority is assigned to low-resolution events: EXTEVNT1..10, RESYNC (if SYNCRST or SYNCSTRT is set from a SYNC event or from a software reset), update, and software set (SWT). The high-resolution delay of the update event is set to high-resolution part of the PRD. In summary, in the case of event density (events occurring within the same f_{SHRTIM} clock period), the effective set (reset) events is arbitrated between:

- Any TIMEVNT1..9 event
- Any source from the master timer
- A single source from the timer
- The "Low-resolution events"

The same arbitration principles apply to simultaneous reset requests. In this case, the reset request has the highest priority.

Scenario 2: Clock prescaler CKPSC[2:0] ≥ 5

When high resolution is not effective, narrow pulse management is simplified.

A set or reset events occurring within the prescaler clock cycle is delayed until the next active edge of the prescaler clock (similar to counter reset), even though arbitration is still performed in each t_{SHRTIM} cycle. If a reset event closely follows a set event within the same prescaler clock cycle, the last event to occur will be considered.

9.3.8 External Events Global Conditioning

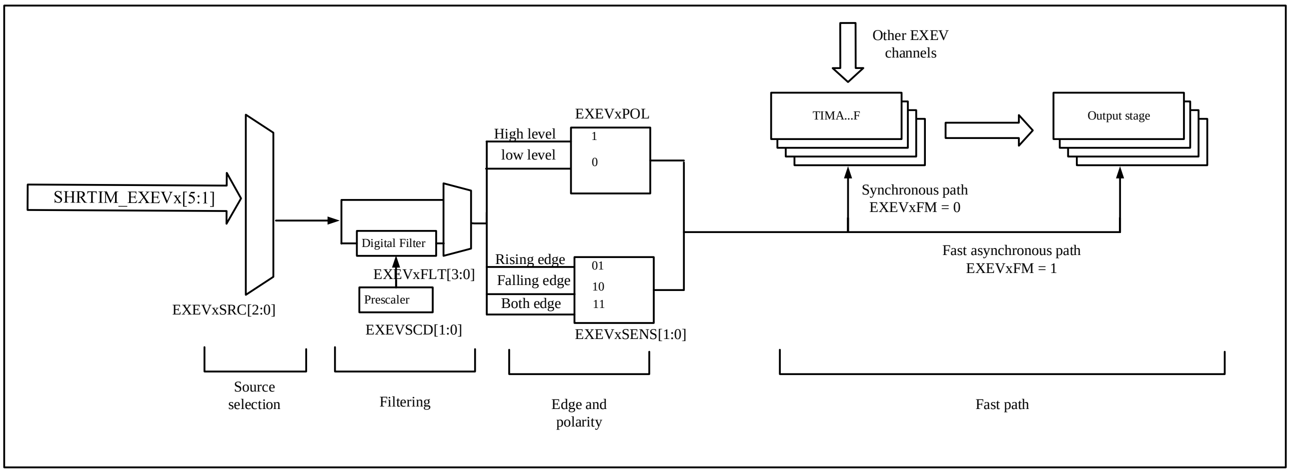
The SHRTIM timer can handle events that are not generated within the timer, known as "external events." The external events can come from various on-chip or off-chip sources, such as:

- Internal comparators
- Digital input pins (typically connected to off-chip comparators and zero-crossing detectors)
- Other on-chip events from external peripherals (such as the analog watchdog of the ADC and trigger outputs of general-purpose timers)

The external event conditioning circuit allows selecting the signal source for a given channel (using a 5:1 multiplexer) and can convert the signal source into information that can be processed by the crossbar unit (for example, triggering output reset by detecting a falling edge on the external event channel).

Up to 10 external event channels can be conditioned, and these external event channels can be used simultaneously for any of the 6 timers. Since this conditioning typically depends on external components (such as zero-crossing detectors) and environmental conditions (filter settings are usually related to application noise levels and signature), it is universal for all timers. Figure9-37 shows an overview of the conditioning logic for a single channel.

Figure9-37 External Event Conditioning Overview (1 Channel Represented)



9.3.8.1Latency to external events

The 10 external events are initialized through the SHRTIM_EXEVCTRL1 and SHRTIM_EXEVCTRL2 registers:

- Use the EXEVxSRC[1:0] bits to select up to 5 sources
- Use the EXEVxSNS[1:0] bits to choose level or edge sensitivity (rising edge, falling edge, or both edges)
- If level sensitivity is selected, use the EXEVxPOL bit to choose the polarity
- Use the EXEVxFM bit for low-latency mode for external events 1 to 10 (refer to Section 9.3.8.1 Latency to external events)

Note: Even if the EXEVSNS bit is reset (selecting level sensitivity), external events used as triggers for reset, capture, burst mode, ADC triggers, and delayed protection are edge-sensitive: if POL = 0, the trigger is active on the rising edge of the external event; if POL = 1, the trigger is active on the falling edge of the external event.

Once the counter is disabled (TxCNTEN bit reset), external events are immediately discarded to prevent any output state changes and counter resets, except for external events used as ADC trigger signals.

Additionally, the EXEVxF[3:0] bits in the SHRTIM_EXEVCTRL3 and SHRTIM_EXEVCTRL4 registers can be used to enable digital noise filtering for external events 1 to 10.

The digital filter consists of a counter that requires N valid samples to confirm an output transition. If the input value changes before the counter reaches the N value, the counter is reset, and the transition is discarded (considered a spurious event). If the counter reaches N, the transition is considered valid and is transmitted as the correct external event. Therefore, the digital filter adds a latency to the external events undergoing filtering, with the delay depending on the sampling clock and the filter length (expected number of valid samples).

The sampling clock is either the f_{SHRTIM} clock or a specific clock f_{EXEVS} derived from f_{SHRTIM} prescaling, defined by the EXEVSCD[1:0] bits in the SHRTIM_EXEVCTRL4 register.

The table below summarizes the available sources and characteristics associated with the 10 external event channels.

Table9-16 External Event Mapping and Associated Characteristics

External event channel	Fast mode(1)	Digital filter(1)	Balanced fault timer A,B,C	Balanced fault timer D,E,F
shrtim_exev1[5:1]	Yes	Yes	-	-
shrtim_exev2[5:1]	Yes	Yes	-	-
shrtim_exev3[5:1]	Yes	Yes	-	-
shrtim_exev4[5:1]	Yes	Yes	-	-
shrtim_exev5[5:1]	Yes	Yes	-	-
shrtim_exev6[5:1]	Yes	Yes	Yes	-
shrtim_exev7[5:1]	Yes	Yes	Yes	-
shrtim_exev8[5:1]	Yes	Yes	-	Yes
shrtim_exev9[5:1]	Yes	Yes	-	Yes
shrtim_exev10[5:1]	Yes	Yes	-	-

1. Fast mode and digital filtering cannot be used simultaneously for the same external event.

9.3.8.1 Latency to external events

The external event conditioning allows adjusting the external event processing time (and associated latency) based on performance expectations:

- A regular operating mode, where the external events is resampled with the clock before acting on the output crossbar. This process adds some latency but provides access to all crossbar functionalities. It enables the generation of high-resolution pulses triggered by external events.
- A fast operating mode, where the latency between external events and the action on the output is minimized. This mode is convenient for ultra-fast over-current protection functions, for instance.

The EXEVxFM bit in the SHRTIM_EXEVCTRL3 and SHRTIM_EXEVCTRL4 registers can be used to define the operating mode for channels 1 to 5. This affects the latency and jitter present in the output pulses, as detailed in the table below.

Table9-17 Output Set/Reset Latency and Jitter versus External Event Operating Mode

EXEVxFM	Response Time Latency	Response Time Jitter	Jitter on the Output Pulse (Counter Reset via External Event)
0	4 to 5 cycles of f_{SHRTIM} clock	1 cycles of f_{SHRTIM} clock	No jitter, pulse width maintained with high-resolution
1	Minimal latency (depends whether the comparator or digital input is used)	Minimal jitter	1 cycle of f_{SHRTIM} clock jitter, pulse width resolution down to t_{SHRTIM}

When level sensitivity is set (EXEVxSNS[1:0] = 00), the EXEVxFM mode can be used; it cannot be used for edge sensitivity.

The event filtering (blanking and windowing, EXEVxFLT[3:0] != 0000, refer to Section 9.3.9) can be applied to external events. In this case, the EXEVxLATCHx bit must be reset: the postponed mode is not supported, and window timeout functionality is not supported.

Note: once the relevant EXEVxFM bit is set to 1, external event configuration (source and polarity) must not be modified.

A fast external events cannot be used for toggling outputs: they must be enabled in either the SHRTIM_TxSETy or SHRTIM_TxRSTy registers but not both simultaneously.

If a set event and a reset event (from 2 separate fast external events) occur simultaneously, the reset event has the highest priority in the crossbar, and the output becomes inactive.

If the EXEVxFM bit is set to 1, the output cannot be changed for 11 f_{SHRTIM} clock cycles following the occurrence of the external event.

Figure 9-38 and Figure9-39 provide examples of the actual response times of external events when output set/reset and counter reset.

Figure 9-38 Latency to External Events Falling Edge (Counter Reset and Output Set)Figure

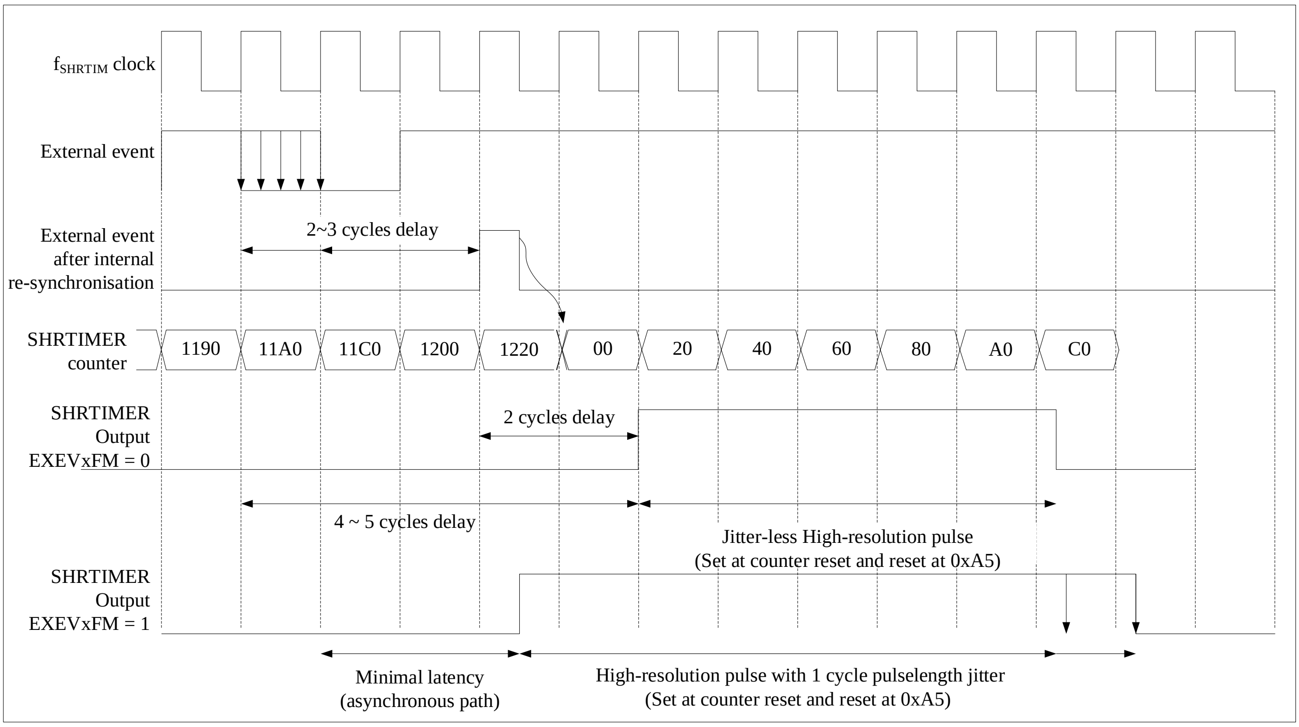
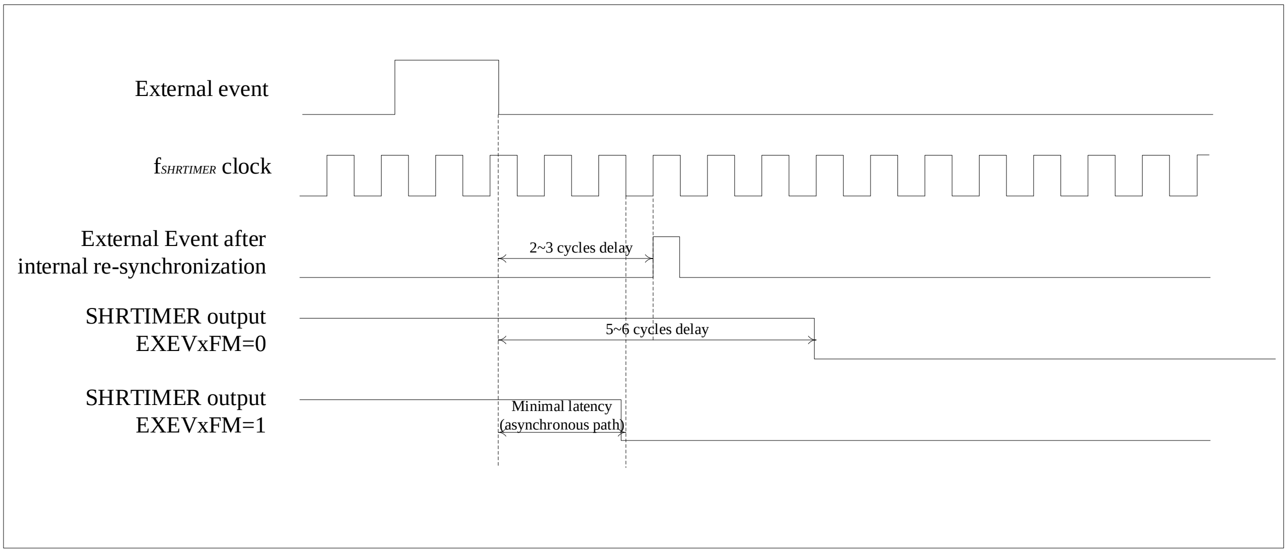


Figure9-39 Latency to External Events (Output Reset on External Event)



9.3.9 External Event Filtering in Timing Units

After conditioning, the 10 external events are available for all timing units.

These events can be used directly and take effect immediately as soon as the timing unit counter is enabled (TxCNTEN bit set to 1).

Additionally, these events can be filtered to have an action limited in time, typically related to the counting period. Two operations can be performed:

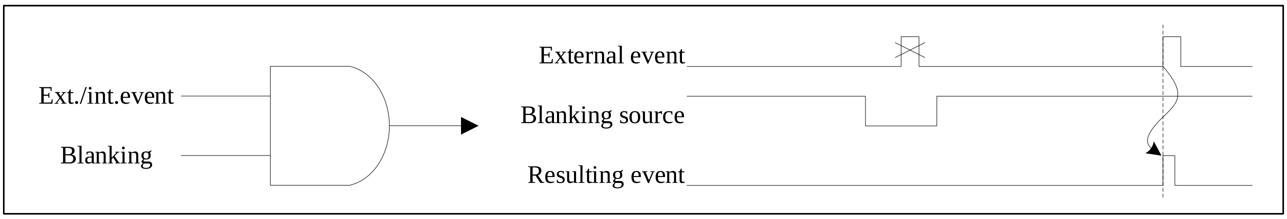
- Blanking mode, which mask external events during a defined time period
- Windowing mode, which enables external events only during a defined time period

These modes are enabled using the SHRTIM_EXEVxFLT[3:0] bits in the SHRTIM_TxEXEVFLT1 and SHRTIM_TxEXEVFLT2 registers. Each of the 6 Timer A..F timing units has programmable filter settings for these 10 external events.

9.3.9.1 Blanking mode

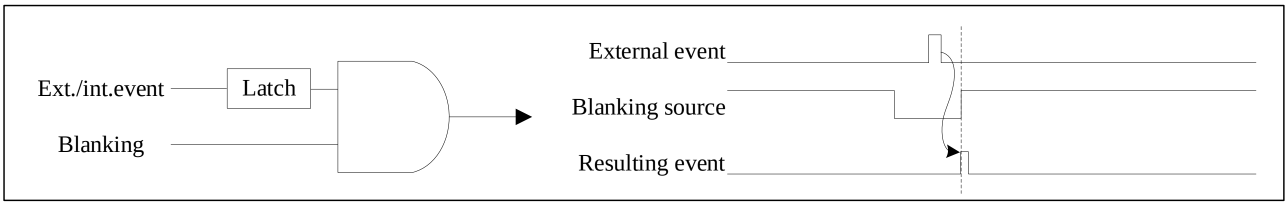
In event blanking mode (refer to the diagram below), if an external event occurs during a given blanking period, it will be ignored. This is convenient, for instance, to avoid a current limit to trip on switching noise at the beginning of a PWM period. This mode is active when the value of the EXEVxFLT[3:0] bit field is in the range of 0001 to 1100.

Figure9-40 Event Blanking Mode



In event postpone mode, the external event is not immediately considered but is memorized (latched) and generated as soon as the blanking period is completed, as shown in Figure9-41. This mode can be enabled by setting the EXEVxLATCH bit to 1 in the SHRTIM_TxEXEVFLT1 and SHRTIM_TxEXEVFLT2 registers.

Figure9-41 Event Postpone Mode



The blanking signal comes from several sources:

- The timer itself: The blanking lasts from the counter reset to the compare match (for compare 1 to compare 4, EXEVxFLT[3:0] = 0001 to 0100).
- From other timing units (EXEVxFLT[3:0] = 0101 to 1100): The blanking lasts from the selected timing unit counter reset to one of its compare matches, or it can be fully programmed as a waveform on the Tx2 output. In this case, the event is masked as long as the Tx2 signal is inactive (no need to enable the output, and the signal is acquired before the output stage).

The configuration of EXEVxFLT[3:0] from 0101 to 1100 is referred to as TIMFLTR1 to TIMFLTR8 in the bit description, with different meanings for each timing unit. 8 options are available for each timer: CMPx refers to blanking from counter reset to compare match, and Tx2 refers to the TIMx output 2 waveform defined by the SHRTIM_TxSET2 and SHRTIM_TxRST2 registers. For example, Timer B (TIMFLTR6) is the output 2 waveform of Timer C.

Table9-18 Filtering Signals Mapping per Timer

Source		Timer A				Timer B				Timer C				Timer D				Timer E				Timer F			
		CMP1	CMP2	CMP4	TA2	CMP1	CMP2	CMP4	TB2	CMP1	CMP2	CMP4	TC2	CMP1	CMP2	CMP4	TD2	CMP1	CMP2	CMP4	TE2	CMP1	CMP2	CMP4	TF2
Destination	Timer A	-				1	-	2	3	4	-	5	-	7	-	-	-	-	8	-	-	6	-	-	-
	Timer B	1	-	2	3	-				4	5	-	-	-	7	-	-	8	-	-	-	-	6	-	-
	Timer C	-	1	-	-	2	-	3	-	-				5	-	6	7	-	-	8	-	4	-	-	-
	Timer D	1	-	-	-	-	2	-	-	3	4	-	5	-				6	-	7	-	-	-	8	-
	Timer E	-	1	-	-	2	-	-	-	3	-	-	-	6	-	7	8	-				-	-	4	5
	Timer F	-	-	1	-	-	2	-	-	-	-	3	-	-	4	5	-	6	-	7	8	-			

The following diagram illustrates and provides examples of event blanking for all edge-sensitive and level-sensitive external events in regular mode and postponed mode.

Figure9-42 External Trigger Blanking with Edge-Sensitive Trigger

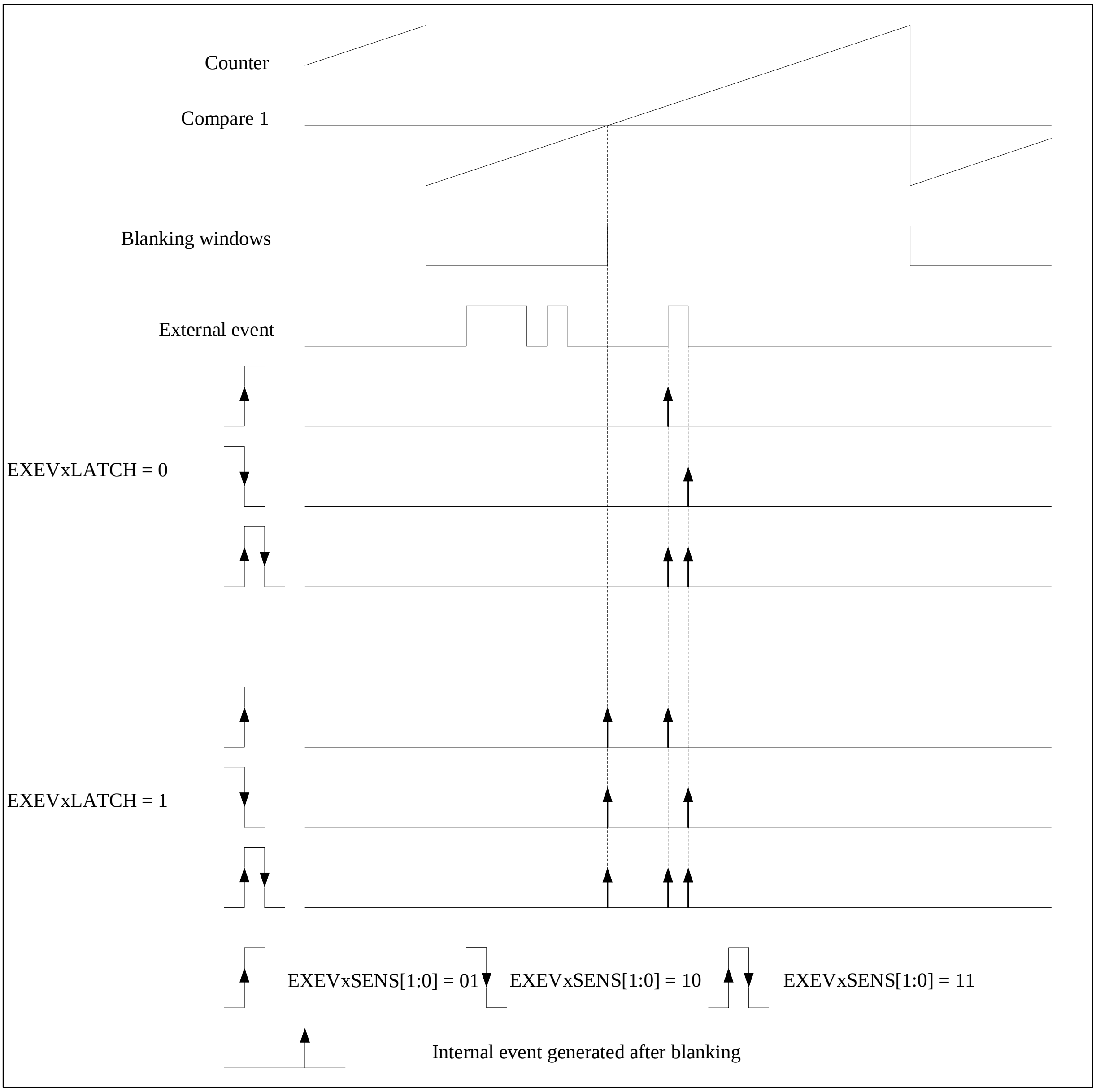
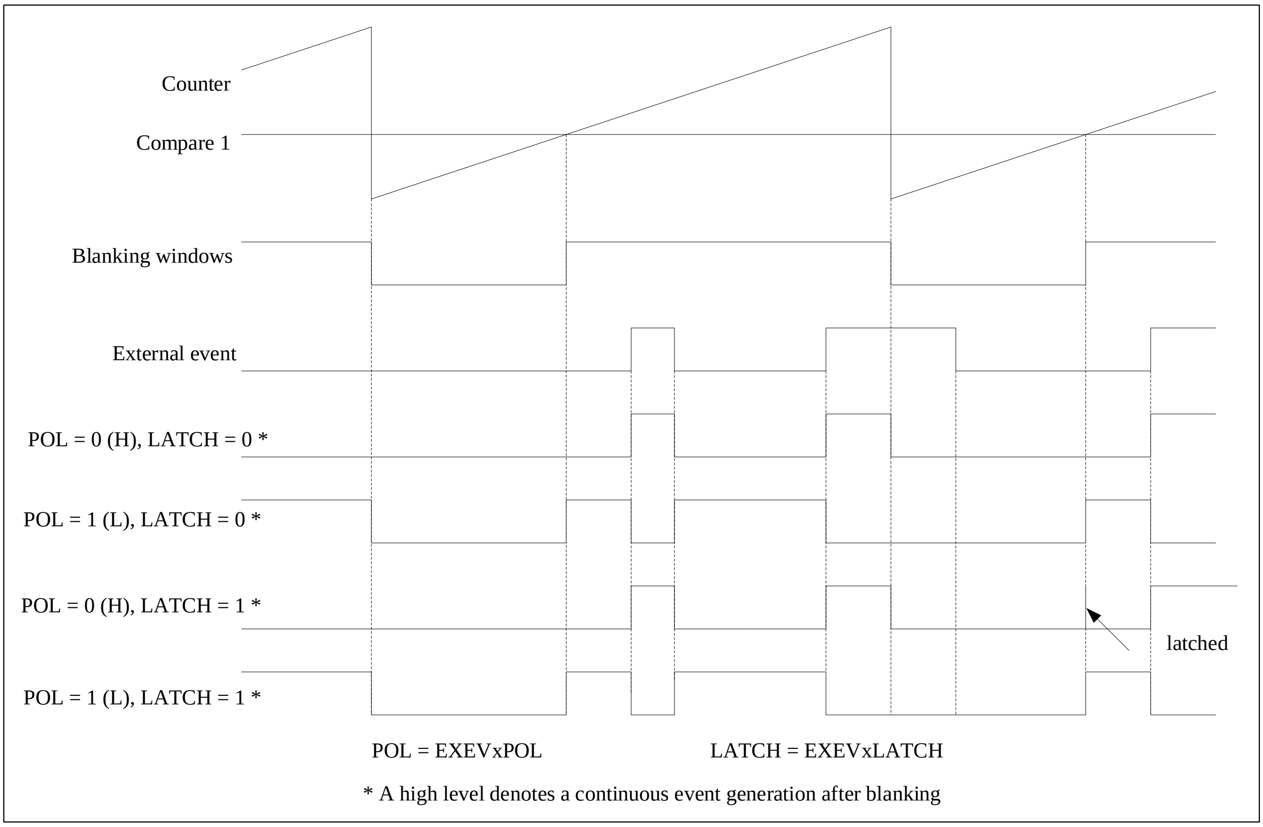


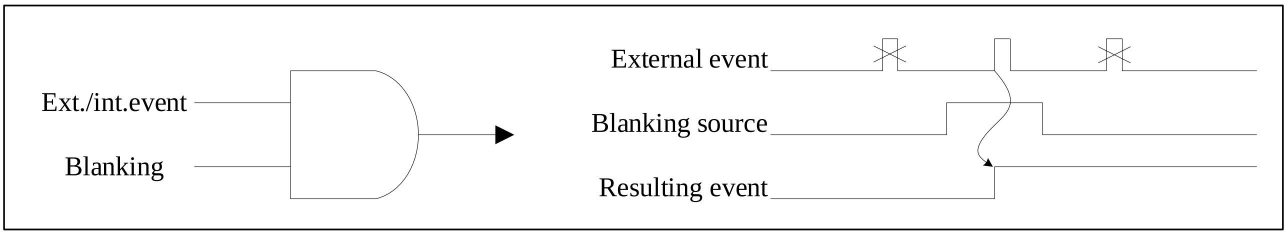
Figure9-43 External Trigger Blanking, Level Sensitive Triggering



9.3.9.2 Windowing mode

In event windowing mode, the event is only considered if it occurs within a given time window; otherwise, it is ignored. This mode is active when the value of EXEVxFLT[3:0] is in the range of 1101 to 1111.

Figure9-44 Event Windowing Mode



The EXEVxLATCH in the EXEVFLT1 and EXEVFLT2 registers allows to latch the signal, when this bit is set to 1: in this case, an event is accepted if it occurs during the window but is delayed at the end of the window,.

- If the EXEVxLATCH bit is reset, and the signal occurs during the window, it is passed through directly.
- If the EXEVxLATCH bit is reset and no signal occurs, a timeout event is generated at the end of the window.

The windowing mode can be used to filter synchronization signals. When the expected synchronization event is lacking (such as during a converter star-tup), the timeout feature can be used to force a default synchronization event.

Each external event windowing has 3 sources, coded as follows:

- 1101 and 1110: The windowing lasts from the counter reset to the compare match in up-counting mode (compare

2 and compare 3, respectively).

- 1111: The windowing is related to another timing unit, and lasts from its counter reset to its compare 2 match. The source is described as TIMWIN in the bit description and is given in **Table** 9-19. For example, the external events in timer B can be filtered by a window starting from the counter reset of timer A to the compare 2 of timer A.

Table 9-19 Windowing Signals Mapping per Timer (EXEVxFLT [3:0] = 1111)

Destination	**Timer A**	**Timer B**	**Timer C**	**Timer D**	**Timer E**	**Timer F**
TIMWIN (source)	Timer B CMP2	Timer A CMP2	Timer D CMP2	Timer C CMP2	Timer F CMP2	Timer E CMP2

Note: If the external event is programmed in fast mode, the timeout event generation is not supported.

The following diagram explains how the events are generated based on the setting of the EXEVxLATCH bit to achieve various edge-triggered and level-triggered operations. A specific explanation is provided for timeout events to facilitate understanding.

Figure9-45 External Trigger Windowing with Edge-Sensitive Trigger

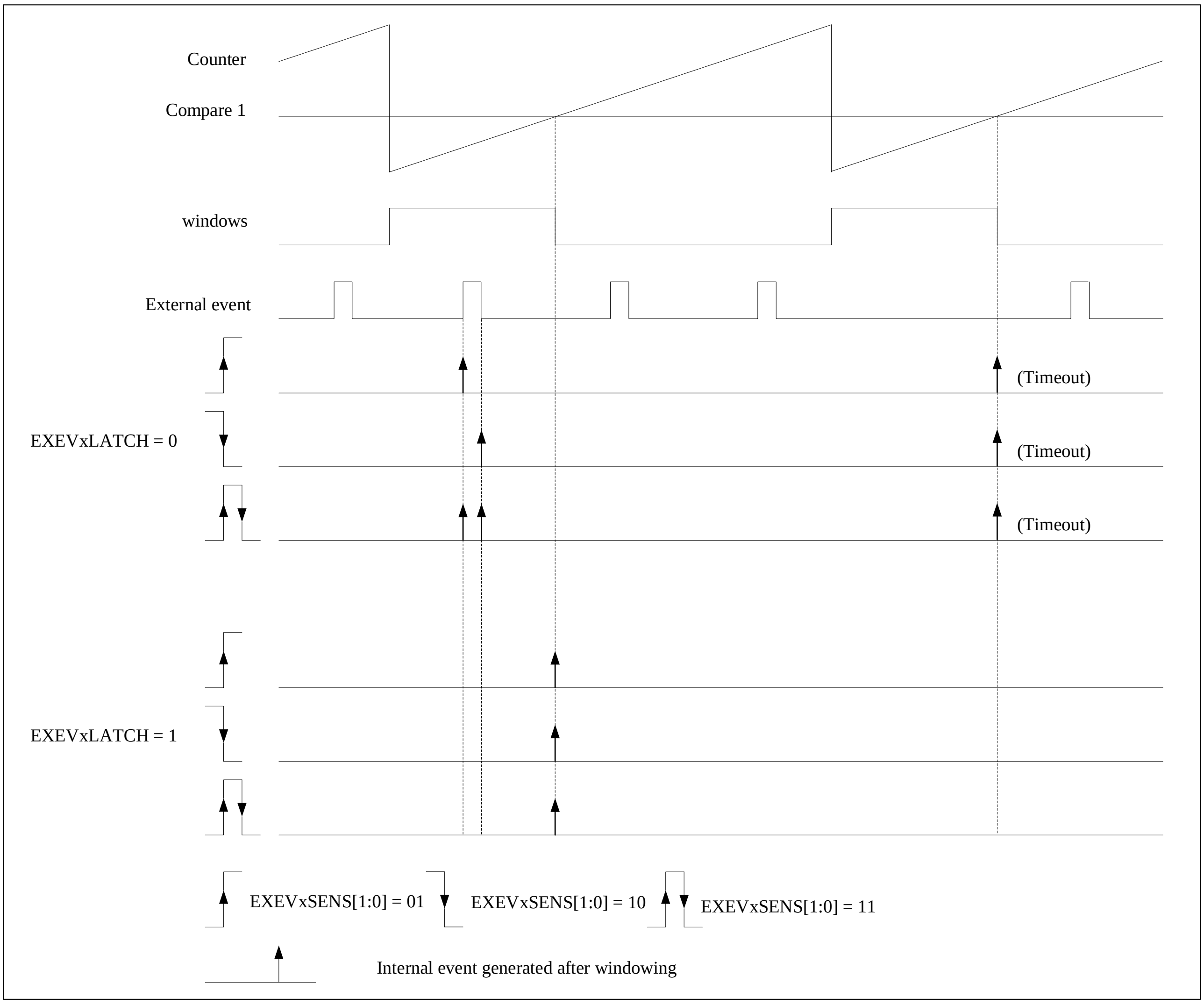
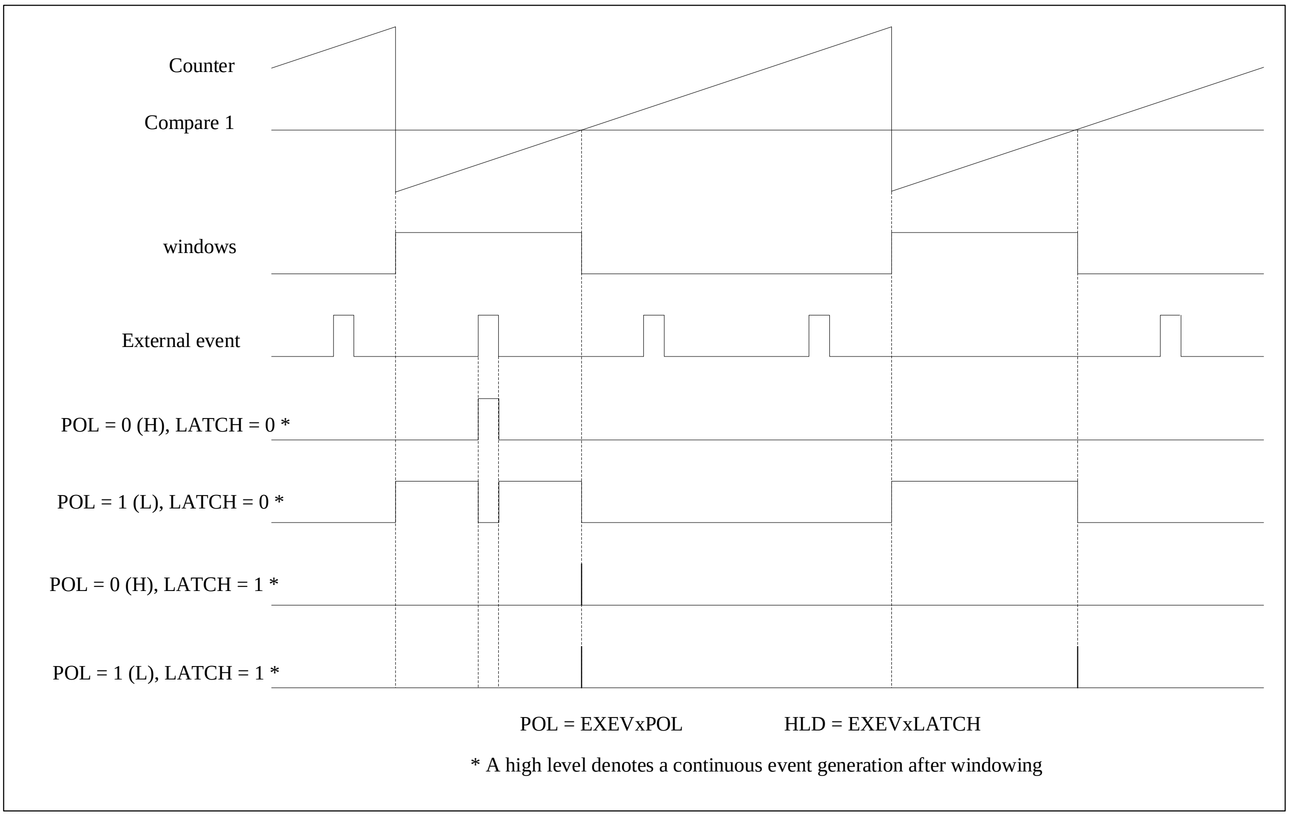


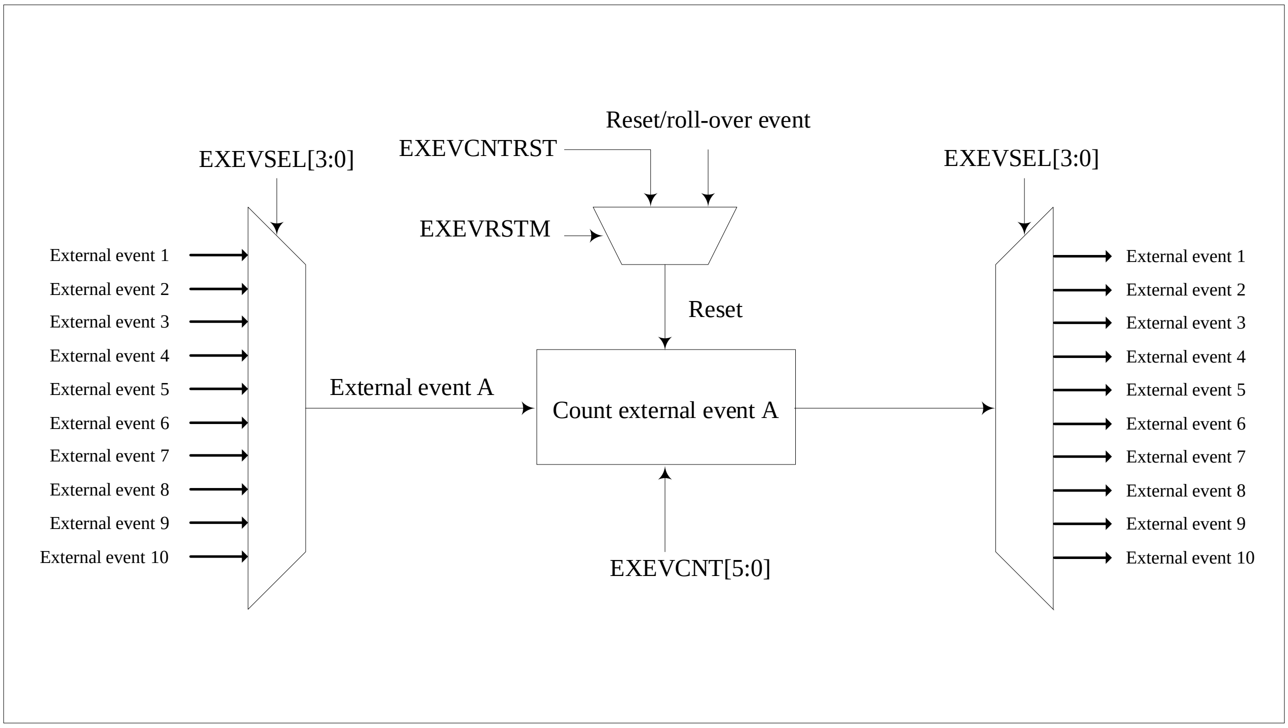
Figure9-46 External Trigger Windowing, Level Sensitive Triggering



9.3.9.3 External event counter

Each timing unit also includes an external event counter that follows the filtering unit, typically for valley skipping implementation. The circuit allows to filter any of the 10 external events filtered, as shown in the diagram below.

Figure9-47 External Event Counter – Channel A

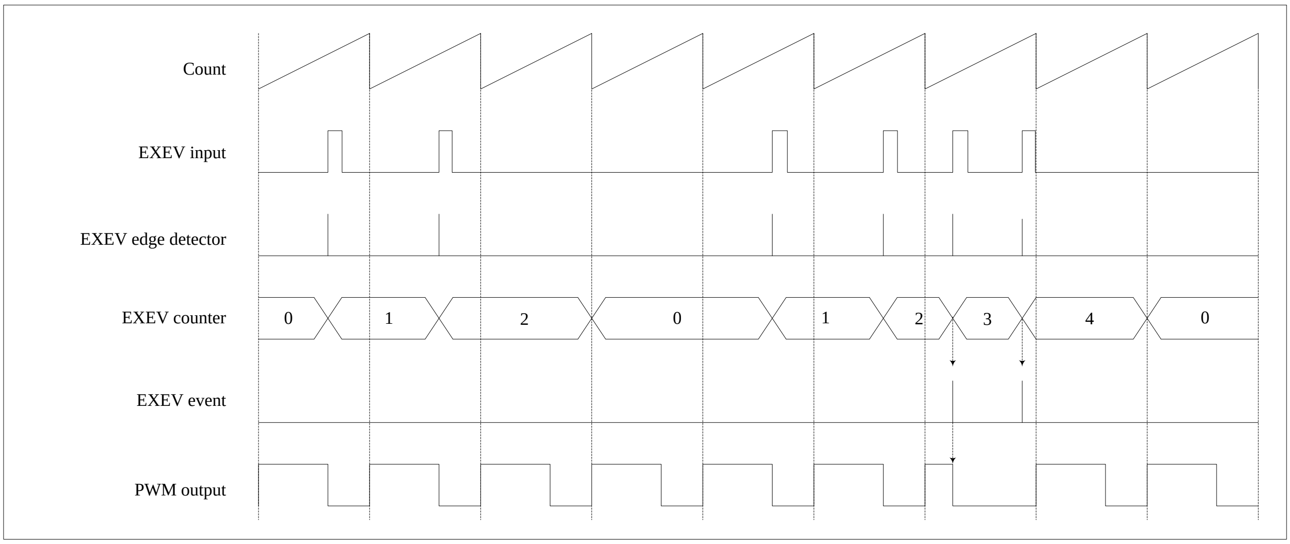


The counter is enabled by setting the EXEVCNTEN bit in the SHRTIM_TAEXEVFLT3 register. This mode is only valid for external events that are edge-sensitive (EXEVxSNS[1:0] bit = 01, 10, or 11). The external events is propagated to the timer only when the number of active edges is greater than or equal to the value programmed in (EXEVCNT[5:0] + 1). There are two available operation modes:

- When the EXEVRSTM bit is reset, the external event counter is reset on each reset/rollover event: the external event is only valid when they occur multiple times within a given PWM period.
- When the EXEVRSTM bit is set, the external event counter is reset only when no events occurred during the last PWM period. This is an accumulation mode where events must occur at least once during multiple PWM periods, as shown in the diagram below.

The external event counter must be enabled after having programmed the counter value (the EXEVCNTEN bit must be set after having written the EXEVCNT[5:0] bit). Once the counter is enabled, the EXEVCNT[5:0] bits can be changed on-the-fly at any time. The new value will be considered based on the programming of the EXEVRSTM bit at the next reset/rollover event or after a software reset (set the EXEVCNTRST bit). Once the EXEVCNTEN bit is set, the EXEVSEL[3:0] bits cannot be modified.

Figure9-48 External Event Counter Accumulation Mode (EXEVRSTM = 1, EXEVCNT = 2)



9.3.10 Delayed Protection

The SHRTIM features specific protection schemes, typically for resonant converters when it is necessary to shut down the PWM outputs in a delayed manner, either once the active pulse is completed or once a push-pull period is completed. These functions are enabled by the DPEN bit in the SHRTIM_TxOUT register and are using specific external event channels.

9.3.10.1 Delayed idle

In this mode, the active pulse is completed before the protection is activated. The selected external event will causes the output to enter idle mode at the end of the active pulse (defined by the output reset event in SHRTIM_TxRST1 or SHRTIM_TxRST2).

Once the protection is triggered, it will remain in idle mode permanently, but the counter continue to run until the output is re-enabled. The Tx1OEN and Tx2OEN bits are not affected by the delayed idle mode entry. To exit the delayed idle mode and resume operation, the Tx1OEN and Tx2OEN bits need to be rewritten as 1. The output state changes on the first transition to an active state following the output enable command.

Note: The delayed idle mode cannot be exited immediately after having been entered, before the active pulse is completed: it is mandatory to make sure that the outputs are in idle state before resuming the run mode. This can be done by waiting until the next period or by polling the O1BCKUP and/or O2BCKUP status bits in the TxINTSTS register.

The delayed idle mode can be applied to a single output (DP[2:0] = x00 or x01) or to two outputs (DP[2:0] = x10).

Upon entering the delayed idle mode, An interrupt or a DMA request can be generated in response to a Delayed Idle mode entry. When an external event occurs, the DP flag in SHRTIM_TxINTSTS will be set to 1 immediately, independently from the end of the active pulse on output.

After triggering the delayed idle mode, the output status can be determined by O1DIPSTS and O2DIPSTS in SHRTIM_TxINTSTS. These status bits will be updated even if the delayed idle mode is applied to a single output. If the push-pull mode is enabled, the IPPSTS flag in SHRTIM_TxINTSTS will indicate during which period the delayed protection request occurred.

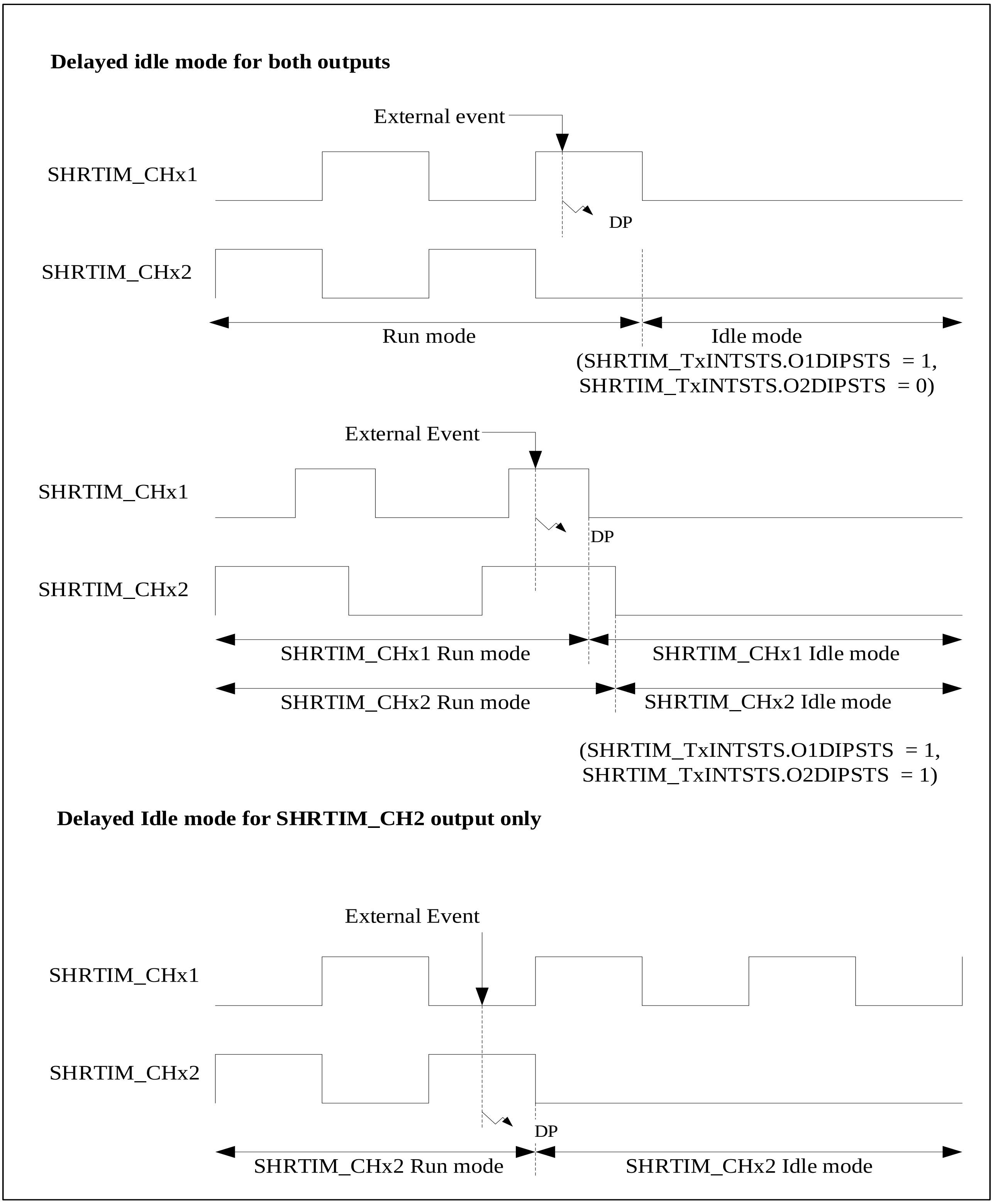
This mode can be used whatever the operating mode of the timer (normal, push-pull, dead time). This mode supports only 2 external events:

- shrtim_exev6 and shrtim_exev7 (for timer A, B and C)
- shrtim_exev8 and shrtim_exev9 (for timer D, E and F)

A software event can trigger delayed protection by configuring SFTDPxy in SHRTIM_SFTDP to select which output channel is triggered.

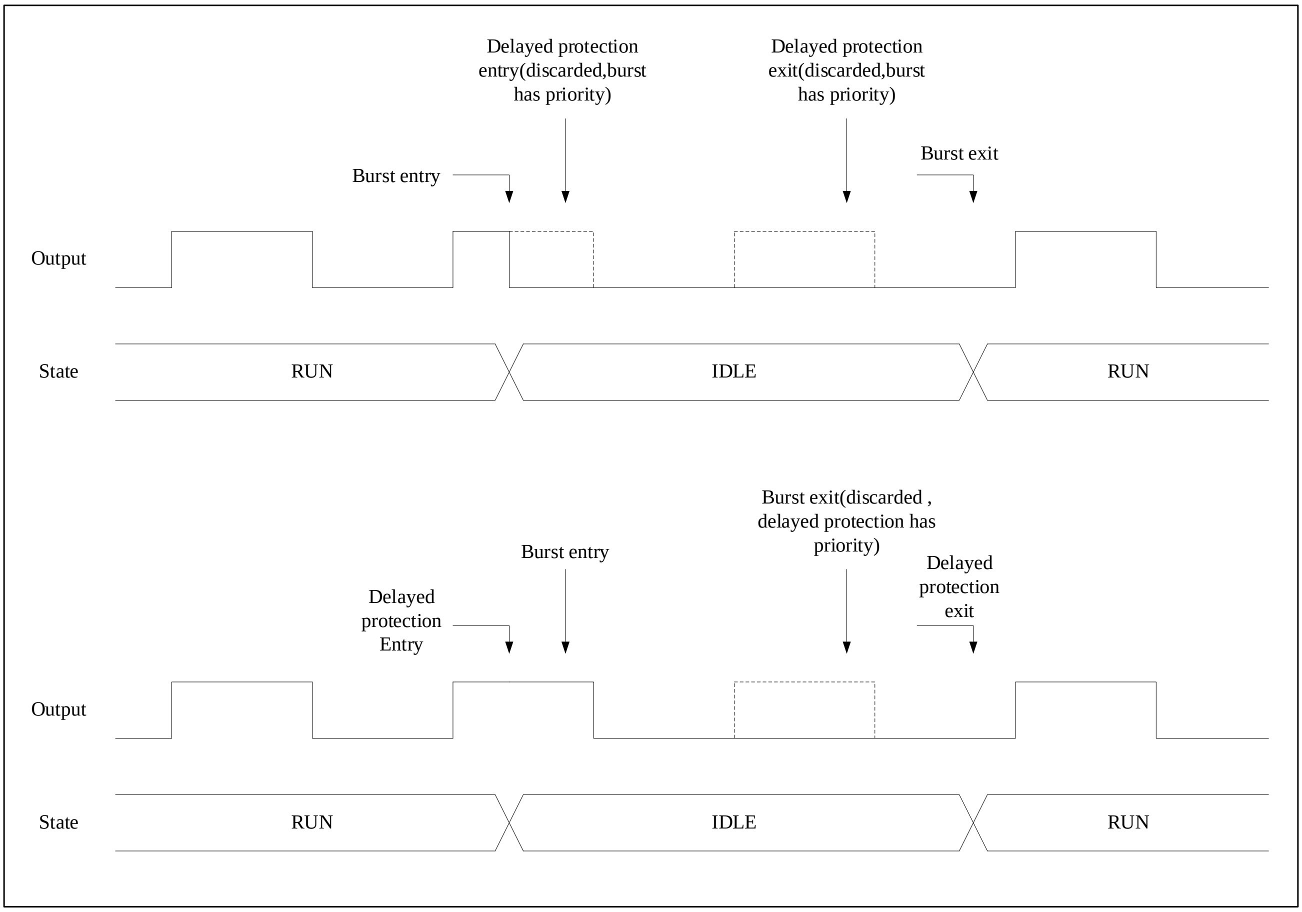
The delayed protection mode can only be triggered when the counter is enabled (TxCNTEN set to 1). Even if the TxCNTEN bit has been reset, the delayed protection mode remains active until TxyOEN is set to 1.

Figure9-49 Delayed Idle Mode Entry



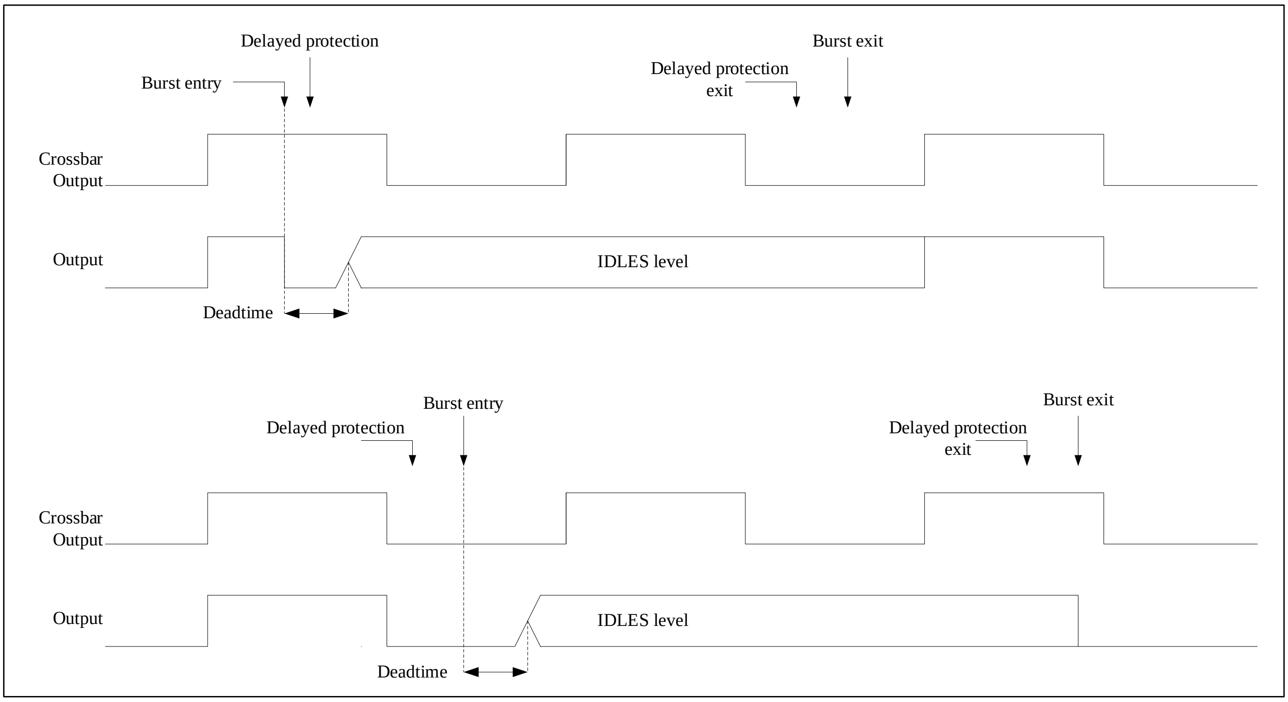
The delayed idle mode has a higher priority than burst mode: once the delayed idle protection is triggered, any burst mode exit request will be discarded. Conversely, if the delayed protection is exited while burst mode is active, the burst mode will resume normally, and the output will maintain in the idle state until exiting burst mode. The following diagram provides an overview of these different scenarios.

Figure9-50 Burst Mode and Delayed Protection Priorities (DIDL = 0)


Delayed protection entry(discarded,burst has priority)
Delayed protection exit(discarded,burst has priority)
Burst entry
Burst exit
Output
State
RUN
IDLE
RUN
Burst exit(discarded , delayed protection has priority)
Burst entry
Delayed protection Entry
Delayed protection exit
Output
State
RUN
IDLE
RUN

If the delayed burst mode entry is enabled (DIDL set to 1), the same priority will be applied, as shown below.

Figure9-51 Burst Mode and Delayed Protection Priorities (DIDL = 1)



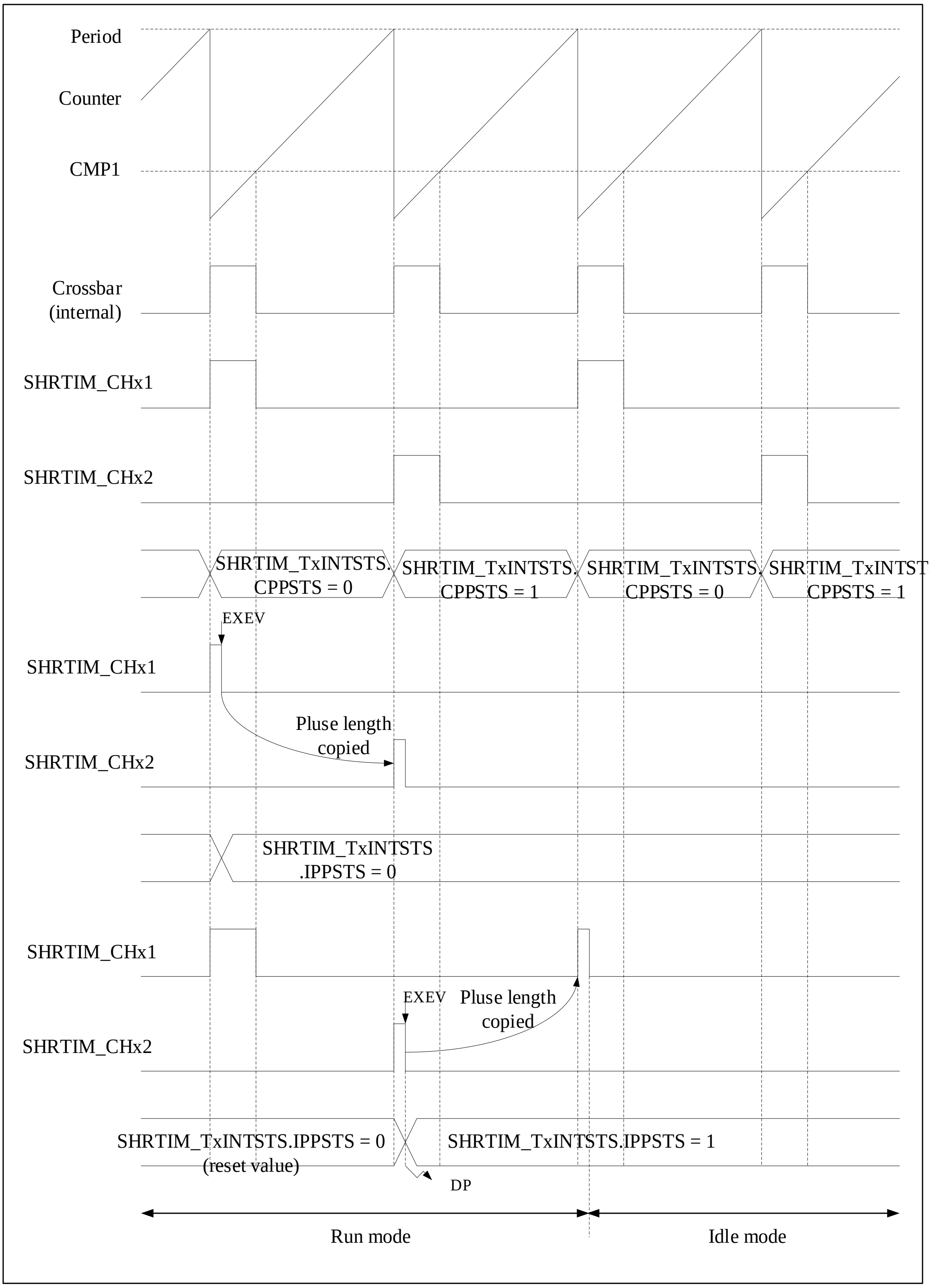
9.3.10.2 Balanced idle

This feature is only available in push-pull mode. The balanced idle allows to have a balanced pulsewidth on the two outputs when one of the active pulse is shortened due to a protection. The pulsewidth, terminated earlier than the programmed, will be copide on the alternate output channel, and then both channels will enter idle state until normal operation is resumed by software. This mode is enabled by writing x11 to the DP[2:0] field in SHRTIM_TxOUT.

This mode supports only 2 external events:

- shrtim_exev6 and shrtim_exev7 (for timer A, B and C)
- shrtim_exev8 and shrtim_exev9 (for timer D, E and F)

Figure9-52 Balanced Idle Protection Example



When balanced idle mode is enabled, the selected external event will trigger a capture of the counter value into the compare4 active register (this value is not user-accessible). The push-pull is maintained for one additional period so that the shorten pulse can be repeated: a new output reset event is generated while the regular output set event is maintained.

Subsequently, it will enter idle mode, and the output will take the level defined by the IDLESx bits in the SHRTIM_TxOUT register. The balanced idle mode entry is indicated by the DP flag, while the IPPSTS flag indicates during which period the external event occurred, which can be used to determine the sequence of the shortened pulses (first SHRTIM_CHA1, then SHRTIM_CHA2, and vice versa).

The timer operation is not interrupted (the counter continues to run).

To enable balanced idle mode, the following initialization steps need to be performed:

- The timer operates in continuous mode (CONT = 1)
- Enable push-pull mode
- The SHRTIM_TxCMP4DAT must be set to 0 and the content transferred into the active register (e.g. by forcing a software update)
- The DELCMP4M[1:0] field must be set to 00 (disable auto-delay mode)
- DP[2:0] = x11 (enable delayed protection)

Note: during balanced idle operation, writing to the SHRTIM_TxCMP4DAT register is not allowed. The CMP4 event is reserved and cannot be used for other purposes.

In balanced idle mode, it is recommended to avoid using multiple external events or software-based reset event causing an output reset. If such an event arrives before a balanced idle request within the same period, it can lead to unbalanced output pulses (where the length of the first pulse is defined by the external event or software reset, and the length of the second pulse is defined by balanced idle mode entry).

The minimum pulsewidth that can be handled in balanced idle mode is 4 f_{SHRTIM} clock periods (when CKPSC[2:0] = 0, 0x80; when CKPSC[2:0] = 1, 0x40; when CKPSC[2:0] = 2, 0x20; ...).

If the capture occurs before the counter has reached this minimum value, the current pulse is extended up to 4 f_{SHRTIM} clock periods before being copied into the secondary output. In any case, the pulsewidths are always balanced.

Tx1OEN and Tx2OEN bits are not affected by the balanced idle entry. To exit from balanced idle and resume the operation, it is necessary to overwrite Tx1OEN and Tx2OEN bits to 1 simultaneously. The output state changes on the first active transition following the output enable.

It is possible to resume operation similarly to the delayed idle entry. For instance, if the external event arrives while output 1 is active (delayed idle effective after output 2 pulse), the re-start sequence can be initiated for output 1 first. To do so, it is necessary to poll CPPSTS bit in the SHRTIM_TxINTSTS register. Using the above example (IPPSTS flag equal to 0), the operation is resumed when CPPSTAT bit is 0.

In order to have a specific re-start sequence, it is possible to poll the CPPSTS to know which output is active first. This allows, for instance, to re-start with the same sequence as the idle entry sequence: if the EXEV arrives during output 1 active, the re-start sequence is initiated when the output 1 is active (CPPSTS = 0).

Note: While a pulse balancing sequence is ongoing, the balanced idle mode should not be disabled. The DPEN bit should only be reset after the CMP4 flag is set to 1, indicating that the sequence is completed.

The balanced idle protection mode can be triggered only when the counter is enabled (TxCNTEN bit set to 1). It remains active even if the TxCNTEN bit is reset, until TxyOEN bits are set.

Balanced idle mode can be used together with the burst mode under the following conditions:

- The TxBM bit must be reset (the counter clock maintained during burst, see Section 9.3.16)
- No balanced idle protection must be triggered while the outputs are in a burst idle state

The balanced idle mode has a higher priority than the burst mode: any burst mode exit request is discarded once the balanced idle protection has been triggered. On the contrary, if the delayed protection is exited while the burst mode is active, the burst mode is resumed normally.

Note: Although the output state is frozen in idle mode, a number of events are still generated on the auxiliary outputs (refer to Section 9.3.18） during the idle period following the delayed protection：

- *Output set/reset interrupts or DMA requests*
- *External event filtering based on output signals*
- *Capture events triggered by set/reset*

9.3.10.3 Balanced idle automatic resuming

The balanced idle mode can be configured to have an automatic resuming of operation after a trigger.

Once the shortened pulse has been copied to the alternate output, the pulse width is reset to its original value, and the timer resumes operation: both outputs keep on being in RUN mode.

This mode is enabled by setting the BIAR bit in the SHRTIM_TxOUT register. This mode can only be used when the period in SHRTIM_TxPRD is greater than 6 periods of the f_{SHRTIM} clock, i.e., if CKPSC[2:0] = 0, it should be 0xC0; if CKPSC[2:0] = 1, it should be 0x60; if CKPSC[2:0] = 2, it should be 0x30, and so on.

Note: this bit is only significant when DP[2:0] = 011 or 111, otherwise it will be ignored.

Note: in balanced idle automatic resuming mode, it is mandatory to set the IDLES state to inactive.

9.3.11 Register Preload and Update Management

Most of the SHRTIM registers are buffered and can be preloaded as needed. This is typically done to avoid the waveforms from being altered by a register update not synchronized with the active events (set/reset).

When preload mode is enabled, the accessed registers become shadow registers. Upon receiving a software-initiated update request or an update request synchronized with an event, the contents of the shadow registers are transferred into the active registers.

By default, the PLEN bit in the SHRTIM_MCTRL and SHRTIM_TxCTRL registers is reset, and the registers are not preloaded: any write operation will directly update the active registers. If the PLEN bit is reset while the timer is running and preload is enabled, the content of the preload registers will be directly transferred into the active registers.

Each timing unit and the master timer have their own PLEN bit. If PLEN is set to 1, the preload registers are enabled and their contents are transferred to the active registers only when an update event occurs.

If preload functionality is needed, two methods can be used to initialize the timer:

- Enable PLEN bit at the very end of the timer initialization to have the preload registers transferred into the active registers before the timer is enabled (by setting MCNTEN and TxCNTEN bits).
- Enable PLEN bit at any time during the initialization and force a software update immediately before starting.

The following table lists the registers that can be preloaded and summarizes the available update events.

Table9-20 SHRTIM Preloadable Control Registers and Associated Update Sources

Timer	Preloadable Registers	Preload Enable	Update Sources
Master timer	SHRTIM_MIDEN SHRTIM_MPRD SHRTIM_MREPT SHRTIM_MCMP1DAT SHRTIM_MCMP2DAT SHRTIM_MCMP3DAT SHRTIM_MCMP4DAT	PLEN bit in SHRTIM_MCTRL	Software Repetition event Burst DMA event Repetition event following a burst DMA event
Timer x x = A..F	SHRTIM_TxIDEN SHRTIM_TxPRD SHRTIM_TxREPT SHRTIM_TxCMP1DAT SHRTIM_TxRCMP1DAT SHRTIM_TxCMP2DAT SHRTIM_TxCMP3DAT SHRTIM_TxCMP4DAT SHRTIM_TxDT SHRTIM_TxSET1 SHRTIM_TxRST1 SHRTIM_TxSET2 SHRTIM_TxRST2 SHRTIM_TxCNTRST	PLEN bit in SHRTIM_TxCTRL	Software TIMx repetition event TIMx reset event Burst DMA event Update event from other timers (TIMy, master) Update event following a burst DMA event shrtim_upd_en [3:1] Update event following shrtim_upd_en [3:1]
SHRTIM Common	SHRTIM_ADTG1SRC1 SHRTIM_ADTG1SRC2 SHRTIM_ADTG2SRC1 SHRTIM_ADTG2SRC2 SHRTIM_ADTG3SRC1 SHRTIM_ADTG3SRC2 SHRTIM_ADTG4SRC1 SHRTIM_ADTG4SRC2	TIMx or master timer update, depending on ADTGxUPDSRC[2:0] bits in SHRTIM_CTRL1 (if PLEN = 1 in the selected timer)	

The master timer has 4 update options:

1. Software: writing 1 into MSWUPD bit in SHRTIM_CTRL2 forces an immediate update of the registers. In this case, any pending hardware update request is cancelled.
2. Update done when the master counter rolls over and the master repetition counter is equal to 0. This is enabled when MREPTUEN bit is set in SHRTIM_MCTRL.
3. Update done once burst DMA is completed (refet to Section 9.3.23 DMA for details). This is enabled when BRSTDMA[1:0] = 01 in SHRTIM_MCTRL. It is possible to have both MREPTUEN =1 and BRSTDMA =01.

   Note: The update can take place immediately after the end of the burst sequence if SWUPD bit is set (i.e. forced update mode). If SWUPD bit is reset, the update is done on the next update event following the end of the burst sequence.
4. Update done when the master counter rolls over following a burst DMA completion. This is enabled when BRSTDMA[1:0] = 10 in SHRTIM_MCTRL.

An interrupt or a DMA request can be generated by the master update event.

Each timer (TIMA..F) can also be updated in the following ways:

- Software: writing 1 into TxSWUPD bit in SHRTIM_CTRL2 forces an immediate update of the registers. In this case, any pending hardware update request is canceled.
- Update done when the counter rolls over and the repetition counter is equal to 0. This is enabled when TxREPTUEN bit is set in SHRTIM_TxCTRL.
- Update done when the counter is reset or rolls over in continuous mode. This is enabled when RSTROUEN bit is set in SHRTIM_TxCTRL. This is used for a timer operating in single-shot mode, for instance.
- Update done once a burst DMA is completed. This is enabled when UPDGAT [3:0] = 0001 in SHRTIM_TxCTRL.
- Update done on the update event following a burst DMA completion (the event can be enabled with RSTROUEN, REPTUEN, MUEN or TxUEN). This is enabled when UPDGAT[3:0] = 0010 in SHRTIM_TxCTRL.
- Update done when receiving a request on shrtim_upd_en[3:1]. This is enabled when UPDGAT[3:0] = 0011, 0100, 0101 in SHRTIM_TxCTRL.
- Update done on the update event following a request on shrtim_upd_en[3:1] (the event can be enabled with RSTROUEN, TxREPTUEN, MUEN or TxUEN). This is enabled when UPDGAT[3:0] = 0110, 0111, 1000 in SHRTIM_TxCTRL.
- Update done synchronously with any other timer or master update (for instance TIMA can be updated simultaneously with TIMB). This is used for converters requiring several timers, and is enabled by setting bits MUEN and TxUEN in SHRTIM_TxCTRL register.

The shrtim_upd_en[3:1] allow to have an update event synchronized with on-chip events coming from the general-purpose timer. These inputs are active on the rising edge.

Table9-21 lists the connections between the update enable inputs and on-chip sources.

Table9-21 The update Enable Inputs and Sources

SHRTIM update enable signal	SHRTIM update enable assignment
SHRTIM_upd_en1	gtim8_oc1
SHRTIM_upd_en2	gtim9_oc1
SHRTIM_upd_en3	gtim10_oc1

This allows to synchronize low frequency update requests with high-frequency signals (for instance an update on the counter roll-over of a 100 kHz PWM that has to be done at a 100 Hz rate).

Note: When CKPSC[2:0] > 5, the update events will be synchronized to the prescaler clock.

The updates from adjacent timers (when the MUEN, TAUEN, TBUEN, TCUEN, TDUEN, TEUEN, TFUEN bits are set) or from software updates (TxSWUPD bit) can be considered immediately or resynchronized with the timers reset/rollover event. This is done with the RSYNCUPD bit in the SHRTIM_TxCTRL register, as shown in Figure9-53:

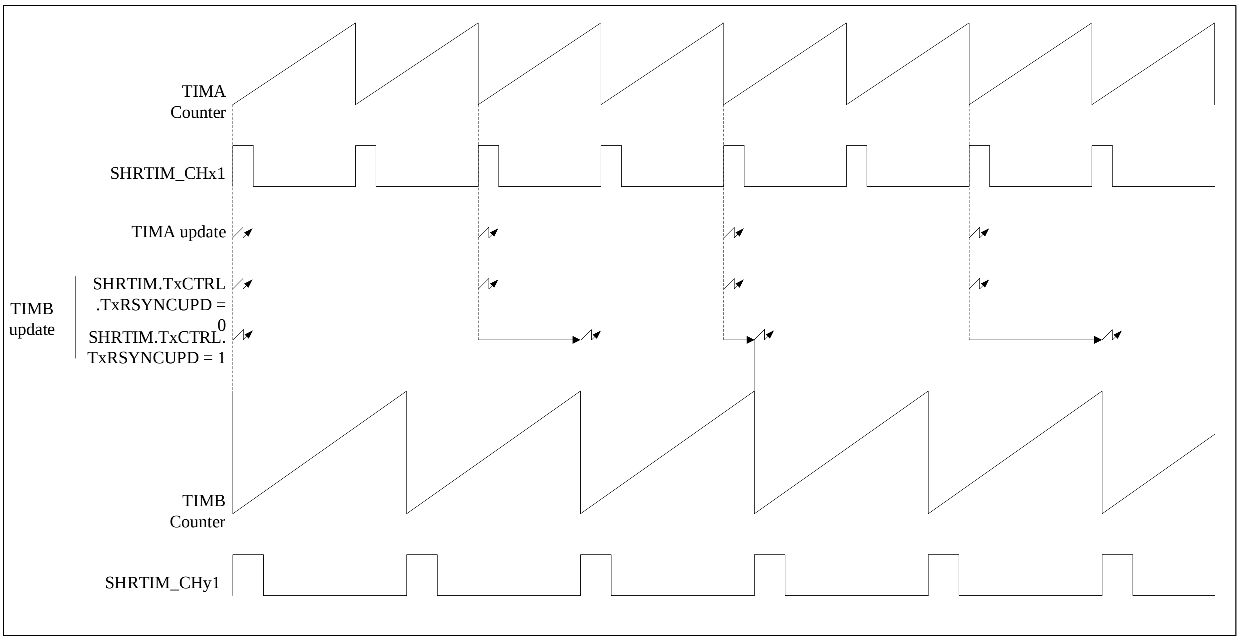
- RSYNCUPD = 0: the update coming from adjacent timers is taken into account immediately
- RSYNCUPD = 1: the update coming from adjacent timers is taken into account on the following reset/roll-over

event.

The RSYNCUPD bit is significant only when UPDGAT[3:0] = 0000, bit is significant only when

The update events of timer Timx can generate interrupts or DMA requests.

Figure9-53 Resynchronized Timer Update（SHRTIM_TBCTRL.TAUEN = 1）



The MUPDDIS and TxUPDDIS bits in SHRTIM_CTRL1 register allow to temporarily disable the transfer from preload to active registers, whatever the selected update event. This allows to modify several registers in multiple timers. The regular update event takes place once these bits are reset.

The MUPDDIS and TxUPDDIS bits are grouped together in the same register. This allows the update of multiple timers (not necessarily synchronized) to be disabled and resumed simultaneously.

Here is a practical use case: The first power converter is controlled by the master timer, TIMB, and TIMC. TIMB and TIMC must be updated simultaneously with the master timer repeat event. The second converter operates in parallel with TIMA, TIMD, and TIME. TIMD and TIME must be updated with TIMA repeat event.

First converter

MREPTUEN bit in SHRTIM_MCTRL register is set to 1: the update occurs at the end of the master timer counter repetition period. In SHRTIM_TBCTRL and SHRTIM_TCCTRL, MUEN bits are set to have TIMB and TIMC timers updated simultaneously with the master timer.

If the power converter set-point has to be adjusted by software, MUPDDIS , TBUPDDIS and TCUPDDIS bits of the SHRTIM_CTRL register must be set prior to write accessing registers to update the values (for instance the compare values). From this time on, any hardware update request is ignored and the preload registers can be accessed without any risk to have them transferred into the active registers. Once the software processing is over, MUPDDIS, TBUPDDIS and TCUPDDIS bits must be reset. The transfer from preload to active registers is done as soon as the master repetition event occurs.

Second converter

TAREPU bit in SHRTIM_TACTRL register is set to 1: the update occurs at the end of the timer A counter repetition period. In SHRTIM_TDCTRL and SHRTIM_TECTRL, TAUEN bits are set to have TIMD and TIME timers updated simultaneously with timer A.

If the power converter set-point has to be adjusted by software, TAUPDDIS, TDUPDDIS and TEUPDDIS bits of the SHRTIM_CTRL register must be set prior to write accessing the registers to update the values (for instance the compare values). From this time on, any hardware update request is ignored and the preload registers can be accessed without any risk to have them transferred into the active registers. Once the software processing is over, TAUPDDIS, TDUPDDIS and TEUPDDIS bits can be reset: the transfer from preload to active registers is done as soon as the timer A repetition event occurs.

9.3.12 PWM Mode with "Greater Than" Comparison

A specific no-latency update mode is available for PWM signals generated with the CMP1 and CMP3 registers. It allows to have a new duty cycle value applied as soon as possible within the PWM cycle, without having to wait the completion of the current PWM period. This reduces the overall delay time in software control loops. As shown below, this eventually allows to have:

- An early turn-off of the output if the new compare value is below the current counter value and the current compare value is above the counter, at the time the new value is written.
- An early turn-on of the output, re-enabling the output if the new compare value is above the counter value and the current compare value is below the counter, at the time the new value is written.

When both the new compare value and the current compare value are both lower than the counter, the output signal remains unchanged.

This feature only applies to the CMP1 or CMP3 reset events and is enabled by using the GTCMP1 and GTCMP3 bits in the SHRTIM_TxCTRL2 register. When the corresponding GTCMPx bit is set, the preload mechanism of the compare register is inactive, whatever the value of the PLEN bit. This mode is designed to make the new compare value take effect as quickly as possible after being written, without waiting for the transfer to the active register. These bits define the operation mode for compare 1 and compare 3 as follows:

- GTCMPx = 0: the compare x event is generated when the counter is equal to the compare value (compare match mode). If the compare value is changed on-the fly, the compare event may not be generated.
- GTCMPx = 1: the compare x event is generated when the counter is equal to the compare value. If the compare value is changed on-the-fly, the new compare value is compared with the current counter value and an output SET or RESET can be generated.

The "greater than" compare mode causes the crossbar to act differently depending on comparison result. Assuming the CMP1 event is doing an output reset, consider the following two scenarios when writing a new compare value:

- If the new compare value is below the counter value, the reset event is issued and can eventually cause an early turn-off.
- If the new compare value is above the counter value, a set event is generated (early turn-on).

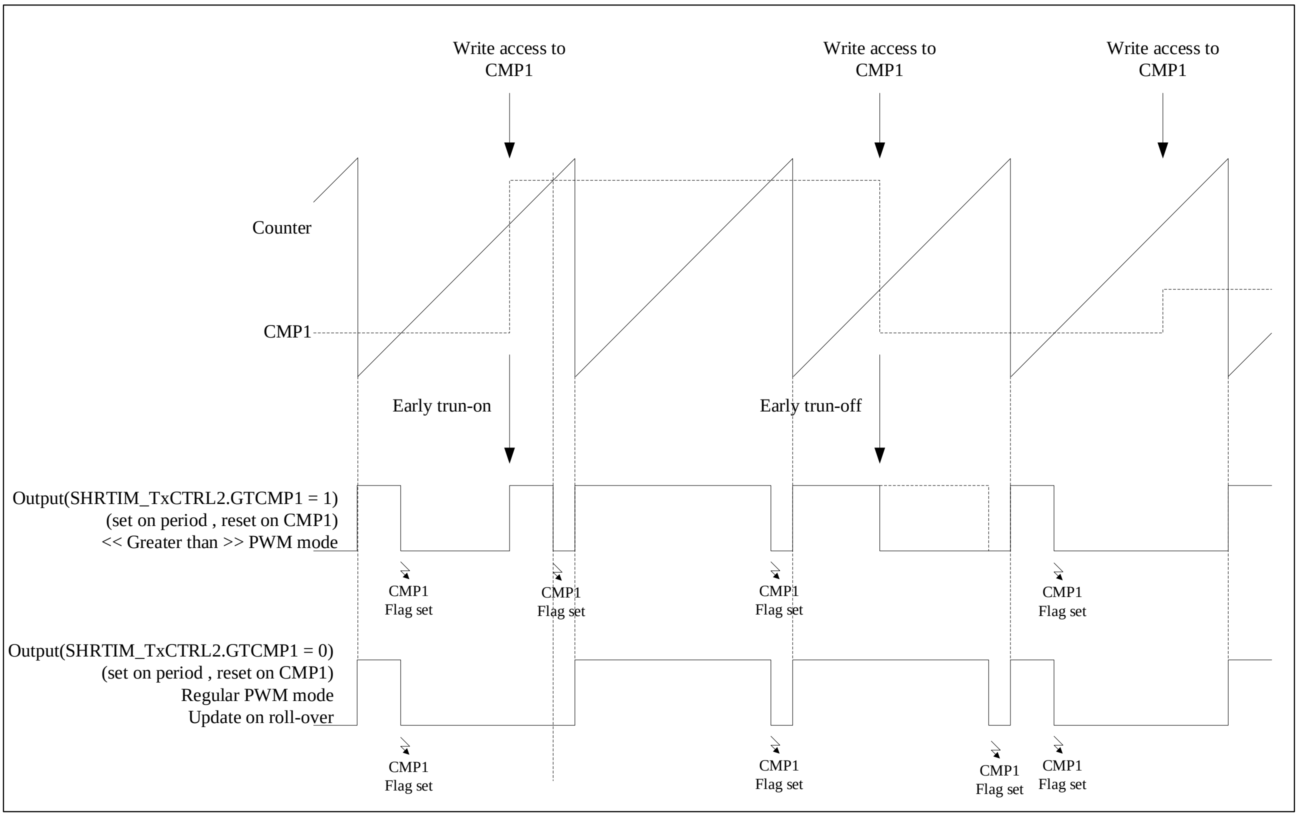
The "greater than" compare mode is supported for both set and reset actions.

The "greater than" compare mode must only be used for the following configuration:

1. In a fixed-frequency configuration, the period event must trigger an output set, while the "greater than" compare triggers an output reset (or vice versa, the period must trigger a reset if the "greater than" compare triggers a set).
2. For variable-frequency configurations, the event chosen as the counter reset source must also be chosen as the source for set or reset the timer output (opposite direction as the "greater than" compare event).

Note: The "greater-than" modes must not be used when the CMP1 and/or CMP3 modes are controlled by hardware in half and interleaved modes

Figure9-54 Early Turn-On and Early Turn-Off Behavior in "Greater Than" PWM Mode



The immediate update mode implies that the content of the preload register is transferred into the active register at the very same time the register is written. When GTCMP1 and/or GTCMP3 bits are set, their respective preload mechanism is disabled (for SHRTIM_TxCMP1DAT and/or SHRTIM_TxCMP3DAT registers), whatever the PLEN bit value.

Note: the compare interrupt flags (CMP1 and CMP3 in SHRTIM_TxINTSTS) are not generated in case of late turn-on and early turn-off.

Note: The "Greater than" comparison must not be done on both CMP1 and CMP3 for the same output (GTCMP1 and GTCMP3 bits should not be set simultaneously).

9.3.13 Events Propagation within or across Multiple Timers

The SHRTIM provides many possibilities for cascading events or sharing them between multiple timing units (including the master timer) to get full benefits from its modular architecture. These are key features for converters that require the use of multi-synchronized outputs.

This section summarizes the various options and details whether events will propagate in SHRTIM and how they propagate.

9.3.13.1 TIMx update triggered by the master timer update

The sources listed in Table 9-22 are generating a master timer update. The table indicates if the source event can be used to trigger a simultaneous update in any of TIMx timing units.

Operating condition: MUEN bit is set in SHRTIM_TxCTRL register.

Table9-22 Master Timer Update Event Propagation

Source	Condition	Propagation	Comment
Burst DMA end	BRSTDMA[1:0] = 01	No	Must be done in TxCTRL (UPDGAT[3:0] = 0001)
Roll-over event following a burst DMA end	BRSTDMA[1:0] = 10	Yes	-
Repetition event caused by a counter roll-over	MREPTU = 1	Yes	-
Repetition event caused by a counter reset (from SHRTIM_SCIN or software)		No	-
Software update	MSWUPD = 1	No	All software update bits (TxSWUPD) are grouped inthe SHRTIM_CTRL2 register and can be used for asimultaneous update

9.3.13.2 TIMx update triggered by the TIMy update

The sources listed in Table 9-23 are generating a TIMy update. The table indicates if the given event can be used to trigger a simultaneous update in another or multiple TIMx timers.

Operating condition: TyUEN bit set in SHRTIM_TxCTRL register (source = TIMy and destination = TIMx).

Table9-23 TIMx Update Event Propagation

Source	Condition	Propagation	Comment
Burst DMA end	UPDGAT[3:0] = 0001	No	Must be done directly in SHRTIM_TxCTRL (UPDGAT[3:0] = 0001)
Update caused by the update enable inputs shrtim_upd_en[3:1]	UPDGAT[3:0] = 0011, 0100, 0101	No	Must be done directly in SHRTIM_TxCTRL (UPDGAT[3:0] = 0011,0100,0101)
Master update	MUEN = 1 in SHRTIM_TyCTRL	No	Must be done with MUEN = 1 in SHRTIM_TxCTRL

Another TIMx update (TIMz>TIMy>TIMx)	TzUEN = 1 in SHRTIM_TyCTRL
	TyUEN =1 in TxCTRL
		No	Must be done with TzUEN=1 in SHRT_TxCTRL
			TzUEN=1 in SHRT_TyCTRL

Repetition event caused by a counter roll-over	REPTUEN = 1 in SHRTIM_TyCTRL	Yes	-
Repetition event caused by a counter reset	REPTUEN = 1 in SHRTIM_TyCTRL	-	Refer to counter reset cases below
Counter roll-over	RSTROUEN = 1 in SHRTIM_TyCTRL	Yes	-
Counter software reset	TySWCNTRST = 1 SHRTIM_CTRL2	YES	Can be done simultaneously with update in SHRTIM_CTRL2 register
Counter reset caused by a TIMz compare	TzCMPn in SHRTIM_TyCNTRST	YES	-
Counter reset caused by external events	EXEVn in SHRTIM_TyCNTRST	YES	-
Counter reset caused by a master compare or a master period	MCMPn or MPRD in SHRTIM_TyCNTRST TIMy's PLEN	YES	-
Counter reset caused by a TIMy compare	CMPn in SHRTIM_TyCNTRST	YES	-
Counter reset caused by an update	UPD in SHRTIM_TyCNTRST	YES	Propagation would result in a lock-up situation (update causing reset causing update)
Counter reset caused by SHRTIM_SCIN	SYNCRST in SHRTIM_TyCTRL	NO	-
Software update	TySWUPD = 1	NO	All software update bits (TxSWUPD) are grouped in the SHRTIM_CTRL2 register and can be used for a simultaneous update

9.3.13.3 TIMx counter reset causing a TIMx update

Table 9-24 lists the counter reset sources and indicates whether they can be used to generate an update.

Operating condition: RSTROUEN bit set in SHRTIM_TxCTRL register.

Table9-24 Reset Events Able to Generate an Update

Source	Condition	Propagation	Comment
Counter roll-over	-	Yes	-

Update event	UPD in SHRTIM_TxCNTRST	No	Propagation would result in a lock-up situation (update causing a reset causing an update)
External event	EXEVn in SHRTIM_TxCNTRST	Yes	-
TIMy compare	TyCMPn in SHRTIM_TxCNTRST	Yes	-
Master compare	MCMPn in SHRTIM_TxCNTRST	Yes	-
Master period	MPRD in SHRTIM_TxCNTRST	Yes	-
Compare 2 and 4	CMPn in SHRTIM_TxCNTRST	Yes	-
Software	TxSWCNTRST=1 in SHRTIM_CTRL2	Yes	-
SHRTIM_SCIN	SYNCRST in SHRTIM_TxCTRL	Yes	-

9.3.13.4 TIMx update causing a TIMx counter reset

Table 9-25 lists the update event sources and indicates whether they can be used to generate a counter reset.

Operating condition: UPD bit set in SHRTIM_TACNTRST.

Table9-25 Update Event Propagation for a Timer Reset

Source	Condition	Propagation	Comment
Burst DMA end	UPDGAT[3:0] = 0001	Yes	-
Update caused by the update enable inputs shrtim_upd_en[3:1]	UPDGAT [3:0] = 0011, 0100, 0101	Yes	-
Master update caused by a roll-over after a burst DMA	MUEN = 1 in SHRTIM_TxCTRL BRSTDMA[1:0] = 10 in SHRTIM_MCTRL	Yes	-
Master update caused by a repetition event following a roll-over	MUEN = 1 in SHRTIM_TxCTRL	Yes	-

Master update caused by a repetition event following a counter reset (software or due to SHRTIM_SCIN)	MREPTUEN = 1 in SHRTIM_MCTRL	No	-

Software triggered master timer update	MUEN = 1 in SHRTIM_TxCTRL
	MSWUPD = 1 in SHRTIM_CTRL2
		No	All software update bits

			(TxSWUPD) are grouped in the SHRTIM_CTRL2 register and can be used for a simultaneous update

TIMy update caused by a TIMy counter roll-over	TyUEN = 1 in SHRTIM_TxCTRL
	RSTROUEN = 1 in SHRTIM_TyCTRL
		Yes	-

TIMy update caused by a TIMy repetition event	TyUEN = 1 in SHRTIM_TxCTRL
	REPTUEN = 1 in SHRTIM_TyCTRL
		Yes	-

TIMy update caused by an external event or a TIMy compare (through a TIMy reset)	TyUEN = 1 in SHRTIM_TxCTRL
	RSTROUEN = 1 in SHRTIM_TyCTRL
	EXEVn or CMP4/2 in SHRTIM_TyCNTRST
		Yes	-

TIMy update caused by sources other than those listed above	TyUEN = 1 in SHRTIM_TxCTRL	yes	--timx update event come from timy reset sources and timy reset sources can from other timer(cmp event,external event) TA-TF and master(y is not include)
Repetition event following a roll-over	REPTUEN = 1 in SHRTIM_TxCTRL	Yes	-
Repetition event following a counter reset		No	-
Timer reset	RSTROUEN = 1 in SHRTIM_TxCTRL	No	Propagation would result in a lock-up situation (reset causing an update causing a reset)
Software	TxSWUPD in SHRTIM_CTRL2	yes	--timex update event is come from timx SWU can reset counter timx

9.3.14 Output Management

Each timing unit controls a pair of outputs. The outputs have three operational states:

- RUN: this is the main operating mode, where the output can take the active or inactive level as programmed in the crossbar unit.
- IDLE: this state is the default operating state after an HRTIM reset, when the outputs are disabled by software or during a burst mode operation (where outputs are temporary disabled during a normal operating mode; refer to Section 9.3.16 for more details). It is either permanently active or inactive.
- FAULT: this is the safety state, entered in case of a shut-down request on FALTx inputs. It can be permanently active, inactive or Hi-Z.

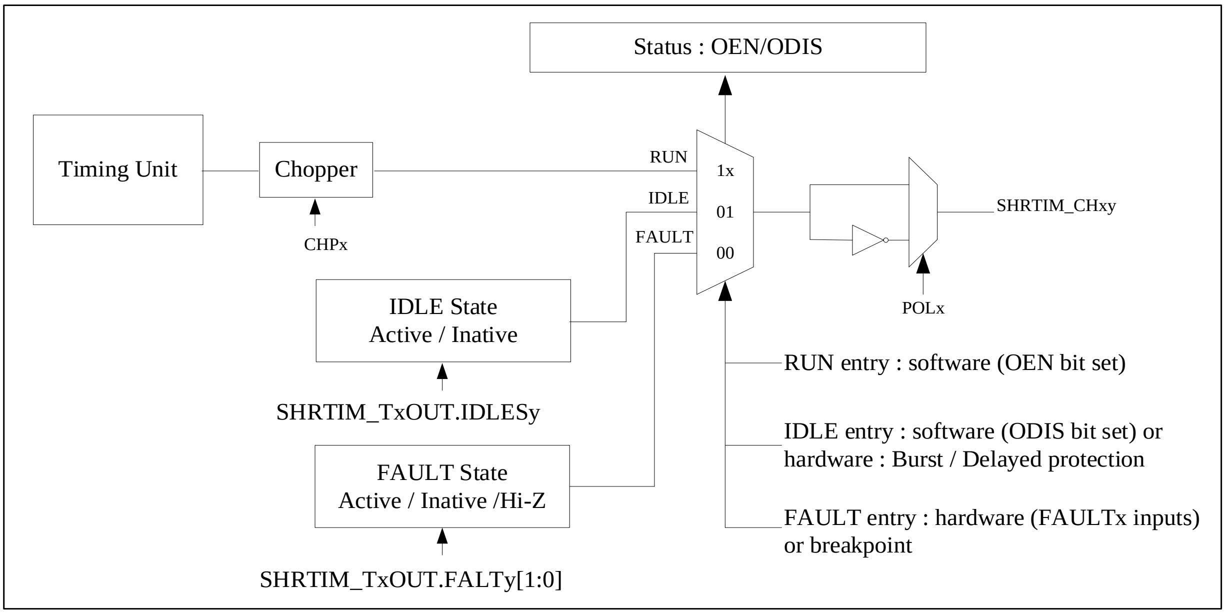
The output status is indicated by TxyOEN bit in SHRTIM_OEN register and TxyODISSTS bit in SHRTIM_ODISSTS register, as in Table9-26.

Table9-26 Output State Programming, x= A..F, y = 1 or 2

TxyOEN (Control/Status) (Set by Software, Cleared by Hardware)	TxyODISSTS (Status)	Output Operating State
1	x	RUN
0	0	IDLE
0	1	FAULT

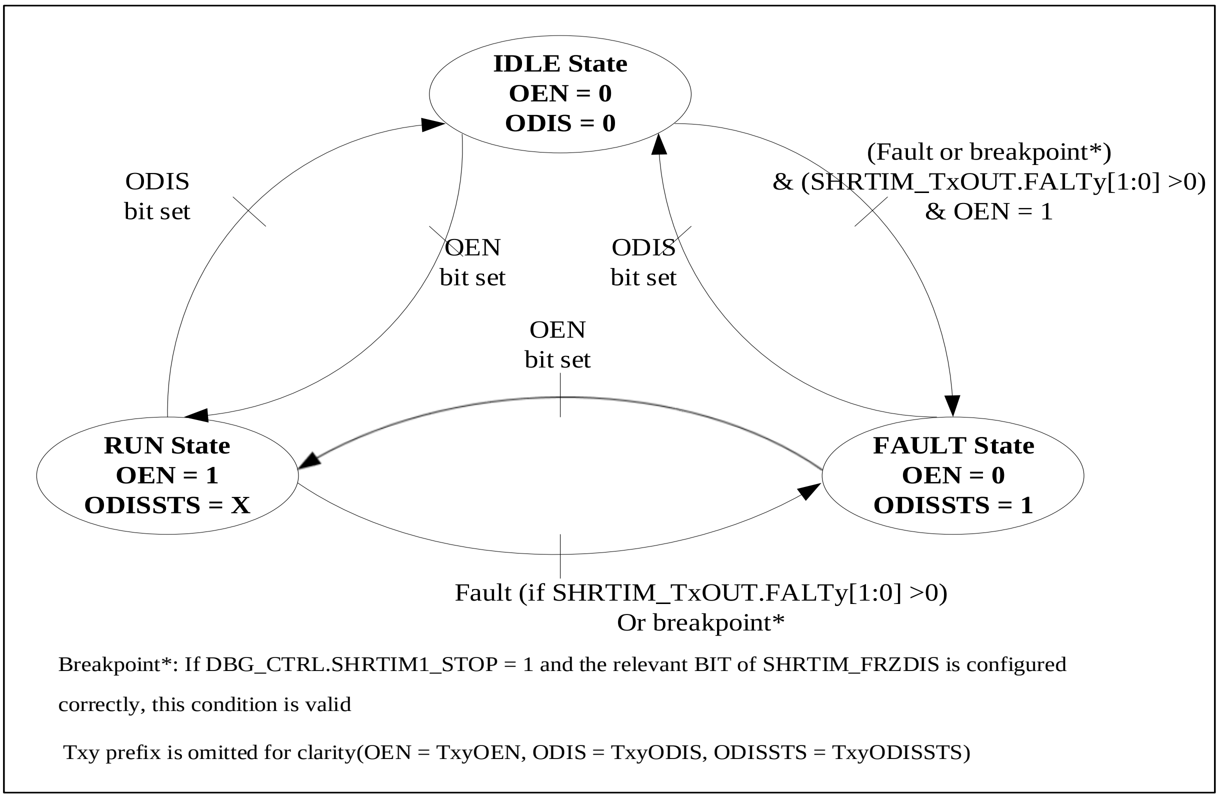
TxyOEN bit is both a control and a status bit: it must be set by software to have the output in RUN mode. It is cleared by hardware when the output goes back in IDLE or FAULT mode. When TxyOEN bit is cleared, TxyODISSTS bit indicates whether the output is in the IDLE or FAULT state. A third bit in the SHRTIM_ODIS register allows to disable the output by software.

Figure9-55 Output Management Overview



The following diagram summarize the bit values for the three states and how the transitions are triggered. Faults can be triggered by any external or internal fault source (refer to Section 9.3.17), and the idle state can be entered when burst mode or delayed protection are active.

Figure9-56SHRTIM Output States and Transitions



Breakpoint*: If DBG_CTRL.SHRTIM1_STOP = 1 and the relevant BIT of SHRTIM_FRZDIS is configured correctly, this condition is valid

Txy prefix is omitted for clarity(OEN = TxyOEN, ODIS = TxyODIS, ODISSTS = TxyODISSTS)

The FAULT and IDLE levels are defined as active or inactive. Active (or inactive) refers to the level on the timer output that causes a power switch to be closed (or opened for an inactive state).

The IDLE state has the highest priority: the transition FAULT → IDLE is possible even if the FAULT condition is still valid, triggered by ODIS bit set.

The FAULT state has a higher priority than the RUN state: if a fault event occurs while TxyOEN is set to 1, the system will transition to the FAULT state. The conditions for transitioning from IDLE → FAULT is given (as shown in Figure9-56): fault protection must be enabled (FALTx[1:0] bits = 01, 10, 11), and the TxyOEN bit should be set to 1 with a fault active (or during a breakpoint if SHRTIM1_STOP = 1).

The output polarity is programmed by the POLx bit in the SHRTIM_TxOUT register. If POLx = 0, the polarity is positive (output is high-level active); if POLx = 1, the polarity is negative (output is low-level active). In practice, the polarity definition depends on the power switches (PMOS and NMOS) or gate driver polarity being driven.

The output level in the FAULT state is configured using FALTx[1:0] bits in SHRTIM_TxOUT, for each output, as follows:

- 00: output never enters the fault state and stays in RUN or IDLE state
- 01: output at active level when in FAULT
- 10: output at inactive level when in FAULT

- 11: output is tri-stated when in FAULT. The safe state must be forced externally with pull-up or pull-down resistors, for instance.

Note: The FALTx[1:0] bits should not be changed as long as the output is in the FAULT state.

The output level in the IDLE state is configured by the IDLESx bit in SHRTIM_TxOUT as follows:

- 0: The output is at inactive level when in the IDLE state.
- 1: The output is at activelevel when in the IDLE state.

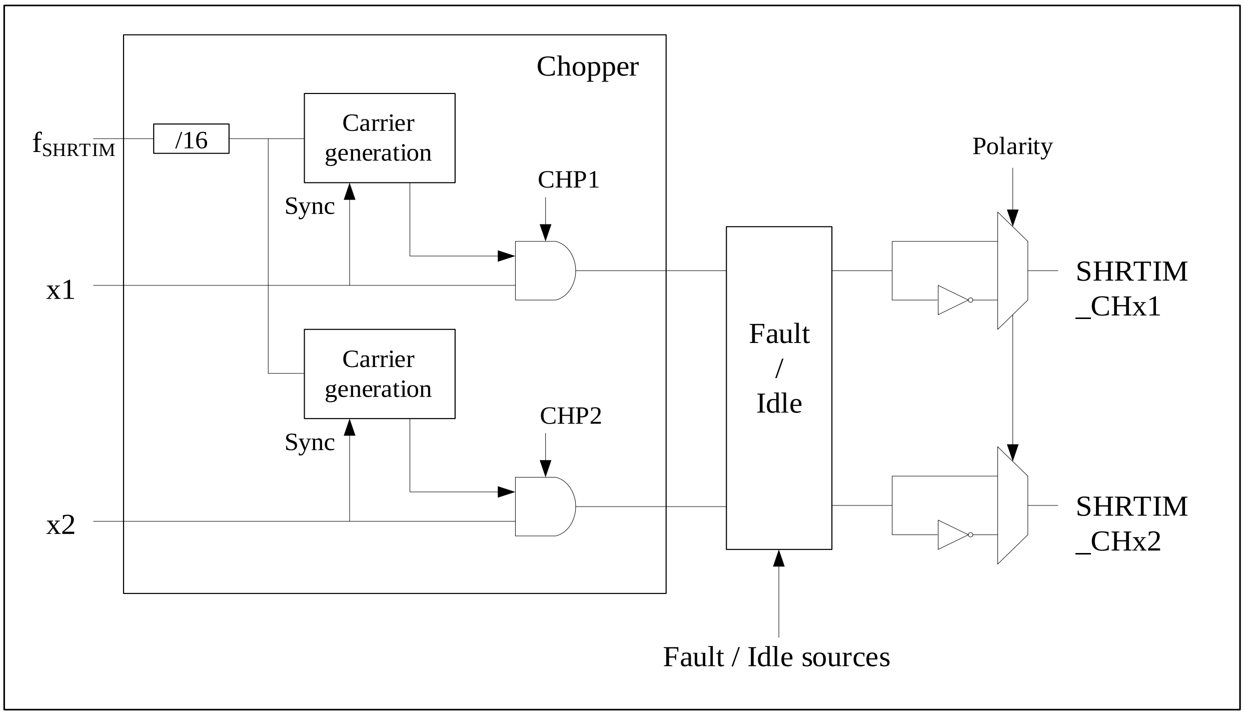
When TxyOEN is set to 1 to enter the RUN state, the output will immediately connect to the crossbar output. If the timer clock stops, the level is either inactive (after an HRTIM reset) or corresponds to the RUN level (when the timer is stopped and the output disabled).

During the SHRTIM initialization process, the output level can be prepositioned prior to have it in RUN mode, using the software forced output set and reset in the SHRTIM_TxSET1 and SHRTIM_TxRST1 registers.

9.3.15 Chopper

A high-frequency carriers can be added to the output signals of the timing unit to drive an isolation transformer. This operation is performed on the output stage before polarity insertion (as shown in Figure9-57) and use the CHP1 and CHP2 bits in the SHRTIM_TxOUT register to enable chopper on output 1 and output 2, respectively.

Figure9-57 Carrier Frequency Signal Insertion



The chopper parameters can be adjusted using the SHRTIM_TxCHOP register to define a specific pulsewidth at the beginning of the pulse, followed by a carrier frequency with programmable frequency and duty cycle, as shown in Figure9-58.

The CARFRQ[3:0] bits define the frequency according to the formula $f_{CHPFRQ} = f_{SHRTIM} / (16 \times (CARFRQ[3:0]+1))$, ranging from 976.56 kHz to 15.265 MHz (for f_{SHRTIM} = 250 MHz).

The duty cycle can be adjusted by CARDCY[2:0] (in steps of 1/8), ranging from 0/8 to 7/8.

When CARDCY[2:0] = 000 (duty cycle = 0/8), the output waveform only contains the starting pulse following the rising edge of the reference waveform, without any added carrier.

The initial pulsewidth is defined by the STARTPW[3:0] field as follows:

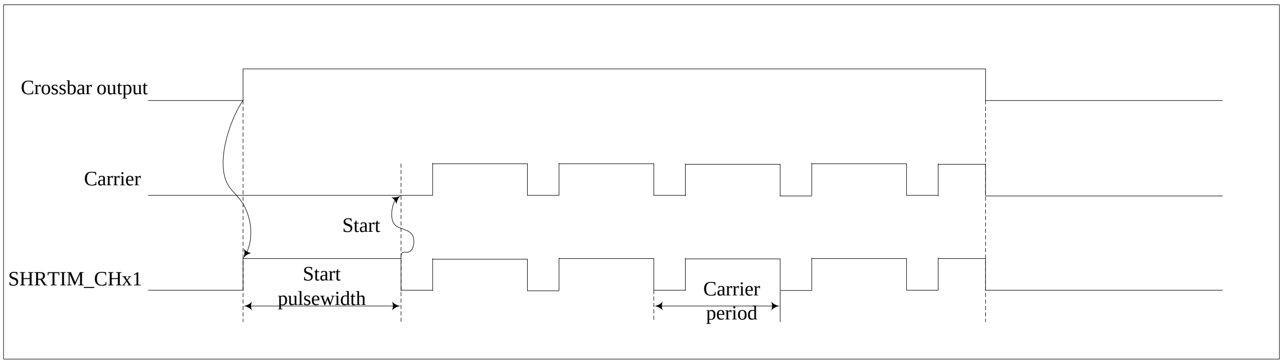
t_{1STPW} = (STARTPW[3:0]+1) x 16 x t_{SHRTIM}. The pulse width ranges from 63.93 ns to 1.03 µs (for f_{SHRTIM} = 250 MHz).

The carrier frequency parameter is defined based on the f_{SHRTIM} frequency and is independent of the CKPSC[2:0] settings.

In chopper mode, the carrier frequency and initial pulsewidth are combined with the reference waveform using a logical AND function. A Synchronization occurs at the end of the initial pulse to have a repeated signal shape.

The chopper signal is stopped at the end of the output waveform active state, without waiting for the current carrier period to be completed. Therefore, the pulses contained in the chopper signal may be shorter than the programmed pulse.

Figure9-58 SHRTIM Outputs with Chopper Mode Enabled



Note: CHP1 and CHP2 bits must be set prior to the output enable done with TxyOEN bits in the SHRTIM_OEN register.

CARFRQ[2:0], CARDCY[2:0] and STARTPW[3:0] bitfields cannot be modified while the chopper mode is active (at least one of the two CHPx bits is set)

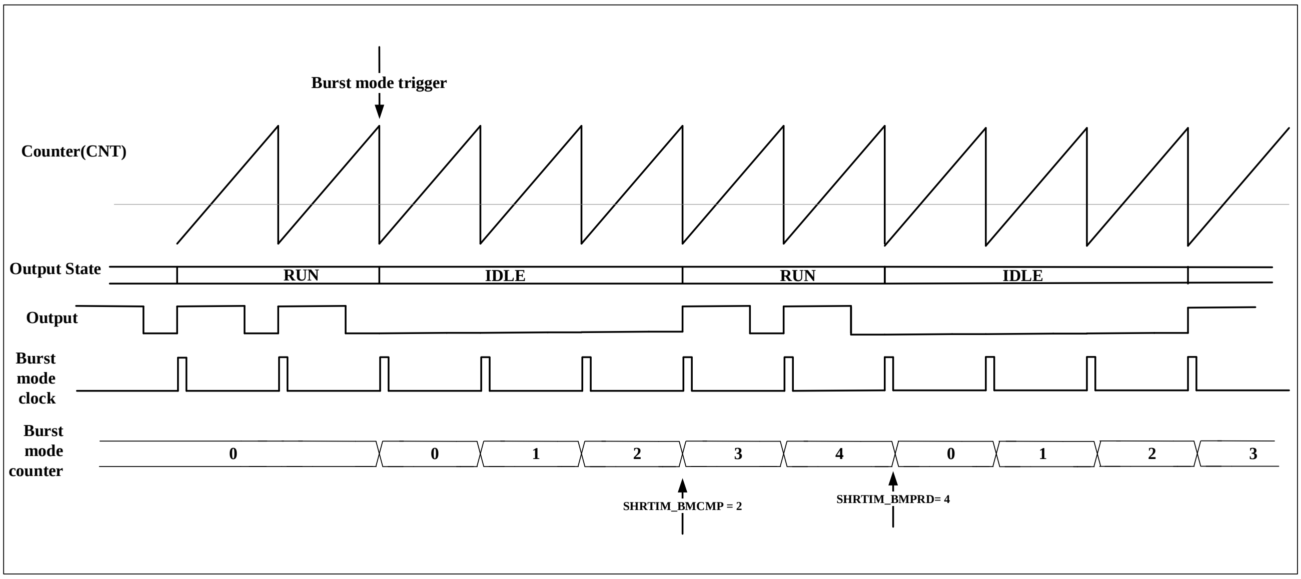
9.3.16 Burst Mode Controller

The burst mode controller allows to have the outputs alternately in IDLE and RUN states by hardware, so as to skip some switching periods with a programmable periodicity and duty cycle.

The Burst mode operation is commonly used in power converters when operating under light loads. This mode significantly increase the efficiency of the converter by reducing the number of transitions on the outputs and the associated switching losses.

When operating in burst mode, one or more pulses are generated after an idle period (equal to several counting periods), typically, where no output pulses are generated, as shown in the example in Figure9-59.

Figure9-59 Burst Mode Operation Example



The burst mode controller includes:

- A counter that can be clocked by various sources (inside or outside of SHRTIM, typically at the end of a PWM period).
- A compare register defining the number of idle periods: SHRTIM_BMCMP.
- A period register defining the burst repetition rate (equivalent to the sum of idle and run periods): SHRTIM_BMPRD.

Note: The duration of IDLE and RUN state in burst mode must be greater than the duration of a PWM output period.

The burst mode controller can take over the control of any one of the 12 PWM outputs. During burst mode operation, the state of each output is programmed through the IDLESx bit in the SHRTIM_TxOUT register, as described in the table.

Table9-27 Timer Output Programming for Burst Mode

IDLE Mx	IDLE Sx	**Output State during Burst Mode**
0	X	No action: the output is not affected by the burst mode operation.
1	0	Output inactive during the burst
1	1	Output active during the burst

The burst mode controller only affects the output stage. Some events will still occur during the idle period:

- Output set/reset interrupts or DMA requests
- External event filtering based on the Tx2 output signal
- Capture events triggered by output set/reset

Operating mode

The counter must be enabled (TxCNTEN set to 1) before using burst mode on a given timing unit. The burst mode is enabled using the BMEN bit in the SHRTIM_BMCTRL register.

The timing unit can operate in continuous mode or single-shot mode based on the BMOM bit in the SHRTIM_BMCTRL register. When BMOM = 1, continuous mode is enabled. The burst mode operation will be maintained until the BMSTS bit in the SHRTIM_BMCTRL register is reset to terminate the burst operation.

In single-shot mode (BMOM = 0), an idle sequence is executed once, following the burst mode trigger, and then normal timer operation is immediately resumed.

The durations of the idle and run periods are defined by the burst mode counter and 2 registers. The SHRTIM_BMCMP register defines the number of counts during which the selected timer(s) are in an idle state (idle period). SHRTIM_BMPRD defines the total burst mode period (sum of idle and run periods). once the initial burst mode is triggered, the idle period length is SHRTIM_BMCMP+1, and the total burst period is SHRTIM_BMPRD+1.

Note: The burst mode period must not be less than or equal to the deadtime duration defined by the DTR[8:0] and DTF[8:0] fields.

During burst mode operation, the counters of the timing unit and master timer can stop and reset. SHRTIM_BMCTRL holds 6 control bits for this purpose: MBM (for master timer) and TABM..TFBM (for timers A..F).

When the MBM or TxBM bit is reset, the counter clock is maintained. For example, this can maintain phase relationships with other timers in a multi-phase system.

If the MBM or TxBM bit is set to 1, the respective counter will stop during the burst idle period and maintain in the reset state. This allows the timer to restart a full period when exiting from the idle state. If SYNCOSRC[1:0] = 00 or 10 (synchronization output when the main timer or timer A starts), a pulse will be sent on the SHRTIM_SCOUT output when exiting burst mode.

Note: When balanced idle mode is active (DP[1:0] = 0x11), the TxBM bit must not be set.

Note: When TxBM is set, the counter reset event of timer x cannot be used for other purposes, including burst mode clock sources.

9.3.16.1 Burst mode clock

The burst mode controller counter can be clocked by multiple clock sources, selected with the BMCK[3:0] bits in the SHRTIM_BMCTRL register:

- BMCK[3:0] = 0000 to 0101: master timer and TIMA..F reset/roll-over events. This allows to have burst mode idle and run periods aligned with the timing unit counting period (both in free-running and counter reset mode).
- BMCK[3:0] = 0110 to 1001: the clock provided by shrtim_bm_ck[4:1], as shown in the table below. In this case, the burst mode idle and run periods are not necessarily aligned with the timing unit counting period (pulses on the output may be interrupted, potentially modifying the waveform duty cycle).
- BMCK[3:0] = 1010: the f_{SHRTIM} clock prescaled by a factor defined by the BMPSC[3:0] bits in the SHRTIM_BMCTRL register. In this case, the burst mode idle and run periods are not necessarily aligned with the timing unit counting period (pulses on the output may be interrupted, potentially modifying the waveform duty cycle).

Table9-28 Burst Mode Clock Sources from the General-Purpose Timer

SHRTIM Burst mode trigger event/ clock signal	SHRTIM Burst mode trigger event/ clock signal ssignment
shrtim_bm_trg	btim1_trgo
shrtim_bm_ck1	gtim8_oc1
shrtim_bm_ck2	gtim9_oc1
shrtim_bm_ck3	gtim10_oc1
shrtim_bm_ck4	btim1_trgo

The pulse width on the TIMxx OC output must be at least N f_{SHRTIM} clock cycles for the SHRTIM burst mode controller to detect it.

9.3.16.2 Burst mode triggers

To trigger burst mode operation, 32 sources are available and are selected through the SHRTIM_BMTG register:

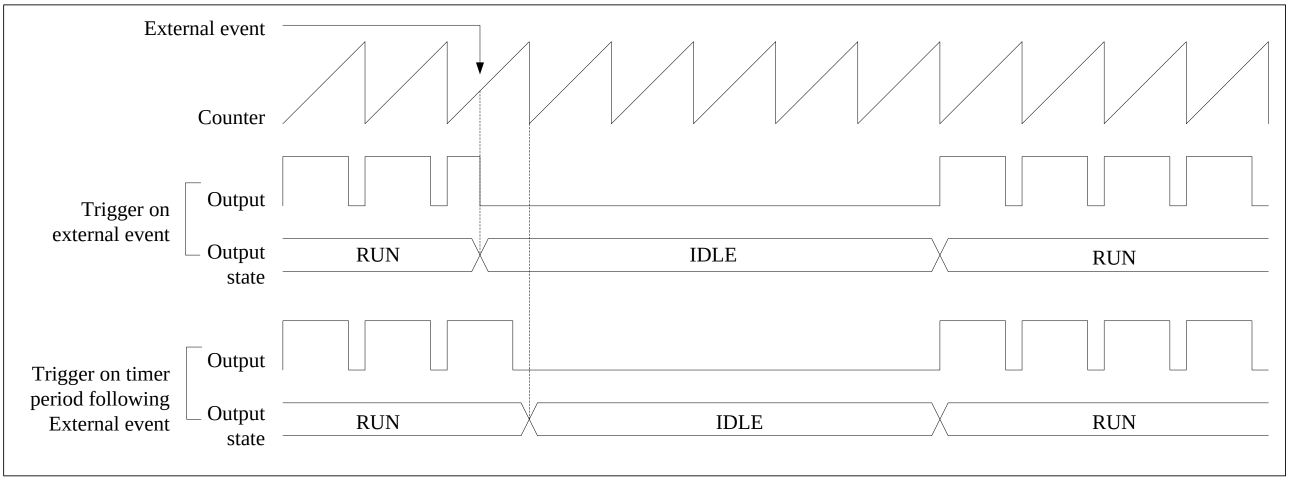
- Software trigger (set by software, reset by hardware)
- 6 master timer events: repeat, reset/roll-over, compare 1 to 4
- 6 x 4 events from timers A..F: repeat, reset/roll-over, compare 1 to 2
- shrtim_exev7 (including TIMA event filtering) and shrtim_exev8 (including TIMD event filtering)
- Timer A period following shrtim_exev7 (including TIMA event filtering)
- Timer D period following shrtim_exev8 (including TIMD event filtering)
- An on-chip events from other general timers (shrtim_bm_trg input: refer to Table 9-1).

These sources can be used in combination to achieve multi-source concurrent triggering.

The burst mode is not re-triggerable. In continuous mode, new triggers are ignored until the burst mode is terminated; in single-shot mode, the triggers are ignored until the current burst completion including run periods (SHRTIM_BMPRD+1 cycle). This applies to software triggers as well (the software bit is reset by hardware even if it is discarded).

Figure9-60 shows how the burst mode is started in response to an external events (immediate activation or activation at a timer period after the event).

Figure9-60 Burst Mode Trigger on External Event



For the TAEXEV7 and TDEXEV8 combined triggers (triggering at a timer period following an external events), the external event detection is always active, regardless of the burst mode programming.

- When the burst mode is enabled (BMEN =1) or the trigger is enabled (TAEXEV7 or TDEXEV8 bit set in the BMTGG register) in between the external event and the timer period event, the burst is triggered.

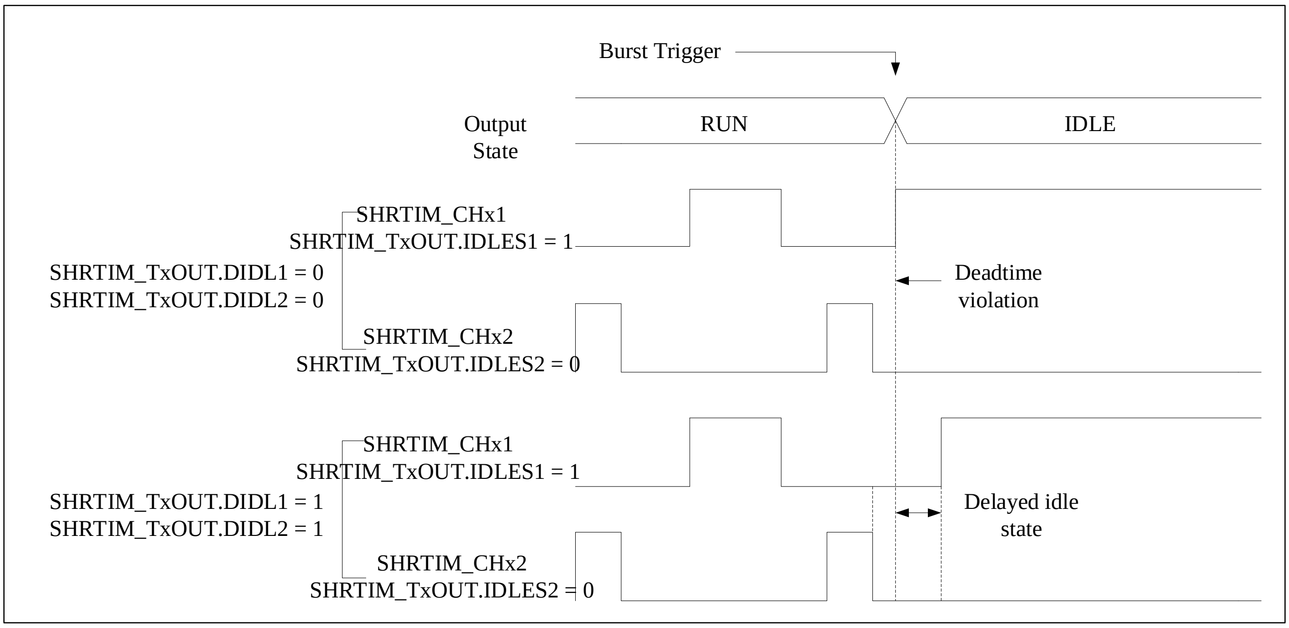
Note: TAEXEV7 and TDEXEV8 triggers are only valid after the period event. If the counter is reset before the period event, any pending shrtim_exev7/8 events will be discarded.

9.3.16.3 Burst mode delayed entry

By default, the output will immediately take their idle level (as per the IDLES1 and IDLES2 settings) after the burst mode trigger.

It is also possible to delay the burst mode entry and force the output to an inactive state during a programmable period before the output takes its idle state. This feature is useful when driving two complementary outputs (where one output has an active idle state) to avoid a deadtime violation, as shown in Figure9-61. This helps prevent the risk of shoot-through current in the half-bridge, but it causes a delayed response to burst mode entry.

Figure9-61 Delayed Burst Mode Entry with Deadtime Enabled and IDLESx = 1



The delay burst entry mode is enabled by the DIDLx bits in the SHRTIM_TxOUT register (one enable bit per output). This mode forces an insertion of deadtime before the output takes its idle state. Each TIMx output has its own deadtime value:

- DTR[8:0] on output 1 when DIDL1 = 1
- DTF[8:0] on output 2 when DIDL2 = 1

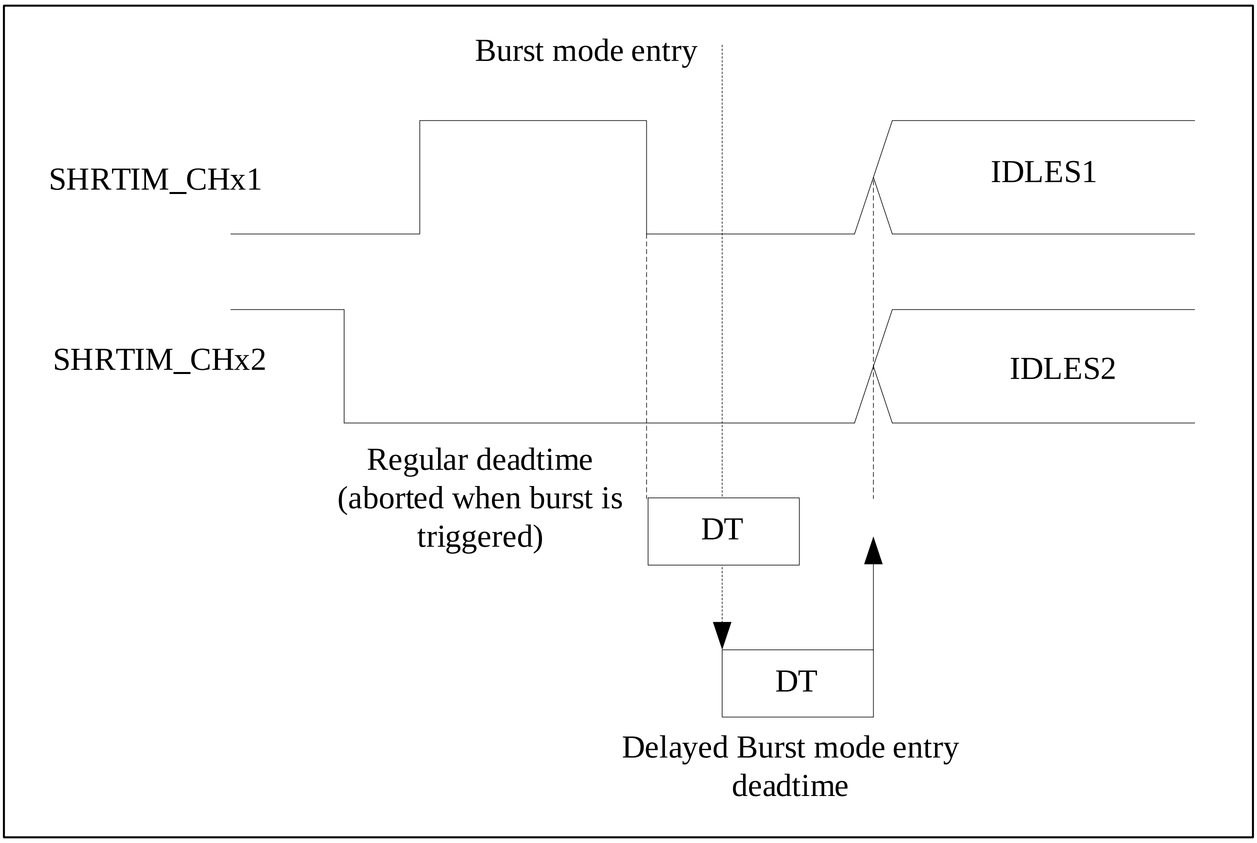
The DIDLx bit can only be set to 1 if one of the outputs has an active idle level during burst mode (IDLES=1) and a positive deadtime is used (SDTR/SDTF set to 0).

Note: the delayed burst entry mode uses the deadtime generator resources. Therefore, if either of the two DIDLx bits is set to 1 and the corresponding timer unit uses deadtime insertion (DTEN set to 1 in SHRTIM_TxOUT), timerx output 2 cannot be used as an external event filter (Tx2 filter signal is not available).

When durations defined by DTR[8:0] and DTF[8:0] are lower than 3 f_{HRTIM} clock cycle periods, the limitations related to the narrow pulse management listed in Section 9.3.7 must be applied.

When the burst mode entry arrives during the regular deadtime, it is aborted and a new deadtime is re-started corresponding to the inactive period, as on Figure9-62.

Figure9-62 Delayed Burst Mode Entry during Deadtime



9.3.16.4 Burst mode exit

The burst mode exit is either forced by software (in continuous mode) or once the idle period is elapsed (in single-shot mode). In both cases, the counter restarts immediately (if the counter is held in reset state by the MBM or TxBM bit = 1), but the effective output state transition from the idle mode to active mode only happens after the programmed set/reset event.

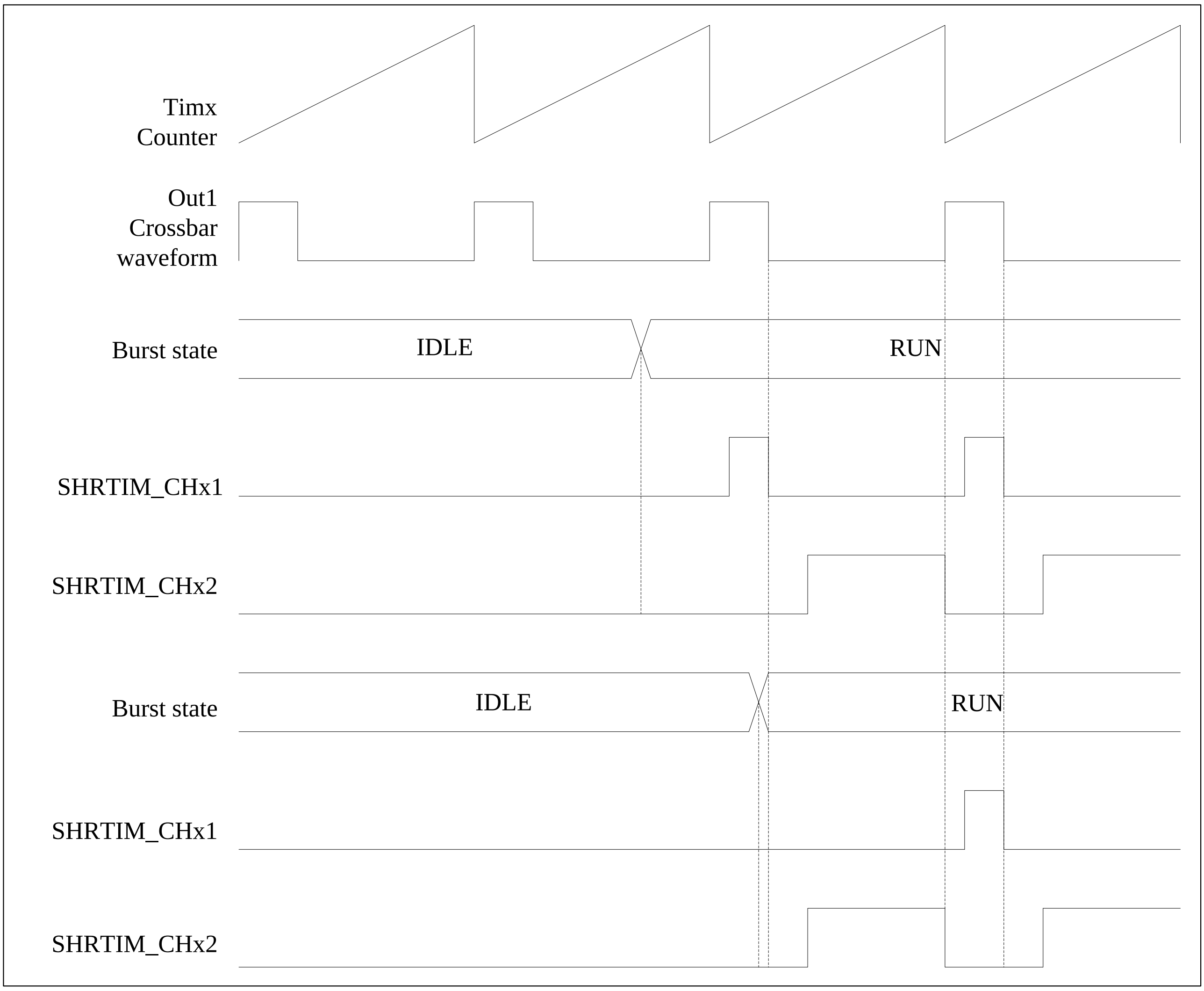
If the BMPRDIEN enable bit in the SHRTIM_INTEN register is set to 1, a burst period interrupt will be generated in both single-shot and continuous modes. This interrupt can be used to synchronize burst mode exit with a burst period in continuous burst mode.

Figure9-63 illustrates how a normal operation is resumed when the deadtime is enabled. Although the burst mode exit is immediate, this is only effective on the first set event on any of the complementary outputs.

The diagram presents two scenarios:

- The burst mode ends while the signal is inactive on the crossbar output waveform. The active state is resumed on Tx1 and Tx2 on the set event for the Tx1 output, and the Tx2 output does not take the complementary level on burst exit.
- The burst mode ends while the crossbar output waveform is active: the activity is resumed on the set event of Tx2 output, and Tx1 does not take the active level immediately on burst exit.

Figure9-63 Burst Mode Exit When the Deadtime Generator Is Enabled



If push-pull mode is enabled, the behavior described above will be slightly different. The push-pull mode forces an output reset at the beginning of the period if the output is inactive, or symmetrically forces an active level if the output was high during the preceding period.

Therefore, an output with an active idle state can be reset at the time the burst mode is exited even if no transition is explicitly programmed. For symmetrical reasons, an output can be set at the time the burst mode is exited even if no transition is explicitly programmed, in case it was active when it entered in idle state.

9.3.16.5 Burst mode registers preloading and update

The BMPLEN bit (Burst Mode Preload Enable) allows for burst mode compare and preload of the period registers (SHRTIM_BMCMP and SHRTIM_BMPRD).

When BMPLEN is set to 1, the transfer from preload to active register happens:

- When the burst mode is enabled (BMEN = 1)
- At the end of the burst mode period

A write into the SHRTIM_BMPRD period register disables the updates temporarily, until the SHRTIM_BMCMP compare register is written, to ensure the consistency of the two registers when they are modified.

If only the compare register needs to be changed, a single write operation is sufficient. However, If the period only needs to be changed, it is also necessary to re-write the compare to have the new values taken into account.

When BMPLEN bits is reset, the write access into BMCMPR and BMPRD directly updates the active register. In this case, it is necessary to consider when the update is done during the overall burst period, for the 2 cases below:

- Compare register update

  If the new compare value is greater than the current burst mode counter value, the new compare value will be considered in the current period.

  If the new compare value is less than the current burst mode counter value, in continuous mode, the new compare value will be considered in the next burst period; in single-shot mode, the new compare value will be ignored (there will be no compare match, and the output will remain the idle state until the idle period ends).

- Period register update

If the new period value is greater than the current burst mode counter value, the change will be considered in the current period.

Note: If the new period value is below the current burst mode counter value, the new period is not taken into account, the burst mode counter overflows (at 0xFFFF) and the change is effective in the next period. In single-shot mode, the counter rolls over at 0xFFFF and the burst mode re-starts for another period up to the new programmed value.

9.3.16.6 Burst mode emulation using a compound register

The burst mode controller only controls one or a set of timers for a single converter. When the burst mode is necessary for multiple independent timers, it is possible to emulate a simple burst mode controller using the DMA and the SHRTIM_TxRCMP1DAT compound register, which holds aliases of both the repetition and the compare 1 registers.

This method is suitable for converters (typically a buck converters) that only require a simple PWM. In this case, the CMP1 register resets the output (and defines the duty cycle), while CMPx (x=2~4) sets it. During the burst idle, the value of CMP1 is the same as that of CMP2 (reset takes priority over set, and the output remains in the reset state)."

Note: The time when the counter equals 0 to CMPx must be greater than the time required for DMA transfer (DMA transfer requires 5 to 6 tHCLK cycles without arbitration).

In this case, a single 32-bit write access in SHRTIM_TxRCMP1DAT is sufficient to define the duty cycle (with the CMP1 value) and the number of periods during which this duty cycle is maintained (with the repetition value). To implement a burst mode, it is then only necessary to transfer by DMA (upon repetition event) two 32-bit data in continuous mode, organized as follows:

SHRTIM_TxRCMP1DAT = {REPT_Run; CMP1 = Duty_Cycle}, {REPT_Idle; CMP1 = CMPx}

For example, the values:

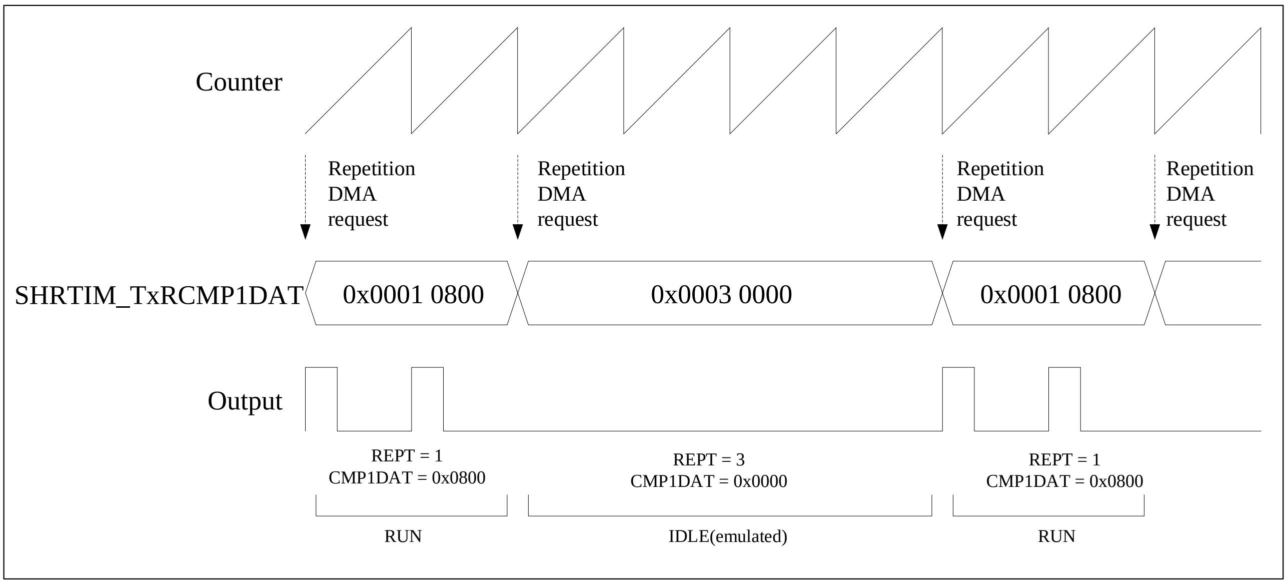
{0x0003 0200}：CMP1 = 0x0200. Output 0% duty cycle, for 4 periods.

{0x0001 0A00}：CMP1 = 0x0A00. Output 50% duty cycle, for 2 periods.

DMA transfer (0x00030200, 0x00010A00). For burst idle, 0x00030200 of DMA transfer, CMP1=0x0200 is equal to CMP2's value of 0x0200. Reset takes priority over set, and the output remains in the reset state. The repeat counter value is 3, and the output maintains a 0% duty cycle for 4 periods. For burst run, 0x00010A00 of DMA transfer, with

CMP2=0x0200 setting the output, and CMP1=0x0A00 resetting it, resulting in a 50% duty cycle. The repeat counter value is 1, and the output maintains a waveform with a 50% duty cycle for 2 periods. As shown on Figure9-64.

Figure9-64 Burst Mode Emulation Example

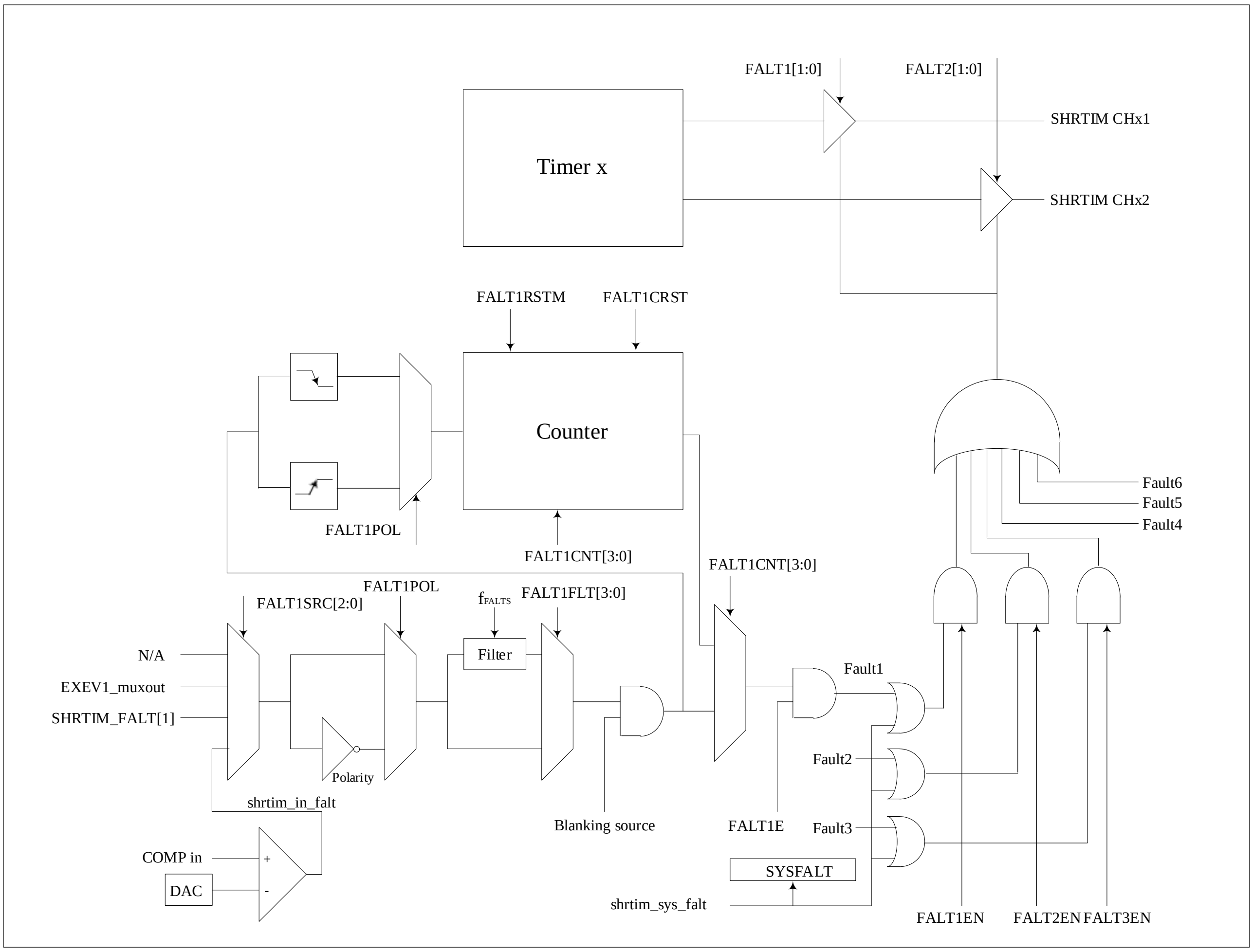


9.3.17 Fault Protection

The SHRTIMER features a general fault protection circuit that can disable the output in case of an abnormal operation. Once a fault is triggered, the output will enter a predefined safe state. The output will maintain in this state until re-enabled by software. In the case of a permanent fault request, it will remain in its fault state, even if software attempts to re-enable them,until the fault source is disappered.

The SHRTIM has 6 FAULT input channels, all of them are available and can be combined for each of the 6 timing units, as shown in Figure9-65.

Figure9-65 Fault Protection Circuitry (FAULT1 Fully Represented, FAULT2..6 Partially)



Each fault channel can be fully configured through the SHRTIM_FALTIN1 and SHRTIM_FALTIN2 registers before being routed to the timing unit. The FALTxSRC[1:0] bits are used to select the fault signal source, which can be a digital input or an internal event (output of the built-in comparator).

The sources available for each of the 6 fault channels are summarized as follows:

Table9-29 Fault Inputs

SHRTIM Fault channel	SHRTIM External Input FALTxSRC[1:0] = 00	On-chip source FALTxSRC[1:0] = 01	External Input FALTxSRC[1:0] = 10	On-chip source FALTxSRC[1:0] = 11
shrtimx_falt1[4:1]	SHRTIMx_FALT1	compx_out(1~7)	EXEV1_muxout	N/A
shrtimx_falt2[4:1]	SHRTIMx_FALT2	compx_out(1~7)	EXEV2_muxout	N/A
shrtimx_falt3[4:1]	SHRTIMx_FALT3	compx_out(1~7)	EXEV3_muxout	N/A
shrtimx_falt4[4:1]	SHRTIMx_FALT4	compx_out(1~7)	EXEV4_muxout	N/A
shrtimx_falt5[4:1]	SHRTIMx_FALT5	compx_out(1~7)	EXEV5_muxout	N/A
shrtimx_falt6[4:1]	SHRTIMx_FALT6	compx_out(1~7)	EXEV6_muxout	N/A

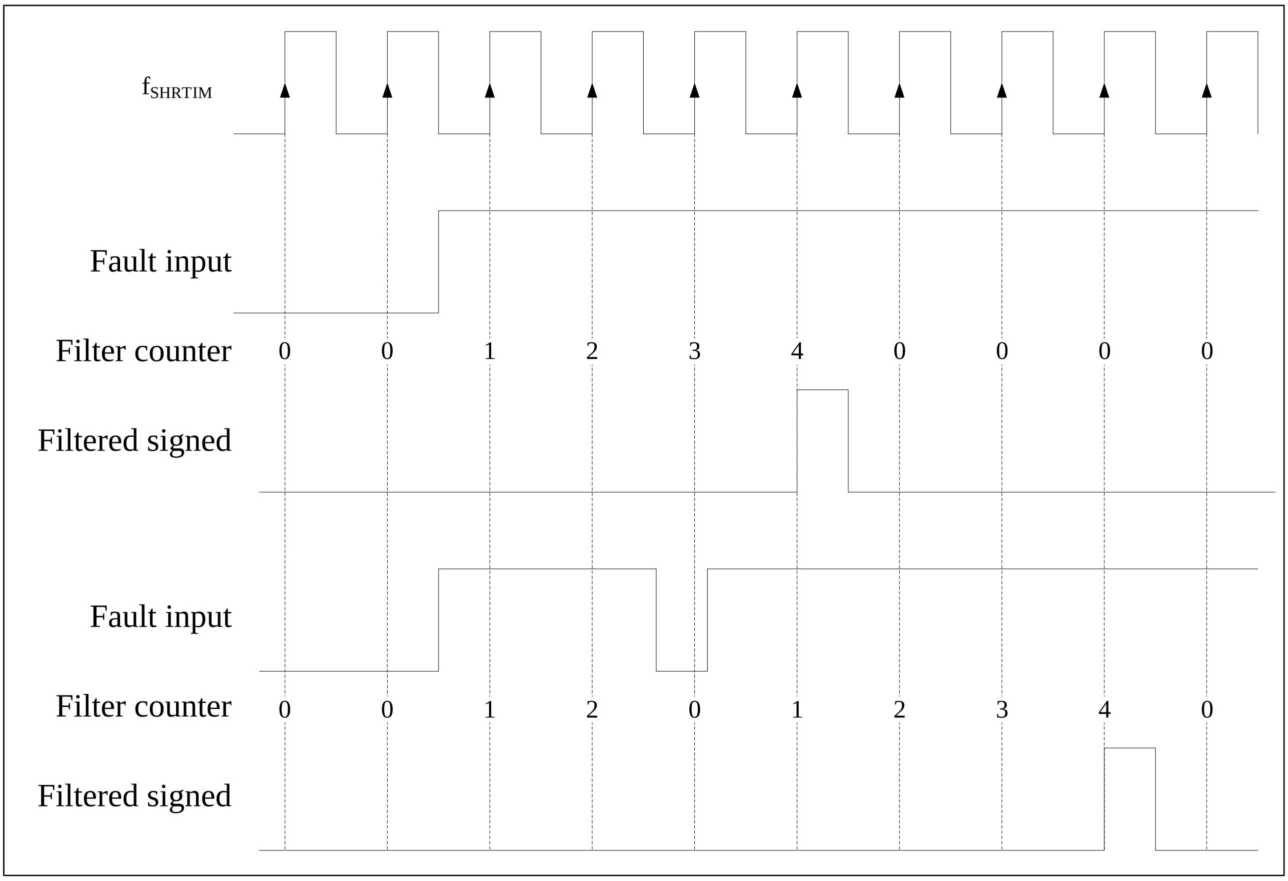
1. compx_out can be mapped to any comparator.

It is possible to trigger a fault using software by configuring the SFTFALTx in SHRTIM_SFTFALT to select which fault channel to trigger.

The EXEVx_muxout event mentioned in Table9-29 above is taken after the shrtim_exevx[5:1] input multiplexer controlled by the EXEVxSRC[1:0] bits.Refer to Figure9-37 for details.

The polarity of the signal can be selected to define the active level, using the FALTxPOL polarity bit in SHRTIM_FALTINx registers. If FALTxPOL = 0, the signal is active at low level; if FALTxPOL = 1, it is active when high.The fault information can be filtered after the polarity setting. If FALTxFLT[3:0] bitfield is set to 0000, the signal is not filtered and acts asynchronously, independently from the f_{SHRTIM} clock. For all other FALTxFLT[3:0] bitfield values, the signal is digitally filtered. The digital filter is made of a counter in which a number N of valid samples is needed to validate a transition on the output. If the input value changes before the counter has reached the value N, the counter is reset and the transition is discarded (considered as a spurious event). If the counter reaches N, the transition is considered as valid and transmitted as a correct external event. Consequently, the digtal filter adds a latency to the external events being filtered, depending on the sampling clock and on the filter length (number of valid samples expected). Figure9-66 shows how a spurious fault signal is filtered.

Figure9-66 Fault Signal Filtering（FALTxFLT[3:0]= 0010：$f_{SAMPLING} = f_{SHRTIM}$，N = 4）



The filtering period ranges from 2 cycles of the f_{SHRTIM} clock up to 8 cycles of the f_{FALTS} clock divided by 32. f_{FALTS} is defined using FALTSCD[1:0] bits in the SHRTIM_FALTIN2 register. The following table summarizes the sampling rate and the filter length. A jitter of 1 sampling clock period must be subtracted from the filter length to take into account the uncertainty due to the sampling and have the effective filtering.

Table9-30 Sampling Rate and Filter Length vs FALTxFLT[3:0] and Clock Setting

	f_{FALTS} Vs FALTSCD [1:0]				Filter length when f_{SHRTIM} = 250 MHz	
FALTxFLT[3:0]	00	01	10	11	Min	Max
0001,0010,0011	f_{SHRTIM}	f_{SHRTIM}	f_{SHRTIM}	f_{SHRTIM}	f_{SHRTIM}， N =2 8 ns	f_{SHRTIM}， N =8 32 ns
0100, 0101	f_{SHRTIM}/2	f_{SHRTIM}/4	f_{SHRTIM}/8	f_{SHRTIM}/16	f_{SHRTIM}/2， N = 6 48 ns	f_{SHRTIM}/16， N = 8 512 ns
0110, 0111	f_{SHRTIM}/4	f_{SHRTIM}/8	f_{SHRTIM}/16	f_{SHRTIM}/32	f_{SHRTIM}/4， N = 6 96 ns	f_{SHRTIM}/32， N = 8 512 ns
1000, 1001	f_{SHRTIM}/8	f_{SHRTIM}/16	f_{SHRTIM}/32	f_{SHRTIM}/64	f_{SHRTIM}/8， N = 6 192 ns	f_{SHRTIM}/64， N = 8 2.48 µs
1010, 1011, 1100	f_{SHRTIM}/16	f_{SHRTIM}/32	f_{SHRTIM}/64	f_{SHRTIM} /128	f_{SHRTIM}/16， N = 5 320 ns	f_{SHRTIM}/128， N = 8 4.96 µs
1101, 1110, 1111	f_{SHRTIM}/32	f_{SHRTIM}/64	f_{SHRTIM} /128	f_{SHRTIM} /256	f_{SHRTIM}/32， N = 5 640 ns	f_{SHRTIM}/256， N = 8 9.92 µs

Fault blanking and event counting

The fault inputs can be temporarily disabled to blank spurious fault events. The table below lists the sources of fault blanking.

Table9-31 Fault Input Blanking Events

-	FALTxBLKS = 0, reset-aligned window		FALTxBLKS = 1 moving window	
Fault input	Blanking window start	Blanking window end	Blanking window start	Blanking window end
shrtim_falt1[4:1]	Timer A reset/roll-over	Timer A CMP3 event	Timer A CMP4 event	Timer A CMP3 event
shrtim_falt2[4:1]	Timer B reset/roll-over	Timer B CMP3 event	Timer B CMP4 event	Timer B CMP3 event
shrtim_falt3[4:1]	Timer C reset/roll-over	Timer C CMP3 event	Timer C CMP4 event	Timer C CMP3 event
shrtim_falt4[4:1]	Timer D reset/roll-over	Timer D CMP3 event	Timer D CMP4 event	Timer D CMP3 event
shrtim_falt5[4:1]	Timer E reset/roll-over	Timer E CMP3 event	Timer E CMP4 event	Timer E CMP3 event
shrtim_falt6[4:1]	Timer F reset/roll-over	Timer F CMP3 event	Timer F CMP4 event	Timer F CMP3 event

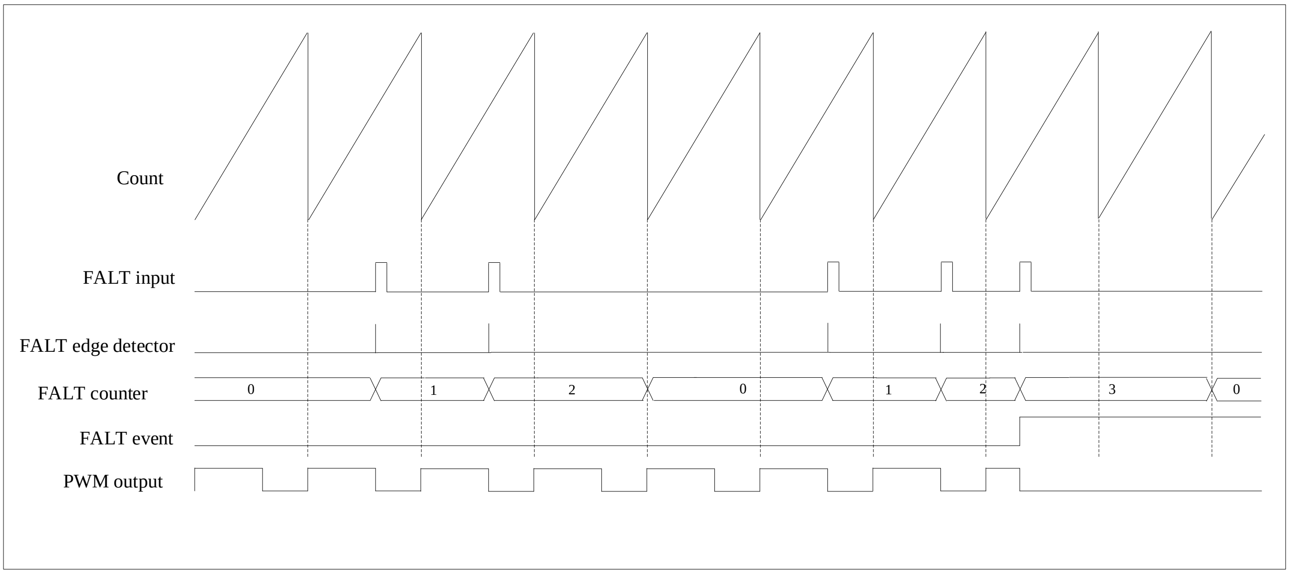
A fault counter also allows to discard multiple spurious fault events and define an acceptance criteria.

The FALTxCNT[3:0] bitfield selects the FAULTx counter threshold. A fault is considered valid when the number of events is equal to the FALTxCNT[3:0]value. This FALTxRSTM selects the FAULTx counter reset mode.

- 0: the fault counter is hardware reset on reset/roll-over events, as shown in Table9-32.
- 1: the fault counter is reset on each reset/roll-over event only if no event occurs during last counting period, as shown on Figure9-66.

The fault counter can be reset by software with the FALTxCRST bit at any time.

Figure9-67 Fault Counter Cumulative Mode (FALTxRSTM = 1, FALTxCNT[3:0] = 2)



A given FALTx input counters can be reset by a single source. The table below indicates the timer units associated with a given fault. This does not prevent to have a fault line shared by multiple timer (for example, FALT1 with event counter enabled, acting on Timer A, Timer B, and Timer C simultaneously).

Table9-32 Faule 1..6 Counter Reset Source

Fault input	Fault counter reset source
shrtim_falt1[4:1]	Timer A reset/roll-over
shrtim_falt2[4:1]	Timer B reset/roll-over
shrtim_falt3[4:1]	Timer C reset/roll-over
shrtim_falt4[4:1]	Timer D reset/roll-over
shrtim_falt5[4:1]	Timer E reset/roll-over
shrtim_falt6[4:1]	Timer F reset/roll-over

These features provide the SHRTIM with flexible fault management capabilities, allowing for adaptability to specific application requirements.

System fault input (shrtim_sys_flt)

System faults from the following sources:

- The clock security system
- The SRAM ECC（Detects two-bit errors and issues an error signal.）
- The SRAM parity checker

- The Cortex^{TM}-M4F-lockup signal
- The PVD detector
- The FLASH ECC double error detection

This input overrides the FAULT inputs and disables all outputs having FAULTy[1:0][1:0] = 01, 10, 11.

For each FAULT channel, the FALTxLCK bit in the SHRTIM_TxFALT register can be written only once to lock the FALTxE, FALTxPOL, FALTxSRC, and FALTxFLT[3:0] bits (making them read-only) to achieve functional safety. Once enabled, the fault conditioning settings will be frozen until the next SHRTIM reset or system reset.

Once the fault signal is conditioned as explained above, it is routed to the timing units. For any timing unit, the FALT1EN to FALT6EN bits in the SHRTIM_TxFALT register enable the 6 fault channels, and all 6 channels can be selected simultaneously (as long as the output is protected by the fault mechanism, the sysfault will be automatically enabled). This allows for:

- Disabling multiple timing units simultaneously using a single fault channel
- Disabling a single timing unit by performing an OR operation on multiple fault channels

The FALTLCK bit in the SHRTIM_TxFALT register, which can only be written once, allows to lock the FALTxEN bits (it renders them read-only) until the next reset, for functional safety purpose. Once enabled, the fault-related settings for the timing unit will be frozen until the next SHRTIM reset or system reset.

For each of the timer, the output state during a fault is defined by the FALT1[1:0] and FALT2[1:0] bits in the SHRTIM_TxOUT register (refer to Section 9.3.14).

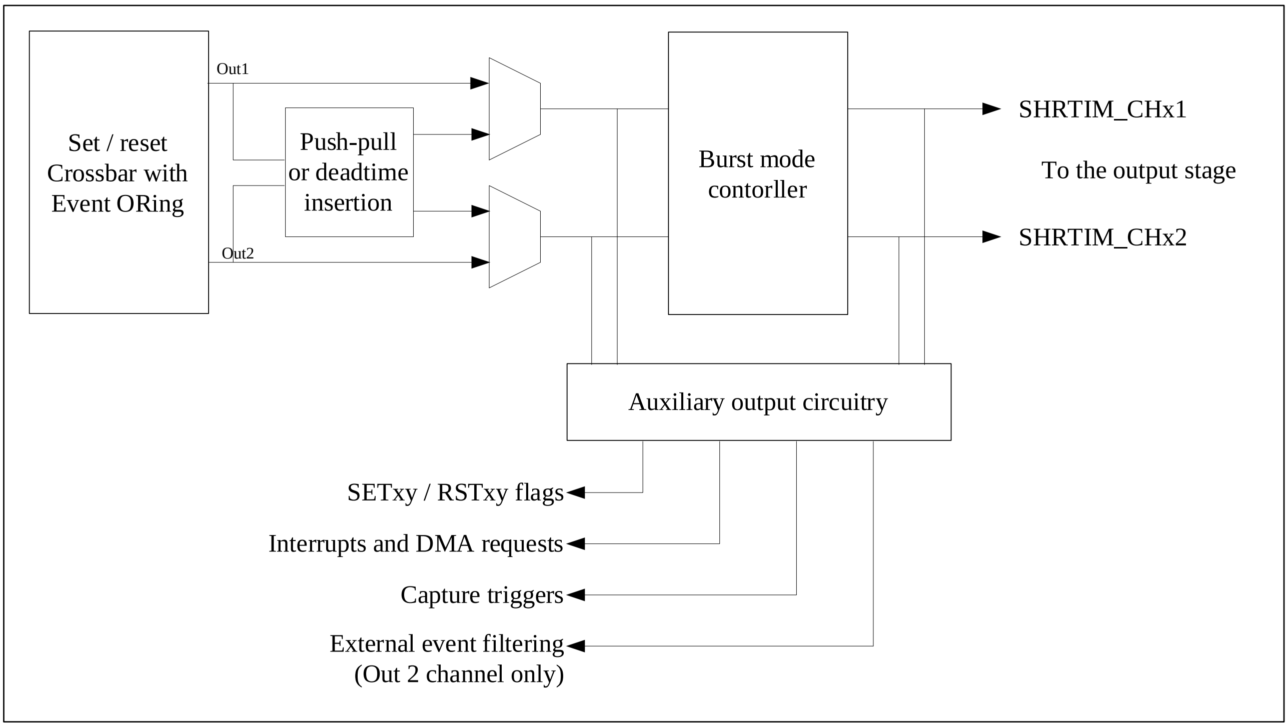
9.3.18 Auxiliary Outputs

Timer A to F have auxiliary outputs in parallel with the regular outputs going to the output stage. They provide the following internal status, events and signals:

- SETyITF and RSTyITF status flags, together with the corresponding interrupts and DMA requests
- Capture triggers upon output set/reset
- External event filters following a Tx2 output copy (refer to Section 9.3.9 for details)

The auxiliary outputs are taken either before or after the burst mode controller, depending on the SHRTIM operating mode. An overview is given on Figure9-68.

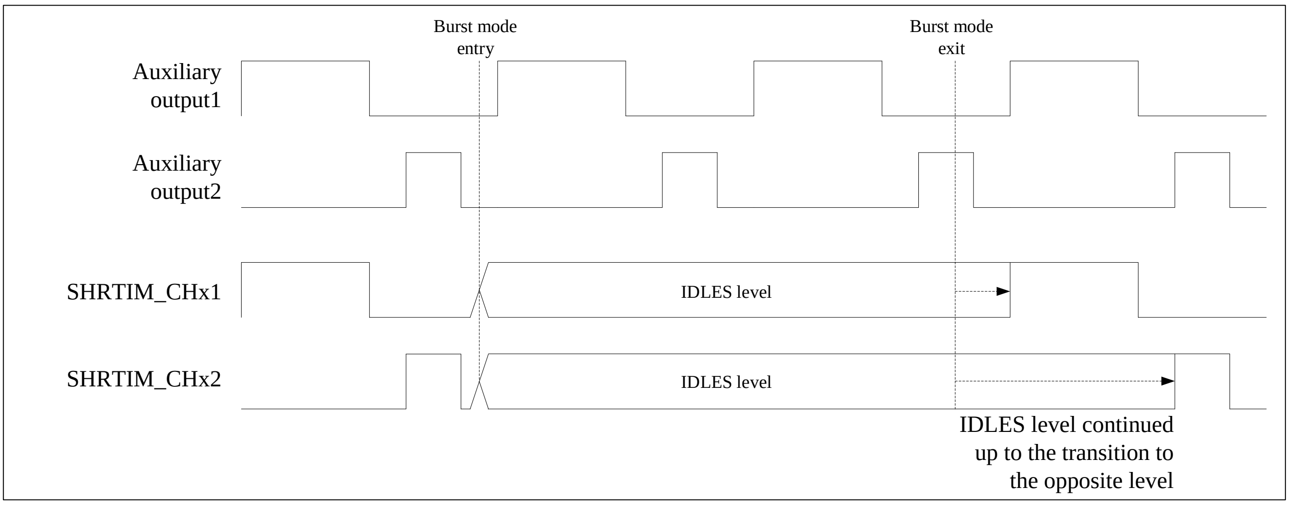
Figure9-68 Auxiliary Outputs



By default, the auxiliary outputs are copies of the outputs Tx1 and Tx2. The exceptions include:

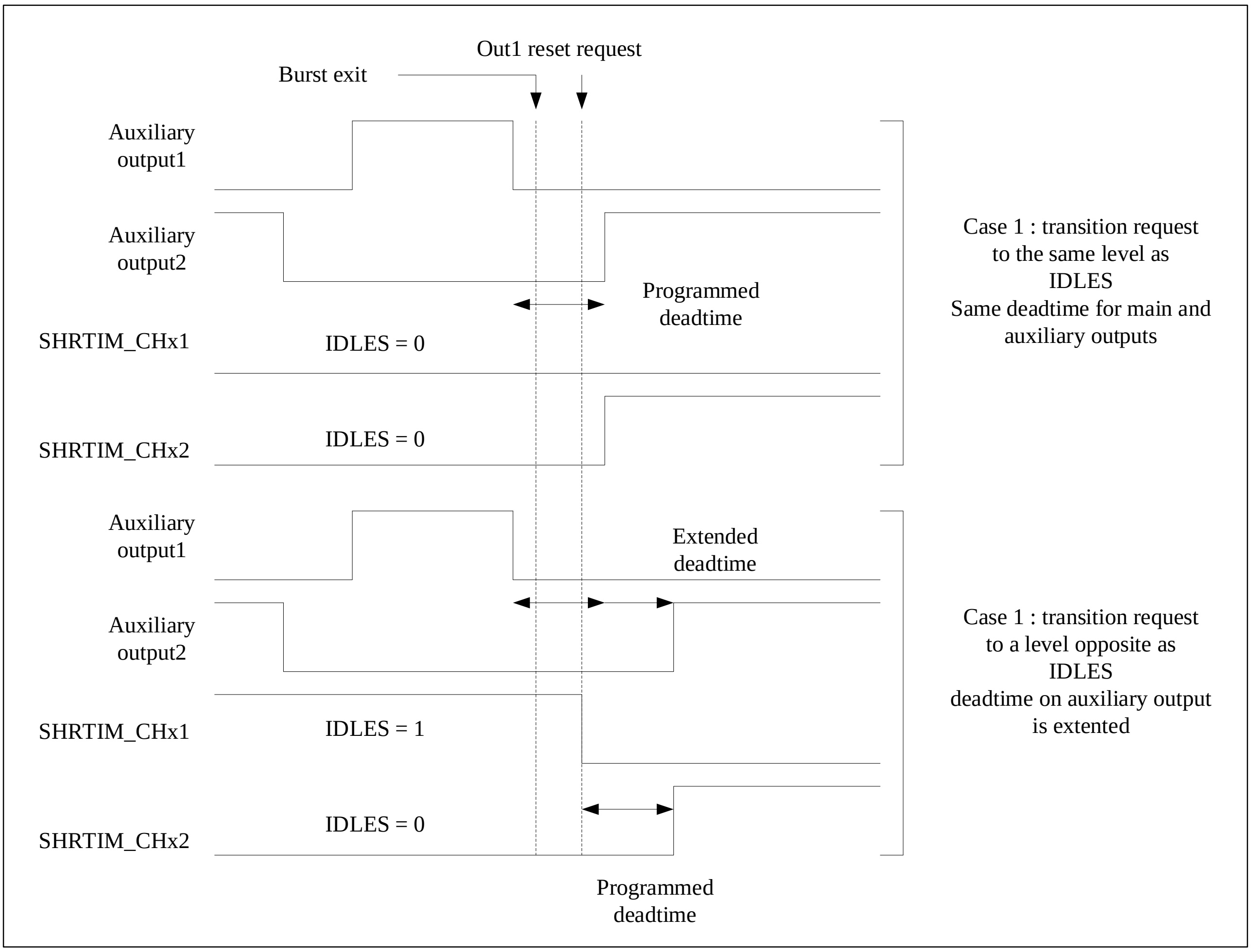
- The delayed idle and balanced idle protection when deadtime is disabled (DTEN = 0). In this case, When the protection is triggered,, the auxiliary output remains unchanged and follows the signal issued coming out of the crossbar. Conversely, if deadtime is enabled (DTEN = 1), both the main output and auxiliary output are forced to an inactive level.
- Burst mode; there are two scenarios:
  - If DTEN=0 or DIDLx=0, the auxiliary outputs are not affected by burst mode entry and continues to follow the reference signal coming out of the crossbar (refer to Figure9-69)
  - If deadtime (DTEN=1) and delayed burst mode entry (DIDLx=1) are both enabled, the behavior of the auxiliary outputs have the same behavior as the main output. They are forced to the IDLES level after the deadtime duration and will maintain this level throughout all burst periods. After the burst mode terminates, the IDLES level is maintained until a transition occurs to the opposite level, similarly to the main output.

Figure9-69 Auxiliary and Main Outputs During Burst Mode (DIDLx = 0)



When exiting from the burst mode or when re-enabling the outputs after a delayed protection, if this happens during a deadtime, the signal on the auxiliary output may be slightly distorted. In such cases, the deadtime applied to the auxiliary output is extended so that the deadtime on the main outputs is respected. The following figure provides some examples.

Figure9-70 Deadtime Distortion on Auxiliary Output When Exiting Burst Mode



9.3.19 Synchronize the SHRTIM with other timers or SHRTIM instances

The SHRTIM can act as a master unit (generating synchronization signals) or as a slave unit (waiting for synchronization triggers) to synchronize multiple instances of SHRTIM. This functionality can also be used to synchronize the SHRTIM with other timers, either external or on-chip. The synchronization circuitry is controlled inside the master timer.

9.3.19.1 Synchronization output

This section describes how the SHRTIM must be configured to synchronize external resources and act as the master unit. Four events can be selected as the source to be sent to the synchronization output by using the SYNCOSRC[1:0] bits in the SHRTIM_MCTRL register, as follows:

- 00: master timer start

  This event is generated when MCNTEN bit is set or when the timer is re-started after having reached the period value in single-shot mode. It is also generated on a reset which occurs during the counting (when CONT or RTG bits are set).

- 01: master timer compare 1 event

- 10: timer A start

  This event is generated when TACNTEN bit is set or when the counter is reset and restarts counting in response to this reset. The following counter reset events are not propagated to the synchronization output: counter roll-over in continuous mode, and discarded reset request in single-shot non-retriggerable mode. The reset is only taken into account when it occurs during the counting (CONT or RTG bits are set).

- 11: timer A compare 1 event

Note: SYNCOSRC[1:0] uniformly configure the sources for shrtim_out_sync1 and shrtim_out_sync2.

The SHRTIM_SYNCOUTxPUS[1:0] bits in the SHRTIM_SYNCOUT register specify how the synchronization event are generated.

If SYNCOUTxPUS[1:0] = 1x , the synchronization pulses are generated on the shrtim_out_syncx output pin, shrtim_out_sync1 is sent to an external pin, specifically the SHRTIM_SCOUT pin, while shrtim_out_sync2 is sent to an internal peripheral on-chip.

SYNCOUTxPUS[0] bit specifies the polarity of the synchronization signal. If SYNCOUTxPUS[0] = 0, the shrtim_out_syncx pin has a low idle level and issues a positive pulse of 16 f_{SHRTIM} clock cycles length for the synchronization). If SYNCOUTxPUS[0] = 1, the idle level is high and a negative pulse is generated.

Note: The synchronization pulse is followed by an idle level of 16 f_{SHRTIM} clock cycles during which any new synchronization request is discarded. Consequently, the maximum synchronization frequency is $f_{SHRTIM}/32$.

The idle level on the shrtim_out_syncx pin is applied as soon as the SYNCOUTxPUS[1:0] bits are enabled (i.e. the bitfield value is different from 00).

The synchronization output initialization steps must be performed before configuring the MCU output and enabling the counter, in the following order:

1. Configure the SYNCOUTxPUS[1:0] in the SHRTIM_SYNCOUT register and SYNCOSRC[1:0] fields in the SHRTIM_MCTRL register.
2. Configure the SHRTIM_SCOUT pin (refer to the GPIO and AFIO sections) or the internal peripheral connected to shrtim_out_sync2.
3. Enable the master timer or Timer A counter (set MCNTEN or TACNTEN to 1)

When the synchronization input mode is enabled and starts the counter (using SYNCSTRT/ SYNCRST bits) simultaneously with the synchronization output mode (SYNCOSRC[1:0] = 00 or 10), the output pulse is generated only when the counter is starting or is reset while running. Any reset request clearing the counter without causing it to start does not affect the synchronization output.

9.3.19.2 Synchronization input

The SHRTIM can be synchronized by an external source by programming the SYNCIN[1:0] bits in the SHRTIM_MCTRL register.

- 000: synchronization input is disabled
- 001: shrtim_in_sync[0] input (connected to the TRGO output of an on-chip timer, refer toTable 9-1)
- 010: shrtim_in_sync[1] input (connected to the TRGO output of an on-chip timer, refer to Table 9-1)

- 011: shrtim_in_sync[2] input (connected to the TRGO output of an on-chip timer, refer to Table 9-1)
- 100: a positive pulse on the SHRTIM_SCIN input pin

This bitfield cannot be changed once the destination timer (master timer or timing unit) is enabled (MCNTEN and/or TxCNTEN bit set).

The SHRTIM_SCIN input is active on the rising edge. The timer behavior is defined by the following bits in the SHRTIM_MCTRL and SHRTIM_TxCTRL registers (for detailed information, refer to Table9-33):

- Synchronous start: the incoming signal starts the timer's counter (SYNCSTRTM and/or SYNCSTRTx bits set). TxCNTEN (MCNTEN) bits must be set to have the timer enabled and the counter ready to start. In continuous mode, the counter does not start until the synchronization signal is received.
- Synchronous reset: the incoming signal resets the counter (SYNCRSTM and/or SYNCRSTx bits set). This event decrements the repetition counter as any other reset event.

The synchronization events are only considered when the respective counter is enabled (MCNTEN or TxCNTEN set to 1). A synchronization request will trigger a SYNC interrupt.

Note: If the current counter value is greater than the active period value, the synchronization start event will reset the counter.

The effect of the synchronization event depends on the timer operating mode, as summarized in Table9-33.

.

Table9-33 Effect of Sync Event Versus Timer Operating Modes

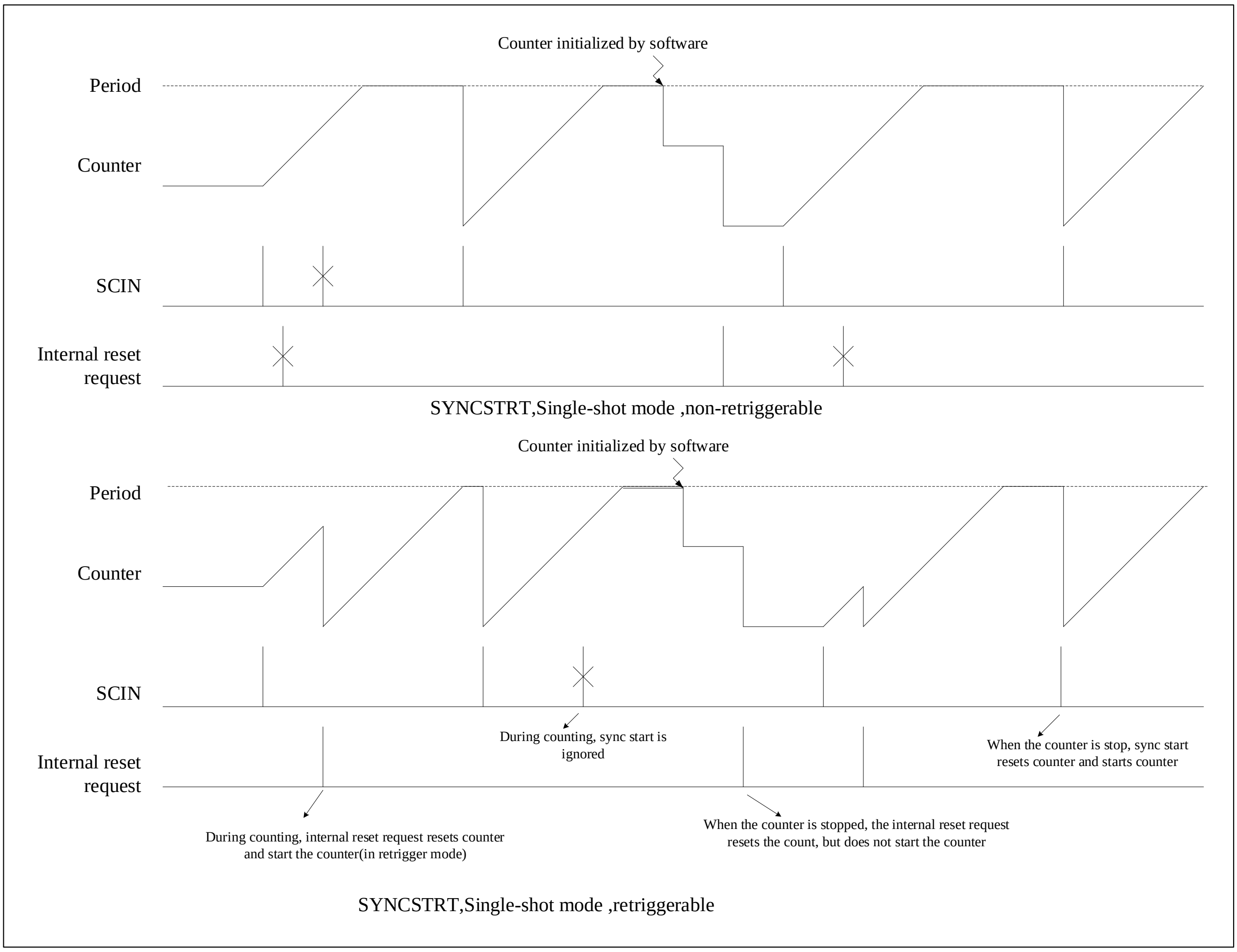
Operating Mode	SYNCRSTx	SYNCSTRTx	Behavior Following a SYNC Reset or Start Event
Single-shot non-retriggerable	0	1	Start events are taken into account when the counter is stopped and: - Once the MCNTEN or TxCNTEN bits are set - Once the period has been reached A start occurring when the counter is stopped at the period value resets the counter. A reset request clears the counter but does not start it (the counter can solely be re-started with the synchronization). Any reset occurring during the counting is ignored (as during regular non-retriggerable mode).
	1	X	Reset events are starting the timer counting. They are taken into account only if the counter is stopped and: – once the MCNTEN or TxCNTEN bits are set – once the period has been reached. When multiple reset requests are selected (from SHRTIM_SCIN and from internal events), only the first arriving request is taken into account.

Single-shot retriggerable	0	1	The counter start is effective only if the counter is not started or period is elapsed. Any synchronization event occurring after counter start has no effect. A start occurring when the counter is stopped at the period value resets the counter. A reset request clears the counter but does not start it (the counter can solely be started by the synchronization). A reset occurring during counting is taken into account (as during regular retriggerable mode).
	1	X	The reset from SHRTIM_SCIN is taken into account as any SHRTIM's timer counter reset from internal events and is starting or re-starting the timer counting. When multiple reset requests are selected, the first arriving request is taken into account.
Continuous mode	0	1	The timer is enabled (MCNTEN or TxCNTEN bit set) and is waiting for the synchronization event to start the counter. Any synchronization event occurring after the counter start has no effect (the counter can solely be started by the synchronization). A reset request clears the counter but does not start it.
	1	X	The reset from SHRTIM_SCIN is taken into account as any HRTIM's timer counter reset from internal events and is starting or re-starting the timer counting.

When a synchronization reset event occurs within the same f_{SHRTIM} clock cycle as the period event, this period is postponed to a programmed reset event (since both events are causing a counter roll-over).This applies only when the high resoulution is active (CKPSC[2:0] < 5).

The following diagram illustrates how synchronous start is performed in single-shot mode.

Figure9-71 Counter Behavior in Synchronized Start Mode


Counter initialized by software
Period
Counter
SCIN
Internal reset request
SYNCSTRT,Single-shot mode ,non-retriggerable
Counter initialized by software
Period
Counter
SCIN
Internal reset request
During counting, sync start is ignored
When the counter is stop, sync start resets counter and starts counter
During counting, internal reset request resets counter and start the counter(in retrigger mode)
When the counter is stopped, the internal reset request resets the count, but does not start the counter
SYNCSTRT,Single-shot mode ,retriggerable

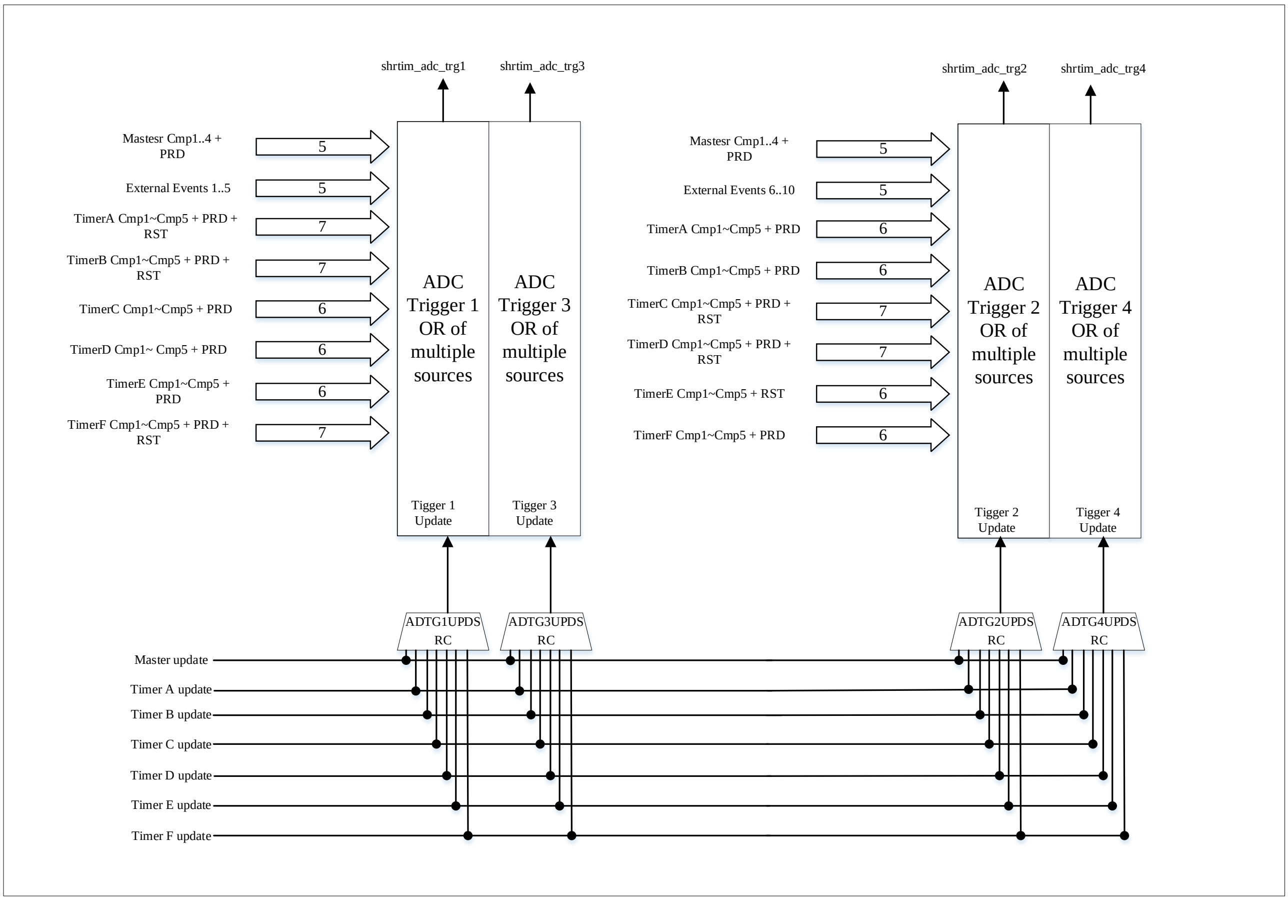
9.3.20 ADC Triggers

The ADC (Analog-to-Digital Converter) can be triggered by the master timer and the 6 timing units.

10 independent triggers are availabl for both the regular and injected sequences of the ADCs. The external events can be used as triggers. They are take right after the conditioning defined in the SHRTIM_EXEVCTRLx register, and do not depend on the settings of TxEXEVFLT1 and TxEXEVFLT2 registers.

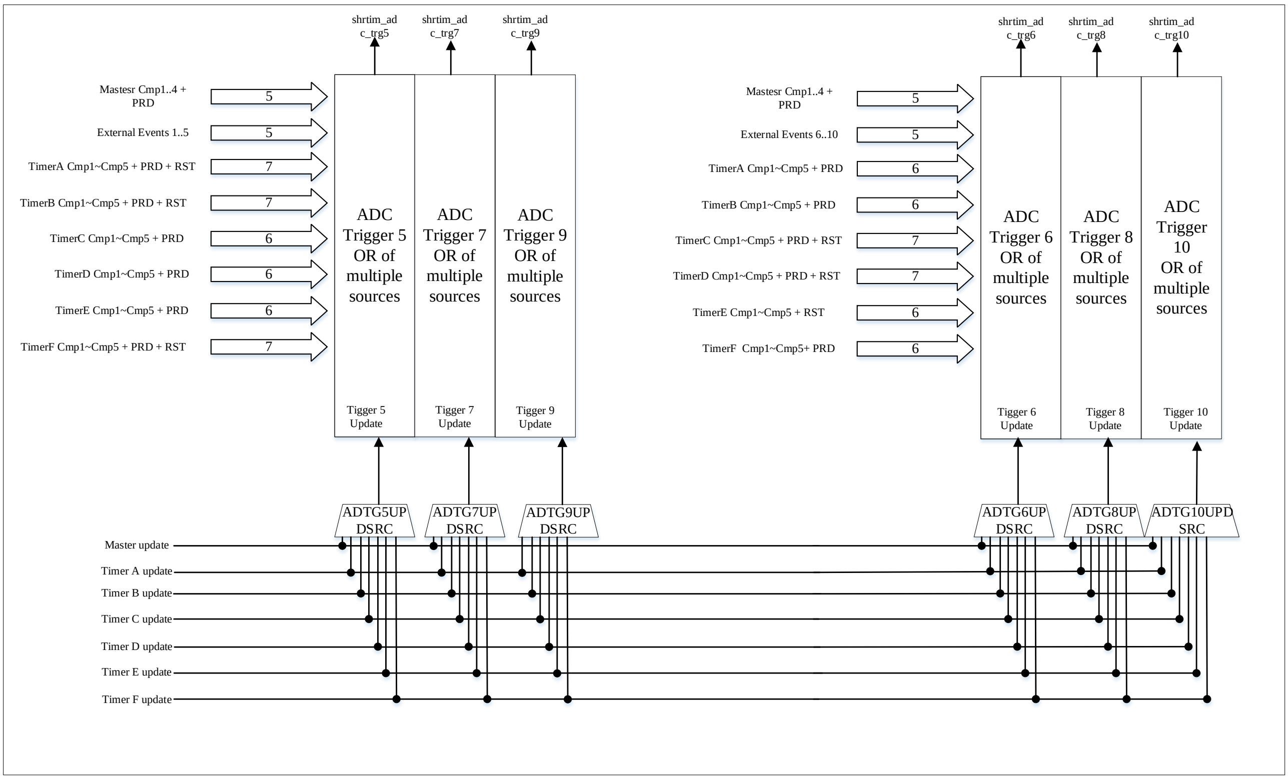
Up to 49 events can be combined (ORed) for ADC triggers 1 to 4, in the SHRTIM_ADTG1SRC1 to SHRTIM_ADTG4SRC2 registers, as shown in the diagram below. ADC triggers 1/3 and 2/4 use the same set of sources. By selecting multiple sources simultaneously, multiple triggers can be achieved in a single switching period. A typical application is for a non-overlapping multi-phase converter, where all phases can be sampled in a row using a single ADC trigger output.

Figure9-72 ADC Trigger Selection Overview



The ADC triggers 5 to 10 are configured in the SHRTIM_ADTGEX1 and SHRTIM_ADTGEX2 registers, as shown in the diagram below. The ADC triggers 5/7/9 and 6/8/10 use the same set of sources. These triggers can only select one source at a time (1 out of the 48/49 possible events).

Figure9-73 ADC Triggers



The SHRTIM_ADTG1SRC1 to SHRTIM_ADTG4SRC2 and SHRTIM_ADTGEX1 to SHRTIM_ADTGEX2 registers are preloaded and can be synchronously updated with the associated timers. The update source is defined by the ADTGxUPDSRC[2:0] bits in the SHRTIM_CTRL1 and SHRTIM_ADTGUPD registers.

For example, if ADC trigger 1 outputs the Timer A CMP2 event (SHRTIM_ADTG1SRC1 = 0x0000 0800), SHRTIM_ADTG1SRC1 is typically updated simultaneously with Timer A (ADTG1UPDSRC[2:0] = 001).

When preload is disabled in the source timer (PLEN bit reset), the SHRTIM_ADTGxSRC1 and SHRTIM_ADTGxSRC2 registers are not preloaded either: a write access will result in an immediate update of the trigger source.

9.3.20.1 ADC post-scaler

A post-scaler unit allows to reduce the ADC trigger rate, as shown in the diagram below.

Each ADC trigger rate can be individually adjusted using the ADCxPSC[4:0] bits in the SHRTIM_ADCPSC1 and SHRTIM_ADCPSC2 registers.

In center-aligned mode, the ADC trigger rate also depends on the ADCROM[1:0] bits programmed in the source timer, as shown below. The ADCROM[1:0] bits is coding for any event that can trigger the ADC: reset, roll-over (period), and compare events:

- ADCROM[1:0] = 00: event generated both during up and down-counting phases
- ADCROM[1:0] = 01: event generated during down-counting phases
- ADCROM[1:0] = 10: event generated during up-counting phases

The ADC post-scaler programming register are preloaded and can be updated on-the-fly without stopping the timers.

Figure9-74 ADC Trigger Post-Scaling in Up-Counting Mode

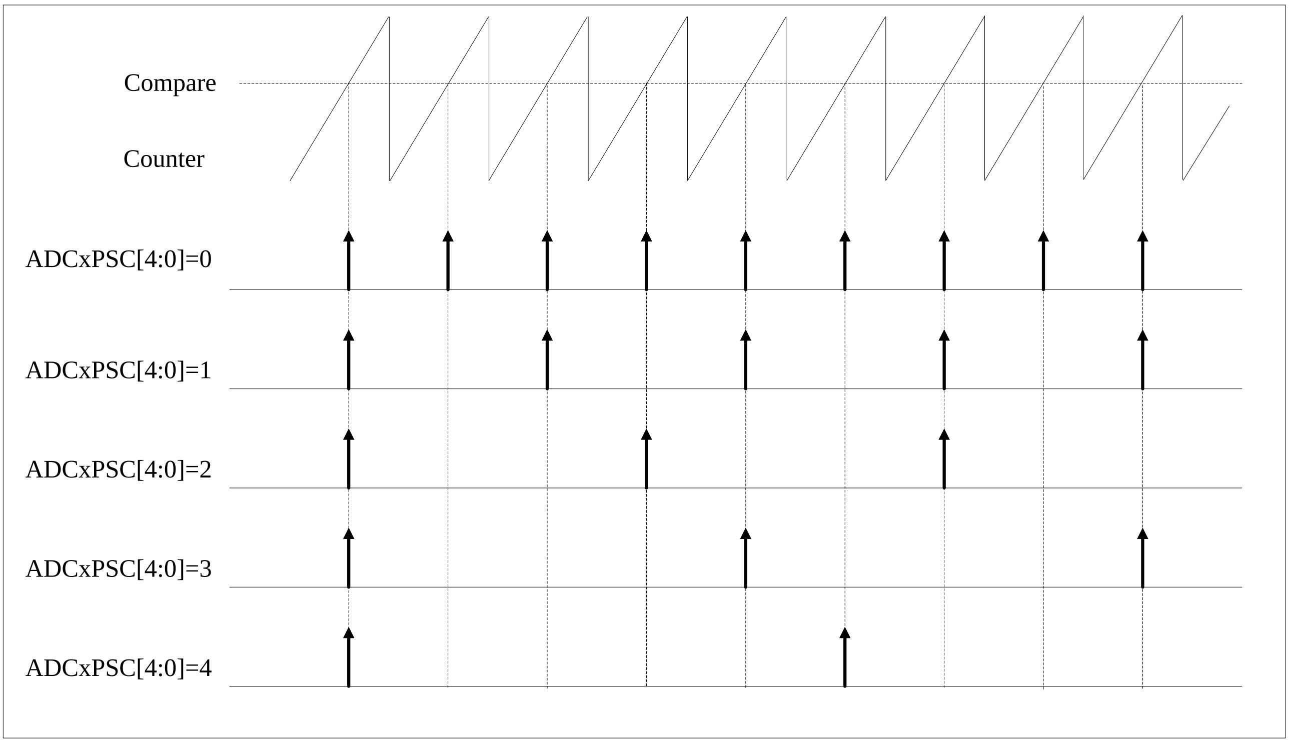
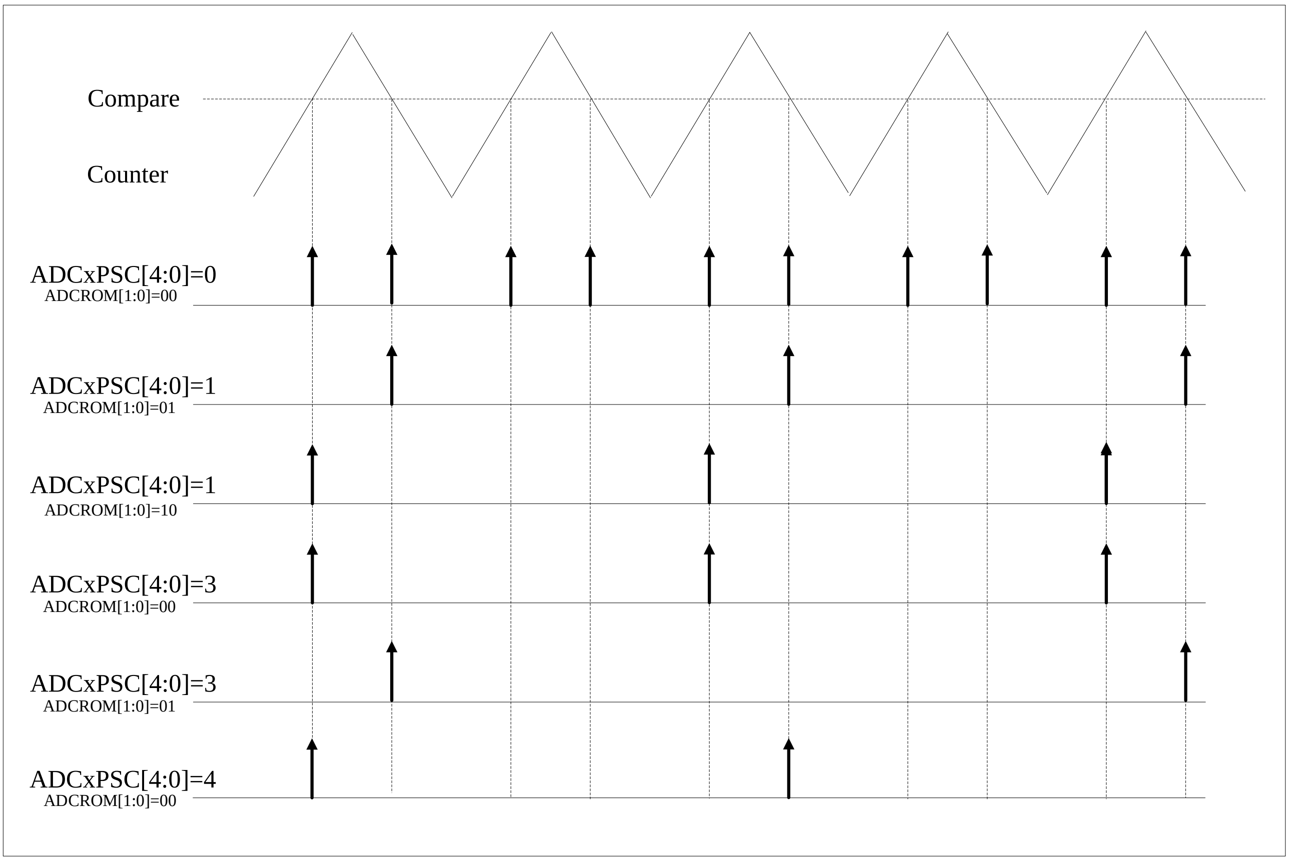


Figure9-75 ADC Trigger Post-Scaling in Up/Down Counting Mode



9.3.21 DAC Triggers

The SHRTIM allows to have the embedded DACs (Digital-to-Analog Converter) updated synchronously with the timer updates. The update events from the master timer and timing units can generate DAC update triggers on any of the shrtim_dac_trgx output.

Note: each timer has its own set of control registers related to the DAC.

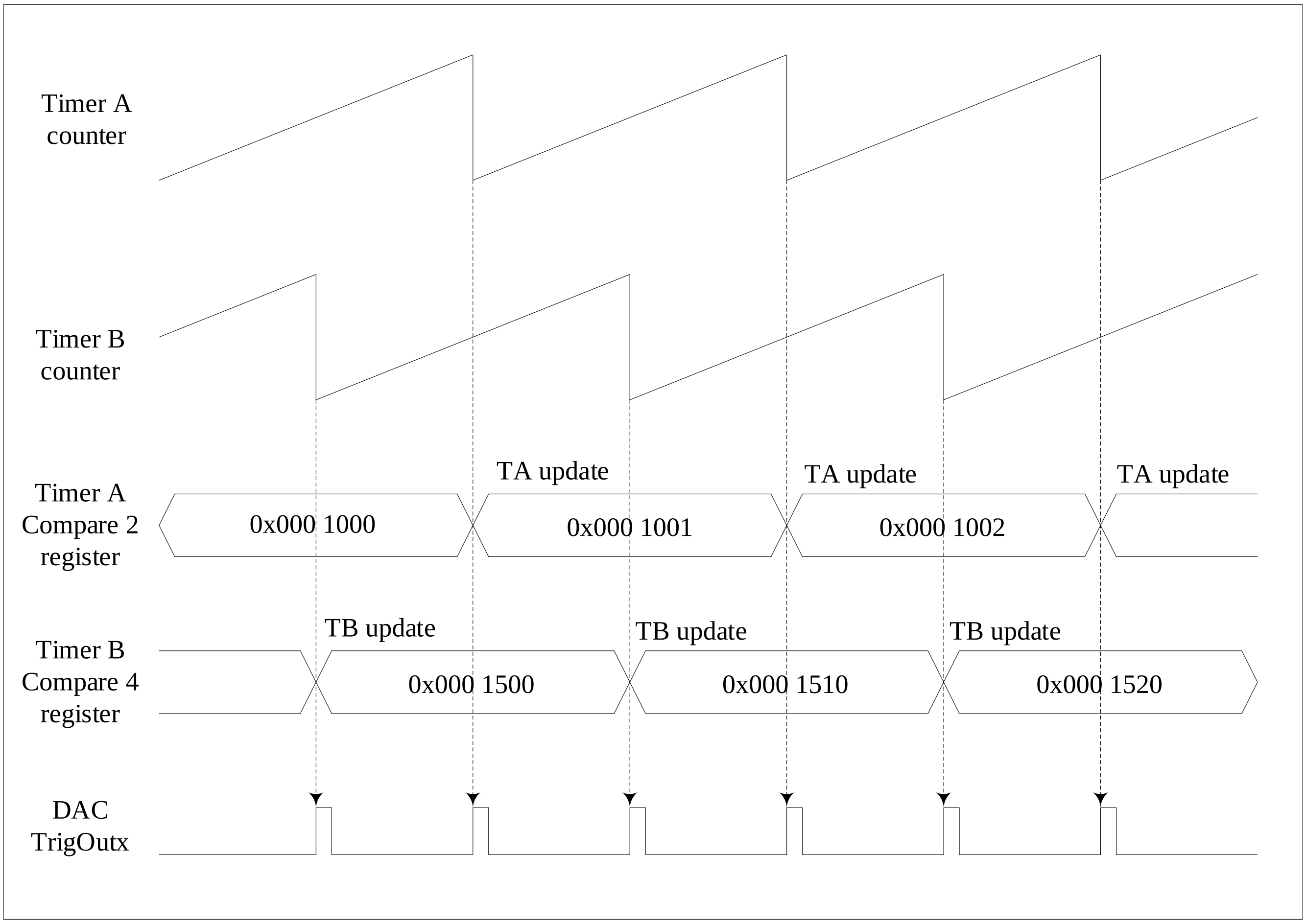
The DACTRIG[1:0] bits in the SHRTIM_MCTRL and SHRTIM_TxCTRL registers are programmed as follows:

- 00: no update generated
- 01: update generated on shrtim_dac_trg1
- 10: update generated on shrtim_dac_trg2
- 11: update generated on shrtim_dac_trg3

An output pulse of 1 f_{SHRTIM} clock periods is generated on the shrtim_dac_trgx output.

When the DACTRIG[1:0] bits are enabled in multiple timers, the shrtim_dac_trgx output is composed of the logical OR of update events from all timers. For example, if DACTRIG = 1 in Timer A and Timer B, the update events from Timer A and Timer B are logically ORed to generate the DAC update trigger on the respective shrtim_dac_trgx output, as shown in the diagram below.

Figure9-76 Combining Several Updates on a Single hrtim_dac_trgx Output



Please refer to Table 9-6 SHRTIM DAC.

Dual DAC trigger

The slope compensation techniques and hysteretic control to be easily implemented using SHRTIM built-in features and the DAC sawtooth generator. The principle is to have a DAC generating a decreasing saw-tooth synchronized with the PWM period, or a square wave synchronized with PWM signal. This mode is enabled with the DCDACEN bit in the TxCTRL2 register. This bit cannot be changed once the timer is operating (TxCNTEN bit set).

It uses two trigger outputs, as shown below:

- The shrtim_dac_reset_trgx generates DAC reset/update events
- The shrtim_dac_step_trgx generates requests for incremental DAC value changes

The DUDACRST bit in the TxCTRL2 register defines when the shrtim_dac_reset_trgx trigger is generated：

- DUDACRST = 0: the trigger is generated on counter reset or roll-over event
- DUDACRST = 1: the trigger is generated on output 1 set event

Note: the DUDACRST bit is not significant when the DCDACEN bit is reset (Dual DAC trigger disabled).

The DUDACSTEP bit in the TxCTRL2 register defines when the hrtim_dac_step_trgx trigger is generated:

- DUDACSTEP = 0: the trigger is generated on compare 2 event

- DUDACSTEP = 1: the trigger is generated on output 1 reset event

The DUDACRST and DUDACSTEP bits allows the following use cases to be covered:

- Edge-aligned slope compensation (DUDACRST = DUDACSTEP = 0): the DAC's sawtooth starts on PWM period beginning and multiple triggers are generated during the period
- Center-aligned slope compensation (DUDACRST = 1, DUDACSTEP = 0): the DAC's sawtooth starts on the output set event and multiple triggers are generated during the period
- Hysteretic controller: the DAC value must be changed twice per period, when the output state changes. 2 triggers are generated per PWM period. In edge-aligned mode (DUDACRST=0, DUDACSTEP =1), the triggers are generated on counter reset or rollover. In center-aligned mode (DUDACRST=1, DUDACSTEP=1), the triggers are generate when the output is set.

The compare 2 has a particular operating mode when the DCDACEN is set and the DUDACSTEP bit is reset. The active comparison value is automatically updated as soon as a compare match has occured, so that the trigger can be repeated periodically with a period equal to the CMP2 value, as represented on Figure9-78.

The dual DAC trigger with DUDACSTEP bit reset (compare 2 event used) must not be used simultaneously with modes using CMP2 (triple / quad interleaved and triggered-half modes).

Note: the CMP2 value can be changed on-the-fly. The new value is taken into account on the next coming compare match.

Note: When the DUDACSTEP bit is reset, the CMP2 value must not be modified by other mechanisms: the interleaved, triggered half and balanced idle modes, auto-delayed mode must be disabled.

Note: The DAC step trigger signal does not have high-resolution features. If the value of CMP2 is not divisible by 32, the actual trigger time will be at a time point before CMP2 that is divisible by 32. For example, if CMP2 is configured as 500, the actual step trigger to the DAC will occur at 480. When CMP2 steps to 1000, the actual step trigger to the DAC will occur at 992, and so on.

The table below provides an example of generating 6 triggers within a PWM period. It shows that it is necessary to round up the division result to the upper value. For example, if the timer period TxPRD = 8192. Dividing 8192 by 6 results in 1365.33.

- Round down value: 1365: 7 triggers are generated, the 6th and 7th being very close (respectively for counter = 8190 and 8192)
- Round up value:1366: 6 triggers are generated. The 6th trigger on dac_step_trg (for counter = 8192) is aborted by the counter roll-over from 8192 to 0.

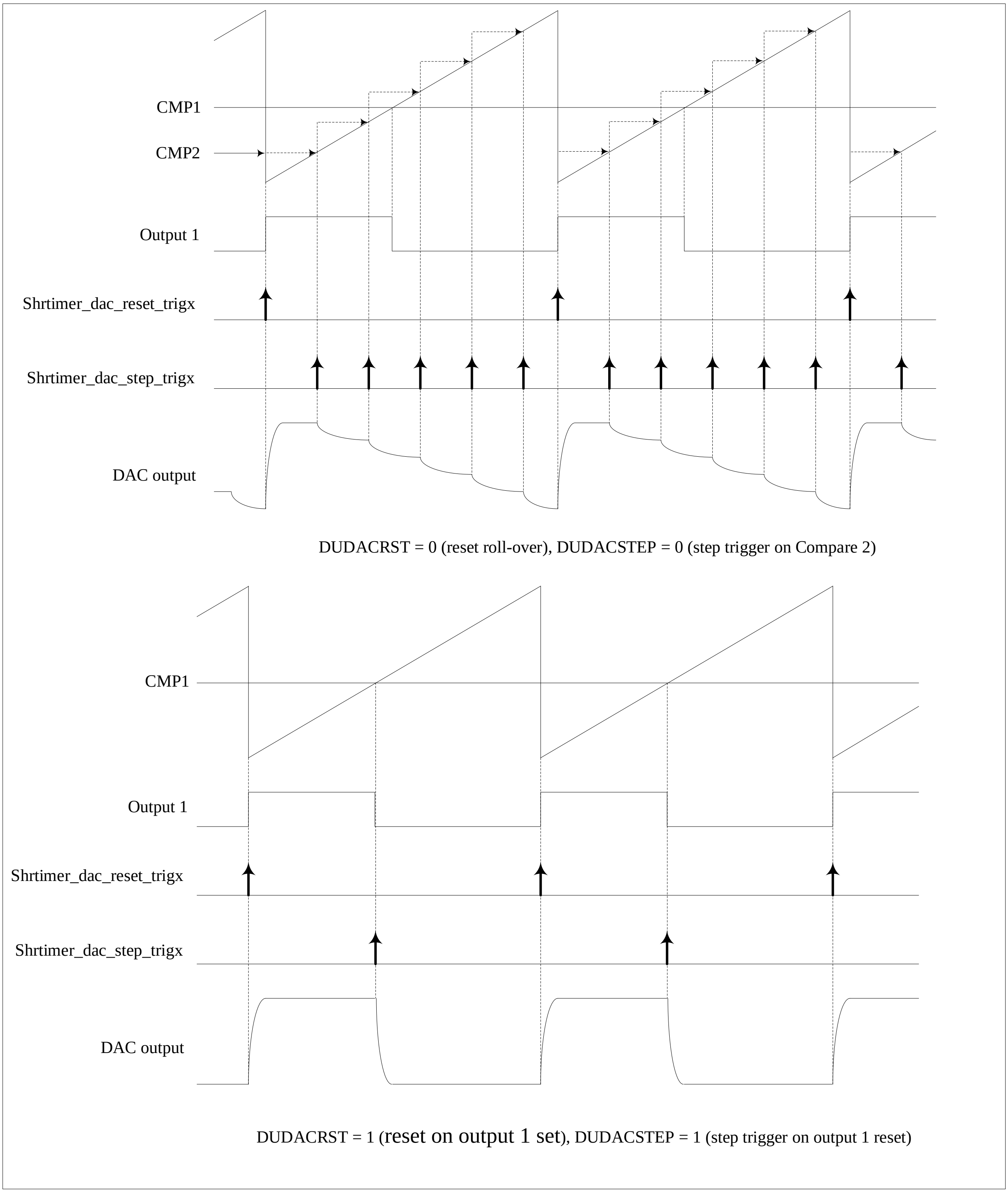
Figure9-77 Dual DAC Trigger Example

-	CMP2 = 1365	dac_trg	dac_step_trg	CMP2 = 1366	dac_trg	dac_step_trg
Counter value	1365	-	1	1366	-	1
	2730	-	2	2732	-	2
	4095	-	3	4098	-	3
	5460	-	4	5464	-	4
	6825	-	5	6830	-	5

	8190	-	6	8192	6	-
	8192	7	-	1366	-	1
	1365	-	1	2732	-	2
	...	-	-	...	-	-

Note: In center-aligned mode, it is mandatory to have an even number of triggers per switching period, so as to avoid unevenly spaced triggers around counter's peak value.

Figure9-78 DAC Triggers for Slope Compensation



DUDACRST = 0 (reset roll-over), DUDACSTEP = 0 (step trigger on Compare 2)

DUDACRST = 1 (reset on output 1 set), DUDACSTEP = 1 (step trigger on output 1 reset)

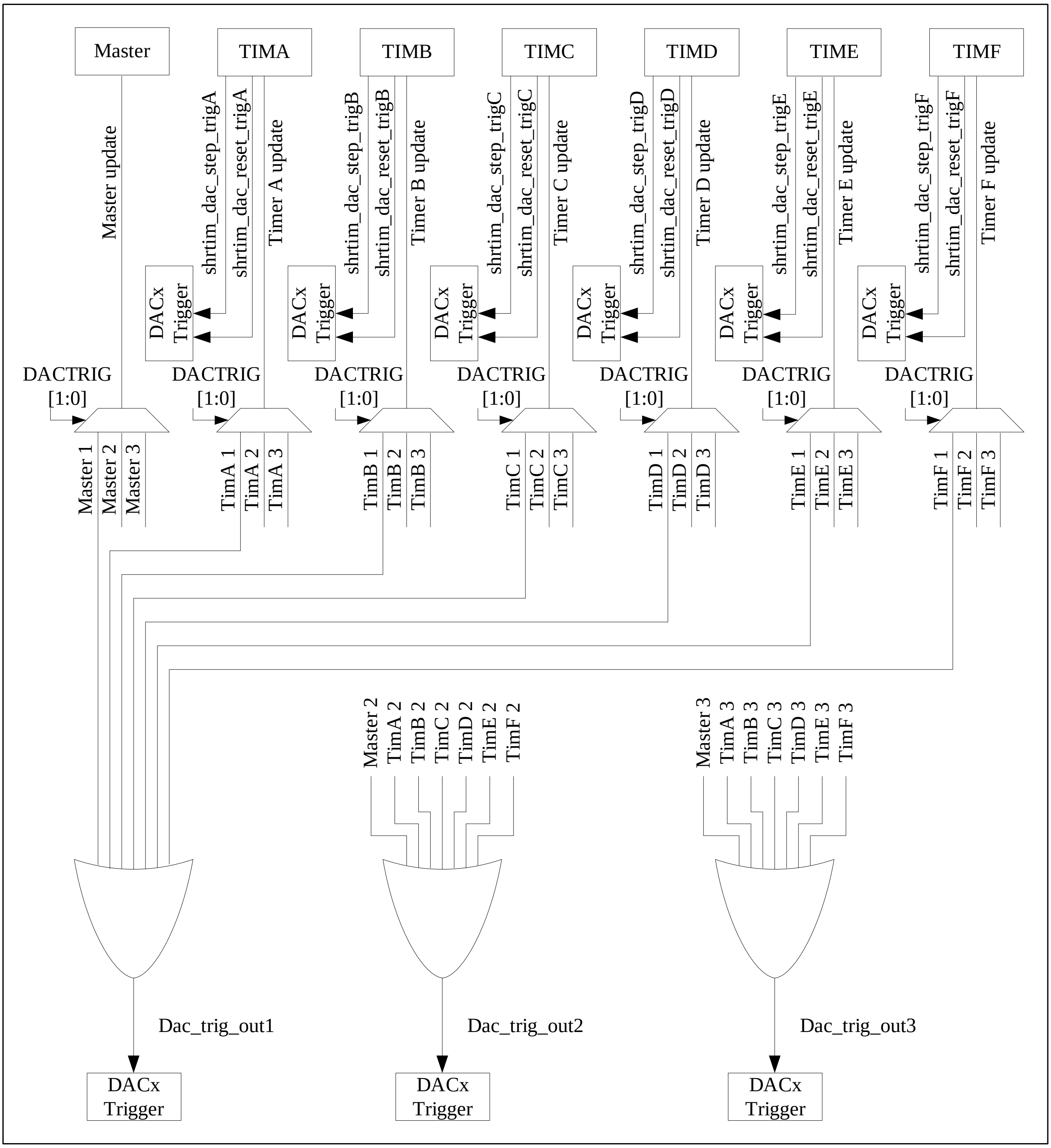
Notes:

1. Dual DAC just support in up-counting mdoe

2. Output1 set/reset is from crossbar output

The following figure provides an overview of all the available DAC triggers

Figure9-79 DAC Triggers Overview



9.3.22 SHRTIM Interrupts

The master timer can generate 7 interrupts:

- Master timer registers update
- Synchronization event received

- Master timer repetition event
- Master compare 1 to 4 event

Each timing unit can generate 14 interrupts:

- Delayed protection triggered
- Counter reset or roll-over event
- Output 1 and output 2 reset (transition active to inactive)
- Output 1 and output 2 set (transition inactive to active)
- Capture 1 and 2 events
- Timing unit registers update
- Repetition event
- Compare 1 to 4 event

7 global interrupts will be generated for the whole SHRTIM:

- System fault and fault 1 to 6 (regardless of the timing unit attribution)
- Burst mode period completed

The interrupt requests are grouped in 8 vectors as follows:

- shrtim_it1: master timer interrupts (master update, sync input, repetition, MCMP1..4) and global interrupt except faults (burst mode period)
- shrtim_it2: TIMA interrupts
- shrtim_it3: TIMB interrupts
- shrtim_it4: TIMC interrupts
- shrtim_it5: TIMD interrupts
- shrtim_it6: TIME interrupts
- shrtim_it7: TIMF interrupts
- shrtim_it8: dedicated vector all fault interrupts to allow high-priority interrupt handling

The following table is a summary of the interrupt requests, their mapping and associated control, and status bits.

Table9-34 SHRTIM Interrupt Summary

Interrupt vector	Interrupt event	Event flag	control bit Enable	Flag clearing bit
SHRTIM_global	Burst mode period completed	BMPRDITF	BMPRDIEN	BMPRDIC
	Master timer registers update	MUPDITF	MUPDIEN	MUPDIC
	Synchronization event received	SYNCINITF	SYNCINIEN	SYNCINIC
	Master timer repetition event	MREPTITF	MREPTIEN	MREPTIC

	Master compare 1 to 4 event	MCMP4ITF	MCMP4IEN	MCMP4IC
		MCMP3ITF	MCMP3IEN	MCMP3IC
		MCMP2ITF	MCMP2IEN	MCMP2IC
		MCMP1ITF	MCMP1IEN	MCMP1IC
SHRTIM_it2 SHRTIM_it3 SHRTIM_it4 SHRTIM_it5 SHRTIM_it6 SHRTIM_it7	Capture 1 and 2 events	CPT2ITF	CPT2IEN	CPT2IC
		CPT1ITF	CPT1IEN	CPT1IC
	Compare 1 to 4 event	CMP4ITF	CMP4IEN	CMP4IC
		CMP3ITF	CMP3IEN	CMP3IC
		CMP2ITF	CMP2IEN	CMP2IC
		CMP1ITF	CMP1IEN	CMP1IC
	Delayed protection triggered	DPITF	DPIEN	DPIC
	Counter reset or roll-over event	RSTROITF	RSTROIEN	RSTROIC
	Output 1 and output 2 reset (transition active to inactive)	RST2ITF	RST2IEN	RST2IC
		RST1ITF	RST1IEN	RST1IC
	Output 1 and output 2 set (transition inactive to active)	SET2ITF	SET2IEN	SET2IC
		SET1ITF	SET1IEN	SET1IC
	Timing unit registers update	UPDITF	UPDIEN	UPDIC
	Repetition event	REPTITF	REPTIEN	REPTIC
SHRTIM_fault	System fault	SYSFALTITF	SYSFALTIEN	SYSFALTIC
	Fault 1 to 6	FALT6ITF	FALT6IEN	FALT6IC
		FALT5ITF	FALT5IEN	FALT5IC
		FALT4ITF	FALT4IEN	FALT4IC
		FALT3ITF	FALT3IEN	FALT3IC
		FALT2ITF	FALT2IEN	FALT2IC
		FALT1ITF	FALT1IEN	FALT1IC

9.3.23 DMA

Most events that can generate interrupts can also generate DMA requests, and they can even generate both interrupts and DMA requests simultaneously. Each timer (master timer, TIMA...F) has its own DMA enable register.

The DMA requests are sent to the 7 channels after OR operation, as follows:

- 1 channel for the master timer
- 1 channel per timing unit (TIMA...F)

Note: before disabling a DMA channel (by resetting the DMA enable bit in TxIDEN), the DMA controller must be disabled first.

The following table is a summary of the events with their associated DMA enable bits.

Table9-35 SHRTIM DMA Request Summary

DMA Channel	Event	DMA capable	DMA enable bit
shrtim_dma1	Burst mode period completed	No	N/A

(master timer)	Master timer registers update	Yes	MUPDDEN
	Synchronization event received	Yes	SYNCINDEN
	Master timer repetition event	Yes	MREPTDEN
	Master compare 1 to 4 event	Yes	MCMP4DEN
			MCMP3DEN
			MCMP2DEN
			MCMP1DEN
shrtim_dma2 (timer A) shrtim_dma3 (timer B) shrtim_dma4 (timer C) shrtim_dma5 (timer D) shrtim_dma6 (timer E) shrtim_dma7 (timer F)	Delayed protection triggered	Yes	DPDEN
	Counter reset or roll-over event	Yes	RSTRODEN
	Output 1 and output 2 reset (transition active to inactive)	Yes	RST2DEN
		Yes	RST1DEN
	Output 1 and output 2 set (transition inactive to active)	Yes	SET2DEN
		Yes	SET1DEN
	Capture 1 and 2 events	Yes	CPT2DEN
		Yes	CPT1DEN
	Timing unit registers update	Yes	UPDEN
	Repetition event	Yes	REPTDEN
	Compare 1 to 4 event	Yes	CMP4DEN
		Yes	CMP3DEN
		Yes	CMP2DEN
		Yes	CMP1DEN
N/A	System fault	Yes	N/A
	Fault 1 to 6	Yes	N/A
	Burst mode period completed	Yes	N/A

Burst DMA transfers

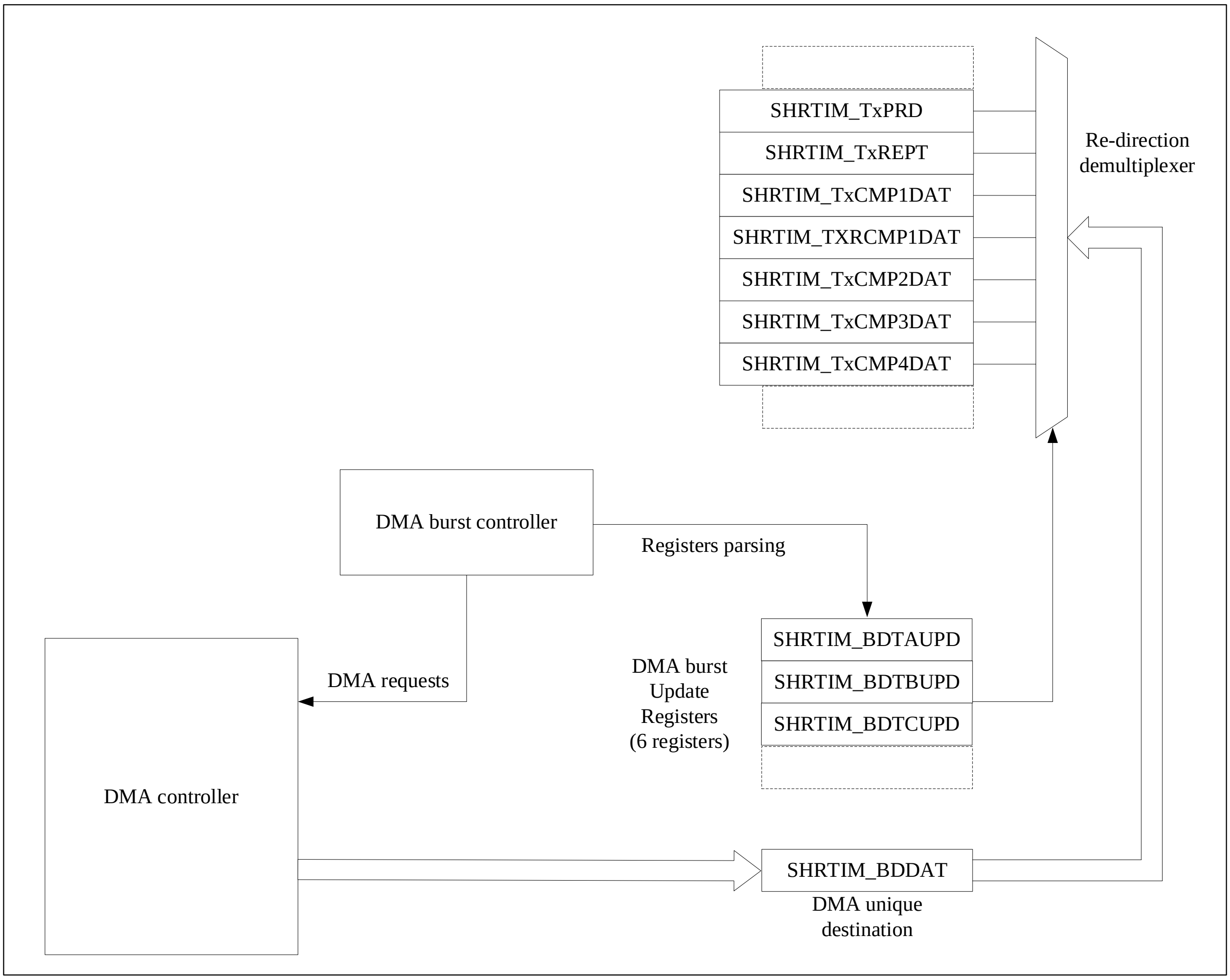
In addition to standard DMA requests, the SHRTIM is equipped with a DMA burst controller that can update multiple registers with a single DMA request. This allows to:

- Updating multiple data registers with a single DMA channel
- Dynamically reprogramming one or more timing units if the converter uses multiple timer outputs

The burst DMA feature is available for one DMA channel only, but any one of the 6 channels can be selected for burst DMA transfers.

The principle is to program which registers are to be written by DMA. The master timer and TIMA..F contain burst DMA update registers, where most control and data registers are associated with selection bits: SHRTIM_BDMTUPD, SHRTIM_BDTAUPD to SHRTIM_BDTFUPD (this is applicable only for registers with write accesses). A redirection mechanism allows to forward the DMA write accesses to the HRTIM registers automatically, as shown in the diagram below.

Figure9-80 DMA Burst Overview



When the DMA trigger occurs, the SHRTIM generates multiple 32-bit DMA requests and parses the update register. If the control bit is set, the write access is redirected to the associated register. If the bit is reset, the register update is skipped and the register parsing is resumed until a new bit set is detected, to trigger a new request. Once the 7 update registers (SHRTIM_BDMTUPD, 6x SHRTIM_BDTFUPD) are parsed, the burst is completed and the system is ready for another DMA trigger (see the flowchart on Figure9-80).

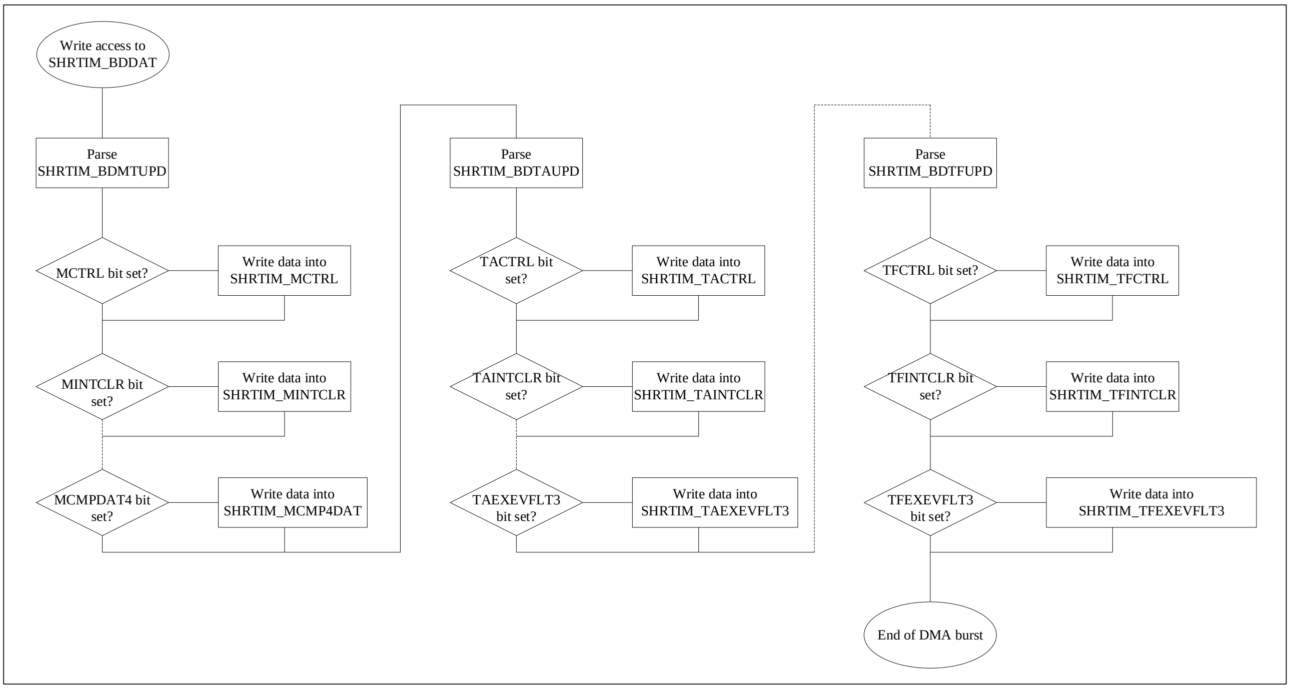
Note: Any trigger occurring while the burst is on-going is discarded, except if it occurs during the very last data transfer.

The burst DMA mode is permanently enabled (there is no enable bit). A burst DMA operation is started by the first write access into the SHRTIM_BDDAT register.

It is only necessary to have the DMA controller pointing to the SHRTIM_BDDAT register as the destination, in the memory, to the peripheral configuration with the peripheral increment mode disabled (the SHRTIM handles internally the data re-routing to the final destination register).

To re-initialize the burst DMA mode if the burst DMA mode is interrupted during the a transaction, reinitializi, it is necessary to write at least to one of the 6 update registers.

Figure9-81 Burst DMA Operation Flowchart


Write access to SHRTIM_BDDAT
Parse SHRTIM_BDMTUPD
MCTRL bit set?
Write data into SHRTIM_MCTRL
MINTCLR bit set?
Write data into SHRTIM_MINTCLR
MCMPDAT4 bit set?
Write data into SHRTIM_MCMP4DAT
Parse SHRTIM_BDTAUPD
TACTRL bit set?
Write data into SHRTIM_TACTRL
TAINTCLR bit set?
Write data into SHRTIM_TAINTCLR
TAEXEVFLT3 bit set?
Write data into SHRTIM_TAEXEVFLT3
Parse SHRTIM_BDTFUPD
TFCTRL bit set?
Write data into SHRTIM_TFCTRL
TFINTCLR bit set?
Write data into SHRTIM_TFINTCLR
TFEXEVFLT3 bit set?
Write data into SHRTIM_TFEXEVFLT3
End of DMA burst

Several options are available once the DMA burst is completed, depending on the register update strategy.

If the PLEN bit is reset (disabling preload), the values written by DMA will be immediately transferred to the active registers and the register will be update sequentially, following the DMA transaction pace..

When preload is enabled (PLEN set to 1), there are 3 use cases:

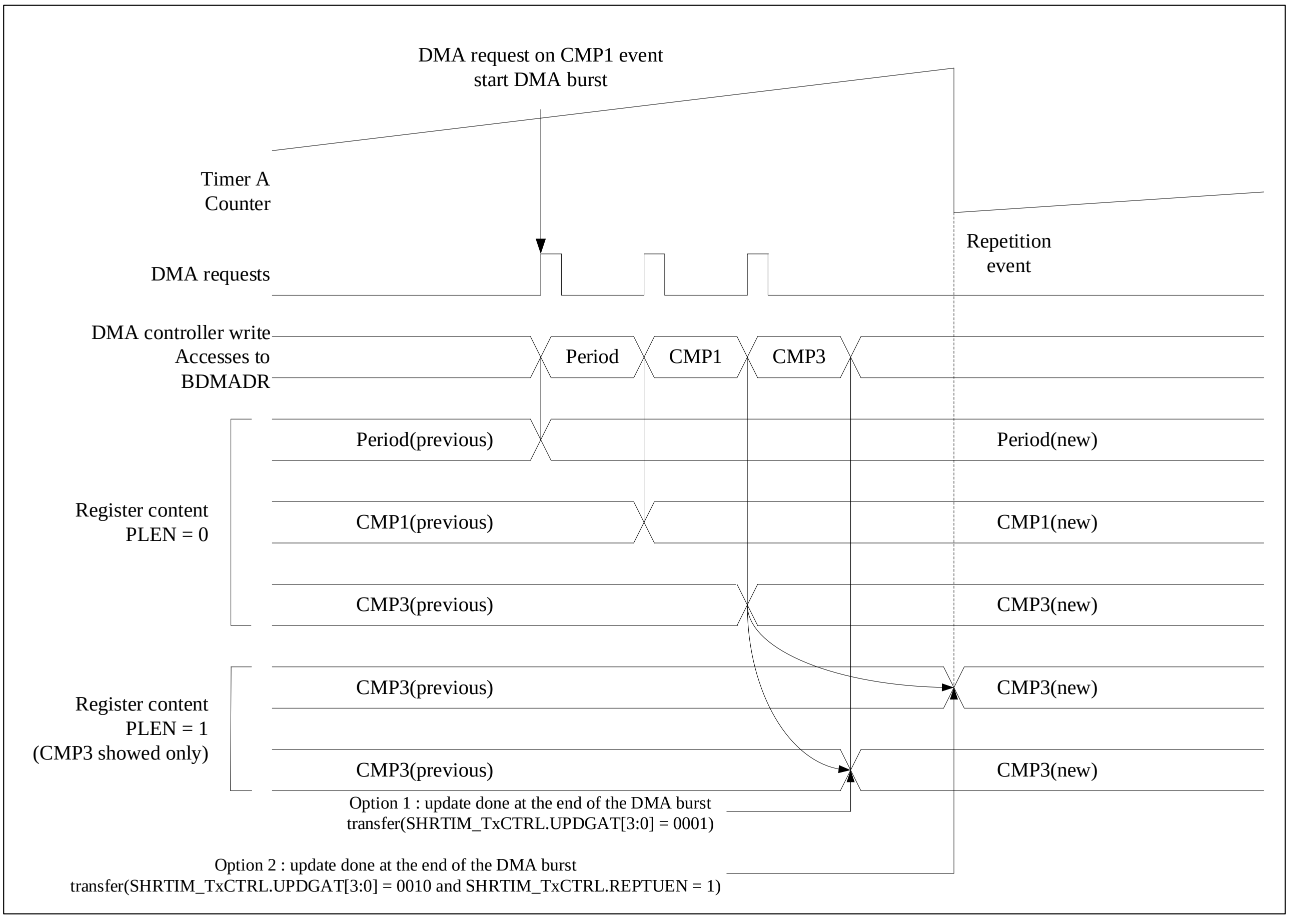
1. The update is done independently from DMA burst transfers (UPDGAT[3:0] = 0000 in SHRTIM_TxCTRL and BRSTDMA[1:0]= 00 in SHRTIM_MCTRL). In this case, and if it is necessary to have all transferred data taken into account simultaneously, the user must check that the DMA burst is completed before the update event takes place. On the contrary, if the update event happens while the DMA transfer is on-going, only part of the registers is loaded and the complete register update requires 2 consecutive update events.

2. The update is done when the DMA burst transfer is completed (UPDGAT[3:0] = 0001 in SHRTIM_TxCTRL and BRSTDMA[1:0] = 01 in SHRTIM_MCTRL). This mode guarantees that all new register values are transferred simultaneously.

3. The update is done on the update event following the DMA burst transfer completion (UPDGAT[3:0] = 0010 in SHRTIM_TxCTRL and BRSTDMA[1:0] = 10 in SHRTIM_MCTRL). This mode guarantees both a coherent update of all transferred data and the synchronization with regular update events, with the timer counter. In this case, if a regular update request occurs while the transfer is on-going, it is discarded and the effective update happens on the next coming update request.

As shown in the diagram below, it illustrates the contents of the active registers in 3 scenarios: PLEN=0, UPDGAT[3:0]=0001, and UPDGAT[3:0]=0001 (when PLEN=1).

Figure9-82 DMA Registers Update Following DMA Burst Transfer



9.3.24 SHRTIM Initialization

This section describes the recommended initialization procedure for the SHRTIM, including other relevant MCU peripherals.

The clock source for the SHRTIM must be enabled in the RCC. The f_{SHRTIM} is provided by the dedicated PLL for the SHRTIM, known as SHRTPLL, detailed in the section Reset and Clock Control (RCC) - Clock Control Unit - SHRTPLL Clock.

The SHRTIM control registers should be initialized based on the power converter topology and timing unit use cases. All inputs (sources, polarity, edge effectiveness) must be configured.

Finally, the SHRTIM outputs must be set in the following order:

- The polarity must be defined using the POLx bits in SHRTIM_TxOUT.
- The FAULT and IDLE states must be configured using the FALTx[1:0] and IDLESx bits in SHRTIM_TxOUT.

The SHRTIM output is ready to be connected to the MCU I/O. In the GPIO controller, In the GPIO controller, the selected HRTIM I/Os have to be configured as per the alternate function mapping table in the product datasheet.

From this point on, the SHRTIM will control the outputs in the IDLE state.

The outputs are configured in RUN mode by setting TxyOEN bits in the SHRTIM_OEN register. The 2 outputs are in the inactive state until the first valid set/reset event in RUN mode. Any output set/reset event (except software requests using SWT) are ignored as long as TxCNTEN bit is reset, as well as burst mode requests. Similarly, any counter reset request coming from the burst mode controller is ignored (if TxBM bit is set).

Note: If deadtime insertion is enabled (DTEN set to 1), it is necessary to force the output state through software (using the SWT and the RSTROITF bit) to ensure the output is immediately in the complementary state upon entering RUN mode.

Finally, the SHRTIM can be started by setting the TxCNTEN or MCNTEN bit in SHRTIM_MCTRL to 1.

When the SHRTIM peripheral is reset by the Reset and Clock Control, the SHRTIM outputs will enter IDLE mode and be at a low level. It is recommended to disconnect the connection between the SHRTIMER and the outputs (using the GPIO controller) before performing the peripheral reset.

9.3.25 Debug Mode

When the microcontroller enters debug mode (Cortex^{TM}-M4F core halted), the TIMx counter will continue to operate normally or stop based on the configuration of the SHRTIM1_STOP bit in the DBG_CTRL register in the DBG module.

- SHRTIM1_STOP= 0 : No change in behavior, SHRTIM continues to operate.
- SHRTIM1_STOP= 1: All SHRTIM timers (including the master timer) stop operating. If FALTx[1:0]=01, 10, 11, the outputs in RUN mode will enter the FAULT state. If FALTx[1:0]=00, the outputs will maintain their current state. Outputs in the IDLE state will remain in that state. Even when the MCU exits the stop mode, they will permanently remain in this state. This helps maintain a safe state during stepping operations. To re-enable the outputs, they can be enabled again by setting the TxyOEN bit to 1 (requires the use of a debugger).

The master timer and each timing unit can be controlled by the SHRTIM_FRZDIS register to determine whether they are affected by DBG_CTRL.SHRTIM1_STOP.

If SHRTIM_FRZDIS.ALLTIMEN=1, when DBG_CTRL.SHRTIM1_STOP is set to 1 and the microcontroller enters debug mode (Cortex^{TM}-M4F core halted), all timers is unaffected.

If only TxDBGEN is set and DBG_CTRL.SHRTIM1_STOP is set to 1, when the microcontroller enters debug mode (Cortex^{TM}-M4 core stops), only timer x is unaffected.

9.3.25.1 Timer behavior during MCU halt (when SHRTIM1_STOP= 1)

The set/reset crossbar, the deadtime and push-pull unit, the idle/balanced fault detection and all the logic driving the normal output in RUN mode are not affected by debug. The output keeps on toggling internally, so as to retrieve regular signals of the outputs when TxyOEN is set again (during or after the MCU halt). Associated triggers and filters are also following internal waveforms when the outputs are disabled.

FAULT inputs and events (any source) are enabled during the MCU halt.

Fault status bits can be set and TxyOEN bits reset during the MCU halt if a fault occurs at that time (TxyOEN and TxyODISSTS are not affected by SHRTIM1_STOP bit state).

Synchronization, counter reset, start and reset-start events are discarded in debug mode, as well as capture events. This is to keep all related registers stable as long as the MCU is halted.

The counter stops counting when a breakpoint is reached. However, the counter enable signal is not reset; consequently no start event is emitted when exiting from debug. All counter reset and capture triggers are disabled, as well as external events (ignored as long as the MCU is halted). The outputs SET and RST flags are frozen, except in case of forced software set/reset. A level-sensitive event is masked during the debug but is active again as soon as the debug is exited. For edge-sensitive events, if the signal is maintained active during the MCU halt, a new edge is not generated when exiting from debug.

The update events are discarded. This prevents any update trigger on shrtim_upd_en[3:1] inputs. The DMA triggers will be disabled. The burst mode circuit is frozen: the triggers will be ignored, and the burst mode counter will stop operating.

9.4 SHRTIM Register

9.4.1 The register of master timer of SHRTIM

9.4.1.1 SHRTIM Master Timer Control Register（SHRTIM_MCTRL）

Address offset: 0x00

Reset value: 0x0000

31 30	29	28	27	26 25	24 23	22	21	20	19	18	17	16
BRSTDMA	MREPTU	Reserved	PLEN	DACTRIG	Reserved	TFCNTE	TECNTE	TDCNTE	TCCNTE	TBCNTE	TACNTE	MCNTE
rw	rw	r	rw	rw	r	rw	rw	rw	rw	rw	rw	rw

15 14	13	12	11	10 9 8	7 6	5	4	3	2 1 0
SYNCOSRC	SYNCSTR	SYNCR	Reserv	SYNCIN	ILV	HLF	RTG	CONT	CKPSC
rw	rw	rw	r	rw	rw	rw	rw	rw	rw

Bit Field	Name	Description
[31:30]	BRSTDMA	Burst DMA Update These bits define during a burst DMA transaction how the update of registers happen 00: Update done independently from the DMA burst transfer completion 01: Update done when the DMA burst transfer is completed 10: Update done on master timer roll-over following a DMA burst transfer completion. This mode only works in continuous mode. 11: Reserved
[29]	MREPTUEN	Master Timer Repetition update This bit defines whether an update occurs when the master timer repetition period is completed (either due to roll-over or reset events). MREPTUEN can be set only if BRSTDMA[1:0] = 00 or 01. 0: Repetition event does not cause register update 1: Repetition event cause register update
[28]	Reserved	Reserved, the reset value must be maintained

[27]	PLEN	Preload enable This bit enables the registers preload mechanism and defines whether the write accesses to the memory mapped registers are done into SHRTIM's active or preload registers. 0: Preload disabled: the write access is directly done into the active register 1: Preload enabled: the write access is done into the preload register
[26:25]	DACTRIG	DAC Synchronization A DAC synchronization event can be enabled and generated when the master timer update occurs. These bits are defining on which output the DAC synchronization is sent (refer to Section 9.3.21 DAC triggers for connections details). 00: No DAC trigger generated 01: Trigger generated on shrtim_dac_trg1 10: Trigger generated on shrtim_dac_trg2 11: Trigger generated on shrtim_dac_trg3
[24:23]	Reserved	Reserved, the reset value must be maintained
[22]	TFCNTEN	Timer F counter enable 0: Timer F counter disabled 1: Timer F counter enabled *Note: This bit must not be changed within 8 cycles of* f_{SHRTIM} *clock*
[21]	TECNTEN	Timer E counter enable 0: Timer E counter disabled 1: Timer E counter enabled *Note: This bit must not be changed within 8 cycles of* f_{SHRTIM} *clock*
[20]	TDCNTEN	Timer D counter enable 0: Timer D counter disabled 1: Timer D counter enabled *Note: This bit must not be changed within 8 cycles of* f_{SHRTIM} *clock*
[19]	TCCNTEN	Timer C counter enable 0: Timer C counter disabled 1: Timer Ccounter enabled *Note: This bit must not be changed within 8 cycles of* f_{SHRTIM} *clock*
[18]	TBCNTEN	Timer B counter enable 0: Timer B counter disabled 1: Timer B counter enabled *Note: This bit must not be changed within 8 cycles of* f_{SHRTIM} *clock*
[17]	TACNTEN	Timer A counter enabl 0: Timer A counter disabled 1: Timer A counter enabled *Note: This bit must not be changed within 8 cycles of* f_{SHRTIM} *clock*
[16]	MCNTEN	Master timer counter enable 0: Master timer counter disabled 1: Master timer counter enabled

		Note: This bit must not be changed within 8 cycles of f_{SHRTIM} clock
[15:14]	SYNCOSRC	Synchronization source These bits define the source and event that will be sent on the synchronous outputs SYNCOUT2 and SYNCOUT1. 00 : Master timer start 01: Master timer compare 1 event 10: Timer A start/reset 11: Timer A compare 1 event
[13]	SYNCSTRT	Synchronization Starts Master This bit enables the master timer start when receiving a synchronization input event: 0: No effect on the master timer 1: A synchronization input event starts the master timer
[12]	SYNCRST	Synchronization Resets Master This bit enables the master timer reset when receiving a synchronization input event: 0: No effect on the master timer 1: A synchronization input event resets the master timer
[11]	Reserved	Reserved, the reset value must be maintained
[10:8]	SYNCIN	Synchronization input These bits are defining the synchronization input source. 000：SHRTIM runs in standalone mode because synchronization input have been disabled. 001: The shrtim_in_sync[0] input is driven by the TRGO of an on-chip timer 010: The shrtim_in_sync[1] input is driven by the TRGO of an on-chip timer 011: The shrtim_in_sync[2] input is driven by the TRGO of an on-chip timer 100: The shrtim_in_sync[3] input is driven by an external source via SHRTIM_SCIN(IOM) 101: Reserved 110: Reserved 111: Reserved For the cascading of shrtim_in_sync, please refer to section 9.3.19.2 Synchronous Inputs. *Note: This parameter cannot be changed once the impacted timers are enabled.*
[7:6]	ILV	Master interleaved mode 00: Interleaved mode disabled 01: Triple interleaved mode: this mode uses SHRTIM_MCMP1DAT and SHRTIM_MCMP2DAT, as well as SHRTIM_MPRD . When SHRTIM_MPRD register is written, the SHRTIM_MCMP1DAT active register is automatically updated with SHRTIM_MPRD/3 value，and the SHRTIM_MCMP2DAT active register is automatically updated with SHRTIM_MPRD*2/3 value. 10: Quad interleaved mode: this mode uses SHRTIM_MCMP1DAT, SHRTIM_MCMP2DAT and SHRTIM_MCMP3DAT, as well as

		SHRTIM_MPRD. When SHRTIM_MPRD register is written, the SHRTIM_MCMP1DAT active register is automatically updated with MPRD/4 value，the SHRTIM_MCMP2DAT active register is automatically updated with SHRTIM_MPRD/2 value, and the SHRTIM_MCMP3DAT active register is automatically updated with SHRTIM_MPRD*3/4 value. 11: Interleaved mode disabled *Note: ILV[1:0] enables the interleaved mode and this bitfield is significant only when the HLF bit is reset.*
[5]	HLF	Half mode This bit enables half-mode, the SHRTIM_MCMP1DAT active register is automatically updated with SHRTIM_MPRD/2 value when SHRTIM_MPRD register is written. 0: Half mode disabled 1: Half mode enabled
[4]	RTG	Re-triggerable mode This bit defines the behavior of the master timer counter in single-shot mode. 0: The timer is not re-triggerable: a counter reset can be done only if the counter is stopped (period elapsed) 1: The timer is re-triggerable: a counter reset is done whatever the counter state (running or stopped)
[3]	CONT	Continuous mode 0: The timer operates in single-shot mode and stops when it reaches the SHRTIM_MPRD value 1: The timer operates in continuous (free-running) mode and rolls over to zero when it reaches the SHRTIM_MPRD value
[2:0]	CKPSC	Master clock prescaler These 3 bits define the 8 possible clock prescaling ratio for the master timer. The counter clock equivalent frequency ($f_{COUNTER}$) is equal to f_{HRCK} / $2^{CKPSC[2:0]}$. *Note: The prescaling ratio cannot be modified once the timer is enabled.*

9.4.1.2 SHRTIM Master Timer Interrupt Status Register（SHRTIM_MINTSTS）

Address offset: 0x04

Reset value: 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									MUPDITF	SYNCINITF	MREPTITF	MCMP4ITF	MCMP3ITF	MCMP2ITF	MCMP1ITF

r r r r r r r

Bit Field	Name	Description
[31:7]	Reserved	Reserved, the reset value must be maintained
[6]	MUPDITF	Master Update Interrupt Flag This interrupt line is asserted each time master timer registers are updated. 0: No update register request received. 1: Update register was received. This bit is set by hardware only. This bit is only set by hardware and can be cleared by software by writing 0 to the register.
[5]	SYNCINITF	Sync Input Interrupt Flag This interrupt line is asserted each time a synchronization input is received by the SHRTIM if interrupt is enabled. 0: No synchronization input was received. 1: Synchronization input was received. This bit is set by hardware only. This bit is only set by hardware and can be cleared by software by writing 0 to the register
[4]	MREPTITF	Master Repetition Interrupt Flag This interrupt line is asserted when enabled and when the zero value for Repetition counter is reached everytime. 0: Repetition counter has not reached the zero value. 1: Repetition counter has reached the zero value causing an interrupt. This bit is only set by hardware and can be cleared by software by writing 0 to the register
[3]	MCMP4ITF	Master Compare 4 Interrupt Flag This interrupt line is asserted when enabled and when the master timer count matches compare 4 register value. 0: No match between master timer counter and compare 4 event. 1: Master timer counter matched with compare 4 event. This bit is only set by hardware and can be cleared by software by writing 0 to the register
[2]	MCMP3ITF	Master Compare 3 Interrupt Flag This interrupt line is asserted when enabled and when the master timer count matches compare 3 register value. 0: No match between master timer counter and compare 3 event. 1: Master timer counter matched with compare 3 event. This bit is only set by hardware and can be cleared by software by writing 0 to the register
[1]	MCMP2ITF	Master Compare 2 Interrupt Flag This interrupt line is asserted when enabled and when the master

		timer count matches compare 2 register value. 0: No match between master timer counter and compare 2 event. 1: Master timer counter matched with compare 2 event. This bit is only set by hardware and can be cleared by software by writing 0 to the register
[0]	MCMP1ITF	Master Compare 1 Interrupt Flag This interrupt line is asserted when enabled and when the master timer count matches compare 1 register value. 0: No match between master timer counter and compare 1 event. 1: Master timer counter matched with compare 1 event. This bit is only set by hardware and can be cleared by software by writing 0 to the register

9.4.1.3 SHRTIM Master Timer Interrupt Clear Register（SHRTIM_MINTCLR）

Address offset: 0x08

Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									MUPDIC	SYNCINIC	MREPTIC	MCMP4IC	MCMP3IC	MCMP2IC	MCMP1IC
									w	w	w	w	w	w	w

Bit Field	Name	Description
[31:7]	Reserved	Reserved, the reset value must be maintained
[6]	MUPDIC	Master update interrupt flag clear 0: SHRTIM_MINTSTS.MUPDITF holds it's value 1: SHRTIM_MINTSTS.MUPDITF value is cleared
[5]	SYNCINIC	Master sync interrupt flag clear 0: SHRTIM_MINTSTS.SYNCINITF holds it's value 1: SHRTIM_MINTSTS.SYNCINITF value is cleared
[4]	MREPTIC	Master repetition interrupt flag clear 0: SHRTIM_MINTSTS.MREPTITF holds it's value 1: SHRTIM_MINTSTS.MREPTITF value is cleared
[3]	MCMP4IC	Master compare4 interrupt flag clear 0: SHRTIM_MINTSTS.MCMP4ITF holds it's value 1: SHRTIM_MINTSTS.MCMP4ITF value is cleared
[2]	MCMP3IC	Master compare3 interrupt flag clear 0: SHRTIM_MINTSTS.MCMP3ITF holds it's value 1: SHRTIM_MINTSTS.MCMP3ITF value is cleared
[1]	MCMP2IC	Master compare2 interrupt flag clear 0: SHRTIM_MINTSTS.MCMP2ITF holds it's value

		1: SHRTIM_MINTSTS.MCMP2ITF value is cleared
[0]	MCMP1IC	Master compare1 interrupt flag clear 0: SHRTIM_MINTSTS.MCMP1ITF holds it's value 1: SHRTIM_MINTSTS.MCMP1ITF value is cleared

9.4.1.4 SHRTIM Master Timer Interrupt/DMA Enable Register（SHRTIM_MIDEN）

Address offset: 0x0C

Reset value: 0x 00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved									MUPDDEN	SYNCINDEN	MREPTDEN	MCMP4DEN	MCMP3DEN	MCMP2DEN	MCMP1DEN
									rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									MUPDIEN	SYNCINIEN	MREPTIEN	MCMP4IEN	MCMP3IEN	MCMP2IEN	MCMP1IEN
									rw	rw	rw	rw	rw	rw	rw

Bit Field	Name	Description
[31:23]	Reserved	Reserved, the reset value must be maintained
[22]	MUPDDEN	Master registers update DMA request enable Software bit to enable DMA request on Master registers update. 0: Disables the Update register generated DMA request . 1: Enables the Update register generated DMA request.
[21]	SYNCINDEN	Synchronization input DMA request enable Software bit to enable Synchronization input DMA request enable 0: Disables the Synchronization input generated DMA request 1: Enables the Synchronization input generated DMA request
[20]	MREPTDEN	Master timer DMA request enable Software bit to enable Master Timer Repetition Counter DMA request enable 0: Disables the Master Timer Repetition Counter generated DMA request 1: Enables the Master Timer Repetition Counter generated DMA request
[19]	MCMP4DEN	Master timer compare 4 DMA request enable Software bit to enable Master Timer compare 4 DMA request enable 0: Disables the Master Timer compare 4 generated DMA request 1: Enables the Master Timer compare 4 generated DMA request .
[18]	MCMP3DEN	Master timer compare 3 DMA request enable Software bit to enable Master Timer compare 3 DMA request enable 0: Disables the Master Timer compare 3 generated DMA request 1: Enables the Master Timer compare 3 generated DMA request
[17]	MCMP2DEN	Master timer compare 2 DMA request enable Software bit to enable Master Timer compare 2 DMA request enable 0: Disables the Master Timer compare 2 generated DMA request 1: Enables the Master Timer compare 2 generated DMA request .

[16]	MCMP1DEN	Master timer compare 1 DMA request enable Software bit to enable Master Timer compare 1 DMA request enable 0: Disables the Master Timer compare 1 generated DMA request 1: Enables the Master Timer compare 1 generated DMA request .
[15:7]	Reserved	Reserved, the reset value must be maintained
[6]	MUPDIEN	Master registers update interrupt request enable Software bit to enable Interrupt request on Master registers update 0: Disables the Update register generated interrupt request. 1: Enables the Update register generated interrupt request.
[5]	SYNCINIEN	Synchronization input interrupt request enable Software bit to enable Synchronization input DMA request enable 0: Disables the Synchronization input generated interrupt request 1: Enables the Synchronization input generated interrupt request
[4]	MREPTIEN	Master repetition counter interrupt request enable Software bit to enable Master Timer Repetition Counter DMA request enable 0: Disables the Master Timer Repetition Counter generated interrupt request 1: Enables the Master Timer Repetition Counter generated interrupt request
[3]	MCMP4IEN	Master timer compare 4 interrupt request enable Software bit to enable Master Timer compare 4 DMA request enable 0: Disables the Master Timer compare 4 generated interrupt request 1: Enables the Master Timer compare 4 generated interrupt request .
[2]	MCMP3IEN	Master timer compare 3 interrupt request enable Software bit to enable Master Timer compare 3 DMA request enable 0: Disables the Master Timer compare 3 generated interrupt request 1: Enables the Master Timer compare 3 generated interrupt request.
[1]	MCMP2IEN	Master timer compare 2 interrupt request enable Software bit to enable Master Timer compare 2 DMA request enable 0: Disables the Master Timer compare 2 generated interrupt request 1: Enables the Master Timer compare 2 generated interrupt request.
[0]	MCMP1IEN	Master timer compare 1 interrupt request enable Software bit to enable Master Timer compare 1 interrupt request enable 0: Disables the Master Timer compare 1 generated interrupt request 1: Enables the Master Timer compare 1 generated interrupt request

9.4.1.5 SHRTIM Master Timer Counter Register（SHRTIM_MCNT）

Address offset: 0x10

Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MCOUNT															

rw

Bit Field	Name	Description
[31:16]	Reserved	Reserved, the reset value must be maintained
[15:0]	MCOUNT	Master timer counter Holds the master timer counter value. This register can be written only when the Master Timer is disabled (SHRTIM_MCTRL.MCNTEN = 0). *Note: For SHRTIM_MCTRL.CKPSC[2:0] < 5 the least significant bits of the counter are not significant. They cannot be written and return 0 when read.* *Note: The counter value loaded into this register should never exceed the value written into SHRTIM_MPRD register value*

9.4.1.6 SHRTIM Master Timer Period Register（SHRTIM_MPRD）

Address offset: 0x14

Reset value: 0x0000FFDF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MPRD															

rw

Bit Field	Name	Description
[31:16]	Reserved	Reserved, the reset value must be maintained
[15:0]	MPRD	Master timer period This register defines the counter overflow value. The period value must be above or equal to 3 periods of the f_{SHRTIM} clock, that is 0x60 if CKPSC[2:0] = 0, 0x30 if CKPSC[2:0] = 1, 0x18 if CKPSC[2:0] = 2,... When SHRTIM_MCTRL.PLEN is set to 1, the register will configure the value to the period preload register. If SHRTIM_MCTRL.PLEN is reset, the register value will be directly loaded to the active period register.

9.4.1.7 SHRTIM Master Timer Repetition Register（SHRTIM_MREPT）

Address offset: 0x18

Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

Reserved

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								MREPT							
								rw							

Bit Field	Name	Description
[31:8]	Reserved	Reserved, the reset value must be maintained
[7:0]	MREPT	Master timer repetition period This register holds the master timer repetition period value. This register configures the value into a repetition preload register when SHRTIM_MCTRL.PLEN bit is set. If SHRTIM_MCTRL.PLEN is reset then this register value is directly loaded to the active repetition register.

9.4.1.8 SHRTIM Master Timer Compare 1 Register（SHRTIM_MCMP1DAT）

Address offset: 0x1C

Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MCMP1DAT															
rw															

Bit Field	Name	Description
[31:16]	Reserved	Reserved, the reset value must be maintained
[15:0]	MCMP1DAT	Master timer compare 1 data The compare 1 value must be greater than or equal to 3 cycles of the f_{SHRTIM} clock, i.e., if CKPSC[2:0] = 0, it is 0x60; if CKPSC[2:0] = 1, it is 0x30; if CKPSC[2:0] = 2, it is 0x18;... This register configures the value into a compare 1 preload register when SHRTIM_MCTRL.PLEN bit is set. If SHRTIM_MCTRL.PLEN is reset then this register value is directly loaded to the active compare 1 register.

9.4.1.9 SHRTIM Master Timer Compare 2 Register（SHRTIM_MCMP2DAT）

Address offset: 0x24

Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MCMP2DAT															

rw

Bit Field	Name	Description
[31:16]	Reserved	Reserved, the reset value must be maintained
[15:0]	MCMP2DAT	Master timer compare 2 data The compare 2 value must be greater than or equal to 3 cycles of the f_{SHRTIM} clock, i.e., if CKPSC[2:0] = 0, it is 0x60; if CKPSC[2:0] = 1, it is 0x30; if CKPSC[2:0] = 2, it is 0x18;... This register configures the value into a compare 2 preload register when SHRTIM_MCTRL.PLEN bit is set. If SHRTIM_MCTRL.PLEN is reset then this register value is directly loaded to the active compare 2 register.

9.4.1.10 SHRTIM Master Timer Compare 3 Register（SHRTIM_MCMP3DAT）

Address offset: 0x28

Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MCMP3DAT															

rw

Bit Field	Name	Description
[31:16]	Reserved	Reserved, the reset value must be maintained
[15:0]	MCMP3DAT	Master timer compare 3 data The compare 3 value must be greater than or equal to 3 cycles of the f_{SHRTIM} clock, i.e., if CKPSC[2:0] = 0, it is 0x60; if CKPSC[2:0] = 1, it is 0x30; if CKPSC[2:0] = 2, it is 0x18;... This register configures the value into a compare 3 preload register when SHRTIM_MCTRL.PLEN bit is set. If SHRTIM_MCTRL.PLEN is reset then this register value is directly loaded to the active compare 3 register.

9.4.1.11 SHRTIM Master Timer Compare 4 Register（SHRTIM_MCMP4DAT）

Address offset: 0x2C

Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

MCMP4DAT
rw

Bit Field	Name	Description
[31:16]	Reserved	Reserved, the reset value must be maintained
[15:0]	MCMP4DAT	Master timer compare 4 data The compare 4 value must be greater than or equal to 3 cycles of the f_{SHRTIM} clock, i.e., if CKPSC[2:0] = 0, it is 0x60; if CKPSC[2:0] = 1, it is 0x30; if CKPSC[2:0] = 2, it is 0x18;... This register configures the value into a compare 4 preload register when SHRTIM_MCTRL.PLEN bit is set. If SHRTIM_MCTRL.PLEN is reset then this register value is directly loaded to the active compare 4 register.

9.4.1.12 SHRTIM Synchronous Output Register（SHRTIM_SYNCOUT）

Address offset: 0x30

Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										SYNCOUT2PUS		Reserved		SYNCOUT1PUS	
										rw				rw	

Bit Field	Name	Description
[31:8]	Reserved	Reserved, the reset value must be maintained
[5:4]	SYNCOUT2PUS	Synchronization output shrtim_out_sync2 to internal peripherals These bits define the routing and conditioning of the synchronization output event. 00: Disabled the synchronization output 01: Reserved. 10: Generate a positive pulse of 16 SHRTIM clock cycles on shrtim_out_sync2 11: Generate a negative pulse of 16 SHRTIM clock cycles on shrtim_out_sync2 *Note: This bitfield must not be modified once the counter is enabled (SHRTIM_MCTRL.TxCNTEN bit set)*
[3:2]	Reserved	Reserved, the reset value must be maintained
[1:0]	SYNCOUT1PUS	Synchronization output shrtim_out_sync1 to SHRTIMx_SCOUT pin These bits define the routing and conditioning of the synchronization output event. 00: Disabled the synchronization output 01: Reserved.

		10: Generate a positive pulse of 16 SHRTIM clock cycles on SHRTIMx_SCOUT 11: Generate a negative pulse of 16 SHRTIM clock cycles on SHRTIMx_SCOUT *Note: This bitfield must not be modified once the counter is enabled (SHRTIM_MCTRL.TxCNTEN bit set)*

9.4.1.13 SHRTIM Debug Freeze Disable Register（SHRTIM_FRZDIS）

Address offset: 0x34

Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								ALLTIM	TFDBG	TEDBG	TDDBG	TCDBG	TBDBG	TADBG	MDBG
								rw	rw	rw	rw	rw	rw	rw	rw

Bit Field	Name	Description
[31:8]	Reserved	Reserved, the reset value must be maintained
[7]	ALLTIMEN	DBG_CTRL.SHRTIM1_STOP controls all timers (master timer + timing units) 0: When DBG_CTRL.SHRTIM1_STOP is set to 1, all timers stop operating when the microcontroller enters debug mode (Cortex^{TM}-M4F core halts). 1: When DBG_CTRL.SHRTIM1_STOP is set to 1, all timers are not affected when the microcontroller enters debug mode (Cortex^{TM}-M4F core halts).
[6]	TFDBGEN	Refer to the description of TADBGEN.
[5]	TEDBGEN	Refer to the description of TADBGEN.
[4]	TDDBGEN	Refer to the description of TADBGEN.
[3]	TCDBGEN	Refer to the description of TADBGEN.
[2]	TBDBGEN	Refer to the description of TADBGEN.
[1]	TADBGEN	Skip the control of Timer A by DBG_CTRL.SHRTIM1_STOP. 0: When DBG_CTRL.SHRTIM1_STOP is set to 1, Timer A stops operating when the microcontroller enters debug mode (Cortex^{TM}-M4F core halts). 1: When DBG_CTRL.SHRTIM1_STOP is set to 1, Timer A is not affected when the microcontroller enters debug mode (Cortex^{TM}-M4F core halts).
[0]	MDBGEN	Skip the control of Master Timer by DBG_CTRL.SHRTIM1_STOP. 0: When DBG_CTRL.SHRTIM1_STOP is set to 1, the master timer stops operating when the microcontroller enters debug mode (Cortex^{TM}-M4F core halts). 1: When DBG_CTRL.SHRTIM1_STOP is set to 1, the master timer is not affected when the microcontroller enters debug mode (Cortex^{TM}-M4F core halts).

9.4.2 The registers of timer unit of SHRTIM

9.4.2.1 SHRTIM Timer x Control Register（SHRTIM_TxCTRL）

Address offset: TIMA: 0x080

Address offset: TIMB: 0x100

Address offset: TIMC: 0x180

Address offset: TIMD: 0x200

Address offset: TIME: 0x280

Address offset: TIMF: 0x300

Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UPDGAT				PLEN	DACTRIG		MUE	TEUE	TDUE	TCUE	TBUE	Reserv	RSTROU	REPTUE	TFUE
rw				rw	rw		rw	rw	rw	rw	rw		rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DELCMP4M		DELCMP2M		SYNCST	SYNCR	RSYNCU	ILV		PP	HLF	RTG	CONT	CKPSC		
rw		rw		rw	rw	rw	rw		rw	rw	rw	rw	rw		

Bit Field	Name	Description
[31:28]	UPDGAT	Update Gating Updating the registers could share a relation with the DMA burst along with the input pins shrtim_upd_en[3:1]. The configuration register bits below show what this dependency is. 0000: The update occurs independently from the DMA burst transfer 0001: The update occurs when the DMA burst transfer is completed 0010: The update occurs on the update event following the DMA burst transfer completion 0011: The update occurs on a rising edge on shrtim_upd_en1 0100: The update occurs on a rising edge on shrtim_upd_en2 0101: The update occurs on a rising edge on shrtim_upd_en3 0110: The update occurs on the update event following a rising edge on shrtim_upd_en1 0111: The update occurs on the update event following a rising edge on shrtim_upd_en2 1000: The update occurs on the update event following a rising edge on shrtim_upd_en3 Others: Reserved
[27]	PLEN	Timer x preload enable This bit enables the registers preload mechanism and defines whether a write access into a preloadable register is done into the active or the preload register.

		0: Preload register access disabled: the write access is directly done into the active register 1: Preload register access enabled: the write access is done into the preload register
[26:25]	DACTRIG	DAC Synchronization trig A DAC synchronization event is generated when the timer update occurs. These bits are defining on which output the DAC synchronization is sent (refer to Section 9.3.21 for DAC triggers for connections details). 00 : No DAC trigger is selected 01 : Trigger generated on shrtim_dac_trig1 10 : Trigger generated on shrtim_dac_trig2 11 : Trigger generated on shrtim_dac_trig3
[24]	MUEN	Master timer update Register Update is triggered by Master timer update 0 : Master timer as register update trigger is disabled 1 : Master timer as register update trigger is enabled
[23]	TEUEN	It is available for SHRTIM_TACTRL 、SHRTIM_TBCTRL 、SHRTIM_TCCTRL、SHRTIM_TDCTRL、SHRTIM_TFCTRL: TIM E update Register Update is triggered by timer E update 0 : timer E as register update trigger is disabled 1 : timer E as register update trigger is enabled
[22]	TDUEN	It is available for SHRTIM_TACTRL 、SHRTIM_TBCTRL 、SHRTIM_TCCTRL、SHRTIM_TECTRL、SHRTIM_TFCTRL: TIM D update Register Update is triggered by timer D update 0 : timer D as register update trigger is disabled 1 : timer D as register update trigger is enabled
[21]	TCUEN	It is available for SHRTIM_TACTRL 、SHRTIM_TBCTRL 、SHRTIM_TDCTRL、SHRTIM_TECTRL、SHRTIM_TFCTRL: TIM C update Register Update is triggered by timer C update 0 : timer C as register update trigger is disabled 1 : timer C as register update trigger is enabled
[20]	TBUEN	It is available for SHRTIM_TACTRL 、SHRTIM_TCCTRL 、SHRTIM_TDCTRL、SHRTIM_TECTRL、SHRTIM_TFCTR: TIM B update Register Update is triggered by timer B update 0 : timer B as register update trigger is disabled 1 : timer B as register update trigger is enabled
[19]	TAUEN	It is available for SHRTIM_TBCTRL 、SHRTIM_TCCTRL 、SHRTIM_TDCTRL、SHRTIM_TECTRL、SHRTIM_TFCTRL: TIM A update Register Update is triggered by timer A update 0 : timer A as register update trigger is disabled 1 : timer A as register update trigger is enabled
[18]	RSTROUEN	Timerx reset update Register update is triggered by timer x counter reset or roll-over to 0 after

		reaching the period value. 0: Update by timer x reset / roll-over disabled 1: Update by timer x reset / roll-over enabled
[17]	REPTUEN	Timer x Repetition update Register update is triggered when the counter rolls over and SHRTIM_TxREPT = 0 0: Update on repetition disabled 1: Update on repetition enabled
[16]	TFUEN	It is available for SHRTIM_TACTRL 、 SHRTIM_TBCTRL 、 SHRTIM_TCCTRL、SHRTIM_TDCTRL、SHRTIM_TECTRL: TIM F update Register Update is triggered by timer F update 0 : timer F as register update trigger is disabled 1 : timer F as register update trigger is enabled
[15:14]	DELCMP4M	CMP4 auto-delayed mode Following bits determine if the SHRTIM_TxCMP4DAT behaves in standard mode or in various Auto-Delayed mode 00: SHRTIM_TxCMP4DAT is in regular compare mode 01: SHRTIM_TxCMP4DAT is in Auto Delayed mode with internal SHRTIM_TxCMP4DAT = SHRTIM_TxCPT.CPT2 register + original SHRTIM_TxCMP4DAT value. If capture 2 event failed to occur before SHRTIM_TxPRD, then the comparison is canceled until reset/rollover to next period and no CMP4 event occurs. 10: SHRTIM_TxCMP4DAT is in Auto Delayed mode with internal SHRTIM_TxCMP4DAT = SHRTIM_TxCPT.CPT2 register + original SHRTIM_TxCMP4DAT value. If capture 2 event is yet to happen meanwhile a timeout feature is implemented by waiting till the count reaches (SHRTIM_TxCMP4DAT + SHRTIM_TxCMP1DAT). The CMP4 event is issued when the match happens. 11: SHRTIM_TxCMP4DAT is in Auto Delayed mode with internal SHRTIM_TxCMP4DAT = SHRTIM_TxCPT.CPT2 register + original SHRTIM_TxCMP4DAT value. If capture 2 event is yet to happen meanwhile a timeout feature is implemented by waiting till the count reaches (SHRTIM_TxCMP4DAT + SHRTIM_TxCMP3DAT). The CMP4 event is issued when the match happens.
[13:12]	DELCMP2M	CMP2 auto-delayed mode Following bits determine if the SHRTIM_TxCMP2DAT behaves in standard mode or in various Auto-Delayed mode 00: SHRTIM_TxCMP2DAT is in regular compare mode 01: SHRTIM_TxCMP2DAT is in Auto Delayed mode with internal SHRTIM_TxCMP2DAT = SHRTIM_TxCPT.CPT1 register + original SHRTIM_TxCMP2DAT value. If capture 1 event failed to occur before SHRTIM_TxPRD, then the comparison is canceled until reset/rollover to next period and no CMP2 event occurs.

		10: SHRTIM_TxCMP2DAT is in Auto Delayed mode with internal SHRTIM_TxCMP2DAT = SHRTIM_TxCPT.CPT1 register + original SHRTIM_TxCMP2DAT value. If capture 1 event is yet to happen meanwhile a timeout feature is implemented by waiting till the count reaches (SHRTIM_TxCMP2DAT + SHRTIM_TxCMP1DAT). The CMP2 event is issued when the match happens. 11: SHRTIM_TxCMP2DAT is in Auto Delayed mode with internal SHRTIM_TxCMP2DAT = SHRTIM_TxCPT.CPT1 register + original SHRTIM_TxCMP2DAT value. If capture 1 event is yet to happen meanwhile a timeout feature is implemented by waiting till the count reaches (SHRTIM_TxCMP2DAT + SHRTIM_TxCMP3DAT). The CMP2 event is issued when the match happens.
[11]	SYNCSTRT	Synchronization Starts Timer x This bit defines the timer x behavior following the synchronization event: 0: No effect on timer x 1: A synchronization input event starts the timer x
[10]	SYNCRST	Synchronization Resets Timer x This bit defines the timer x behavior following the synchronization event: 0: No effect on timer x 1: A synchronization input event resets the timer x
[9]	RSYNCUPD	Update on resynchronization This bit specifies whether update source coming outside from the timing unit must be synchronized: 0: The update coming from adjacent timers (when MUEN, TAUEN, TBUEN, TCUEN, TDUEN, TEUEN, TFUEN bit is set) or from a software update (TxSWUPD bit) is taken into account immediately 1: The update coming from adjacent timers (when MUEN, TAUEN, TBUEN, TCUEN, TDUEN, TEUEN, TFUEN bit is set) or from a software update (TxSWUPD bit) is taken into account on the following reset/roll-over event. *Note: This bit is significant only when UPDGAT[3:0] = 4'b0000, it is ignored otherwise.*
[8:7]	ILV	Interleaved mode This bitfield is significant only when the HLF bit is reset. It enables the interleaved mode. 00: Interleaved mode disabled 01: Triple interleaved mode: when SHRTIM_TxPRD register is written, the SHRTIM_TxCMP1DAT active register is automatically updated with SHRTIM_TxPRD/3 value, and the SHRTIM_TxCMP2DAT active register is automatically updated with 2x (SHRTIM_TxPRD/3) value. 10: Quad interleaved mode: when SHRTIM_TxPRD register is written, the SHRTIM_TxCMP1DAT active register is automatically updated with SHRTIM_TxPRD/4 value, the SHRTIM_TxCMP2DAT active register is automatically updated with SHRTIM_TxPRD/2 value and the

		SHRTIM_TxCMP3DAT active register is automatically updated with 3x (SHRTIM_TxPRD/4) value. 11: Interleaved mode disabled
[6]	PP	Push-Pull mode enable This bit enables the push-pull mode. 0: Push-pull mode disabled 1: Push-pull mode enabled Note: This bitfield must not be modified once the counter is enabled (*TxCNTEN* bit set).
[5]	HLF	Half mode enable This bit enables the half duty-cycle mode: the SHRTIM_TxCMP1DAT active register is automatically updated with SHRTIM_TxPRD/2 value when SHRTIM_TxPRD register is written. 0: Half mode disabled 1: Half mode enabled
[4]	RTG	Re-triggerable mode This bit defines the counter behavior in single shot mode. 0: The timer is not re-triggerable: a counter reset is done if the counter is stopped (period elapsed in single-shot mode or counter stopped in continuous mode) 1: The timer is re-triggerable: a counter reset is done whatever the counter state
[3]	CONT	Continuous mode 0: This setting configures timer x in single shot mode and stops counting on reaching SHRTIM_TxPRD. 1: Timer x works continuosly in Free run mode and rolls over to zero each time SHRTIM_TxPRD is reached.
[2:0]	CKPSC	Timer x clock prescaler These 3 bits define the 8 possible prescale clock factors for Timer x. The effective counting clock frequency ($f_{COUNTER}$) is equal to $F_{HRCK}/2^{PSC[2:0]}$. The prescaling frequency must be made final before enabling the timer.

9.4.2.2 SHRTIM Timer x Interrupt Status Register（SHRTIM_TxINTSTS）

Address offset: TIMA: 0x084

Address offset: TIMB: 0x104

Address offset: TIMC: 0x184

Address offset: TIMD: 0x204

Address offset: TIME: 0x284

Address offset: TIMF: 0x304

Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

Reserved	O2BCK	O1BCK	O2DIPS	O1DIPS	IPPSTS	CPPST
	r	r	r	r	r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserv	DPIT	RSTROI	RST2I	SET2I	RST1I	SET1I	CPT2I	CPT1I	UPDI	REPTIT	CMP5IT	CMP4IT	CMP3IT	CMP2I	CMP1I
	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

Bit Field	Name	Description
[31:22]	Reserved	Reserved, the reset value must be maintained
[21]	O2BCKUP	Timer x channel 2 output backup This status bit is a raw copy of the output 2 state, before the output stage (chopper, polarity). It allows to check the current output state before re-enabling the output after a delayed protection. 0: Output 2 is inactive 1: Output 2 is active
[20]	O1BCKUP	Timer x channel 1 output backup This status bit is a raw copy of the output 1 state, before the output stage (chopper, polarity). It allows to check the current output state before re-enabling the output after a delayed protection. 0: Output 1 is inactive 1: Output 1 is active
[19]	O2DIPSTS	Timer x channnel 2 delayed idle protection status This status bit indicates the output 2 state when the delayed idle protection was triggered. This bit is updated upon any new delayed protection entry. This bit is not updated in balanced idle. 0: Output 2 was inactive 1: Output 2 was active
[18]	O1DIPSTS	Timer x channnel 1 delayed idle protection status This status bit indicates the output 1 state when the delayed idle protection was triggered. This bit is updated upon any new delayed protection entry. This bit is not updated in balanced idle. 0: Output 1 was inactive 1: Output 1 was active
[17]	IPPSTS	Idle Push Pull Status This status bit indicates on which output the signal was applied, in push-pull mode, balanced fault mode or delayed idle mode, when the protection was triggered (whatever the output state, active or inactive). 0: Protection occurred when the output 1 was active and output 2 forced inactive 1: Protection occurred when the output 2 was active and output 1 forced inactive
[16]	CPPSTS	Current Push Pull Status This status bit indicates on which output the signal is currently applied, in push-pull mode. It is only significant in this configuration.

		0: Signal applied on output 1 and output 2 forced inactive 1: Signal applied on output 2 and output 1 forced inactive
[15]	Reserved	Reserved, the reset value must be maintained
[14]	DPITF	Delayed protection interrupt flag 0: No delayed protection interrupt occured 1: Delayed idle or balanced Idle mode entry occurred causing interrupt
[13]	RSTROITF	Reset and/or roll-over Interrupt Flag This bit is set by hardware when the timer x counter is reset or rolls over in continuous mode. 0: No TIMx counter reset/roll-over interrupt occurred 1: TIMX counter reset/roll-over interrupt occurred
[12]	RST2ITF	Output 2 reset interrupt flag Refer to the RST1ITF description.
[11]	SET2ITF	Output 2 set interrupt flag Refer to the SET1ITF description.
[10]	RST1ITF	Output 1 reset interrupt flag This bit is set by hardware when the Tx1 output is reset (goes from active to inactive mode). 0: No Tx1 output reset interrupt occurred 1: Tx1 output reset interrupt occurred
[9]	SET1ITF	Output 1 set interrupt flag This bit is set by hardware when the Tx1 output is set (goes from inactive to active mode). 0: No Tx1 output set interrupt occurred 1: Tx1 output set interrupt occurred
[8]	CPT2ITF	Capture 2 interrupt flag Refer to the CPT1ITF description.
[7]	CPT1ITF	Capture 1 interrupt flag This bit is set by hardware when the timer x capture 1 event occurs. 0: No timer x capture 1 interrupt occurred 1: Timer x capture 1 interrupt occurred
[6]	UPDITF	Update interrupt flag This bit is set by hardware when the timer x update event occurs. 0: No timer x update interrupt occurred 1: Timer x update interrupt occurred
[5]	REPTITF	Repetition interrupt flag This bit is set by hardware when the timer x repetition period has elapsed. 0: No timer x repetition interrupt occurred 1: Timer x repetition interrupt occurred
[4]	CMP5ITF	Compare 5 flag Refer to the CMP1ITF description.
[3]	CMP4ITF	Compare 4 flag Refer to the CMP1ITF description.

[2]	CMP3ITF	Compare 3 flag Refer to the CMP1ITF description.
[1]	CMP2ITF	Compare 2 flag Refer to the CMP1ITF description.
[0]	CMP1ITF	Compare 1 flag This bit is set by hardware when the timer x counter matches the value programmed in the compare 1 register. 0: No compare 1 interrupt occurred 1: Compare 1 interrupt occurred

9.4.2.3 SHRTIM Timer x Interrupt Clear Register（SHRTIM_TxINTCLR）

Address offset: TIMA: 0x088

Address offset: TIMB: 0x108

Address offset: TIMC: 0x188

Address offset: TIMD: 0x208

Address offset: TIME: 0x288

Address offset: TIMF: 0x308

Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	DPIC	RSTROIC	RST2IC	SET2IC	RST1IC	SET1IC	CPT2IC	CPT1IC	UPDIC	REPTIC	CMP5IC	CMP4IC	CMP3IC	CMP2IC	CMP1IC
	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

Bit Field	Name	Description
[31:15]	Reserved	Reserved, the reset value must be maintained
[14]	DPIC	Delayed protection interrupt flag clear Writing 1 to this bit clears the SHRTIM_TxINTSTS.DPIF
[13]	RSTROIC	Reset and/or roll-over interrupt flag clear Writing 1 to this bit clears the SHRTIM_TxINTSTS.RSTROITF
[12]	RST2IC	Output channel 2 reset interrupt flag clear Writing 1 to this bit clears the SHRTIM_TxINTSTS.RST2ITF
[11]	SET2IC	Output channel 2 set interrupt flag clear Writing 1 to this bit clears the SHRTIM_TxINTSTS.SET2ITF
[10]	RST1IC	Output channel 1 reset interrupt flag clear Writing 1 to this bit clears the SHRTIM_TxINTSTS.RST1ITF
[9]	SET1IC	Output channel 1 set interrupt flag clear Writing 1 to this bit clears the SHRTIM_TxINTSTS.SET1ITF

[8]	CPT2IC	Capture 2 interrupt flag clear Writing 1 to this bit clears the SHRTIM_TxINTSTS.CPT2ITF
[7]	CPT1IC	Capture 1 interrupt flag clear Writing 1 to this bit clears the SHRTIM_TxINTSTS.CPT1ITF
[6]	UPDIC	Update interrupt flag clear Writing 1 to this bit clears the SHRTIM_TxINTSTS.UPDITF
[5]	REPTIC	Repetition interrupt flag clear Writing 1 to this bit clears the SHRTIM_TxINTSTS.REPTITF
[4]	CMP5IC	Compare 5 flag clear Writing 1 to this bit clears the SHRTIM_TxINTSTS.CMP5ITF
[3]	CMP4IC	Compare 4 interrupt flag clear Writing 1 to this bit clears the SHRTIM_TxINTSTS.CMP4ITF
[2]	CMP3IC	Compare 3 interrupt flag clear Writing 1 to this bit clears the SHRTIM_TxINTSTS.CMP3ITF
[1]	CMP2IC	Compare 2 interrupt flag clear Writing 1 to this bit clears the SHRTIM_TxINTSTS.CMP2ITF
[0]	CMP1IC	Compare 1 interrupt flag clear Writing 1 to this bit clears the SHRTIM_TxINTSTS.CMP1ITF

9.4.2.4 SHRTIM Timer x Interrupt/DMA Enable Register（SHRTIM_TxIDEN）

Address offset: TIMA: 0x08C

Address offset: TIMB: 0x10C

Address offset: TIMC: 0x18C

Address offset: TIMD: 0x20C

Address offset: TIME: 0x28C

Address offset: TIMF: 0x30C

Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved	DPDEN	RSTRODEN	RST2DEN	SET2DEN	RST1DEN	SET1DEN	CPT2DEN	CPT1DEN	UPDDEN	Reserved	REPTDEN	CMP4DEN	CMP3DEN	CMP2DEN	CMP1DEN
	rw	rw	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	DPIEN	RSTROIEN	RST2IEN	SET2IEN	RST1IEN	SET1IEN	CPT2IEN	CPT1IEN	UPDIEN	REPTIEN	Reserved	CMP4IEN	CMP3IEN	CMP2IEN	CMP1IEN
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw	rw

Bit Field	Name	Description
[31]	Reserved	Reserved, the reset value must be maintained
[30]	DPDEN	Delayed protection DMA request enable This bit is set and cleared by software to enable/disable DMA requests on delayed protection.

		0: Delayed protection DMA request disabled 1: Delayed protection DMA request enabled
[29]	RSTRODEN	Reset/roll-over DMA request enable This bit is set and cleared by software to enable/disable DMA requests on timer x counter reset or roll-over in continuous mode. 0: Timer x counter reset/roll-over DMA request disabled 1: Timer x counter reset/roll-over DMA request enabled
[28]	RST2DEN	Output 2 reset DMA request enable Refer to the RST1DEN description.
[27]	SET2DEN	Output 2 set DMA request enable Refer to the SET1DEN description.
[26]	RST1DEN	Output 1 reset DMA request enable This bit is set and cleared by software to enable/disable Tx1 output reset DMA requests. 0: Tx1 output reset DMA request disabled 1: Tx1 output reset DMA request enabled
[25]	SET1DEN	Output 1 reset DMA request enable This bit is set and cleared by software to enable/disable Tx1 output set DMA requests. 0: Tx1 output set DMA request disabled 1: Tx1 output set DMA request enabled
[24]	CPT2DEN	Capture 2 DMA request enable Refer to the CPT1DEN description.
[23]	CPT1DEN	Capture 1 DMA request enable This bit is set and cleared by software to enable/disable capture 1 DMA requests. 0: Capture 1 DMA request disabled 1: Capture 1 DMA request enabled
[22]	UPDDEN	Update DMA request enable This bit is set and cleared by software to enable/disable DMA requests on update event. 0: Update DMA request disabled 1: Update DMA request enabled
[21]	Reserved	Reserved, the reset value must be maintained
[20]	REPTDEN	Repetition DMA request enable This bit is set and cleared by software to enable/disable DMA requests on repetition event. 0: Repetition DMA request disabled 1: Repetition DMA request enabled
[19]	CMP4DEN	Compare 4 DMA request enable Refer to the CMP1DEN description.
[18]	CMP3DEN	Compare 3 DMA request enable Refer to the CMP1DEN description.

[17]	CMP2DEN	Compare 2 DMA request enable Refer to the CMP1DEN description.
[16]	CMP1DEN	Compare 1 DMA request enable This bit is set and cleared by software to enable/disable the compare 1 DMA requests. 0: Compare 1 DMA request disabled 1: Compare 1 DMA request enabled
[15]	Reserved	Reserved, the reset value must be maintained
[14]	DPIEN	Delayed protection interrupt enable This bit is set and cleared by software to enable/disable interrupts on delayed protection. 0: Delayed protection interrupts disabled 1: Delayed protection interrupts enabled
[13]	RSTROIEN	Reset/roll-over interrupt enable This bit is set and cleared by software to enable/disable interrupts on timer x counter reset or rollover in continuous mode. 0: Timer x counter reset/roll-over interrupt disabled 1: Timer x counter reset/roll-over interrupt enabled
[12]	RST2IEN	Output 2 reset interrupt enable Refer to the RST1IEN description.
[11]	SET2IEN	Output 2 set interrupt enable Refer to the SET1IEN description.
[10]	RST1IEN	Output 1 reset interrupt enable This bit is set and cleared by software to enable/disable Tx1 output reset interrupts. 0: Tx1 output reset interrupts disabled 1: Tx1 output reset interrupts enabled
[9]	SET1IEN	Output 2 set interrupt enable This bit is set and cleared by software to enable/disable Tx1 output set interrupts. 0: Tx1 output set interrupts disabled 1: Tx1 output set interrupts enabled
[8]	CPT2IEN	Capture 2 interrupt enable Refer to the CPT1IEN description.
[7]	CPT1IEN	Capture 1 interrupt enable This bit is set and cleared by software to enable/disable capture 1 interrupts. 0: Capture 1 interrupts disabled 1: Capture 1 interrupts enabled
[6]	UPDIEN	Update interrupt enable This bit is set and cleared by software to enable/disable update event interrupts. 0: Update interrupts disabled 1: Update interrupts enabled

[5]	REPTIEN	Repetition interrupt enable This bit is set and cleared by software to enable/disable repetition event interrupts. 0: Repetition interrupts disabled 1: Repetition interrupts enabled
[4]	Reserved	Reserved, the reset value must be maintained
[3]	CMP4IEN	Compare 4 interrupt enable Refer to the CMP1IEN description.
[2]	CMP3IEN	Compare 3 interrupt enable Refer to the CMP1IEN description.
[1]	CMP2IEN	Compare 2 interrupt enable Refer to the CMP1IEN description.
[0]	CMP1IEN	Compare 1 interrupt enable This bit is set and cleared by software to enable/disable the compare 1 interrupts. 0: Compare 1 interrupt disabled 1: Compare 1 interrupt enabled

9.4.2.5 SHRTIM Timer x Counter Register（SHRTIM_TxCNT）

Address offset: TIMA: 0x090

Address offset: TIMB: 0x110

Address offset: TIMC: 0x190

Address offset: TIMD: 0x210

Address offset: TIME: 0x290

Address offset: TIMF: 0x310

Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COUNT															
rw															

Bit Field	Name	Description
[31:16]	Reserved	Reserved, the reset value must be maintained
[15:0]	COUNT	Timer x counter value Holds the timer x counter value. This register can be written only when the timer x is disabled (SHRTIM_MCTRL.TxCNTEN=0). *Note: For CKPSC[2:0] < 5 the least significant bits of the counter are not*

		significant. They cannot be written and return 0 when read. *Note: The count loaded into this register should never exceed the value written into PRD register value*

9.4.2.6 SHRTIM Timer x Period Register（SHRTIM_TxPRD）

Address offset: TIMA: 0x094

Address offset: TIMB: 0x114

Address offset: TIMC: 0x194

Address offset: TIMD: 0x214

Address offset: TIME: 0x294

Address offset: TIMF: 0x314

Reset value: 0x0000FFDF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRD															
rw															

Bit Field	Name	Description
[31:16]	Reserved	Reserved, the reset value must be maintained
[15:0]	PRD	Timer x period value The period value must be above or equal to 3 periods of the f_{SHRTIM} clock, that is 0x60 if SHRTIM_TxCTRL.CKPSC[2:0] = 0, 0x30 if SHRTIM_TxCTRL.CKPSC[2:0] = 1, 0x18 if SHRTIM_TxCTRL.CKPSC[2:0] = 2,... When PLEN is set to 1, the register will configure the value to the period preload register. If PLEN is reset, the register value will be directly loaded to the active period register.

9.4.2.7 SHRTIM Timer x Repetition Register（SHRTIM_TxREPT）

Address offset: TIMA: 0x098

Address offset: TIMB: 0x118

Address offset: TIMC: 0x198

Address offset: TIMD: 0x218

Address offset: TIME: 0x298

Address offset: TIMF: 0x318

Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								REPT							
								rw							

Bit Field	Name	Description
[31:8]	Reserved	Reserved, the reset value must be maintained
[7:0]	REPT	Timer x repetition period value This register holds the timer x repetition period value. This register configures the value into a repetition preload register when PLEN bit is set. If PLEN is reset then this register value is directly loaded to the active repetition register.

9.4.2.8 SHRTIM Timer x Compare 1 Register（SHRTIM_TxCMP1DAT）

Address offset: TIMA: 0x09C

Address offset: TIMB: 0x11C

Address offset: TIMC: 0x19C

Address offset: TIMD: 0x21C

Address offset: TIME: 0x29C

Address offset: TIMF: 0x31C

Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMP1DAT															
rw															

Bit Field	Name	Description
[31:16]	Reserved	Reserved, the reset value must be maintained
[15:0]	CMP1DAT	Timer x compare 1 data The compare 1 value must be above or equal to 3 periods of the f_{SHRTIM} clock, that is 0x60 if SHRTIM_TxCTRL.CKPSC[2:0] = 0, 0x30 if SHRTIM_TxCTRL.CKPSC[2:0] = 1, 0x18 if SHRTIM_TxCTRL.CKPSC[2:0] = 2,... When PLEN is set to 1, the register will configure the value to the compare 1 preload register. If PLEN is reset, the register value will be directly loaded to

		the active compare 1 register.

9.4.2.9 SHRTIM Timer x Compare 1 Compound Register (SHRTIM_TxRCMP1DAT)

Address offset: TIMA: 0x0A0

Address offset: TIMB: 0x120

Address offset: TIMC: 0x1A0

Address offset: TIMD: 0x220

Address offset: TIME: 0x2A0

Address offset: TIMF: 0x320

Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								REPT							
								rw							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMP1DAT															
rw															

Bit Field	Name	Description
[31:24]	Reserved	Reserved, the reset value must be maintained
[23:16]	REPT	Timer x repetition value Aliased from SHRTIM_TxREPT register This bitfield is an alias from the REPT[7:0] bitfield in the SHRTIM_TxREPT register.
[15:0]	CMP1DAT	Timer x compare 1 data This bitfield is an alias from the CMP1DAT[15:0] bitfield in the SHRTIM_TxCMP1DAT register.

9.4.2.10 SHRTIM Timer x Compare 2 Register（SHRTIM_TxCMP2DAT）

Address offset: TIMA: 0x0A4

Address offset: TIMB: 0x124

Address offset: TIMC: 0x1A4

Address offset: TIMD: 0x224

Address offset: TIME: 0x2A4

Address offset: TIMF: 0x324

Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

<table>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="16">CMP2DAT</td></tr>
<tr><td colspan="16">rw</td></tr>
</table>

Bit Field	Name	Description
[31:16]	Reserved	Reserved, the reset value must be maintained
[15:0]	CMP2DAT	Timer x compare 2 data The compare 2 value must be above or equal to 3 periods of the f_{SHRTIM} clock, that is 0x60 if SHRTIM_TxCTRL.CKPSC[2:0] = 0, 0x30 if SHRTIM_TxCTRL.CKPSC[2:0] = 1, 0x18 if SHRTIM_TxCTRL.CKPSC[2:0] = 2,... When PLEN is set to 1, the register will configure the value to the compare 2 preload register. If PLEN is reset, the register value will be directly loaded to the active compare 2 register.

9.4.2.11 SHRTIM Timer x Compare 3 Register（SHRTIM_TxCMP3DAT）

Address offset: TIMA: 0x0A8

Address offset: TIMB: 0x128

Address offset: TIMC: 0x1A8

Address offset: TIMD: 0x228

Address offset: TIME: 0x2A8

Address offset: TIMF: 0x328

Reset value: 0x00000000

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="16">Reserved</td></tr>
</table>

<table>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="16">CMP3DAT</td></tr>
<tr><td colspan="16">rw</td></tr>
</table>

Bit Field	Name	Description
[31:16]	Reserved	Reserved, the reset value must be maintained
[15:0]	CMP3DAT	Timer x compare 3 data The compare 3 value must be above or equal to 3 periods of the f_{SHRTIM} clock, that is 0x60 if SHRTIM_TxCTRL.CKPSC[2:0] = 0, 0x30 if SHRTIM_TxCTRL.CKPSC[2:0] = 1, 0x18 if SHRTIM_TxCTRL.CKPSC[2:0] = 2,... When PLEN is set to 1, the register will configure the value to the compare 3 preload register. If PLEN is reset, the register value will be directly loaded to

		the active compare 3 register.

9.4.2.12 SHRTIM Timer x Compare 4 Register（SHRTIM_TxCMP4DAT）

Address offset: TIMA: 0xAC

Address offset: TIMB: 0x12C

Address offset: TIMC: 0x1AC

Address offset: TIMD: 0x22C

Address offset: TIME: 0x2AC

Address offset: TIMF: 0x32C

Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMP4DAT															
rw															

Bit Field	Name	Description
[31:16]	Reserved	Reserved, the reset value must be maintained
[15:0]	CMP4DAT	Timer A compare 4 data The compare 4 value must be above or equal to 3 periods of the f_{SHRTIM} clock, that is 0x60 if SHRTIM_TxCTRL.CKPSC[2:0] = 0, 0x30 if SHRTIM_TxCTRL.CKPSC[2:0] = 1, 0x18 if SHRTIM_TxCTRL.CKPSC[2:0] = 2,... When PLEN is set to 1, the register will configure the value to the compare 4 preload register. If PLEN is reset, the register value will be directly loaded to the active compare 4 register.

9.4.2.13 SHRTIM Timer x Capture 1 Register（SHRTIM_TxCPT1）

Address offset: TIMA: 0x0B0

Address offset: TIMB: 0x130

Address offset: TIMC: 0x1B0

Address offset: TIMD: 0x230

Address offset: TIME: 0x2B0

Address offset: TIMF: 0x330

Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

Reserved															UDSTS1
															r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CPT1															
r															

Bit Field	Name	Description
[31:17]	Reserved	Reserved, the reset value must be maintained
[16]	UDSTS1	Status for timer x counting direction while capture 1 happened 0: Timer x is up-counting 1: Timer x is down-counting In up-counting mode, the UDSTS1 bit is always read as 0.
[15:0]	CPT1	Timer x Capture 1 value This register holds the counter value when the capture 1 event occurred.

Notes: The UDSTS1 bit allows to discriminate the up-down phases when reading the captured value.

Note: This is a regular resolution register: for SHRTIMER clock prescaling ratio below 32 (CKPSC[2:0] < 5), the least significant bits of the counter are not significant. They cannot be written and return 0 when read.

9.4.2.14 SHRTIM Timer x Capture 2 Register（SHRTIM_TxCPT2）

Address offset: TIMA: 0x0B4

Address offset: TIMB: 0x134

Address offset: TIMC: 0x1B4

Address offset: TIMD: 0x234

Address offset: TIME: 0x2B4

Address offset: TIMF: 0x334

Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															UDSTS2
															r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CPT2															
r															

Bit Field	Name	Description
[31:17]	Reserved	Reserved, the reset value must be maintained
[16]	UDSTS2	Status for timer x counting direction while capture 2 happened 0: Timer x is up-counting 1: Timer x is down-counting In up-counting mode, the UDSTS2 bit is always read as 0.

[15:0]	CPT2	Timerx Capture 2 value This register holds the counter value when the capture 2 event occurred.

Notes:The UDSTS2 bit allows to discriminate the up-down phases when reading the captured value.

Note: This is a regular resolution register: for SHRTIMER clock prescaling ratio below 32 (CKPSC[2:0] < 5), the least significant bits of the counter are not significant. They cannot be written and return 0 when read.

9.4.2.15 SHRTIM Timer x Deadtime Register（SHRTIM_TxDT）

Address offset: TIMA: 0x0B8

Address offset: TIMB: 0x138

Address offset: TIMC: 0x1B8

Address offset: TIMD: 0x238

Address offset: TIME: 0x2B8

Address offset: TIMF: 0x338

Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DTFLCK	DTFSLCK	Reserved				SDTF	DTF								
rw	rw					rw	rw								

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DTRLCK	DTRSLCK	Reserved	DTPSC			SDTR	DTR								
rw	rw		rw			rw	rw								

Bit Field	Name	Description
[31]	DTFLCK	Deadtime Falling Lock This write-once bit prevents the deadtime (sign and value) to be modified, if enabled. 0: Deadtime falling value and sign is writable 1: Deadtime falling value and sign is read-only *Note: This bit is not preloaded*
[30]	DTFSLCK	Deadtime Falling Sign Lock This write-once bit prevents the sign of falling deadtime to be modified, if enabled. 0: Deadtime falling sign is writable 1: Deadtime falling sign is read-only *Note: This bit is not preloaded.*
[29:26]	Reserved	Reserved, the reset value must be maintained
[25]	SDTF	Sign Deadtime Falling value This register determines whether the deadtime is positive (signals not overlapping) or negative (signals overlapping). 0: Positive deadtime on falling edge 1: Negative deadtime on falling edge
[24:16]	DTF	Deadtime Falling data value

		This register holds the value of the deadtime following a falling edge of reference PWM signal. t_{DTF} = DTF[8:0] x t_{DTG}
[15]	DTRLCK	Deadtime Rising Lock This write-once bit prevents the deadtime (sign and value) to be modified, if enabled. 0: Deadtime rising value and sign is writable 1: Deadtime rising value and sign is read-only *Note: This bit is not preloaded.*
[14]	DTRSLCK	Deadtime rising sign lock This write-once bit prevents the sign of deadtime to be modified, if enabled. 0: Deadtime rising sign is writable 1: Deadtime rising sign is read-only *Note: This bit is not preloaded.*
[13]	Reserved	Reserved, the reset value must be maintained
[12:10]	DTPSC	Deadtime prescaler This register holds the value of the deadtime clock prescaler. t_{DTG} = (2(DTPSC[2:0])) x (t_{SHRTIM} / 8) This bitfield is read-only as soon as any of the lock bit is enabled (DTFLCK, DTFSLCK, DTRLCK, DTRSLCK).
[9]	SDTR	Sign deadtime rising value This register determines whether the deadtime is positive or negative (overlapping signals). 0: Positive deadtime on rising edge 1: Negative deadtime on rising edge
[8:0]	DTR	Deadtime rising value This register holds the value of the deadtime following a rising edge of reference PWM signal. t_{DTR} = DTR[8:0] x t_{DTG}

9.4.2.16 SHRTIM Timer x Output 1 Set Register（SHRTIM_TxSET1）

Address offset: TIMA: 0x0BC

Address offset: TIMB: 0x13C

Address offset: TIMC: 0x1BC

Address offset: TIMD: 0x23C

Address offset: TIME: 0x2BC

Address offset: TIMF: 0x33C

Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UPD	EXEV10	EXEV9	EXEV8	EXEV7	EXEV6	EXEV5	EXEV4	EXEV3	EXEV2	EXEV1	TIMEV9	TIMEV8	TIMEV7	TIMEV6	TIMEV5
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIMEV 4	TIMEV 3	TIMEV 2	TIMEV 1	MCMP 4	MCMP 3	MCMP 2	MCMP 1	MPRD	CMP4	CMP3	CMP2	CMP1	PRD	RSYNC	SWT
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit Field	Name	Description
[31]	UPD	Registers update (transfer preload to active) Register update event forces the output to its active state.
[30]	EXEV10	External Event 10 Refer to the EXEV1 description
[29]	EXEV9	External Event 9 Refer to the EXEV1 description
[28]	EXEV8	External Event 8 Refer to the EXEV1 description
[27]	EXEV7	External Event 7 Refer to the EXEV1 description
[26]	EXEV6	External Event 6 Refer to the EXEV1 description
[25]	EXEV5	External Event 5 Refer to the EXEV1 description
[24]	EXEV4	External Event 4 Refer to the EXEV1 description
[23]	EXEV3	External Event 3 Refer to the EXEV1 description
[22]	EXEV2	External Event 2 Refer to the EXEV1 description
[21]	EXEV1	External Event 1 External event 1 forces the output to its active state.
[20]	TIMEV9	Timer Event 9 Refer to the TIMEV1 description
[19]	TIMEV8	Timer Event 8 Refer to the TIMEV1 description
[18]	TIMEV7	Timer Event 7 Refer to the TIMEV1 description
[17]	TIMEV6	Timer Event 6 Refer to the TIMEV1 description
[16]	TIMEV5	Timer Event 5 Refer to the TIMEV1 description
[15]	TIMEV4	Timer Event 4 Refer to the TIMEV1 description
[14]	TIMEV3	Timer Event 3 Refer to the TIMEV1 description
[13]	TIMEV2	Timer Event 2 Refer to the TIMEV1 description

[12]	TIMEV1	Timer Event 1 Timers event 1 forces the output to its active state (refer to Table9-10 for timer events assignments)
[11]	MCMP4	Master Compare 4 Master timer compare 4 event forces the output to its active state.
[10]	MCMP3	Master Compare 3 Master timer compare 3 event forces the output to its active state.
[9]	MCMP2	Master Compare 2 Master timer compare 2 event forces the output to its active state.
[8]	MCMP1	Master Compare 1 Master timer compare 1 event forces the output to its active state.
[7]	MPRD	Master Period The master timer counter roll-over in continuous mode, or the master timer reset in single-shot mode forces the output to its active state.
[6]	CMP4	Timer x Compare 4 Timer x compare 4 event forces the output to its active state.
[5]	CMP3	Timer x Compare 3 Timer x compare 3 event forces the output to its active state.
[4]	CMP2	Timer x Compare 2 Timer x compare 2 event forces the output to its active state.
[3]	CMP1	Timer x Compare 1 Timer x compare 1 event forces the output to its active state.
[2]	PRD	Timer x Period Timer x period event forces the output to its active state
[1]	RSYNC	Timer x resynchronization Timer x reset event coming solely from software or SYNC input forces the output to its active state. *Note: Other timer reset are not affecting the output when RSYNC=1.*
[0]	SWT	Software Set trigger This bit forces the output to its active state. This bit can only be set by software and is reset by hardware. *Note: This bit is not preloaded.*

9.4.2.17 SHRTIM Timer x Output 1 Reset Register（SHRTIM_TxRST1）

Address offset: TIMA: 0x0C0

Address offset: TIMB: 0x140

Address offset: TIMC: 0x1C0

Address offset: TIMD: 0x240

Address offset: TIME: 0x2C0

Address offset: TIMF: 0x340

Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UPD	EXEV10	EXEV9	EXEV8	EXEV7	EXEV6	EXEV5	EXEV4	EXEV3	EXEV2	EXEV1	TIMEV9	TIMEV8	TIMEV7	TIMEV6	TIMEV5
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIMEV4	TIMEV3	TIMEV2	TIMEV1	MCMP4	MCMP3	MCMP2	MCMP1	MPRD	CMP4	CMP3	CMP2	CMP1	PRD	RSYNC	SWT
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Refer to SHRTIM_TxSET1 bits description. These bits are defining the source which can force the Tx1 output to its inactive state.

9.4.2.18 SHRTIM Timer x Output 2 Set Register（SHRTIM_TxSET2）

Address offset: TIMA: 0x0C4

Address offset: TIMB: 0x144

Address offset: TIMC: 0x1C4

Address offset: TIMD: 0x244

Address offset: TIME: 0x2C4

Address offset: TIMF: 0x344

Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UPD	EXEV10	EXEV9	EXEV8	EXEV7	EXEV6	EXEV5	EXEV4	EXEV3	EXEV2	EXEV1	TIMEV9	TIMEV8	TIMEV7	TIMEV6	TIMEV5
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIMEV4	TIMEV3	TIMEV2	TIMEV1	MCMP4	MCMP3	MCMP2	MCMP1	MPRD	CMP4	CMP3	CMP2	CMP1	PRD	RSYNC	SWT
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Refer to SHRTIM_TxSET1 bits description. These bits are defining the source which can force the Tx2 output to its active state.

9.4.2.19 SHRTIM Timer x Output 2 Reset Register（SHRTIM_TxRST2）

Address offset: TIMA: 0x0C8

Address offset: TIMB: 0x148

Address offset: TIMC: 0x1C8

Address offset: TIMD: 0x248

Address offset: TIME: 0x2C8

Address offset: TIMF: 0x348

Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UPD	EXEV10	EXEV9	EXEV8	EXEV7	EXEV6	EXEV5	EXEV4	EXEV3	EXEV2	EXEV1	TIMEV9	TIMEV8	TIMEV7	TIMEV6	TIMEV5
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIMEV4	TIMEV3	TIMEV2	TIMEV1	MCMP4	MCMP3	MCMP2	MCMP1	MPRD	CMP4	CMP3	CMP2	CMP1	PRD	RSYNC	SWT
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Refer to SHRTIM_TxSET2 bits description. These bits are defining the source which can force the Tx2 output to its inactive state.

9.4.2.20 SHRTIM Timer x External Event Filtering Register 1（SHRTIM_TxEXEVFLT1）

Address offset: TIMA: 0x0CC

Address offset: TIMB: 0x14C

Address offset: TIMC: 0x1CC

Address offset: TIMD: 0x24C

Address offset: TIME: 0x2CC

Address offset: TIMF: 0x34C

Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved			EXEV5FLT				EXEV5LATCH	Reserved	EXEV4FLT				EXEV4LATCH	Reserved	EXEV3FLT
			rw				rw		rw				rw	r	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			EXEV3LATCH	Reserved	EXEV2FLT				EXEV2LATCH	Reserved	EXEV1FLT				EXEV1LATCH
			rw		rw				rw		rw				rw

Bit Field	Name	Description
[31:29]	Reserved	Reserved, the reset value must be maintained
[28:25]	EXEV5FLT	External Event 5 filter Refer to the EXEV1FLT[3:0] description
[24]	EXEV5LATCH	External event 5 latch Refer to the EXEV1LATCH description
[23]	Reserved	Reserved, the reset value must be maintained
[22:19]	EXEV4FLT	External Event 4 filter Refer to the EXEV1FLT[3:0] description
[18]	EXEV4LATCH	External event 4 latch Refer to the EXEV1LATCH description
[17]	Reserved	Reserved, the reset value must be maintained
[16:13]	EXEV3FLT	External Event 3 filter Refer to the EXEV1FLT[3:0] description

[12]	EXEV3LATCH	External event 3 latch Refer to the EXEV1LATCH description
[11]	Reserved	Reserved, the reset value must be maintained
[10:7]	EXEV2FLT	External Event 2 filter Refer to the EXEV1FLT[3:0] description
[6]	EXEV2LATCH	External event 2 latch Refer to the EXEV1LATCH description
[5]	Reserved	Reserved, the reset value must be maintained
[4:1]	EXEV1FLT	External Event 1 filter 0000: No filtering 0001: Blanking from counter reset/roll-over to compare 1 0010: Blanking from counter reset/roll-over to compare 2 in up-counting mode (SHRTIM_TxCTRL2.UPDOWNM bit reset). In up-down counting mode (SHRTIM_TxCTRL2.UPDOWNM bit set): blanking from compare 1 to compare 2, only during the up-counting phase. 0011: Blanking from counter reset/roll-over to compare 3 0100: Blanking from counter reset/roll-over to compare 4 0100: Blanking from counter reset/roll-over to compare 4 in up-counting mode (SHRTIM_TxCTRL2.UPDOWNM bit reset). In up-down counting mode (SHRTIM_TxCTRL2.UPDOWNM bit set): blanking from compare 3 to compare 4, only during the up-counting phase. 0101: Blanking from another timing unit: TIMFLTR1 source (refer to Table9-18 for details） 0110: Blanking from another timing unit: TIMFLTR2 source (refer to Table9-18 for details） 0111: Blanking from another timing unit: TIMFLTR3 source (refer to Table9-18 for details) 1000: Blanking from another timing unit: TIMFLTR4 source (refer to Table9-18 for details） 1001: Blanking from another timing unit: TIMFLTR5 source (refer to Table9-18 for details） 1010: Blanking from another timing unit: TIMFLTR6 source (refer to Table9-18 for details） 1011: Blanking from another timing unit: TIMFLTR7 source (refer to Table9-18 for details） 1100: Blanking from another timing unit: TIMFLTR8 source (refer to Table9-18 for details） 1101: Windowing from counter reset/roll-over to compare 2 in up-counting mode (SHRTIM_TxCTRL2.UPDOWNM bit reset). In up-down counting mode (SHRTIM_TxCTRL2.UPDOWNM bit set): windowing from compare 2 to compare 3, only during the up-counting phase. 1110: Windowing from counter reset/roll-over to compare 3 in up-counting mode (SHRTIM_TxCTRL2.UPDOWNM bit reset). In up-down counting mode

		(SHRTIM_TxCTRL2.UPDOWNM bit set): windowing from compare 2 to compare 3, only during the down-counting phase. 1111：Windowing from another timing unit: TIMWIN source (refer to **Table** 9-19 for details）in upcounting mode (SHRTIM_TxCTRL2.UPDOWNM bit reset). In up-down counting mode (SHRTIM_TxCTRL2.UPDOWNM bit set): windowing from compare 2 during the up-counting phase to compare 3 during the down-counting phase. If EXEVWINSEL = 0, compare 2 is from another timer unit(refer to **Table** 9-19); if EXEVWINSEL = 1, compare 2 is from timer unit itself. *Note: Whenever a compare register is used for filtering, the value must be strictly above 0.* *This bitfield must not be modified once the counter is enabled (TxCNTEN bit set)*
[0]	EXEV1LATCH	External Event 1 latch 0: Event 1 is ignored if it happens during a blank, or passed through during a window. 1: Event 1 is latched and delayed till the end of the blanking or windowing period. *Note: A timeout event is generated in window mode (EXEV1FLT[3:0]=1101、1110、1111) if EE1LTCH = 0, except if the external event is programmed in fast mode (EXEV1FM = 1).* *This bitfield must not be modified once the counter is enabled (TxCNTEN bit set)*

9.4.2.21 SHRTIM Timer x External Event Filtering Register 2（SHRTIM_TxEXEVFLT2）

Address offset: TIMA: 0x0D0

Address offset: TIMB: 0x150

Address offset: TIMC: 0x1D0

Address offset: TIMD: 0x250

Address offset: TIME: 0x2D0

Address offset: TIMF: 0x350

Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved			EXEV10FLT				EXEV10LATCH	Reserved	EXEV9FLT				EXEV9LATCH	Reserved	EXEV8FLT
			rw				rw		rw				rw	r	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			EXEV8LATCH	Reserved	EXEV7FLT				EXEV7LATCH	Reserved	EXEV6FLT				EXEV6LATCH
			rw		rw				rw		rw				rw

Bit Field	Name	Description
[31:29]	Reserved	Reserved, the reset value must be maintained
[28:25]	EXEV10FLT	External Event 10 filter Refer to the EXEV1FLT[3:0] description
[24]	EXEV10LATCH	External Event 10 latch Refer to the EXEV1LATCH description
[23]	Reserved	Reserved, the reset value must be maintained
[22:19]	EXEV9FLT	External Event 9 filter Refer to the EXEV1FLT[3:0] description
[18]	EXEV9LATCH	External Event 9 latch Refer to the EXEV1LATCH description
[17]	Reserved	Reserved, the reset value must be maintained
[16:13]	EXEV8FLT	External Event 8 filter Refer to the EXEV1FLT[3:0] description
[12]	EXEV8LATCH	External Event 8 latch Refer to the EXEV1LATCH description
[11]	Reserved	Reserved, the reset value must be maintained
[10:7]	EXEV7FLT	External Event 7 filter Refer to the EXEV1FLT[3:0] description
[6]	EXEV7LATCH	External Event 7 latch Refer to the EXEV1LATCH description
[5]	Reserved	Reserved, the reset value must be maintained
[4:1]	EXEV6FLT	External Event 6 filter Refer to the EXEV1FLT[3:0]
[0]	EXEV6LATCH	External Event 6 latch Refer to the EXEV1LATCH description

9.4.2.22 SHRTIM Timer A Counter Reset Register（SHRTIM_TACNTRST）

Address offset: 0xD4

Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TFCM P2	TECM P4	TECM P2	TECM P1	TDCM P4	TDCM P2	TDCM P1	TCCM P4	TCCM P2	TCCM P1	TBCM P4	TBCM P2	TBCM P1	EXEV 10	EXEV 9	EXEV8
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXEV7	EXEV6	EXEV5	EXEV4	EXEV3	EXEV2	EXEV1	MCMP 4	MCMP 3	MCMP 2	MCMP 1	MPRD	CMP4	CMP2	UPD	TFCM P1
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit Field	Name	Description
[31]	TFCMP2	Timer F Compare 2 The timer A counter is reset upon timer F compare 2 event.
[30]	TECMP4	Timer E Compare 4 The timer A counter is reset upon timer E compare 4 event.

[29]	TECMP2	Timer E Compare 2 The timer A counter is reset upon timer E compare 2 event.
[28]	TECMP1	Timer E Compare 1 The timer A counter is reset upon timer E compare 1 event.
[27]	TDCMP4	Timer D Compare 4 The timer A counter is reset upon timer D compare 4 event.
[26]	TDCMP2	Timer D Compare 2 The timer A counter is reset upon timer D compare 2 event.
[25]	TDCMP1	Timer D Compare 1 The timer A counter is reset upon timer D compare 1 event.
[24]	TCCMP4	Timer C Compare 4 The timer A counter is reset upon timer C compare 4 event.
[23]	TCCMP2	Timer C Compare 2 The timer A counter is reset upon timer C compare 2 event.
[22]	TCCMP1	Timer C Compare 1 The timer A counter is reset upon timer C compare 1 event.
[21]	TBCMP4	Timer B Compare 4 The timer A counter is reset upon timer B compare 4 event.
[20]	TBCMP2	Timer B Compare 2 The timer A counter is reset upon timer B compare 2 event.
[19]	TBCMP1	Timer B Compare 1 The timer A counter is reset upon timer B compare 1 event.
[18]	EXEV10	External event 10 The timer A counter is reset upon external event 10.
[17]	EXEV9	External event 9 The timer A counter is reset upon external event 9.
[16]	EXEV8	External event 8 The timer A counter is reset upon external event 8.
[15]	EXEV7	External event 7 The timer A counter is reset upon external event 7.
[14]	EXEV6	External event 6 The timer A counter is reset upon external event 6.
[13]	EXEV5	External event 5 The timer A counter is reset upon external event 5.
[12]	EXEV4	External event 4 The timer A counter is reset upon external event 4.
[11]	EXEV3	External event 3 The timer A counter is reset upon external event 3.
[10]	EXEV2	External event 2 The timer A counter is reset upon external event 2.
[9]	EXEV1	External event 1 The timer A counter is reset upon external event 1.
[8]	MCMP4	Master timer compare 4

		The timer A counter is reset upon master timer compare 4 event.
[7]	MCMP3	Master timer compare 3 The timer A counter is reset upon master timer compare 3 event.
[6]	MCMP2	Master timer compare 2 The timer A counter is reset upon master timer compare 2 event.
[5]	MCMP1	Master timer compare 1 The timer A counter is reset upon master timer compare 1 event.
[4]	MPRD	Master timer period The timer A counter is reset upon master timer period event.
[3]	CMP4	TIMA compare 4 The timer A counter is reset upon timer A compare 4 event.
[2]	CMP2	TIMA compare 2 The timer A counter is reset upon timer A compare 2 event.
[1]	UPD	TIMA update The timer A counter is reset upon update event.
[0]	TFCMP1	TIMF compare 1 The timer A counter is reset upon timer F compare 1 event.

9.4.2.23 SHRTIM Timer B Counter Reset Register（SHRTIM_TBCNTRST）

Address offset: 0x154

Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TFCM P2	TECM P4	TECM P2	TECM P1	TDCM P4	TDCM P2	TDCM P1	TCCM P4	TCCM P2	TCCM P1	TACM P4	TACM P2	TACM P1	EXEV 10	EXEV 9	EXEV8
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXEV7	EXEV6	EXEV5	EXEV4	EXEV3	EXEV2	EXEV1	MCMP 4	MCMP 3	MCMP 2	MCMP 1	MPRD	CMP4	CMP2	UPD	TFCM P1
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Please refer to the SHRTIM_TACNTRST description.

Bits 30:19 and bit 0 are different (the reset signals from TIMA, TIMC, TIMD, TIME, and TIMF).

9.4.2.24 SHRTIM Timer C Counter Reset Register（SHRTIM_TCCNTRST）

Address offset: 0x1D4

Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TFCM P2	TECM P4	TECM P2	TECM P1	TDCM P4	TDCM P2	TDCM P1	TBCM P4	TBCM P2	TBCM P1	TACM P4	TACM P2	TACM P1	EXEV 10	EXEV 9	EXEV8
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXEV7	EXEV6	EXEV5	EXEV4	EXEV3	EXEV2	EXEV1	MCMP 4	MCMP 3	MCMP 2	MCMP 1	MPRD	CMP4	CMP2	UPD	TFCM P1

rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Please refer to the SHRTIM_TACNTRST description.

Bits 30:19 and bit 0 are different (the reset signals from TIMA, TIMC, TIMD, TIME, and TIMF).

9.4.2.25 SHRTIM Timer D Counter Reset Register（SHRTIM_TDCNTRST）

Address offset: 0x254

Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TFCM P2	TECM P4	TECM P2	TECM P1	TCCM P4	TCCM P2	TCCM P1	TBCM P4	TBCM P2	TBCM P1	TACM P4	TACM P2	TACM P1	EXEV 10	EXEV 9	EXEV8
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXEV7	EXEV6	EXEV5	EXEV4	EXEV3	EXEV2	EXEV1	MCMP 4	MCMP 3	MCMP 2	MCMP 1	MPRD	CMP4	CMP2	UPD	TFCM P1
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Please refer to the SHRTIM_TACNTRST description.

Bits 30:19 and bit 0 are different (the reset signals from TIMA, TIMC, TIMD, TIME, and TIMF).

9.4.2.26 SHRTIM Timer E Counter Reset Register（SHRTIM_TECNTRST）

Address offset: 0x2D4

Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TFCM P2	TDCM P4	TDCM P2	TDCM P1	TCCM P4	TCCM P2	TCCM P1	TBCM P4	TBCM P2	TBCM P1	TACM P4	TACM P2	TACM P1	EXEV 10	EXEV 9	EXEV8
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXEV7	EXEV6	EXEV5	EXEV4	EXEV3	EXEV2	EXEV1	MCMP 4	MCMP 3	MCMP 2	MCMP 1	MPRD	CMP4	CMP2	UPD	TFCM P1
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Please refer to the SHRTIM_TACNTRST description.

Bits 30:19 and bit 0 are different (the reset signals from TIMA, TIMC, TIMD, TIME, and TIMF).

9.4.2.27 SHRTIM Timer F Counter Reset Register（SHRTIM_TFCNTRST）

Address offset: 0x354

Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TECM P2	TDCM P4	TDCM P2	TDCM P1	TCCM P4	TCCM P2	TCCM P1	TBCM P4	TBCM P2	TBCM P1	TACM P4	TACM P2	TACM P1	EXEV 10	EXEV 9	EXEV8
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

EXEV7	EXEV6	EXEV5	EXEV4	EXEV3	EXEV2	EXEV1	MCMP 4	MCMP 3	MCMP 2	MCMP 1	MPRD	CMP4	CMP2	UPD	TECM P1
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Please refer to the SHRTIM_TACNTRST description.

Bits 30:19 and bit 0 are different (the reset signals from TIMA, TIMC, TIMD, TIME, and TIMF).

9.4.2.28 SHRTIM Timer x Chopper Register（SHRTIM_TxCHOP）

Address offset: TIMA: 0x0D8

Address offset: TIMB: 0x158

Address offset: TIMC: 0x1D8

Address offset: TIMD: 0x258

Address offset: TIME: 0x2D8

Address offset: TIMF: 0x358

Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved					STARTPW				CARDCY			CARFRQ			
					rw				rw			rw			

Bit Field	Name	Description
[31:11]	Reserved	Reserved, the reset value must be maintained
[10:7]	STARTPW	Timer x start pulse width This register defines the initial pulsewidth following a rising edge on output signal. This bitfield cannot be modified when one of the SHRTIM_TxOUT.CHPy bits is set. t_{1STPW} = (STARTPW [3:0]+1) x 16 x t_{SHRTIM}. 0000：64.0 ns 0001：128.0 ns ... 1111：1024.0 ns
[6:4]	CARDCY	Timer x chopper duty cycle value This register defines the duty cycle of the carrier signal. This bitfield cannot be modified when one of the SHRTIM_TxOUT.CHPy bits is set. 000：0/8（i.e. only 1st pulse is present） ... 111：7/8
[3:0]	CARFRQ	Timer x carrier frequency value This register defines the carrier frequency f_{CHPFRQ} = f_{SHRTIM} / (16 x

		(CARFRQ[3:0]+1)). This bitfield cannot be modified when one of the SHRTIM_TxOUT.CHPCHy bits is set. 0000: 15.625 MHz (f_{SHRTIM}/16) ... 1111: 976.563khz (f_{SHRTIM}/256)

9.4.2.29 SHRTIM Timer A Capture 1 Control Register (SHRTIM_TACPT1CTRL)

Address offset: 0x0DC

Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TECMP2	TECMP1	TE1RST	TE1SET	TDCMP2	TDCMP1	TD1RST	TD1SET	TCCMP2	TCCMP1	TC1RST	TC1SET	TBCMP2	TBCMP1	TB1RST	TB1SET
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TFCMP2	TFCMP1	TF1RST	TF1SET	EXEV10CPT	EXEV9CPT	EXEV8CPT	EXEV7CPT	EXEV6CPT	EXEV5CPT	EXEV4CPT	EXEV3CPT	EXEV2CPT	EXEV1CPT	UPDCPT	SWCPT
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rt_w

Bit Field	Name	Description
[31]	TECMP2	TIME compare 2 0: No action 1: Timer E compare 2 triggers the timer A capture 1
[30]	TECMP1	TIME compare 1 0: No action 1: Timer E compare 1 triggers the timer A capture 1
[29]	TE1RST	TIME channel 1 output reset 0: No action 1: The timer A capture 1 is triggered by TE1 output active to inactive
[28]	TE1SET	TIME channel 1 output set 0: No action 1: The timer A capture 1 is triggered by TE1 output inactive to active
[27]	TDCMP2	TIMD compare 2 0: No action 1: Timer D compare 2 triggers the timer A capture 1
[26]	TDCMP1	TIMD compare 1 0: No action 1: Timer D compare 1 triggers the timer A capture 1
[25]	TD1RST	TIMD output 1 reset 0: No action 1: The timer A capture 1 is triggered by TD1 output active to inactive
[24]	TD1SET	TIMD output 1 set 0: No action 1: The timer A capture 1 is triggered by TD1 output inactive to active
[23]	TCCMP2	TIMC compare 2

		0: No action 1: Timer C compare 2 triggers the timer A capture 1
[22]	TCCMP1	TIMC compare 1 0: No action 1: Timer C compare 1 triggers the timer A capture 1
[21]	TC1RST	TIMC output 1 reset 0: No action 1: The timer A capture 1 is triggered by TC1 output active to inactive
[20]	TC1SET	TIMC output 1 set 0: No action 1: The timer A capture 1 is triggered by TC1 output inactive to active
[19]	TBCMP2	TIMB compare 2 0: No action 1: Timer B compare 2 triggers the timer A capture 1
[18]	TBCMP1	TIMB compare 1 0: No action 1: Timer B compare 1 triggers the timer A capture 1
[17]	TB1RST	TIMB ouput 1 reset 0: No action 1: The timer A capture 1 is triggered by TB1 output active to inactive transition
[16]	TB1SET	TIMB output 1 set 0: No action 1: The timer A capture 1 is triggered by TB1 output inactive to active transition
[15]	TFCMP2	TIMF compare 2 0: No action 1: Timer F compare 2 triggers the timer A capture 1
[14]	TFCMP1	TIMF compare 1 0: No action 1: Timer F compare 1 triggers the timer A capture 1
[13]	TF1RST	TIMF output 1 reset 0: No action 1: The timer A capture 1 is triggered by TF1 output active to inactive transition
[12]	TF1SET	TIMF output 1 set 0: No action 1: The timer A capture 1 is triggered by TF1 output inactive to active transition
[11]	EXEV10CPT	External event 10 capture 0: No action 1: The external event 10 triggers the timer A capture 1
[10]	EXEV9CPT	External event 10 capture

		0: No action 1: The external event 9 triggers the timer A capture 1
[9]	EXEV8CPT	External event 8 capture 0: No action 1: The external event 8 triggers the timer A capture 1
[8]	EXEV7CPT	External event 7 capture 0: No action 1: The external event 7 triggers the timer A capture 1
[7]	EXEV6CPT	External event 6 capture 0: No action 1: The external event 6 triggers the timer A capture 1
[6]	EXEV5CPT	External event 5 capture 0: No action 1: The external event 5 triggers the timer A capture 1
[5]	EXEV4CPT	External event 4 capture 0: No action 1: The external event 4 triggers the timer A capture 1
[4]	EXEV3CPT	External event 3 capture 0: No action 1: The external event 3 triggers the timer A capture 1
[3]	EXEV2CPT	External event 2 capture 0: No action 1: The external event 2 triggers the timer A capture 1
[2]	EXEV1CPT	External event 1 capture 0: No action 1: The external event 1 triggers the timer A capture 1
[1]	UPDCPT	Update capture 0: No action 1: The update event triggers the timer A capture 1
[0]	SWCPT	Software capture 0: No action 1: This bit forces the timer A capture 1 by software Note: This bit is set only, reset by hardware

9.4.2.30 SHRTIM Timer A Capture 2 Control Register (SHRTIM_TACPT2CTRL)

Address offset: 0x0E0

Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TECMP2	TECMP1	TE1RST	TE1SET	TDCMP2	TDCMP1	TD1RST	TD1SET	TCCMP2	TCCMP1	TC1RST	TC1SET	TBCMP2	TBCMP1	TB1RST	TB1SET
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

TFCM P2	TFCM P1	TF1R ST	TF1S ET	EXEV10 CPT	EXEV9 CPT	EXEV8 CPT	EXEV7 CPT	EXEV6 CPT	EXEV5 CPT	EXEV4 CPT	EXEV3 CPT	EXEV2 CPT	EXEV1 CPT	UPDC PT	SWC PT
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rt_w

Refer to the SHRTIM_TACPT1CTRL description.

9.4.2.31 SHRTIM Timer B Capture 1 Control Register (SHRTIM_TBCPT1CTRL)

Address offset: 0x15C

Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TECM P2	TECM P1	TE1R ST	TE1S ET	TDCMP2	TDCMP 1	TD1RS T	TD1SE T	TCCMP 2	TCCMP 1	TC1RS T	TC1SET	TFCMP 2	TFCMP 1	TF1R ST	TF1S ET
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	Rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAC MP2	TAC MP1	TA1R ST	TA1S ET	EXEV10 CPT	EXEV9 CPT	EXEV8 CPT	EXEV7 CPT	EXEV6 CPT	EXEV5 CPT	EXEV4 CPT	EXEV3 CPT	EXEV2 CPT	EXEV1 CPT	UPDC PT	SWC PT
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	Rw	rt_w

Refer to the SHRTIM_TACPT1CTRL description.

Bits 31:12 are different (from TIMA, TIMC, TIMD, TIME, and TIMF).

9.4.2.32 SHRTIM Timer B Capture 2 Control Register (SHRTIM_TBCPT2CTRL)

Address offset: 0x160

Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TECM P2	TECM P1	TE1R ST	TE1S ET	TDCMP2	TDCMP 1	TD1RS T	TD1SE T	TCCMP 2	TCCMP 1	TC1RS T	TC1SET	TFCMP 2	TFCMP 1	TF1R ST	TF1S ET
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAC MP2	TAC MP1	TA1R ST	TA1S ET	EXEV10 CPT	EXEV9 CPT	EXEV8 CPT	EXEV7 CPT	EXEV6 CPT	EXEV5 CPT	EXEV4 CPT	EXEV3 CPT	EXEV2 CPT	EXEV1 CPT	UPDC PT	SWC PT
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rt_w

Refer to the SHRTIM_TACPT1CTRL description.

Bits 31:12 are different (from TIMA, TIMC, TIMD, TIME, and TIMF).

9.4.2.33 SHRTIM Timer C Capture 1 Control Register (SHRTIM_TCCPT1CTRL)

Address offset: 0x1DC

Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TECM P2	TECM P1	TE1R ST	TE1S ET	TDCMP 2	TDCMP 1	TD1RS T	TD1SE T	TFCMP 2	TFCMP 1	TF1RST	TF1SET	TBCMP 2	TBCMP 1	TB1R ST	TB1S ET
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	Rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

TACMP2	TACMP1	TA1RST	TA1SET	EXEV10CPT	EXEV9CPT	EXEV8CPT	EXEV7CPT	EXEV6CPT	EXEV5CPT	EXEV4CPT	EXEV3CPT	EXEV2CPT	EXEV1CPT	UPDCPT	SWCPT
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	Rw	rw	rw	rw	rw	rt_w

Refer to the SHRTIM_TACPT1CTRL description.

Bits 31:12 are different (from TIMA, TIMB, TIMD, TIME, and TIMF).

9.4.2.34 SHRTIM Timer C Capture 2 Control Register (SHRTIM_TCCPT2CTRL)

Address offset: 0x1E0

Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TECMP2	TECMP1	TE1RST	TE1SET	TDCMP2	TDCMP1	TD1RST	TD1SET	TFCMP2	TFCMP1	TF1RST	TF1SET	TBCMP2	TBCMP1	TB1RST	TB1SET
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TACMP2	TACMP1	TA1RST	TA1SET	EXEV10CPT	EXEV9CPT	EXEV8CPT	EXEV7CPT	EXEV6CPT	EXEV5CPT	EXEV4CPT	EXEV3CPT	EXEV2CPT	EXEV1CPT	UPDCPT	SWCPT
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rt_w

Refer to the SHRTIM_TACPT1CTRL description.

Bits 31:12 are different (from TIMA, TIMB, TIMD, TIME, and TIMF).

9.4.2.35 SHRTIM Timer D Capture 1 Control Register (SHRTIM_TDCPT1CTRL)

Address offset: 0x25C

Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TECMP2	TECMP1	TE1RST	TE1SET	TFCMP2	TFCMP1	TF1RST	TF1SET	TCCMP2	TCCMP1	TC1RST	TC1SET	TBCMP2	TBCMP1	TB1RST	TB1SET
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TACMP2	TACMP1	TA1RST	TA1SET	EXEV10CPT	EXEV9CPT	EXEV8CPT	EXEV7CPT	EXEV6CPT	EXEV5CPT	EXEV4CPT	EXEV3CPT	EXEV2CPT	EXEV1CPT	UPDCPT	SWCPT
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rt_w

Refer to the SHRTIM_TACPT1CTRL description.

Bits 31:12 are different (from TIMA, TIMB, TIMC, TIME, and TIMF).

9.4.2.36 SHRTIM Timer D Capture 2 Control Register (SHRTIM_TDCPT2CTRL)

Address offset: 0x260

Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TECMP2	TECMP1	TE1RST	TE1SET	TFCMP2	TFCMP1	TF1RST	TF1SET	TCCMP2	TCCMP1	TC1RST	TC1SET	TBCMP2	TBCMP1	TB1RST	TB1SET
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TACMP2	TACMP1	TA1RST	TA1SET	EXEV10CPT	EXEV9CPT	EXEV8CPT	EXEV7CPT	EXEV6CPT	EXEV5CPT	EXEV4CPT	EXEV3CPT	EXEV2CPT	EXEV1CPT	UPDCPT	SWCPT
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rt_w

Refer to the SHRTIM_TACPT1CTRL description.

Bits 31:12 are different (from TIMA, TIMB, TIMC, TIME, and TIMF).

9.4.2.37 SHRTIM Timer E Capture 1 Control Register (SHRTIM_TECPT1CTRL)

Address offset: 0x2DC

Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TFCMP2	TFCMP1	TF1RST	TF1SET	TDCMP2	TDCMP1	TD1RST	TD1SET	TCCMP2	TCCMP1	TC1RST	TC1SET	TBCMP2	TBCMP1	TB1RST	TB1SET
rw	rw	rw	rw	Rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TACMP2	TACMP1	TA1RST	TA1SET	EXEV10CPT	EXEV9CPT	EXEV8CPT	EXEV7CPT	EXEV6CPT	EXEV5CPT	EXEV4CPT	EXEV3CPT	EXEV2CPT	EXEV1CPT	UPDCPT	SWCPT
rw	rw	rw	rw	Rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rt_w

Refer to the SHRTIM_TACPT1CTRL description.

Bits 31:12 are different (from TIMA, TIMB, TIMC, TIMD, and TIMF).

9.4.2.38 SHRTIM Timer E Capture 2 Control Register (SHRTIM_TECPT2CTRL)

Address offset: 0x2E0

Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TFCMP2	TFCMP1	TF1RST	TF1SET	TDCMP2	TDCMP1	TD1RST	TD1SET	TCCMP2	TCCMP1	TC1RST	TC1SET	TBCMP2	TBCMP1	TB1RST	TB1SET
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TACMP2	TACMP1	TA1RST	TA1SET	EXEV10CPT	EXEV9CPT	EXEV8CPT	EXEV7CPT	EXEV6CPT	EXEV5CPT	EXEV4CPT	EXEV3CPT	EXEV2CPT	EXEV1CPT	UPDCPT	SWCPT
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rt_w

Refer to the SHRTIM_TACPT1CTRL description.

Bits 31:12 are different (from TIMA, TIMB, TIMC, TIMD, and TIMF).

9.4.2.39 SHRTIM Timer F Capture 1 Control Register (SHRTIM_TFCPT1CTRL)

Address offset: 0x35C

Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TECMP2	TECMP1	TE1RST	TE1SET	TDCMP2	TDCMP1	TD1RST	TD1SET	TCCMP2	TCCMP1	TC1RST	TC1SET	TBCMP2	TBCMP1	TB1RST	TB1SET

rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TACMP2	TACMP1	TA1RST	TA1SET	EXEV10CPT	EXEV9CPT	EXEV8CPT	EXEV7CPT	EXEV6CPT	EXEV5CPT	EXEV4CPT	EXEV3CPT	EXEV2CPT	EXEV1CPT	UPDCPT	SWCPT
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rt_w

Refer to the SHRTIM_TACPT1CTRL description.

Bits 31:12 are different (from TIMA, TIMB, TIMC, TIMD, and TIME).

9.4.2.40 SHRTIM Timer F Capture 2 Control Register (SHRTIM_TFCPT2CTRL)

Address offset: 0x360

Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TECMP2	TECMP1	TE1RST	TE1SET	TDCMP2	TDCMP1	TD1RST	TD1SET	TCCMP2	TCCMP1	TC1RST	TC1SET	TBCMP2	TBCMP1	TB1RST	TB1SET
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TACMP2	TACMP1	TA1RST	TA1SET	EXEV10CPT	EXEV9CPT	EXEV8CPT	EXEV7CPT	EXEV6CPT	EXEV5CPT	EXEV4CPT	EXEV3CPT	EXEV2CPT	EXEV1CPT	UPDCPT	SWCPT
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rt_w

Refer to the SHRTIM_TACPT1CTRL description.

Bits 31:12 are different (from TIMA, TIMB, TIMC, TIMD, and TIME).

9.4.2.41 SHRTIM Timer x Output Register (SHRTIM_TxOUT)

Address offset: TIMA: 0x0E4

Address offset: TIMB: 0x164

Address offset: TIMC: 0x1E4

Address offset: TIMD: 0x264

Address offset: TIME: 0x2E4

Address offset: TIMF: 0x364

Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								DIDL2	CHP2	FALT2		IDLES2	IDLEM2	POL2	Reserved
								rw	rw	rw		rw		rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	BIAR	Reserved	DP			DPEN	DTEN	DIDL1	CHP1	FALT1		IDLES1	IDLEM1	POL1	Reserved
	rw		rw			rw	rw	rw	rw	rw		rw		rw	

Bit Field	Name	Description

[31:24]	Reserved	Reserved, the reset value must be maintained
[23]	DIDL2	Output 2 Deadtime upon burst mode Idle entry This bit delays the idle mode entry by forcing a deadtime insertion before switching the outputs to their idle state. This setting only applies when entering in idle state during a burst mode operation. 0: The programmed idle state is applied immediately to the output 2 1: Deadtime (inactive level) is inserted on output 2 before entering the idle mode. The deadtime value is set by DTF[8:0]. *Note: This parameter cannot be changed once the timer x is enabled.* *Note: DIDL=1 is set only if one of the outputs is active during the burst mode (IDLES=1), and with positive deadtimes (SDTR/SDTF set to 0).*
[22]	CHP2	Output 2 Chopper enable This bit enables the chopper on output 2. 0: Output signal is not changed 1: Output signal is chopped by a carrier signal *Note: This parameter cannot be changed once the timer x is enabled.*
[21:20]	FALT2	Output 2 Fault state These bits select the output 2 state after a fault event. 00: No action: the output is not affected by the fault input and stays in run mode. 01: Active 10: Inactive 11: High-Z *Note: This parameter cannot be changed once the timer x is enabled (TxCNTEN bit set), if FALTENx bit is set or if the output is in FAULT state.*
[19]	IDLES2	Output 2 Idle State This bit selects the output 2 idle state. 0: Inactive 1: Active *Note: This parameter must be set prior to have the SHRTIM controlling the outputs*
[18]	IDLEM2	This bit selects the output 2 idle mode. 0: No action: the output is not affected by the burst mode operation. 1: The output is in idle state when requested by the burst mode controller.
[17]	POL2	Output 2 polarity This bit selects the output 2 polarity. 0: Positive polarity (output active high) 1: Negative polarity (output active low) *Note: This parameter cannot be changed once the timer x is enabled.*
[16:15]	Reserved	Reserved, the reset value must be maintained
[14]	BIAR	Balanced idle automatic resume This bit selects the outputs are automatically re-enabled after a balanced idle event. This bit is only significant if DP[2:0] = 011 or 111, it is ignored otherwise. 0: Disabled

		1: Enabled *Note: This parameter cannot be changed once the timer x is enabled.*
[13]	Reserved	Reserved, the reset value must be maintained
[12:10]	DP	Delayed Protection These bits define the source and outputs on which the delayed protection schemes are applied. **In SHRTIM_TAOUT, SHRTIM_TBOUT, SHRTIM_TCOUT:** 000: Output 1 enters delayed idle on external event 6 001: Output 2 enters delayed idle on external event 6 010: Output 1 and output 2 both enter delayed idle on external event 6 011: Enters balanced idle on external event 6 100: Output 1 enters delayed idle on external event 7 101: Output 2 enters delayed idle on external event 7 110: Output 1 and output 2 both enter delayed idle on external event 7 111: Enters balanced idle on external event 7 **In SHRTIM_TDOUT、SHRTIM_TEOUT、SHRTIM_TFOUT:** 000: Output 1 enters delayed idle on external event 8 001: Output 2 enters delayed idle on external event 8 010: Output 1 and output 2 both enter delayed idle on external event 8 011: Enters balanced idle on external event 8 100: Output 1 enters delayed idle on external event 9 101: Output 2 enters delayed idle on external event 9 110: Output 1 and output 2 both enter delayed idle on external event 9 111: Enters balanced idle on external event 9
[9]	DPEN	Delayed Protection Enable This bit enables the delayed protection scheme. 0: No action 1: Delayed protection is enabled, as per DP[2:0] bits *Note: This parameter cannot be changed once the timer x is enabled (TxCNTEN bit set).*
[8]	DTEN	Deadtime enable This bit enables the deadtime insertion on output 1 and output 2. 0: Output 1 and output 2 signals are independent. 1: Deadtime is inserted between output 1 and output 2 (reference signal is output 1 signal generator) *Note: This parameter cannot be changed once the timer is operating (TxCNTEN bit set) or if its outputs are enabled and set/reset by another timer*
[7]	DIDL1	Output 1 Deadtime upon burst mode Idle entry This bit delays the idle mode entry by forcing a deadtime insertion before switching the outputs to their idle state. This setting only applies when entering in idle state during a burst mode operation. 0: The programmed idle state is applied immediately to the output 1 1: Deadtime (inactive level) is inserted on output 1 before entering the idle mode.

		The deadtime value is set by DTF[8:0]. *Note: This parameter cannot be changed once the timer x is enabled.* *Note: DIDL=1 is set only if one of the outputs is active during the burst mode (IDLES=1), and with positive deadtimes (SDTR/SDTF set to 0).*
[6]	CHP1	Output 1 Chopper enable This bit enables the chopper on output 1. 0: Output signal is not changed 1: Output signal is chopped by a carrier signal *Note: This parameter cannot be changed once the timer x is enabled.*
[5:4]	FALT1	Output 1 Fault state These bits select the output 1 state after a fault event. 00: No action: the output is not affected by the fault input and stays in run mode. 01: Active 10: Inactive 11: High-Z *Note: This parameter cannot be changed once the timer x is enabled (TxCNTEN bit set), if FALTENx bit is set or if the output is in FAULT state.*
[3]	IDLES1	Output 1 Idle State This bit selects the output 1 idle state. 0: Inactive 1: Active *Note: This parameter must be set prior to have the SHRTIM controlling the outputs*
[2]	IDLEM1	Output 1 Idle mode. This bit selects the output 1 idle mode. 0: No action: the output is not affected by the burst mode operation. 1: The output is in idle state when requested by the burst mode controller.
[1]	POL1	Output 1 polarity This bit selects the output 1 polarity. 0: Positive polarity (output active high) 1: Negative polarity (output active low) *Note: This parameter cannot be changed once the timer x is enabled.*
[0]	Reserved	Reserved, the reset value must be maintained

9.4.2.42 SHRTIM Timer x Fault Register (SHRTIM_TxFALT)

Address offset: TIMA: 0x0E8

Address offset: TIMB: 0x168

Address offset: TIMC: 0x1E8

Address offset: TIMD: 0x268

Address offset: TIME: 0x2E8

Address offset: TIMF: 0x368

Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FALTLCK	Reserved														
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										FALT6EN	FALT5EN	FALT4EN	FALT3EN	FALT2EN	FALT1EN
										rw	rw	rw	rw	rw	rw

Bit Field	Name	Description
[31]	FALTLCK	Fault sources lock 0: FALT1EN..FALT6EN bits are read/write 1: FALT1EN..FALT6EN bits are read only The FALTLCK bit is write-once. Once it has been set, it cannot be modified till the next system reset.
[30:6]	Reserved	Reserved, the reset value must be maintained
[5]	FALT6EN	FAULT 6 enable 0: Fault 6 input is ignored 1: Fault 6 input is active and disables SHRTIM outputs
[4]	FALT5EN	FAULT 5 enable 0: Fault 5 input is ignored 1: Fault 5 input is active and disables SHRTIM outputs
[3]	FALT4EN	FAULT 4 enable 0: Fault 4 input is ignored 1: Fault 4 input is active and disables SHRTIM outputs
[2]	FALT3EN	FAULT 3 enable 0: Fault 3 input is ignored 1: Fault 3 input is active and disables SHRTIM outputs
[1]	FALT2EN	FAULT 2 enable 0: Fault 2 input is ignored 1: Fault 2 input is active and disables SHRTIM outputs
[0]	FALT1EN	FAULT 1 enable 0: Fault 1 input is ignored 1: Fault 1 input is active and disables SHRTIM outputs

9.4.2.43 SHRTIM Timer x Control Register 2 (SHRTIM_TxCTRL2)

Address offset: TIMA: 0x0EC

Address offset: TIMB: 0x16C

Address offset: TIMC: 0x1EC

Address offset: TIMD: 0x26C

Address offset: TIME: 0x2EC

Address offset: TIMF: 0x36C

Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved											TRGHLF	Reserved		GTCMP3	GTCMP1
											rw			rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FEROM		BMROM		ADCROM		OUTROM		ROM		Reserved	UPDOWNM	Reserved	DUDACRST	DUDACSTEP	DUDACEN
rw		rw		rw		rw		rw			rw		rw	rw	rw

Bit Field	Name	Description
[31:21]	Reserved	Reserved, the reset value must be maintained
[20]	TRGHLF	Triggered-half mode This bitfield defines whether the compare 2 register is behaving in standard mode (compare match issued as soon as counter equal compare), or in triggered-half mode (refer to 9.3.4.12). 0: The SHRTIM_TxCMP2DAT register value is written by the user only (standard compare mode) 1: The SHRTIM_TxCMP2DAT register value is set by hardware as soon as a capture 1 event occurs. It is loaded with the (capture 1 divided by 2) value. The initial value can be written by the user (as long as TRGHLF is reset), but is ignored once the first capture has been triggered (the preload mechanism is disabled for SHRTIM_TxCMP2DAT when the TRGHLF bit is set). Note: This bitfield must not be modified once the counter is enabled (*SHRTIM_MCTRL.TxCNTEN* bit set).
[19:18]	Reserved	Reserved, the reset value must be maintained
[17]	GTCMP3	Greater than compare 3 PWM mode This bit defines the compare 3 operating mode. 0: The compare 3 event is generated when the counter is equal to the compare value (compare match mode) 1: The compare 3 event is generated when the counter is greater than the compare value. If the compare value is changed on-the-fly, the new compare value is compared with the current counter value and an output SET or RESET can be generated.
[16]	GTCMP1	Greater than compare 1 PWM mode 0: The compare 1 event is generated when the counter is equal to the compare value (compare match mode) 1: The compare 1 event is generated when the counter is greater than the compare value. If the compare value is changed on-the-fly, the new compare value is compared with the current counter value and an output SET or RESET can be generated.
[15:14]	FEROM	Fault and event roll-over mode This bit defines when the roll-over is generated in up-down counting mode. It only relates to the rollover event used by the fault and event counters.

		00: An event is generated when the counter is equal to 0 or to SHRTIM_TxPRD value 01: An event is generated when the counter is equal to 0 10: An event is generated when the counter is equal to SHRTIM_TxPRD 11: Reserved *Note: This setting only applies when the SHRTIM_TxCTRL2.UPDOWNM bit is set. It is not significant otherwise.* *Note: This bitfield cannot be changed once the timer is operating (SHRTIM_MCTRL.TxCNTEN bit set).*
[13:12]	BMROM	Burst mode roll-over mode This bit defines when the roll-over is generated in up-down counting mode. It only relates to the rollover event used in the burst mode controller, as burst mode trigger clock. 00: An event is generated when the counter is equal to 0 or to SHRTIM_TxPRD value 01: An event is generated when the counter is equal to 0 10: An event is generated when the counter is equal to SHRTIM_TxPRD 11: Reserved *Note: This setting only applies when the SHRTIM_TxCTRL2.UPDOWNM bit is set. It is not significant otherwise.* *Note: This bitfield cannot be changed once the timer is operating (SHRTIM_MCTRL.TxCNTEN bit set).*
[11:10]	ADCROM	ADC roll-over mode This bit defines when the roll-over is generated in up-down counting mode. It only relates to the rollover event which is used for ADC trigger. 00: An event is generated when the counter is equal to 0 or to SHRTIM_TxPRD value 01: An event is generated when the counter is equal to 0 10: An event is generated when the counter is equal to SHRTIM_TxPRD 11: Reserved *Note: This setting only applies when the SHRTIM_TxCTRL2.UPDOWNM bit is set. It is not significant otherwise.* *Note: This bitfield cannot be changed once the timer is operating (SHRTIM_MCTRL.TxCNTEN bit set).*
[9:8]	OUTROM	Output roll-over mode This bit defines when the roll-over is generated in up-down counting mode. It only relates to the rollover event which sets and/or resets the ouputs, as per SHRTIM_TxSETy and SHRTIM_TxRSTy settings. 00: An event is generated when the counter is equal to 0 or to SHRTIM_TxPRD value 01: An event is generated when the counter is equal to 0 10: An event is generated when the counter is equal to SHRTIM_TxPRD 11: Reserved

		Note: This setting only applies when the SHRTIM_TxCTRL2.UPDOWNM bit is set. It is not significant otherwise. *Note: This bitfield cannot be changed once the timer is operating (SHRTIM_MCTRL.TxCNTEN bit set)*
[7:6]	ROM	Roll-over mode This bit defines when the roll-over is generated in up-down counting mode. It only relates to the rollover event with the following destinations: update trigger (transfer content from preload to active registers), IRQ and DMA requests, repetition counter decrement and external event filtering. 00: An event is generated when the counter is equal to 0 or to SHRTIM_TxPRD value 01: An event is generated when the counter is equal to 0 10: An event is generated when the counter is equal to SHRTIM_TxPRD 11: Reserved *Note: This setting only applies when the SHRTIM_TxCTRL2.UPDOWNM bit is set. It is not significant otherwise.* *Note: This bitfield cannot be changed once the timer is operating (SHRTIM_MCTRL.TxCNTEN bit set)*
[5]	Reserved	Reserved, the reset value must be maintained
[4]	UPDOWNM	Up-down mode This bit defines whether the counter operates in up counting mode or up-down counting mode. 0: The counter operates in up counting mode 1: The counter operates in up-down counting mode *Note: This bit cannot be changed once the timer is operating (SHRTIM_MCTRL.TxCNTEN bit set).*
[3]	Reserved	Reserved, the reset value must be maintained
[2]	DUDACRST	Dual DAC reset trigger This bit defines when the shrtim_dac_reset_trgx trigger is generated. 0: The trigger is generated on counter reset or roll-over events 1: The trigger is generated on output 1 set events *Note: The DUDACRST bit is not significant when the DUDACEN bit is reset (dual DAC trigger disabled).*
[1]	DUDACSTEP	Dual DAC step trigger This bit defines when the shrtim_dac_step_trgx trigger is generated. 0: The trigger is generated on compare 2 events 1: The trigger is generated on output 1 reset events *Note: The DUDACSTEP bit is not significant when the DUDACEN bit is reset (dual DAC trigger disabled).*
[0]	DUDACEN	Dual DAC trigger enable This bit enables the dual DAC trigger mechanism. 0: Disables the dual DAC trigger 1: Enables the dual DAC trigger

		Note: This bit cannot be changed once the timer is operating (SHRTIM_MCTRL.TxCNTEN bit set).

9.4.2.44 SHRTIM Timer x External Event Filtering Register 3 (SHRTIM_TxEXEVFLT3)

Address offset: 0x0F0

Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		EXEVCNT						EXEVSEL				Reserved	EXEVRSTM	EXEVCNTRST	EXEVCNTEN
		rw						rw					rw	rt_w	rw

Bit Field	Name	Description
[31:14]	Reserved	Reserved, the reset value must be maintained
[13:8]	EXEVCNT	Channel x (x = A~F)'s external event counter This bitfield selects the channel x (x = A~F)'s external event counter threshold. An event is taken into account valid when the number of events is equal to the (EXEVCNT[5:0]+1) value.
[7:4]	EXEVSEL	Channel x (x = A~F)'s s external event selection This bitfield selects the channel x (x = A~F)'s external event source 0000: The external event 1 is used as the channel x (x = A~F)'s external event source 0001: The external event 2 is used as the channel x (x = A~F)'s external event source ... 1001: The external event 10 is used as the channel x (x = A~F)'s external event source Others: Reserved
[3]	Reserved	Reserved, the reset value must be maintained
[2]	EXEVRSTM	Channel x (x = A~F)'s external event reset mode This bitfield selects the channel x (x = A~F)'s external event counter reset mode. 0: The channel x (x = A~F)'s external event counter is reset on each reset/roll-over event 1: The channel x (x = A~F)'s external event counter is reset on each reset/roll-over event only if no event occurs during last counting period
[1]	EXEVCNTRST	Channel x (x = A~F)'s external event counter reset This bit resets the channel x (x = A~F)'s external event counter. It is set by software and reset by hardware. 0: No action 1: The channel x (x = A~F)'s external event counter is reset

[0]	EXEVCNTEN	Channel x (x = A~F)’s external event counter enable This bit enables the channel x (x = A~F)’s external event counter. 0: The channel x (x = A~F)’s external event counter disabled 1: The channel x (x = A~F)’s external event counter enabled

9.4.2.45 SHRTIM Timer x Compare 5 Register (SHRTIM_TxCMP5DAT)

Address offset: 0x0F4

Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMP5DAT															
rw															

Bit Field	Name	Description
[31:16]	Reserved	Reserved, the reset value must be maintained
[15:0]	CMP5DAT	Timer x compare 5 data 1． The Compare unit 5 is dedicated to trigger ADC and does not have the same functions as CMP1 to CMP4. It only supports the following features: 2． CMP5 has regular resolution and does not has high-resolution characteristics. 3． When the counter matches CMP5 value, it will generate the CMP5 status flag, which can also be cleared. However, CMP5 events do not connect to NVIC, so they do not generate interrupts. 4． CMP5 events can be used to trigger the ADC.

9.4.3 The registers of common registers of SHRTIM

9.4.3.1 SHRTIM Control Register 1 (SHRTIM_CTRL1)

Address offset: 0x380

Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				ADTG4UPDSRC			ADTG3UPDSRC			ADTG2UPDSRC			ADTG1UPDSRC		
				rw			rw			rw			rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									TFUPDDIS	TEUPDDIS	TDUPDDIS	TCUPDDIS	TBUPDDIS	TAUPDDIS	MUPDDIS
									rw	rw	rw	rw	rw	rw	rw

Bit Field	Name	Description
[31:28]	Reserved	Reserved, the reset value must be maintained
[27:25]	ADTG4UPDSRC	ADC trigger 4 update source Refer to the ADTG1UPDSRC[2:0] description
[24:22]	ADTG3UPDSRC	ADC trigger 3 update source Refer to the ADTG1UPDSRC[2:0] description
[21:19]	ADTG2UPDSRC	ADC trigger 2 update source Refer to the ADTG1UPDSRC[2:0] description
[18:16]	ADTG1UPDSRC	ADC trigger 1 update source These bits define the source which triggers the update of the SHRTIM_ADTG1SRC1 and SHRTIM_ADTG1SRC2 register (transfer from preload to active register). It only defines the source timer. The precise condition is defined within the timer itself, in SHRTIM_MCTRL or SHRTIM_TxCTRL. 000: Master timer 001: Timer A 010: Timer B 011: Timer C 100: Timer D 101: Timer E 110: Timer F 111: Reserved
[15:7]	Reserved	Reserved, the reset value must be maintained
[6]	TFUPDDIS	TIM F update disable Refer to the TAUPDDIS description
[5]	TEUPDDIS	TIM E update disable Refer to the TAUPDDIS description
[4]	TDUPDDIS	TIM D update disable Refer to the TAUPDDIS description
[3]	TCUPDDIS	TIM C update disable Refer to the TAUPDDIS description
[2]	TBUPDDIS	TIM B update disable Refer to the TAUPDDIS description
[1]	TAUPDDIS	TIM A update disable This bit is set and cleared by software to enable/disable an update event generation temporarily on timer A. 0: Update enabled. The update occurs upon generation of the selected source. 1: Update disabled. The updates are temporarily disabled to allow the software to write multiple registers that have to be simultaneously taken into account.
[0]	MUPDDIS	Master timer update disable This bit is set and cleared by software to enable/disable an update event

		generation temporarily on master timer. 0: Update enabled. 1: Update disabled. The updates are temporarily disabled to allow the software to write multiple registers that have to be simultaneously taken into account.

9.4.3.2 SHRTIM Control Register 2 (SHRTIM_CTRL2)

Address offset: 0x384

Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved										SWAP F	SWAP E	SWAP D	SWAP C	SWAP B	SWA PA
										rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Rese rved	TFSWC NTRST	TESWC NTRST	TDSWC NTRST	TCSWC NTRST	TBSWC NTRST	TASWC NTRST	MSWC NTRST	Rese rved	TFSW UPD	TESW UPD	TDSW UPD	TCSW UPD	TBSW UPD	TASW UPD	MSW UPD
	rt_w	rt_w	rt_w	rt_w	rt_w	rt_w	rt_w	rt_w	rt_w	rt_w	rt_w	rt_w	rt_w	rt_w	rt_w

Bit Field	**Name**	**Description**
[31:22]	Reserved	Reserved, the reset value must be maintained
[21]	SWAPF	TIMF output swap Refer to the SWAPA description *Note: This bit is not significant when push-pull mode is enabled (PP = 1).*
[20]	SWAPE	TIME output swap Refer to the SWAPA description *Note: This bit is not significant when push-pull mode is enabled (PP = 1).*
[19]	SWAPD	TIMD output swap Refer to the SWAPA description *Note: This bit is not significant when push-pull mode is enabled (PP = 1).*
[18]	SWAPC	TIMC output swap Refer to the SWAPA description *Note: This bit is not significant when push-pull mode is enabled (PP = 1).*
[17]	SWAPB	TIMB output swap Refer to the SWAPA description *Note: This bit is not significant when push-pull mode is enabled (PP = 1).*
[16]	SWAPA	TIMA output swap This bit allows to swap the timer A outputs. 0: SHRTIM_TARST1 and SHRTIM_TASET1 are coding for the output A1, SHRTIM_TARST2 and SHRTIM_TASET2 are coding for the output A2. 1: SHRTIM_TARST1 and SHRTIM_TASET1 are coding for the output A2, SHRTIM_TARST2 and SHRTIM_TASET2 are coding for the output A1. *Note: This bit is not significant when push-pull mode is enabled (PP = 1).*
[15]	Reserved	Reserved, the reset value must be maintained
[14]	TFSWCNTRST	TIMF's software counter reset

		Refer to the TASWCNTRST description
[13]	TESWCNTRST	TIME’s software counter reset Refer to the TASWCNTRST description
[12]	TDSWCNTRST	TIMD’s software counter reset Refer to the TASWCNTRST description
[11]	TCSWCNTRST	TIMC’s software counter reset Refer to the TASWCNTRST description
[10]	TBSWCNTRST	TIMB’s software counter reset Refer to the TASWCNTRST description
[9]	TASWCNTRST	TIMA’s software counter reset Setting this bit resets the timer A counter. 0: No action 1: Timer A is reset by software The bit is automatically clear by hardware.
[8]	MSWCNTRST	Master timer’s software counter reset Setting this bit resets the master timer counter. 0: No action 1: Master timer is reset by software The bit is automatically clear by hardware.
[7]	Reserved	Reserved, the reset value must be maintained
[6]	TFSWUPD	TIMF software update） Refer to the TASWUPD description
[5]	TESWUPD	TIME software update） Refer to the TASWUPD description
[4]	TDSWUPD	TIMD software update） Refer to the TASWUPD description
[3]	TCSWUPD	TIMC software update Refer to the TASWUPD description
[2]	TBSWUPD	TIMB software update Refer to the TASWUPD description
[1]	TASWUPD	Timer A Software Update This bit is set by software and automatically reset by hardware. It forces an immediate transfer from the preload to the active register and any pending update request is canceled.
[0]	MSWUPD	Master Timer Software update This bit is set by software and automatically reset by hardware. It forces an immediate transfer from the preload to the active register in the master timer and any pending update request is canceled.

9.4.3.3 SHRTIM Interrupt Status Register (SHRTIM_INTSTS)

Address offset: 0x388

Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved														BMPRDITF	Reserved
														r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									FALT6ITF	SYSFALTITF	FALT5ITF	FALT4ITF	FALT3ITF	FALT2ITF	FALT1ITF
									r	r	r	r	r	r	r

Bit Field	Name	Description
[31:18]	Reserved	Reserved, the reset value must be maintained
[17]	BMPRDITF	Burst mode Period Interrupt Flag This bit is set by hardware when a single-shot burst mode operation is completed, or at the end of a burst mode period in continuous mode. It is cleared by software writing it at 1. 0: No burst mode period interrupt occurred 1: Burst mode period interrupt occurred
[16]	Reserved	Reserved, the reset value must be maintained
[15:7]	Reserved	Reserved, the reset value must be maintained
[6]	FALT6ITF	Fault 6 interrupt flag Refer to the FALT1ITF description
[5]	SYSFALTITF	System Fault Interrupt Flag Refer to the FALT1ITF description
[4]	FALT5ITF	Fault 5 interrupt flag Refer to the FALT1ITF description
[3]	FALT4ITF	Fault 4 interrupt flag Refer to the FALT1ITF description
[2]	FALT3ITF	Fault 3 interrupt flag Refer to the FALT1ITF description
[1]	FALT2ITF	Fault 2 interrupt flag Refer to the FALT1ITF description
[0]	FALT1ITF	Fault 1 Interrupt Flag This bit is set by hardware when fault 1 event occurs. It is cleared by software writing it at 1. 0: No fault 1 interrupt occurred 1: Fault 1 interrupt occurred

9.4.3.4 SHRTIM Interrupt Clear Register (SHRTIM_INTCLR)

Address offset: 0x38C

Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved														BMPRDIC	Reserved
														w	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									FALT6IC	SYSFALTIC	FALT5IC	FALT4IC	FALT3IC	FALT2IC	FALT1IC
									w	w	w	w	w	w	w

Bit Field	Name	Description
[31:18]	Reserved	Reserved, the reset value must be maintained
[17]	BMPRDIC	Burst mode period flag clear Writing a 1 to this bit will clear the SHRTIM_INTSTS.BMPRDITF.
[16]	Reserved	Reserved, the reset value must be maintained
[15:7]	Reserved	Reserved, the reset value must be maintained
[6]	FALT6IC	Fault 6 interrupt flag clear Writing a 1 to this bit will clear the SHRTIM_INTSTS.FALT6ITF
[5]	SYSFALTIC	System fault interrupt flag clear Writing a 1 to this bit will clear the SHRTIM_INTSTS. SYSFALTITF
[4]	FALT5IC	Fault 5 interrupt flag clear Writing a 1 to this bit will clear the SHRTIM_INTSTS.FALT5ITF
[3]	FALT4IC	Fault 4 interrupt flag clear Writing a 1 to this bit will clear the SHRTIM_INTSTS.FALT4ITF
[2]	FALT3IC	Fault 3 interrupt flag clear Writing a 1 to this bit will clear the SHRTIM_INTSTS.FALT3ITF
[1]	FALT2IC	Fault 2 interrupt flag clear Writing a 1 to this bit will clear the SHRTIM_INTSTS.FALT2ITF
[0]	FALT1IC	Fault 1 interrupt flag clear Writing a 1 to this bit will clear the SHRTIM_INTSTS.FALT1ITF

9.4.3.5 SHRTIM Interrupt Enable Register (SHRTIM_INTEN)

Address offset: 0x390

Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved														BMPRDIEN	Reserved
														rw	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									FALT6IEN	SYSFALTIEN	FALT5IEN	FALT4IEN	FALT3IEN	FALT2IEN	FALT1IEN
									rw	rw	rw	rw	rw	rw	rw

Bit Field	Name	Description
[31:18]	Reserved	Reserved, the reset value must be maintained
[17]	BMPRDIEN	Burst mode period interrupt enable This bit is set and cleared by software to enable/disable the burst mode period interrupt. 0: Disable burst mode period interrupt 1: Enable burst mode period interrupt

[16]	Reserved	Reserved, the reset value must be maintained
[15:7]	Reserved	Reserved, the reset value must be maintained
[6]	FALT6IEN	Fault 6 interrupt enable This bit is set and cleared by software to enable/disable the fault 6 interrupt. 0: Disable fault 6 interrupt 1: Enable fault 6 interrupt
[5]	SYSFALTIEN	System fault interrupt enable This bit is set and cleared by software to enable/disable the system fault interrupt. 0: Disable system fault interrupt 1: Enable system fault interrupt
[4]	FALT5IEN	Fault 5 interrupt enable This bit is set and cleared by software to enable/disable the fault 5 interrupt. 0: Disable fault 5 interrupt 1: Enable fault 5 interrupt
[3]	FALT4IEN	Fault 4 interrupt enable This bit is set and cleared by software to enable/disable the fault 4 interrupt. 0: Disable fault 4 interrupt 1: Enable fault 4 interrupt
[2]	FALT3IEN	Fault 3 interrupt enable This bit is set and cleared by software to enable/disable the fault 3 interrupt. 0: Disable fault 3 interrupt 1: Enable fault 3 interrupt
[1]	FALT2IEN	Fault 2 interrupt enable This bit is set and cleared by software to enable/disable the fault 2 interrupt. 0: Disable fault 2 interrupt 1: Enable fault 2 interrupt
[0]	FALT1IEN	Fault 1 interrupt enable This bit is set and cleared by software to enable/disable the fault 1 interrupt. 0: Disable fault 1 interrupt 1: Enable fault 1 interrupt

9.4.3.6 SHRTIM Output Enable Register (SHRTIM_OEN)

Address offset: 0x394

Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

Reserved

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				TF2OEN	TF1OEN	TE2OEN	TE1OEN	TD2OEN	TD1OEN	TC2OEN	TC1OEN	TB2OEN	TB1OEN	TA2OEN	TA1OEN
				rt_w	rt_w	rt_w	rt_w	rt_w	rt_w	rt_w	rt_w	rt_w	rt_w	rt_w	rt_w

Bit Field	Name	Description
[31:12]	Reserved	Reserved, the reset value must be maintained
[11]	TF2OEN	Timer F output 2 enable Refer to the TA1OEN description
[10]	TF1OEN	Timer F output 1 enable Refer to the TA1OEN description
[9]	TE2OEN	Timer E output 1 enable Refer to the TA1OEN description
[8]	TE1OEN	Timer E output 1 enable Refer to the TA1OEN description
[7]	TD2OEN	Timer D output 2 enable Refer to the TA1OEN description
[6]	TD1OEN	Timer D output 1 enable Refer to the TA1OEN description
[5]	TC2OEN	Timer C output 2 enable Refer to the TA1OEN description
[4]	TC1OEN	Timer C output 1 enable Refer to the TA1OEN description
[3]	TB2OEN	Timer B output 2 enable Refer to the TA1OEN description
[2]	TB1OEN	Timer B output 1 enable Refer to the TA1OEN description
[1]	TA2OEN	Timer A output 2 enable Refer to the TA1OEN description
[0]	TA1OEN	Timer A output 1 enable Setting this bit to 1 will enable timer A output 1. Writing "0" has no effect. Reading this bit will return the output enable/disable status. This bit is asynchronously cleared by hardware immediately when the timer-related fault input/inputs is/are active. 0: SHRTIM_CHA1 output is disabled. The output is either in a fault or idle state. 1: SHRTIM_CHA1 output is enabled. *Note: The disabled state corresponds to both idle and fault states. The output disable state is provided by SHRTIM_ODISSTS.TA1ODISSTS.*

9.4.3.7 SHRTIM Output Disable Register (SHRTIM_ODIS)

Address offset: 0x398

Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				TF2ODIS	TF1ODIS	TE2ODIS	TE1ODIS	TD2ODIS	TD1ODIS	TC2ODIS	TC1ODIS	TB2ODIS	TB1ODIS	TA2ODIS	TA1ODIS
				w	w	w	w	w	w	w	w	w	w	w	w

Bit Field	Name	Description
[31:12]	Reserved	Reserved, the reset value must be maintained
[11]	TF2ODIS	TIMF output 2 disable Refer to the TA1ODIS description.
[10]	TF1ODIS	TIMF output 1 disable Refer to the TA1ODIS description.
[9]	TE2ODIS	TIME output 2 disable Refer to the TA1ODIS description.
[8]	TE1ODIS	TIME output 1 disable Refer to the TA1ODIS description.
[7]	TD2ODIS	TIMD output 2 disable Refer to the TA1ODIS description.
[6]	TD1ODIS	TIMD output 1 disable Refer to the TA1ODIS description.
[5]	TC2ODIS	TIMC output 2 disable Refer to the TA1ODIS description.
[4]	TC1ODIS	TIMC output 1 disable Refer to the TA1ODIS description.
[3]	TB2ODIS	TIMB output 2 disable Refer to the TA1ODIS description.
[2]	TB1ODIS	TIMB output 1 disable Refer to the TA1ODIS description.
[1]	TA2ODIS	TIMA output 2 disable Refer to the TA1ODIS description.
[0]	TA1ODIS	TIMA output 1 disable Setting this bit to 1 will disable timer A output 1. The output will transition from the running or fault state to the idle state. Writing "0" has no effect.

9.4.3.8 SHRTIM Output Disable Status Register (SHRTIM_ODISSTS)

Address offset: 0x39C

Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				TF2ODISSTS	TF1ODISSTS	TE2ODISSTS	TE1ODISSTS	TD2ODISSTS	TD1ODISSTS	TC2ODISSTS	TC1ODISSTS	TB2ODISSTS	TB1ODISSTS	TA2ODISSTS	TA1ODISSTS
				r	r	r	r	r	r	r	r	r	r	r	r

Bit Field	Name	Description
[31:12]	Reserved	Reserved, the reset value must be maintained
[11]	TF2ODISSTS	TIMF output 2 disable status Refer to the TA1ODISSTS description.
[10]	TF1ODISSTS	TIMF output 1 disable status Refer to the TA1ODISSTS description.
[9]	TE2ODISSTS	TIME output 2 disable status Refer to the TA1ODISSTS description.
[8]	TE1ODISSTS	TIME output 1 disable status Refer to the TA1ODISSTS description.
[7]	TD2ODISSTS	TIMD output 2 disable status Refer to the TA1ODISSTS description.
[6]	TD1ODISSTS	TIMD output 1 disable status Refer to the TA1ODISSTS description.
[5]	TC2ODISSTS	TIMC output 2 disable status Refer to the TA1ODISSTS description.
[4]	TC1ODISSTS	TIMC output 1 disable status Refer to the TA1ODISSTS description.
[3]	TB2ODISSTS	TIMB output 2 disable status Refer to the TA1ODISSTS description.
[2]	TB1ODISSTS	TIMB output 1 disable status Refer to the TA1ODISSTS description.
[1]	TA2ODISSTS	TIMA output 2 disable status Refer to the TA1ODISSTS description.
[0]	TA1ODISSTS	TIMA output 1 disable status Reading this bit will return the output disable state. This bit is not significant when the output is active (Tx1OEN or Tx2OEN = 1). 0: SHRTIM_CHA1 output is disabled, in idle state. 1: SHRTIM_CHA1 output is disabled, in fault state.

9.4.3.9 SHRTIM Burst Mode Control Register (SHRTIM_BMCTRL)

Address offset: 0x3A0

Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BMSTS	Reserved								TFBM	TEBM	TDBM	TCBM	TBBM	TABM	MBM
rc_w0									rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved					BMPLEN	BMPSC				BMCK				BMOM	BMEN
					rw	rw				rw				rw	rw

Bit Field	Name	Description
[31]	BMSTS	Burst Mode Status This bit gives the current operating state 0: Normal operation 1: Burst operation on-going. Writing this bit to 0 causes a burst mode early termination.
[30:23]	Reserved	Reserved, the reset value must be maintained
[22]	TFBM	TIM F burst mode Refer to the TABM description.
[21]	TEBM	TIM E burst mode Refer to the TABM description.
[20]	TDBM	TIM D burst mode Refer to the TABM description.
[19]	TCBM	TIM C burst mode Refer to the TABM description.
[18]	TBBM	TIM B burst mode Refer to the TABM description.
[17]	TABM	TIM A burst mode This bit defines the behavior of the timer during burst mode operation. This bit field cannot be changed once the burst mode is enabled. 0: Timer A counter clock is maintained, timer operates normally 1: Timer A counter clock is stopped, counter is reset *Note: This bit must not be set when the balanced idle mode is active (DP[2:0] = 0x11).*
[16]	MBM	Master timer Burst Mode This bit defines the behavior of the timer during burst mode operation. This bit field cannot be changed once the burst mode is enabled. 0: Master timer counter clock is maintained, timer operates normally 1: Master timer counter clock is stopped, counter is reset
[15:11]	Reserved	Reserved, the reset value must be maintained
[10]	BMPLEN	Burst mode Preload Enable This bit enables the register preload mechanism and defines whether write accesses to the preloadable registers (SHRTIM_BMCMP, SHRTIM_BMPRD) are done in the active registers or in the preload registers. 0: Disable preload: the write access is directly done into active registers 1: Enable preload: the write access is done into preload registers
[9:6]	BMPSC	Burst mode prescaler This bit field defines the prescaler ratio for the f_{SHRTIM} clock of the burst mode controller. It cannot be changed when burst mode is enabled.

		0000: Clock not divided 0001: Division by 2 0010: Division by 4 0011: Division by 8 0100: Division by 16 0101: Division by 32 0110: Division by 64 0111: Division by 128 1000: Division by 256 1001: Division by 512 1010: Division by 1024 1011: Division by 2048 1100: Division by 4096 1101: Division by 8192 1110: Division by 16384 1111: Division by 32768
[5:2]	BMCK	Burst Mode Clock source) This bit field defines the clock source for the burst mode counter. It cannot be changed when burst mode is enabled (for detailed information on the on-chip events 1 to 4 connections, please refer to Table 9-4). 0000: Master timer counter reset/roll-over 0001 : Timer A counter reset/roll-over 0010 : Timer B counter reset/roll-over 0011 : Timer C counter reset/roll-over 0100 : Timer D counter reset/roll-over 0101 : Timer E counter reset/roll-over 0110: On-chip event 1 (shrtim_bm_ck1) serves as the clock for the burst mode counter. 0111: On-chip event 1 (shrtim_bm_ck2) serves as the clock for the burst mode counter. 1000: On-chip event 1 (shrtim_bm_ck3) serves as the clock for the burst mode counter. 1001: On-chip event 1 (shrtim_bm_ck4) serves as the clock for the burst mode counter. 1010: Prescaled f_{SHRTIM} clock (as per BMPSC[3:0] setting) 1011: Timer F counter reset/roll-over event Others: Reserved
[1]	BMOM	Burst Mode operating mode This bit defines whether to enter single-shot burst mode or continuous operation. 0: Single-shot mode 1: Continuous operation
[0]	BMEN	Burst Mode enable

		This bit is used to start the burst mode controller which is ready to receive the start trigger signal. Writing a 0 to this bit will cause a burst mode early termination. 0: Disable burst mode 1: Enable burst mode

9.4.3.10 SHRTIM Burst Mode Trigger Register (SHRTIM_BMTG)

Address offset: 0x3A4

Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OCEV	EXEV 8	EXEV 7	TDPRDEX EV8	TAPRDEX EV7	TECM P2	TECM P1	TERE PT	TFCMP 1	TDCM P2	TFRE PT	TDRE PT	TDRST RO	TFRST RO	TCCM P1	TCRE PT
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TCRST RO	TBCM P2	TBCM P1	TBREPT	TBRSTRO	TACM P2	TACM P1	TARE PT	TARST RO	MCM P4	MCM P3	MCM P2	MCMP 1	MREP T	MRST RO	SWST RT
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rt_w

Bit Field	Name	Description
[31]	OCEV	On-chip event A rising edge on the shrtim_bm_trg input triggers a burst mode entry (refer to Table 9-4).
[30]	EXEV8	External event 8 (TIMD filters applied) The external event 8 conditioned by TIMD filters is starting the burst mode operation.
[29]	EXEV7	External event 7 (TIMA filters applied) The external event 7 conditioned by TIMA filters is starting the burst mode operation
[28]	TDPRDEXEV8	Timer D period following external event 8 The timer D period following an external event 8 (conditioned by TIMD filters) is starting the burst mode operation.
[27]	TAPRDEXEV7	Timer A period following external event 7 The timer A period following an external event 7 (conditioned by TIMA filters) is starting the burst mode operation.
[26]	TECMP2	Timer E compare 2 event Refer to the TACMP2 description.
[25]	TECMP1	Timer E compare 1 event Refer to the TACMP1 description.
[24]	TEREPT	Timer E repetition event Refer to the TAREPT description.
[23]	TFCMP1	Timer F compare 1event Refer to the TACMP1 description.
[22]	TDCMP2	Timer E Compare 2 event Refer to the TACMP2 description.

[21]	TFREPT	Timer F repetition event Refer to the TAREPT description.
[20]	TDREPT	Timer D repetition event Refer to the TAREPT description.
[19]	TDRSTRO	Timer D reset/roll-over event Refer to the TARSTRO description.
[18]	TFRSTRO	Timer F reset/roll-over event Refer to the TARSTRO description.
[17]	TCCMP1	Timer C compare 1 event Refer to the TACMP1 description.
[16]	TCREPT	Timer C repetition event Refer to the TAREPT description.
[15]	TCRSTRO	Timer C reset/roll-over event Refer to the TARSTRO description.
[14]	TBCMP2	Timer B compare 2 event Refer to the TACMP2 description.
[13]	TBCMP1	Timer B compare 1 event Refer to the TACMP1 description.
[12]	TBREPT	Timer B repetition event Refer to the TAREPT description.
[11]	TBRSTRO	Timer B reset/roll-over event Refer to the TARSTRO description.
[10]	TACMP2	Timer A compare 2 event Refer to the TACMP1 description.
[9]	TACMP1	Timer A Compare 1 event Timer A compare 1 event will start burst mode operation.
[8]	TAREPT	Timer A repetition Timer A repetition event will start burst mode operation.
[7]	TARSTRO	Timer A counter reset or roll-over Timer A reset or roll-over event will start burst mode operation.
[6]	MCMP4	Master timer compare 4 Refer to the MCMP1 description.
[5]	MCMP3	Master timer compare 3 Refer to the MCMP1 description.
[4]	MCMP2	Master timer compare 2 Refer to the MCMP1 description.
[3]	MCMP1	Master timer compare 1 Master timer compare 1 event will start burst mode operation.
[2]	MREPT	Master timer repetition Master timer repetition event will start burst mode operation.
[1]	MRSTRO	Master timer reset/roll-over Master timer reset or roll-over event will start burst mode operation.
[0]	SWSTRT	Software start

		This bit is set by software and automatically reset by hardware. If this bit is set to 1, it starts the burst mode operation immediately. This bit is not active if the burst mode is not enabled (BMEM bit is reset).

9.4.3.11 SHRTIM Burst Mode Compare Register (SHRTIM_BMCMP)

Address offset: 0x3A8

Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BMCMP															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BMCMP															
rw															

Bit Field	**Name**	**Description**
[31:0]	BMCMP	Burst Mode compare value This register defines the number of periods during which the selected timers are in idle state. The register holds the contents of the preload register, or if preload is disabled, it holds the contents of the active register. *Note: The duration of IDLE and RUN in Burst mode must be greater than the duration of one PWM output period.*

9.4.3.12 SHRTIM Burst Mode Period Register (SHRTIM_BMPRD)

Address offset: 0x3AC

Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BMPRD															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BMPRD															
rw															

Bit Field	**Name**	**Description**
[31:0]	BMPRD	Burst Mode Period This register defines the burst mode repetition period The register holds the contents of the preload register, or if preload is disabled, it holds the contents of the active register. *Note: The duration of IDLE and RUN in Burst mode must be greater than the duration of one PWM output period.*

9.4.3.13 SHRTIM External Event Control Register 1 (SHRTIM_EXEVCTRL1)

Address offset: 0x3B0

Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		EXEV5SENS		EXEV5POL	EXEV5SRC			EXEV4SENS		EXEV4POL	EXEV4SRC			EXEV3SENS	
		rw		rw	rw			rw		rw	rw			rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXEV3POL	EXEV3SRC			EXEV2SENS		EXEV2POL	EXEV2SRC			EXEV1SENS		EXEV1POL	EXEV1SRC		
rw	rw			rw		rw	rw			rw		rw	rw		

Bit Field	Name	Description
[31:30]	Reserved	Reserved, the reset value must be maintained
[29:28]	EXEV5SENS	External event 5 sensitivity Refer to the EXEV1SENS description.
[27]	EXEV5POL	External event 5 polarity Refer to the EXEV1POL description.
[26:24]	EXEV5SRC	External event 5 source Refer to the EXEV1SRC description.
[23:22]	EXEV4SENS	External event 4 sensitivity Refer to the EXEV1SENS description.
[21]	EXEV4POL	External event 4 polarity Refer to the EXEV1POL description.
[20:18]	EXEV4SRC	External event 4 source Refer to the EXEV1SRC description.
[17:16]	EXEV3SENS	External event 3 sensitivity Refer to the EXEV1SENS description.
[15]	EXEV3POL	External event 3 polarity Refer to the EXEV1POL description.
[14:12]	EXEV3SRC	External event 3 source Refer to the EXEV1SRC description.
[11:10]	EXEV2SENS	External event 2 sensitivity Refer to the EXEV1SENS description.
[9]	EXEV2POL	External event 2 polarity Refer to the EXEV1POL description.
[8:6]	EXEV2SRC	External event 2 source Refer to the EXEV1SRC description.
[5:4]	EXEV1SENS	External event 1 sensitivity 00: On active level defined by EXEV1POL bit 01: Rising edge, whatever EXEV1POL bit value 10: Falling edge, whatever EXEV1POL bit value 11: Both edges, whatever EXEV1POL bit value

[3]	EXEV1POL	External event 1 polarity This bit is only significant if EXEV1SENS[1:0] = 00. 0: External event is active high 1: External event is active low *Note: This parameter cannot be changed once the timer x is enabled. It must be configured prior to setting EXEV1FM bit.*
[2:0]	EXEV1SRC	External event 4 source This bitfield selects the external event 1 source. 000: shrtim_exev1_1 001: shrtim_exev1_2 010: shrtim_exev1_3 011: shrtim_exev1_4 100: shrtim_exev1_5 101~111:Reserved *Note: This parameter cannot be changed once the timer x is enabled. It must be configured prior to setting EXEV1FM bit.*

9.4.3.14 SHRTIM External Event Control Register 2 (SHRTIM_EXEVCTRL2)

Address offset: 0x3B4

Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		EXEV10SENS		EXEV10POL	EXEV10SRC			EXEV9SENS		EXEV9POL	EXEV9SRC			EXEV8SENS	
		rw		rw	rw			rw		rw	rw			rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXEV8POL	EXEV8SRC			EXEV7SENS		EXEV7POL	EXEV7SRC			EXEV6SENS		EXEV6POL	EXEV6SRC		
rw	rw			rw		rw	rw			rw		rw	rw		

Bit Field	Name	Description
[31:30]	Reserved	Reserved, the reset value must be maintained
[29:28]	EXEV10SENS	External event 10 sensitivity Refer to the EXEV1SENS description.
[27]	EXEV10POL	External event 10 polarity Refer to the EXEV1POL description.
[26:24]	EXEV10SRC	External event 10 source Refer to the EXEV1SRC description.
[23:22]	EXEV9SENS	External event 9 sensitivity Refer to the EXEV1SENS description.
[21]	EXEV9POL	External event 9 polarity Refer to the EXEV1POL description.
[20:18]	EXEV9SRC	External event 9 source Refer to the EXEV1SRC description.
[17:16]	EXEV8SENS	External event 8 sensitivity

		Refer to the EXEV1SENS description.
[15]	EXEV8POL	External event 8 polarity Refer to the EXEV1POL description.
[14:12]	EXEV8SRC	External event 8 source Refer to the EXEV1SRC description.
[11:10]	EXEV7SENS	External event 7 sensitivity Refer to the EXEV1SENS description.
[9]	EXEV7POL	External event 7 polarity Refer to the EXEV1POL description.
[8:6]	EXEV7SRC	External event 7 source Refer to the EXEV1SRC description.
[5:4]	EXEV6SENS	External event 6 sensitivity Refer to the EXEV1SENS description.
[3]	EXEV6POL	External event 6 polarity Refer to the EXEV1POL description.
[2:0]	EXEV6SRC	External event 6 source Refer to the EXEV1SRC description.

9.4.3.15 SHRTIM External Event Control Register 3 (SHRTIM_EXEVCTRL3)

Address offset: 0x3B8

Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved			EXEV5FM	EXEV5F				Reserved	EXEV4FM	EXEV4F				Reserved	EXEV3FM
			rw	rw					rw	rw					rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXEV3F				Reserved	EXEV2FM	EXEV2F				Reserved	EXEV1FM	EXEV1F			
rw					rw	rw					rw	rw			

Bit Field	Name	Description
[31:29]	Reserved	Reserved, the reset value must be maintained
[28]	EXEV5FM	External event 5 fast mode Refer to the EXEV1FM description.
[27:24]	EXEV5F	External event 5 filter Refer to the EXEV1F description.
[23]	Reserved	Reserved, the reset value must be maintained
[22]	EXEV4FM	External event 4 fast mode Refer to the EXEV1FM description.
[21:18]	EXEV4F	External event 4 filter Refer to the EXEV1F description.
[17]	Reserved	Reserved, the reset value must be maintained
[16]	EXEV3FM	External event 3 fast mode Refer to the EXEV1FM description.

[15:12]	EXEV3F	External event 3 filter Refer to the EXEV1F description.
[11]	Reserved	Reserved, the reset value must be maintained
[10]	EXEV2FM	External event 2 fast mode Refer to the EXEV1FM description.
[9:6]	EXEV2F	External event 2 filter Refer to the EXEV1F description.
[5]	Reserved	Reserved, the reset value must be maintained
[4]	EXEV1FM	External event 1 fast mode 0: External event 1 is re-synchronized by the SHRTIM logic before acting on outputs, which adds a f_{SHRTIM} clock-related latency 1: External event 1 is acting asynchronously on outputs (low latency mode) *Note: This bit must not be modified once the counter in which the event is used is enabled (SHRTIM_MCTRL.TxCNTEN bit set).*
[3:0]	EXEV1F	External event 1 filter This bitfield defines the frequency used to sample external event 1 input and the length of the digital filter applied to shrtim_exev1. The digital filter is made of a counter in which N valid samples are needed to validate a transition on the output. 0000: Filter disabled 0001: $f_{SAMPLING}$= f_{SHRTIM}, N=2 0010: $f_{SAMPLING}$= f_{SHRTIM}, N=4 0011: $f_{SAMPLING}$= f_{SHRTIM}, N=8 0100: $f_{SAMPLING}$= f_{EXEVS}/2, N=6 0101: $f_{SAMPLING}$= f_{EXEVS}/2, N=8 0110: $f_{SAMPLING}$= f_{EXEVS}/4, N=6 0111: $f_{SAMPLING}$= f_{EXEVS}/4, N=8 1000: $f_{SAMPLING}$= f_{EXEVS}/8, N=6 1001: $f_{SAMPLING}$= f_{EXEVS}/8, N=8 1010: $f_{SAMPLING}$= f_{EXEVS}/16, N=5 1011: $f_{SAMPLING}$= f_{EXEVS}/16, N=6 1100: $f_{SAMPLING}$= f_{EXEVS}/16, N=8 1101: $f_{SAMPLING}$= f_{EXEVS}/32, N=5 1110: $f_{SAMPLING}$= f_{EXEVS}/32, N=6 1111: $f_{SAMPLING}$= f_{EXEVS}/32, N=8

9.4.3.16 SHRTIM External Event Control Register 4 (SHRTIM_EXEVCTRL4)

Address offset: 0x3BC

Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EXEVSCD		Reserved	EXEV10FM	EXEV10F				Reserved	EXEV9FM	EXEV9F				Reserved	EXEV8FM
rw			rw	rw					rw	rw					rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXEV8F				Reserved	EXEV7FM	EXEV7F				Reserved	EXEV6FM	EXEV6F			
rw					rw	rw					rw	rw			

Bit Field	Name	Description
[31:30]	EXEVSCD	External event sampling clock division This bitfield indicates the division ratio between the timer clock frequency (f_{SHRTIM}) and the external event signal sampling clock (f_{EXEVS}) used by the digital filters. 00: $f_{EXEVS}=f_{SHRTIM}$ 01: $f_{EXEVS}=f_{SHRTIM}/2$ 10: $f_{EXEVS}=f_{SHRTIM}/4$ 11: $f_{EXEVS}=f_{SHRTIM}/8$
[29]	Reserved	Reserved, the reset value must be maintained
[28]	EXEV10FM	External event 10 fast mode Refer to the EXEV1FM description.
[27:24]	EXEV10F	External event 10 filter Refer to the EXEV1F description.
[23]	Reserved	Reserved, the reset value must be maintained
[22]	EXEV9FM	External event 9 fast mode Refer to the EXEV1FM description.
[21:18]	EXEV9F	External event 9 filter Refer to the EXEV1F description.
[17]	Reserved	Reserved, the reset value must be maintained
[16]	EXEV8FM	External event 8 fast mode Refer to the EXEV1FM description.
[15:12]	EXEV8F	External event 8 filter Refer to the EXEV1F description.
[11]	Reserved	Reserved, the reset value must be maintained
[10]	EXEV7FM	External event 7 fast mode Refer to the EXEV1FM description.
[9:6]	EXEV7F	External event 7 filter Refer to the EXEV1F description.
[5]	Reserved	Reserved, the reset value must be maintained
[4]	EXEV6FM	External event 6 fast mode Refer to the EXEV1FM description.
[3:0]	EXEV6F	External event 6 filter Refer to the EXEV1Fdescription.

9.4.3.17 SHRTIM ADC Trigger 1 Sources Register 1 (SHRTIM_ADTG1SRC1)

Address offset: 0x3C0

Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

Reserved					ADTG1TCPRD	ADTG1TCCMP5	ADTG1TCCMP4	ADTG1TCCMP3	ADTG1TCCMP2	ADTG1TCCMP1	ADTG1TBRSTRO	ADTG1TBPRD	ADTG1TBCMP5	ADTG1TBCMP4	ADTG1TBCMP3
					rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADTG1TBCMP2	ADTG1TBCMP1	ADTG1TARSTRO	ADTG1TAPRD	ADTG1TACMP5	ADTG1TACMP4	ADTG1TACMP3	ADTG1TACMP2	ADTG1TACMP1	Reserved	ADTG1MPRD	Reserved	ADTG1MCMP4	ADTG1MCMP3	ADTG1MCMP2	ADTG1MCMP1
rw	rw	rw	rw	rw	rw	rw	rw	rw		rw		rw	rw	rw	rw

Bit Field	Name	Description
[31]	Reserved	Reserved, the reset value must be maintained
[30]	Reserved	Reserved, the reset value must be maintained
[29]	Reserved	Reserved, the reset value must be maintained
[28]	Reserved	Reserved, the reset value must be maintained
[27]	Reserved	Reserved, the reset value must be maintained
[26]	ADTG1TCPRD	ADC trigger 1 driven by timer C period event 0: No action 1: ADC trigger 1 is configured to be driven by timer C period event
[25]	ADTG1TCCMP5	ADC trigger 1 driven by timer C compare 5 event 0: No action 1: ADC trigger 1 is configured to be driven by timer C compare 5 event
[24]	ADTG1TCCMP4	ADC trigger 1 driven by timer C compare 4 event 0: No action 1: ADC trigger 1 is configured to be driven by timer C compare 4 event
[23]	ADTG1TCCMP3	ADC trigger 1 driven by timer C compare 3 event 0: No action 1: ADC trigger 1 is configured to be driven by timer C compare 3 event
[22]	ADTG1TCCMP2	ADC trigger 1 driven by timer C compare 2 event 0: No action 1: ADC trigger 1 is configured to be driven by timer C compare 2 event
[21]	ADTG1TCCMP1	ADC trigger 1 driven by timer C compare 1 event 0: No action 1: ADC trigger 1 is configured to be driven by timer C compare 1 event
[20]	ADTG1TBRSTRO	ADC trigger 1 driven by timer B reset or roll-over event 0: No action 1: ADC trigger 1 is configured to be driven by timer B reset or roll-over event
[19]	ADTG1TBPRD	ADC trigger 1 driven by timer B period event 0: No action 1: ADC trigger 1 is configured to be driven by timer B period event
[18]	ADTG1TBCMP5	ADC trigger 1 driven by timer B compare 5 event 0: No action 1: ADC trigger 1 is configured to be driven by timer B compare 5 event
[17]	ADTG1TBCMP4	ADC trigger 1 driven by timer B compare 4 event 0: No action 1: ADC trigger 1 is configured to be driven by timer B compare 4 event

[16]	ADTG1TBCMP3	ADC trigger 1 driven by timer B compare 3 event 0: No action 1: ADC trigger 1 is configured to be driven by timer B compare 3 event
[15]	ADTG1TBCMP2	ADC trigger 1 driven by timer B compare 2 event 0: No action 1: ADC trigger 1 is configured to be driven by timer B compare 2 event
[14]	ADTG1TBCMP1	ADC trigger 1 driven by timer B compare 1 event 0: No action 1: ADC trigger 1 is configured to be driven by timer B compare 1 event
[13]	ADTG1TARSTRO	ADC trigger 1 driven by timer A reset/roll-over event 0: No action 1: ADC trigger 1 is configured to be driven by timer A reset/roll-over event
[12]	ADTG1TAPRD	ADC trigger 1 driven by timer A period event 0: No action 1: ADC trigger 1 is configured to be driven by timer A period event
[11]	ADTG1TACMP5	ADC trigger 1 driven by timer A compare 5 event 0: No action 1: ADC trigger 1 is configured to be driven by timer A compare 5 event
[10]	ADTG1TACMP4	ADC trigger 1 driven by timer A compare 4 event 0: No action 1: ADC trigger 1 is configured to be driven by timer A compare 4 event
[9]	ADTG1TACMP3	ADC trigger 1 driven by timer A compare 3 event 0: No action 1: ADC trigger 1 is configured to be driven by timer A compare 3 event
[8]	ADTG1TACMP2	ADC trigger 1 driven by timer A compare 2 event 0: No action 1: ADC trigger 1 is configured to be driven by timer A compare 2 event
[7]	ADTG1TACMP1	ADC trigger 1 driven by timer A compare 1 event 0: No action 1: ADC trigger 1 is configured to be driven by timer A compare 1 event
[6]	Reserved	Reserved, the reset value must be maintained
[5]	ADTG1MPRD	ADC trigger 1 driven by master timer period event 0: No action 1: ADC trigger 1 is configured to be driven by master timer period event
[4]	Reserved	Reserved, the reset value must be maintained
[3]	ADTG1MCMP4	ADC trigger 1 driven by master timer compare 4 event 0: No action 1: ADC trigger 1 is configured to be driven by master timer compare 4 event
[2]	ADTG1MCMP3	ADC trigger 1 driven by master timer compare 3 event 0: No action 1: ADC trigger 1 is configured to be driven by master timer compare 3

		event
[1]	ADTG1MCMP2	ADC trigger 1 driven by master timer compare 2 event 0: No action 1: ADC trigger 1 is configured to be driven by master timer compare 2 event
[0]	ADTG1MCMP1	ADC trigger 1 driven by master timer compare 1 event 0: No action 1: ADC trigger 1 is configured to be driven by master timer compare 1 event

9.4.3.18 SHRTIM ADC Trigger 1 Sources Register 2 (SHRTIM_ADTG1SRC2)

Address offset: 0x3C4

Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved						ADTG1EXEV5	ADTG1EXEV4	ADTG1EXEV3	ADTG1EXEV2	ADTG1EXEV1	ADTG1TFRSTRO	ADTG1TFPRD	ADTG1TFCMP5	ADTG1TFCMP4	ADTG1TFCMP3
						rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADTG1TFCMP2	ADTG1TFCMP1	Reserved	ADTG1TEPRD	ADTG1TECMP5	ADTG1TECMP4	ADTG1TECMP3	ADTG1TECMP2	ADTG1TECMP1	Reserved	ADTG1TDPRD	ADTG1TDCMP5	ADTG1TDCMP4	ADTG1TDCMP3	ADTG1TDCMP2	ADTG1TDCMP1
rw	rw		rw	rw	rw	rw	rw	rw		rw	rw	rw	rw	rw	rw

Bit Field	Name	Description
[31]	Reserved	Reserved, the reset value must be maintained
[30]	Reserved	Reserved, the reset value must be maintained
[29]	Reserved	Reserved, the reset value must be maintained
[28]	Reserved	Reserved, the reset value must be maintained
[27]	Reserved	Reserved, the reset value must be maintained
[26]	Reserved	Reserved, the reset value must be maintained
[25]	ADTG1EXEV5	ADC trigger 1 driven by external event 5 0: No action 1: ADC trigger 1 is configured to be driven by external event 5
[24]	ADTG1EXEV4	ADC trigger 1 driven by external event 4 0: No action 1: ADC trigger 1 is configured to be driven by external event 4
[23]	ADTG1EXEV3	ADC trigger 1 driven by external event 3 0: No action 1: ADC trigger 1 is configured to be driven by external event 3
[22]	ADTG1EXEV2	ADC trigger 1 driven by external event 2 0: No action 1: ADC trigger 1 is configured to be driven by external event 2
[21]	ADTG1EXEV1	ADC trigger 1 driven by external event 1 0: No action 1: ADC trigger 1 is configured to be driven by external event 1

[20]	ADTG1TFRSTRO	ADC trigger 1 driven by timer F reset and roll-over events 0: No action 1: ADC trigger 1 is configured to be driven by timer F reset and roll-over events
[19]	ADTG1TFPRD	ADC trigger 1 driven by timer F period events 0: No action 1: ADC trigger 1 is configured to be driven by timer F period event
[18]	ADTG1TFCMP5	ADC trigger 1 driven by timer F compare 5 event 0: No action 1: ADC trigger 1 is configured to be driven by timer F compare 5 event
[17]	ADTG1TFCMP4	ADC trigger 1 driven by timer F compare 4 event 0: No action 1: ADC trigger 1 is configured to be driven by timer F compare 4 event
[16]	ADTG1TFCMP3	ADC trigger 1 driven by timer F compare 3 event 0: No action 1: ADC trigger 1 is configured to be driven by timer F compare 3 event
[15]	ADTG1TFCMP2	ADC trigger 1 driven by timer F compare 2 event 0: No action 1: ADC trigger 1 is configured to be driven by timer F compare 2 event
[14]	ADTG1TFCMP1	ADC trigger 1 driven by timer F compare 1 event 0: No action 1: ADC trigger 1 is configured to be driven by timer F compare 1 event
[13]	Reserved	Reserved, the reset value must be maintained
[12]	ADTG1TEPRD	ADC trigger 1 driven by timer E period event 0: No action 1: ADC trigger 1 is configured to be driven by timer E period event
[11]	ADTG1TECMP5	ADC trigger 1 driven by timer E compare 5 event 0: No action 1: ADC trigger 1 is configured to be driven by timer E compare 5 event
[10]	ADTG1TECMP4	ADC trigger 1 driven by timer E compare 4 event 0: No action 1: ADC trigger 1 is configured to be driven by timer E compare 4 event
[9]	ADTG1TECMP3	ADC trigger 1 driven by timer E compare 3 event 0: No action 1: ADC trigger 1 is configured to be driven by timer E compare 3 event
[8]	ADTG1TECMP2	ADC trigger 1 driven by timer E compare 2 event 0: No action 1: ADC trigger 1 is configured to be driven by timer E compare 2 event
[7]	ADTG1TECMP1	ADC trigger 1 driven by timer E compare 1 event 0: No action 1: ADC trigger 1 is configured to be driven by timer E compare 1 event
[6]	Reserved	Reserved, the reset value must be maintained
[5]	ADTG1TDPRD	ADC trigger 1 driven by timer D period event

		0: No action 1: ADC trigger 1 is configured to be driven by timer D period event
[4]	ADTG1TDCMP5	ADC trigger 1 driven by timer D compare 5 event 0: No action 1: ADC trigger 1 is configured to be driven by timer D compare 5 event
[3]	ADTG1TDCMP4	ADC trigger 1 driven by timer D compare 4 event 0: No action 1: ADC trigger 1 is configured to be driven by timer D compare 4 event
[2]	ADTG1TDCMP3	ADC trigger 1 driven by timer D compare 3 event 0: No action 1: ADC trigger 1 is configured to be driven by timer D compare 3 event
[1]	ADTG1TDCMP2	ADC trigger 1 driven by timer D compare 2 event 0: No action 1: ADC trigger 1 is configured to be driven by timer D compare 2 event
[0]	ADTG1TDCMP1	ADC trigger 1 driven by timer D compare 1 event 0: No action 1: ADC trigger 1 is configured to be driven by timer D compare 1 event

9.4.3.19 SHRTIM ADC Trigger 2 Sources Register 1 (SHRTIM_ADTG2SRC1)

Address offset: 0x3C8

Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				ADTG2TCRSTRO	ADTG2TCPRD	ADTG2TCCMP5	ADTG2TCCMP4	ADTG2TCCMP3	ADTG2TCCMP2	ADTG2TCCMP1	Reserved	ADTG2TBPRD	ADTG2TBCMP5	ADTG2TBCMP4	ADTG2TBCMP3
				rw	rw	rw	rw	rw	rw	rw		rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADTG2TBCMP2	ADTG2TBCMP1	Reserved	ADTG2TAPRD	ADTG2TACMP5	ADTG2TACMP4	ADTG2TACMP3	ADTG2TACMP2	ADTG2TACMP1	Reserved	ADTG2MPRD	Reserved	ADTG2MCMP4	ADTG2MCMP3	ADTG2MCMP2	ADTG2MCMP1
rw	rw		rw	rw	rw	rw	rw	rw		rw		rw	rw	rw	rw

Refer to the SHRTIM_ADTG1SRC1 description.

9.4.3.20 SHRTIM ADC Trigger 2 Sources Register 2 (SHRTIM_ADTG2SRC2)

Address offset: 0x3CC

Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved						ADTG2EXEV10	ADTG2EXEV9	ADTG2EXEV8	ADTG2EXEV7	ADTG2EXEV6	Reserved	ADTG2TFPRD	ADTG2TFCMP5	ADTG2TFCMP4	ADTG2TFCMP3
						rw	rw	rw	rw	rw		rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADTG2TFCMP2	ADTG2TFCMP1	ADTG2TERSTRO	Reserved	ADTG2TECMP5	ADTG2TECMP4	ADTG2TECMP3	ADTG2TECMP2	ADTG2TECMP1	ADTG2TDRSTRO	ADTG2TDPRD	ADTG2TDCMP5	ADTG2TDCMP4	ADTG2TDCMP3	ADTG2TDCMP2	ADTG2TDCMP1
rw	rw	rw		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Refer to the SHRTIM_ADTG1SRC2 description.

9.4.3.21 SHRTIM ADC Trigger 3 Sources Register 1 (SHRTIM_ADTG3SRC1)

Address offset: 0x3D0

Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved					ADTG3TCPRD	ADTG3TCCMP5	ADTG3TCCMP4	ADTG3TCCMP3	ADTG3TCCMP2	ADTG3TCCMP1	ADTG3TBRSTRO	ADTG3TBPRD	ADTG3TBCMP5	ADTG3TBCMP4	ADTG3TBCMP3
					rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADTG3TBCMP2	ADTG3TBCMP1	ADTG3TARSTRO	ADTG3TAPRD	ADTG3TACMP5	ADTG3TACMP4	ADTG3TACMP3	ADTG3TACMP2	ADTG3TACMP1	Reserved	ADTG3MPRD	Reserved	ADTG3MCMP4	ADTG3MCMP3	ADTG3MCMP2	ADTG3MCMP1
rw	rw	rw	rw	rw	rw	rw	rw	rw		rw		rw	rw	rw	rw

Refer to the SHRTIM_ADTG1SRC1 description.

9.4.3.22 SHRTIM ADC Trigger 3 Sources Register 2 (SHRTIM_ADTG3SRC2)

Address offset: 0x3D4

Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved						ADTG3EXEV5	ADTG3EXEV4	ADTG3EXEV3	ADTG3EXEV2	ADTG3EXEV1	ADTG3TFRSTRO	ADTG3TFPRD	ADTG3TFCMP5	ADTG3TFCMP4	ADTG3TFCMP3
						rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADTG3TFCMP2	ADTG3TFCMP1	Reserved	ADTG3TEPRD	ADTG3TECMP5	ADTG3TECMP4	ADTG3TECMP3	ADTG3TECMP2	ADTG3TECMP1	Reserved	ADTG3TDPRD	ADTG3TDCMP5	ADTG3TDCMP4	ADTG3TDCMP3	ADTG3TDCMP2	ADTG3TDCMP1
rw	rw		rw	rw	Rw	rw	rw	rw		rw	rw	rw	rw	rw	rw

Refer to the SHRTIM_ADTG1SRC2 description.

9.4.3.23 SHRTIM ADC Trigger 4 Sources Register 1 (SHRTIM_ADTG4SRC1)

Address offset: 0x3D8

Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				ADTG4TCRSTRO	ADTG4TCPRD	ADTG4TCCMP5	ADTG4TCCMP4	ADTG4TCCMP3	ADTG4TCCMP2	ADTG4TCCMP1	Reserved	ADTG4TBPRD	ADTG4TBCMP5	ADTG4TBCMP4	ADTG4TBCMP3
				rw	rw	rw	rw	rw	rw	rw		rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADTG4TBCMP2	ADTG4TBCMP1	Reserved	ADTG4TAPRD	ADTG4TACMP5	ADTG4TACMP4	ADTG4TACMP3	ADTG4TACMP2	ADTG4TACMP1	Reserved	ADTG4MPRD	Reserved	ADTG4MCMP4	ADTG4MCMP3	ADTG4MCMP2	ADTG4MCMP1
rw	rw		rw	rw	rw	rw	rw	rw	r	rw		rw	rw	rw	rw

Refer to the SHRTIM_ADTG1SRC1 description.

9.4.3.24 SHRTIM ADC Trigger 4 Sources Register 2 (SHRTIM_ADTG4SRC2)

Address offset: 0x3DC

Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved						ADTG4 EXEV10	ADTG4 EXEV9	ADTG4 EXEV8	ADTG4 EXEV7	ADTG4EXEV6	Reserved	ADTG4 TFPRD	ADTG4 TFCMP5	ADTG4 TFCMP4	ADTG4 TFCMP3
						rw	rw	rw	rw	rw		rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADTG4 TFCMP2	ADTG4 TFCMP1	ADTG4 TERSTRO	Reserved	ADTG4 TECMP5	ADTG4 TECMP4	ADTG4 TECMP3	ADTG4 TECMP2	ADTG4 TECMP1	ADTG4 TDRSTRO	ADTG 4TDPRD	ADTG4 TDCMP5	ADTG4 TDCMP4	ADTG4 TDCMP3	ADTG4 TDCMP2	ADTG4 TDCMP1
rw	rw	rw		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Refer to the SHRTIM_ADTG1SRC2 description.

9.4.3.25 SHRTIM Fault Input Register 1 (SHRTIM_FALTIN1)

Address offset: 0x3E0

Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FALT4LCK	FALT4FLT				FALT4SRC0	FALT4POL	FALT4E	FALT3LCK	FALT3FLT				FALT3SRC0	FALT3POL	FALT3E
rw	rw				rw	rw	rw	rw	rw				rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FALT2LCK	FALT2FLT				FALT2SRC0	FALT2POL	FALT2E	FALT1LCK	FALT1FLT				FALT1SRC0	FALT1POL	FALT1E
rw	rw				rw	rw	rw	rw	rw				rw	rw	rw

Bit Field	Name	Description
[31]	FALT4LCK	Fault 4 Lock Refer to the SHRTIM_FALTIN1.FALT1LCK description.
[30:27]	FALT4FLT	Fault 4 filter Refer to the SHRTIM_FALTIN1.FALT1FLT[3:0] description.
[26]	FALT4SRC0	Fault 4 source bit 0 Refer to the SHRTIM_FALTIN1.FALT1SRC0 description.
[25]	FALT4POL	Fault 4 polarity Refer to the SHRTIM_FALTIN1.FALT1POL description.
[24]	FALT4E	Fault 4 enable Refer to the SHRTIM_FALTIN1.FALT1E description.
[23]	FALT3LCK	Fault 3 Lock Refer to the SHRTIM_FALTIN1.FALT1LCK description.
[22:19]	FALT3FLT	Fault 3 filter Refer to the SHRTIM_FALTIN1.FALT1FLT[3:0] description.
[18]	FALT3SRC0	Fault 3 source bit 0 Refer to the SHRTIM_FALTIN1.FALT1SRC0 description.
[17]	FALT3POL	Fault 3 polarity

		Refer to the SHRTIM_FALTIN1.FALT1POL description.
[16]	FALT3E	Fault 3 enable Refer to the SHRTIM_FALTIN1.FALT1E description.
[15]	FALT2LCK	Fault 2 Lock Refer to the SHRTIM_FALTIN1.FALT1LCK description.
[14:11]	FALT2FLT	Fault 2 filter Refer to the SHRTIM_FALTIN1.FALT1FLT[3:0] description.
[10]	FALT2SRC0	Fault 2 source bit 0 Refer to the SHRTIM_FALTIN1.ALT1SRC0 description.
[9]	FALT2POL	Fault 2 polarity Refer to the SHRTIM_FALTIN1.FALT1POL description.
[8]	FALT2E	Fault 2 enable Refer to the SHRTIM_FALTIN1.FALT1Edescription.
[7]	FALT1LCK	Fault 1 Lock The FALT1LCK bit modifies the write attributes of the fault programming bit, so that they are protected against spurious write accesses. This bit is write-once. Once it has been set, it cannot be modified till the next system reset. 0: FALT1E, FALT1POL, FALT1SRC[1:0], FALT1FLT[3:0] and FALT1CSEL[2:0] bits are read/write. 1: FALT1E, FALT1POL, FALT1SRC[1:0], FALT1FLT[3:0] and FALT1CSEL[2:0] bits cannot be written (read-only mode)
[6:3]	FALT1FLT	Fault 1 filter This bitfield defines the frequency used to sample FALT1 input and the length of the digital filter applied to FALT1. The digital filter is made of an event counter in which N events are needed to validate a transition on the output. 0000: No filter, FALT1 acts asynchronously 0001: $f_{SAMPLING} = f_{SHRTIM}$, N = 2 0010: $f_{SAMPLING} = f_{SHRTIM}$, N = 4 0011: $f_{SAMPLING} = f_{SHRTIM}$, N = 8 0100: $f_{SAMPLING} = f_{FALTS}/2$, N = 6 0101: $f_{SAMPLING} = f_{FALTS}/2$, N = 8 0110: $f_{SAMPLING} = f_{FALTS}/4$, N = 6 0111: $f_{SAMPLING} = f_{FALTS}/4$, N = 8 1000: $f_{SAMPLING} = f_{FALTS}/8$, N = 6 1001: $f_{SAMPLING} = f_{FALTS}/8$, N = 8 1010: $f_{SAMPLING} = f_{FALTS}/16$, N = 5 1011: $f_{SAMPLING} = f_{FALTS}/16$, N = 6 1100: $f_{SAMPLING} = f_{FALTS}/16$, N = 8 1101: $f_{SAMPLING} = f_{FALTS}/32$, N = 5 1110: $f_{SAMPLING} = f_{FALTS}/32$, N = 6 1111: $f_{SAMPLING} = f_{FALTS}/32$, N = 8

		Note: This bitfield is written only when FALT1E enable bit is reset. *Note: This bitfield is modified when FALT1LCK has been programmed.*
[2]	FALT1SRC0	Fault 1 source bit 0 The FALT1SRC[1:0] bitfield selects the FAULT1 input source (refer to Table9-29 for connection details). 00: Fault 1 input is SHRTIM_FALT1 input pin 01: Fault 1 input is a COMPx output 10: Fault 1 input is EXEV1_muxout input pin *Note: This bitfield is written only when FALT1E enable bit is reset.*
[1]	FALT1POL	Fault 1 polarity This bit selects the FAULT1 input polarity. 0: Fault 1 input is active low 1: Fault 1 input is active high *Note: This bitfield is written only when FALT1E enable bit is reset.*
[0]	FALT1E	Fault 1 enable This bit enables the global FAULT1 input circuitry. 0: Fault 1 input disabled 1: Fault 1 input enabled

9.4.3.26 SHRTIM Fault Input Register 2 (SHRTIM_FALTIN2)

Address offset: 0x3E4

Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SFALTCKSECEN	SFALTLOCKUPEN	SFALTPVDEN	SFALTSMECCEN	SFALTFLECCEN	SFALTSMPAREN	FALTSCD		Reserved		FALT6SRC1	FALT5SRC1	FALT4SRC1	FALT3SRC1	FALT2SRC1	FALT1SRC1
rw	rw	rw	rw	rw	rw	rw				rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FALT6LCK	FALT6FLT				FALT6SRC0	FALT6POL	FALT6E	FALT5LCK	FALT5FLT				FALT5SRC0	FALT5POL	FALT5E
rw	rw				rw	rw	rw	rw	rw				rw	rw	rw

Bit Field	Name	Description
[31]	SFALTCKSECEN	The clock security system error as SHRTIM system fault input enable This bit is set by software and can only be cleared by a system reset or a power-on reset (POR).
[30]	SFALTLOCKUPEN	The core lockup as SHRTIM system fault input enable This bit is set by software and can only be cleared by a system reset or a power-on reset (POR).
[29]	SFALTPVDEN	The PVD error as SHRTIM system fault input enable This bit is set by software and can only be cleared by a system reset or a power-on reset (POR).
[28]	SFALTSMECCEN	The sram ECC error as SHRTIM system fault input enable This bit is set by software and can only be cleared by a system reset or a power-on reset (POR).

[27]	SFALTFLECCEN	The flash ECC double detection error as SHRTIM system fault input enable This bit is set by software and can only be cleared by a system reset or a power-on reset (POR).
[26]	SFALTSMPAREN	The sram parity error as SHRTIM system fault input enable This bit is set by software and can only be cleared by a system reset or a power-on reset (POR).
[25:24]	FALTSCD	Fault Sampling clock division This bitfield indicates the division ratio between the timer clock frequency (f_{SHRTIM}) and the fault signal sampling clock (f_{FALTS}) used by the digital filters. 00: f_{FALTS}=f_{SHRTIM} 01: f_{FALTS}=f_{SHRTIM} / 2 10: f_{FALTS}=f_{SHRTIM} / 4 11: f_{FALTS}=f_{SHRTIM} / 8 *Note: This bitfield must be written prior to any of the FALTxE enable bits.*
[23:22]	Reserved	Reserved, the reset value must be maintained
[21]	FALT6SRC1	Fault 6 source bit 1 Refer to the FALT6SRC_0 description.
[20]	FALT5SRC1	Fault 5 source bit 1 Refer to the FALT5SRC_0 description.
[19]	FALT4SRC1	Fault 4 source bit 1 Refer to the FALT4SRC_0 description.
[18]	FALT3SRC1	Fault 3 source bit 1 Refer to the FALT3SRC_0 description.
[17]	FALT2SRC1	Fault 2 source bit 1 Refer to the FALT2SRC_0 description.
[16]	FALT1SRC1	Fault 1 source bit 1 Refer to the FALT1SRC_0 description.
[15]	FALT6LCK	Fault 6 Lock Refer to the SHRTIM_FALTIN1.FALT1LCK description.
[14:11]	FALT6FLT	Fault 6 filter Refer to the SHRTIM_FALTIN1.FALT1FLT[3:0] description.
[10]	FALT6SRC0	Fault 6 source bit 0 Refer to the SHRTIM_FALTIN1.FALT1SRC0 description.
[9]	FALT6POL	Fault 6 polarity Refer to the SHRTIM_FALTIN1.FALT1POL description.
[8]	FALT6E	Fault 6 enable Refer to the SHRTIM_FALTIN1.FALT1E description.
[7]	FALT5LCK	Fault 5 Lock Refer to the SHRTIM_FALTIN1.FALT1LCK description.
[6:3]	FALT5FLT	Fault 5 filter Refer to the SHRTIM_FALTIN1.FALT1FLT[3:0] description.
[2]	FALT5SRC0	Fault 5 source bit 0

		Refer to the SHRTIM_FALTIN1.FALT1SRC0 description.
[1]	FALT5POL	Fault 5 polarity Refer to the SHRTIM_FALTIN1.FALT1POL description.
[0]	FALT5E	Fault 5 enable Refer to the SHRTIM_FALTIN1.FALT1E description.

9.4.3.27 SHRTIM Fault Input Register 3 (SHRTIM_FALTIN3)

Address offset: 0x3E8

Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FALT4RSTM	FALT4CRST	FALT4CNT				FALT4BLKS	FALT4BLKEN	FALT3RSTM	FALT3CRST	FALT3CNT				FALT3BLKS	FALT3BLKEN
rw	rt_w	rw				rw	rw	rw	rt_w	rw				rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FALT2RSTM	FALT2CRST	FALT2CNT				FALT2BLKS	FALT2BLKEN	FALT1RSTM	FALT1CRST	FALT1CNT				FALT1BLKS	FALT1BLKEN
rw	rt_w	rw				rw	rw	rw	rt_w	rw				rw	rw

Bit Field	Name	Description
[31]	FALT4RSTM	Fault 4 reset mode Refer to the FALT1RSTM description.
[30]	FALT4CRST	Fault 4 counter reset Refer to the FALT1CRST description.
[29:26]	FALT4CNT	Fault 4 counter Refer to the FALT1CNT description.
[25]	FALT4BLKS	Fault 4 blanking source Refer to the FALT1BLKS description.
[24]	FALT4BLKEN	Fault 4 blanking enable Refer to the FALT1BLKEN description.
[23]	FALT3RSTM	Fault 3 reset mode Refer to the FALT1RSTM description.
[22]	FALT3CRST	Fault 3 counter reset Refer to the FALT1CRST description.
[21:18]	FALT3CNT	Fault 3 counter Refer to the FALT1CNT description.
[17]	FALT3BLKS	Fault 3 blanking source Refer to the FALT1BLKS description.
[16]	FALT3BLKEN	Fault 3 blanking enable Refer to the FALT1BLKEN description.
[15]	FALT2RSTM	Fault 2 reset mode Refer to the FALT1RSTM description.
[14]	FALT2CRST	Fault 2 counter reset Refer to the FALT1CRST description.
[13:10]	FALT2CNT	Fault 2 counter

		Refer to the FALT1CNT description.
[9]	FALT2BLKS	Fault 2 blanking source Refer to the FALT1BLKS description.
[8]	FALT2BLKEN	Fault 2 blanking enable Refer to the FALT1BLKEN description.
[7]	FALT1RSTM	Fault 1 reset mode This bit selects the FAULT1 counter reset mode 0: Fault 1 counter is reset on each reset/roll-over event 1: Fault 1 counter is reset on each reset/roll-over event only if no fault occurred during last counting period. *Note: This bitfield is written only when FALT1E enable bit is reset.*
[6]	FALT1CRST	Fault 1 counter reset This bit resets the FAULT1 counter. It is set by software and reset by hardware. 0: No action 1: Fault 1 counter is reset
[5:2]	FALT1CNT	Fault 1 counter This bitfield selects the FAULT1 counter threshold. A fault is considered valid when the number of events is equal to the (FALT1CNT[3:0]+1) value.
[1]	FALT1BLKS	Fault 1 blanking source The FATL1BLKS bit selects the FAULT1 blanking source (refer to Table9-31 for details). 0: Fault 1 reset-aligned blanking window 1: Fault 1 moving blanking window *Note: This bitfield is written only when FALT1E enable bit is reset.*
[0]	FALT1BLKEN	Fault 1 blanking enable The FATL1BLKEN bit selects the FAULT1 blanking mode. The blanking source is defined by the FALT1BLKS bit. 0: No blanking on fault 1 1: Fault 1 blanking mode *Note: This bitfield is written only when FALT1E enable bit is reset*

9.4.3.28 SHRTIM Fault Input Register 4 (SHRTIM_FALTIN4)

Address offset: 0x3EC

Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FALT6RSTM	FALT6CRST	FALT6CNT				FALT6BLKS	FALT6BLKEN	FALT5RSTM	FALT5CRST	FALT5CNT				FALT5BLKS	FALT5BLKEN

rw	rt_w	rw	rw	rw	rw	rt_w	rw	rw	rw

Bit Field	Name	Description
[31:16]	Reserved	Reserved, the reset value must be maintained
[15]	FALT6RSTM	Fault 6 reset mode Refer to the FALT1RSTM description.
[14]	FALT6CRST	Fault 6 counter reset Refer to the FALT1CRST description.
[13:10]	FALT6CNT	Fault 6 counter Refer to the FALT1CNT description.
[9]	FALT6BLKS	Fault 6 blanking source Refer to the FALT1BLKS description.
[8]	FALT6BLKEN	Fault 6 blanking enable Refer to the FALT1BLKEN description.
[7]	FALT5RSTM	Fault 5 reset mode Refer to the FALT1RSTM description.
[6]	FALT5CRST	Fault 5 counter reset Refer to the FALT1CRST description.
[5:2]	FALT5CNT	Fault 5 counter Refer to the FALT1CNT description.
[1]	FALT5BLKS	Fault 5 blanking source Refer to the FALT1BLKS description.
[0]	FALT5BLKEN	Fault 5 blanking enable Refer to the FALT1BLKEN description.

9.4.3.29 SHRTIM Burst DMA Master Timer Update Register (SHRTIM_BDMTUPD)

Address offset: 0x3F0

Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						MCMPDAT4	MCMPDAT3	MCMPDAT2	MCMPDAT1	MREPT	MPRD	MCNT	MIDEN	MINTCLR	MCTRL
						rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit Field	Name	Description
[31:10]	Reserved	Reserved, the reset value must be maintained
[9]	MCMPDAT4	SHRTIM_MCMP4DAT register update enable 0: SHRTIM_MCMP4DAT register is not updated by burst DMA accesses 1: SHRTIM_MCMP4DAT register is updated by burst DMA accesses
[8]	MCMPDAT3	SHRTIM_MCMP3DAT register update enable 0：SHRTIM_MCMP3DAT register is not updated by burst DMA accesses

		1：SHRTIM_MCMP3DAT register is updated by burst DMA accesses
[7]	MCMPDAT2	SHRTIM_MCMP2DAT register update enable 0：SHRTIM_MCMP2DAT register is not updated by burst DMA accesses 1：SHRTIM_MCMP2DAT register is updated by burst DMA accesses
[6]	MCMPDAT1	SHRTIM_MCMP1DAT register update enable 0：SHRTIM_MCMP1DAT register is not updated by burst DMA accesses 1：SHRTIM_MCMP1DAT register is updated by burst DMA accesses
[5]	MREPT	SHRTIM_MREPT register update enable 0：SHRTIM_MREPT register is not updated by burst DMA accesses 1：SHRTIM_MREPT register is updated by burst DMA accesses
[4]	MPRD	SHRTIM_MPRD register update enable 0：SHRTIM_MPRD register is not updated by burst DMA accesses 1：SHRTIM_MPRD register is updated by burst DMA accesses
[3]	MCNT	SHRTIM_MCNT register update enable 0：SHRTIM_MCNT register is not updated by burst DMA accesses 1：SHRTIM_MCNT register is updated by burst DMA accesses
[2]	MIDEN	SHRTIM_MIDEN register update enable 0：SHRTIM_MIDEN register is not updated by burst DMA accesses 1 ：SHRTIM_MIDEN register is updated by burst DMA accesses
[1]	MINTCLR	SHRTIM_MINTCLR register update enable 0：SHRTIM_MINTCLR register is not updated by burst DMA accesses 1：SHRTIM_MINTCLR register is updated by burst DMA accesses
[0]	MCTRL	SHRTIM_CTRL register update enable 0：SHRTIM_CTRL register is not updated by burst DMA accesses 1：SHRTIM_CTRL register is updated by burst DMA accesses

9.4.3.30 SHRTIM Burst DMA Timer x Update Register (SHRTIM_BDTxUPD)

Address offset: 0x3F4

Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved									TxEXEVFLT3	TxCTRL2	TxFALT	TxOUT	TxCHOP	TxCNTRST	TxEXEVFLT2
									rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TxEXEVFLT1	TxRST2	TxSET2	TxRST1	TxSET1	TxDT	TxCMP4DAT	TxCMP3DAT	TxCMP2DAT	TxCMP1DAT	TxREPT	TxPRD	TxCNT	TxIDEN	TxINTCLR	TxCTRL
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit Field	Name	Description
[31:23]	Reserved	Reserved, the reset value must be maintained
[22]	TxEXEVFLT3	SHRTIM_TxEXEVFLT3 register update enable 0: Burst DMA access does not cause the TxEXEVFLT3 to update. 1: Burst DMA access causes the TxEXEVFLT3 to update.
[21]	TxCTRL2	SHRTIM_TxCTRL2 register update enable

		0: Burst DMA access does not cause the SHRTIM_TxCTRL2 to update. 1: Burst DMA access cause the SHRTIM_TxCTRL2 to update.
[20]	TxFALT	SHRTIM_TxFALT register update enable 0: Burst DMA access does not cause the SHRTIM_TxFALT to update. 1: Burst DMA access cause the SHRTIM_TxFALT to update.
[19]	TxOUT	SHRTIM_TxOUT register update enable 0: Burst DMA access does not cause the SHRTIM_TxOUT to update. 1: Burst DMA access cause the SHRTIM_TxOUT to update.
[18]	TxCHOP	SHRTIM_TxCHOP register update enable 0: Burst DMA access does not cause the SHRTIM_TxCHOP to update. 1: Burst DMA access cause the SHRTIM_TxCHOP to update.
[17]	TxCNTRST	SHRTIM_TxCNTRST register update enable 0: Burst DMA access does not cause the SHRTIM_TxCNTRST to update. 1: Burst DMA access cause the SHRTIM_TxCNTRST to update.
[16]	TxEXEVFLT2	SHRTIM_TxEXEVFLT2 register update enable 0: Burst DMA access does not cause the SHRTIM_TxEXEVFLT2 to update. 1: Burst DMA access cause the SHRTIM_TxEXEVFLT2 to update.
[15]	TxEXEVFLT1	SHRTIM_TxEXEVFLT1 register update enable 0: Burst DMA access does not cause the SHRTIM_TxEXEVFLT1 to update. 1: Burst DMA access cause the SHRTIM_TxEXEVFLT1 to update.
[14]	TxRST2	SHRTIM_TxRST2 register update enable 0: Burst DMA access does not cause the SHRTIM_TxRST2 to update. 1: Burst DMA access cause the SHRTIM_TxRST2 to update.
[13]	TxSET2	SHRTIM_TxSET2 register update enable 0: Burst DMA access does not cause the SHRTIM_TxSET2 to update. 1: Burst DMA access cause the SHRTIM_TxSET2 to update.
[12]	TxRST1	SHRTIM_TxRST1 register update enable 0: Burst DMA access does not cause the SHRTIM_TxRST1 to update. 1: Burst DMA access cause the SHRTIM_TxRST1 to update.
[11]	TxSET1	SHRTIM_TxSET1 register update enable 0: Burst DMA access does not cause the SHRTIM_TxSET1 to update. 1: Burst DMA access cause the SHRTIM_TxSET1 to update.
[10]	TxDT	SHRTIM_TxDT register update enable 0: Burst DMA access does not cause the SHRTIM_TxDT to update. 1: Burst DMA access cause the SHRTIM_TxDT to update.
[9]	TxCMP4DAT	SHRTIM_TxCMP4DAT register update enable 0: Burst DMA access does not cause the SHRTIM_TxCMP4DAT to update. 1: Burst DMA access cause the SHRTIM_TxCMP4DAT to update.
[8]	TxCMP3DAT	SHRTIM_TxCMP3DAT register update enable 0: Burst DMA access does not cause the SHRTIM_TxCMP3DAT to update. 1: Burst DMA access cause the SHRTIM_TxCMP3DAT to update.
[7]	TxCMP2DAT	SHRTIM_TxCMP2DAT register update enable 0: Burst DMA access does not cause the SHRTIM_TxCMP2DAT to update.

		1: Burst DMA access cause the SHRTIM_TxCMP2DAT to update.
[6]	TxCMP1DAT	SHRTIM_TxCMP1DAT register update enable 0: Burst DMA access does not cause the SHRTIM_TxCMP1DAT to update. 1: Burst DMA access cause the SHRTIM_TxCMP1DAT to update.
[5]	TxREPT	SHRTIM_TxREPT register update enable 0: Burst DMA access does not cause the SHRTIM_TxREPT to update. 1: Burst DMA access cause the SHRTIM_TxREPT to update.
[4]	TxPRD	SHRTIM_TxPRD register update enable 0: Burst DMA access does not cause the SHRTIM_TxPRD to update. 1: Burst DMA access cause the SHRTIM_TxPRD to update.
[3]	TxCNT	SHRTIM_TxCNT register update enable 0: Burst DMA access does not cause the SHRTIM_TxCNT to update. 1: Burst DMA access cause the SHRTIM_TxCNT to update.
[2]	TxIDEN	SHRTIM_TxIDEN register update enable 0: Burst DMA access does not cause the SHRTIM_TxIDEN to update. 1: Burst DMA access cause the SHRTIM_TxIDEN to update.
[1]	TxINTCLR	SHRTIM_TxINTCLR register update enable 0: Burst DMA access does not cause the SHRTIM_TxINTCLR to update. 1: Burst DMA access cause the SHRTIM_TxINTCLR to update.
[0]	TxCTRL	SHRTIM_TxCTRL register update enable 0: Burst DMA access does not cause the SHRTIM_TxCTRL to update. 1: Burst DMA access cause the SHRTIM_TxCTRL to update.

9.4.3.31 SHRTIM Burst DMA Data Register (SHRTIM_BDDAT)

Address offset: 0x40C

Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BDMADATA															
w															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BDMADATA															
w															

Bit Field	Name	Description
[31:0]	BDMADATA	Burst DMA data register When a write operation is performed on this register, the register value will be loaded into the register enabled by SHRTIM_BDTxUPD or SHRTIM_BDMTUPD, which will also cause the increment of the register pointer to the next location to be filled.

9.4.3.32 SHRTIM ADC Trigger Extended Register 1 (SHRTIM_ADTGEX1)

Address offset: 0x410

Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved										ADTG9SRC					
										rw					

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		ADTG7SRC						Reserved		ADTG5SRC					
		rw								rw					

Bit Field	Name	Description
[31:22]	Reserved	Reserved, the reset value must be maintained
[21:16]	ADTG9SRC	ADC trig 9 source select Refer to the ADTG5SRC description.
[15:14]	Reserved	Reserved, the reset value must be maintained
[13:8]	ADTG7SRC	ADC trig 7 source select Refer to the ADTG5SRC description.
[7:6]	Reserved	Reserved, the reset value must be maintained
[5:0]	ADTG5SRC	ADC trig 5 source select 0: Trigger on master timer compare 1 1: Trigger on master timer compare 2 2: Trigger on master timer compare 3 3: Trigger on master timer compare 4 4: Trigger on master timer period 5: Trigger on external event 1 6: Trigger on external event 2 7: Trigger on external event 3 8: Trigger on external event 4 9: Trigger on external event 5 10: Trigger on timer A compare 1 11: Trigger on timer A compare 2 12: Trigger on timer A compare 3 13: Trigger on timer A compare 4 14: Trigger on timer A compare 5 15: Trigger on timer A period 16: Trigger on timer A reset and counter roll-over 17: Trigger on timer B compare 1 18: Trigger on timer B compare 2 19: Trigger on timer B compare 3 20: Trigger on timer B compare 4 21: Trigger on timer B compare 5 22: Trigger on timer B period 23: Trigger on timer B reset and counter roll-over 24: Trigger on timer C compare 1

		25: Trigger on timer C compare 2 26: Trigger on timer C compare 3 27: Trigger on timer C compare 4 28: Trigger on timer C compare 5 29: Trigger on timer C period 30: Trigger on timer D compare 1 31: Trigger on timer D compare 2 32: Trigger on timer D compare 3 33: Trigger on timer D compare 4 34: Trigger on timer D compare 5 35: Trigger on timer D period 36: Trigger on timer E compare 1 37: Trigger on timer E compare 2 38: Trigger on timer E compare 3 39: Trigger on timer E compare 4 40: Trigger on timer E compare 5 41: Trigger on timer E period 42: Trigger on timer F compare 1 43: Trigger on timer F compare 2 44: Trigger on timer F compare 3 45: Trigger on timer F compare 4 46: Trigger on timer F compare 5 47: Trigger on timer F period 48: Trigger on timer F reset and counter roll-over

9.4.3.33 SHRTIM ADC Trigger Extended Register 2 (SHRTIM_ADTGEX2)

Address offset: 0x414

Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved										ADTG10SRC					
										rw					

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		ADTG8SRC						Reserved		ADTG6SRC					
		rw								rw					

Bit Field	Name	Description
[31:22]	Reserved	Reserved, the reset value must be maintained
[21:16]	ADTG10SRC	ADC trig 10 source select Refer to the ADTG5SRC description.
[15:14]	Reserved	Reserved, the reset value must be maintained
[13:8]	ADTG8SRC	ADC trig 8 source select Refer to the ADTG5SRC description.

[7:6]	Reserved	Reserved, the reset value must be maintained
[5:0]	ADTG6SRC	ADC trig 6 source select 0: Triggered on the master timer compare 1 1: Triggered on the master timer compare 2 2: Triggered on the master timer compare 3 3: Triggered on the master timer compare 4 4: Triggered on the master timer period 5: Triggered on the external event 6 6: Triggered on the external event 7 7: Triggered on the external event 8 8: Triggered on the external event 9 9: Triggered on the external event 10 10: Triggered on the timer A compare 1 11: Triggered on the timer A compare 2 12: Triggered on the timer A compare 3 13: Triggered on the timer A compare 4 14: Triggered on the timer A compare 5 1: Triggered on the timer A period 16: Triggered on the timer B compare 1 17: Triggered on the timer B compare 2 18: Triggered on the timer B compare 3 19: Triggered on the timer B compare 4 20: Triggered on the timer B compare 5 21: Triggered on the timer B period 22: Triggered on the timer C compare 1 23: Triggered on the timer C compare 2 24: Triggered on the timer C compare 3 25: Triggered on the timer C compare 4 26: Triggered on the timer C compare 5 2: Triggered on the timer C period 28: Triggered on the timer C reset and counter roll-over 29: Triggered on the timer D compare 1 30: Triggered on the timer D compare 2 31: Triggered on the timer D compare 3 32: Triggered on the timer D compare 4 33: Triggered on the timer D compare 5 34: Triggered on the timer D period 35: Triggered on the timer D reset and counter roll-over 36: Triggered on the timer E compare 1 37: Triggered on the timer E compare 2 38: Triggered on the timer E compare 3 39: Triggered on the timer E compare 4 40: Triggered on the timer E compare 5

		41: Triggered on the timer E reset and counter roll-over 42: Triggered on the timer F compare 1 43: Triggered on the timer F compare 2 44: Triggered on the timer F compare 3 45: Triggered on the timer F compare 4 46: Triggered on the timer F compare 5 47: Triggered on the timer F period

9.4.3.34 SHRTIM ADC Trigger Update Register (SHRTIM_ADTGUPD)

Address offset: 0x418

Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved									ADTG10UPDSRC			Reserved	ADTG9UPDSRC		
									rw				rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	ADTG8UPDSRC			Reserved	ADTG7UPDSRC			Reserved	ADTG6UPDSRC			Reserved	ADTG5UPDSRC		
	rw				rw				rw				rw		

Bit Field	Name	Description
[31:23]	Reserved	Reserved, the reset value must be maintained
[22:20]	ADTG10UPDSRC	ADC trigger 10 update source Refer to the ADTG5UPDSRC[2:0] description
[19]	Reserved	Reserved, the reset value must be maintained
[18:16]	ADTG9UPDSRC	ADC trigger 9 update source Refer to the ADTG5UPDSRC[2:0] description
[15]	Reserved	Reserved, the reset value must be maintained
[14:12]	ADTG8UPDSRC	ADC trigger 8 update source Refer to the ADTG5UPDSRC[2:0] description
[11]	Reserved	Reserved, the reset value must be maintained
[10:8]	ADTG7UPDSRC	ADC trigger 7 update source Refer to the ADTG5UPDSRC[2:0] description
[7]	Reserved	Reserved, the reset value must be maintained
[6:4]	ADTG6UPDSRC	ADC trigger 6 update source Refer to the ADTG5UPDSRC[2:0] description
[3]	Reserved	Reserved, the reset value must be maintained
[2:0]	ADTG5UPDSRC	ADC trigger 5 update source These bits define the source which triggers the update of SHRTIM_ADTGEX1.ADTG5SRC[5:0] bitfield (from preload to active register transfer). It only defines the source timer. The exact conditions are defined by the SHRTIM_MCTRL.BRSTDMA[1:0] and SHRTIM_TxCTRL.UPDGAT[3:0] bit fields.

		000: Master timer 001: Timer A 010: Timer B 011: Timer C 100: Timer D 101: Timer E 110: Timer F 111: Reserved

9.4.3.35 SHRTIM ADC Post Scaler Register 1 (SHRTIM_ADCPSC1)

Address offset: 0x41C

Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved			ADC5PSC					Reserved	ADC4PSC					Reserved	ADC3PSC
			rw						rw						rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADC3PSC				Reserved	ADC2PSC					Reserved	ADC1PSC				
rw					rw						rw				

Bit Field	Name	Description
[31:29]	Reserved	Reserved, the reset value must be maintained
[28:24]	ADC5PSC	ADC trigger 5 post prescaler This bit selects the post-scaler ratio of ADC trigger 5.
[23]	Reserved	Reserved, the reset value must be maintained
[22:18]	ADC4PSC	ADC trigger 4 post prescaler This bit selects the post-scaler ratio of ADC trigger 4.
[17]	Reserved	Reserved, the reset value must be maintained
[16:12]	ADC3PSC	ADC trigger 3 post prescaler This bit selects the post-scaler ratio of ADC trigger 3.
[11]	Reserved	Reserved, the reset value must be maintained
[10:6]	ADC2PSC	ADC trigger 2 post prescaler This bit selects the post-scaler ratio of ADC trigger 2.
[5]	Reserved	Reserved, the reset value must be maintained
[4:0]	ADC1PSC	ADC trigger 1 post prescaler This bit selects the post-scaler ratio of ADC trigger 1.

9.4.3.36 SHRTIM ADC Post Scaler Register 2 (SHRTIM_ADCPSC2)

Address offset: 0x420

Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved			ADC10PSC					Reserved	ADC9PSC					Reserved	ADC8PSC

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					rw						rw				rw
ADC8PSC				Reserved	ADC7PSC					Reserved	ADC6PSC				
rw					rw						rw				

Bit Field	Name	Description
[31:29]	Reserved	Reserved, the reset value must be maintained
[28:24]	ADC10PSC	ADC trigger 10 post prescaler This bit selects the post-scaler ratio of ADC trigger 10.
[23]	Reserved	Reserved, the reset value must be maintained
[22:18]	ADC9PSC	ADC trigger 9 post prescaler This bit selects the post-scaler ratio of ADC trigger 9.
[17]	Reserved	Reserved, the reset value must be maintained
[16:12]	ADC8PSC	ADC trigger 8 post prescaler This bit selects the post-scaler ratio of ADC trigger 8.
[11]	Reserved	Reserved, the reset value must be maintained
[10:6]	ADC7PSC	ADC trigger 7 post prescaler This bit selects the post-scaler ratio of ADC trigger 7.
[5]	Reserved	Reserved, the reset value must be maintained
[4:0]	ADC6PSC	ADC trigger 6 post prescaler This bit selects the post-scaler ratio of ADC trigger 6.

9.4.3.37 SHRTIM Software Fault Register (SHRTIM_SFTFALT)

Address offset: 0x428

Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										SFTFALT6	SFTFALT5	SFTFALT4	SFTFALT3	SFTFALT2	SFTFALT1
										rt_w	rt_w	rt_w	rt_w	rt_w	rt_w

Bit Field	Name	Description
[31:6]	Reserved	Reserved, the reset value must be maintained
[5]	SFTFALT6	Soft fault 6 1: This bit is only set by software. If the fault is enabled, writing a 1 to this register will cause a software fault. 0: Writing a 0 to this bit has no effect. When the SHRTIM_OEN.TxyOEN bit of both channels corresponding to the given timer output channel is set, this bit is cleared by hardware.
[4]	SFTFALT5	Soft fault 5 1: This bit is only set by software. If the fault is enabled, writing a 1 to this

		register will cause a software fault. 0: Writing a 0 to this bit has no effect. When the SHRTIM_OEN.TxyOEN bit of both channels corresponding to the given timer output channel is set, this bit is cleared by hardware.
[3]	SFTFALT4	Soft fault 4 1: This bit is only set by software. If the fault is enabled, writing a 1 to this register will cause a software fault. 0: Writing a 0 to this bit has no effect. When the SHRTIM_OEN.TxyOEN bit of both channels corresponding to the given timer output channel is set, this bit is cleared by hardware.
[2]	SFTFALT3	Soft fault 3 1: This bit is only set by software. If the fault is enabled, writing a 1 to this register will cause a software fault. 0: Writing a 0 to this bit has no effect. When the SHRTIM_OEN.TxyOEN bit of both channels corresponding to the given timer output channel is set, this bit is cleared by hardware.
[1]	SFTFALT2	Soft fault 2 1: This bit is only set by software. If the fault is enabled, writing a 1 to this register will cause a software fault. 0: Writing a 0 to this bit has no effect. When the SHRTIM_OEN.TxyOEN bit of both channels corresponding to the given timer output channel is set, this bit is cleared by hardware.
[0]	SFTFALT1	Soft fault 1 1: This bit is only set by software. If the fault is enabled, writing a 1 to this register will cause a software fault. 0: Writing a 0 to this bit has no effect. When the SHRTIM_OEN.TxyOEN bit of both channels corresponding to the given timer output channel is set, this bit is cleared by hardware.

9.4.3.38 SHRTIM Software Delay Protection Register (SHRTIM_SFTDP)

Address offset: 0x42C

Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				SFTDP F2	SFTDP E2	SFTDP D2	SFTDP C2	SFTDP B2	SFTDP A2	SFTDP F1	SFTDP E1	SFTDP D1	SFTDP C1	SFTDP B1	SFTDP A1
				rt_w	rt_w	rt_w	rt_w	rt_w	rt_w	rt_w	rt_w	rt_w	rt_w	rt_w	rt_w

Bit Field	Name	Description
[31:12]	Reserved	Reserved, the reset value must be maintained
[11]	SFTDPF2	Software delay protection for timer F channel 2

		1: This bit is only set by software. If the delayed idle mode of timer F channel 2 is enabled, writing a 1 to this register will cause the delayed idle mode entry. 0: Writing a 0 to this bit has no effect. When the SHRTIM_OEN.TxyOEN bits of both channels corresponding to the given timer output channel are set, this bit is cleared by hardware.
[10]	SFTDPE2	Software delay protection for timer E channel 2 1: This bit is only set by software. If the delayed idle mode of timer E channel 2 is enabled, writing a 1 to this register will cause the delayed idle mode entry. 0: Writing a 0 to this bit has no effect. When the SHRTIM_OEN.TxyOEN bits of both channels corresponding to the given timer output channel are set, this bit is cleared by hardware.
[9]	SFTDPD2	Software delay protection for timer D channel 2 1: This bit is only set by software. If the delayed idle mode of timer D channel 2 is enabled, writing a 1 to this register will cause the delayed idle mode entry. 0: Writing a 0 to this bit has no effect. When the SHRTIM_OEN.TxyOEN bits of both channels corresponding to the given timer output channel are set, this bit is cleared by hardware.
[8]	SFTDPC2	Software delay protection for timer C channel 2 1: This bit is only set by software. If the delayed idle mode of timer C channel 2 is enabled, writing a 1 to this register will cause the delayed idle mode entry. 0: Writing a 0 to this bit has no effect. When the SHRTIM_OEN.TxyOEN bits of both channels corresponding to the given timer output channel are set, this bit is cleared by hardware.
[7]	SFTDPB2	Software delay protection for timer B channel 2 1: This bit is only set by software. If the delayed idle mode of timer B channel 2 is enabled, writing a 1 to this register will cause the delayed idle mode entry. 0: Writing a 0 to this bit has no effect. When the SHRTIM_OEN.TxyOEN bits of both channels corresponding to the given timer output channel are set, this bit is cleared by hardware.
[6]	SFTDPA2	Software delay protection for timer A channel 2 1: This bit is only set by software. If the delayed idle mode of timer A channel 2 is enabled, writing a 1 to this register will cause the delayed idle mode entry. 0: Writing a 0 to this bit has no effect. When the SHRTIM_OEN.TxyOEN bits of both channels corresponding to the given timer output channel are set, this bit is cleared by hardware.
[5]	SFTDPF1	Software delay protection for timer F channel 1 1: This bit is only set by software. If the delayed idle mode of timer F channel 1 is enabled, writing a 1 to this register will cause the delayed idle mode entry. 0: Writing a 0 to this bit has no effect. When the SHRTIM_OEN.TxyOEN bits of both channels corresponding to the given timer output channel are set, this bit is cleared by hardware.
[4]	SFTDPE1	Software delay protection for timer E channel 1 1: This bit is only set by software. If the delayed idle mode of timer E channel

		1 is enabled, writing a 1 to this register will cause the delayed idle mode entry. 0: Writing a 0 to this bit has no effect. When the SHRTIM_OEN.TxyOEN bits of both channels corresponding to the given timer output channel are set, this bit is cleared by hardware.
[3]	SFTDPD1	Software delay protection for timer D channel 1 1: This bit is only set by software. If the delayed idle mode of timer D channel 1 is enabled, writing a 1 to this register will cause the delayed idle mode entry. 0: Writing a 0 to this bit has no effect. When the SHRTIM_OEN.TxyOEN bits of both channels corresponding to the given timer output channel are set, this bit is cleared by hardware.
[2]	SFTDPC1	Software delay protection for timer C channel 1 1: This bit is only set by software. If the delayed idle mode of timer C channel 1 is enabled, writing a 1 to this register will cause the delayed idle mode entry. 0: Writing a 0 to this bit has no effect. When the SHRTIM_OEN.TxyOEN bits of both channels corresponding to the given timer output channel are set, this bit is cleared by hardware.
[1]	SFTDPB1	Software delay protection for timer B channel 1 1: This bit is only set by software. If the delayed idle mode of timer B channel 1 is enabled, writing a 1 to this register will cause the delayed idle mode entry. 0: Writing a 0 to this bit has no effect. When the SHRTIM_OEN.TxyOEN bits of both channels corresponding to the given timer output channel are set, this bit is cleared by hardware.
[0]	SFTDPA1	Software delay protection for timer A channel 1 1: This bit is only set by software. If the delayed idle mode of timer A channel 1 is enabled, writing a 1 to this register will cause the delayed idle mode entry. 0: Writing a 0 to this bit has no effect. When the SHRTIM_OEN.TxyOEN bits of both channels corresponding to the given timer output channel are set, this bit is cleared by hardware.

9.4.3.39 SHRTIM Fault Input Register 5 (SHRTIM_FALTIN5)

Address offset: 0x430

Reset value: 0x00000000

z	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved									FALT6CSEL			Reserved	FALT5CSEL		
										rw				rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	FALT4CSEL			Reserved	FALT3SEL			Reserved	FALT2SEL			Reserved	FALT1SEL		
		rw				rw				rw				rw	

Bit Field	Name	Description
[31:23]	Reserved	Reserved, the reset value must be maintained
[22:20]	FALT6CSEL	Fault 6 analog comparator select Refer to the FALT1CSEL[2:0] description.

[19]	Reserved	Reserved, the reset value must be maintained
[18:16]	FALT5CSEL	Fault 5 analog comparator select Refer to the FALT1CSEL[2:0] description.
[15]	Reserved	Reserved, the reset value must be maintained
[14:12]	FALT4CSEL	Fault 4 analog comparator select Refer to the FALT1CSEL[2:0] description.
[11]	Reserved	Reserved, the reset value must be maintained
[10:8]	FALT3CSEL	Fault 3 analog comparator select Refer to the FALT1CSEL[2:0] description.
[7]	Reserved	Reserved, the reset value must be maintained
[6:4]	FALT2CSEL	Fault 2 analog comparator select Refer to the FALT1CSEL[2:0] description.
[3]	Reserved	Reserved, the reset value must be maintained
[2:0]	FALT1CSEL	Fault 1 analog comparator select This bit can be locked by FALT1LCK 000: FAULT1 analog comparator input comes from comp1 out. 001: FAULT1 analog comparator input comes from comp2 out 010: FAULT1 analog comparator input comes from comp3 out 011: FAULT1 analog comparator input comes from comp4 out 100: FAULT1 analog comparator input comes from comp5 out 101: FAULT1 analog comparator input comes from comp6 out 110: FAULT1 analog comparator input comes from comp7 out Other: Reserved

9.4.3.40 SHRTIM External Event Control Register 5 (SHRTIM_EXEVCTRL5)

Address offset: 0x434

Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		EXEV10CSEL			EXEV9CSEL			EXEV8CSEL			EXEV7CSEL			EXEV6CSEL	
		rw			rw			rw			rw			rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	EXEV5CSEL			EXEV4CSEL			EXEV3CSEL			EXEV2CSEL			EXEV1CSEL		
rw	rw			rw			rw			rw			rw		

Bit Field	Name	Description
[31:30]	Reserved	Reserved, the reset value must be maintained
[29:27]	EXEV10CSEL	External event 10's analog comparator select Refer to the EXEV1CSEL description.
[26:24]	EXEV9CSEL	External event 9's analog comparator select Refer to the EXEV1CSEL description.
[23:21]	EXEV8CSEL	External event 8's analog comparator select Refer to the EXEV1CSEL description.

[20:18]	EXEV7CSEL	External event 7's analog comparator select Refer to the EXEV1CSEL description.
[17:15]	EXEV6CSEL	External event 6's analog comparator select Refer to the EXEV1CSEL description.
[14:12]	EXEV5CSEL	External event 5's analog comparator select Refer to the EXEV1CSEL description.
[11:9]	EXEV4CSEL	External event 4's analog comparator select Refer to the EXEV1CSEL description.
[8:6]	EXEV3CSEL	External event 3's analog comparator select Refer to the EXEV1CSEL description.
[5:3]	EXEV2CSEL	External event 2's analog comparator select Refer to the EXEV1CSEL description.
[2:0]	EXEV1CSEL	External event 1's analog comparator select 000: External event 1's analog comparator input comes from comp1 out 001: External event 1's analog comparator input comes from comp2 out 010: External event 1's analog comparator input comes from comp3 out 011: External event 1's analog comparator input comes from comp4 out 100: External event 1's analog comparator input comes from comp5 out 101: External event 1's analog comparator input comes from comp6 out 110: External event 1's analog comparator input comes from comp7 out Other: Reserved

10 Advanced-control Timers（ATIM1/ ATIM2/ ATIM3）

10.1 ATIMx（x=1-3）Introduction

The advanced control timers (ATIM1/ ATIM2/ ATIM3) are mainly used in the following scenarios: counting the input signals, measuring the pulse width of the input signals and generating the output waveforms, etc.

Advanced-control timers have complementary output function with dead-time insertion and break function, which are suitable for motor control.

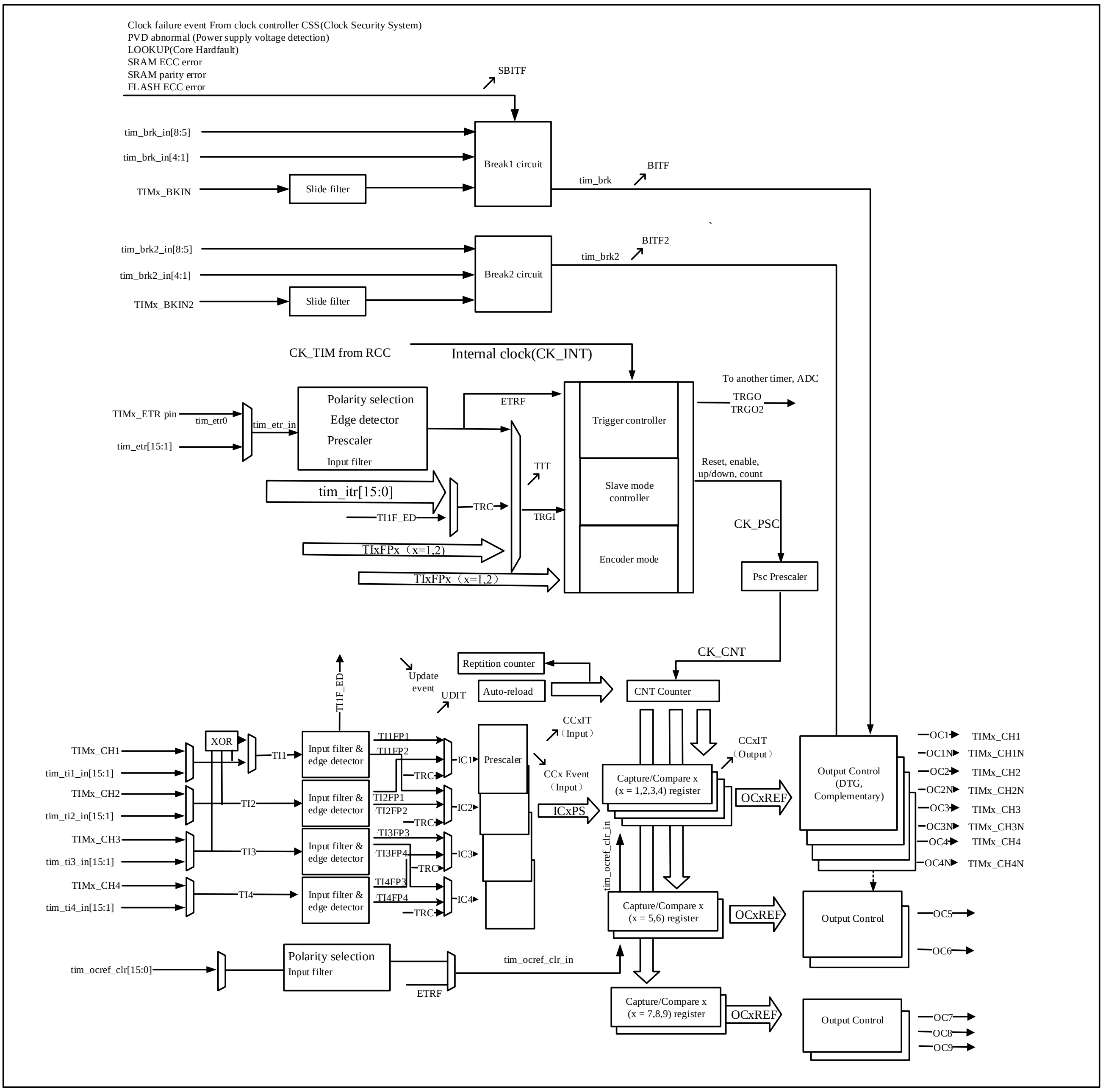
10.2 ATIMx（x=1-3）Main Features

- 16-bit auto-reload counters. (It can realize up-counting, down-counting, up/down counting)
- 16-bit programmable prescaler. (The prescaler factor can be configured with any value between 1 and 65536)
- Programmable repetition counter
- Up to 9 channels
- 4 capture/compare channels, the operating modes are PWM output, ouput compare, one-pulse mode output, input capture
- 2 break input signals with digital filter

- The events that generate the interrupt/DMA are as follows:
  - Update event
  - Trigger event
  - Input capture
  - Output compare
  - Break signal input
- Complementary outputs with programmable dead-time
  - Channel 1,2,3,4 support this feature
- Timer can be controlled by external signal
- Timers can be linked together internally for timer synchronization or chaining
- Incremental (quadrature) encoder interface: used for tracking motion and resolving rotation direction and position
- Hall sensor interface: used to do three-phase motor control
- Trigger input for external clock or cycle-by-cycle current management

10.3 ATIMx（x=1-3）Block Diagram

Figure 10-1 Block Diagram of ATIMx



10.4 ATIMx（x=1-3）Pins and Internal Signals

The following table describes the input and output pins and internal signals of ATIMx.

Table 10-1 ATIMx Input/Output Pins

Pin Name	Type	Description
TIMx_CH1	Input/Output	Each channel can be used for capture, compare or PWM.

TIMx_CH2 TIMx_CH3 TIMx_CH4		TIM_CH1 and TIM_CH2 can also be used as external clock (below 1/4 of the tim_ker_ck clock), external trigger and quadrature encoder inputs. TIM_CH1, TIM_CH2 and TIM_CH3 can be used to interface with digital hall effect sensors.
TIM_CH1N TIM_CH2N TIM_CH3N TIM_CH4N	Output	Complementary outputs with programmable dead-time insertion
TIMx_ETR	Input	External trigger input. This input can be used as external trigger or as external clock source. This input can receive a clock with a frequency higher than the tim_ker_ck if the tim_etr_in prescaler is used.
TIMx_BKIN TIMx_BKIN2	Input/Output	Break and Break2 inputs. Both inputs can be configured in bidirectional mode.

Table 10-2 ATIMx Internal Input/Output Signals

Internal Signals	Type	Description
tim_ti1_in[15:0] tim_ti2_in[15:0] tim_ti3_in[15:0] tim_ti4_in[15:0]	Input	Channels 1/2/3/4 input signals. tim_ti1_in[15:0] and tim_ti2_in[15:0] inputs can be used for capture or as an external clock (lower than 1/4 of the system clock frequency) and for quadrature encoder signals.
tim_etr[15:0]	Input	External trigger channel input signals. These inputs can be used as triggers, external clocks, or for hardware per-cycle pulse width control. If a prescaler is used, the input signal tim_etr can be a signal with a frequency higher than the system clock frequency.
tim_itr[15:0]	Input	Internal trigger input signals. These inputs can be used in mode controllers or as input clocks (lower than 1/4 of the system clock frequency).
tim_trgo/tim_trgo2	Output	Internal trigger signal output. These trigger signals can be used by other timers and/or other peripheral devices.

10.4.1 Interconnect of the tim_ti1/ tim_ti2/ tim_ti3/ tim_ti4 Input

Table 10-3 Interconnect to tim_ti1 Input Multiplexer

tim_ti1 inputs	Sources		
	ATIM1	ATIM2	ATIM3
tim_ti1_in0	ATIM1_CH1	ATIM2_CH1	ATIM3_CH1
tim_ti1_in1	COMP1_OUT	COMP1_OUT	COMP1_OUT
tim_ti1_in2	COMP2_OUT	COMP2_OUT	COMP2_OUT
tim_ti1_in3	COMP3_OUT	COMP3_OUT	COMP3_OUT
tim_ti1_in4	COMP4_OUT	COMP4_OUT	COMP4_OUT
tim_ti1_in[15:5]	Reserved		

Table 10-4 Interconnect to tim_ti2 Input Multiplexer

tim_ti2 inputs	Sources		
	ATIM1	ATIM2	ATIM3
tim_ti2_in0	ATIM1_CH2	ATIM2_CH2	ATIM3_CH2
tim_ti2_in1	GTIM7_CC1	GTIM7_CC2	GTIM7_CC3
tim_ti2_in[15:2]	Reserved		

Table 10-5 Interconnect to tim_ti3 Input Multiplexer

tim_ti3 inputs	Sources		
	ATIM1	ATIM2	ATIM3
tim_ti3_in0	ATIM1_CH3	ATIM2_CH3	ATIM3_CH3
tim_ti3_in[15:1]	Reserved		

Table 10-6 Interconnect to tim_ti4 Input Multiplexer

tim_ti4 inputs	Sources		
	ATIM1	ATIM2	ATIM3

tim_ti4_in0	ATIM1_CH4	ATIM2_CH4	ATIM3_CH4
tim_ti4_in[15:1]	Reserved		

10.4.2 Interconnect of the tim_itr Input

Table 10-7 Interconnect to tim_itr Input Multiplexer

ATIMx	ATIM1	ATIM2	ATIM3
tim_itr0	Reserved	ATIM1_TRGO	ATIM1_TRGO
tim_itr1	GTIM1_TRGO	GTIM1_TRGO	GTIM1_TRGO
tim_itr2	GTIM2_TRGO	GTIM2_TRGO	GTIM2_TRGO
tim_itr3	GTIM3_TRGO	GTIM3_TRGO	GTIM3_TRGO
tim_itr4	GTIM4_TRGO	GTIM4_TRGO	GTIM4_TRGO
tim_itr5	ATIM2_TRGO	Reserved	ATIM2_TRGO
tim_itr6	GTIM8_TRGO	GTIM8_TRGO	GTIM8_TRGO
tim_itr7	GTIM9_TRGO	GTIM9_TRGO	GTIM9_TRGO
tim_itr8	GTIM10_TRGO	GTIM10_TRGO	GTIM10_TRGO
tim_itr9	ATIM3_TRGO	ATIM3_TRGO	Reserved
tim_itr10	SHRTIM1_OUT_SYNC2	SHRTIM1_OUT_SYNC2	SHRTIM1_OUT_SYNC2
tim_itr11	Reserve	Reserve	Reserve
tim_itr12	GTIM5_TRGO	GTIM5_TRGO	GTIM5_TRGO
tim_itr13	GTIM6_TRGO	GTIM6_TRGO	GTIM6_TRGO
tim_itr14	GTIM7_TRGO	GTIM7_TRGO	GTIM7_TRGO
tim_itr15	Reserve	Reserve	Reserve

10.4.3 Interconnect of the tim_etr Input

Table 10-8 Interconnect to tim_etr Input Multiplexer

ATIMx	ATIM1	ATIM2	ATIM3
tim_etr0	ATIM1_ETR	ATIM2_ETR	ATIM3_ETR
tim_etr1	COMP1_OUT	COMP1_OUT	COMP1_OUT
tim_etr2	COMP2_OUT	COMP2_OUT	COMP2_OUT
tim_etr3	COMP3_OUT	COMP3_OUT	COMP3_OUT
tim_etr4	COMP4_OUT	COMP4_OUT	COMP4_OUT
tim_etr5	COMP5_OUT	COMP5_OUT	COMP5_OUT
tim_etr6	COMP6_OUT	COMP6_OUT	COMP6_OUT
tim_etr7	COMP7_OUT	COMP7_OUT	COMP7_OUT
tim_etr8	ADC1_AWD1	ADC2_AWD1	ADC3_AWD1
tim_etr9	ADC1_AWD2	ADC2_AWD2	ADC3_AWD2
tim_etr10	ADC1_AWD3	ADC2_AWD3	ADC3_AWD3
tim_etr11	ADC4_AWD1	ADC3_AWD1	ADC1_AWD1
tim_etr12	ADC4_AWD2	ADC3_AWD2	ADC1_AWD2
tim_etr13	ADC4_AWD3	ADC3_AWD3	ADC1_AWD3
tim_etr_in[15:14]	Reserved		

10.4.4 Interconnect of the tim_brk Input

Table 10-9 ATIMx Break Interconnect

tim_brk Input	ATIM1	ATIM2	ATIM3
TIM_BKIN	ATIM1_BKIN pin	ATIM2_BKIN pin	ATIM3_BKIN pin
tim_brk_in1	COMP1_OUT	COMP1_OUT	COMP1_OUT

tim_brk_in2	COMP2_OUT	COMP2_OUT	COMP2_OUT
tim_brk_in3	COMP3_OUT	COMP3_OUT	COMP3_OUT
tim_brk_in4	COMP4_OUT	COMP4_OUT	COMP4_OUT
tim_brk_in5	COMP5_OUT	COMP5_OUT	COMP5_OUT
tim_brk_in6	COMP6_OUT	COMP6_OUT	COMP6_OUT
tim_brk_in7	COMP7_OUT	COMP7_OUT	COMP7_OUT
tim_brk_in8	Reserved		

10.4.5 Interconnect of the tim_brk2 Input

Table 10-10 ATIMx Break2 Interconnect

tim_brk2 Input	**ATIM1**	**ATIM2**	**ATIM3**
TIM_BKIN2	ATIM1_BKIN2 pin	ATIM2_BKIN2 pin	ATIM3_BKIN2 pin
tim_brk2_in1	COMP1_OUT	COMP1_OUT	COMP1_OUT
tim_brk2_in2	COMP2_OUT	COMP2_OUT	COMP2_OUT
tim_brk2_in3	COMP3_OUT	COMP3_OUT	COMP3_OUT
tim_brk2_in4	COMP4_OUT	COMP4_OUT	COMP4_OUT
tim_brk2_in5	COMP5_OUT	COMP5_OUT	COMP5_OUT
tim_brk2_in6	COMP6_OUT	COMP6_OUT	COMP6_OUT
tim_brk2_in7	COMP7_OUT	COMP7_OUT	COMP7_OUT
tim_brk2_in8	Reserved		

10.4.6 Interconnect of the tim_ocref_clr Input

Table 10-11 Interconnect to the tim_ocref_clr Input Multiplexer

ATIMx OCREF clear signal	**ATIMx OCREF Clear Signals Assignment**		
	ATIM1	**ATIM2**	**ATIM3**

tim_ocref_clr0	COMP1_OUT	COMP1_OUT	COMP1_OUT
tim_ocref_clr1	COMP2_OUT	COMP2_OUT	COMP2_OUT
tim_ocref_clr2	COMP3_OUT	COMP3_OUT	COMP3_OUT
tim_ocref_clr3	COMP4_OUT	COMP4_OUT	COMP4_OUT
tim_ocref_clr4	COMP5_OUT	COMP5_OUT	COMP5_OUT
tim_ocref_clr5	COMP6_OUT	COMP6_OUT	COMP6_OUT
tim_ocref_clr6	COMP7_OUT	COMP7_OUT	COMP7_OUT
tim_ocref_in[15:7]	Reserved		

10.5 ATIMx（x=1-3）Function Description

10.5.1 Time-base Unit

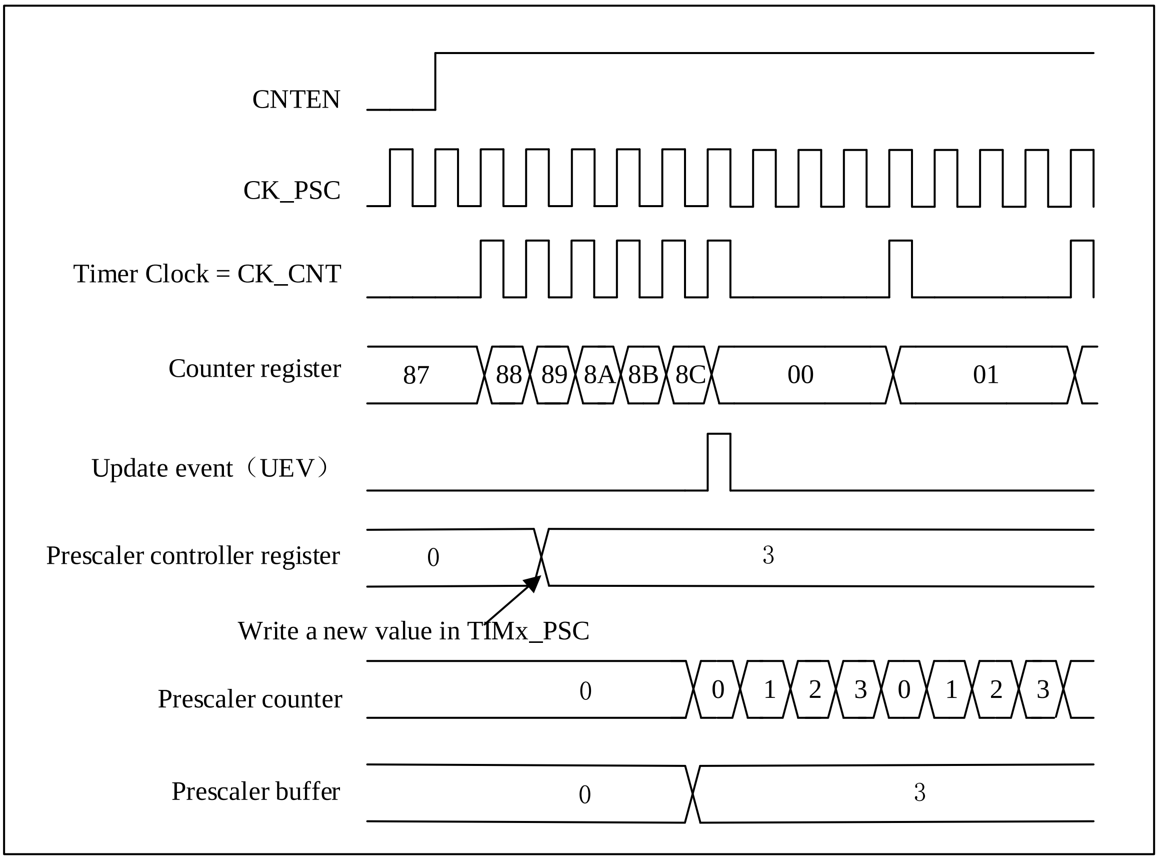
The advanced-control timer's time-base unit mainly includes: prescaler, counter, auto-reload and repetition counter. When the time base unit is operating, the software can read and write the corresponding registers (TIMx_PSC, TIMx_CNT, TIMx_AR and TIMx_REPCNT) at any time.

Depending on the setting of the auto-reload preload enable bit (TIMx_CTRL1.ARPEN), the value of the preload register is transferred to the shadow register immediately or at each update event UEV. When TIMx_CTRL1.UPDIS=0, a counter overflow/underflow or software setting TIMx_EVTGEN.UDGN will generate an update event. The counter CK_CNT is valid only when the TIMx_CTRL1.CNTEN bit is set. The counter starts counting one clock cycle after the TIMx_CTRL1.CNTEN bit is set.

10.5.1.1 Prescaler Description

The TIMx_PSC register consists of a 16-bit counter that can be used to divide the counter clock frequency by any factor between 1 and 65536. It can be changed on the fly as this register is buffered. The new prescaler value is only taken into account at the next update event.

Figure 10-2 Counter Timing Diagram with Prescaler Division Change from 1 to 4



10.5.2 Counter Mode

10.5.2.1 Up-counting mode

In up-counting mode, the counter will count from 0 to the value of the register TIMx_AR, then it restarts from 0 and a counter overflow event is generated.

If the TIMx_CTRL1.UPRS bit (select update request) and the TIMx_EVTGEN.UDGN bit are set, an update event (UEV) will be generated, but the TIMx_STS.UDITF will not be set by hardware, therefore, no update interrupts or update DMA requests are generated.This is to avoid generating an update interrupt when clearing the counter.

Depending on the TIMx_CTRL1.UPRS, when an update event occurs, all registers are updated and the TIMx_STS.UDITF is set:

- The repetition counter reloads the contents of the TIMx_REPCNT
- The auto-reload shadow registers is updated with preload value(TIMx_AR), when TIMx_CTRL1.ARPEN = 1
- The prescaler shadow register is reloaded with the preload value(TIMx_PSC).

To avoid updating the shadow registers when new values are written to the preload registers, you can disable the update event by setting TIMx_CTRL1.UPDIS=1.

When an update event is generated, the counter will still be cleared and the prescaler counter will also be set to 0 (but the prescaler value will remain unchanged).

The figure below shows some examples of the counter behavior for different prescaler factors in the up-counting mode.

Figure 10-3 Timing Diagram of Up-Counting with Internal Clock Divider Factor = 2/N

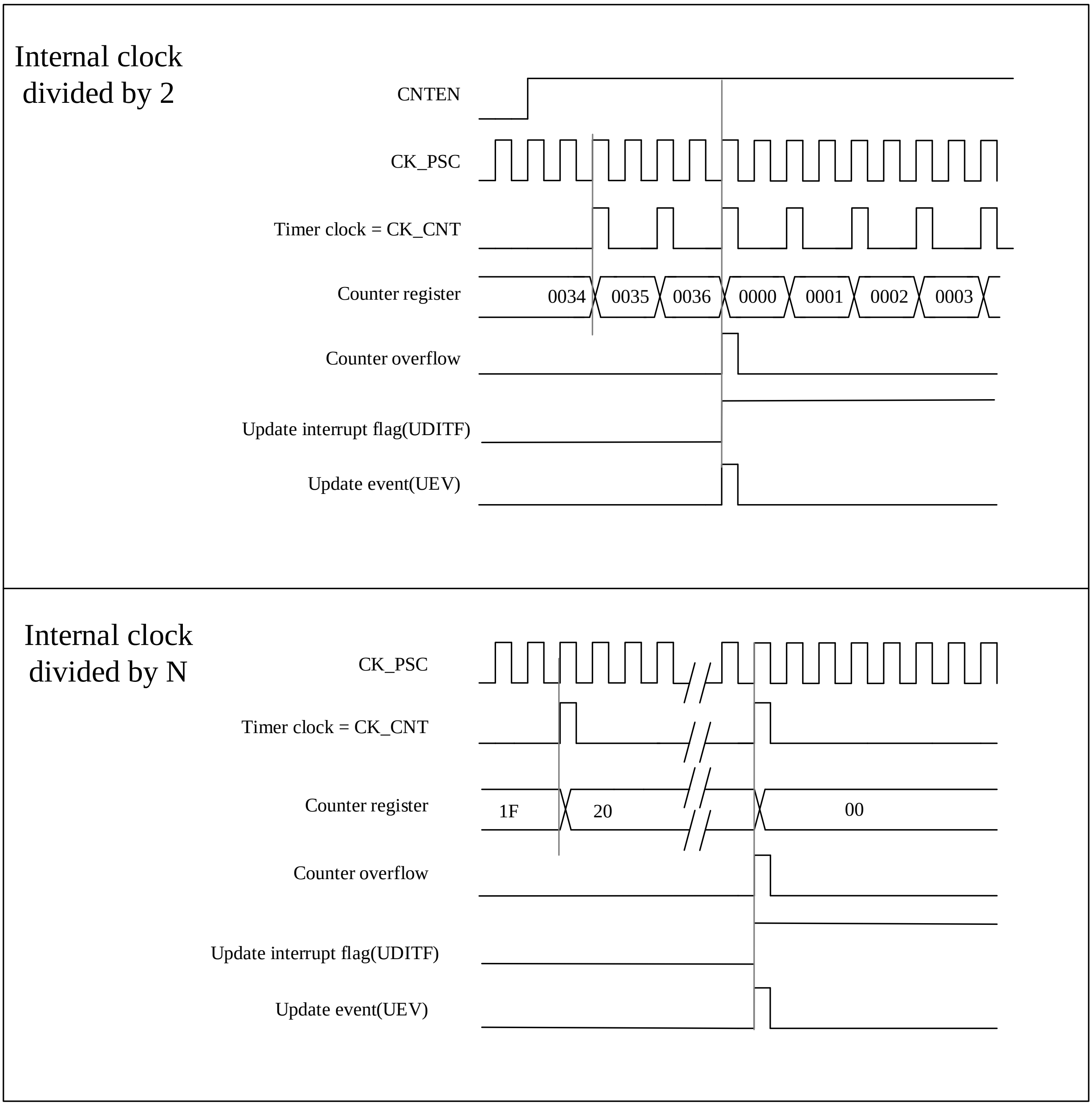
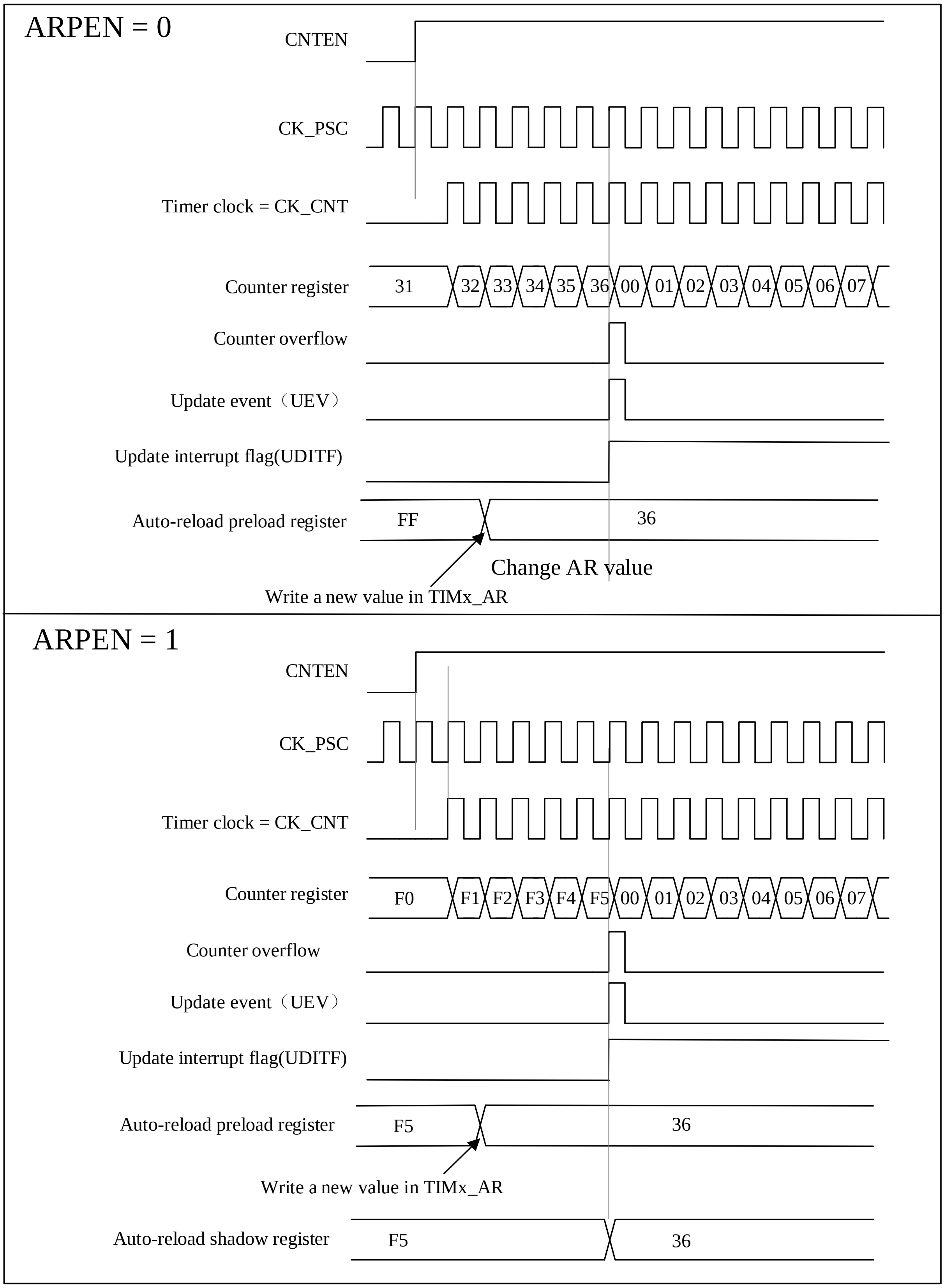


Figure 10-4 Timing Diagram of The Up-Counting with Update Event When ARPEN=0/1



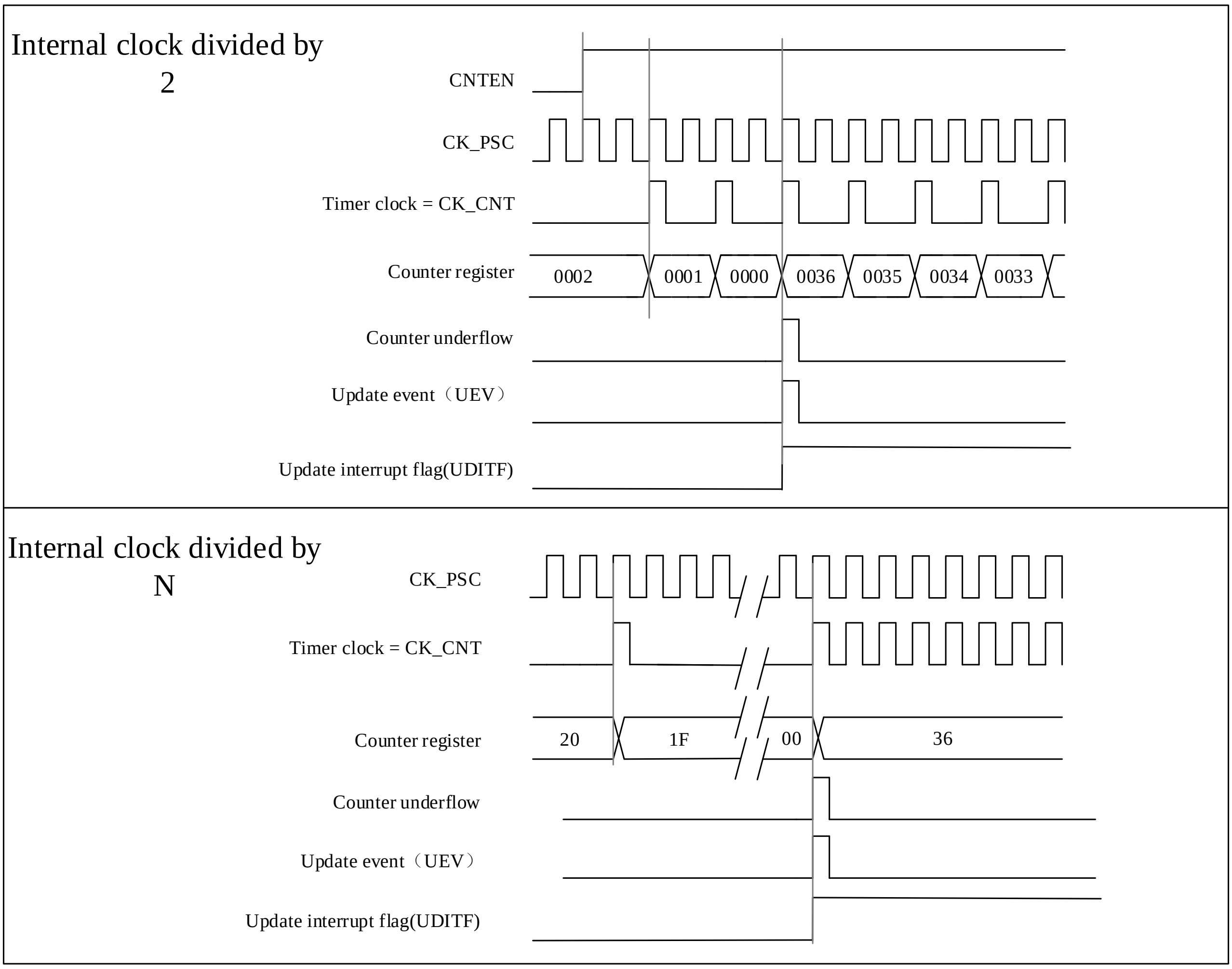
10.5.2.2 Down-counting mode

In down-counting mode, the counter will count from the value of the register TIMx_AR down to 0, then restart from the auto-reload value and generate a counter underflow event.

The process of configuring update events and updating registers in down-counting mode is the same as in up-counting mode, refer to Section 10.5.2.1.

The figure below shows some examples of the counter behavior and the update flags for different prescaler factors in the down-counting mode.

Figure 10-5 Timing Diagram of The Down-Counting with Internal Clock Divided Factor = 2/N



10.5.2.3 Center-aligned mode

10.5.2.3.1 Center-aligned symmetric mode

In center-aligned symmetric mode, the counter counts from 0 to the value (TIMx_AR) – 1, a counter overflow event is generated. It then counts from the auto-reload value (TIMx_AR) down to 1 and generates a counter underflow event. Then the counter resets to 0 and starts counting up again.

In this mode, the TIMx_CTRL1.DIR direction bits cannot be written and the count direction is updated and specified by hardware. Center-aligned mode is active when the TIMx_CTRL1. CAMSEL bit is not equal to "00".

An update event can be generated at each counter overflows and at each counter underflows. Alternatively, an update event can also be generated by setting the TIMx_EVTGEN. UDGN bit (either by software or using a slave mode controller). In this case, the counter restarts from 0, and the prescaler counter also restarts from 0.

Note: if an update is generated due to a counter overflow, the auto-reload value will be updated before the counter is reload.

Figure 10-6 Timing Diagram of the Center-Aligned with Internal Clock Divided Factor =2/N

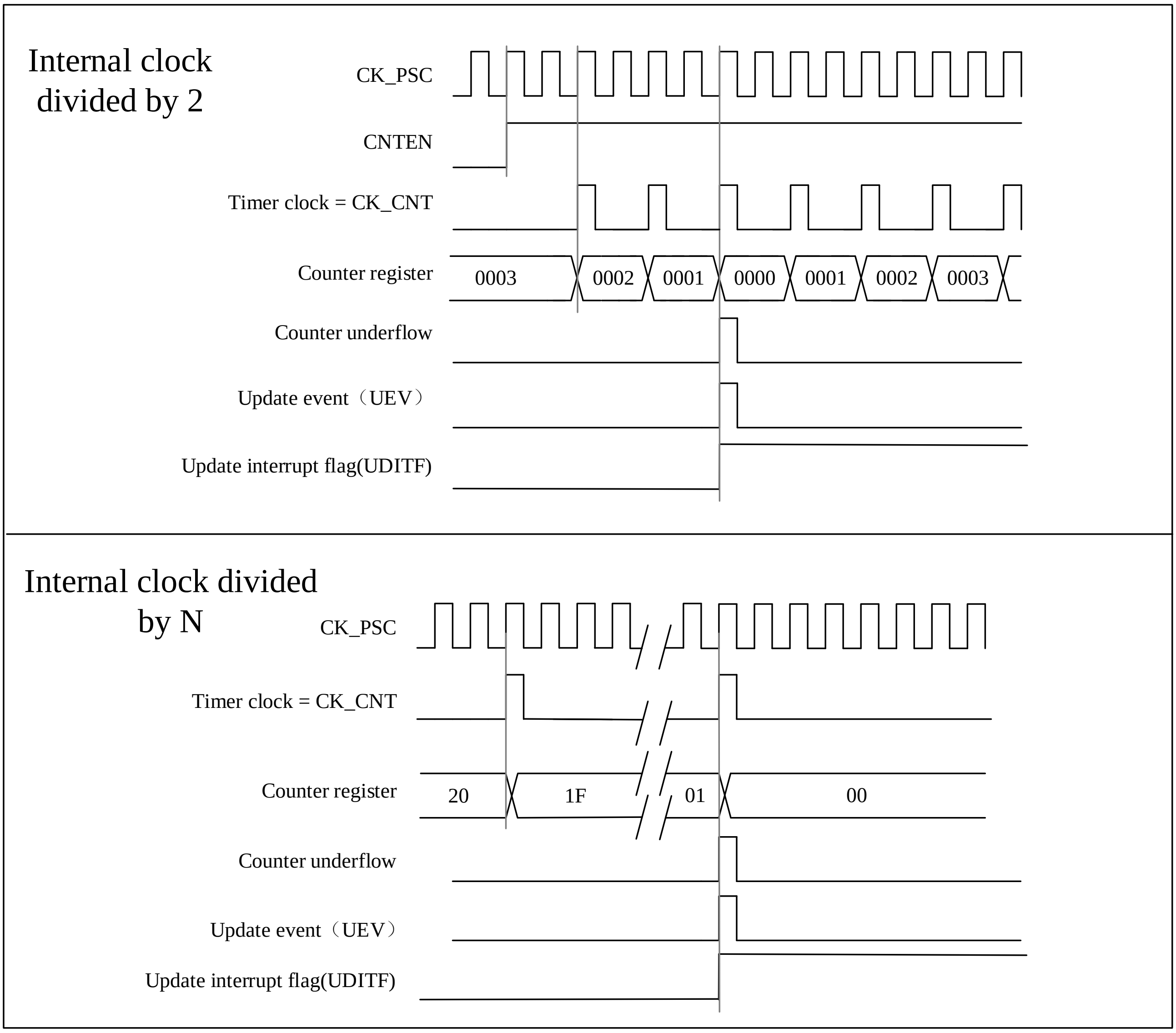
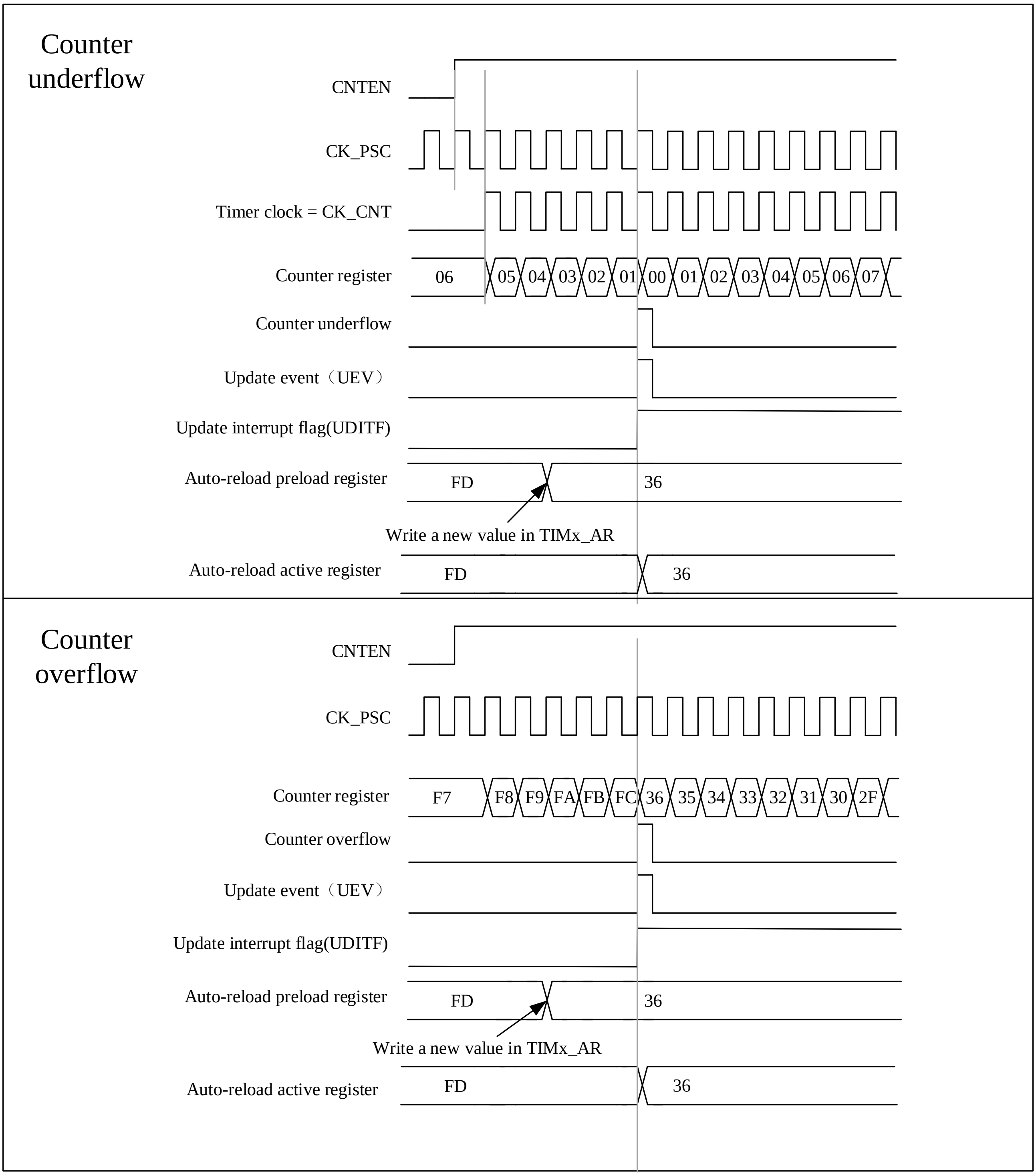


Figure 10-7 A Center-Aligned Sequence Diagram That Includes Counter Overflows and Underflows (ARPEN = 1)



10.5.2.3.2 Center-aligned asymmetric mode

In center-aligned asymmetric mode(TIMx_CTRL1. ASYMMETRIC is 1 and TIMx_CTRL1.CAMSEL[1:0] is non-zero), the counter counts from 0 to the auto-reload value (TIMx_AR) – 1 and generates a counter overflow event, then counts from the auto-reload value down to 1 and generates a counter underflow event and then restarts counting from 0.

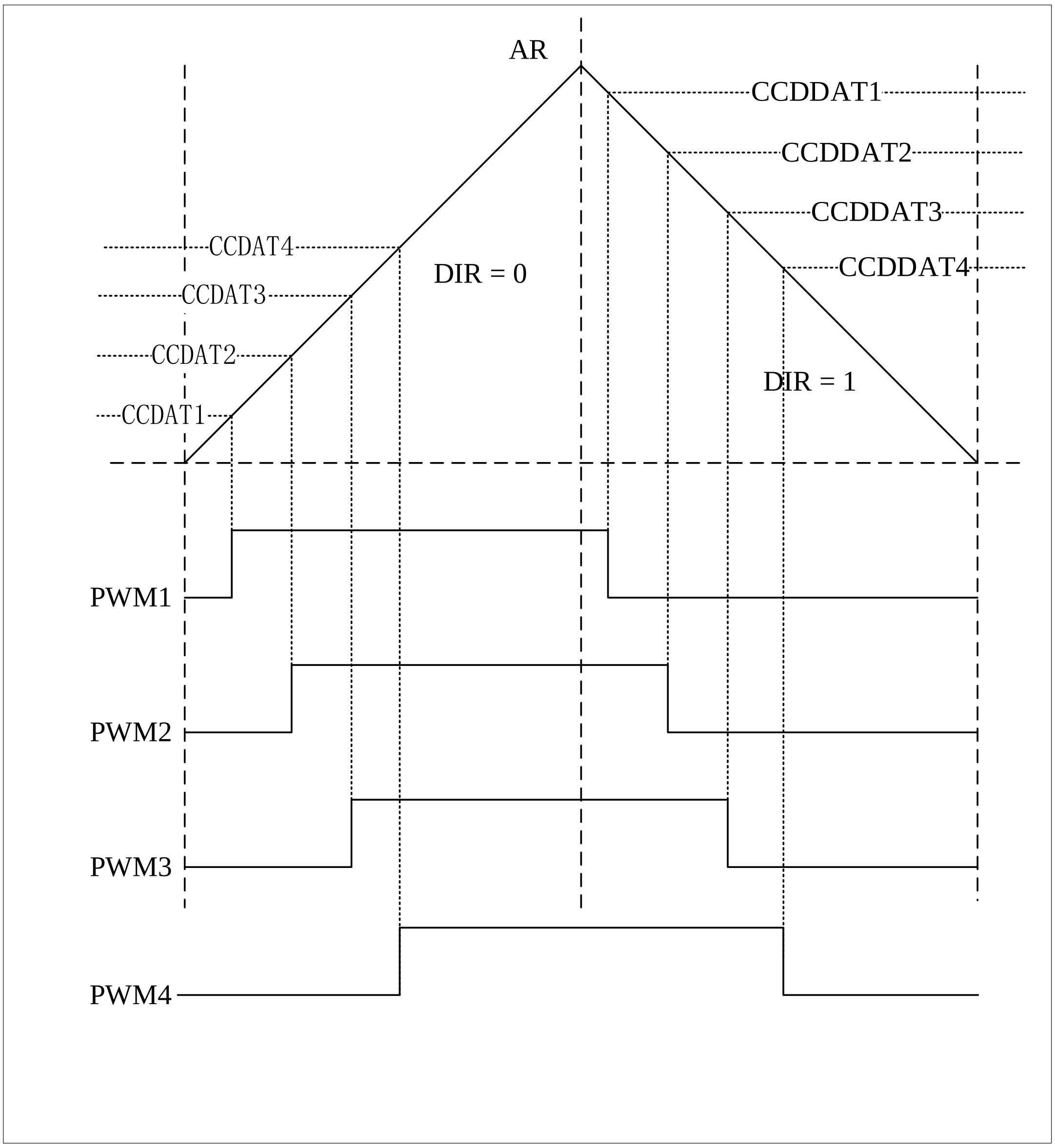
The TIMx_CTRL1.DIR cannot be written in this mode. It is updated by hardware and indicates the current direction of the counter.

When the channel is not 1,2,3,4, the comparison value are compared with CCDATx. When the dead time generator is turned on, note that when DIR = 0, the dead time insertion point is at which the counter value is equal to CCDATx(x=1,2,3,4), and when DIR = 1, the dead time insertion point is at which the counter value is equal to CCDDATx(x=1,2,3,4).

An update event can be generated each time the counter overflows and each time the counter underflows. Alternatively, an update event can also be generated by setting the TIMx_EVTGEN. UDGN bit (either by software or using a slave mode controller). In this case, the counter restarts from 0, and the prescaler counter also restarts from 0.

Note: if an update is generated due to a counter overflow, the auto-reload value will be updated before the counter is reloaded.

Figure 10-8 The Output Waveform Corresponding to the Asymmetric Mode

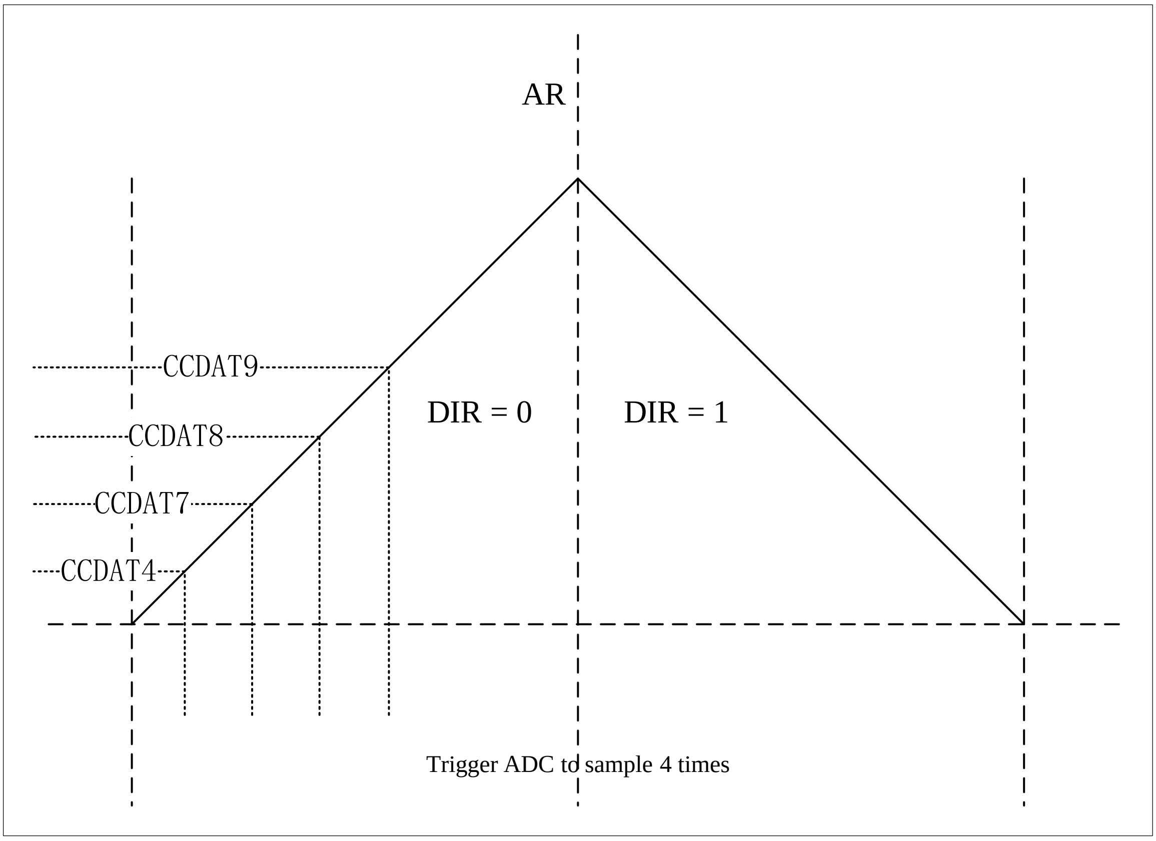


Since the triggering function of CC7/CC8/CC9 three channels is added, and the triggering function of CC4 has been modified, Here is the description of the CC4/CC7/CC8/CC9 channels triggering the ADC.

When the timer is operating in center-aligned asymmetric mode, each channel(CC4/CC7/CC8/CC9) can individually trigger the ADC only when MMSEL3 = 1.

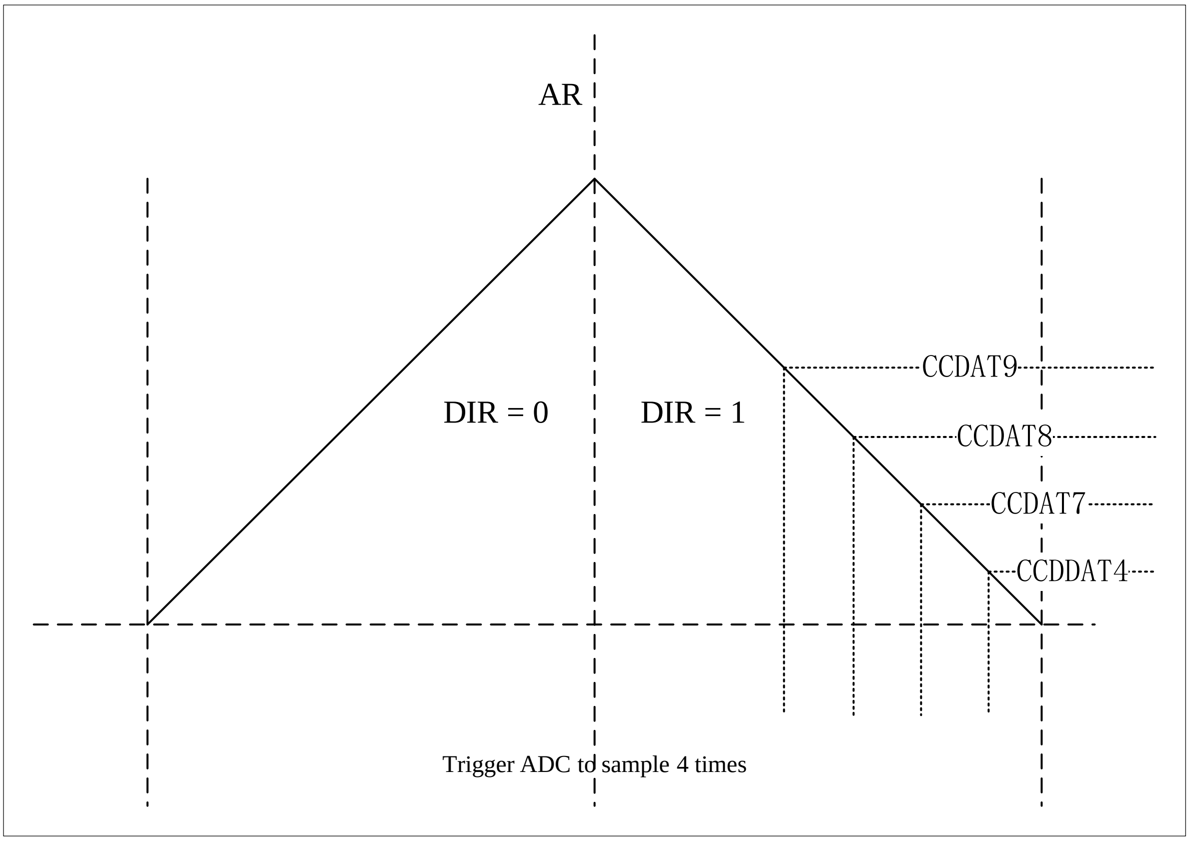
If TIMx_CTRL1.CMODE[1:0]=00, the CCDATx.CCDAT (x=4,7,8,9) value will only trigger ADC when DIR =0.

Figure 10-9 CCDATx(x=4,7,8,9) Triggers ADC When DIR = 0



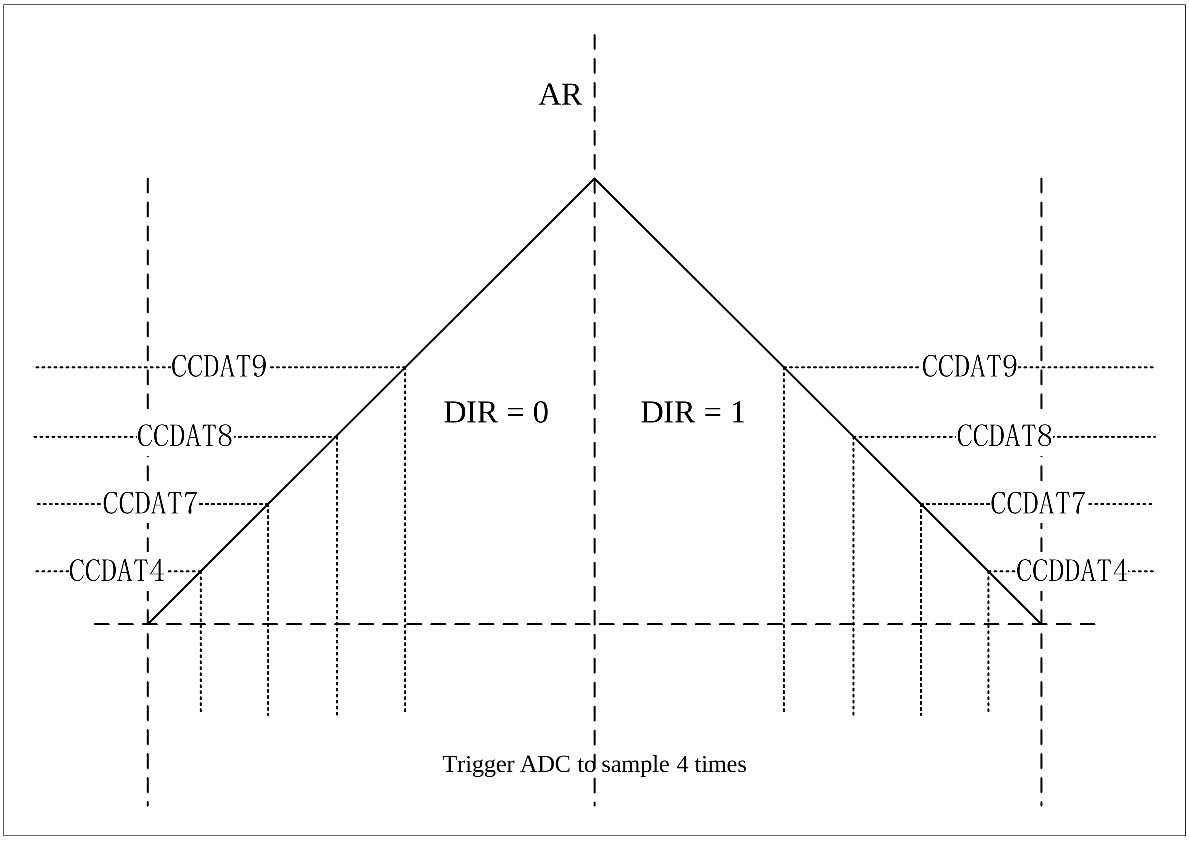
If TIMx_CTRL1.CMODE [1:0]=01, CCDATx.CCDDAT (x=4) and CCDATx.CCDAT (x=7,8,9) value will only trigger ADC when DIR =1.

Figure 10-10 CCDATx(x=4,7,8,9) Triggers ADC When DIR = 1



If TIMx_CTRL1.CMODE[1:0]=1x, the CCDATx.CCDAT and CCDATx.CCDDAT (x=4,7,8,9) value will trigger ADC when DIR =0 or DIR=1.

Figure 10-11 CCDATx(x=4,7,8,9) Triggers ADC When DIR = 0 or DIR = 1



In the above figure, channel 4 up counting to CCDAT4 or down counting to CCDDAT4, channel 7/8/9 up counting or down counting to CCDAT7/8/9, trigger valid.

10.5.3 Repetition Counter

The basic unit of Section 10.5.1 describes the conditions for generating an update event (UEV). An update event (UEV) is actually only generated when the repetition counter reaches zero, which is valuable for generating PWM signals.

This means that data are transferred from the preload registers to the shadow registers every N+1 counter overflow or underflow, where N is the value in the TIMx_REPCNT.

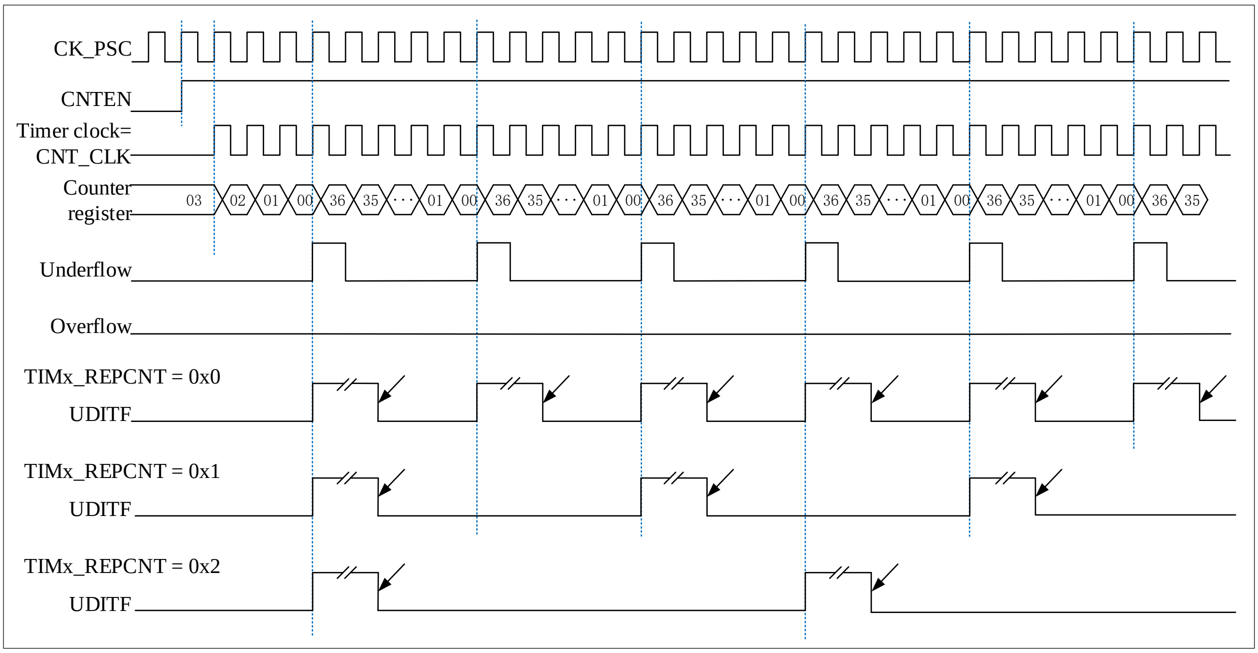
The repetition counter is decremented:

- In the up-counting mode, each time the counter reaches the maximum value, an overflow occurs.
- In down-counting mode, each time the counter decrements to the minimum value, an underflow occurs.
- In center-aligned mode, each time the counter overflows or underflows.

Its repetition rate is defined by the value of the TIMx_REPCNT register.

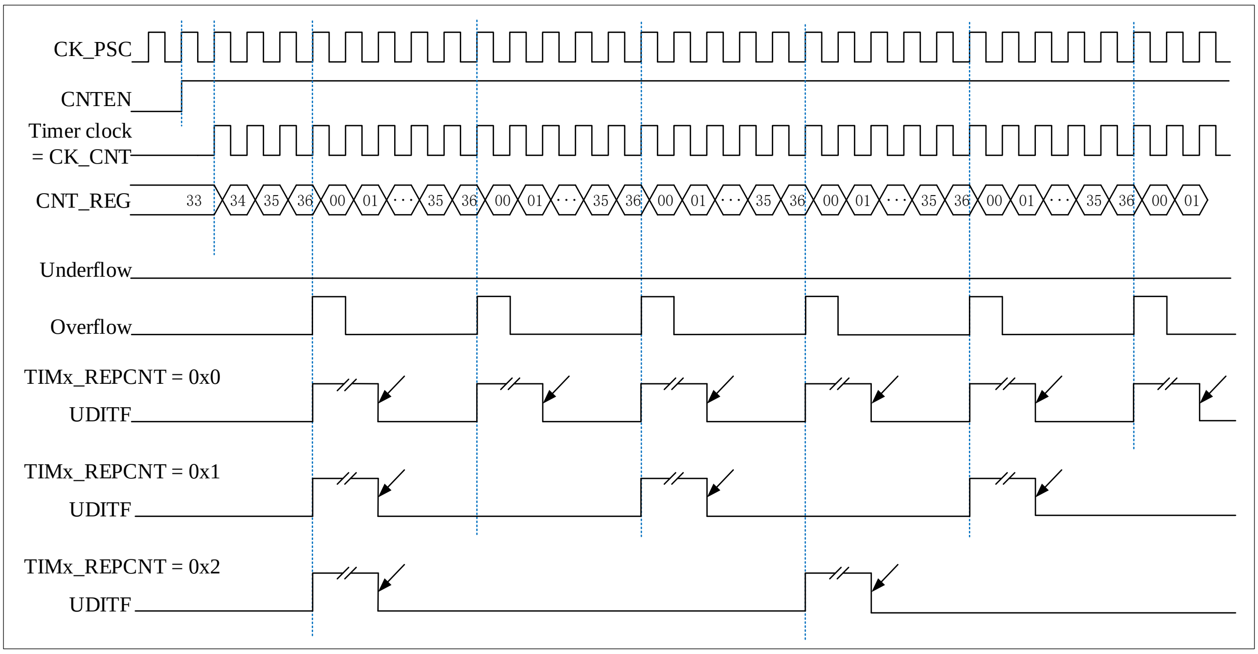
Repetition counters feature automatic reloading. The update event (generated by setting TIMx_EVTGEN.UDGN or hardware through slave mode controller) occurs immediately, regardless of the value of the repeatition counter.

Figure 10-12 Repeat Count Sequence Diagram in Down-Counting Mode



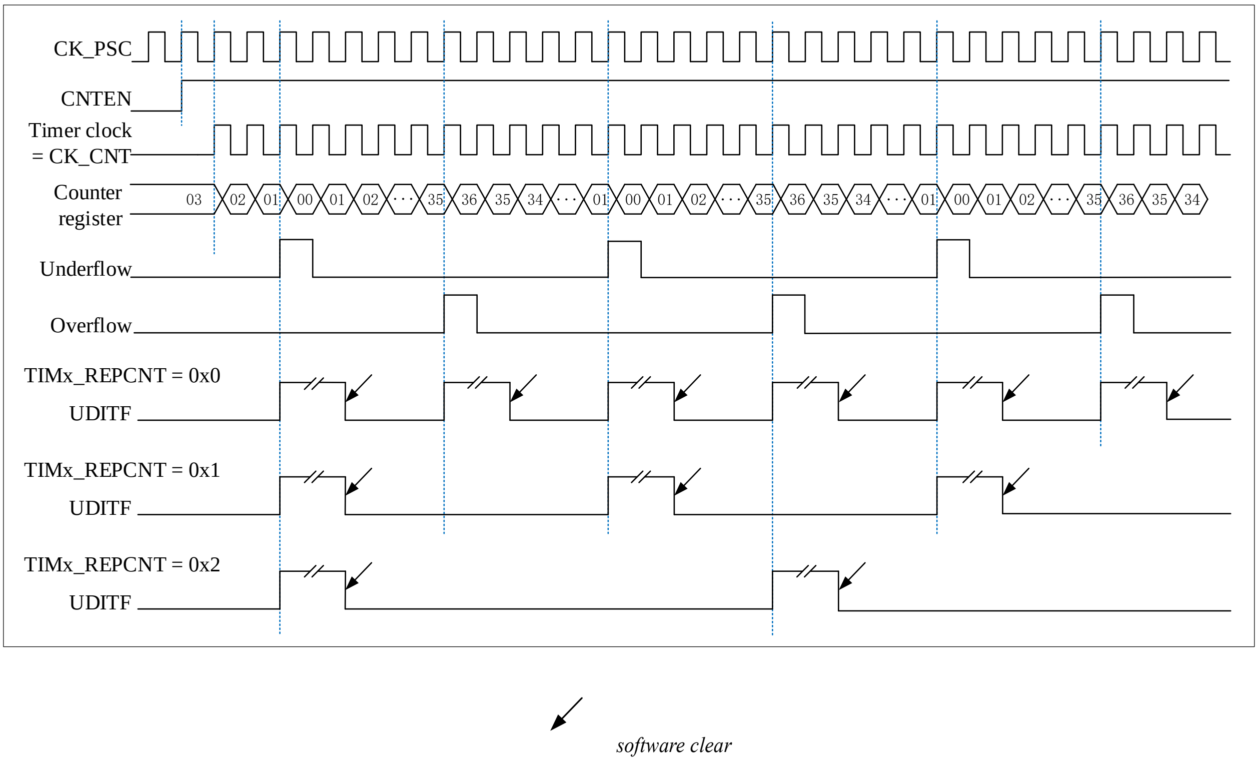
software clear

Figure 10-13 Repeat Count Sequence Diagram in Up-Counting Mode



software clear

Figure 10-14 Repeat Count Sequence Diagram in Center-Aligned Mode



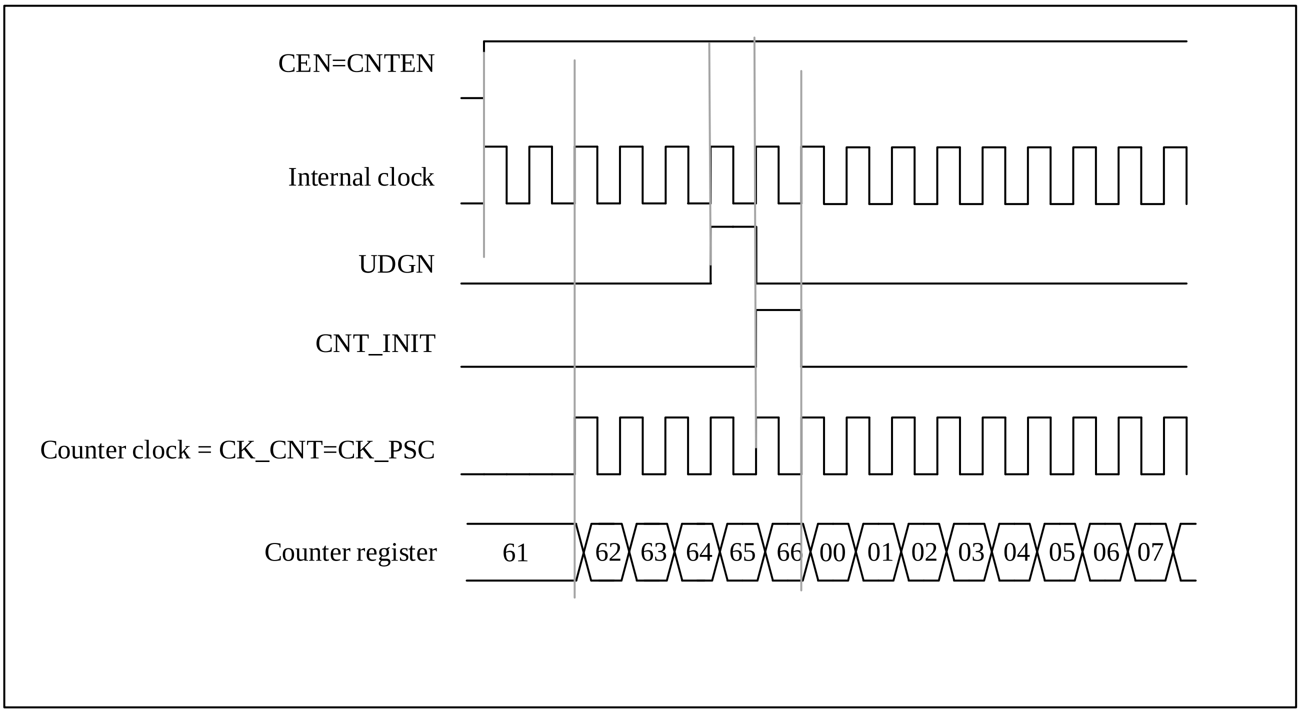
10.5.4 Clock Selection

- The internal clock of Advanced-control timers : CK_INT
- Two kinds of external clock mode :
  - External input pin
  - External trigger input ETR
- Internal trigger input (ITRx): one timer is used as a prescaler for another timer.

10.5.4.1 Internal clock source (CK_INT)

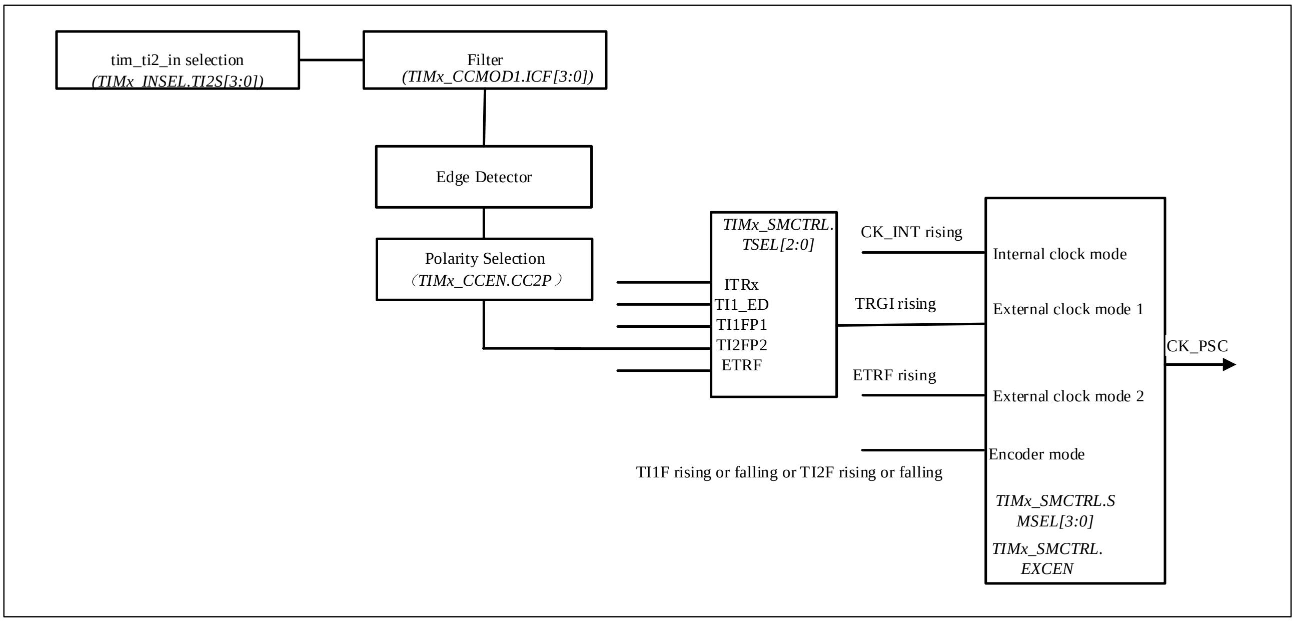
When the TIMx_SMCTRL.SMSEL is equal to “000”, the slave mode controller is disabled. The three control bits (TIMx_CTRL1.CNTEN、TIMx_CTRL1. DIR、TIMx_EVTGEN. UDGN) can only be changed by software (except TIMx_EVTGEN. UDGN, which remains cleared automatically). It is provided that the TIMx_CTRL1.CNTEN bit is written as '1' by software, the clock source of the prescaler is provided by the internal clock CK_INT.

Figure 10-15 Control Circuit in Normal Mode, Internal Clock Divided by 1



10.5.4.2 External clock source mode 1

Figure 10-16 TI2 TI2 External Clock Connection Example



This mode is selected by configuring TIMx_SMCTRL.SMSEL=111. The counter can be configured to count on the rising or falling edge of the clock at the selected input.

For example, to configure up-counting mode to count on the rising edge of the clock at the T12 input, the configuration steps are as follows:

- Configure TIMx_CCMOD1.CC2SEL equal to ‘01’, CC2 channel is configured as input, IC2 is mapped to TI2

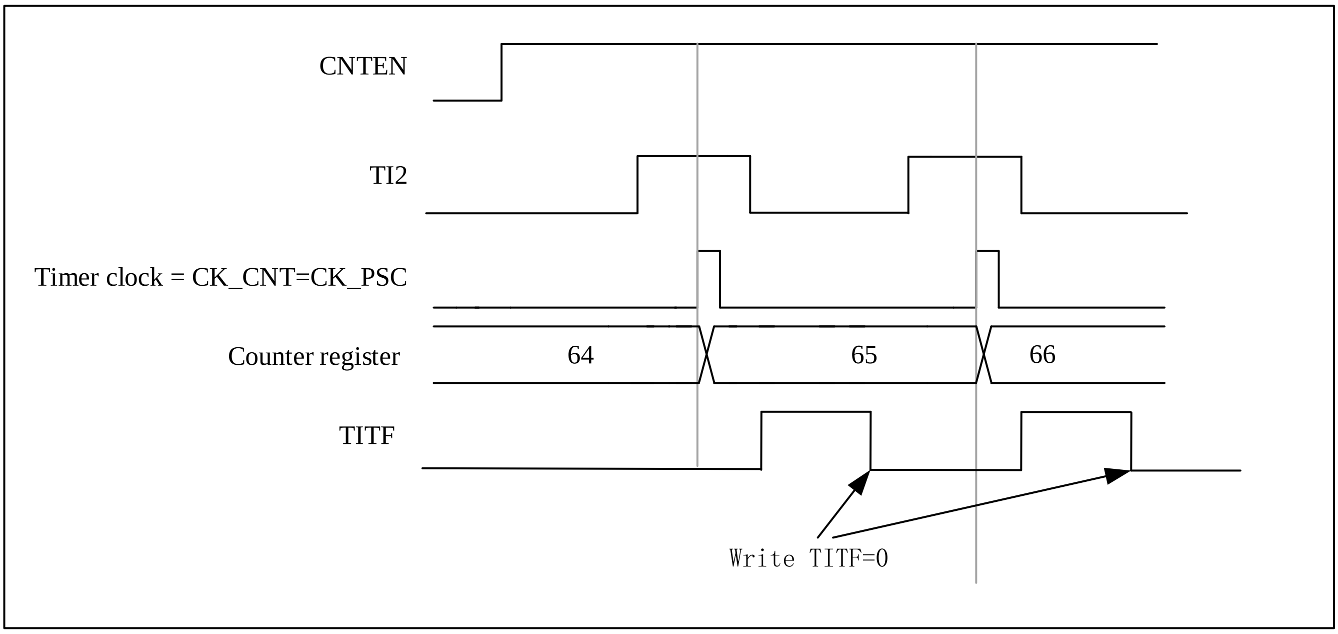
- Configure TIMx_CCEN.CC2P equal to '0', select clock rising edge polarity
- To select input filter bandwidth by configuring TIMx_CCMOD1.IC2F[3:0] (if filter is not needed, keep IC2F bit at '0000')
- Configure TIMx_SMCTRL.SMSEL equal to '111', select timer external clock mode 1
- Configure TIMx_SMCTRL.TSEL equal to '110', select TI2 as the trigger input source
- Configure TIMx_CTRL1.CNTEN equal to '1' to start the counter

Note: the capture prescaler is not used for triggering, so it does not need to be configured

When the rising edge of the timer clock occurs at TI2=1, the counter counts once and the TIMx_STS .TITF flag is set.

The delay between the rising edge of TI2 and the actual clock of the counter depends on the resynchronization circuit at the input of TI2.

Figure 10-17 Control Circuit in External Clock Mode 1

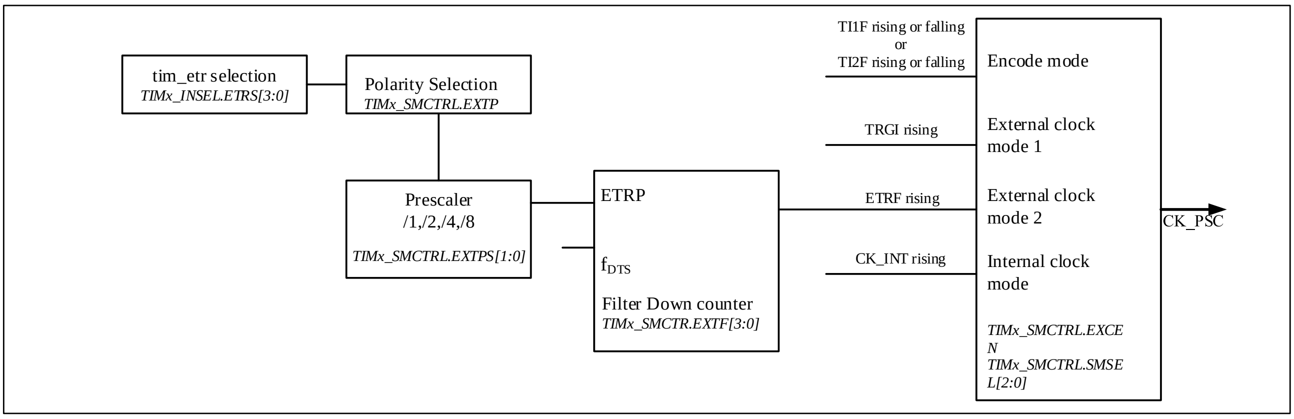


10.5.4.3 External clock source mode 2

This mode is set by configuring TIMx_SMCTRL .EXCEN equal to 1. The counter can count on every rising or falling edge of the external trigger input ETR.

The following figure is a schematic diagram of the external trigger input module in external clock source mode 2.

Figure 10-18 External Trigger Input Block Diagram

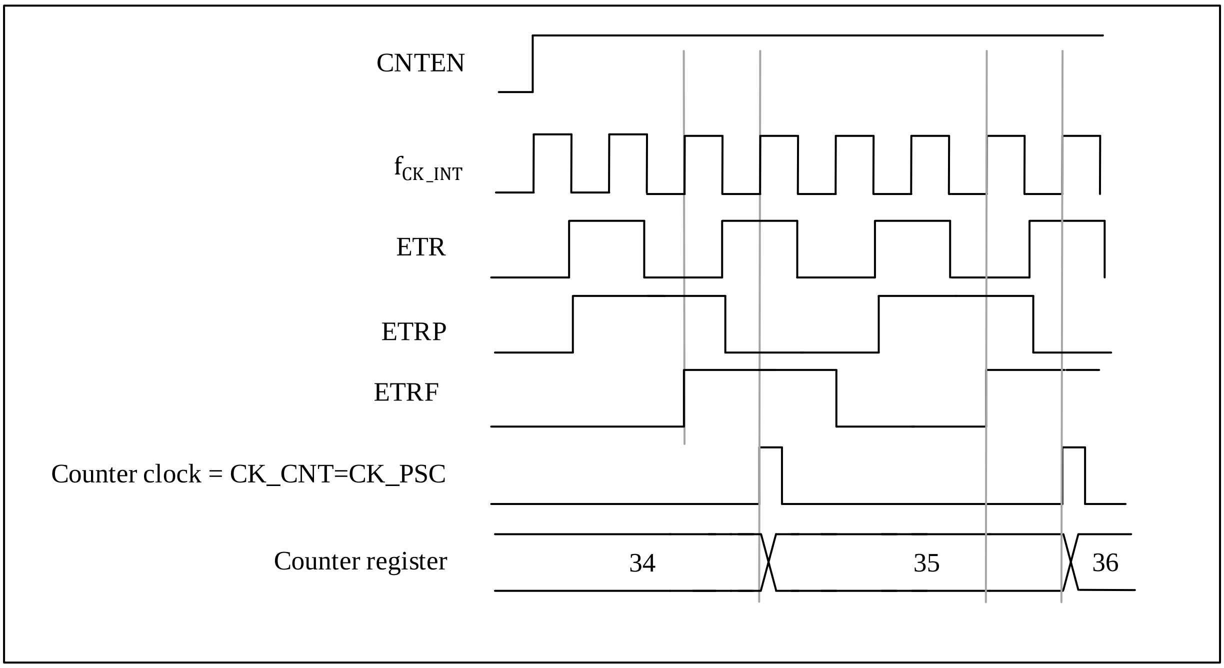


For example, use the following configuration steps to make the upcounter count every 2 rising edges on ETR.

- Since no filter is needed in this case, make TIMx_SMCTRL .EXTF[3:0] equal to ‘0000’
- Configure the prescaler by making TIMx_SMCTRL.EXTPS[1:0] equal to ‘01’
- Select the polarity on ETR pin by setting TIMx_SMCTRL.EXTP equal to ‘0’, the rising edge of ETR is valid
- External clock mode 2 is selected by setting TIMx_SMCTRL .EXCEN equal to ‘1’
- Turn on the counter by setting TIMx_CTRL1. CNTEN equal to ‘1’

The counter counts every 2 rising edges of ETR. The delay between the rising edge of ETR and the actual clock of the counter is due to a resynchronization circuit on the ETRP signal.

Figure 10-19 Control Circuit in External Clock Mode 2

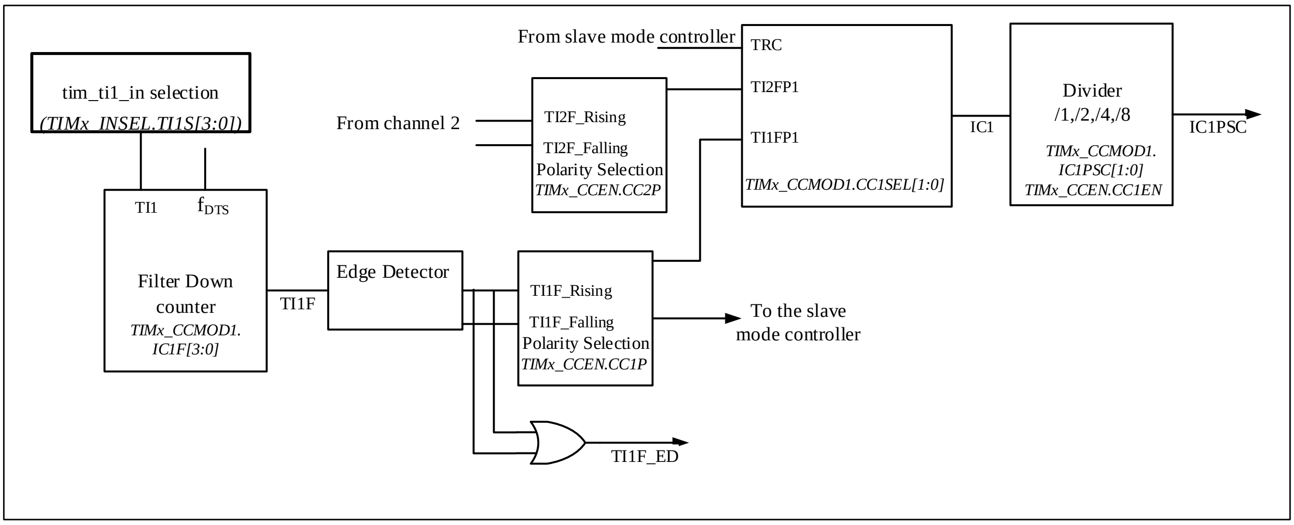


10.5.5 Capture/Compare Channels

The capture/compare channels include capture/compare registers and shadow registers. The input stage consists of digital filters, multiplexers and prescalers. The output section includes comparators and output controls.

The input signal TIx is sampled and filtered to generate the signal TIxF. A signal (TIxF_rising or TIxF_falling) is then generated by the edge detector of the polarity select function, the polarity of which is selected by the TIMx_CCEN.CCxP bits. This signal can be used as a trigger input for the slave mode controller. At the same time, the signal ICx is sent to the capture register after frequencyprecaler. The following figure shows a block diagram of a capture/compare channel.

Figure 10-20 Capture/Compare Channel (Example: Channel 1 Input Stage)



The output stage generates an intermediate waveform OCxRef (active high) as reference. The polarity acts at the end of the chain.

Figure 10-21 Capture/Compare Channel 1 Main Circuit

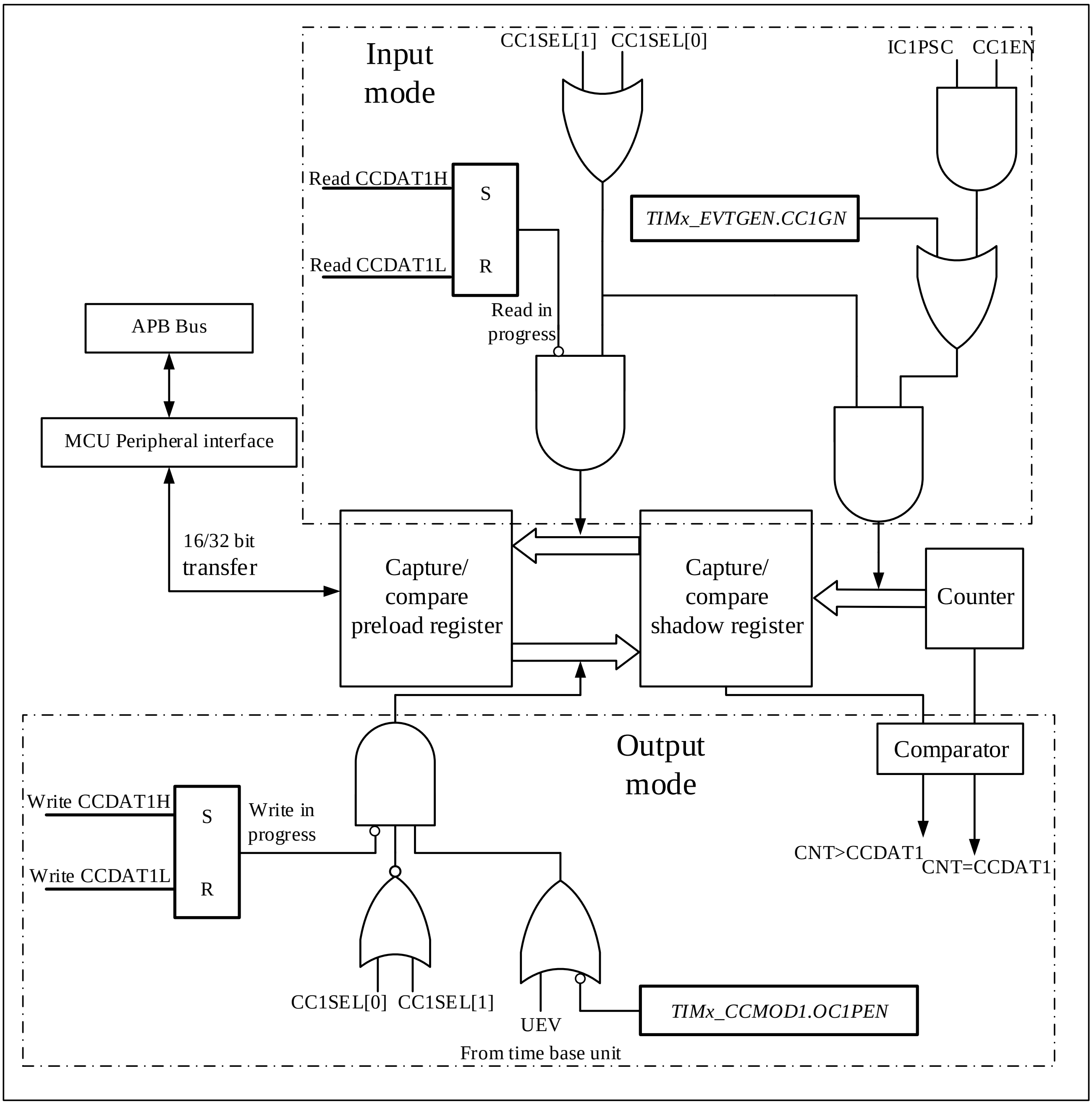
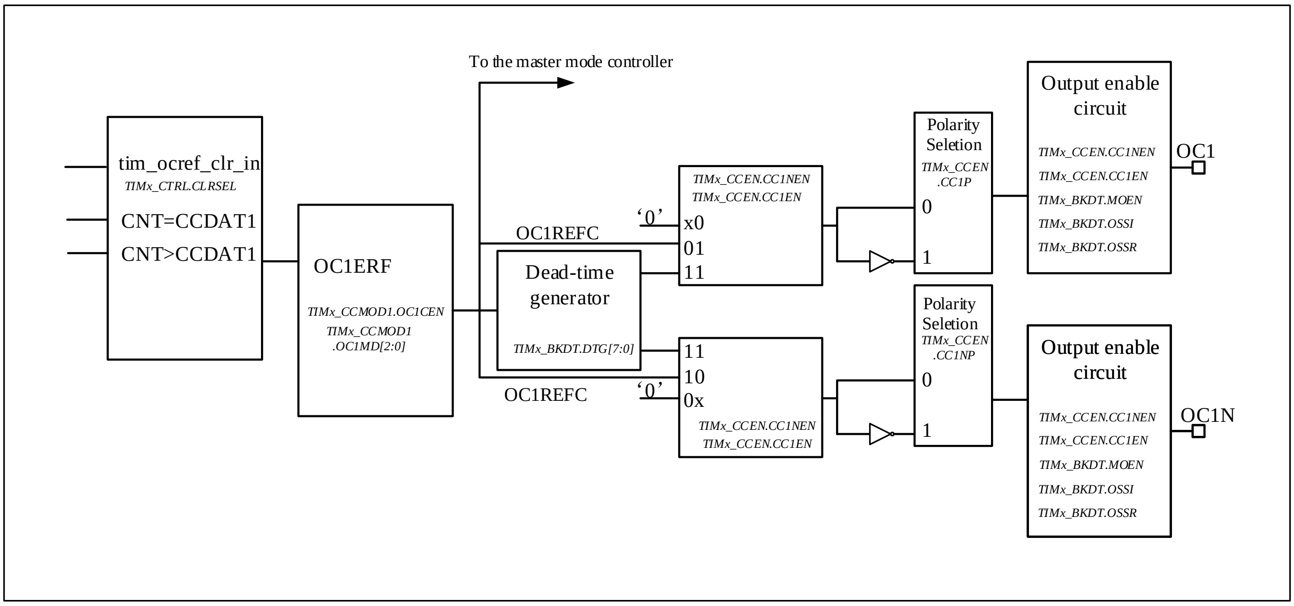


Figure 10-22 Output Part of Channelx (x= 1,2,3,4, Take Channel 1 as Example)



Reads and writes operations always access the preload registers when capturing/comparing. The two specific working processes are as follows:

In capture mode, the capture is actually done in the shadow register, and then the value in the shadow register is copied into the preload register.

In compare mode, the value of the preload register is copied into the shadow register, which is compared with the counter.

10.5.6 Input Capture Mode

In input capture mode, the TIMx_CCDATx registers are used to latch the counter value after the ICx signal detects.

There is a capture interrupt flag TIMx_STS.CCxITF, which can trigger an interrupt or DMA request if the corresponding interrupt enable is set.

The TIMx_STS. CCxITF bit is set by hardware when a capture event occurs and is cleared by software or by reading the TIMx_CCDATx register.

The overcapture flag TIMx_STS.CCxOCF is set equal to 1 when the counter value is captured in the TIMx_CCDATx register and TIMx_STS.CC1ITF is already set. Unlike the former, TIMx_STS.CCxOCF is cleared by writing 0 to it.

To achieve a rising edge of the TI1 input to capture the counter value into the TIMx_CCDAT1 register, the configuration flow is as follows:

- To select a valid input:

  Configure TIMx_CCMOD1.CC1SEL to '01'. At this time, the input is the CC1 channel, and IC1 is mapped to TI1.

- Define the input filter duration required for programming:

  Define the sampling frequency of the TI1 input and the length of the digital filter by configuring the

TIMx_CCMODx.ICxF bits. Example: If the input signal jitters up to 5 internal clock cycles, we must choose a filter duration longer than these 5 clock cycles. When 8 consecutive samples (sampled at f_{DTS} frequency) with the new level are detected, we can validate the transition on TI1. Then configure TIMx_CCMOD1. IC1F to '0011'.

- Select the rising edge as the valid transition polarity on the TI1 channel by configuring TIMx_CCEN .CC1P=0.
- Configure the input prescaler. In this example, configure TIMx_CCMOD1.IC1PSC= '00' to disable the prescaler because we want to capture every valid transition.
- Enable capture by configuring TIMx_CCEN. CC1EN = '1'.

If you want to enable DMA request, you can configure TIMx_DINTEN.CC1DEN=1.If you want enable related interrupt request, you can configureTIMx_DINTEN.CC1IEN bit=1

10.5.7 PWM Input Mode

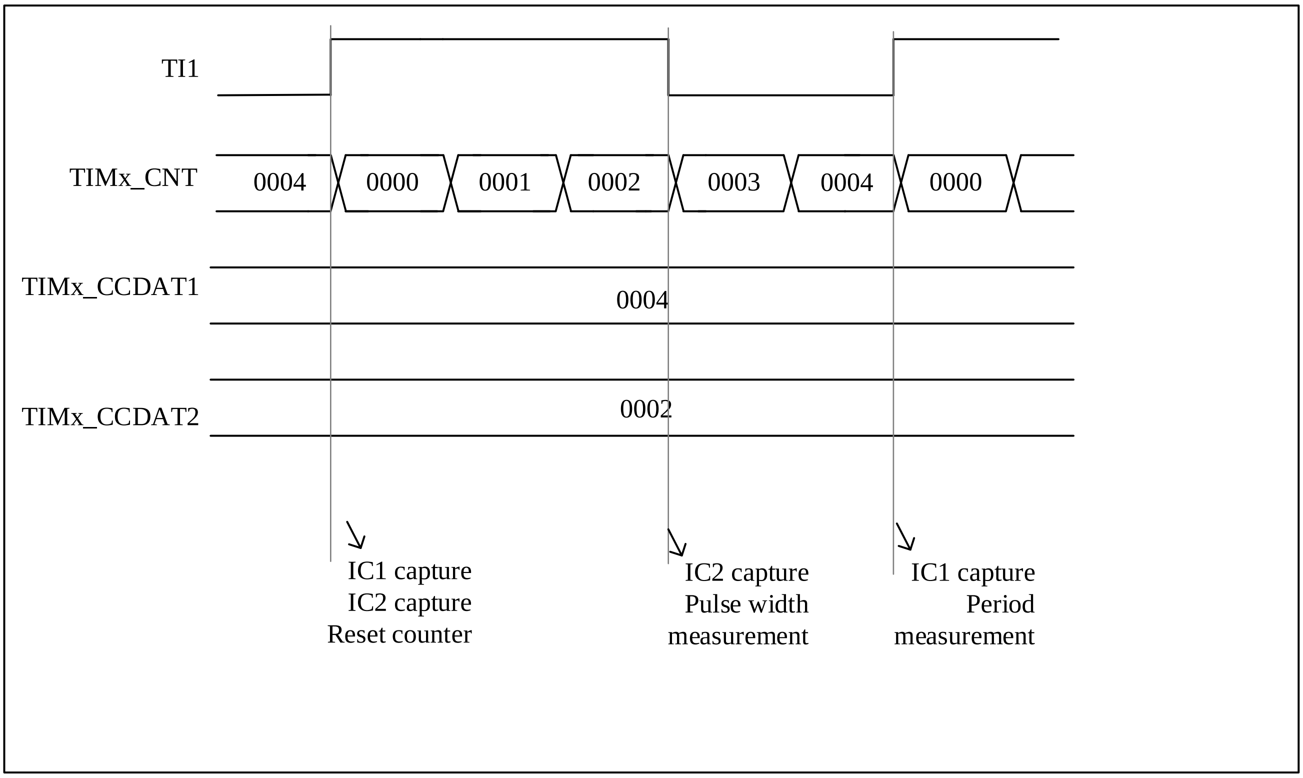
There are some differences between PWM input mode and normal input capture mode, including:

- Two ICx signals are mapped to the same TIx input.
- The two ICx signals are active on edges of opposite polarity.
- Select one of two TIxFP signals as trigger input.
- The slave mode controller is configured in reset mode.

For example, the following configuration flow can be used to know the period and duty cycle of the PWM signal on TI1 (It depends on the frequency of CK_INT and the value of the prescaler).

- Configure TIMx_CCMOD1.CC1SEL equal to '01' to select TI1 as valid input for TIMx_CCDAT1.
- Configure TIMx_CCEN.CC1P equal to '0' to select the active polarity of filtered timer input 1(TI1FP1), active on the rising edge.
- Configure TIMx_CCMOD1.CC2SEL equal to '10' select TI1 as valid input for TIMx_CCDAT2.
- Configure TIMx_CCEN.CC2P equal to 1 to select the valid polarity of filtered timer input 2(TI1FP2), active on the falling edge.
- Configure TIMx_SMCTRL.TSEL=101 to select filtered timer input 1 (TI1FP1) as valid trigger input.
- Configure TIMx_SMCTRL.SMSEL=100 to configure the slave mode controller to reset mode.
- Configure TIMx_CCEN. CC1EN=1 and TIMx_CCEN.CC2EN=1 to enable capture.

Figure 10-23 PWM Input Mode Timing



Because of only filter timer input 1 (TI1FP1) and filter timer input 2 (TI2FP2) are connected to the slave mode controller, the PWM input mode can only be used with the TIMx_CH1/TIMx_CH2 signals.

10.5.8 Forced Output Mode

In output mode (TIMx_CCMODx.CCxSEL=00) , software can force output compare signals to active or inactive level directly.

User can set TIMx_CCMODx. OCxMD=101 to force the output compare signal to active level. And the OCxREF will be forced high, OCx get opposite value to CCxP polarity bit. On the other hand, user can set TIMx_CCMODx. OCxMD=100 to force the output compare signal to low level.

The values of the TIMx_CCDATx shadow register and the counter still comparing with each other in this mode.

The comparison between the output compare register TIMx_CCDATx and the counter TIMx_CNT has no effect on OCxREF. And the flag still can be set. Therefore, the interrupt and DMA requests still can be sent.

10.5.9 Output Compare Mode

User can use this mode to control the output waveform, or to indicate that a period of time has elapsed.

When the capture/compare register and the counter have the same value, the output compare function’s operations are as follow:

- TIMx_CCMODx.OCxMD is for output compare mode, and TIMx_CCEN.CCxP is for output polarity. When the compare matches, if set TIMx_CCMODx.OCxMD=000, the output pin will keep its level;if set

TIMx_CCMODx.OCxMD=001, the output pin will be set active;if set TIMx_CCMODx.OCxMD=010, the output pin will be set inactive;if set TIMx_CCMODx.OCxMD=011, the output pin will be set to toggle.

- Set TIMx_STS.CCxITF.
- If user set TIMx_DINTEN.CCxIEN, a corresponding interrupt will be generated.
- If user set TIMx_DINTEN.CCxDEN and set TIMx_CTRL2.CCDSEL to select DMA request, and DMA request will be sent.

The TIMx_CCDATx registers can be programmed with or without preload registers using the TIMx_CCMODx.OCxPEN register.

The time resolution is one count period of the counter.

In one-pulse mode, the output compare mode can also be used to output a single pulse.

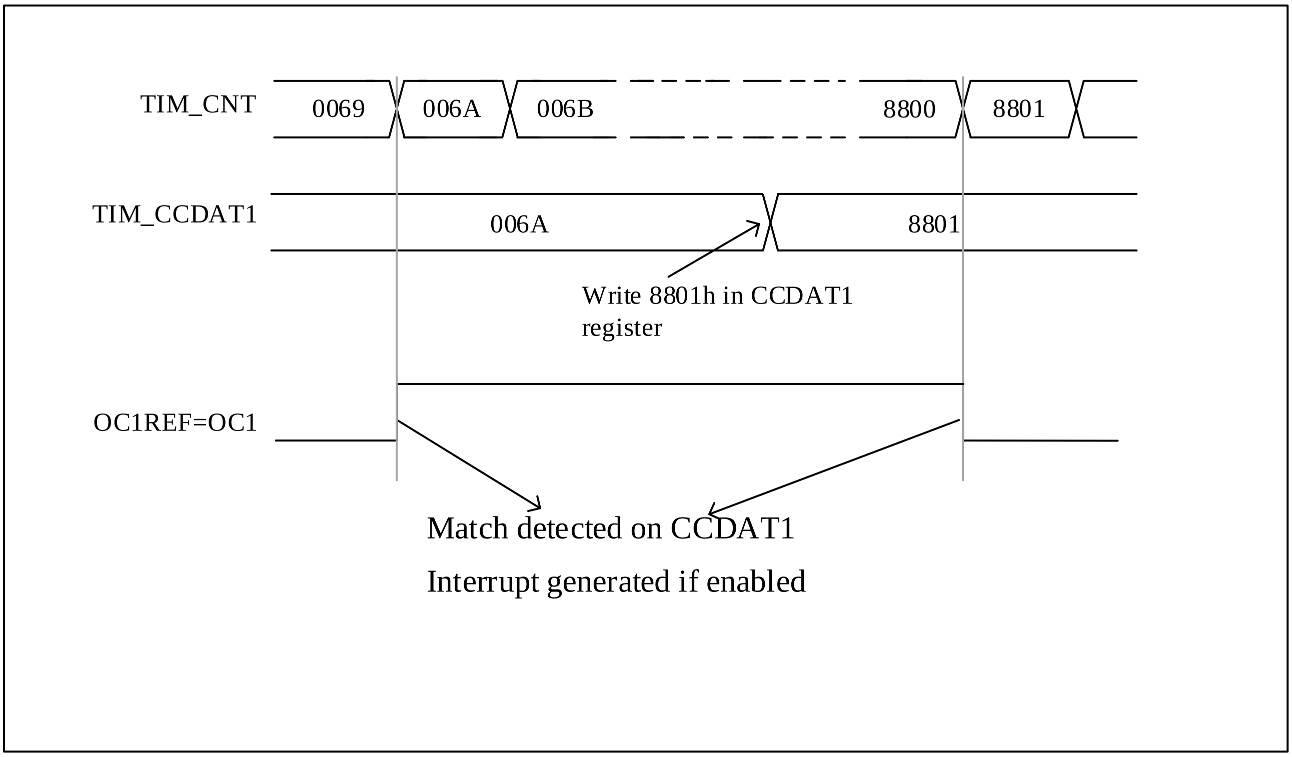
Here are the configuration steps for output compare mode:

- First of all, user should select the counter clock.
- Secondly, write the desired data in the TIMx_AR and TIMx_CCDATxregisters.
- If user need to generate an interrupt, set TIMx_DINTEN.CCxIEN.
- Then select the output mode by set TIMx_CCEN.CCxP, TIMx_CCMODx.OCxMD, TIMx_CCEN.CCxEN, etc.
- At last, set TIMx_CTRL1.CNTEN to enable the counter.

User can update the output waveform by writting TIMx_CCDATx at any time, as long as the preload register is not enabled. Otherwise the TIMx_CCDATx shadow register will be updated at the next update event.

Here is an example.

Figure 10-24 Output Compare Mode, Toggle on OC1



10.5.10 PWM Mode

Pulse width modulation mode is used to generate a signal with a frequency determined by the value of the TIMx_AR register and a duty cycle determined by the value of the TIMx_CCDATx register.. And depending on the value of TIMx_CTRL1.CAMSEL, the TIM can generate PWM signal in edge-aligned mode or center-aligned mode.

User can select PWM mode 1 or PWM mode 2 by setting TIMx_CCMODx. OCxMD=110 or setting TIMx_CCMODx. OCxMD=111. To enable preload register, user must set corresponding TIMx_CCMODx.OCxPEN, and then set TIMx_CTRL1.ARPEN to auto-reload preload register eventually.

User can program polarity of OCx by setting TIMx_CCEN.CCxP. On the other hand, to enable the output of OCx, user need to set the combination of the value of CCxEN, CCxNEN, MOEN, OSSI, and OSSR in TIMx_CCEN and TIMx_BKDT.

The values of TIMx_CNT and TIMx_CCDATx are always compared with each other when the TIM is under PWM mode.

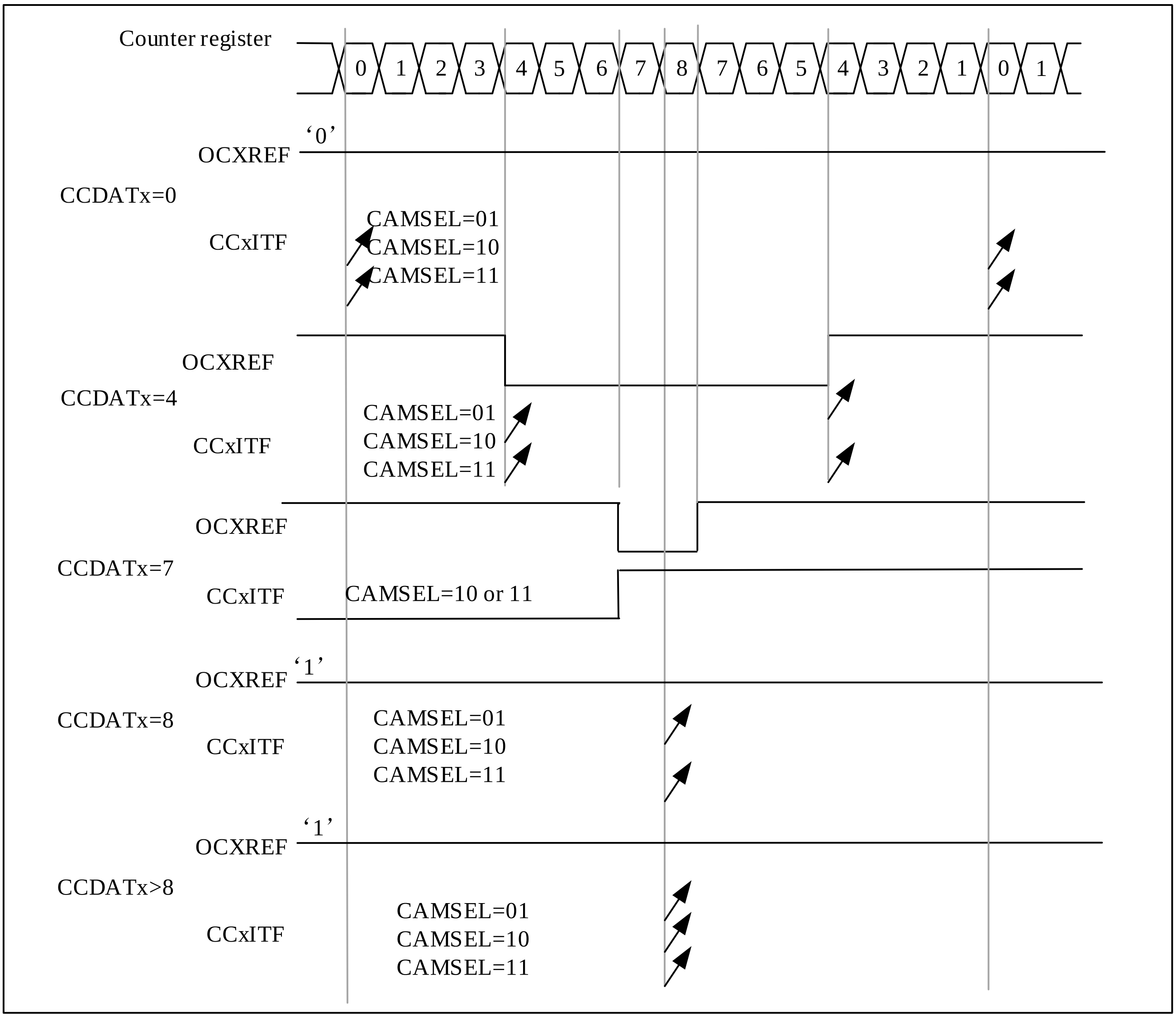
Only when an update event occurs, the preload register will be transferred to the shadow register. Therefore user must reset all the registers by setting TIMx_EVTGEN.UDGN before the counter starts counting.

10.5.10.1 PWM center-aligned mode

If user set TIMx_CTRL1.CAMSEL equal to 01, 10 or 11, the PWM center-aligned mode will be active. The setting of the compare flag depends on the value of TIMx_CTRL1.CAMSEL. There are three kinds of situation that the compare flag is set, only when the counter counts up, only when the counter counts down, or both when the counter counts up and counts down. User should not modified TIMx_CTRL1.DIR by software, it is updated by hardware.

Examples of center-aligned PWM waveforms is as follow, and the setting of the waveform are: TIMx_AR=8, PWM mode 1, the compare flag is set when the counter counts down corresponding to TIMx_CTRL1. CAMSEL=01.

Figure 10-25 Center-Aligned PWM Waveform (AR=8)



When using center-aligned mode, users should pay attention to the following considrations:

- It depends on the value of TIMx_CTRL1.DIR that the counter counts up or down. Cautions that the DIR and CAMSEL bits should not be changed at the same time.
- User should not write the counter while running in center-aligned mode, otherwise it will cause unexpected results. Here are some example:
  - If the value written into the counter is 0 or is the value of TIMx_AR, the direction will be updated but the update event will not be generated.
  - If the value written into the counter is greater than the value of auto-reload, the direction will not be updated.
- For safety reasons, It is recommended that users set TIMx_EVTGEN.UDGN to generate an update by software before starting the counter, and do not write the counter while it is running.

10.5.10.2 PWM center-aligned asymmetric mode

About PWM center-aligned asymmetric mode, refer to Section 10.5.2.3.2。

10.5.10.3 PWM edge-aligned mode

There are two kinds of configuration in edge-aligned mode, up-counting and down-counting.

- Up-counting

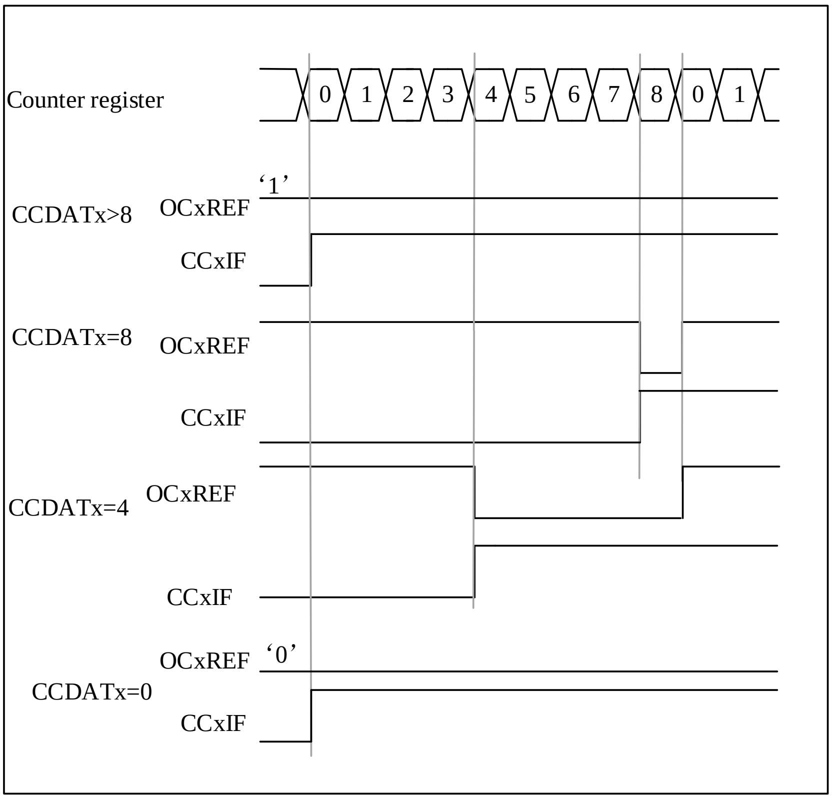
User can set TIMx_CTRL1.DIR=0 to make counter count up.

Example for PWM mode1:

When TIMx_CNT < TIMx_CCDATx, the reference PWM signal OCxREF is high. Otherwise it will be low. If the compare value in TIMx_CCDATx is greater than the auto-reload value, the OCxREF will remains 1. Conversely, if the compare value is 0, the OCxREF will remains 0.

When TIMx_AR=8, the PWM waveforms are as follows.

Figure 10-26 Edge-Aligned PWM Waveform (APR=8)



- Down-counting

User can set TIMx_CTRL1.DIR=1 to make counter counts down.

Example for PWM mode1:

When TIMx_CNT > TIMx_CCDATx, the reference PWM signal OCxREF is low. Otherwise it will be high. If the compare value in TIMx_CCDATx is greater than the auto-reload value, the OCxREF will remains 1.

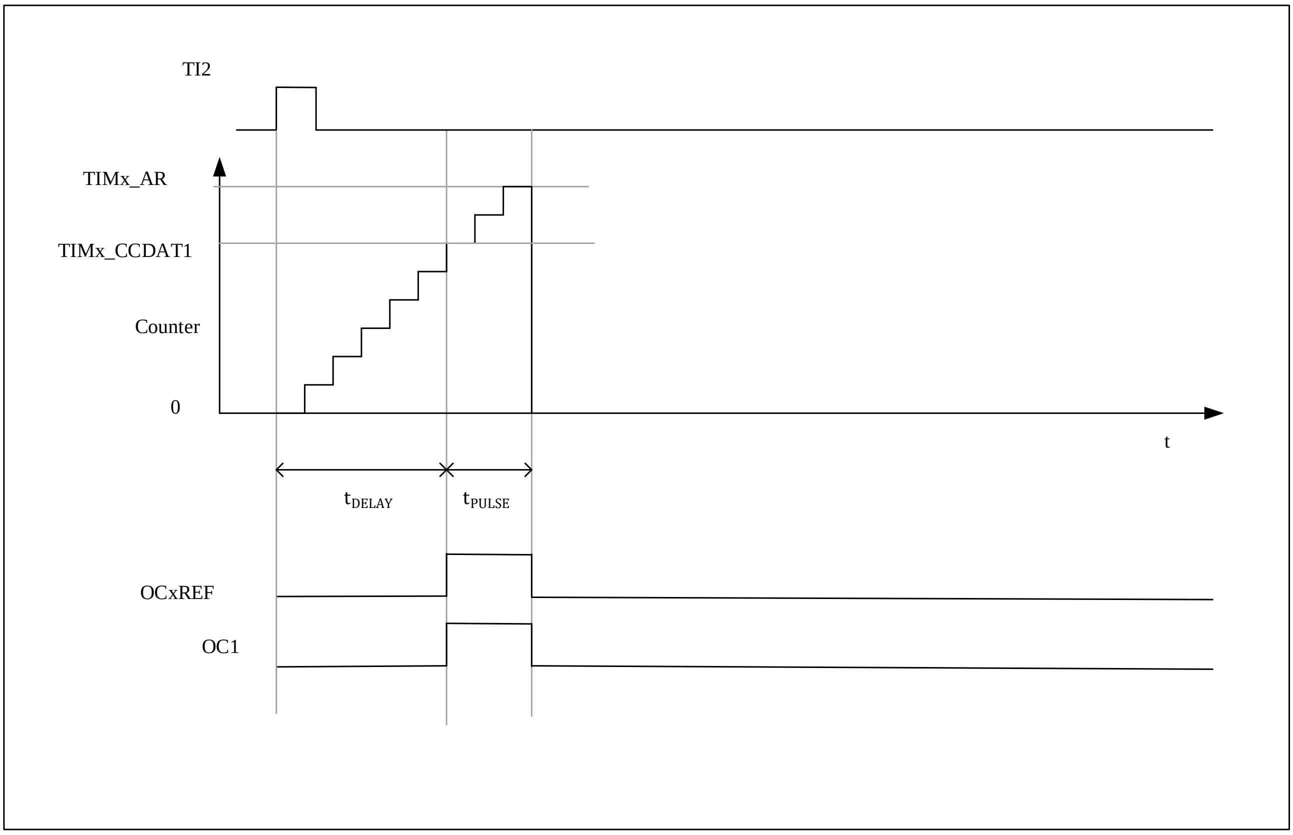
Note: if the n_{th} PWM cycle CCDATx shadow register >= AR value, the shadow register value of CCDATx in the

$(n+1)_{th}$ PWM cycle is 0. At the moment when the counter is 0 in the (n+1)th PWM cycle, although the value of the counter = CCDATx shadow register = 0 and OCxREF = '0', no compare event will be generated.

10.5.11 One-pulse Mode

In the one-pulse mode (ONEPM), a trigger signal is received, and a pulse t_{PULSE} with a controllable pulse width is generated after a controllable delay t_{DELAY}. The output mode needs to be configured as output compare mode or PWM mode. After selecting one-pulse mode, the counter will stop counting after the update event UEV is generated.

Figure 10-27 Example of One-Pulse Mode



The following is an example of a one-pulse mode:

A rising edge trigger is detected from the TI2 input, and a pulse with a width of t_{PULSE} is generated on OC1 after a delay of t_{DELAY}.

1. Counter configuration: count up, counter TIMx_CNT < TIMx_CCDAT1 ≤ TIMx_AR;
2. TI2FP2 is mapped to TI2, TIMx_CCMOD1.CC2SEL= ‘01’; TI2FP2 is configured for rising edge detection, TIMx_CCEN.CC2P= ‘0’;
3. TI2FP2 acts as the trigger (TRGI) of the slave mode controller and starts the counter, TIMx_SMCTRL.TSEL= ‘110’, TIMx_SMCTRL.SMSEL= ‘110’ (trigger mode);
4. TIMx_CCDAT1 writes the count value to be delayed (t_{DELAY}), TIMx_AR - TIMx_CCDAT1 is the count value of the pulse width t_{PULSE};

5. Configure TIMx_CTRL1.ONEPM=1 to enable single pulse mode, configure TIMx_CCMOD1.OC1MD = '111' to select PWM2 mode;

6. Wait for an external trigger event on TI2, and a one pulse waveform will be output on OC1.

10.5.11.1 Special case: OCx fast enable:

In one-pulse mode, an edge is detected through the TIx input, and triggers the start of the counter to count to the comparison value and then output a pulse. These operations limit the minimum delay t_{DELAY} that can be achieved.

You can set TIMx_CCMODx.OCxFEN=1 to turn on OCx fast enable, after triggering the rising edge, the OCxREF signal will be forced to be converted to the same level as the comparison match occurs immediately, regardless of the comparison result. OCxFEN fast enable only takes effect when the channel mode is configured for PWM1 and PWM2 modes.

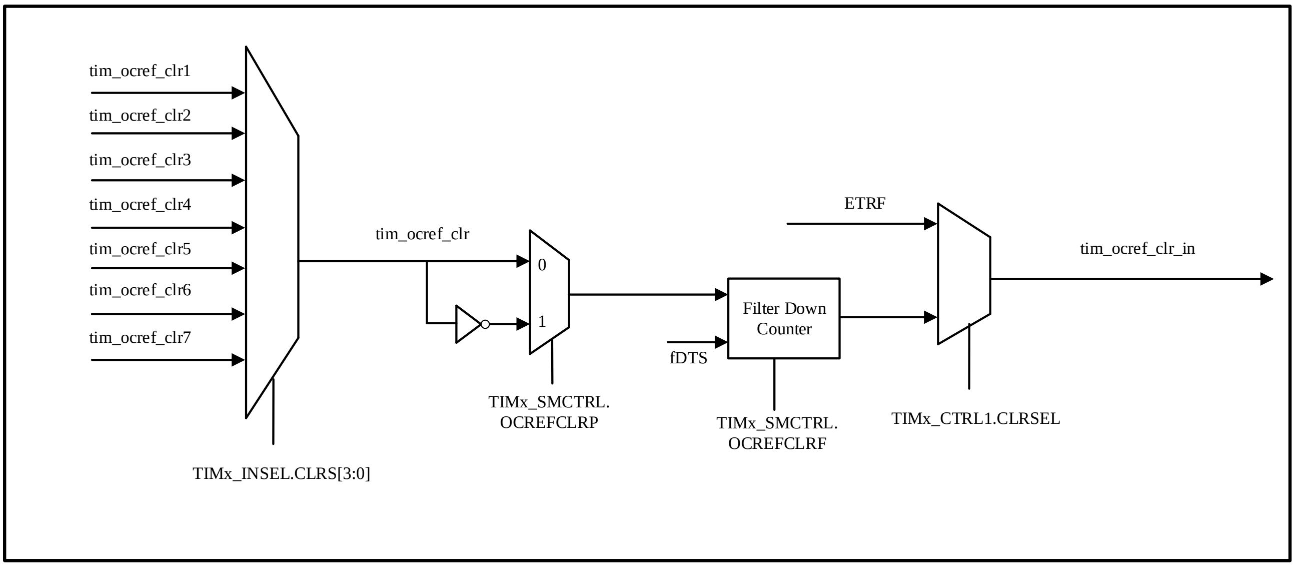
10.5.12 Clearing the Ocxref Signal on an External Event

If the user sets TIMx_CCMODx.OCxCEN=1, high level of ETRF input can be used to driven the OCxREF signal to low, and the OCxREF signal will remains low, until the next UEV happens. Only Output Compare and PWM modes can use this function. This cannot be used when it is in forced mode.

The input clear signal tim_ocref_clr_in can be selected as tim_ocref_clr or ETRF through the CLRSEL bit in the TIMx_CTRL1 register.

The tim_ocref_clr signal can be selected through the CLRS[3:0] in the TIMx_INSEL register, as shown in the following diagram.

Figure 10-28 Diagram of External Event Clearing Ocxref Signal



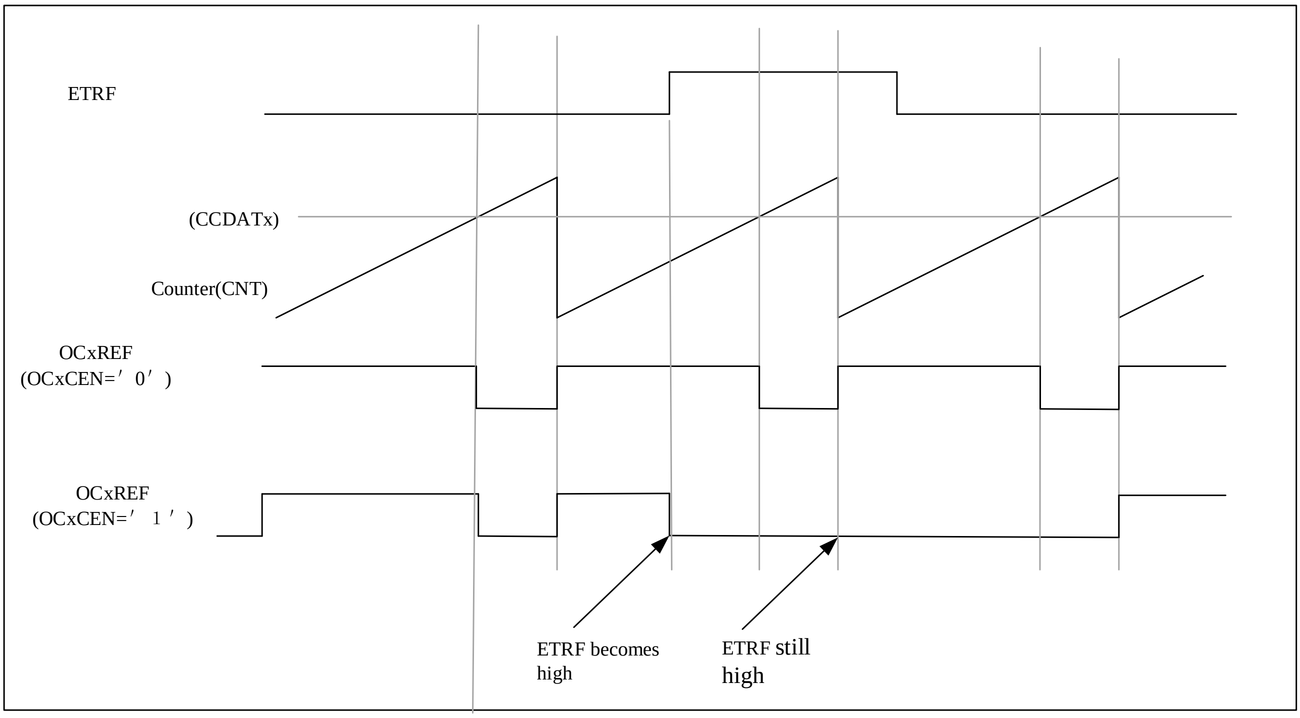
Example: When the tim_ocref_clr_in signal is selected as ETRF, the tim_etr_in configuration is as follows:

- Set TIMx_SMCTRL.EXTPS=00 to disable the external trigger prescaler.
- Set TIMx_SMCTRL.EXCEN=0 to disable the external clock mode 2.
- Set TIMx_SMCTRL.EXTP and TIMx_SMCTRL.EXTF to configure the external trigger polarity and external

trigger filter according to the need.

For example: The following diagram shows when ETRF input becomes high, the behavior of OCxREF signal for different value of OCxCEN. Timer is set to be in PWM mode in this case.

Figure 10-29 Clearing the Ocxref of TIMx



10.5.13 Complementary Outputs with Dead-Time Insertion

The advanced-control timer can output two complementary signals, and manage the switching-off and switching-on instants of outputs , which involves inserting dead time. Users should adjust the dead time based on the characteristics of the connected device to ensure proper operation and avoid potential issues.

User can select the polarity of outputs by setting TIMx_CCEN.CCxP and TIMx_CCEN.CCxNP. And this selection is independently for each output.

User can control the complementary signals OCx and OCxN by setting the combination of several control bits, which are TIMx_CCEN.CCxEN, TIMx_CCEN.CCxNEN, TIMx_BKDT.MOEN, TIMx_CTRL2.OIx, TIMx_CTRL2.OIxN, TIMx_BKDT.OSSI, and TIMx_BKDT.OSSR. When switching to the IDLE state, the dead-time will be activated.

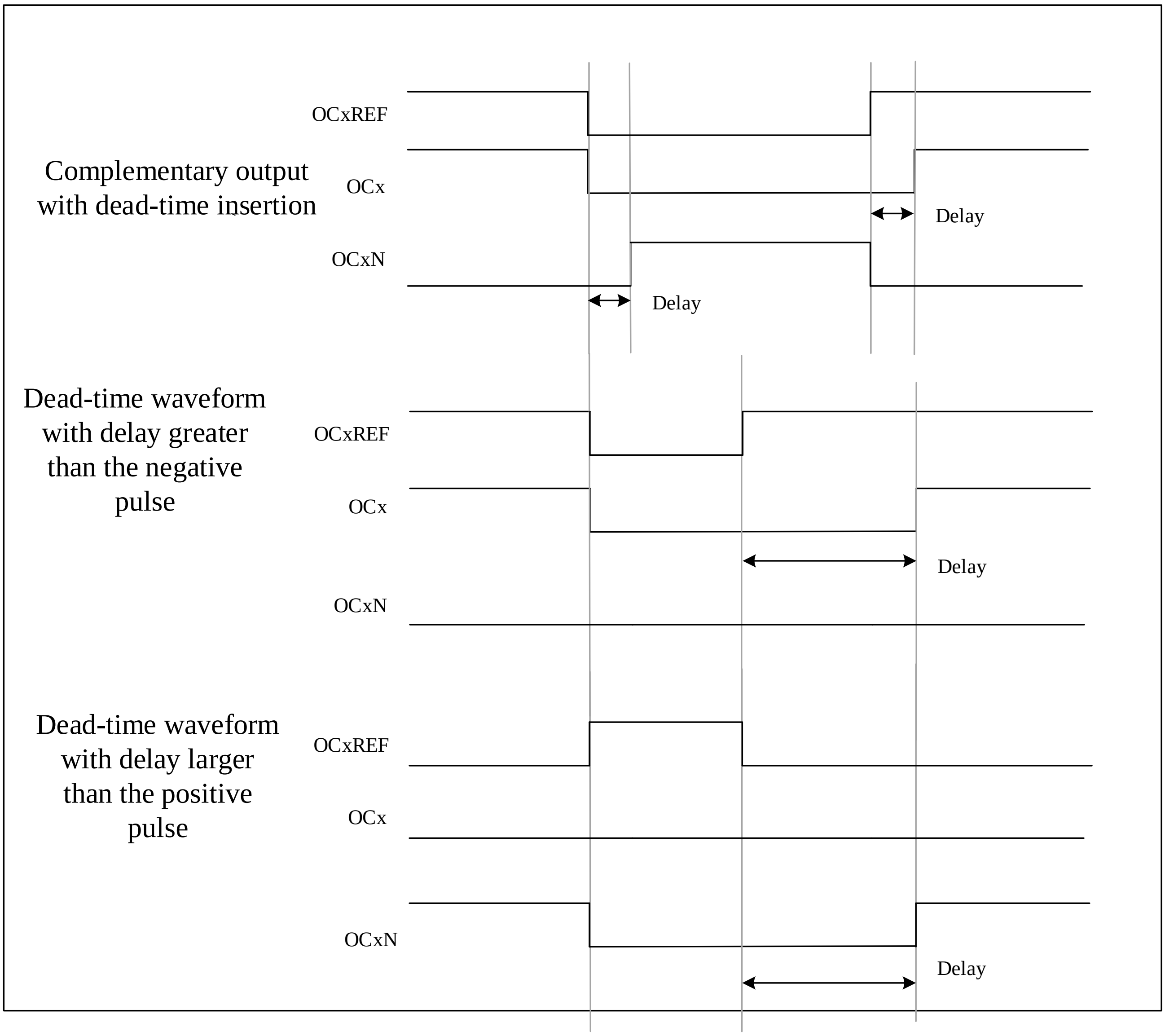
If user set TIMx_CCEN.CCxEN and TIMx_CCEN.CCxNEN at the same time, a dead-time will be insert. If there is a break circuit, the TIMx_BKDT.MOEN should be set too. There are 10-bit dead-time generators for each channel.

Reference waveform OCxREF can generates 2 outputs OCx and OCxN. And if OCx and OCxN are active high, the OCx ouput signal is the same as the reference signal and the OCxN output signal is the opposite of the reference signal. However, OCx output signal will be delayed relative to the reference rising edge and the OCxN output signal will be delayed relative to the reference falling edge. If the delay is greater than the width of the active OCx or OCxN output, the corresponding pulse will not generated.

The relationships between the output signals of the dead-time generator and the reference signal OCxREF are as follow.

Assume that TIMx_CCEN.CCxP=0, TIMx_CCEN.CCxNP=0, TIMx_BKDT.MOEN=1, TIMx_CCEN.CCxEN=1, TIMx_CCEN.CCxNEN=1.

Figure 10-30 Complementary Output with Dead-Time Insertion


Complementary output with dead-time insertion
OCxREF
OCx
OCxN
Delay
Delay
Dead-time waveform with delay greater than the negative pulse
OCxREF
OCx
Delay
OCxN
Dead-time waveform with delay larger than the positive pulse
OCxREF
OCx
OCxN
Delay

User can set TIMx_BKDT.DTGN to programme the dead-time delay for each of the channels.

10.5.13.1 Redirecting OCxREF to OCx or OCxN

In output mode, user can set TIMx_CCEN.CCxEN and TIMx_CCEN.CCxNEN to re-directed OCxREF to the OCx output or to OCxN output.

Here are two ways to use this function. When the complementary remains at its inactive level, user can use this function to send a specific waveform, such as PWM or static active level. User can also use this function to set both outputs in their inactive level or both outputs active and complementary with dead-time.

If user set TIMx_CCEN.CCxEN=0 and TIMx_CCEN.CCxNEN=1, it will not complemented, and OCxN will become active when OCxREF is high. On the other hand, if user set TIMx_CCEN.CCxEN=1 and TIMx_CCEN.CCxNEN=1, OCx will become active when OCxREF is high. On the contrary, OCxN will become active when OCxREF is low.

10.5.14 Break Function

The output enable signals and inactive levels will be modified when setting the corresponding control bits when using the break function. However, the output of OCx and OCxN cannot at the active level at the same time no matter when, that is, (CCxP ^ OIx) ^ (CCxNP^OIxN)! = 0.

When multiple break signals are enabled, each break signal constitutes an OR logic. Here are some signal which can be the source of break and break2.

Break：

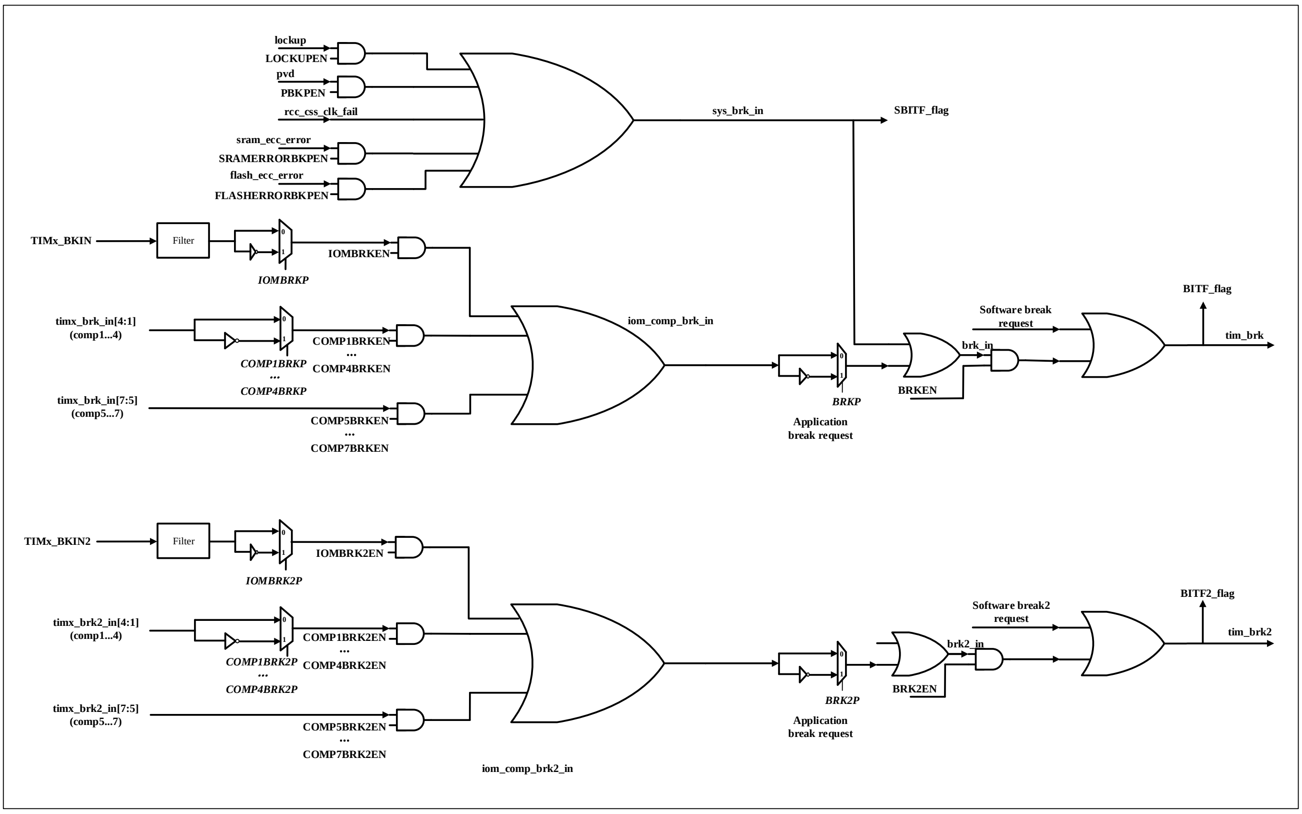
- The break input pin
- A clock failure event, generated by the clock security system in the clock controller
- A PVD failure event
- Core Hardfault event
- SRAM ECC error
- SRAM parity check error
- FLASH ECC error
- The output signal of the comparator
- By software through the TIMx_EVTGEN.BGN

Break2：

- The break2 input pin
- The output signal of the comparator
- By software through the TIMx_EVTGEN.BGN

Perform an OR operation on all sources entering the tim_brk or tim_brk2 input before they are input, as shown in the diagram below.

Figure 10-31 Break and Break2 circuitry overview



Note: An asynchronous (clockless) operation is only guaranteed when the programmable filter is disabled. If it is enabled, a fail safe clock mode (for example by using the internal PLL and/or the CSS) must be used to guarantee that break events are handled.

The break circuit will be disable after reset. And the MOEN bit will be low. User can set TIMx_BKDT.BKEN/ BK2EN to enable the break function. The polarity of break input signal can be selected by setting TIMx_BKDT.BKP/BK2P. User can modify the TIMx_BKDT.BKEN/BK2EN and TIMx_BKDT.BKP/BK2P at the same time. After user set the TIMx_BKDT.BKEN/BK2EN and TIMx_BKDT.BKP/BK2P, there is 1 APB clock cycle delay before the configuration take effect. Therefore, user need to wait 1 APB clock cycle to read back the written bit value.

The falling edge of MOEN can be asynchronous, so a resynchronization circuit has been inserted between the actual signal and the synchronous control bit. This circuit will cause a delay between the asynchronous and the synchronous signal. When user set TIMx_BKDT.MOEN while it is low, user need to insert a delay before reading the value. Because an asynchronous signal was written but user read the synchronous signal.

The behaviors that after a break occurs are as follow:

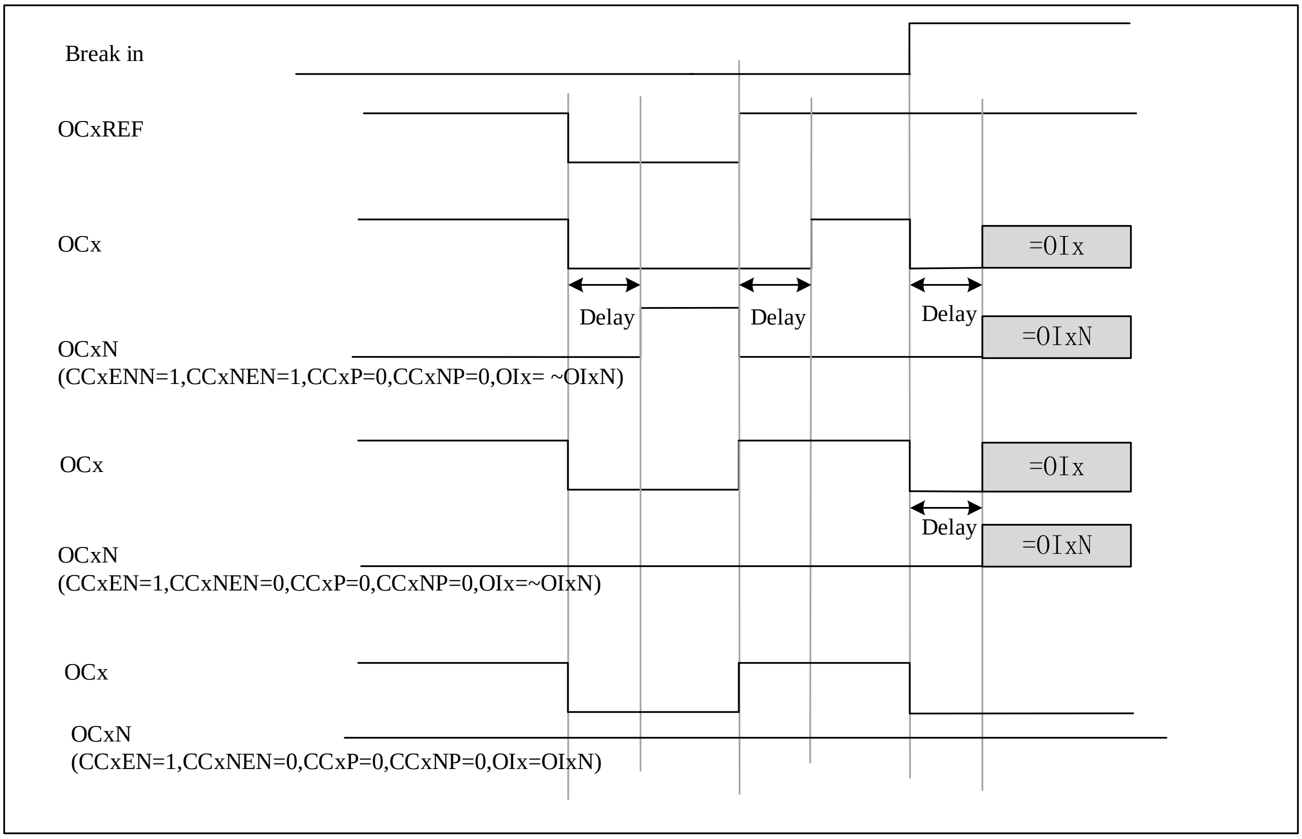
- TIMx_BKDT.MOEN will be cleared asynchronously, and then the outputs will be put in inactive state, idle state or reset state. The state of output is selected by setting TIMx_BKDT.OSSI. This will take effect even if the MCU oscillator is off.
- Once TIMx_BKDT.MOEN=0, the output of each output channel will be driven with the level programmed in TIMx_CTRL2.OIx. Timer will release the enable outputs(taken over by GPIO controller) if TIMx_BKDT.OSSI=0, otherwise it will remain high.

- If user choose to use complementary outputs, the behaviors of TIM are as follow:
  - Depends on the polarity, the outputs will be set in reset state first. It is an asynchronous option so it still works even if there is no clock provided to the timer.
  - The dead-time generator will be reactivated if the timer clock is still provided, and drive the outputs according to the value of TIMx_CTRL2.OIx and TIMx_CTRL2.OIxN after the dead-time when (CCxP ^ OIx) ^ (CCxNP^OIxN)! = 0, that is, the OCx and OCxN still cannot be driven to active level at the same time. Note that the dead-time will be longer than usual because of the resynchronization on MOEN (almost 2 cycles of ck_tim).
  - Timer will release the output control if TIMx_BKDT.OSSI=0. Otherwise, if the enable output was high, it will remain high. If it was low, it will become high when TIMx_CCEN.CCxEN or TIMx_CCEN.CCxNEN is high.
- If TIMx_DINTEN.BIEN=1, when TIMx_STS.BITF=1, an interrupt will be generated.
- If user set TIMx_BKDT.AOEN, the TIMx_BKDT.MOEN will be set automatically when the next UEV happened. User can use this to regulate. If user did not set TIMx_BKDT.AOEN, the TIMx_BKDT.MOEN will remain low until been set 1 again. At this situation, user can use this for security. User can connect the break input to thermal sensors, alarm for power drivers, or other security components.
- When the break input is active, TIMx_BKDT.MOEN cannot be set automatically or by software at the same time, and the TIMx_STS.BITF cannot be cleared. Because the break inputs are active on level.

To ensure the security of application, the break circuit has the write protection function, and there is break input and output management too. It allow user to freeze some parameters, such as dead-time duration, OCx/OCxN polarities and state when disabled, OCxMD configurations, break enable and polarity. User can choose one of the 3 levels of protection to use by setting TIMx_BKDT.LCKCFG. However, the TIMx_BKDT.LCKCFG can only be written once after an MCU reset.

An example for output behavior in response to a break is as follow:

Figure 10-32 Output Behavior in Response to a Break


Break in
OCxREF
OCx
Delay
Delay
Delay
=OIx
=OIxN
OCxN
(CCxENN=1,CCxNEN=1,CCxP=0,CCxNP=0,OIx= ~OIxN)
OCx
=OIx
Delay
=OIxN
OCxN
(CCxEN=1,CCxNEN=0,CCxP=0,CCxNP=0,OIx=~OIxN)
OCx
OCxN
(CCxEN=1,CCxNEN=0,CCxP=0,CCxNP=0,OIx=OIxN)

Two brake inputs have different behaviors for the timer output:

- The tim_brk input can disable (invalid state) the PWM output or force the PWM output to a predefined safe state.
- The tim_brk2 can only disable (invalid state) the PWM output.

The priority of the tim_brk input is higher than the tim_brk2 input, as shown in the table below.

Note: tim_brk2 must be used only when OSSR = OSSI = 1.

Figure 10-12 Behavior of Timer Outputs versus tim_brk/tim_brk2 Inputs

tim_brk	tim_brk2	Timer Outputs State	Typical Use Case	
			OCxN Output	OCx Output
Active	X	–Invalid, then forced output state (after a deadtime) –Outputs disabled if OSSI = 0 (control taken over by GPIO logic)	ON after deadtime insertion	OFF

Inactive	Active	Inactive	OFF	OFF

The following diagram provides an example of the behavior of OCx and OCxN outputs when a valid signal appears on the tim_brk and tim_brk2 inputs. In this case, the polarity of both outputs is active high (CCxP = CCxNP = 0 in the TIMx_CCEN register).

Figure 10-33 PWM Output State Following tim_brk and tim_brk2 Assertion (OSSI=1)

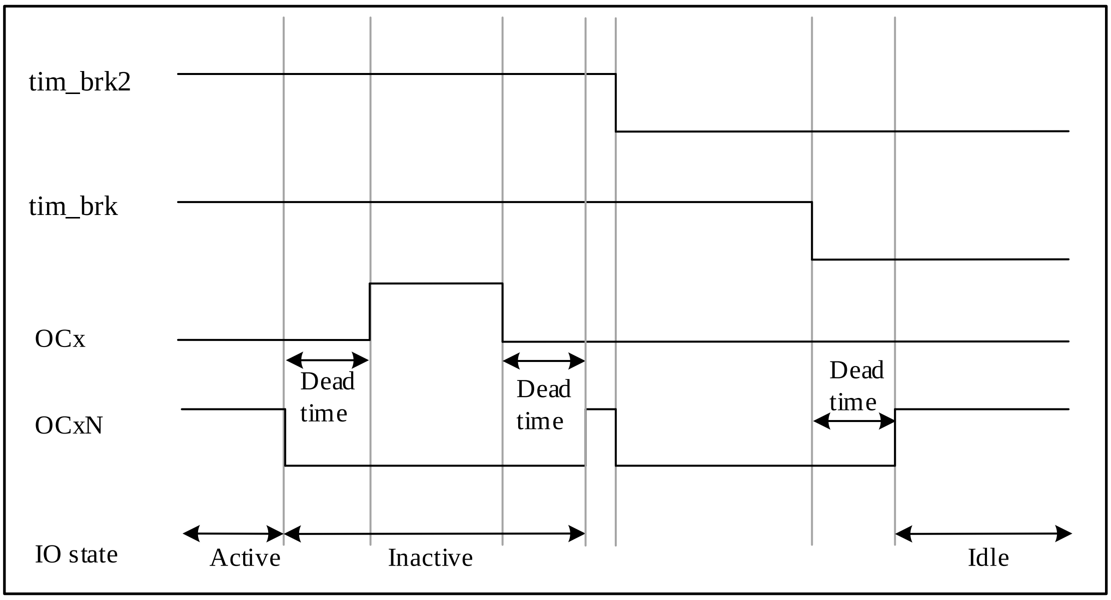
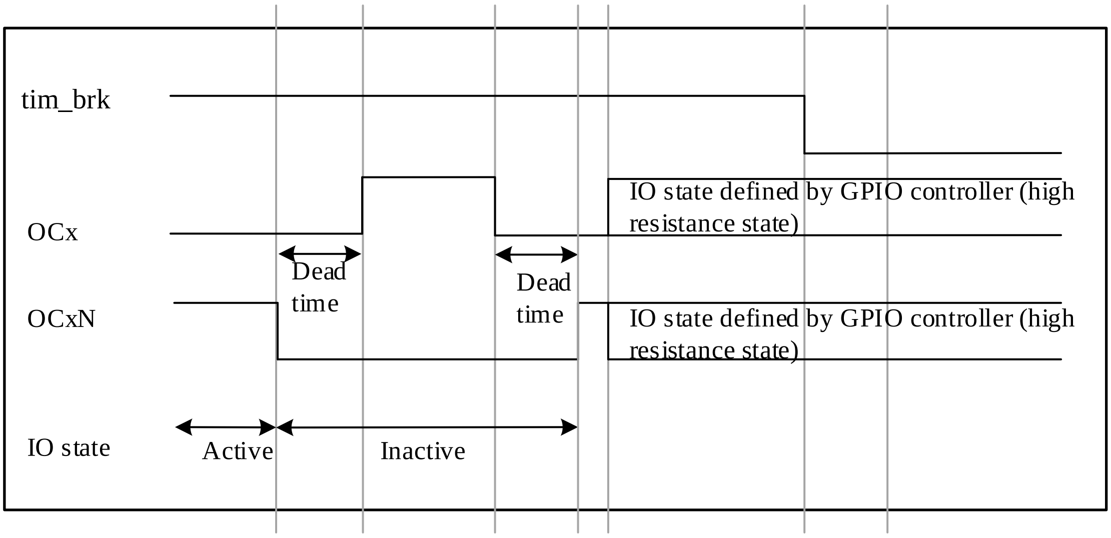


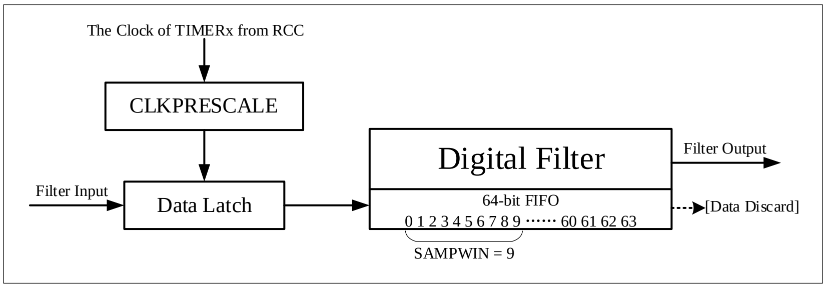
Figure 10-34 PWM Output State Following tim_brk Assertion (OSSI=0)


tim_brk
OCx
OCxN
Dead time
Dead time
IO state defined by GPIO controller (high resistance state)
IO state defined by GPIO controller (high resistance state)
IO state
Active
Inactive

10.5.14.1 Break filter

Register TIMx_BKFR description are as follow:

Figure 10-35 Slide Filter



- The digital filter samples break signal at the clock of TIMx from RCC, accumulating samples in a 64-bits FIFO. Only sampled data within window size defined in TIMx_BRKFR.WSIZE [5:0] with maximum size 64.
- The filter outputs the majority value inside sample window which is defined by the threshold value in TIMx_BRKFR.THRESH [5:0] with maximum threshold of 63. This value should be equal or more than half of window size. If neither logic 1 nor logic 0 counts inside sampling window is more than threshold, digital filter maintain previous output value.
- TIMx_ SLIDFPSC. SLIDFPSC[15:0] register determines sample rate of corresponding digital filter. Filter FIFO capture one sample value from input at every sample clock.
- If digital filter is off, the filter will be bypassed.

10.5.15 Bidirectional Break

ATIM1/ATIM2/ATIM3 have bidirectional brake I/O functionality.

This provides support for:

- A board-level global brake signal, which can be used to send a fault signal to an external MCU or gate driver through a unique IO (input and output status pin).
- When multiple internal and external brake sources need to be combined, they are connected together through an "OR" operation to generate a single brake event.

The tim_brk and tim_brk2 inputs are configured for bidirectional mode using the BRKBID/BRK2BID in the TIMx_BDTR register. BRKBID/BRK2BID can be locked in read-only mode using the LOCK bit in the TIMx_BDTR register (at LOCK level 1 or above).

Bidirectional mode is available for both the tim_brk and tim_brk2 inputs, requiring the I/O configuration to be in open-drain mode with a low-level polarity (through the TIMx_AF1.IOMBRKP, TIMx_BKDT.BKP, TIMx_AF1.IOMBRK2P, and TIMx_BKDT.BK2P bits). Any break request from the system (such as CSS), on-chip peripherals, or break inputs will force the break inputs to a low level to indicate a fault event. For safety reasons, if the polarity bit is not set correctly, bidirectional mode will be inhibited (e.g. set to active high, bidirectional mode will not take effect).

Software break events (TIMx_EVTGEN.BGN and BGN2) also force the break IO to "0" to indicate to external devices that the timer has entered a break state. However, this is only effective when the break is enabled (TIMx_BKDT.BKEN or BK2EN=1). When a software break event occurs (TIMx_BKDT.BKEN or BK2EN=0), the output is put in a safe state, and the break flag is set. However, it has no effect on the TIMx_BKIN and TIMx_BKIN2 I/O.

A safe disarming mechanism prevents the system to be completely locked-up (a low level on the break input triggers a break which enforces a low level on the same input).

When the TIMx_BKDT.BRKDSRM (BRK2DSRM) bit is set to 1, the break output is released to clear a fault signal and provide the possibility to re-arm the system.

At any time, the break protection circuit cannot be disabled:

- The break input path is always active: the break event remains active even if the TIMx_BKDT.BRKDSRM (BRK2DSRM) bit is set and the open-drain control is released. This prevents the PWM output to be re-started when break conditions are present.
- As long as the output is enabled (TIMx_BKDT.MOEN bit is set), the TIMx_BKDT.BRKDSRM (BRK2DSRM) bit cannot disarm the break protection.

Table 10-13 Break Protection Disarming Conditions

MOEN	BRKBID (BRK2BID)	BRKDSRM (BRK2DSRM)	Break Protection State
0	0	X	Armed
0	1	0	Armed
0	1	1	Disarmed
1	X	X	Armed

Arming and re-arming break circuitry

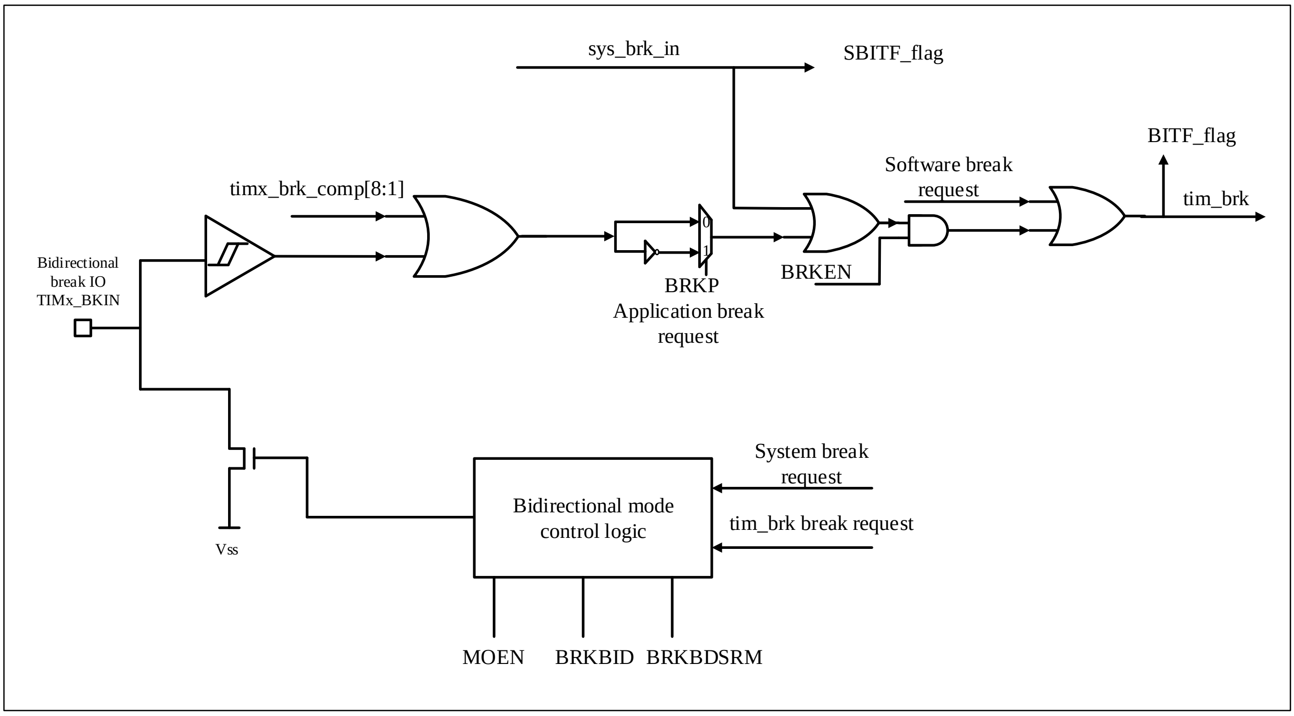
The break circuit (in input or bidirectional mode) is armed by default (peripheral reset configuration).

After a break (break2) event, the following procedure must be followed to re-arm the protection:

- The TIMx_BKDT.BRKDSRM (BRK2DSRM) bit must be set to release the output control.
- The software must wait for the system break conditions to disappear (if any) and clear the TIMx_STS.SBIF status flag (or clears it systematically before re-arming).
- The software must poll the TIMx_BKDT.BRKDSRM (BRK2DSRM) bit until it is cleared by hardware (when the application break conditions disappear).

From this point, the break circuit is armed and active, and the TIMx_BKDT.MOEN bit can be set to re-enable the PWM output.

Figure 10-36 Output Redirection (tim_brk2 Request Not Represented)



10.5.16 Debug Mode

When the microcontroller is in debug mode (the Cortex^{TM}-M4F core halted), depending on the DBG_TIMx_STOP configuration in the DBG module, the TIMx counter can either continue to work normally or stop. For more details, refer to Section 37.3.2。

10.5.17 ATIMx and External Trigger Synchronization

TIMx can be synchronized by a trigger in slave modes (reset, trigger and gated).

10.5.17.1 Slave mode: Reset mode

In reset mode, the trigger event can reset the counter and the prescaler updates the preload registers TIMx_AR, TIMx_CCDATx, and generates the update event UEV (TIMx_CTRL1.UPRS=0).

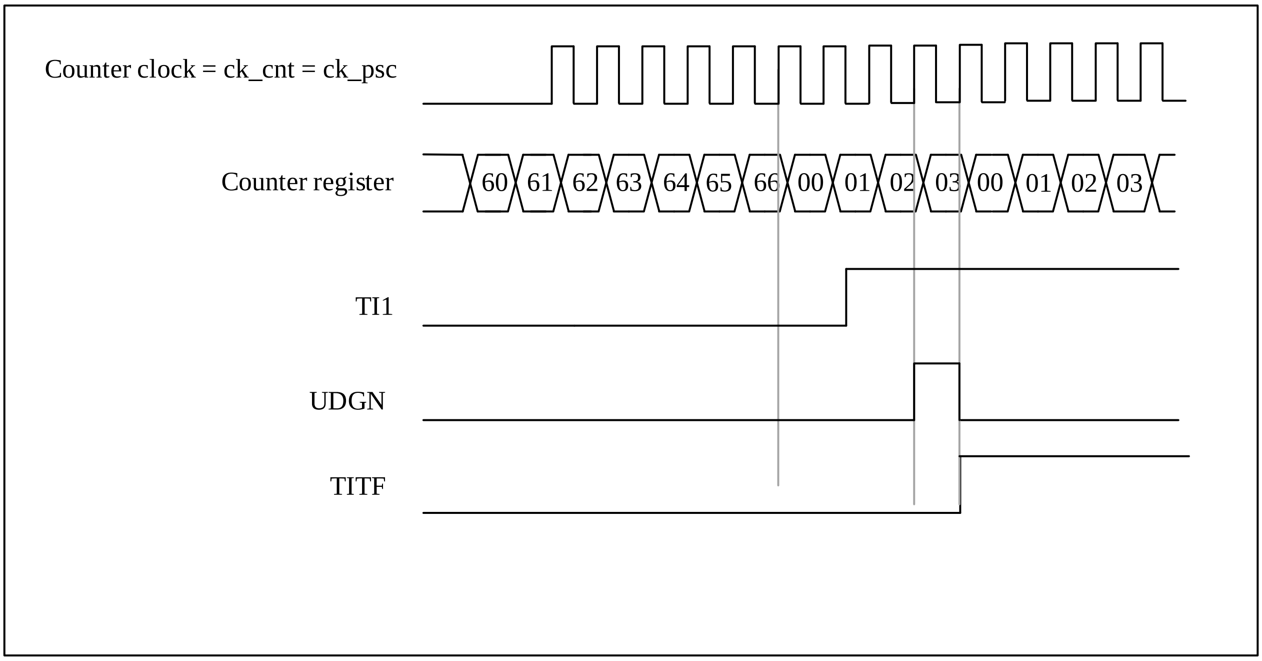
The following is an example of a reset mode:

- Channel 1 is configured as input to detect the rising edge of TI1 (TIMx_CCMOD1.CC1SEL=01, TIMx_CCEN.CC1P=0);
- The slave mode is selected as reset mode (TIMx_SMCTRL.SMSEL=0100), and the trigger input is selected as TI1 (TIMx_SMCTRL.TSEL=101);
- Setting TIMx_CTRL1.CNTEN = 1 to start counter;

After starting the timer, when TI1 detects a rising edge, the counter resets and restarts counting, and the trigger flag is set (TIMx_STS.TITF=1);

The delay between the rising edge on TI1 and the actual reset of the counter is due to the resynchronization circuit on TI1 input.

Figure 10-37 Control Circuit in Reset Mode



10.5.17.2 Slave mode: Trigger mode

In trigger mode, the trigger event (rising edge/falling edge) of the input port can trigger the counter to start counting.

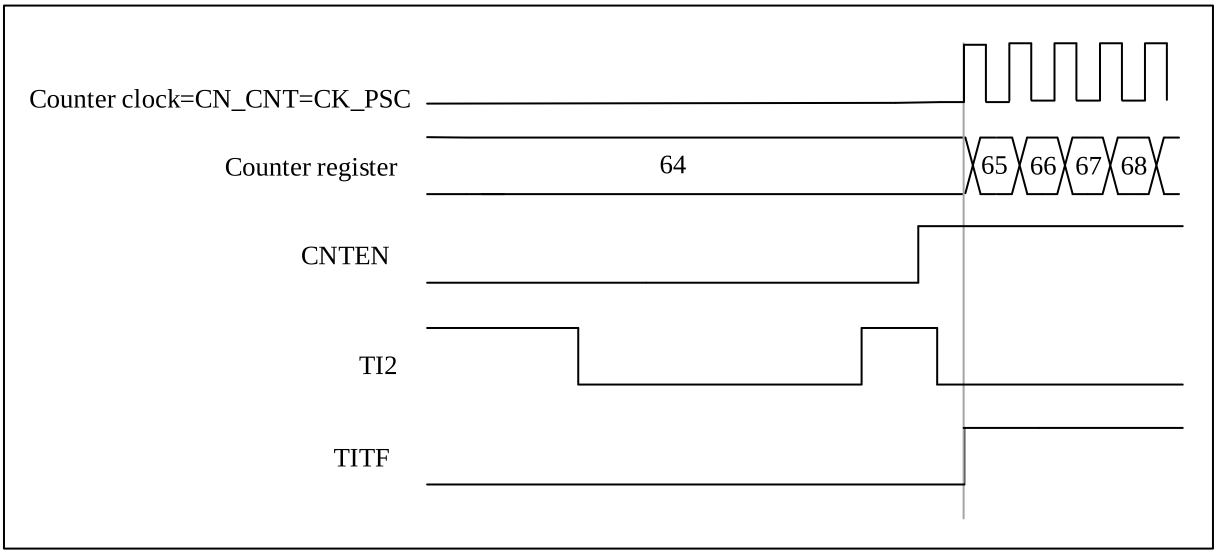
The following is an example of a trigger pattern:

- Channel 2 is configured as input to detect the rising edge of TI2 (TIMx_CCMOD1.CC2SEL=01, TIMx_CCEN.CC2P=0);
- Select from mode to trigger mode (TIMx_SMCTRL.SMSEL=0110), select TI2 for trigger input (TIMx_SMCTRL.TSEL=110);

When a rising edge is detected on TI2, the counter starts counting, and the trigger flag is set (TIMx_STS.TITF=1);

The delay between the rising edge on TI2 and the actual start of the counter is due to the resynchronization circuit on TI2 input.

Figure 10-38 Control Circuit in Trigger Mode



10.5.17.3 Slave mode: Gated mode

In gated control mode, the level polarity of the input port can control whether the counter counts or not.

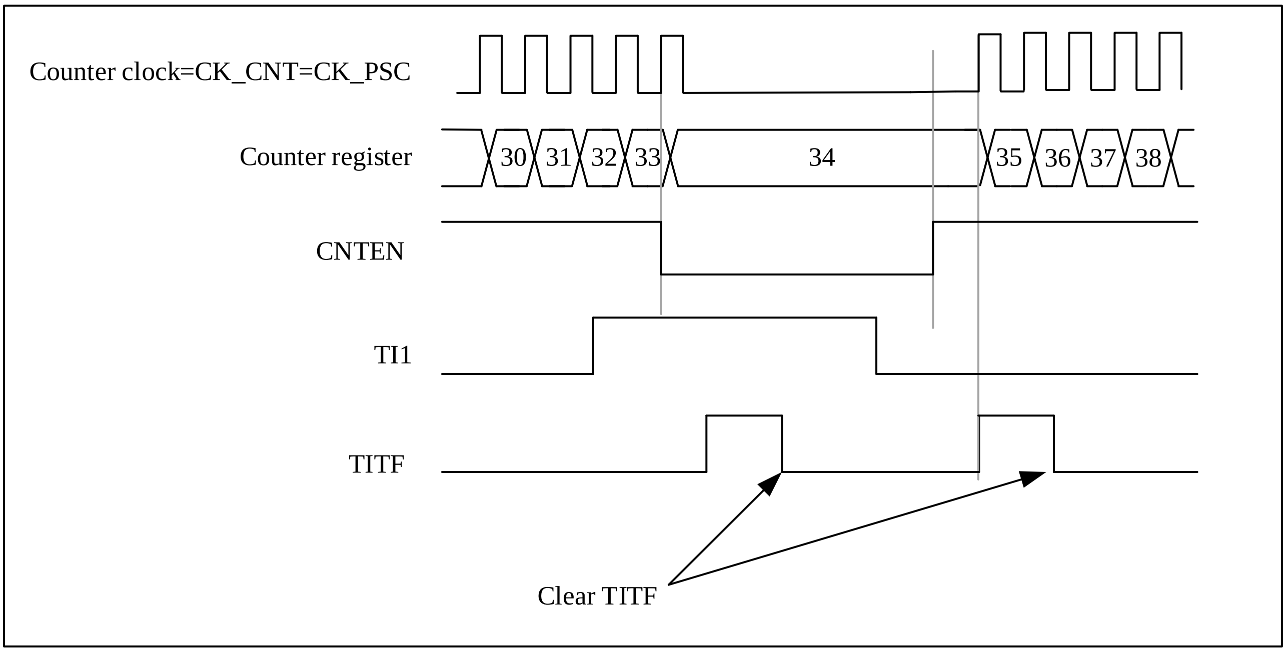
The following is an example of a gated mode:

- Channel 1 is configured as input detection active low on TI1 (TIMx_CCMOD1.CC1SEL=01, TIMx_CCEN.CC1P=1);
- Select the slave mode as the gated mode (TIMx_SMCTRL.SMSEL=0101), and select TI1 as the trigger input (TIMx_SMCTRL.TSEL=101);
- Setting TIMx_CTRL1.CNTEN = 1 to start counter

When TI1 detects that the level changes from low to high, the counter stops counting, and when TI1 detects that the level changes from high to low, the counter starts counting, and the trigger flag will be set (TIMx_STS.TITF=1) when it starts or stops counting;

The delay between the rising edge on TI1 and the actual stop of the counter is due to the resynchronization circuit on TI1 input.

Figure 10-39 Control Circuit in Gated Mode



10.5.17.4 Slave mode: Trigger mode + External clock mode 2

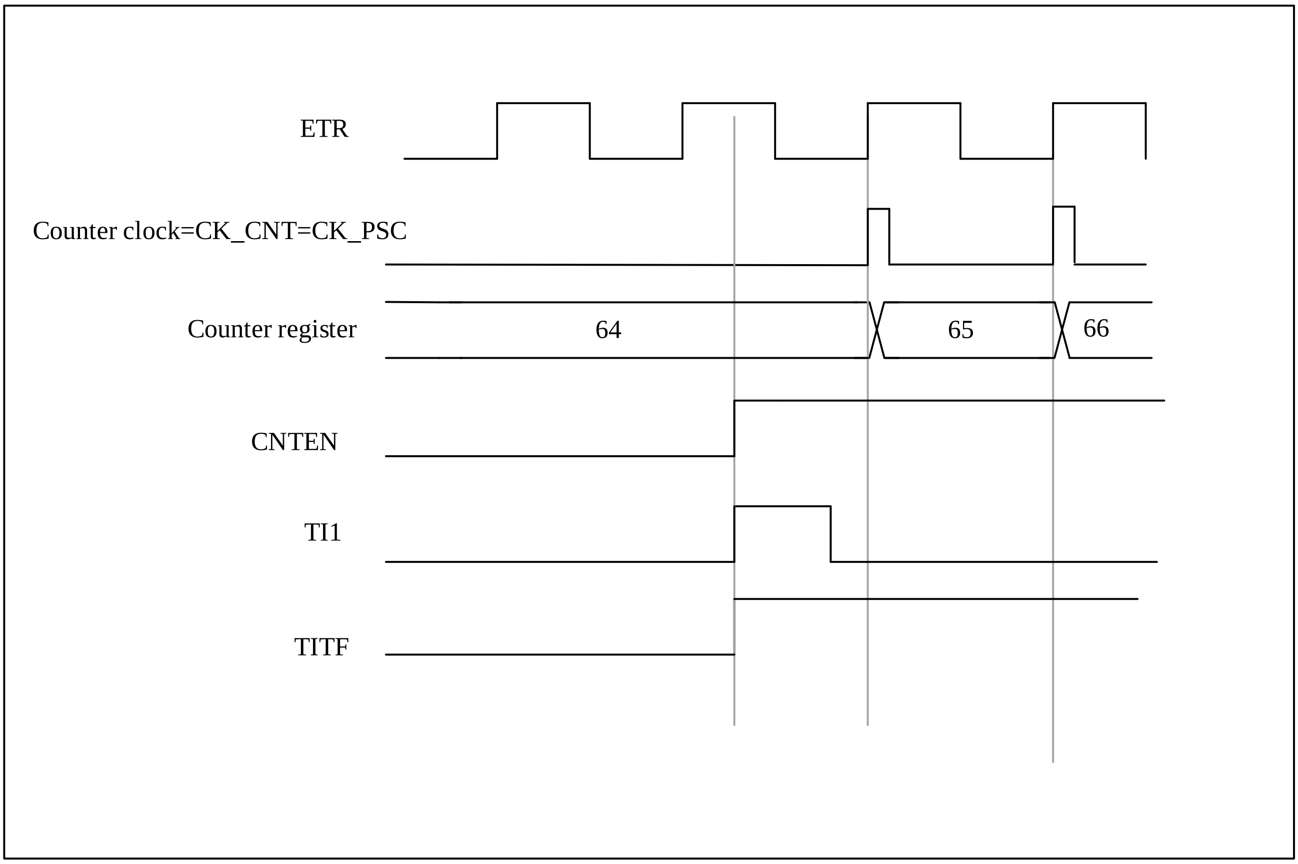
In reset mode, trigger mode and gate control mode, the counter clock can be selected as external clock mode 2, and the ETR signal is used as the external clock source input. At this time, the trigger selection needs to select non-ETRF (TIMx_SMCTRL.TSEL=111).

Here is an example:

- Channel 1 is configured as input to detect the rising edge of TI1 (TIMx_CCMOD1.CC1SEL=01, TIMx_CCEN.CC1P=0),
- Enable external clock mode 2 (TIMx_SMCTRL.EXCEN=1), select rising edge for external trigger polarity (TIMx_SMCTRL.EXTP=0), select slave mode as trigger mode (TIMx_SMCTRL.SMSEL=0110), select TI1 for trigger input (TIMx_SMCTRL.TSEL=101),

When TI1 detects a rising edge, the counter starts counting on the rising edge of ETR, and the trigger flag is set (TIMx_STS.TITF=1).

Figure 10-40 Control Circuit in Trigger Mode + External Clock Mode2



10.5.17.5 Slave mode: Combined Reset + Trigger Mode

In this case, the rising edge of the selected trigger input (TRGI) will reset the counter, generate register updates, and start the counter.

This mode is used for one-pulse mode.

The event selected on the input resets and enables the counter.

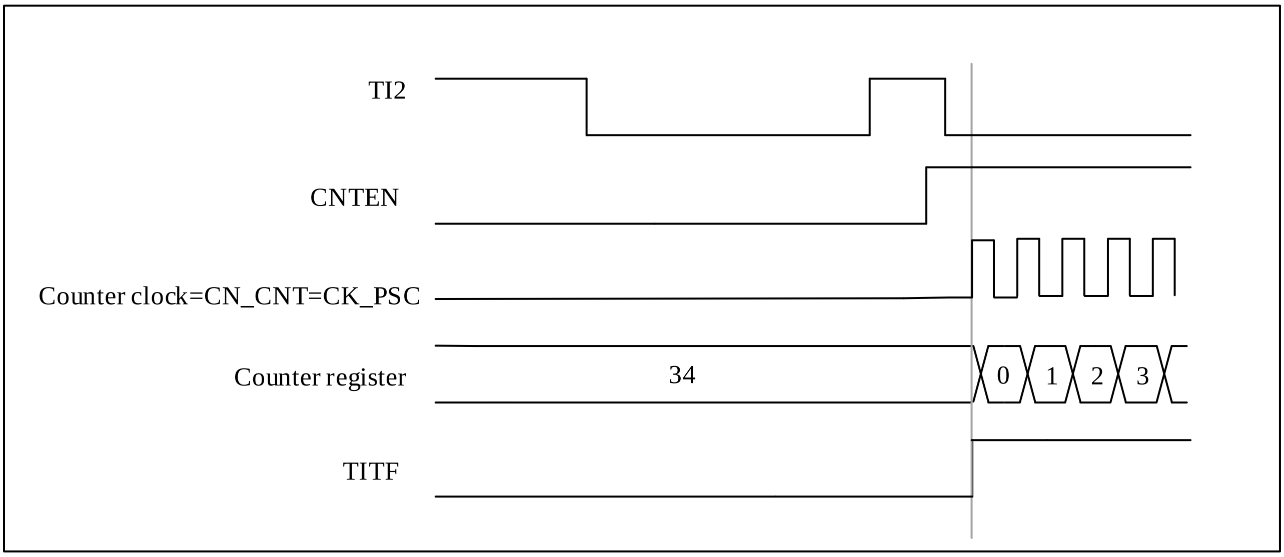
In the following example, the counter is reset and starts counting up on the rising edge of the TI2 input:

- Configure channel 2 to detect the rising edge of TI2. Set the input filter bandwidth (in this example, no filter is needed, keep TIMx_CCMOD1.IC2F=0000). No capture prescaler is used in the trigger operation, so no configuration is required. The TIMx_CCMOD1.CC2SEL bit is used to select the input capture source, set TIMx_CCMOD1.CC2SEL=01. Set TIMx_CCEN.CC2P=1 to determine the polarity (detect only low level).
- Configure TIMx_SMCTRL.SMSEL=1110 to set the timer in Combined Reset + Trigger mode; set TIMx_SMCTRL.TSEL=110 to select TI2 as the input source.

When a rising edge occurs on TI2, the counter starts counting under the internal clock drive, and the TITF flag is set.

The delay between the rising edge on TI2 and the actual stop of the counter is due to the resynchronization circuit on TI2 input.

Figure 10-41 Control Circuit in Combined Reset + Trigger Mode



10.5.17.6 Slave mode: Combined Gated + Reset Mode

When the trigger input (TRGI) is high, the counter is enabled. Once the trigger goes low, the counter stops and is reset. The start and stop of the counter are both controlled.

This mode can detect out-of-range PWM signals (duty cycle exceeding the maximum expected value).

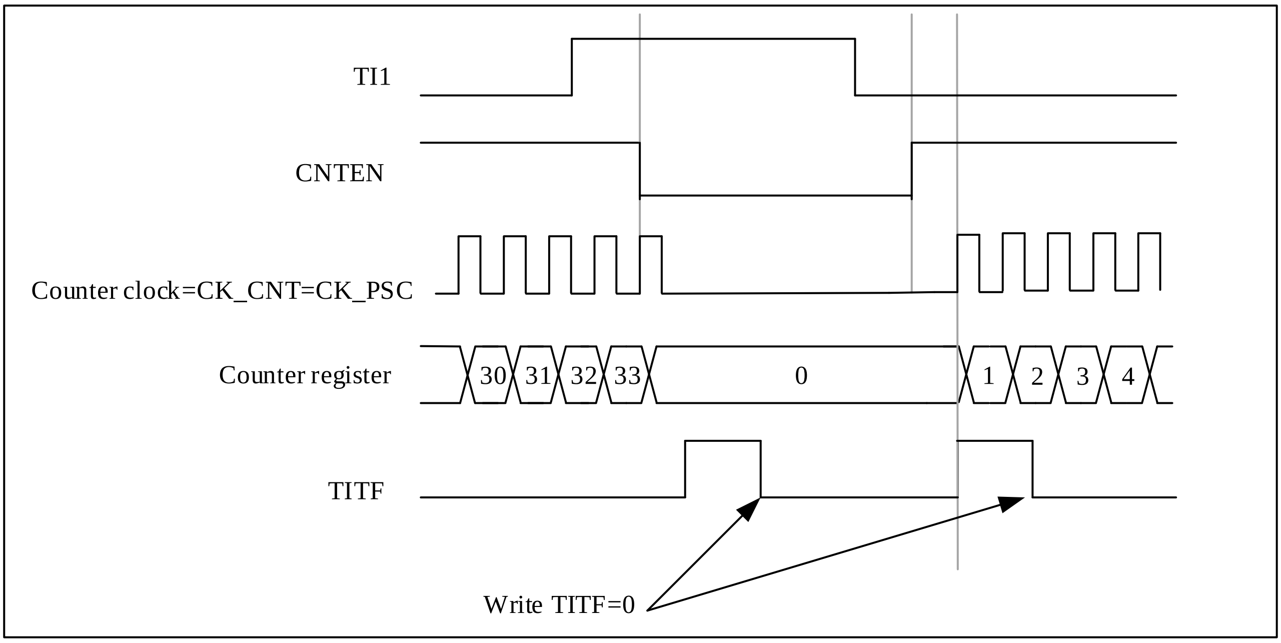
In the following example, the counter counts up only when TI1 is low, and the counter stops and resets when TI1 goes high:

- Configure channel 1 to detect low level on TI1. Set the input filter bandwidth (in this example, no filter is needed, so keep TIMx_CCMOD1.IC1F=0000). No capture prescaler is used in the trigger operation, so no configuration is required. The TIMx_CCMOD1.CC1SEL bit is used to select the input capture source, set TIMx_CCMOD1.CC1SEL=01. Set TIMx_CCEN.CC1P=1 to determine the polarity (detect only low level).
- Configure TIMx_SMCTRL.SMSEL=1101 to set the timer in Gated + Reset mode; set TIMx_SMCTRL.TSEL=101 to select TI1 as the input source.
- Set CNTEN=1 in the TIMx_CTRL1 register to start the counter. In Gated + Reset mode, if CNTEN=0, the counter cannot start, regardless of the trigger input level.

The counter starts counting based on the internal clock as long as TI1 is low, and it stops counting once TI1 goes high. The TITF flag in TIMx_STS is set when the counter starts or stops.

The delay between the rising edge on TI1 and the actual stop of the counter is due to the resynchronization circuit on TI1 input.

Figure 10-42 Control Circuit in Combined Gated + Reset Mode



10.5.18 Timer Synchronization

All TIMx are internally interconnected for timer synchronization or chaining. For more details, refer to Section 12.5.20。

10.5.19 ADC Triggers

The timer can generate ADC trigger events through various internal signals, such as reset, enable, or comparison events, and can also generate pulse triggers issued by internal edge detectors.

users can trigger the ADC by selecting the internal lines TRGO/TRGO2 using the MMSEL2[3:0] and MMSEL[3:0] bits in the TIMx_CTRL2 register.

Channels 4/7/8/9 can also trigger the ADC by configuring TRIG4/TRIG7/TRIG8/TRIG9 to '1', causing channels 4/7/8/9 to generate pulse triggers for the ADC during comparison matches.

10.5.20 Generating Six-step PWM Output

In order to modify the configuration of all channels at the same time, the configuration of the next step can be set in advance (the preloaded bits are OCxMD, CCxEN and CCxNEN). When a COM commutation event occurs, the OCxMD, CCxEN, and CCxNEN preload bits are transferred to the shadow register bits.

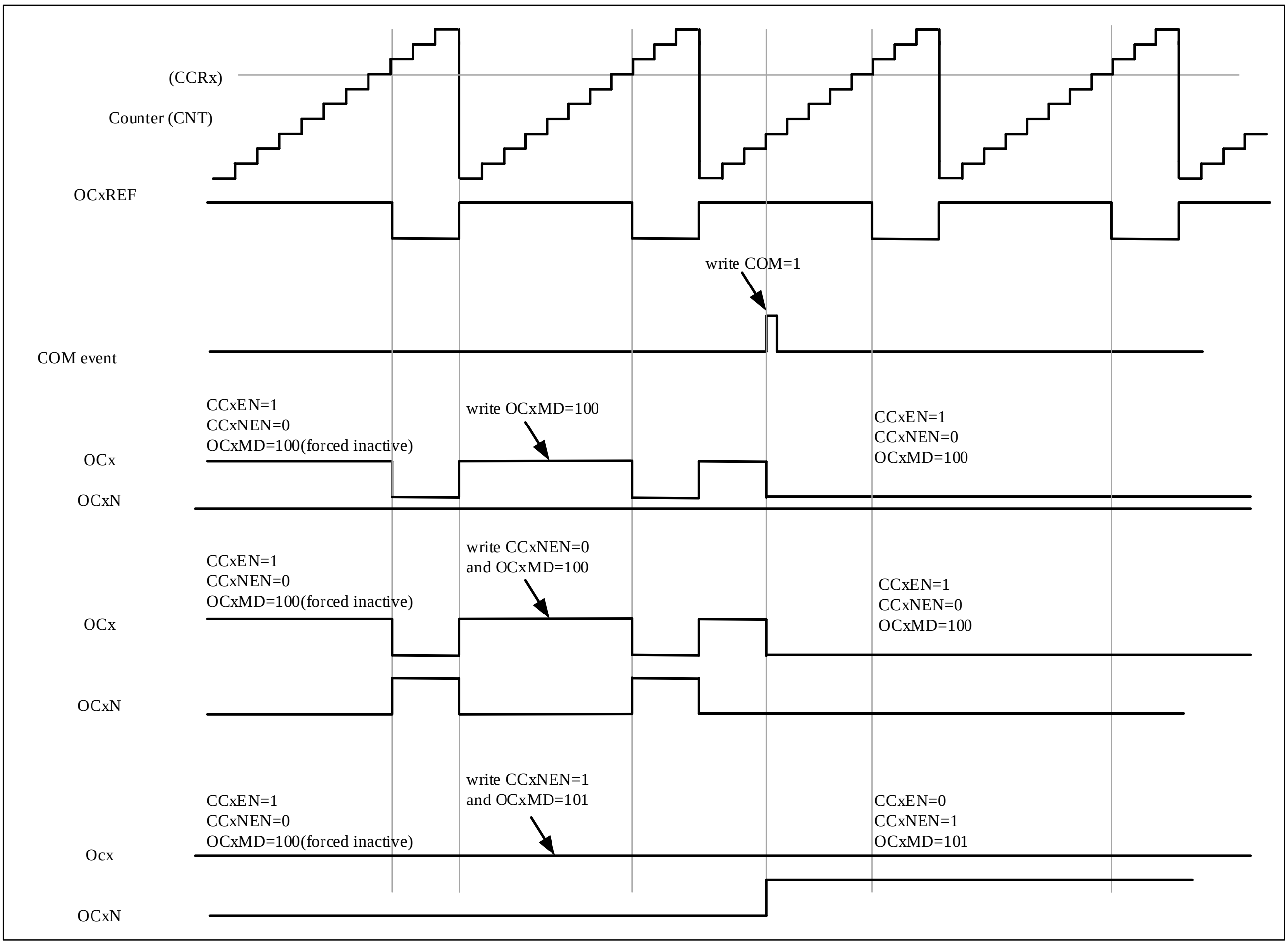
Methods to generate a COM commutation event:

- A software sets TIMx_EVTGEN.CCUDGN;
- Generated by hardware on the rising edge of TRGI;

When a COM commutation event occurs, the TIMx_STS.COMITF flag will be set, enabling interrupts (TIMx_DINTEN.COMIEN) will generate interrupts, and enabling DMA requests (TIMx_DINTEN.COMDEN) will generate DMA requests.

The following figure shows the output timing diagram of OCx and OCxN when a COM commutation event occurs in three different configurations:

Figure 10-43 6-Step PWM Generation, COM Example (OSSR=1)



10.5.21 Encoder Interface Mode

10.5.21.1 Quadrature Encoder Mode

The encoder uses two inputs, TI1 and TI2 as the interface and the counter counts on every edge change on TI1FP1 or TI2FP2. The counting direction is automatically controlled by hardware TIMx_CTRL1.DIR. There are five types of quadrature encoder counting modes:

- Encoder mode 1: The counter counts only on the edges of TI1, TIMx_SMCTRL.SMSEL = '0001';
- Encoder mode 2: The counter counts only on the edges of TI2, TIMx_SMCTRL.SMSEL = '0010';
- Encoder mode 3: The counter counts on the edges of both TI1 and TI2, TIMx_SMCTRL.SMSEL = '0011';
- Encoder mode 4: The counter counts only on the edges of TI1 when T2 is high, TIMx_SMCTRL.SMSEL = '1001';
- Encoder mode 5: The counter counts only on the edges of TI2 when T1 is high, TIMx_SMCTRL.SMSEL =

'1010';

The encoder interface is equivalent to using an external clock with direction selection, and the counter only counts continuously between 0 and the auto-reload value (TIMx_AR.AR [15:0]). Therefore, it is necessary to configure the auto-reload register TIMx_AR in advance.

Note: encoder mode and external clock mode 2 are not compatible and must not be selected together.

The relationship between the counting direction and the encoder signal is as follows:

Table 10-14 The Relationship between the Counting Direction and the Encoder Signal（CC1P=CC2P=0）

Active Edge	**SMSEL[3:0]**	**Level on Opposite Signals (TI1FP1 for TI2, TI2FP2 for TI1)**	**TI1FP1 Signal**		**TI2FP2 Signal**	
			Rising	**Falling**	**Rising**	**Falling**
Counting only on TI1	0001	Hign	Counting down	Counting up	Don't count	Don't count
		Low	Counting up	Counting down	Don't count	Don't count
Counting only on TI2	0010	Hign	Don't count	Don't count	Counting up	Counting down
		Low	Don't count	Don't count	Counting down	Counting up
Counting on Both TI1 and TI2	0011	Hign	Counting down	Counting up	Counting up	Counting down
		Low	Counting up	Counting down	Counting down	Counting up
Counting only on TI1 when T2 is high	1001	Hign	Counting down	Counting up	Don't count	Don't count
		Low	Don't count	Don't count	Don't count	Don't count
Counting only on TI2 when T1 is high	1010	Hign	Don't count	Don't count	Counting up	Counting down
		Low	Don't count	Don't count	Don't count	Don't count

The change in the counter value in each encoder mode is as follows:

Figure 10-44 The Encoder Counts Only on TI1

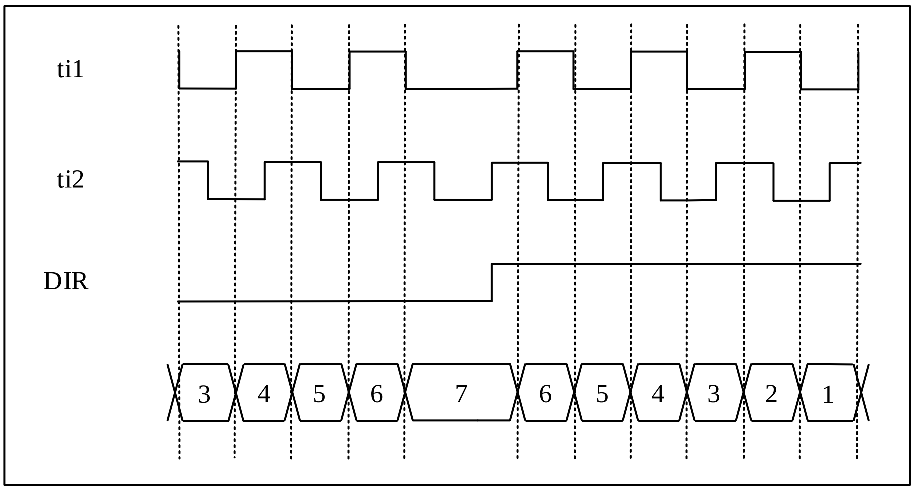


Figure 10-45 The Encoder Counts Only on TI2

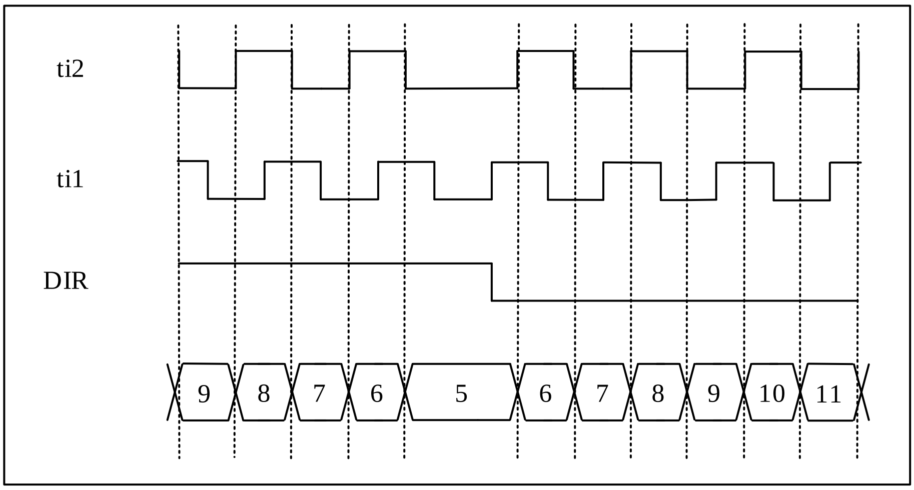


Figure 10-46 The Encoder Counts on Both TI1 and TI2

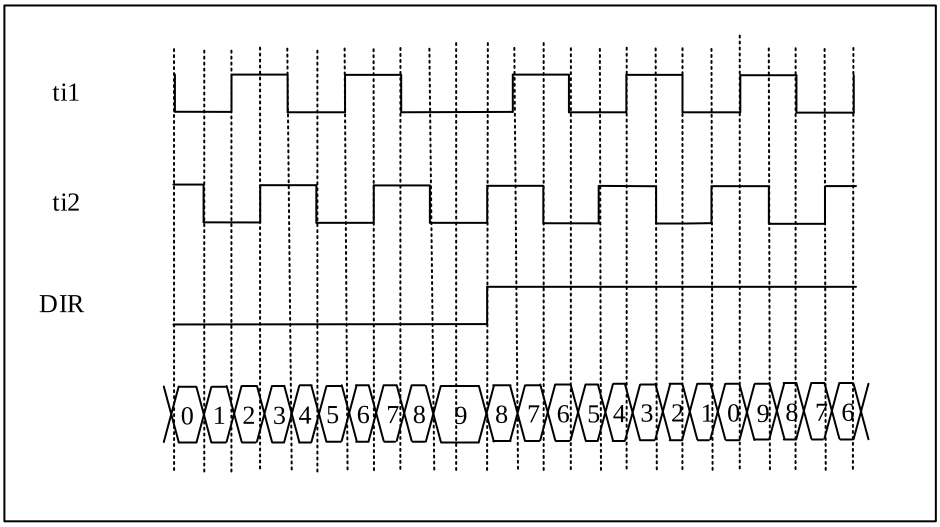


Figure 10-47 The Counter Counts Only on TI1 When T2 is High

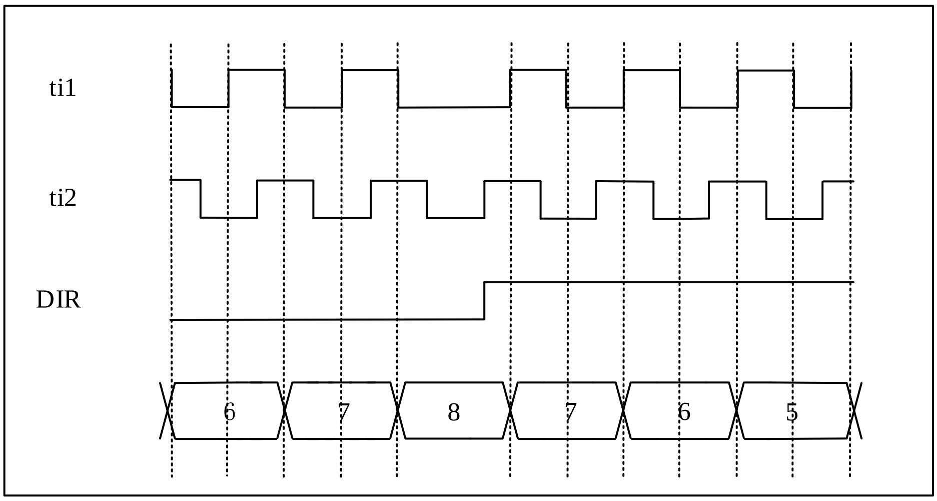
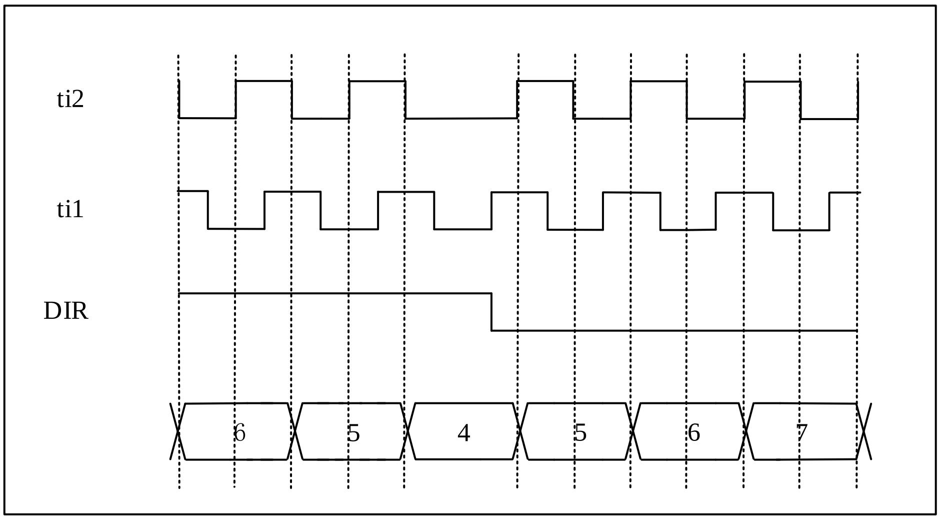


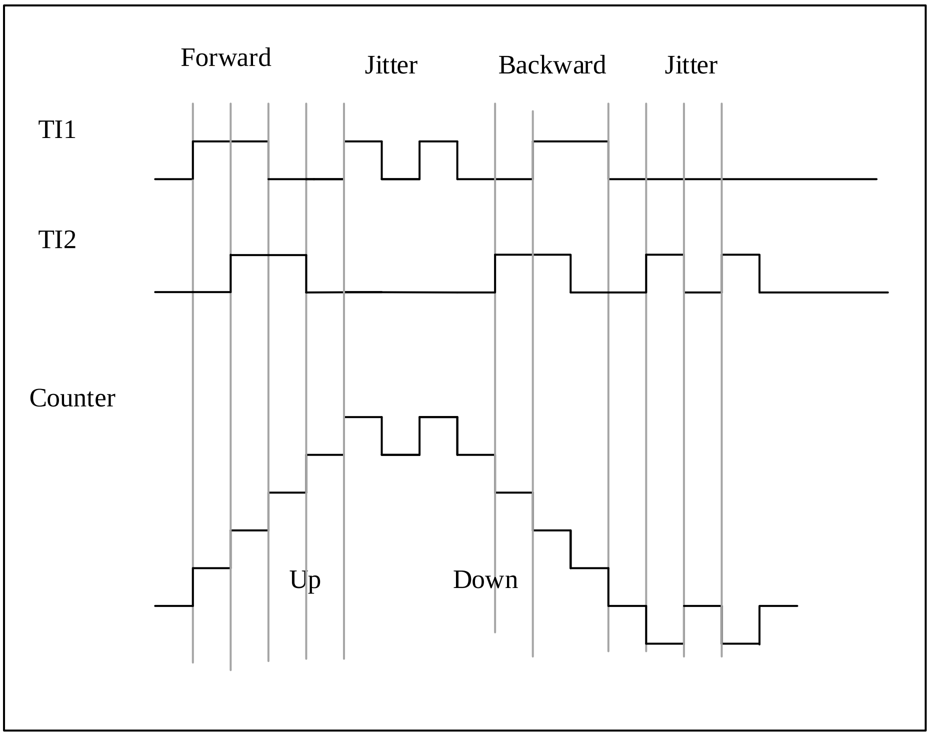
Figure 10-48 The Counter Counts Only on TI2 When TI1 is High



Here is an example of an encoder with dual edge selected for triggering to suppress input jitter:

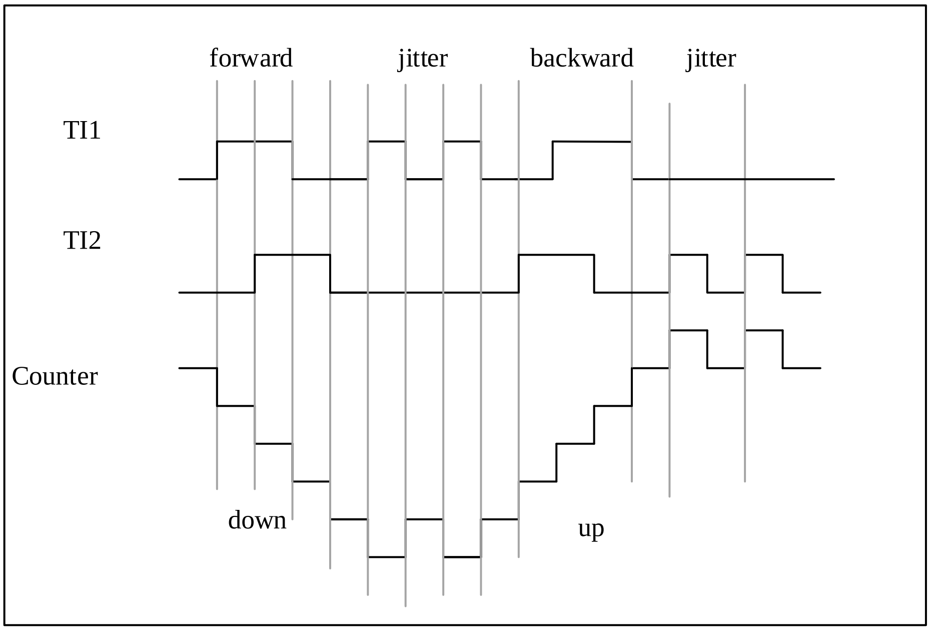
- IC1FP1 is mapped to TI1 (TIMx_CCMOD1.CC1SEL= ‘01’), IC1FP1 is not inverted (TIMx_CCEN.CC1P= ‘0’);
- IC1FP2 is mapped to TI2 (TIMx_CCMOD2.CC2SEL= ‘01’), IC2FP2 is not inverted (TIMx_CCEN.CC2P= ‘0’);
- The input is valid on both rising and falling edges (TIMx_SMCTRL.SMSEL = ‘011’);
- Enable counter TIMx_CTRL1.CNTEN= ‘1’.

Figure 10-49 Example of Counter Operation in Encoder Interface Mode



The following figure shows the example of counter behavior when IC1FP1 polarity is inverted (CC1P= '1', other configurations are the same as above).

Figure 10-50 Encoder Interface Mode Example with IC1FP1 Polarity Inverted



10.5.21.2 Pulse-Level Encoder Mode

In the Pulse-Level encoder mode, the clock is provided on a single line on TI2, while the counting direction is provided by the TI1 input.

This mode is enabled through the SMSEL[3:0] in the TIMx_SMCTRL register, as follows.

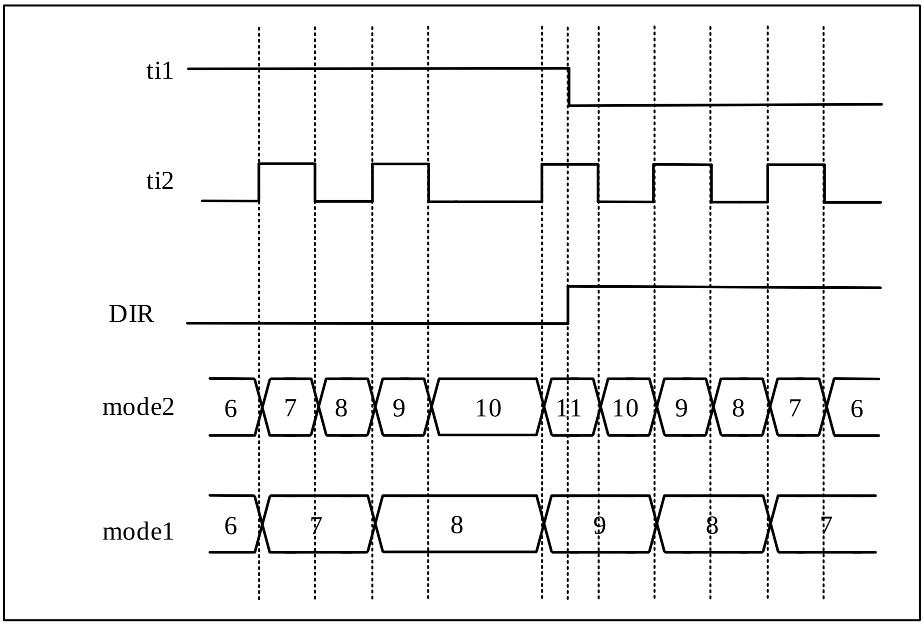
1011：Pulse-Level encoder mode 2, the counter is updated on both the rising and falling edges of the clock.

1100：Pulse-Level encoder mode 1, the counter is updated on a single clock edge based on the value of CC2P. CC2P = 0 corresponds to counting on the rising edge, and CC2P = 1 corresponds to counting on the falling edge.

The polarity of the direction signal for TI1 is set by the CC1P bit. When CC2P = 0, counting up occurs when TI1 is high and counting down occurs when TI1 is low; when CC1P = 1, counting up occurs when TI1 is low and counting down occurs when TI1 is high.

The following diagram illustrates the case where CC1P=CC2P=0:

Figure 10-51 Pulse-Level Encoder Mode（CC1P=CC2P=0）



10.5.21.3 Dual-Pulse Encoder Mode

In the Dual-Pulse encoder mode, the clock is provided on two lines, with only one line provided at a time based on the direction. This results in one clock line for counting up and another clock line for counting down.

This mode is enabled through the SMSEL[3:0] field in the TIMx_SMCTRL register, as follows.

- 1000：Dual-Pulse encoder mode 2, The counter is updated on both the rising and falling edges of any of the two clock line. The CC1P and CC2P bits encode the idle state of the clock. CCxP=0 corresponds to a high-level idle state, while CCxP=1 corresponds to a low-level idle state.
- 1111：Dual-Pulse encoder mode 1, the counter is updated on a single clock edge based on the values of the CC1P and CC2P bits. CCxP = 0 corresponds to the falling edge and high-level state, while CCxP = 1 corresponds to the rising edge and low-level state.

The following table describes the relationship between counting direction, encoder signals, and polarity settings:

Table 10-15 The Relationship between Counting Direction, Encoder Signals, and Polarity Settings

Dual-Pulse Encoder Mode	SMSEL [3:0]	Level on Opposite Signal (TI1FP1 for TI2, TI2FP2 for TI1)	TI1FP1 Signal		TI2FP2 Signal	
			Rising	Falling	Rising	Falling
Mode 2 CCxP=0	1000	High	Counting down	Counting down	Counting up	Counting up
		Low	Don't count	Don't count	Don't count	Don't count
Mode 2 CCxP=1	1000	High	Don't count	Don't count	Don't count	Don't count
		Low	Counting down	Counting down	Counting up	Counting up
Mode 1 CCxP=0	1111	High	Don't count	Counting down	Don't count	Counting up
		Low	Don't count	Don't count	Don't count	Don't count
Mode 1 CCxP=1	1111	High	Don't count	Don't count	Don't count	Don't count
		Low	Counting down	Don't count	Counting up	Don't count

The following diagram illustrates the counting method of the counter in Dual-Pulse encoder mode:

Figure 10-52 Dual-Pulse Encoder Mode (CC1P = CC2P = 0)

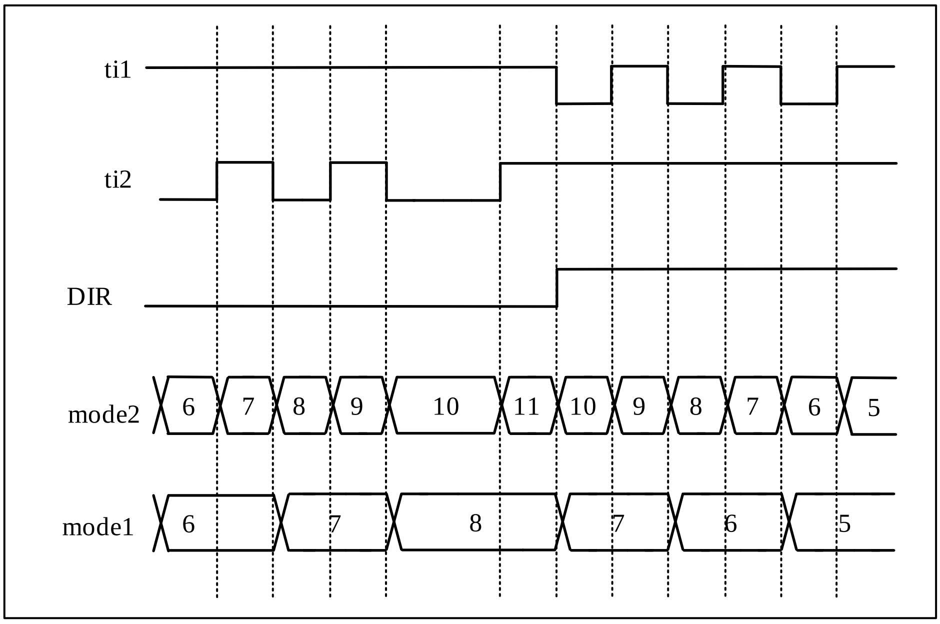
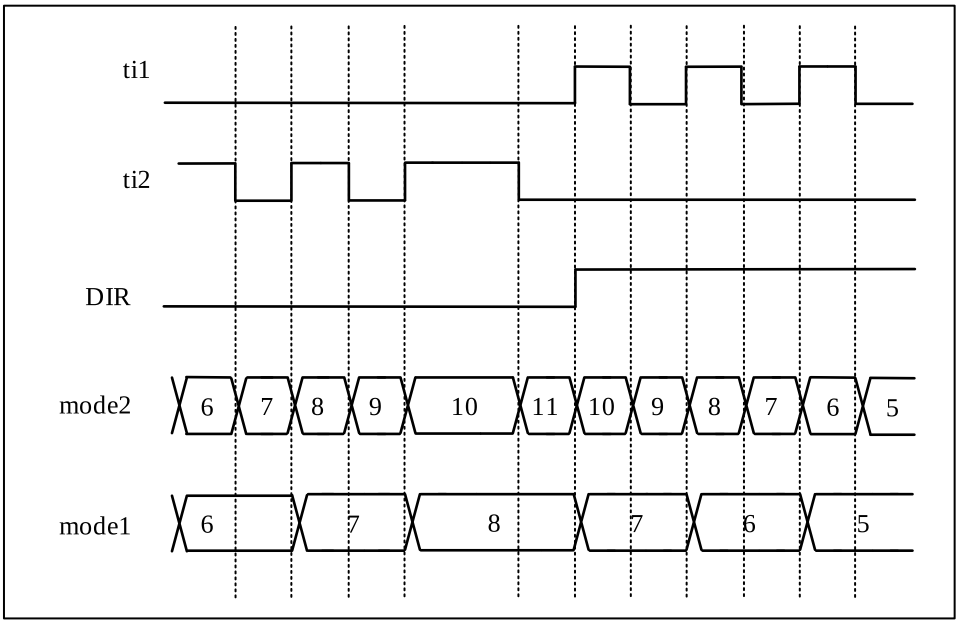


Figure 10-53 Dual-Pulse Encoder Mode (CC1P = CC2P = 1)



10.5.22 Interfacing with Hall Sensor

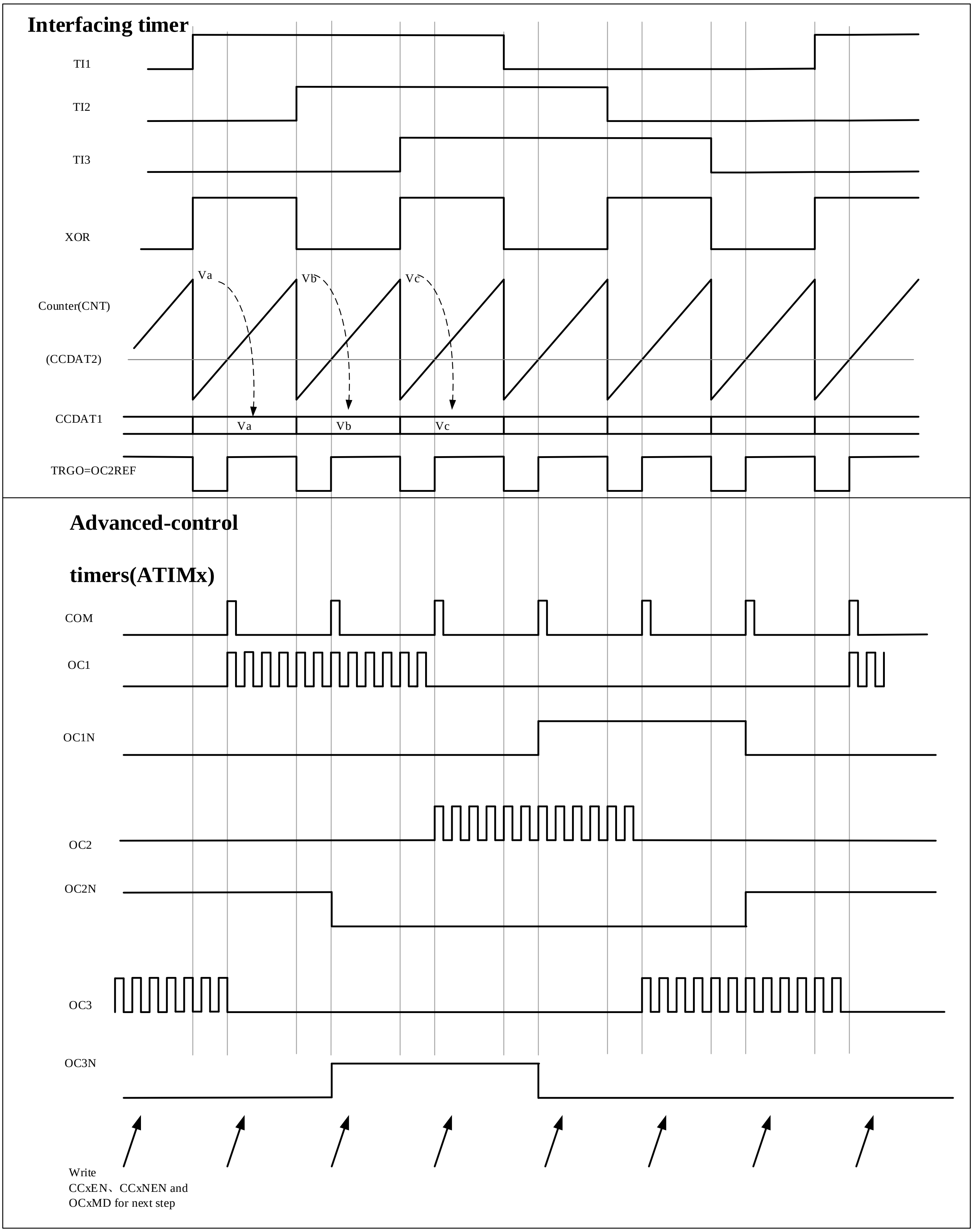
Connect the Hall sensor to the three input pins (CC1, CC2 and CC3) of the timer, and then select the XOR function to pass the inputs of TIMx_CH1, TIMx_CH2 and TIMx_CH3 through the XOR gate as the output of TI1 to channel 1 for capture signal.

The timer needs to be configured as the reset mode in slave mode (TIMx_SMCTRL.SMSEL= ‘0100’); the edge of the trigger select TI1 triggers TI1F_ED (TIMx_SMCTRL.TSEL= ‘100’), any change in the Hall 3 inputs will trigger the counter to recount, so it is used as a time reference; the capture/compare channel 1 is configured to capture the TRC signal in capture mode (TIMx_CCMOD1.CC1SEL= ‘11’), which is used to calculate the two input time intervals, thereby reflecting the motor speed.

Select timer channel 2 to output pulses to the advanced timer to trigger the COM event of the advanced timer to update the control bits of the output PWM. The trigger selection of the advanced timer needs to select the corresponding internal trigger signal (TIMx_SMCTRL.TSEL="ITRx"), the capture/compare preload control bit needs to be configured to support preload (TIMx_CTRL2.CCPCTL=1) and support the rising edge of TRGI Trigger an update (TIMx_CTRL2.CCUSEL=1).

This example is shown in the following figure.

Figure 10-54 Example of Hall Sensor Interface



10.6 ATIMx Register Description

For abbreviations used in registers, refer to Section 1.1.

These peripheral registers can be operated as half word (16-bits) or one word (32-bits).

10.6.1 Control Register 1（TIMx_CTRL1）

Offset address: 0x00

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								ASYMMETRIC	Reserved	CMODE[1:0]		Reserved			
								rw		rw					

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SRAMECCERREN	FLASHECCERREN	CLRSEL	SRAMPARERREN	PBKPEN	LBKPEN	ARPEN	ONEPM	CLKD[1:0]		UPDIS	UPRS	CAMSEL[1:0]		DIR	CNTEN
w	w	rw	w	w	w	rw	rw	rw		rw	rw	rw		rw	rw

Bit Field	Name	Description
31:24	Reserved	Reserved, the reset value must be maintained
23	ASYMMETRIC	Asymmetric mode enable in center-aligned mode 0：Disabled 1：Enabled (valid when TIMx_CTRL1.CAMSEL[1:0] is non-zero, each channel will compare to CCDATx when counting up, and compare to CCDDATx when counting down)
22	Reserved	Reserved, the reset value must be maintained
21:20	CMODE	In center-aligned asymmetric mode, channel 4/7/8/9 trigger mode, only when TIMx_CTRL2.MMSEL3 = '1xxx', the TRGO output will be valid. 00：Up-counting to CCDAT4/7/8/9, trigger valid 01：Down-counting, channel 4 count to CCDDAT4, channel 7/8/9 count to CCDAT7/8/9, trigger valid 1x：Channel 4 up-counting to CCDAT4 or down-counting to CCDDAT4, channel 7/8/9 up-counting or down-counting to CCDAT7/8/9, trigger valid In center-aligned symmetry mode, channel 4/7/8/9 trigger mode, only when TIMx_CTRL2.MMSEL3 = '1xxx', the TRGO output will be valid. 00: Up-counting to CCDAT4/7/8/9, trigger valid 01: Down-counting to CCDAT4/7/8/9, trigger valid 1x: Up-counting or down-counting to CCDAT4/7/8/9, trigger valid
19:16	Reserved	Reserved, the reset value must be maintained
15	SMECCERREN	SRAM ECC error break Enable 0: Disable 1: Enable

Bit Field	Name	Description
		Note: System reset and power-on reset can clear this bit to 0, but timer reset cannot clear this bit to 0
14	FLECCERREN	FLASH ECC error break Enable 0: Disable 1: Enable *Note: System reset and power-on reset can clear this bit to 0, but timer reset cannot clear this bit to 0*
13	CLRSEL	OcxRef clear selection 0：Select the external Ocxclr (TIMx_ETR) signal，Specific selection can refer to TIMx_INSEL.ETRS register 1：Select the internal Ocxclr (tim_ocref_clr) signal，Specific selection can refer to TIMx_INSEL.CLRS register
12	SMPARERREN	SRAM parity error break Enable 0: Disable 1: Enable *Note: System reset and power-on reset can clear this bit to 0, but timer reset cannot clear this bit to 0*
11	PBKPEN	PVD break Enable 0: Disable 1: Enable *Note: System reset and power-on reset can clear this bit to 0, but timer reset cannot clear this bit to 0*
10	LBKPEN	LockUp break Enable（Core Hardfault） 0: Disable 1: Enable *Note: System reset and power-on reset can clear this bit to 0, but timer reset cannot clear this bit to 0*
9	ARPEN	Auto-reload preload enable 0: Shadow register disable for TIMx_AR register 1: Shadow register enable for TIMx_AR register
8	ONEPM	One pulse mode 0: Disable one-pulse mode, the counter counts are not affected when an update event occurs. 1: Enable one-pulse mode, the counter stops counting when the next update event occurs
7:6	CLKD[1:0]	Clock division CLKD[1:0] indicates the division ratio between CK_INT (timer clock) and DTS (clock used for dead-time generator and digital filters (ETR, TIx)) 00: $t_{DTS} = t_{CK\_INT}$. 01: $t_{DTS} = 2 \times t_{CK\_INT}$ 10: $t_{DTS} = 4 \times t_{CK\_INT}$ 11: Reserved, do not use this configuration

Bit Field	Name	Description
5	UPDIS	Update disable This bit is set and cleared by software to enable/disable UEV event generation. 0: UEV Enable. UEV will be generated if one of following condition been fulfilled: – Counter overflow/underflow – The TIMx_EVTGEN.UDGN bit is set – Update generation from the slave mode controller Shadow registers will update with preload value. 1: UEV disabled. No update event is generated, and the shadow registers (AR, PSC, and CCDATx) keep their values. If the TIMx_EVTGEN.UDGN bit is set or a hardware reset is issued by the slave mode controller, the counter and prescaler are reinitialized.
4	UPRS	Update request source This bit is used to select the UEV event sources by software. 0: If update interrupt or DMA request is enabled, any of the following events will generate an update interrupt or DMA request: – Counter overflow/underflow – The TIMx_EVTGEN.UDGN bit is set – Update generation from the slave mode controller 1: If update interrupt or DMA request is enabled, only counter overflow/underflow will generate update interrupt or DMA request
3:2	CAMSEL[1:0]	Center-aligned mode selection 00: Edge-aligned mode. TIMx_CTRL1.DIR specifies up-counting or down-counting. 01: Center-aligned mode 1. The counter counts in center-aligned mode, and the output compare interrupt flag bit is set to 1 when down-counting. 10: Center-aligned mode 2. The counter counts in center-aligned mode, and the output compare interrupt flag bit is set to 1 when up-counting. 11: Center-aligned mode 3. The counter counts in center-aligned mode, and the output compare interrupt flag bit is set to 1 when up-counting or down-counting. *Note: switching from edge-aligned mode to center-aligned mode is not allowed when the counter is still enabled (TIMx_CTRL1.CNTEN = 1).*
1	DIR	Direction 0: Up-counting 1: Down-counting *Note: this bit is read-only when the counter is configured in center-aligned mode or encoder mode.*
0	CNTEN	Counter Enable 0: Disable counter 1: Enable counter *Note: external clock, gating mode and encoder mode can only work after TIMx_CTRL1.CNTEN bit is set in the software. Trigger mode can automatically set TIMx_CTRL1.CNTEN bit by hardware.*

10.6.2 Control Register 2（TIMx_CTRL2）

Offset address: 0x04

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				MMSEL2[3:0]				TRIG9	TRIG8	TRIG7	TRIG4	TI1SEL	CCPCTL	CCDSEL	CCUSEL
				rw				rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MMSEL[3:0]				Reserved	OI6	Reserved	OI5	OI4N	OI4	OI3N	OI3	OI2N	OI2	OI1N	OI1
rw					rw		rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit Field	Name	Description
31:28	Reserved	Reserved, the reset value must be maintained
27:24	MMSEL2[3:0]	Master mode selection 2 These 4 bits are used to select the synchronization information (TRGO2) sent to the slave timer in the master mode. Possible combinations are as follows: 0000: Reset –When the TIMx_EVTGEN.UDGN is set or a reset is generated by the slave mode controller, a TRGO2 pulse occurs. And in the latter case, the signal on TRGO is delayed compared to the actual reset. 0001: Enable - The TIMx_CTRL1.CNTEN bit is used as the trigger output (TRGO2). Sometimes you need to start multiple timers at the same time or enable slave timer for a period of time. The counter enable signal is set when TIMx_CTRL1.CNTEN bit is set or the trigger input in gated mode is high. When the counter enable signal is controlled by the trigger input, there is a delay on TRGO2 except if the master/slave mode is selected (refer to the description of the TIMx_SMCTRL.MSMD bit). 0010: Update - The update event is selected as the trigger output (TRGO2). For example, a master timer clock can be used as a slave timer prescaler. 0011: Compare pulse - Triggers the output to send a positive pulse (TRGO2) when the TIMx_STS.CC1ITF is to be set (even if it is already high), when a capture or a comparison succeeds. 0100: Compare - OC1REF signal is used as the trigger output (TRGO2) 0101: Compare – OC2REF signal is used as the trigger output (TRGO2) 0110: Compare – OC3REF signal is used as the trigger output (TRGO2) 0111: Compare – OC4REF signal is used as the trigger output (TRGO2) 1000: Compare – OC5REF signal is used as the trigger output (TRGO2) 1001: Compare – OC6REF signal is used as the trigger output (TRGO2) 1010: Compare Pulse - OC4REF rising or falling edges generate a positive pulses on TRGO2 1011: Compare Pulse – OC6REF rising or falling edges generate a positive pulses on TRGO2 1100: Compare Pulse - OC4REF rising edges or OC6REF rising edges generate a positive pulses on TRGO2

Bit Field	Name	Description
		1101: Compare Pulse - OC4REF rising edges or OC6REF falling edges generate a positive pulses on TRGO2 1110: Compare Pulse – OC5REF rising edges or OC6REF rising edges generate a positive pulses on TRGO2 1111: Compare Pulse – OC5REF rising edges or OC6REF falling edges generate a positive pulses on TRGO2
23	TRIG9	Enable triggering ADC when channel 9 comparison matches 0: Trigger disable 1: Trigger enable
22	TRIG8	Enable triggering ADC when channel 8 comparison matches 0: Trigger disable 1: Trigger enable
21	TRIG7	Enable triggering ADC when channel 7 comparison matches 0: Trigger disable 1: Trigger enable
20	TRIG4	Enable triggering ADC when channel 4 comparison matches 0: Trigger disable 1: Trigger enable
19	TI1SEL	TI1 selection 0: TIMx_CH1 pin connected to TI1 input. 1: TIMx_CH1, TIMx_CH2, and TIMx_CH3 pins are XOR connected to the TI1 input.
18	CCPCTL	Capture/compare preloaded control 0: CCxEN, CCxNEN and OCxMD bits are not preloaded. 1: CCxEN, CCxNEN and OCxMD bits are preloaded. they are updated only when a commutation event COM occurs (CCUDGN bit set or rising edge on TRGI depending on CCUSEL bit) *Note: this bit only applied to channels with complementary outputs.*
17	CCDSEL	Capture/compare DMA selection 0: When a CCx event occurs, a DMA request for CCx is sent. 1: When an update event occurs, a DMA request for CCx is sent.
16	CCUSEL	Capture/compare control update selection 0: If TIMx_CTRL2.CCPCTL = 1, they can only be updated by setting CCUDGN bit 1: If TIMx_CTRL2.CCPCTL = 1, they can be updated by setting CCUDGN bit or a rising edge on TRGI. *Note: this bit only applied to channels with complementary outputs.*
15:12	MMSEL[3:0]	Master Mode Selection These 4 bits are used to select the synchronization information (TRGO) sent to the slave timer in the master mode. Possible combinations are as follows: 0000: Reset –When the TIMx_EVTGEN.UDGN is set or a reset is generated by the slave mode controller, a TRGO pulse occurs. And in the latter case, the signal on TRGO is delayed compared to the actual reset. 0001: Enable - The TIMx_CTRL1.CNTEN bit is used as the trigger output (TRGO). Sometimes you need to start multiple timers at the same time or enable slave timer for a period of time.

Bit Field	Name	Description
		The counter enable signal is set when TIMx_CTRL1.CNTEN bit is set or the trigger input in gated mode is high. When the counter enable signal is controlled by the trigger input, there is a delay on TRGO except if the master/slave mode is selected (refer to the description of the TIMx_SMCTRL.MSMD bit). 0010: Update - The update event is selected as the trigger output (TRGO). For example, a master timer clock can be used as a slave timer prescaler. 0011: Compare pulse - Triggers the output to send a positive pulse (TRGO) when the TIMx_STS.CC1ITF is to be set (even if it is already high), when a capture or a comparison succeeds. 0100: Compare - OC1REF signal is used as the trigger output (TRGO). 0101: Compare - OC2REF signal is used as the trigger output (TRGO). 0110: Compare - OC3REF signal is used as the trigger output (TRGO). 0111: Compare - OC4REF signal is used as the trigger output (TRGO). 1xxx: Compare-If the counter is center-aligned mode: The corresponding edge signal of OC4REF/OC7REF/OC8REF/OC9REF used as the trigger output (TRGO), up counting and down counting are configurable, refer specifically to the TIMx_CTRL1.CMODE. If the counter is edge alignment mode: The OC4REF signal is used as the trigger output (TRGO).
11	Reserved	Reserved, the reset value must be maintained
10	OI6	Output idle state 6 (OC6 output). Refer to TIMx_CTRL2.OI1 bit.
9	Reserved	Reserved, the reset value must be maintained
8	OI5	Output idle state 5 (OC5 output). Refer to TIMx_CTRL2.OI1 bit.
7	OI4N	Output idle state 4 (OC4N output). Refer to TIMx_CTRL2.OI1N bit.
6	OI4	Output idle state 4 (OC4 output). Refer to TIMx_CTRL2.OI1 bit.
5	OI3N	Output idle state 3 (OC3N output). Refer to TIMx_CTRL2.OI1N bit.
4	OI3	Output idle state 3 (OC3 output). Refer to TIMx_CTRL2.OI1 bit.
3	OI2N	Output idle state 2 (OC2N output). Refer to TIMx_CTRL2.OI1N bit.
2	OI2	Output idle state 2 (OC2 output). Refer to TIMx_CTRL2.OI1 bit.
1	OI1N	Output Idle state 1 (OC1N Output) 0: When TIMx_BKDT.MOEN = 0, after dead-time OC1N = 0 1: When TIMx_BKDT.MOEN = 0, after dead-time OC1N = 1 *Note: Once TIMx_BKDT.LCKCFG level 1, 2, or 3 has been set, this bit cannot be modified.*
0	OI1	Output Idle state 1 0: When TIMx_BKDT.MOEN = 0, if OC1N is implemented, after dead-time OC1 = 0 1: When TIMx_BKDT.MOEN = 0, if OC1N is implemented, after dead-time OC1 = 1 *Note: Once TIMx_BKDT.LCKCFG level 1, 2, or 3 has been set, this bit cannot be modified.*

10.6.3 Status Registers（TIMx_STS）

Offset address: 0x08

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved					CC9ITF	CC8ITF	CC7ITF	Reserved		SBITF	BITF2	BITF	TITF	COMITF	UDITF
					rc_w0	rc_w0	rc_w0			rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				CC4OCF	CC3OCF	CC2OCF	CC1OCF	Reserved		CC6ITF	CC5ITF	CC4ITF	CC3ITF	CC2ITF	CC1ITF
				rc_w0	rc_w0	rc_w0	rc_w0			rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0

Bit Field	Name	Description
31:27	Reserved	Reserved, the reset value must be maintained
26	CC9ITF	Capture/Compare 9 interrupt flag Refer to TIMx_STS.CC1ITF description.
25	CC8ITF	Capture/Compare 8 interrupt flag Refer to TIMx_STS.CC1ITF description.
24	CC7ITF	Capture/Compare 7 interrupt flag Refer to TIMx_STS.CC1ITF description.
23:22	Reserved	Reserved, the reset value must be maintained
21	SBITF	System break interrupt flag This bit is set by hardware once the system break input is active. This bit is cleared by software when the system break input becomes inactive. 0: No system break event occurred 1: An active level has been detected on system break input
20	BITF2	Break2 interrupt flag This bit is set by hardware once the break2 input is active. This bit is cleared by software when the break2 input becomes inactive. 0: No break2 event occurred 1: An active level has been detected on break2 input
19	BITF	Break1 interrupt flag This bit is set by hardware once the break1 input is active. This bit is cleared by software when the break input becomes inactive. 0: No break1 event occurred 1: An active level has been detected on break1 input
18	TITF	Trigger interrupt flag When a trigger event occurs (when an effective edge is detected at the TRGI input, except in gated mode, or any edge in gated mode) the hardware sets this bit to '1'. It is cleared by software to '0'. 0: No trigger event occurred

Bit Field	Name	Description
		1: Trigger interrupt pending
17	COMITF	COM interrupt flag This bit is set by hardware once a COM event is generated (when TIMx_CCEN.CCxEN, TIMx_CCEN.CCxNEN, TIMx_CCMOD1.OCxMD have been updated). This bit is cleared by software. 0: No COM event occurred 1: COM interrupt pending
16	UDITF	Update interrupt flag This bit is set by hardware when an update event occurs under the following conditions: – When TIMx_CTRL1.UPDIS = 0, and repeat counter value overflow or underflow (An update event is generated when the repeat counter equals 0). – When TIMx_CTRL1.UPRS = 0, TIMx_CTRL1.UPDIS = 0, and set the TIMx_EVTGEN.UDGN bit by software to reinitialize the CNT. – When TIMx_CTRL1.UPRS = 0, TIMx_CTRL1.UPDIS = 0, and the counter CNT is reinitialized by the trigger event. (See TIMx_SMCTRL Register description) This bit is cleared by software. 0: No update event occurred 1: Update interrupt occurred
15:12	Reserved	Reserved, the reset value must be maintained
11	CC4OCF	Capture/Compare 4 overcapture flag Refer to TIMx_STS.CC1OCF description.
10	CC3OCF	Capture/Compare 3 overcapture flag Refer to TIMx_STS.CC1OCF description.
9	CC2OCF	Capture/Compare 2 overcapture flag Refer to TIMx_STS.CC1OCF description.
8	CC1OCF	Capture/Compare 1 overcapture flag This bit is set by hardware only when the corresponding channel is configured in input capture mode. Cleared by software writing 0. 0: No overcapture occurred 1: TIMx_STS.CC1ITF was already set when the value of the counter has been captured in the TIMx_CCDAT1 register.
7:6	Reserved	Reserved, the reset value must be maintained
5	CC6ITF	Capture/Compare 6 interrupt flag Refer to TIMx_STS.CC1ITF description.
4	CC5ITF	Capture/Compare 5 interrupt flag Refer to TIMx_STS.CC1ITF description.
3	CC4ITF	Capture/Compare 4 interrupt flag Refer to TIMx_STS.CC1ITF description.
2	CC3ITF	Capture/Compare 3 interrupt flag Refer to TIMx_STS.CC1ITF description.
1	CC2ITF	Capture/Compare 2 interrupt flag Refer to TIMx_STS.CC1ITF description.

Bit Field	Name	Description
0	CC1ITF	Capture/Compare 1 interrupt flag **If channel CC1 is configured as an output mode:** Except in center-aligned mode, this bit is set by hardware when the counter value is the same as the compare value (see TIMx_CTRL1.CAMSEL bit description). This bit is cleared by software. 0: No match occurred. 1: The value of TIMx_CNT is the same as the value of TIMx_CCDAT1. When the value of TIMx_CCDAT1 is greater than the value of TIMx_AR, the TIMx_STS.CC1ITF bit will go high if the counter overflows (in up-counting and up/down-counting modes) and underflows in down-counting mode. **If channel CC1 is configured as an input mode:** This bit is set by hardware when the capture event occurs. This bit is cleared by software or by reading TIMx_CCDAT1. 0: No input capture occurred. 1: Input capture occurred. Counter value has captured in the TIMx_CCDAT1. An edge with the same polarity as selected has been detected on IC1.

10.6.4 Event Generation Registers（TIMx_EVTGEN）

Offset address: 0x0C

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			BGN2	BGN	TGN	CCUDGN	UDGN	Reserved				CC4GN	CC3GN	CC2GN	CC1GN
			w	w	w	w	w					w	w	w	w

Bit Field	Name	Description
31:13	Reserved	Reserved, the reset value must be maintained
12	BGN2	Break2 generation This bit can generate a break2 event when set by software. And at this time TIMx_BKDT.MOEN = 0, TIMx_STS.BITF2 = 1, if the corresponding interrupt and DMA are enabled, the corresponding interrupt and DMA will be generated. This bit is automatically cleared by hardware. 0: No action 1: Generated a break2 event

Bit Field	Name	Description
11	BGN	Break1 generation This bit can generate a break1 event when set by software. And at this time TIMx_BKDT.MOEN = 0, TIMx_STS.BITF = 1, if the corresponding interrupt and DMA are enabled, the corresponding interrupt and DMA will be generated. This bit is automatically cleared by hardware. 0: No action 1: Generated a break1 event
10	TGN	Trigger generation This bit can generate a trigger event when set by software. And at this time TIMx_STS.TITF = 1, if the corresponding interrupt and DMA are enabled, the corresponding interrupt and DMA will be generated. This bit is automatically cleared by hardware. 0: No action 1: Generated a trigger event
9	CCUDGN	Capture/Compare control update generation This bit is set by software. And if TIMx_CTRL2.CCPCTL = 1 at this time, the CCxEN, CCxNEN and OCxMD bits are allowed to be updated. This bit is automatically cleared by hardware. 0: No action 1: Generated a COM event *Note: This bit is only valid for channels with complementary outputs.*
8	UDGN	Update generation This bit is set to '1' by software and automatically cleared to '0' by hardware. This bit can generate an update event when set by software. And at this time the counter will be reinitialized, the prescaler counter will be cleared, the counter will be cleared in center-aligned or up-counting mode, but take TIMx_AR in down-counting mode the value of the register. This bit is automatically cleared by hardware. 0: No action 1: Generated an update event
7:4	Reserved	Reserved, the reset value must be maintained
3	CC4GN	Capture/Compare 4 generation See TIMx_EVTGEN.CC1GN description.
2	CC3GN	Capture/Compare 3 generation See TIMx_EVTGEN.CC1GN description.
1	CC2GN	Capture/Compare 2 generation See TIMx_EVTGEN.CC1GN description.

Bit Field	Name	Description
0	CC1GN	Capture/Compare 1 generation This bit can generate a capture/compare event when set by software. This bit is automatically cleared by hardware. **When the corresponding channel of CC1 is in output mode:** The TIMx_STS.CC1ITF flag will be set to 1, if the corresponding interrupt and DMA are enabled, the corresponding interrupt and DMA will be generated. **When the corresponding channel of CC1 is in input mode:** TIMx_CCDAT1 will capture the current counter value, and the TIMx_STS.CC1ITF flag will be set to 1, if the corresponding interrupt and DMA are enabled, the corresponding interrupt and DMA will be generated. If the TIMx_STS.CC1ITF is already pulled high, pull TIMx_STS.CC1OCF high. 0: No action 1: Generated a CC1 capture/compare event

10.6.5 Slave Mode Control Register（TIMx_SMCTRL）

Offset address: 0x10

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								OCREFCLRF[3:0]				OCREF CLRP	Reserved		MSMD
								rw				rw			rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTF[3:0]				EXTP	EXCEN	EXTPS[1:0]		SMSEL[3:0]				Reserved	TSEL[2:0]		
rw				rw	rw	rw		rw					rw		

Bit Field	Name	Description
31:24	Reserved	Reserved, the reset value must be maintained
23:20	OCREFCLRF[3:0]	tim_ocref_clr signal filter These bits are used to define the frequency at which the tim_ocref_clr signal is sampled and the bandwidth of the tim_ocref_clr digital filtering. In effect, the digital filter is an event counter that generates a validate output after consecutive N events are recorded. 0000: No filter, sampling 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: $f_{SAMPLING}=f_{DTS}/4$, 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

19	OCREFCLRP	tim_ocref_clr signal polarity This bit is used to select whether the trigger operation is to use tim_ocref_clr or the inversion of tim_ocref_clr. 0: tim_ocref_clr active at high level or rising edge. 1: tim_ocref_clr active at low level or falling edge.
18:17	Reserved	Reserved, the reset value must be maintained
16	MSMD	Master/slave mode 0: No action 1: Events on the trigger input (TRGI) are delayed to allow a perfect synchronization between the current timer (via TRGO) and its slaves. This is useful when several timers are required to be synchronized to a single external event.
15:12	EXTF[3:0]	External trigger filter These bits are used to define the frequency at which the ETRP signal is sampled and the bandwidth of ETRP digital filtering. In effect, the digital filter is an event counter that generates a validate output after consecutive N events are recorded. 0000: No filter, sampling at f_{DTS} 1000: fSAMPLING=fDTS/8, N=6 0001: fSAMPLING=fCK_INT, N=2 1001: fSAMPLING=fDTS/8, N=8 0010: fSAMPLING=fCK_INT, N=4 1010: fSAMPLING=fDTS/16, N=5 0011: fSAMPLING=fCK_INT, N=8 1011: fSAMPLING=fDTS/16, N=6 0100: fSAMPLING=fDTS/2, N=6 1100: fSAMPLING=fDTS/16, N=8 0101: fSAMPLING=fDTS/2, N=8 1101: fSAMPLING=fDTS/32, N=5 0110: fSAMPLING=fDTS/4, N=6 1110: fSAMPLING=fDTS/32, N=6 0111: fSAMPLING=fDTS/4, N=8 1111: fSAMPLING=fDTS/32, N=8
11	EXTP	External trigger polarity This bit is used to select whether the trigger operation is to use tim_etr_in or the inversion of tim_etr_in. 0: tim_etr_in active at high level or rising edge. 1: tim_etr_in active at low level or falling edge.
10	EXCEN	External clock enable This bit is used to enable external clock mode 2, and the counter is driven by any active edge on the ETRF signal in this mode. 0: External clock mode 2 disable. 1: External clock mode 2 enable. *Note 1: when external clock mode 1 and external clock mode 2 are enabled at the same time, the input of the external clock is ETRF.* *Note 2: the following slave modes can be used simultaneously with external clock mode 2: reset mode, gated mode and trigger mode; However, TRGI cannot connect to ETRF (TIMx_SMCTRL.TSEL ≠ '111').* *Note 3: setting the TIMx_SMCTRL.EXCEN bit has the same effect as selecting external clock mode 1 and connecting TRGI to ETRF (TIMx_SMCTRL.SMSEL = 0111 and TIMx_SMCTRL.TSEL = 111).*
9:8	EXTPS[1:0]	External trigger prescaler The frequency of the external trigger signal ETRP must be at most 1/4 of TIMxCLK frequency.

		When a faster external clock is input, a prescaler can be used to reduce the frequency of ETRP. 00: Prescaler disable 01: ETRP frequency divided by 2 10: ETRP frequency divided by 4 11: ETRP frequency divided by 8
7:4	SMSEL[3:0]	Slave mode selection When an external signal is selected, the active edge of the trigger signal (TRGI) is linked to the selected external input polarity (refer to input control register and control register description) 0000: Disable slave mode. If TIMx_CTRL1.CNTEN = 1, the prescaler is driven directly by the internal clock. 0001: Encoder mode 1. According to the level of TI2FP2, the counter up-counting or down-counting on the edge of TI1FP1. 0010: Encoder mode 2. According to the level of TI1FP1, the counter up-counting or down-counting on the edge of TI2FP2. 0011: Encoder mode 3. According to the input level of another signal, the counter up-counting or down-counting on the edges of TI2FP1 and TI2FP2. 0100: Reset mode. On the rising edge of the selected trigger input (TRGI), the counter is reinitialized and the shadow register is updated. 0101: Gated mode. When the trigger input (TRGI) is high, the clock of the counter is enabled. Once the trigger input becomes low, the counter stops counting, but is not reset. In this mode, the start and stop of the counter are controlled. 0110: Trigger mode. When a rising edge occurs on the trigger input (TRGI), the counter is started but not reset. In this mode, only the start of the counter is controlled. 0111: External clock mode 1. The counter is clocked by the rising edge of the selected trigger input (TRGI). 1000: Dual-Pulse encoder mode 2 1001: Quadrature encoder mode 4 - The counter counts up/down on the edge of TI1FP1 based on the level of TI2FP2. The counting edge is selected by CC1P. 1010: Quadrature encoder mode 5 - The counter counts up/down on the edge of TI2FP2 based on the level of TI1FP1. The counting edge is selected by CC2P. 1011: Pulse-Level encoder mode 2 1100: Pulse-Level encoder mode 1. Set the counting edge of TI2FP2 through CC2P. 1101: Combined Gated + Reset mode, the counter is enabled when the trigger input (TRGI) is high. Once the trigger input goes low, the counter stops (and resets). The start and stop of the counter are both controlled. 1110: Combined Reset + Trigger Mode - The counter starts (and resets) on the rising edge of the trigger input TRGI, only the start of the counter is controlled. 1111: Dual-Pulse encoder mode 1. Set the counting edge of TI1FP1 and TI2FP2 through CC1P and CC2P. *Note: do not use gated mode if TI1F_ED is selected as the trigger input*

		(TIMx_SMCTRL.TSEL=100). This is because TI1F_ED outputs a pulse for each TI1F transition, whereas gated mode checks the level of the triggered input.
3	Reserved	Reserved, the reset value must be maintained
2:0	TSEL[2:0]	Trigger selection These 3 bits are used to select the trigger input of the synchronous counter. 0xx: Internal trigger (ITRx) , Select the ITR signal source based on TIMx_INSEL. 100: TI1 edge detector (TI1F_ED) 101: Filtered timer input 1 (TI1FP1) 110: Filtered timer input 2 (TI2FP2) 111: External triggered Input (ETRF) *Note: these bits must be changed only when not in use (e. g. TIMx_SMCTRL.SMSEL=000) to avoid false edge detection at the transition.*

10.6.6 DMA/Interrupt Enable Registers（TIMx_DINTEN）

Offset address: 0x14

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								CC9IEN	COMIEN	TDEN	COMDEN	UDEN	BIEN	TIEN	UIEN
								rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				CC4DEN	CC3DEN	CC2DEN	CC1DEN	CC8IEN	CC7IEN	CC6IEN	CC5IEN	CC4IEN	CC3IEN	CC2IEN	CC1IEN
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit Field	**Name**	**Description**
31:22	Reserved	Reserved, the reset value must be maintained
23	CC9IEN	Capture/Compare 9 interrupt enable 0: Disable capture/compare 9 interrupt 1: Enable capture/compare 9 interrupt
22	COMIEN	COM interrupt enable 0: Disable COM interrupt 1: Enable COM interrupt
21	TDEN	Trigger DMA request enable 0: Disable trigger DMA request 1: Enable trigger DMA request
20	COMDEN	COM DMA request enable 0: Disable COM DMA request 1: Enable COM DMA request
19	UDEN	Update DMA request enable 0: Disable update DMA request 1: Enable update DMA request

Bit Field	Name	Description
18	BIEN	Break interrupt enable 0: Disable break interrupt 1: Enable break interrupt
17	TIEN	Trigger interrupt enable 0: Disable trigger interrupt 1: Enable trigger interrupt
16	UIEN	Update interrupt enable 0: Disable update interrupt 1: Enables update interrupt
15:12	Reserved	Reserved, the reset value must be maintained
11	CC4DEN	Capture/Compare 4 DMA request enable 0: Disable capture/compare 4 DMA request 1: Enable capture/compare 4 DMA request
10	CC3DEN	Capture/Compare 3 DMA request enable 0: Disable capture/compare 3 DMA request 1: Enable capture/compare 3 DMA request
9	CC2DEN	Capture/Compare 2 DMA request enable 0: Disable capture/compare 2 DMA request 1: Enable capture/compare 2 DMA request
8	CC1DEN	Capture/Compare 1 DMA request enable 0: Disable capture/compare 1 DMA request 1: Enable capture/compare 1 DMA request
7	CC8IEN	Capture/Compare 8 interrupt enable 0: Disable capture/compare 8 interrupt 1: Enable capture/compare 8 interrupt
6	CC7IEN	Capture/Compare 7 interrupt enable 0: Disable capture/compare 7 interrupt 1: Enable capture/compare 7 interrupt
5	CC6IEN	Capture/Compare 6 interrupt enable 0: Disable capture/compare 6 interrupt 1: Enable capture/compare 6 interrupt
4	CC5IEN	Capture/Compare 5 interrupt enable 0: Disable capture/compare 5 interrupt 1: Enable capture/compare 5 interrupt
3	CC4IEN	Capture/Compare 4 interrupt enable 0: Disable capture/compare 4 interrupt 1: Enable capture/compare 4 interrupt
2	CC3IEN	Capture/Compare 3 interrupt enable 0: Disable capture/compare 3 terrupt 1: Enable capture/compare 3 nterrupt

Bit Field	Name	Description
1	CC2IEN	Capture/Compare 2 interrupt enable 0: Disable capture/compare 2 nterrupt 1: Enable capture/compare 2 nterrupt
0	CC1IEN	Capture/Compare 1 interrupt enable 0: Disable capture/compare 1 nterrupt 1: Enable capture/compare 1 nterrupt

10.6.7 Capture/Compare Mode Register 1（TIMx_CCMOD1）

Offset address: 0x18

Reset value: 0x0000 0000

Channels can be used for input (capture mode) or output (compare mode), and the direction of the channel is defined by the corresponding CCxSEL bit. The other bits of the register act differently in input and output modes. OCx describes the function of a channel in output mode, ICx describes the function of a channel in input mode. Hence, please note that the same bit can have different meanings for output mode and for input mode.

Output compare mode:

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC2MD[2:0]			OC2CEN	OC2FEN	OC2PEN	CC2SEL[1:0]		OC1MD[2:0]			OC1CEN	OC1FEN	OC1PEN	CC1SEL[1:0]	
rw			rw	rw	rw	rw		rw			rw	rw	rw	rw	

Bit Field	Name	Description
31:16	Reserved	Reserved, the reset value must be maintained
15:13	OC2MD[2:0]	Output Compare 2 mode
12	OC2CEN	Output Compare 2 clear enable
11	OC2FEN	Output Compare 2 fast enable
10	OC2PEN	Output Compare 2 preload enable
9:8	CC2SEL[1:0]	Capture/Compare 2 selection These bits are used to select the input/output and input mapping of the channel 00: CC2 channel is configured as output 01: CC2 channel is configured as input, IC2 is mapped on TI2 10: CC2 channel is configured as input, IC2 is mapped on TI1 11: CC2 channel is configured as input, IC2 is mapped on TRC. This mode is only active when the internal trigger input is selected by TIMx_SMCTRL.TSEL. *Note: CC2SEL is writable only when the channel is off (TIMx_CCEN.CC2EN = 0).*
7:5	OC1MD[2:0]	Output Compare 1 mode These bits are used to manage the output reference signal OC1REF, which determines the values of OC1 and OC1N, and is valid at high levels, while the active levels of OC1 and

Bit Field	Name	Description
		OC1N depend on the TIMx_CCEN.CC1P and TIMx_CCEN.CC1NP bits. 000: Frozen. Comparison between TIMx_CCDAT1 register and counter TIMx_CNT has no effect on OC1REF signal. 001: Set channel 1 to the active level on match. When TIMx_CCDAT1 = TIMx_CNT, OC1REF signal will be forced high. 010: Set channel 1 as inactive level on match. When TIMx_CCDAT1 = TIMx_CNT, OC1REF signal will be forced low. 011: Toggle. When TIMx_CCDAT1 = TIMx_CNT, OC1REF signal will be toggled. 100: Force to inactive level. OC1REF signal is forced low. 101: Force to active level. OC1REF signal is forced high. 110: PWM mode 1 - In up-counting mode, if TIMx_CNT < TIMx_CCDAT1, OC1REF signal of channel 1 is high, otherwise it is low. In down-counting mode, if TIMx_CNT > TIMx_CCDAT1, OC1REF signal of channel 1 is low, otherwise it is high. 111: PWM mode 2 - In up-counting mode, if TIMx_CNT < TIMx_CCDAT1, OC1REF signal of channel 1 is low, otherwise it is high. In down-counting mode, if TIMx_CNT > TIMx_CCDAT1, OC1REF signal of channel 1 is high, otherwise it is low. *Note 1: In PWM mode 1 or PWM mode 2, the OC1REF level changes only when the comparison result changes or when the output compare mode is switched from frozen mode to PWM mode.*
4	OC1CEN	Output Compare 1 clear enable 0: OC1REF is not affected by tim_ocref_clr_in input level 1: OC1REF is cleared immediately when the tim_ocref_clr_in input level is detected as high (tim_ocref_clr_in is controlled by the TIMx_CTRL1.CLRSEL register).
3	OC1FEN	Output Compare 1 fast enable This bit is used to speed up the response of the CC output to the trigger input event. 0: CC1 behaves normally depending on the counter and CCDAT1 values, even if the trigger is ON. The minimum delay for activating CC1 output when an edge occurs on the trigger input is 5 clock cycles. 1: An active edge of the trigger input acts like a comparison match on CC1 output. Therefore, OC is set to the comparison level regardless of the comparison result. The delay time for sampling the trigger input and activating the CC1 output is reduced to 3 clock cycles. OCxFEN only works if the channel is configured in PWM1 or PWM2 mode.
2	OC1PEN	Output Compare 1 preload enable 0: Disable preload function of TIMx_CCDAT1 register. Supports write operations to TIMx_CCDAT1 register at any time, and the written value is effective immediately. 1: Enable preload function of TIMx_CCDAT1 register. Only read and write operations to preload registers. When an update event occurs, the value of TIMx_CCDAT1 is loaded into the active register. *Note 1: Only when TIMx_CTRL1.ONEPM = 1(In one-pulse mode), PWM mode can be used without verifying the preload register, otherwise no other behavior can be predicted.*
1:0	CC1SEL[1:0]	Capture/compare 1 selection These bits are used to select the input/output and input mapping of the channel

Bit Field	Name	Description
		00: CC1 channel is configured as output 01: CC1 channel is configured as input, IC1 is mapped on TI1 10: CC1 channel is configured as input, IC1 is mapped on TI2 11: CC1 channels are configured as inputs and IC1 is mapped to TRC. This mode is only active when the internal trigger input is selected by TIMx_SMCTRL.TSEL. *Note: CC1SEL is writable only when the channel is off (TIMx_CCEN.CC1EN = 0).*

Input capture mode:

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IC2F[3:0]				IC2PSC[1:0]		CC2SEL[1:0]		IC1F[3:0]				IC1PSC[1:0]		CC1SEL[1:0]	
rw				rw		rw		rw				rw		rw	

Bit field	Name	Description
31:16	Reserved	Reserved, the reset value must be maintained
15:12	IC2F[3:0]	Input capture 2 filter
11:10	IC2PSC[1:0]	Input capture 2 prescaler
9:8	CC2SEL[1:0]	Capture/Compare 2 selection These bits are used to select the input/output and input mapping of the channel 00: CC2 channel is configured as output 01: CC2 channel is configured as input, IC2 is mapped on TI2 10: CC2 channel is configured as input, IC2 is mapped on TI1 11: CC2 channel is configured as input, IC2 is mapped on TRC. This mode is only active when the internal trigger input is selected by TIMx_SMCTRL.TSEL. *Note: CC2SEL is writable only when the channel is off (TIMx_CCEN.CC2EN = 0).*
7:4	IC1F[3:0]	Input Capture 1 filter These bits are used to define sampling frequency of TI1 input and the length of digital filter. The digital filter is an event counter that generates an output transition after N events are recorded. 0000: No filter, sampling at f_{DTS} frequency 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
3:2	IC1PSC[1:0]	Input Capture 1 prescaler These bits are used to select the ratio of the prescaler for IC1 (CC1 input).

		When TIMx_CCEN.CC1EN = 0, the prescaler will be reset. 00: No prescaler, capture is done each time an edge is detected on the capture input 01: Capture is done once every 2 events 10: Capture is done once every 4 events 11: Capture is done once every 8 events
1:0	CC1SEL[1:0]	Capture/Compare 1 selection These bits are used to select the input/output and input mapping of the channel 00: CC1 channel is configured as output 01: CC1 channel is configured as input, IC1 is mapped on TI1 10: CC1 channel is configured as input, IC1 is mapped on TI2 11: CC1 channel is configured as input, IC1 is mapped to TRC. This mode is only active when the internal trigger input is selected by TIMx_SMCTRL.TSEL. *Note: CC1SEL is writable only when the channel is off (TIMx_CCEN.CC1EN = 0).*

10.6.8 Capture/Compare Mode Register 2（TIMx_CCMOD2）

Offset address: 0x1C

Reset value: 0x0000 0000

Refer to the description of the CCMOD1 register above.

Output compare mode:

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC4MD[2:0]			OC4CEN	OC4FEN	OC4PEN	CC4SEL[1:0]		OC3MD[2:0]			OC3CEN	OC3FEN	OC3PEN	CC3SEL[1:0]	
rw			rw	rw	rw	rw		rw			rw	rw	rw	rw	

Bit Field	Name	Description
31:16	Reserved	Reserved, the reset value must be maintained
15:13	OC4MD[2:0]	Output compare 4 mode
12	OC4CEN	Output compare 4 clear enable
11	OC4FEN	Output compare 4 fast enable
10	OC4PEN	Output compare 4 preload enable
9:8	CC4SEL[1:0]	Capture/Compare 4 selection These bits are used to select the input/output and input mapping of the channel 00: CC4 channel is configured as output 01: CC4 channel is configured as input, IC4 is mapped on TI4 10: CC4 channel is configured as input, IC4 is mapped on TI3 11: CC4 channel is configured as input, IC4 is mapped on TRC. This mode is only active when the internal trigger input is selected by TIMx_SMCTRL.TSEL. Note: CC4SEL is writable only when the channel is off (TIMx_CCEN.CC4EN = 0).
7:5	OC3MD[2:0]	Output compare 3 mode

Bit Field	Name	Description
31:16	Reserved	Reserved, the reset value must be maintained
4	OC3CEN	Output compare 3 clear enable
3	OC3FEN	Output compare 3 fast enable
2	OC3PEN	Output compare 3 preload enable
1:0	CC3SEL[1:0]	Capture/Compare 3 selection These bits are used to select the input/output and input mapping of the channel 00: CC3 channel is configured as output 01: CC3 channel is configured as input, IC3 is mapped to TI3 10: CC3 channel is configured as input, IC3 is mapped on TI4 11: CC3 channel is configured as input, IC3 is mapped to TRC. This mode is only active when the internal trigger input is selected by TIMx_SMCTRL.TSEL. *Note: CC3SEL is writable only when the channel is off (TIMx_CCEN.CC3EN = 0).*

Input capture mode:

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IC4F[3:0]				IC4PSC[1:0]		CC4SEL[1:0]		IC3F[3:0]				IC3PSC[1:0]		CC3SEL[1:0]	
rw				rw		rw		rw				rw		rw	

Bit Field	Name	Description
31:16	Reserved	Reserved, the reset value must be maintained
15:12	IC4F[3:0]	Input capture 4 filter
11:10	IC4PSC[1:0]	Input capture 4 prescaler
9:8	CC4SEL[1:0]	Capture/Compare 4 selection These bits are used to select the input/output and input mapping of the channel 00: CC4 channel is configured as output 01: CC4 channel is configured as input, IC4 is mapped on TI4 10: CC4 channel is configured as input, IC4 is mapped on TI3 11: CC4 channel is configured as input, IC4 is mapped on TRC. This mode is only active when the internal trigger input is selected by TIMx_SMCTRL.TSEL. *Note: CC4SEL is writable only when the channel is off (TIMx_CCEN.CC4EN = 0).*
7:4	IC3F[3:0]	Input capture 3 filter
3:2	IC3PSC[1:0]	Input capture 3 prescaler

Bit Field	Name	Description
31:16	Reserved	Reserved, the reset value must be maintained
1:0	CC3SEL[1:0]	Capture/compare 3 selection These bits are used to select the input/output and input mapping of the channel 00: CC3 channel is configured as output 01: CC3 channel is configured as input, IC3 is mapped to TI3 10: CC3 channel is configured as input, IC3 is mapped on TI4 11: CC3 channel is configured as input, IC3 is mapped to TRC. This mode is only active when the internal trigger input is selected by TIMx_SMCTRL.TSEL. *Note: CC3SEL is writable only when the channel is off (TIMx_CCEN.CC3EN = 0).*

10.6.9 Capture/Compare Mode Register 3（TIMx_CCMOD3）

Offset address: 0x20

Reset value: 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved							OC9PEN	Reserved			OC8PEN	Reserved			OC7PEN
							rw				rw				rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC6MD[2:0]			OC6CEN	OC6FEN	OC6PEN	Reserved		OC5MD[2:0]			OC5CEN	OC5FEN	OC5PEN	Reserved	
rw			rw	rw	rw			rw			rw	rw	rw		

Bit Field	Name	Description
31:25	Reserved	Reserved, the reset value must be maintained
24	OC9PEN	Output compare 9 preload enable
23:21	Reserved	Reserved, the reset value must be maintained
20	OC8PEN	Output compare 8 preload enable
19:17	Reserved	Reserved, the reset value must be maintained
16	OC7PEN	Output compare 7 preload enable
15:13	OC6MD[2:0]	Output compare 6 mode
12	OC6CEN	Output compare 6 clear enable
11	OC6FEN	Output compare 6 fast enable
10	OC6PEN	Output compare 6 preload enable
9:8	Reserved	Reserved, the reset value must be maintained
7:5	OC5MD[2:0]	Output compare 5 mode
4	OC5CEN	Output compare 5 clear enable
3	OC5FEN	Output compare 5 fast enable

Bit Field	Name	Description
2	OC5PEN	Output compare 5 preload enable
1:0	Reserved	Reserved, the reset value must be maintained

10.6.10 Capture/Compare Enable Registers（TIMx_CCEN）

Offset address: 0x24

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								CC6P	CC6EN	Reserved		CC5P	CC5EN	Reserved	
								rw	rw			rw	rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CC4P	CC4EN	CC4NP	CC4NEN	CC3P	CC3EN	CC3NP	CC3NEN	CC2P	CC2EN	CC2NP	CC2NEN	CC1P	CC1EN	CC1NP	CC1NEN
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit Field	Name	Description
31:24	Reserved	Reserved, the reset value must be maintained
23	CC6P	Capture/Compare 6 output polarity See TIMx_CCEN.CC1P description.
22	CC6EN	Capture/Compare 6 output enable See TIMx_CCEN.CC1EN description.
21:20	Reserved	Reserved, the reset value must be maintained
19	CC5P	Capture/Compare 5 output polarity See TIMx_CCEN.CC1P description.
18	CC5EN	Capture/Compare 5 output enable See TIMx_CCEN.CC1EN description.
17：16	Reserved	Reserved, the reset value must be maintained
15	CC4P	Capture/Compare 4 output polarity See TIMx_CCEN.CC1P description.
14	CC4EN	Capture/Compare 4 output enable See TIMx_CCEN.CC1EN description.
13	CC4NP	Capture/Compare 4 complementary output polarity See TIMx_CCEN.CC1NP description.
12	CC4NEN	Capture/Compare 4 complementary output enable See TIMx_CCEN.CC1NEN description.
11	CC3P	Capture/Compare 3 output polarity See TIMx_CCEN.CC1P description.
10	CC3EN	Capture/Compare 3 output enable See TIMx_CCEN.CC1EN description.
9	CC3NP	Capture/Compare 3 complementary output polarity See TIMx_CCEN.CC1NP description.

Bit Field	Name	Description
8	CC3NEN	Capture/Compare 3 complementary output enable See TIMx_CCEN.CC1NEN description.
7	CC2P	Capture/Compare 2 output polarity See TIMx_CCEN.CC1P description.
6	CC2EN	Capture/Compare 2 output enable See TIMx_CCEN.CC1EN description.
5	CC2NP	Capture/Compare 2 complementary output polarity See TIMx_CCEN.CC1NP description.
4	CC2NEN	Capture/Compare 2 complementary output enable See TIMx_CCEN.CC1NEN description.
3	CC1P	Capture/Compare 1 output polarity **When the corresponding channel of CC1 is in output mode:** 0: OC1 active high 1: OC1 active low **When the corresponding channel of CC1 is in input mode:** At this time, this bit is used to select whether IC1 or the inverse signal of IC1 is used as the trigger or capture signal. 0: non-inverted: Capture action occurs when IC1 generates a rising edge. When used as external trigger, IC1 is non-inverted. 1: inverted: Capture action occurs when IC1 generates a falling edge. When used as external trigger, IC1 is inverted. *Note: If TIMx_BKDT.LCKCFG = 3 or 2, these bits cannot be modified.*
2	CC1EN	Capture/Compare 1 output enable **When the corresponding channel of CC1 is in output mode:** 0: Disable - Disable output OC1 signal. The level of OC1 depends on the value of these bits TIMx_BKDT.MOEN, TIMx_BKDT.OSSI, TIMx_BKDT.OSSR, TIMx_CTRL2.OI1, TIMx_CTRL2.OI1N and TIMx_CCEN.CC1NEN. 1: Enable - Enable output OC1 signal. The level of OC1N depends on the value of these bits TIMx_BKDT.MOEN, TIMx_BKDT.OSSI, TIMx_BKDT.OSSR, TIMx_CTRL2.OI1, TIMx_CTRL2.OI1N and TIMx_CCEN.CC1NEN. **When the corresponding channel of CC1 is in input mode:** At this time, this bit is used to disable/enable the capture function. 0: Disable capture 1: Enable capture
1	CC1NP	Capture/Compare 1 complementary output polarity 0: OC1N active high 1: OC1N active low
0	CC1NEN	Capture/Compare 1 complementary output enable 0: Disable - Disable output OC1N signal. The level of OC1N depends on the value of these bits TIMx_BKDT.MOEN, TIMx_BKDT.OSSI, TIMx_BKDT.OSSR, TIMx_CTRL2.OI1, TIMx_CTRL2.OI1N and TIMx_CCEN.CC1EN. 1: Enable - Enable output OC1N signal. The level of OC1N depends on the value of these bits

Bit Field	Name	Description
		TIMx_BKDT.MOEN, TIMx_BKDT.OSSI, TIMx_BKDT.OSSR, TIMx_CTRL2.OI1, TIMx_CTRL2.OI1N and TIMx_CCEN.CC1EN.

Table 10-16 Output Control Bits Of Complementary OCx And OCxN Channels With Break Function

Control Bits					Output State (1)	
MOEN	**OSSI**	**OSSR**	**CCxEN**	**CCxNEN**	**OCx Output State**	**OCxN Output State**
1	X	0	0	0	Output disabled（not driven by timer） OCx=0，OCx_EN=0	Output disabled（not driven by timer） OCxN=0，OCxN_EN=0
		0	0	1	Output disabled（not driven by timer） OCx=0，OCx_EN=0	OCxREF + polarity OCxN= OCxREF xor CCxNP， OCxN_EN=1
		0	1	0	OCxREF + polarity OCx= OCxREF xor CCxP， OCx_EN=1	Output disabled（not driven by timer） OCxN=0，OCxN_EN=0
		0	1	1	OCxREF + polarity + dead-time, OCx_EN=1	Complementary to OCxREF + polarity + dead-time , OCxN_EN=1
		1	0	0	Output disabled（not driven by timer） OCx=CCxP，OCx_EN=0	Output disabled（not driven by timer） OCxN=CCxNP，OCxN_EN=0
		1	0	1	Off-state (Output enabled with inactive state) OCx=CCxP，OCx_EN=1	OCxREF + polarity OCxN= OCxREF xor CCxNP， OCxN_EN=1
		1	1	0	OCxREF + polarity OCx= OCxREF xor CCxP， OCx_EN=1	Off-state (Output enabled with inactive state) OCxN=CCxNP，OCxN_EN=1
		1	1	1	OCxREF + polarity + dead-time OCx_EN=1	Complementary to OCxREF + polarity + dead-time OCxN_EN=1
0	0	X	0	0	Output disabled（not driven by timer） Asynchronously: OCx=CCxP, OCx_EN=0, OCxN=CCxNP,OCxN_EN=0; Then if the clock is present: if (CCxP ^ OIx) ^ (CCxNP^OIxN)! = 0， OCx=OIx, OCxN=OIxN after a dead-time	
	0		0	1		
	0		1	0		
	0		1	1		

	1		0	0	Off-state (Output enabled with inactive state) Asynchronously: OCx=CCxP, OCx_EN=1, OCxN=CCxNP,OCxN_EN=1; Then if the clock is present: if (CCxP ^ OIx) ^ (CCxNP^OIxN)! = 0， OCx=OIx , OCxN=OIxN after a dead-time
	1		0	1	
	1		1	0	
	1		1	1	

1. If both outputs of a channel are not used (CCxEN = CCxNEN = 0), OIx, OIxN, CCxP and CCxNP must all be cleared.

Note: the status of external I/O pins connected to complementary OCx and OCxN channels depends on the OCx and OCxN channel states and GPIO and AFIO registers.

10.6.11 Capture/Compare Register 1（TIMx_CCDAT1）

Offset address: 0x28

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CCDDAT1[15:0]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCDAT1[15:0]															
rw															

Bit Field	Name	Description
31:16	CCDDAT1[15:0]	Capture/Compare 1 down-counting value, dedicated to center-aligned asymmetric mode. • CC1 channel can only configured as output: CCDDAT1 contains the value to be compared to the counter TIMx_CNT (only when TIMx_CTRL1.DIR = 1 and in asymmetric mode), signal is sent out on the OC1 output. If the preload feature is not selected in TIMx_CCMOD1.OC1PEN bit, the written value is immediately transferred to the active register. Otherwise, this preloaded value is transferred to the active register only when an update event occurs.
15:0	CCDAT1[15:0]	Capture/Compare 1 value • CC1 channel is configured as output: CCDAT1 contains the value to be compared to the counter TIMx_CNT (except when TIMx_CTRL1.DIR = 1 and in asymmetric mode), signal is sent out on the OC1 output. If the preload feature is not selected in TIMx_CCMOD1.OC1PEN bit, the written value is immediately transferred to the active register. Otherwise, this preloaded value is transferred to the active register only when an update event occurs.

Bit Field	Name	Description
31:16	CCDDAT1[15:0]	Capture/Compare 1 down-counting value, dedicated to center-aligned asymmetric mode. • CC1 channel can only configured as output: CCDDAT1 contains the value to be compared to the counter TIMx_CNT (only when TIMx_CTRL1.DIR = 1 and in asymmetric mode), signal is sent out on the OC1 output. If the preload feature is not selected in TIMx_CCMOD1.OC1PEN bit, the written value is immediately transferred to the active register. Otherwise, this preloaded value is transferred to the active register only when an update event occurs.
		• CC1 channel is configured as input: CCDAT1 contains the counter value transferred by the last input capture 1 event (IC1). When configured as input mode, register CCDAT1 and CCDDAT1 are only readable. When configured as output mode, register CCDAT1 and CCDDAT1 are readable and writable.

10.6.12 Capture/Compare Register 2（TIMx_CCDAT2）

Offset address: 0x2C

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CCDDAT2[15:0]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCDAT2[15:0]															
rw															

Bit Field	Name	Description
31:16	CCDDAT2[15:0]	Capture/Compare 2 down-counting value, dedicated to center-aligned asymmetric mode • CC2 channel can only configured as output: CCDDAT2 contains the value to be compared to the counter TIMx_CNT (only when TIMx_CTRL1.DIR = 1 and in asymmetric mode), signal is sent out on the OC2 output. If the preload feature is not selected in TIMx_CCMOD1.OC2PEN bit, the written value is immediately transferred to the active register. Otherwise, this preloaded value is transferred to the active register only when an update event occurs.
15:0	CCDAT2[15:0]	Capture/Compare 2 value • CC2 channel is configured as output: CCDAT2 contains the value to be compared to the counter TIMx_CNT (except when TIMx_CTRL1.DIR = 1 and in asymmetric mode), signal is sent out on the OC2 output.

Bit Field	Name	Description
31:16	CCDDAT2[15:0]	Capture/Compare 2 down-counting value, dedicated to center-aligned asymmetric mode - CC2 channel can only configured as output: CCDDAT2 contains the value to be compared to the counter TIMx_CNT (only when TIMx_CTRL1.DIR = 1 and in asymmetric mode), signal is sent out on the OC2 output. If the preload feature is not selected in TIMx_CCMOD1.OC2PEN bit, the written value is immediately transferred to the active register. Otherwise, this preloaded value is transferred to the active register only when an update event occurs.
		If the preload feature is not selected in TIMx_CCMOD1.OC2PEN bit, the written value is immediately transferred to the active register. Otherwise, this preloaded value is transferred to the active register only when an update event occurs. - CC2 channel is configured as input: CCDAT2 contains the counter value transferred by the last input capture 2 event (IC2). When configured as input mode, register CCDAT2 and CCDDAT2 are only readable. When configured as output mode, register CCDAT2 and CCDDAT2 are readable and writable.

10.6.13 Capture/Compare Register 3（TIMx_CCDAT3）

Offset address: 0x30

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CCDDAT3[15:0]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCDAT3[15:0]															
rw															

Bit Field	Name	Description
31:16	CCDDAT3[15:0]	Capture/Compare 3 down-counting value, dedicated to center-aligned asymmetric mode - CC3 channel can only configured as output: CCDDAT3 contains the value to be compared to the counter TIMx_CNT (only when TIMx_CTRL1.DIR = 1 and in asymmetric mode), signal is sent out on the OC3 output. If the preload feature is not selected in TIMx_CCMOD2.OC3PEN bit, the written value is immediately transferred to the active register. Otherwise, this preloaded value is transferred to the active register only when an update event occurs.
15:0	CCDAT3[15:0]	Capture/Compare 3 value - CC3 channel is configured as output:

Bit Field	Name	Description
31:16	CCDDAT3[15:0]	Capture/Compare 3 down-counting value, dedicated to center-aligned asymmetric mode - CC3 channel can only configured as output: CCDDAT3 contains the value to be compared to the counter TIMx_CNT (only when TIMx_CTRL1.DIR = 1 and in asymmetric mode), signal is sent out on the OC3 output. If the preload feature is not selected in TIMx_CCMOD2.OC3PEN bit, the written value is immediately transferred to the active register. Otherwise, this preloaded value is transferred to the active register only when an update event occurs.
		CCDAT3 contains the value to be compared to the counter TIMx_CNT (except when TIMx_CTRL1.DIR = 1 and in asymmetric mode), signal is sent out on the OC3 output. If the preload feature is not selected in TIMx_CCMOD2.OC3PEN bit, the written value is immediately transferred to the active register. Otherwise, this preloaded value is transferred to the active register only when an update event occurs. - CC3 channel is configured as input: CCDAT3 contains the counter value transferred by the last input capture 3 event (IC3). When configured as input mode, register CCDAT3 and CCDDAT3 are only readable. When configured as output mode, register CCDAT3 and CCDDAT3 are readable and writable.

10.6.14 Capture/Compare Register 4（TIMx_CCDAT4）

Offset address: 0x34

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CCDDAT4[15:0]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCDAT4[15:0]															
rw															

Bit Field	Name	Description
31:16	CCDDAT4[15:0]	Capture/Compare 4 down-counting value, dedicated to center-aligned asymmetric mode - CC4 channel can only configured as output: CCDDAT4 contains the value to be compared to the counter TIMx_CNT (only when TIMx_CTRL1.DIR = 1 and in asymmetric mode), signaling on the OC4 output. If the preload feature is not selected in TIMx_CCMOD2.OC4PEN bit, the written value is immediately transferred to the active register. Otherwise, this preloaded value is transferred to the active register only when an update event occurs.

15:0	CCDAT4[15:0]	Capture/Compare 4 value • CC4 channel is configured as output: CCDAT4 contains the value to be compared to the counter TIMx_CNT (except when TIMx_CTRL1.DIR = 1 and in asymmetric mode), signaling on the OC4 output. If the preload feature is not selected in TIMx_CCMOD2.OC4PEN bit, the written value is immediately transferred to the active register. Otherwise, this preloaded value is transferred to the active register only when an update event occurs. • CC4 channel is configured as input: CCDAT4 contains the counter value transferred by the last input capture 4 event (IC4). When configured as input mode, register CCDAT4 and CCDDAT4 are only readable. When configured as output mode, register CCDAT4 and CCDDAT4 are readable and writable.

10.6.15 Capture/Compare Register 5（TIMx_CCDAT5）

Offset address: 0x38

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCDAT5[15:0]															
rw															

Bit Field	Name	Description
31:16	Reserved	Reserved, the reset value must be maintained
15:0	CCDAT5[15:0]	Capture/Compare 5 value • CC5 channel can only configured as output: CCDAT5 contains the value to be compared with the counter TIMx_CNT, and signal is sent out on the OC5 output. If the preload function is not selected in the TIMx_CCMOD3.OC5PEN bit, the written value is immediately transferred to the active register. Otherwise, this preloaded value is transferred to the active register only when an update event occurs. CC5 is used for comparator blanking.

10.6.16 Capture/Compare Register 6（TIMx_CCDAT6）

Offset address: 0x3C

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCDAT6[15:0]															
rw															

Bit Field	Name	Description
31:16	Reserved	Reserved, the reset value must be maintained
15:0	CCDAT6[15:0]	Capture/Compare 6 value • CC6 channel can only configured as output: CCDAT6 contains the value to be compared with the counter TIMx_CNT, and signals are sent out on the OC6 output. If the preload function is not selected in the TIMx_CCMOD3.OC6PEN bit, the written value is immediately transferred to the active register. Otherwise, this preloaded value is transferred to the active register only when an update event occurs.

10.6.17 Prescaler（TIMx_PSC）

Offset address: 0x40

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSC[15:0]															
rw															

Bit Field	Name	Description
31:16	Reserved	Reserved, the reset value must be maintained
15:0	PSC[15:0]	Prescaler value Counter clock $f_{CK\_CNT} = f_{CK\_PSC}$/ (PSC [15:0] +1). The PSC value is loaded into the shadow register of the prescaler each time an update event occurs.

10.6.18 Auto-Reload Register（TIMx_AR）

Offset address: 0x44

Reset value: 0x0000 FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AR[15:0]															
rw															

Bit Field	Name	Description
31:16	Reserved	Reserved, the reset value must be maintained
15:0	AR[15:0]	Auto-reload value AR contains the value to be loaded into the actual auto-reload register. Refer to section 10.5.1 for details on the update and action of AR. When the value for auto-reload is empty, the counter will not operate.

10.6.19 Counters（TIMx_CNT）

Offset address: 0x48
Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT[15:0]															
rw															

Bit Field	Name	Description
31:16	Reserved	Reserved, the reset value must be maintained
15:0	CNT[15:0]	Counter value

10.6.20 Repeat Count Registers（TIMx_REPCNT）

Offset address: 0x4C
Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Reserved	REPCNT[7:0]
	rw

Bit Field	Name	Description
31:8	Reserved	Reserved, the reset value must be maintained
7:0	REPCNT[7:0]	Repetition counter value Repetition counter is used to generate the update event or update the timer registers only after a given number (N+1) cycles of the counter, where N is the value of TIMx_REPCNT.REPCNT . The repetition counter is decremented at each counter overflow in up-counting mode, at each counter underflow in down-counting mode or at each counter overflow and at each counter underflow in center-aligned mode. Setting the TIMx_EVTGEN.UDGN bit will reload the content of TIMx_REPCNT.REPCNT and generate an update event.

10.6.21 Break and Dead-Time Registers（TIMx_BKDT）

Offset address: 0x50

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved										BRK2BID	BRKBID	BRK2 DSRM	BRK DSRM	BK2EN	BK2P
										rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LCKCFG[1:0]		OSSR	OSSI	BKEN	BKP	AOEN	MOEN	DTGN[7:0]							
rw		rw	rw	rw	rw	rw	rw	rw							

Note: BRK2BID、BRKBID、BK2EN、BK2P、AOEN、BKP、BKEN、OSSI、OSSR and DTGN[7:0] bits can all be write protected depending on the LOCK configuration, and it is necessary to configure all of them on the first write to the TIMx_BKDT register.

Bit Field	Name	Description
31:22	Reserved	Reserved, the reset value must be maintained
21	BRK2BID	Break2 bidirectional enable 0: Break 2 is in input mode. 1: Break 2 is in bidirectional mode. In bidirectional mode, the break 2 input is configured in input and open-drain output modes. Any break 2 event will generate a low level on the break 2 input IO, indicating an internal break 2 event to the external device. *Note: Once the LOCK level (LCKCFG bit in the TIMx_BKDT register) is set to '1', this bit cannot be modified.* *Note: Any write operation on this bit will take effect after a delay of one APB clock cycle.*
20	BRKBID	Break1 bidirectional enable 0: Break 1 is in input mode. 1: Break 1 is in bidirectional mode.

		In bidirectional mode, the break 1 input is configured in input and open-drain output modes. Any break 1 event will generate a low level on the break 1 input IO, indicating an internal break 1 event to the external device. *Note: Once the LOCK level (LCKCFG bit in the TIMx_BKDT register) is set to '1', this bit cannot be modified.* *Note: Any write operation on this bit will take effect after a delay of one APB clock cycle.*
19	BRK2DSRM	Break 2 disarm 0: Break 2 input is armed 1: Break 2 input is disarmed When the break 2 input is invalid, this bit is automatically cleared by the hardware. BRK2DSRM is set by software to release the bidirectional output control of break 2 (open-drain output in high-impedance state), and then software polls this bit until it is reset by hardware, indicating that the break 2 event has disappeared. *Note: Any write operation on this bit will take effect after a delay of one APB clock cycle.*
18	BRKDSRM	Break 1 disarm 0: Break 1 input is armed 1: Break 1 input is disarmed When the break 1 input is invalid, this bit is automatically cleared by the hardware. BRKDSRM is set by software to release the bidirectional output control of break 1 (open-drain output in high-impedance state), and then software polls this bit until it is reset by hardware, indicating that the break 1 event has disappeared. *Note: Any write operation on this bit will take effect after a delay of one APB clock cycle.*
17	BK2EN	Break 2 enable. This bit enables the entire break 2 protection circuit. 0: Disable break 2 input 0: Enable break 2 input *Note: Break 2 can only be used when OSSI=OSSR=1.* *Note: When LOCK level 1 is set (LCKCFG bit in the TIMx_BKDT register), this bit cannot be modified.* *Note: Any write operation on this bit will take effect after a delay of one APB clock cycle.*
16	BK2P	Break 2 polarity 0: Low level of the break 2 input is valid 1: High level of the break 2 input is valid *Note: When LOCK level 1 is set (LCKCFG bit in the TIMx_BKDT register), this bit cannot be modified.* *Note: Any write operation on this bit will take effect after a delay of one APB clock cycle.*
15:14	LCKCFG[1:0]	Lock configuration. This bit provides write protection to prevent software errors. These bits offer a write protection against software errors. 00: – No write protected. 01： – LOCK Level 1 TIMx_BKDT.DTGN、TIMx_BKDT.BKEN、TIMx_BKDT.BKP、TIMx_BKDT.AOEN、TIMx_CTRL2.OIx、TIMx_CTRL2.OIxN bits enable write protection.

		10： – LOCK Level 2 Except for register write protection in LOCK Level 1 mode, TIMx_CCEN.CCxP and TIMx_CCEN.CCxNP (If the corresponding channel is configured in output mode), TIMx_BKDT.OSSR and TIMx_BKDT.OSSI bits also enable write protection 11: – LOCK Level 3 Except for register write protection in LOCK Level 2, TIMx_CCMODx.OCxMD and TIMx_CCMODx.OCxPEN bits (If the corresponding channel is configured in output mode) also enable write protection. *Note: After the system reset, the LCKCFG bit can only be written once. Once written to the TIMx_BKDT register, LCKCFG will be protected until the next reset.*
13	OSSR	Off-state Selection for Run Mode This bit is used when TIMx_BKDT.MOEN=1 and the channel is a complementary output. The OSSR bit does not exist in timer without complementary outputs. 0: When inactive, OC/OCN outputs are disabled (OC/OCN enable output signal = 0) 1: When inactive, OC/OCN outputs are enabled with their inactive level as soon as CCxEN = 1 or CCxNEN = 1. Then, OC/OCN enable output signal = 1 For more details, See Section 10.6.9 capture/compare enable registers (TIMx_CCEN)。
12	OSSI	Off-state selection for Idle mode This bit is used when TIMx_BKDT.MOEN=0 and the channels configured as outputs. 0: When inactive, OC/OCN outputs are disabled (OC/OCN enable output signal = 0) 1: When inactive, OC/OCN outputs are forced with their with their idle level as soon as CCxEN = 1 or CCxNEN = 1. Then, OC/OCN enable output signal = 1 For more details, See Section 10.6.9 capture/compare enable registers (TIMx_CCEN)。
11	BKEN	Break1 enable 0: Disable break 1 input 1: Enable break 1 input *Note: Any write to this bit requires an APB clock delay to take effect.*
10	BKP	Break1 polarity 0: Low level of the break 1 input is valid 1: High level of the break 1 input is valid *Note: Any write to this bit requires an APB clock delay to take effect.*
9	AOEN	Automatic output enable 0: Only software can set TIMx_BKDT.MOEN; 1: Software sets TIMx_BKDT.MOEN; or if the break input is not active, when the next update event occurs, hardware automatically sets TIMx_BKDT.MOEN.
8	MOEN	Main output enable This bit can be set by software or hardware depending on the TIMx_BKDT.AOEN bit, and is asynchronously cleared to '0' by hardware once the break input is active. It is only valid for channels configured as outputs. 0: OC and OCN outputs are disabled or forced to idle state. 1: OC and OCN outputs are enabled if TIMx_CCEN.CCxEN or TIMx_CCEN.CCxNEN bits are

		set.For more details, see Section 10.6.9 Capture/Compare enable registers (TIMx_CCEN)。
7:0	DTGN[7:0]	Dead-time generator setup These bits define the dead-time duration between inserted complementary outputs. The relationship between the DTGN value and the dead time is as follows: DTGN[7:5]=0xx => DT=DTGN[7:0] × T_{dtgn}， T_{dtgn} = T_{DTS}； DTGN[7:5]=10x => DT=(64+DTGN[5:0]) × T_{dtgn}， T_{dtgn} = 2 × T_{DTS}； DTGN[7:5]=110 => DT=(32+DTGN[4:0]) × T_{dtgn}， T_{dtgn} = 8 × T_{DTS}； DTGN[7:5]=111 => DT=(32+DTGN[4:0])× T_{dtgn}， T_{dtgn} = 16 × T_{DTS}；

10.6.22 Capture/Compare Register 7（TIMx_CCDAT7）

Offset address: 0x54

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCDAT7[15:0]															
rw															

Bit Field	Name	Description
31:16	Reserved	Reserved, the reset value must be maintained
15:0	CCDAT7[15:0]	Capture/Compare 7 value • CC7 channel can only configured as output: CCDAT7 contains the value to be compared with the counter TIMx_CNT, and signals are sent out on OC7 output. If the preload function is not selected in the TIMx_CCMOD3.OC7PEN bit, the written value is immediately transferred to the active register. Otherwise, this preloaded value is transferred to the active register only when an update event occurs.

10.6.23 Capture/Compare Register 8（TIMx_CCDAT8）

Offset address: 0x58

Reset value: 0x0000 0000

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="16">Reserved</td></tr>
</table>

<table>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="16">CCDAT8[15:0]</td></tr>
<tr><td colspan="16">rw</td></tr>
</table>

Bit Field	**Name**	**Description**
31:16	Reserved	Reserved, the reset value must be maintained
15:0	CCDAT8[15:0]	Capture/Compare 8 value • CC7 channel can only configured as output: CCDAT8 contains the value to be compared with the counter TIMx_CNT, and signals are sent out on OC8 output. If the preload function is not selected in the TIMx_CCMOD3.OC8PEN bit, the written value is immediately transferred to the active register. Otherwise, this preloaded value is transferred to the active register only when an update event occurs.

10.6.24 Capture/Compare Register 9（TIMx_CCDAT9）

Offset address: 0x5C
Reset value: 0x0000 0000

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="16">Reserved</td></tr>
</table>

<table>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="16">CCDAT9[15:0]</td></tr>
<tr><td colspan="16">rw</td></tr>
</table>

Bit Field	**Name**	**Description**
31:16	Reserved	Reserved, the reset value must be maintained
15:0	CCDAT9[15:0]	Capture/Compare 9 value • CC7 channel can only configured as output: CCDAT9 contains the value to be compared with the counter TIMx_CNT, and signals are sent out on OC9 output. If the preload function is not selected in the TIMx_CCMOD3.OC9PEN bit, the written value is immediately transferred to the active register. Otherwise, this preloaded value is transferred to the active register only when an update event occurs.

10.6.25 Break 1 Filter Register（TIMx_BKFR）

Offset address: 0x60

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		THRESH[5:0]						Reserved	WSIZE[5:0]						FILTEN
		rw							rw						rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															

Bit Field	Name	Description
31:30	Reserved	Reserved, the reset value must be maintained
29:24	THRESH[5:0]	Threshold number of sample logic level to be valid, maximum 63: Threshold value for a valid logic level. Within sample window if number of logic high is more than or equal to threshold value, next logic level will be logic high. Same rule applies to logic low. If both number of 1's and 0's inside window are smaller than threshold, filter output stays unchanged. Threshold value should set to more than or equal to half of window value. Recommend threshold range is: **Minimum**: 1 pre-scale clock cycle more than ceiling value of max glitch size (in pre-scale clock cycle) and need to larger than half of window size. for example, if glitch size is 3.2*(pre-scale clock period), threshold should be ┌3.2┐ = 4 + 1 =5 **Maximum**: floor value of minimum size of valid signal (in pre-scale clock cycle) and need to be smaller than window size. For example, if minimum message size is 3.2*(pre-scale clock period), threshold should be floor (3.2) = 3.
23	Reserved	Reserved, the reset value must be maintained
22:17	WSIZE[5:0]	Window size value for logic level check, maximum 63: Window size decides how many sampled values will take into consideration for getting next logic level. Build-in FIFO is 64 bits with maximum index 63 which can only set window size to be 63.
16	FILTEN	Filter enable 0: Disable filter 1: Enable filter
15:0	Reserved	Reserved, the reset value must be maintained

10.6.26 Input Selection Register（TIMx_INSEL）

Note:For detailed selection of signal sources, please refer to Chapter 10.4.

Offset address: 0x78

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				CLRS[3:0]				ITRS[3:0]				ETRS[3:0]			
				rw				rw				rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TI4S[3:0]				TI3S[3:0]				TI2S[3:0]				TI1S[3:0]			
rw				rw				rw				rw			

Bit Field	Name	Description
31:28	Reserved	Reserved, the reset value must be maintained
27:24	CLRS[3:0]	Selects tim_ocref_clr input signal 0000：tim_ocref_clr0 0001：tim_ocref_clr1 … 1111 ：tim_ocref_clr15 *Note: Once the LOCK level (LCKCFG bit in the TIMx_BKDT register) is set to '1', this bit cannot be modified.*
23:20	ITRS[3:0]	Selects tim_itr input signal 0000：tim_itr0 0001：tim_itr1 … 1111 ：tim_itr15
19:16	ETRS[3:0]	Selects tim_etr input signal 0000：tim_etr0 0001：tim_etr1 … 1111 ：tim_etr15 *Note: Once the LOCK level (LCKCFG bit in the TIMx_BKDT register) is set to '1', this bit cannot be modified.*
15:12	TI4S[3:0]	Selects tim_ti4[15:0] input signal 0000：tim_ti4_in0 0001：tim_ti4_in1 … 1111 ：tim_ti4_in15
11:8	TI3S[3:0]	Selects tim_ti3[15:0] input signal 0000：tim_ti3_in0 0001：tim_ti3_in1 … 1111 ：tim_ti3_in15

7:4	TI2S[3:0]	Selects tim_ti2[15:0] input signal 0000：tim_ti2_in0 0001：tim_ti2_in1 … 1111 ：tim_ti2_in15
3:0	TI1S[3:0]	Selects tim_ti1[15:0] input signal 0000：tim_ti1_in0 0001：tim_ti1_in1 … 1111 ：tim_ti1_in15

10.6.27 Alternate Function Option Register 1（TIMx_AF1）

Offset address: 0x7C

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		COMP4 BRKP	COMP3 BRKP	COMP2 BRKP	COMP1 BRKP	IOM BRKP	Reserved	COMP7 BRKEN	COMP6 BRKEN	COMP5 BRKEN	COMP4 BRKEN	COMP3 BRKEN	COMP2 BRKEN	COMP1 BRKEN	IOM BRKEN
		rw	rw	rw	rw	rw		rw	rw	rw	rw	rw	rw	rw	rw

Bit Field	**Name**	**Description**
31:14	Reserved	Reserved, the reset value must be maintained
13	COMP4BRKP	tim_brk_comp4 input polarity 0：tim_brk_comp4 input polarity is not inverted (active low if BKP = 0, active high if BKP = 1) 1：tim_brk_comp4 input polarity is inverted (active high if BKP = 0, active low if BKP = 1)
12	COMP3BRKP	tim_brk_comp3 input polarity 0：tim_brk_comp3 input polarity is not inverted (active low if BKP = 0, active high if BKP = 1) 1：tim_brk_comp3 input polarity is inverted (active high if BKP = 0, active low if BKP = 1)
11	COMP2BRKP	tim_brk_comp2 input polarity 0：tim_brk_comp2 input polarity is not inverted (active low if BKP = 0, active high if BKP = 1) 1：tim_brk_comp2 input polarity is inverted (active high if BKP = 0, active low if BKP = 1)
10	COMP1BRKP	tim_brk_comp1 input polarity 0：tim_brk_comp1 input polarity is not inverted (active low if BKP = 0, active high if BKP = 1) 1：tim_brk_comp1 input polarity is inverted (active high if BKP = 0, active low if BKP = 1)
9	IOMBRKP	TIMx_BKIN input polarity 0：TIMx_BKIN input polarity is not inverted (active low if BKP = 0, active high if BKP = 1) 1：TIMx_BKIN input polarity is inverted (active high if BKP = 0, active low if BKP = 1)
8	Reserved	Reserved, the reset value must be maintained

7	COMP7BRKEN	tim_brk_comp7 input enable 0：tim_brk_comp7 input disabled 1：tim_brk_comp7 input enabled
6	COMP6BRKEN	tim_brk_comp6 input enable 0：tim_brk_comp6 input disabled 1：tim_brk_comp6 input enabled
5	COMP5BRKEN	tim_brk_comp5 input enable 0：tim_brk_comp5 input disabled 1：tim_brk_comp5 input enabled
4	COMP4BRKEN	tim_brk_comp4 input enable 0：tim_brk_comp4 input disabled 1：tim_brk_comp4 input enabled
3	COMP3BRKEN	tim_brk_comp3 input enable 0：tim_brk_comp3 input disabled 1：tim_brk_comp3 input enabled
2	COMP2BRKEN	tim_brk_comp2 input enable 0：tim_brk_comp2 input disabled 1：tim_brk_comp2 input enabled
1	COMP1BRKEN	tim_brk_comp1 input enable 0：tim_brk_comp1 input disabled 1：tim_brk_comp1 input enabled
0	IOMBRKEN	TIMx_BKIN input enable 0：TIMx_BKIN input disabled 1：TIMx_BKIN input enabled

10.6.28 Alternate Function Option Register 2（TIMx_AF2）

Offset address: 0x80

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		COMP4 BRK2P	COMP3 BRK2P	COMP2 BRK2P	COMP1 BRK2P	IOM BRK2P	Reserved	COMP7 BRK2EN	COMP6 BRK2EN	COMP5 BRK2EN	COMP4 BRK2EN	COMP3 BRK2EN	COMP2 BRK2EN	COMP1 BRK2EN	IOM BRK2EN
		rw	rw	rw	rw	rw		rw	rw	rw	rw	rw	rw	rw	rw

Bit Field	**Name**	**Description**
31:14	Reserved	Reserved, the reset value must be maintained
13	COMP4BRK2P	tim_brk2_comp4 input polarity 0：tim_brk2_comp4 input polarity is not inverted (active low if BK2P = 0, active high if BK2P

		= 1) 1：tim_brk2_comp4 input polarity is inverted (active high if BK2P = 0, active low if BK2P = 1)
12	COMP3BRK2P	tim_brk2_comp3 input polarity 0：tim_brk2_comp3 input polarity is not inverted (active low if BK2P = 0, active high if BK2P = 1) 1：tim_brk2_comp3 input polarity is inverted (active high if BK2P = 0, active low if BK2P = 1)
11	COMP2BRK2P	tim_brk2_comp2 input polarity 0：tim_brk2_comp2 input polarity is not inverted (active low if BK2P = 0, active high if BK2P = 1) 1：tim_brk2_comp2 input polarity is inverted (active high if BK2P = 0, active low if BK2P = 1)
10	COMP1BRK2P	tim_brk2_comp1 input polarity 0：tim_brk2_comp1 input polarity is not inverted (active low if BK2P = 0, active high if BK2P = 1) 1：tim_brk2_comp1 input polarity is inverted (active high if BK2P = 0, active low if BK2P = 1)
9	IOMBRK2P	TIMx_BKIN2 input polarity 0：TIMx_BKIN2 input polarity is not inverted (active low if BK2P = 0, active high if BK2P = 1) 1：TIMx_BKIN2 input polarity is inverted (active high if BK2P = 0, active low if BK2P = 1)
8	Reserved	Reserved, the reset value must be maintained
7	COMP7BRK2EN	tim_brk2_comp7 input enable 0：tim_brk2_comp7 input disabled 1：tim_brk2_comp7 input enabled
6	COMP6BRK2EN	tim_brk2_comp6 input enable 0：tim_brk2_comp6 input disabled 1：tim_brk2_comp6 input enabled
5	COMP5BRK2EN	tim_brk2_comp5 input enable 0：tim_brk2_comp5 input disabled 1：tim_brk2_comp5 input enabled
4	COMP4BRK2EN	tim_brk2_comp4 input enable 0：tim_brk2_comp4 input disabled 1：tim_brk2_comp4 input enabled
3	COMP3BRK2EN	tim_brk2_comp3 input enable 0：tim_brk2_comp3 input disabled 1：tim_brk2_comp3 input enabled
2	COMP2BRK2EN	tim_brk2_comp2 input enable 0：tim_brk2_comp2 input disabled 1：tim_brk2_comp2 input enabled
1	COMP1BRK2EN	tim_brk2_comp1 input enable 0：tim_brk2_comp1 input disabled 1：tim_brk2_comp1 input enabled
0	IOMBRK2EN	TIMx_BKIN2 input enable 0：TIMx_BKIN2 input disabled 1：TIMx_BKIN2 input enabled

10.6.29 Break 2 Filter Register（TIMx_ BKFR2）

Offset address: 0x84

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		THRESH[5:0]						Reserved	WSIZE[5:0]						FILTEN
		rw							rw						rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															

Bit Field	Name	Description
31:30	Reserved	Reserved, the reset value must be maintained
29:24	THRESH2[5:0]	Threshold number of sample logic level to be valid, maximum 63: Threshold value for a valid logic level. Within sample window if number of logic high is more than or equal to threshold value, next logic level will be logic high. Same rule applies to logic low. If both number of 1's and 0's inside window are smaller than threshold, filter output stays unchanged. Threshold value should set to more than or equal to half of Window value. Recommend threshold range is: **Minimum:** 1 pre-scale clock cycle more than ceiling value of max glitch size (in pre-scale clock cycle) and need to larger than half of window size. for example, if glitch size is 3.2*(pre-scale clock period), threshold should be ┌3.2┐ = 4 + 1 =5 **Maximum:** floor value of minimum size of valid signal (in pre-scale clock cycle) and need to be smaller than window size. For example, if minimum message size is 3.2*(pre-scale clock period), threshold should be floor (3.2) = 3.
23	Reserved	Reserved, the reset value must be maintained
22:17	WSIZE2[5:0]	Window size value for logic level check, maximum 63: Window size decides how many sampled values will take into consideration for getting next logic level. Build-in FIFO is 64 bits with maximum index 63 which can only set window size to be 63.
16	FILTEN2	Filter enable 0: Disable filter 1: Enable filter
15:0	Reserved	Reserved, the reset value must be maintained

10.6.30 Sliding Filter Prescaler Register（TIMx_SLIDFPSC）

Offset address: 0x88

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SLIDFPSC [15:0]															
rw															

Bit Field	Name	Description
31:16	Reserved	Reserved, the reset value must be maintained
15:0	SLIDFPSC [15:0]	Sliding Filter Sampling Clock Prescaler Register (Prescaler): For this filter, it supports a 65535 division (16 bits). The clock prescaler scales the system clock to the sampling clock. The sampling clock determines the distance between two sampling points. Only the valid values of the sampling points are considered for logical level calculation. By configuring these bits, the sampling clock division for break 1 input sliding filter and break 2 input sliding filter can be determined, in conjunction with TIMx_BKFR, TIMx_BKFR2.

10.6.31 DMA Control Register（TIMx_DCTRL）

Offset address: 0x94

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		DBADDR[5:0]						Reserved		DBLEN[5:0]					
		rw								rw					

Bit Field	Name	Description
31:14	Reserved	Reserved, the reset value must be maintained
13:8	DBADDR[5:0]	DMA base address This bit field defines the first address where the DMA accesses the TIMx_DADDR register. When access is done through the TIMx_DADDR first time, this bit-field specifies the address you just access. And then the second access to the TIMx_DADDR, you will access the address of “DMA Base Address + 4” 00000：TIMx_CTRL1, 00001：TIMx_CTRL2, 00010：TIMx_SMCTRL,

Bit Field	Name	Description
		...... 10001：TIMx_BKDT, 10010：TIMx_DCTRL
7:6	Reserved	Reserved, the reset value must be maintained
5:0	DBLEN[5:0]	DMA burst length This bit field defines the number DMA will accesses (write/read) TIMx_DADDR register. 000000: 1 time transfer 000001: 2 time transfer 000010: 3 time transfer … 010001: 18 time transfer ...... 100010: 35 time transfer Example: Consider the following transfer: DBLEN=7, DBADDR=TIMx_CTRL1 If DBLEN=7 and DBADDR=TIMx_CTRL1 represent the address of the data to be transferred, then the address of the transfer is given by the following formula: (Address of TIMx_CTRL1) + DBADDR + (DMA index), where DMA index = DBLEN. Adding 7 to (Address of TIMx_CTRL1) + DBADDR gives the address where the data will be written or read, and the data transfer will occur in 7 registers starting from the address (Address of TIMx_CTRL1) + DBADDR. If the data is set to half-word (16 bits), the data will be transferred to all 7 registers. If the data is set to byte, the data will still be transferred to all 7 registers: the first register contains the first MSB byte, the second register contains the first LSB byte, and so on. Therefore, for the timer, the user must specify the data width to be transferred by DMA.

10.6.32 DMA Address For Full Transfer Register（TIMx_DADDR）

Offset address: 0x98

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BURST[31:0]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BURST[31:0]															
rw															

Bit Field	Name	Description
31:0	BURST[31:0]	DMA Access Buffer When a read or write operation is assigned to this register, the register located at the address range (DMA base address + DMA burst length × 4) will be accessed. DMA base address = The address of TIM_CTRL1 + TIMx_DCTRL. DBADDR * 4； DMA burst len = TIMx_DCTRL.DBLEN + 1. Example: If TIMx_DCTRL.DBLEN = 0x3(4 transfers), TIMx_DCTRL.DBADDR = 0xD (TIMx_CCDAT1), DMA data length = half word, DMA memory address = buffer address in SRAM, DMA peripheral address = TIMx_DADDR address. When an event occurs, TIMx will send requests to the DMA, and transfer data 4 times. For the first time, DMA access to the TIMx_ DADDR register will be mapped to access TIMx_CCDAT1 register; For the second time, DMA access to the TIMx_ DADDR register will be mapped to access TIMx_CCDAT2 register; … … For the fourth time, DMA access to the TIMx_ DADDR register will be mapped to access TIMx_CCDAT4 register;

11 General-Purpose Timers （GTIM1/ GTIM2/ GTIM3/ GTIM4/ GTIM5/ GTIM6/ GTIM7）

11.1 General-Purpose Timers Introduction

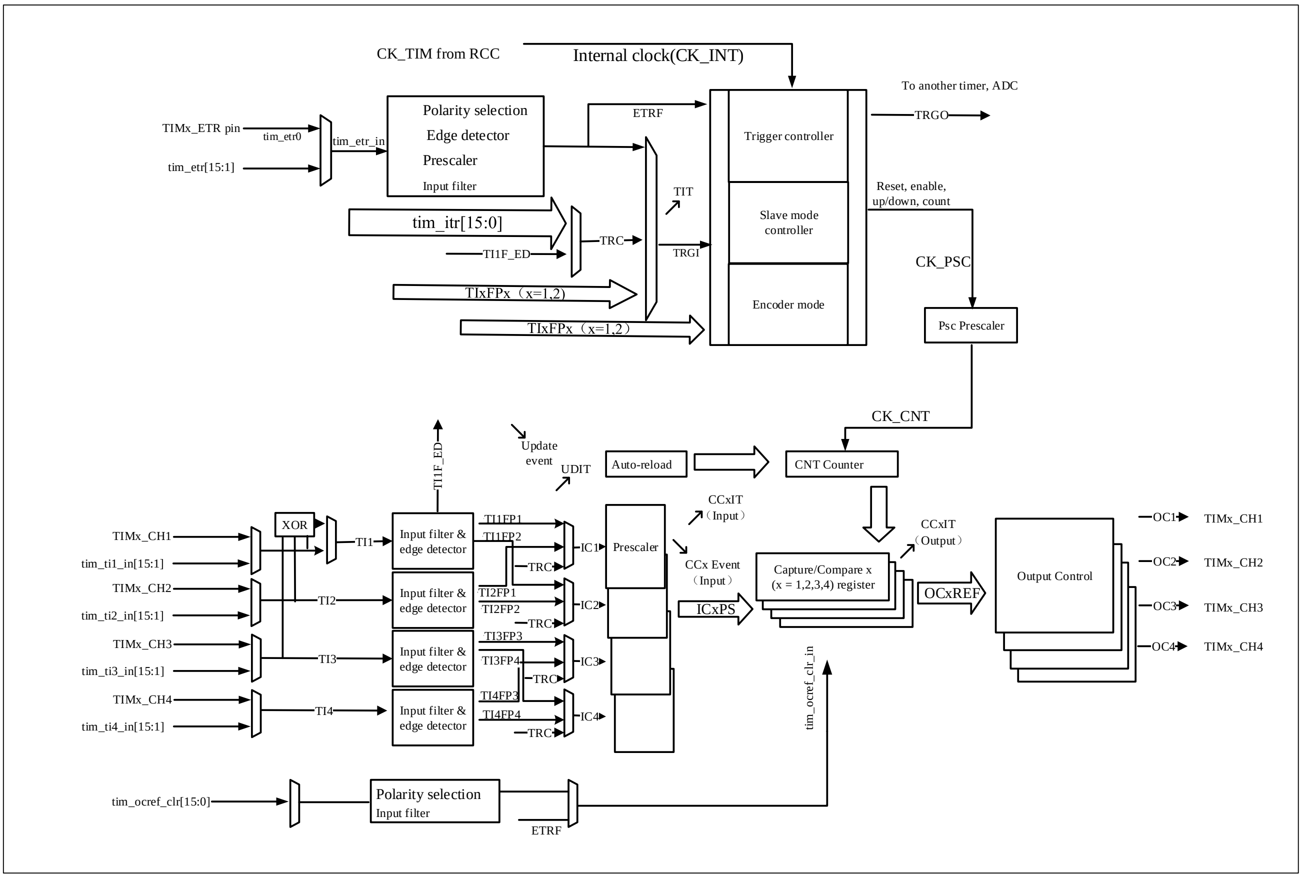
The general-purpose timers (GTIM1/GTIM2/GTIM3/GTIM4/GTIM5/GTIM6/GTIM7) is mainly used in the following scenarios: counting the input signal, measuring the pulse width of the input signal and generating the output waveform, etc.

11.2 Main Features Of GTIMx（x=1-7）

- 16-bit auto-reload counters. (It can realize up-counting, down-counting, up/down counting)
- 16-bit programmable prescaler. (The prescaler factor can be configured with any value between 1 and 65536)
- GTIMx up to 4 channels.
- Channel's working modes: PWM output, ouput compare, one-pulse mode output, input capture.
- The events that generate the interrupt/DMA are as follows:
  - Update event
  - Trigger event
  - Input capture
  - Output compare
- Timer can be controlled by external signal
- Timers can be linked together internally for timer synchronization or chaining
- Incremental (quadrature) encoder interface: used for tracking motion and resolving rotation direction and position
- Hall sensor interface: used to do three-phase motor control
- Supports capture of internal comparator output signal.

11.3 Block Diagram Of GTIMx（x=1-7）

Figure 11-1 Block Diagram Of GTIMx（x=1-7）



11.4 Pinout and Internal Signals of GTIMx（x=1-7）

The following table describes the input and output pins and signals of GTIMx

Table 11-1 GTIMx Input/Output Pins

Pin	Type	Description
TIMx_CH1 TIMx_CH2 TIMx_CH3 TIMx_CH4	Input/Output	Timer multi-purpose channel. Each channel can be used for capture, compare or generating PWM. TIM_CH1 and TIM_CH2 can also be used as an external clock (less than 1/4 of the internal clock frequency), external trigger, and quadrature encoder input. TIM_CH1, TIM_CH2, TIM_CH3 can also be used for interfacing with Hall effect sensors.
TIMx_ETR	Input	External trigger input. This input can be used as an external trigger or external clock source. If uses prescaler, the input signal TIMx_ETR can

		be a signal with a frequency higher than the system clock frequency.

Table 11-2 GTIMx Internal Input/Output Signals

Internal signal	Type	Description
tim_ti1_in[15:0] tim_ti2_in[15:0] tim_ti3_in[15:0] tim_ti4_in[15:0]	Input	Timer channels 1/2/3/4 input signals. The tim_ti1_in[15:0] and tim_ti2_in[15:0] inputs can be used for capture or as an external clock (less than 1/4 of the system clock frequency) and for quadrature encoder signals.
tim_etr[15:0]	Input	External trigger channel input signals. These inputs can be used as triggers, external clocks, or for hardware pulse-width modulation control. If uses prescaler , the input signal TIMx_ETR can be a signal with a frequency higher than the system clock frequency.
tim_itr[15:0]	Input	Internal trigger input signals. These inputs can be used in mode controllers or as input clocks (less than 1/4 of the system clock frequency).
tim_trgo	Output	Internal trigger output signals. These trigger signals can be used by other timers and/or other peripheral devices.

11.4.1 Signal Sources for tim_ti1/ tim_ti2/ tim_ti3/ tim_ti4 in GTIMx

Table 11-3 tim_ti1 Input Signal Sources

tim_ti1 inputs	**Signal Sources**						
	GTIM1	**GTIM2**	**GTIM3**	**GTIM4**	**GTIM5**	**GTIM6**	**GTIM7**
tim_ti1_in0	GTIM1_ CH1	GTIM2_ CH1	GTIM3_ CH1	GTIM4_ CH1	GTIM5_ CH1	GTIM6_ CH1	GTIM7_ CH1
tim_ti1_in1	COMP1_ OUT	COMP1_ OUT	COMP1_ OUT	COMP1_ OUT	COMP1_ OUT	COMP1_ OUT	COMP1_ OUT
tim_ti1_in2	COMP2_ OUT	COMP2_ OUT	COMP2_ OUT	COMP2_ OUT	COMP2_ OUT	COMP2_ OUT	COMP2_ OUT
tim_ti1_in3	COMP3_ OUT	COMP3_ OUT	COMP3_ OUT	COMP3_ OUT	COMP3_ OUT	COMP3_ OUT	COMP3_ OUT

tim_ti1_in4	COMP4_ OUT	COMP4_ OUT	COMP4_ OUT	COMP4_ OUT	COMP4_ OUT	COMP4_ OUT	COMP4_ OUT
tim_ti1_in5	Reserved	Reserved	Reserved	Reserved	LSI	CAN2_TMP	HSE/128
tim_ti1_in6	Reserved	Reserved	Reserved	Reserved	CAN1_TMP	RTC Wake Up	Reserved
tim_ti1_in7	Reserved	Reserved	Reserved	Reserved	CAN1_RTP	CAN2_RTP	Reserved
tim_ti1_in[15:8]	Reserved						

Table 11-4 tim_ti2 Input Signal Sources

tim_ti2 inputs	**Signal Sources**						
	GTIM1	**GTIM2**	**GTIM3**	**GTIM4**	**GTIM5**	**GTIM6**	**GTIM7**
tim_ti2_in0	GTIM1_ CH2	GTIM2_ CH2	GTIM3_ CH2	GTIM4_ CH2	GTIM5_ CH2	GTIM6_ CH2	GTIM7_ CH2
tim_ti2_in1	COMP5_ OUT	COMP5_ OUT	COMP5_ OUT	COMP5_ OUT	COMP5_ OUT	COMP5_ OUT	COMP5_ OUT
tim_ti2_in2	COMP6_ OUT	COMP6_ OUT	COMP6_ OUT	COMP6_ OUT	COMP6_ OUT	COMP6_ OUT	COMP6_ OUT
tim_ti2_in3	COMP7_ OUT	COMP7_ OUT	COMP7_ OUT	COMP7_ OUT	COMP7_ OUT	COMP7_ OUT	COMP7_ OUT
tim_ti2_in4	GTIM6_ CC1	GTIM6_ CC2	GTIM6_ CC3	GTIM6_ CC4	LSE	HSE/128	LSI
tim_ti2_in[15:5]	Reserved						

Table 11-5 tim_ti3 Input Signal Sources

tim_ti3 inputs	**Signal Sources**						
	GTIM1	**GTIM2**	**GTIM3**	**GTIM4**	**GTIM5**	**GTIM6**	**GTIM7**
tim_ti3_in0	GTIM1_	GTIM2_	GTIM3_	GTIM4_	GTIM5_	GTIM6_	GTIM7_

	CH3	CH3	CH3	CH3	CH3	CH3	CH3
tim_ti3_in1	COMP1_ OUT	COMP2_ OUT	COMP3_ OUT	COMP4_ OUT	HSE/128	LSI	Reserved
tim_ti3_in[15:2]	Reserved						

Table 11-6 tim_ti4 Input Signal Sources

tim_ti4 inputs	**Signal Sources**						
	GTIM1	**GTIM2**	**GTIM3**	**GTIM4**	**GTIM5**	**GTIM6**	**GTIM7**
tim_ti4_in0	GTIM1_ CH4	GTIM2_ CH4	GTIM3_ CH4	GTIM4_ CH4	GTIM5_ CH4	GTIM6_ CH4	GTIM7_ CH4
tim_ti3_in1	COMP1_ OUT	COMP2_ OUT	COMP3_ OUT	COMP4_ OUT	Reserved	MCO	MCO2
tim_ti3_in2	Reserved	Reserved	Reserved	Reserved	RTC Wake Up	Reserved	Reserved
tim_ti4_in[15:3]	Reserved						

11.4.2 Signal Sources for tim_itr in GTIMx

Table 11-7 tim_itr Input Signal Sources

GTIMx	**GTIM1**	**GTIM2**	**GTIM3**	**GTIM4**	**GTIM5**	**GTIM6**	**GTIM7**
tim_itr0	ATIM1_ TRGO	ATIM1_ TRGO	ATIM1_ TRGO	ATIM1_ TRGO	ATIM1_ TRGO	ATIM1_ TRGO	ATIM1_ TRGO
tim_itr1	Reserved	GTIM1_ TRGO	GTIM1_ TRGO	GTIM1_ TRGO	GTIM1_ TRGO	GTIM1_ TRGO	GTIM1_ TRGO
tim_itr2	GTIM2_ TRGO	Reserved	GTIM2_ TRGO	GTIM2_ TRGO	GTIM2_ TRGO	GTIM2_ TRGO	GTIM2_ TRGO
tim_it	GTIM3_	GTIM3_	Reserved	GTIM3_	GTIM3_	GTIM3_	GTIM3_

r3	TRGO	TRGO		TRGO	TRGO	TRGO	TRGO
tim_itr4	GTIM4_TRGO	GTIM4_TRGO	GTIM4_TRGO	Reserved	GTIM4_TRGO	GTIM4_TRGO	GTIM4_TRGO
tim_itr5	ATIM2_TRGO	ATIM2_TRGO	ATIM2_TRGO	ATIM2_TRGO	ATIM2_TRGO	ATIM2_TRGO	ATIM2_TRGO
tim_itr6	GTIM8_TRGO	GTIM8_TRGO	GTIM8_TRGO	GTIM8_TRGO	GTIM8_TRGO	GTIM8_TRGO	GTIM8_TRGO
tim_itr7	GTIM9_TRGO	GTIM9_TRGO	GTIM9_TRGO	GTIM9_TRGO	GTIM9_TRGO	GTIM9_TRGO	GTIM9_TRGO
tim_itr8	GTIM10_TRGO	GTIM10_TRGO	GTIM10_TRGO	GTIM10_TRGO	GTIM10_TRGO	GTIM10_TRGO	Reserved
tim_itr9	ATIM3_TRGO	ATIM3_TRGO	ATIM3_TRGO	ATIM3_TRGO	ATIM3_TRGO	ATIM3_TRGO	Reserved
tim_itr10	SHRTIM1_OUT_SYNC2	SHRTIM1_OUT_SYNC2	SHRTIM1_OUT_SYNC2	SHRTIM1_OUT_SYNC2	SHRTIM1_OUT_SYNC2	SHRTIM1_OUT_SYNC2	SHRTIM1_OUT_SYNC2
tim_itr11	USB SOF SYNC	Reserved	Reserved	Reserved	CAN1_SOC	CAN2_SOC	ETH-PTP
tim_itr12	GTIM5_TRGO	GTIM5_TRGO	GTIM5_TRGO	GTIM5_TRGO	Reserved	GTIM5_TRGO	GTIM5_TRGO
tim_itr13	GTIM6_TRGO	GTIM6_TRGO	GTIM6_TRGO	GTIM6_TRGO	GTIM6_TRGO	Reserved	GTIM6_TRGO
tim_itr14	GTIM7_TRGO	GTIM7_TRGO	GTIM7_TRGO	GTIM7_TRGO	GTIM7_TRGO	GTIM7_TRGO	Reserved
tim_itr15	Reserved						

11.4.3 Signal Sources for tim_etr in GTIMx

Table 11-8 tim_etr Input Signal Sources

GTIMx	GTIM1	GTIM2	GTIM3	GTIM4	GTIM5	GTIM6	GTIM7
tim_etr0	GTIM1_ETR	GTIM2_ETR	GTIM3_ETR	GTIM4_ETR	GTIM5_ETR	GTIM6_ETR	GTIM7_ETR
tim_etr1	COMP1_OUT	COMP1_OUT	COMP1_OUT	COMP1_OUT	COMP1_OUT	COMP1_OUT	COMP1_OUT
tim_etr2	COMP2_OUT	COMP2_OUT	COMP2_OUT	COMP2_OUT	COMP2_OUT	COMP2_OUT	COMP2_OUT
tim_etr3	COMP3_OUT	COMP3_OUT	COMP3_OUT	COMP3_OUT	COMP3_OUT	COMP3_OUT	COMP3_OUT
tim_etr4	COMP4_OUT	COMP4_OUT	COMP4_OUT	COMP4_OUT	COMP4_OUT	COMP4_OUT	COMP4_OUT
tim_etr5	COMP5_OUT	COMP5_OUT	COMP5_OUT	COMP5_OUT	COMP5_OUT	COMP5_OUT	COMP5_OUT
tim_etr6	COMP6_OUT	COMP6_OUT	COMP6_OUT	COMP6_OUT	COMP6_OUT	COMP6_OUT	COMP6_OUT
tim_etr7	COMP7_OUT	COMP7_OUT	COMP7_OUT	COMP7_OUT	COMP7_OUT	COMP7_OUT	COMP7_OUT
tim_etr8	GTIM2_ETR	GTIM1_ETR	GTIM2_ETR	GTIM1_ETR	GTIM6_ETR	GTIM5_ETR	GTIM5_ETR
tim_etr9	GTIM8_ETR	GTIM3_ETR	GTIM4_ETR	GTIM2_ETR	GTIM7_ETR	GTIM7_ETR	GTIM6_ETR
tim_etr10	ADC1_AWD3	GTIM9_ETR	GTIM10_ETR	GTIM8_ETR	GTIM9_ETR	GTIM10_ETR	GTIM8_ETR
tim_etr11	LSE	ADC2_AWD1	ADC3_AWD1	ADC4_AWD1	ADC1_AWD1	ADC2_AWD1	ADC3_AWD1

tim_etr12	ADC1_ AWD1	ADC2_ AWD2	ADC3_ AWD2	ADC4_ AWD2	ADC1_ AWD2	ADC2_ AWD2	ADC3_ AWD2
tim_etr13	ADC1_ AWD2	ADC2_ AWD3	ADC3_ AWD3	ADC4_ AWD3	ADC1_ AWD3	ADC2_ AWD3	ADC3_ AWD3
tim_etr_in[15:14]	Reserved						

11.4.4 Signal Source for tim_ocref_clr in GTIMx

Table 11-9 tim_ocref_clr Input Signal Sources

OCREF clear signal	**GTIMx OCREF clear signal assignment**						
	GTIM1	**GTIM2**	**GTIM3**	**GTIM4**	**GTIM5**	**GTIM6**	**GTIM7**
tim_ocref_clr0	COMP1_ OUT	COMP1_ OUT	COMP1_ OUT	COMP1_ OUT	COMP1_ OUT	COMP1_ OUT	COMP1_ OUT
tim_ocref_clr1	COMP2_ OUT	COMP2_ OUT	COMP2_ OUT	COMP2_ OUT	COMP2_ OUT	COMP2_ OUT	COMP2_ OUT
tim_ocref_clr2	COMP3_ OUT	COMP3_ OUT	COMP3_ OUT	COMP3_ OUT	COMP3_ OUT	COMP3_ OUT	COMP3_ OUT
tim_ocref_clr3	COMP4_ OUT	COMP4_ OUT	COMP4_ OUT	COMP4_ OUT	COMP4_ OUT	COMP4_ OUT	COMP4_ OUT
tim_ocref_clr4	COMP5_ OUT	COMP5_ OUT	COMP5_ OUT	COMP5_ OUT	COMP5_ OUT	COMP5_ OUT	COMP5_ OUT
tim_ocref_clr5	COMP6_ OUT	COMP6_ OUT	COMP6_ OUT	COMP6_ OUT	COMP6_ OUT	COMP6_ OUT	COMP6_ OUT
tim_ocref_clr6	COMP7_ OUT	COMP7_ OUT	COMP7_ OUT	COMP7_ OUT	COMP7_ OUT	COMP7_ OUT	COMP7_ OUT
tim_ocref_in[15:7]	Reserved						

11.5 GTIMx（x=1-7）Function Description

11.5.1 Time-Base Unit

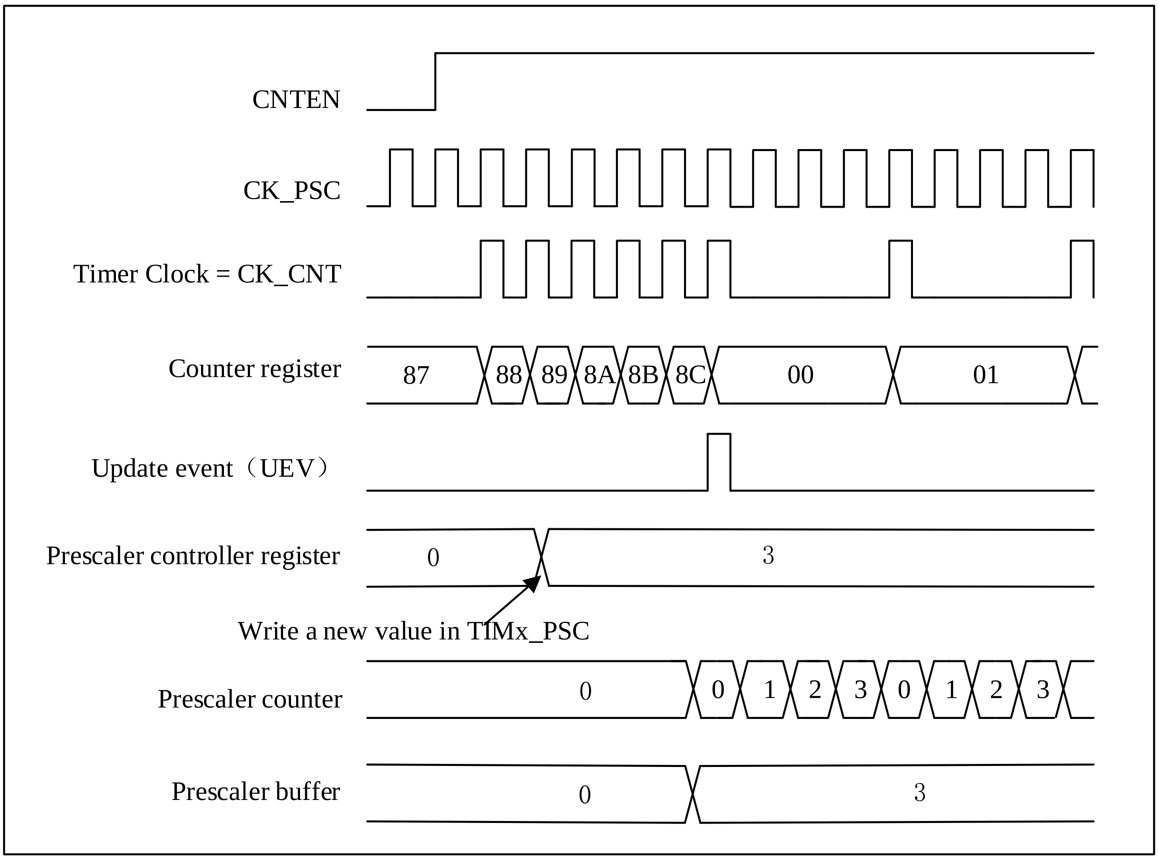
The time-base unit mainly includes: prescaler, counter and auto-reload. When the time base unit is working, the software can read and write the corresponding registers (TIMx_PSC, TIMx_CNT and TIMx_AR) at any time.

Depending on the setting of the auto-reload preload enable bit (TIMx_CTRL1.ARPEN), the value of the preload register is transferred to the shadow register immediately or at each update event UEV. An update event is generated when the counter reaches the overflow/underflow condition and it can be generated by software when TIMx_CTRL1.UPDIS=0. The counter CK_CNT is valid only when the TIMx_CTRL1.CNTEN bit is set. The counter starts counting one clock cycle after the TIMx_CTRL1.CNTEN bit is set.

11.5.1.1 Prescaler Description

The TIMx_PSC register consists of a 16-bit counter that can be used to divide the counter clock frequency by any factor between 1 and 65536. It can be changed on the fly as it is buffered. The prescaler value is only taken into account at the next update event.

Figure 11-2 Counter Timing Diagram With Prescaler Division Change From 1 To 4



11.5.2 Counter Mode

11.5.2.1 Up-counting mode

In up-counting mode, the counter will count from 0 to the value of the register TIMx_AR, then it resets to 0. And a counter overflow event is generated.

If the TIMx_CTRL1.UPRS bit (select update request) and the TIMx_EVTGEN.UDGN bit are set, an update event (UEV) will generate And TIMx_STS.UDITF will not be set by hardware, therefore, no update interrupts or update DMA requests are generated. This setting is used in scenarios where you want to clear the counter but do not want to generate an update interrupt.

Depending on the update request source is configured in the TIMx_CTRL1.UPRS. When an update event occurs, TIMx_STS.UDITF is set, all registers are updated:

- Update auto-reload shadow registers with preload value(TIMx_AR), when TIMx_CTRL1.ARPEN = 1.
- The prescaler shadow register is reloaded with the preload value(TIMx_PSC).

To avoid updating the shadow registers when new values are written to the preload registers, you can disable the update by setting TIMx_CTRL1.UPDIS=1.

When an update event occurs, the counter will still be cleared and the prescaler counter will also be set to 0 (but the prescaler value will remain unchanged).

The figure below shows some examples of the counter behavior and the update flags for different division factors in the up-counting mode.

Figure 11-3 Timing Diagram Of Up-Counting. When The Internal Clock Divider Factor = 2/N

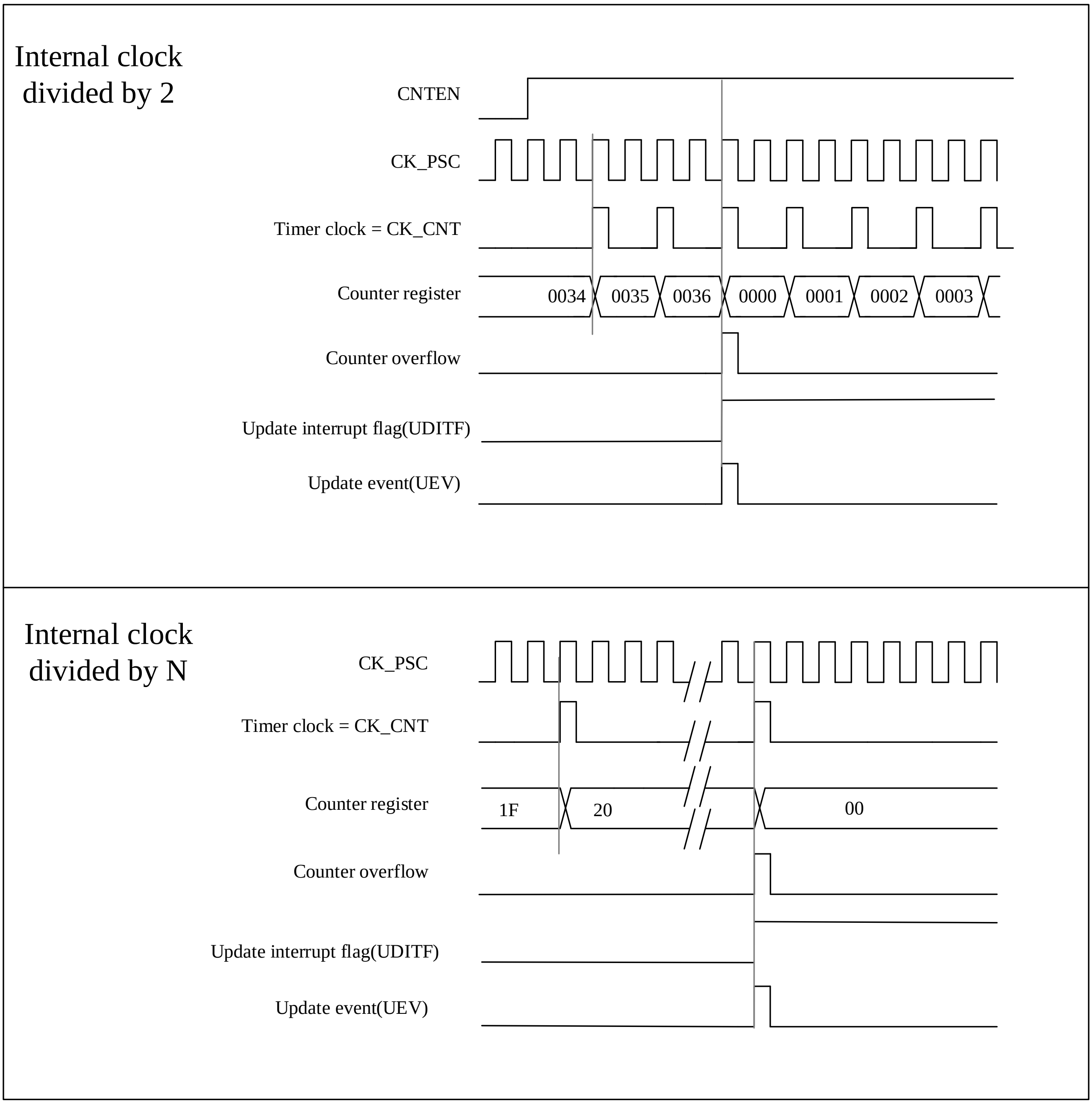
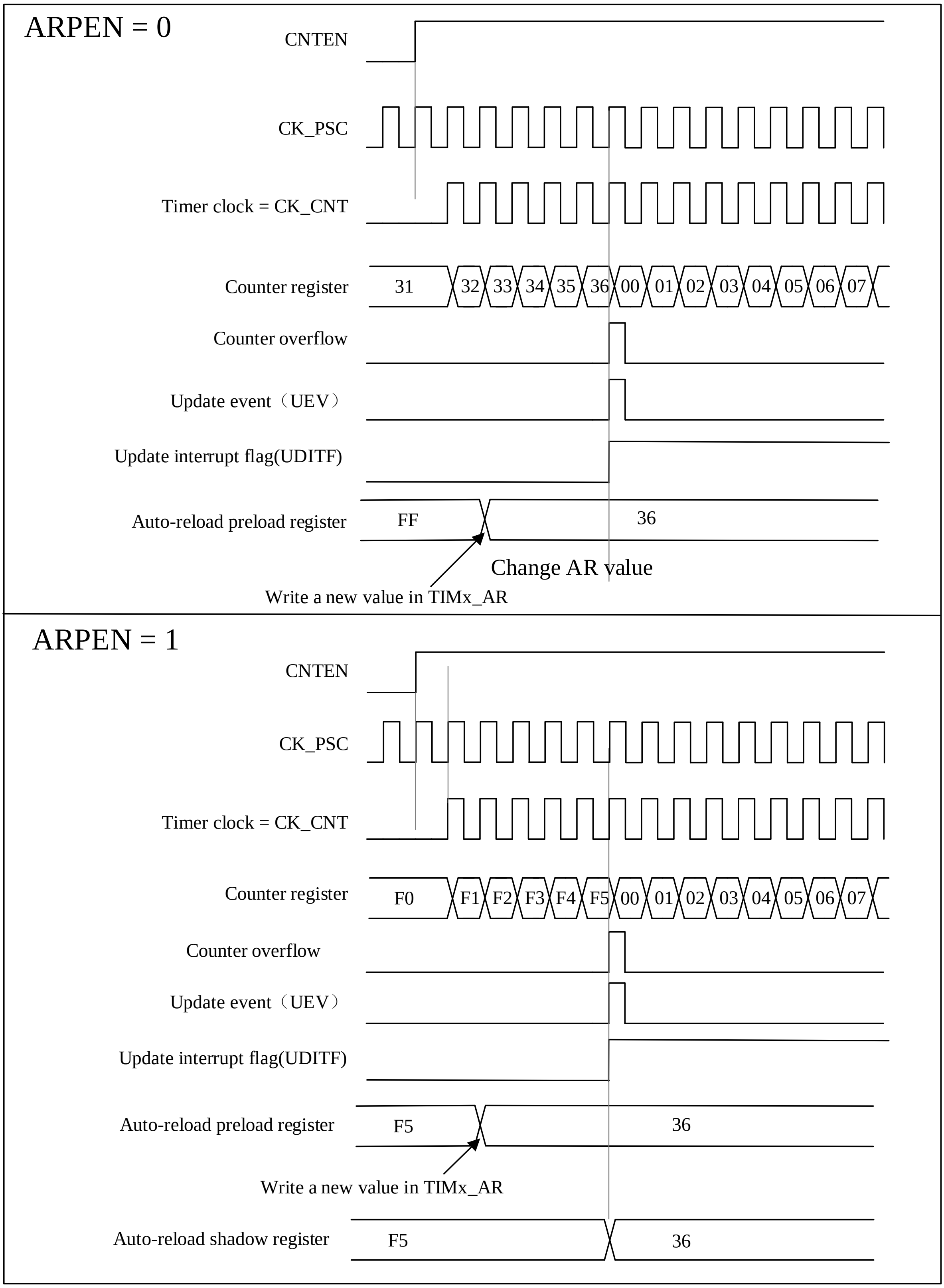


Figure 11-4 Timing Diagram Of The Up-Counting, When ARPEN=0/1 Generates an Update Event



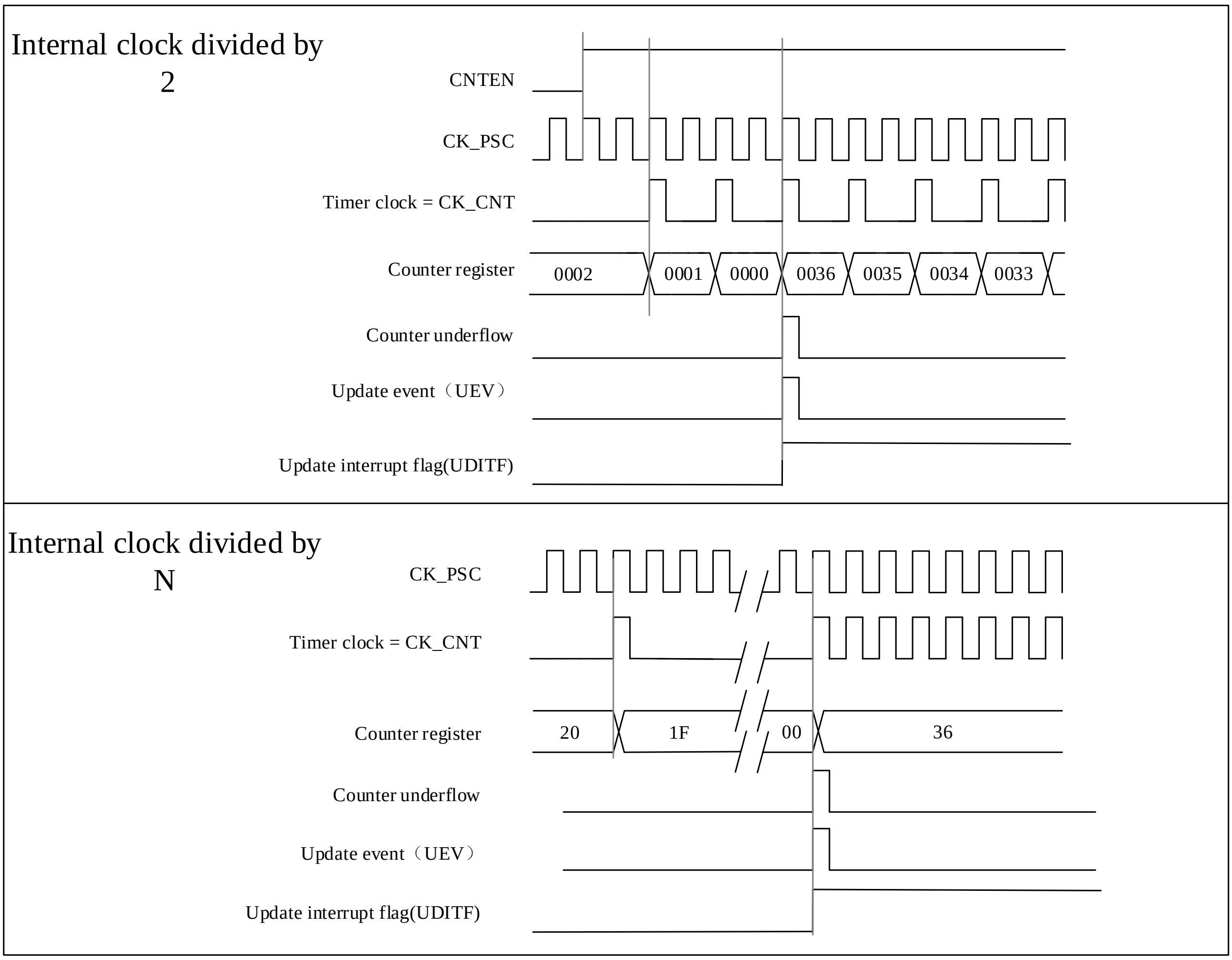
11.5.2.2 Down-counting mode

In down-counting mode, the counter will decrement from the value of the register TIMx_AR to 0, then restart from the auto-reload value and generate a counter underflow event.

The process of configuring update events and updating registers in down-counting mode is the same as in up-counting mode, see Section11.5.2.1.

The figure below shows some examples of the counter behavior and the update flags for different division factors in the down-counting mode.

Figure 11-5 Timing Diagram Of The Down-Counting With Internal Clock Divided Factor = 2/N



11.5.2.3 Center-aligned mode

In center-aligned mode, the counter increments from 0 to the value (TIMx_AR) – 1, a counter overflow event is generated. It then counts down from the auto-reload value (TIMx_AR) to 1 and generates a counter underflow event. Then the counter resets to 0 and starts counting up again.

In this mode, the TIMx_CTRL1.DIR direction bits have no effect and the count direction is updated and specified by hardware. Center-aligned mode is valid when the TIMx_CTRL1. CAMSEL bit is not equal to "00".

The update events can be generated each time the counter overflows and each time the counter underflows. Alternatively, an update event can also be generated by setting the TIMx_EVTGEN. UDGN bit (either by software or using a slave mode controller). In this case, the counter restarts from 0, as does the prescaler's counter.

Note: if the update source is a counter overflow, auto-reload update before reloading the counter.

Figure 11-6 Timing Diagram Of The Center-Aligned With Internal Clock Divided Factor =2/N

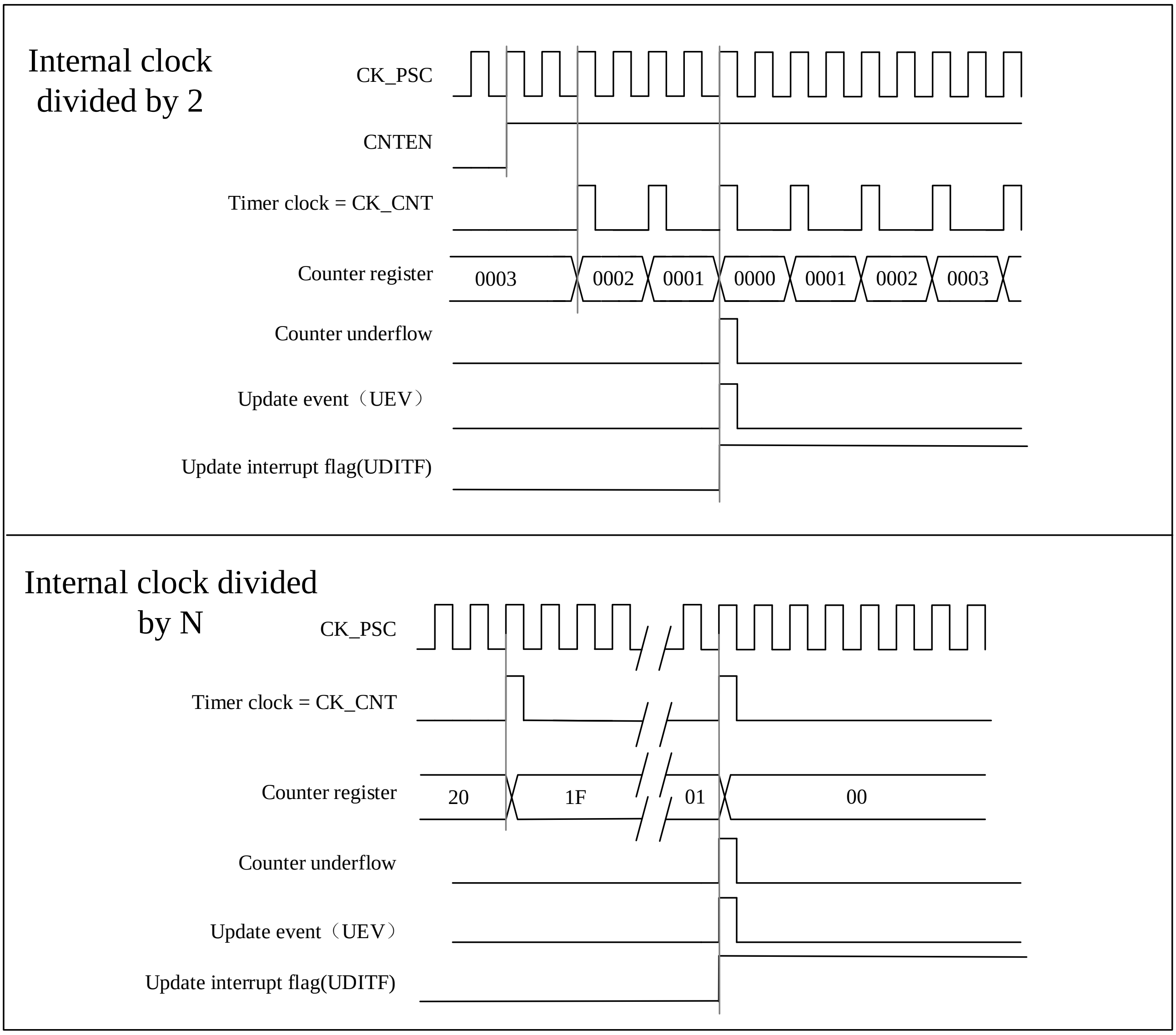
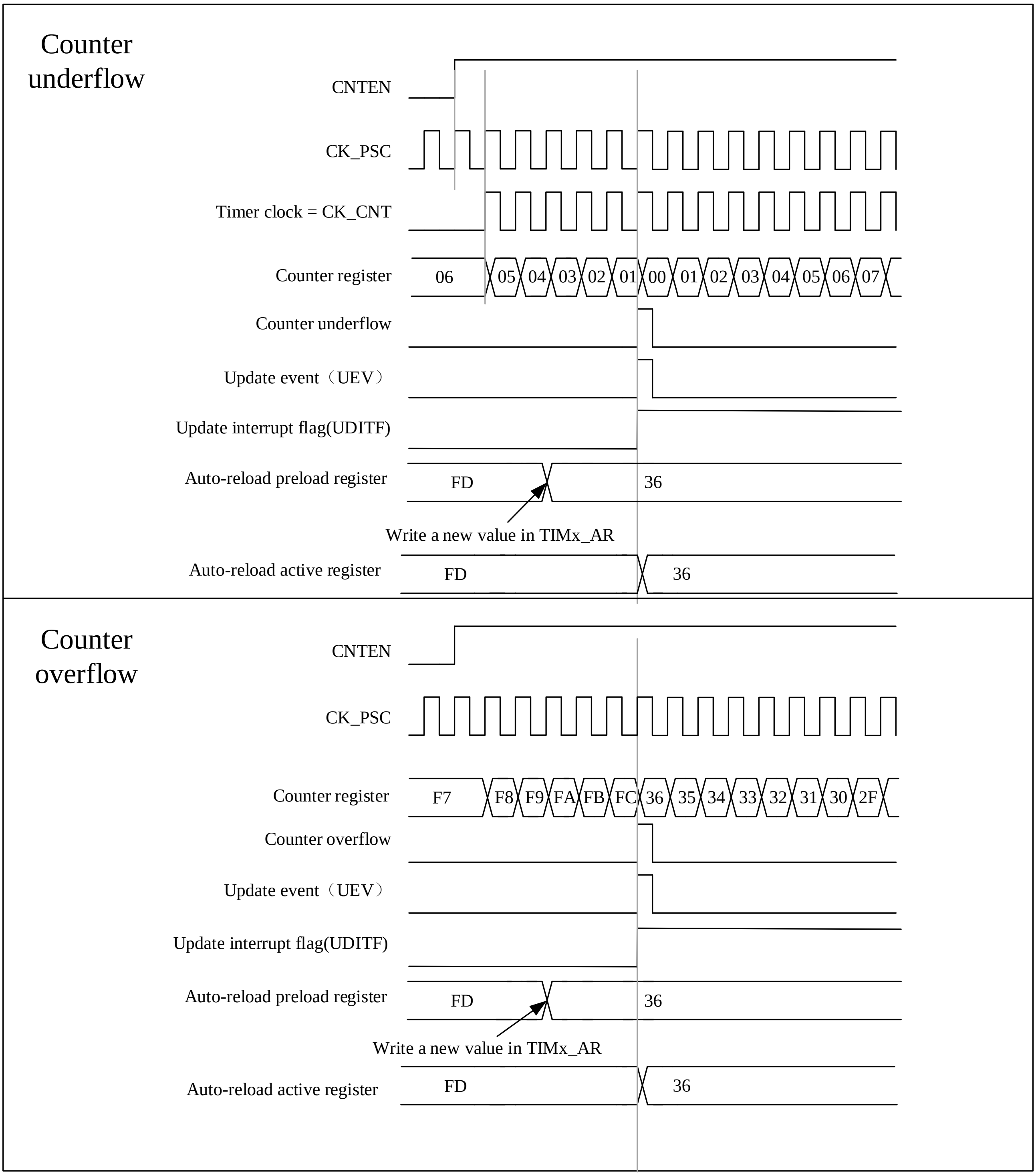


Figure 11-7 A Center-Aligned Sequence Diagram That Includes Counter Overflows And Underflows (ARPEN = 1)



11.5.3 Clock Selection

- The internal clock of timers：CK_INT

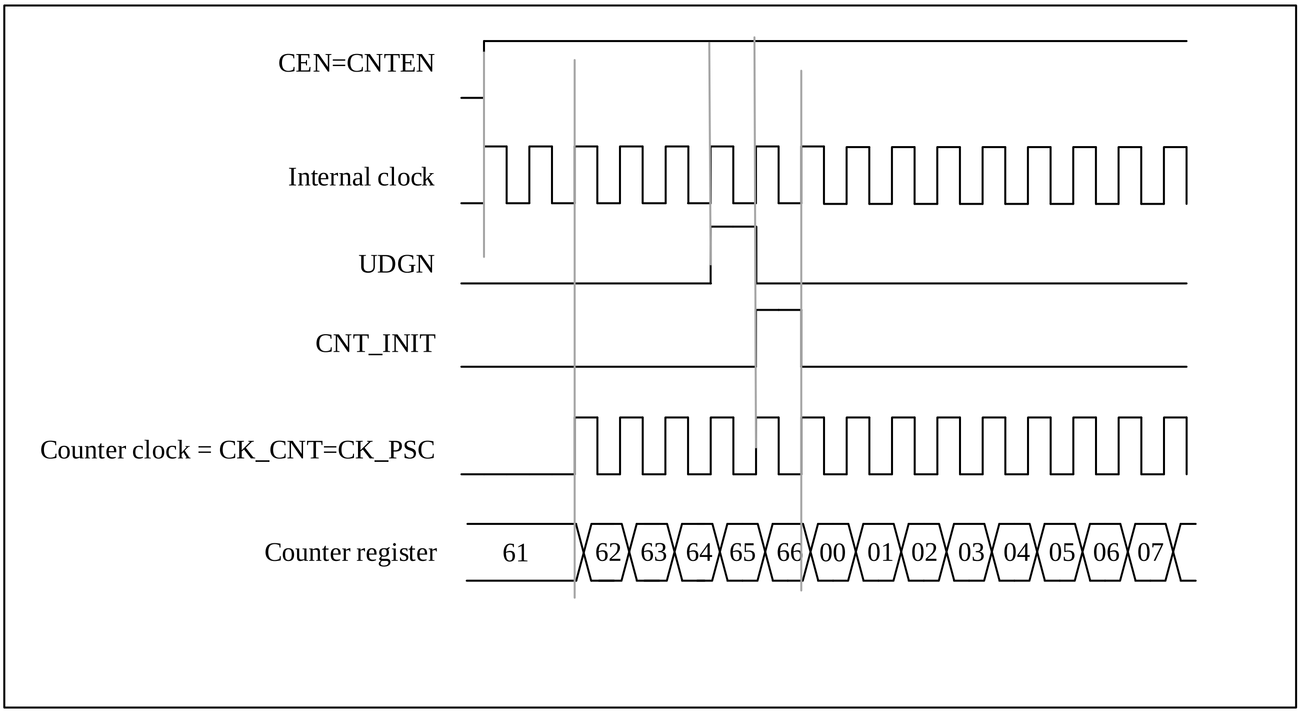
Note：The maximum operating clock for GTIM1-7 is 180MHz. Therefore, when the AHB is greater than 180MHz and GTIM1-7 need to be used, the APB1 cannot be divided by 1 or 2. For example, if the AHB frequency is 240MHz *and GTIM1 needs to be used, then the APB1 can only be divided by 4 to achieve 60MHz, resulting in a GTIM1 operating clock of 120MHz.*

- Two kinds of external clock mode:
  - external input pin
  - external trigger input ETR
- Internal trigger input(ITRx): one timer is used as a prescaler for another timer

11.5.3.1 Internal clock source(CK_INT)

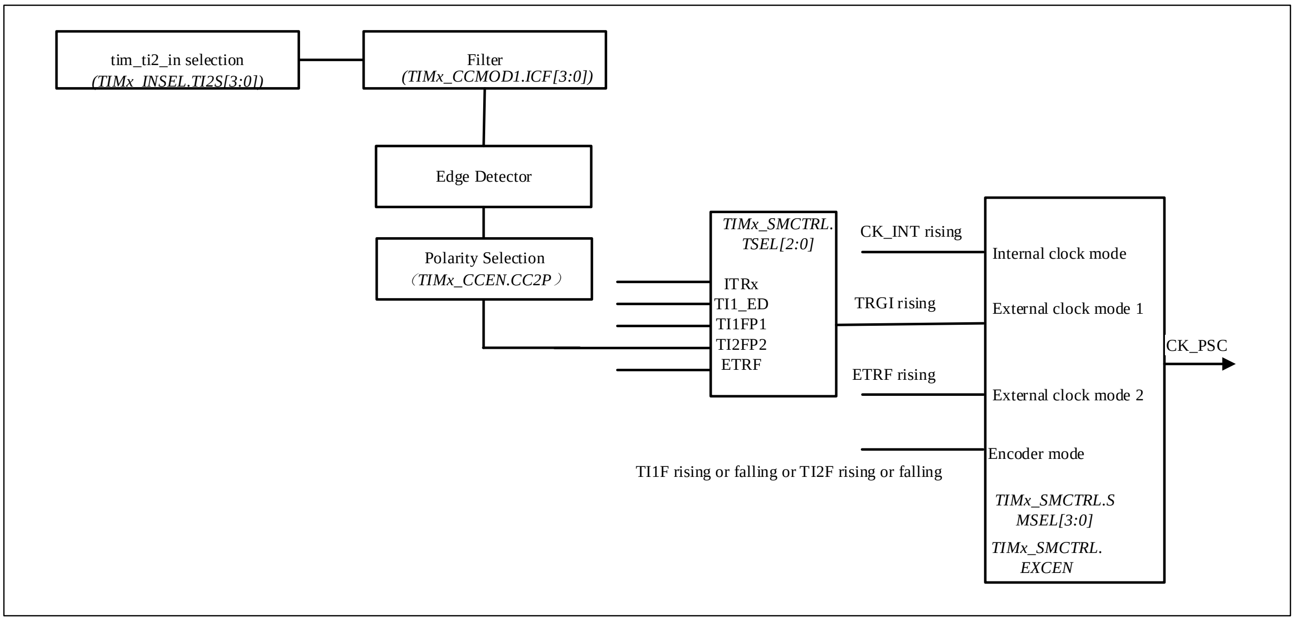
When the TIMx_SMCTRL.SMSEL is equal to “0000”, the slave mode controller is disabled. The three control bits (TIMx_CTRL1.CNTEN、TIMx_CTRL1. DIR、TIMx_EVTGEN. UDGN) can only be changed by software (except TIMx_EVTGEN. UDGN, which remains cleared automatically). It is provided that the TIMx_CTRL1.CNTEN bit is written as' 1 ' by software, the clock source of the prescaler is provided by the internal clock CK_INT.

Figure 11-8 Control Circuit In Normal Mode With Internal Clock Divided By 1



11.5.3.2 External clock source mode 1

Figure 11-9 TI2 External Clock Connection Example



This mode is selected by configuring TIMx_SMCTRL.SMSEL=0111. The counter can be configured to count on the rising or falling edge of the clock at the selected input.

For example, to configure up-counting mode to count on the rising edge of the clock at the T12 input, the configuration steps are as follows:

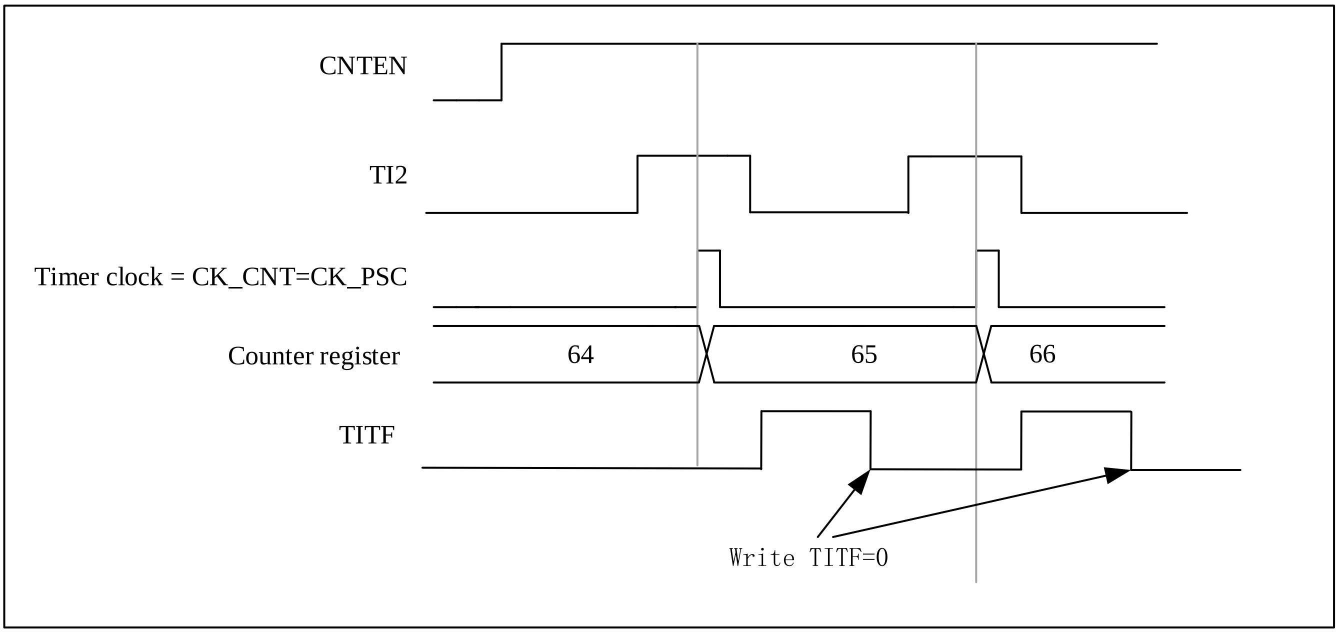
- Configure TIMx_CCMOD1.CC2SEL equal to ‘01’, CC2 channel is configured as input, IC2 is mapped to TI2
- Configure TIMx_CCEN.CC2P equal to ‘0’, select clock rising edge polarity
- To select input filter bandwidth by configuring TIMx_CCMOD1.IC2F[3:0] (if filter is not needed, keep IC2F bit at ‘0000’)
- Configure TIMx_SMCTRL.SMSEL equal to ‘0111’, select timer external clock mode 1
- Configure TIMx_SMCTRL.TSEL equal to ‘110’, select TI2 as the trigger input source
- Configure TIMx_CTRL1.CNTEN equal to ‘1’ to start the counter

Note：The capture prescaler is not used for triggering, so it does not need to be configured

When the rising edge of the timer clock occurs at TI2=1, the counter counts once and the TIMx_STS .TITF flag is pulled high.

The delay between the rising edge of TI2 and the actual clock of the counter depends on the resynchronization circuit at the input of TI2.

Figure 11-10 Control Circuit In External Clock Mode 1

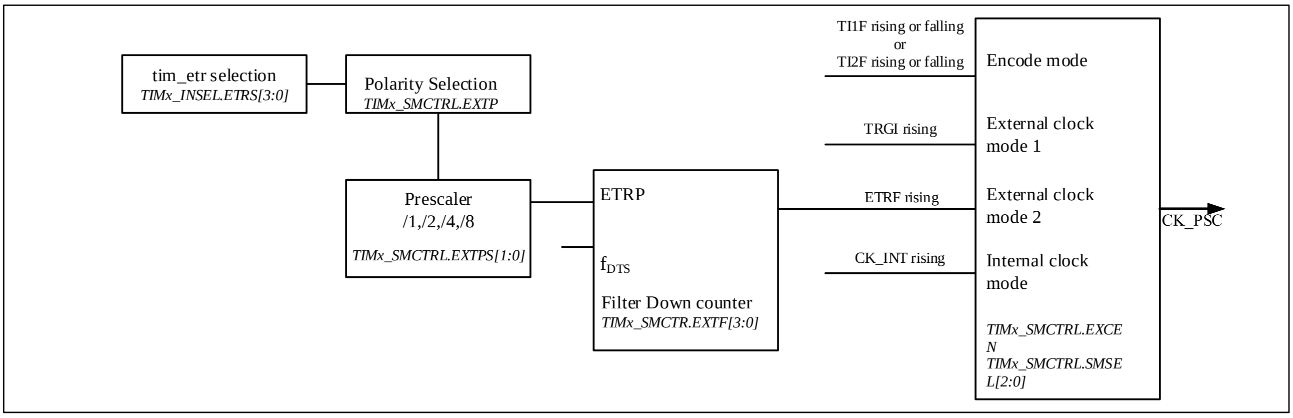


11.5.3.3 External clock source mode 2

This mode is selected by TIMx_SMCTRL .EXCEN equal to 1. The counter can count on every rising or falling edge of the external trigger input ETR.

The following figure is a schematic diagram of the external trigger input module in External clock source mode 2

Figure 11-11 External Trigger Input Block Diagram

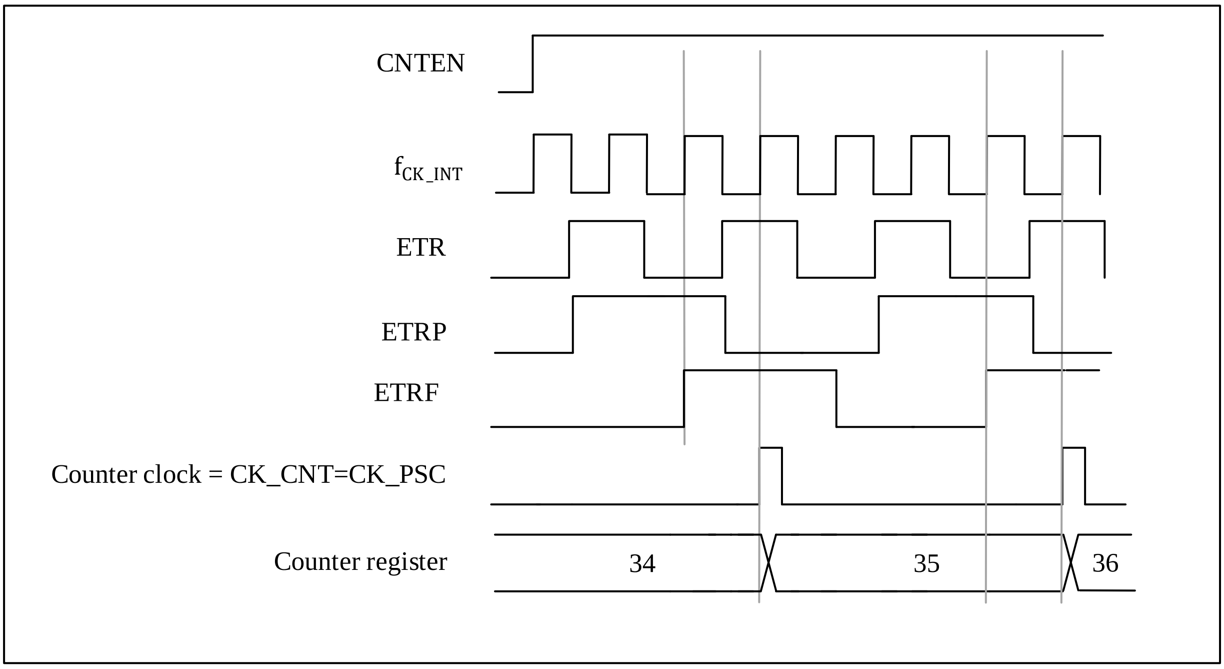


For example, use the following configuration steps to make the up counter count every 2 rising edges on ETR.

- Since no filter is needed in this case, make TIMx_SMCTRL .EXTF[3:0] equal to ‘0000’
- Configure the prescaler by making TIMx_SMCTRL.EXTPS[1:0] equal to ‘01’
- Select the polarity on ETR pin by setting TIMx_SMCTRL.EXTP equal to ‘0’, The rising edge of ETR is valid
- External clock mode 2 is selected by setting TIMx_SMCTRL .EXCEN equal to ‘1’
- Turn on the counter by setting TIMx_CTRL1. CNTEN equal to ‘1’

The counter counts every 2 rising edges of ETR. The delay between the rising edge of ETR and the actual clock to the counter is due to a resynchronization circuit on the ETRP signal.

Figure 11-12 Control Circuit In External Clock Mode 2

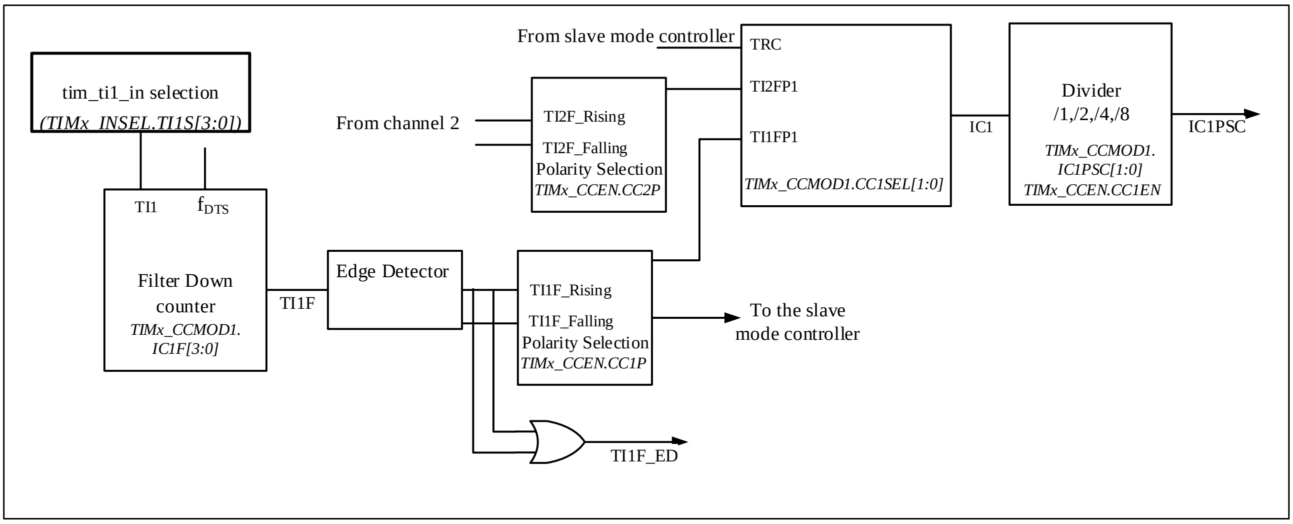


11.5.4 Capture/Compare Channels

Capture/Compare channels include Capture/Compare registers and shadow registers. The input section consists of digital filters, multiplexers and prescalers. The output section includes comparators and output controls.

The input signal TIx is sampled and filtered to generate the signal TIxF. A signal (TIxF_rising or TIxF_falling) is then generated by the edge detector of the polarity select function, the polarity of which is selected by the TIMx_CCEN.CCxP bits. This signal can be used as a trigger input for the slave mode controller. At the same time, the signal ICx is sent to the capture register after prescale. The following figure shows a block diagram of a capture/compare channel.

Figure 11-13 Capture/Compare Channel (Example: Channel 1 Input Stage)



The output part generates an intermediate waveform OCxRef (active high) as reference. The polarity acts at the end of the chain.

Figure 11-14 Capture/Compare Channel 1 Main Circuit

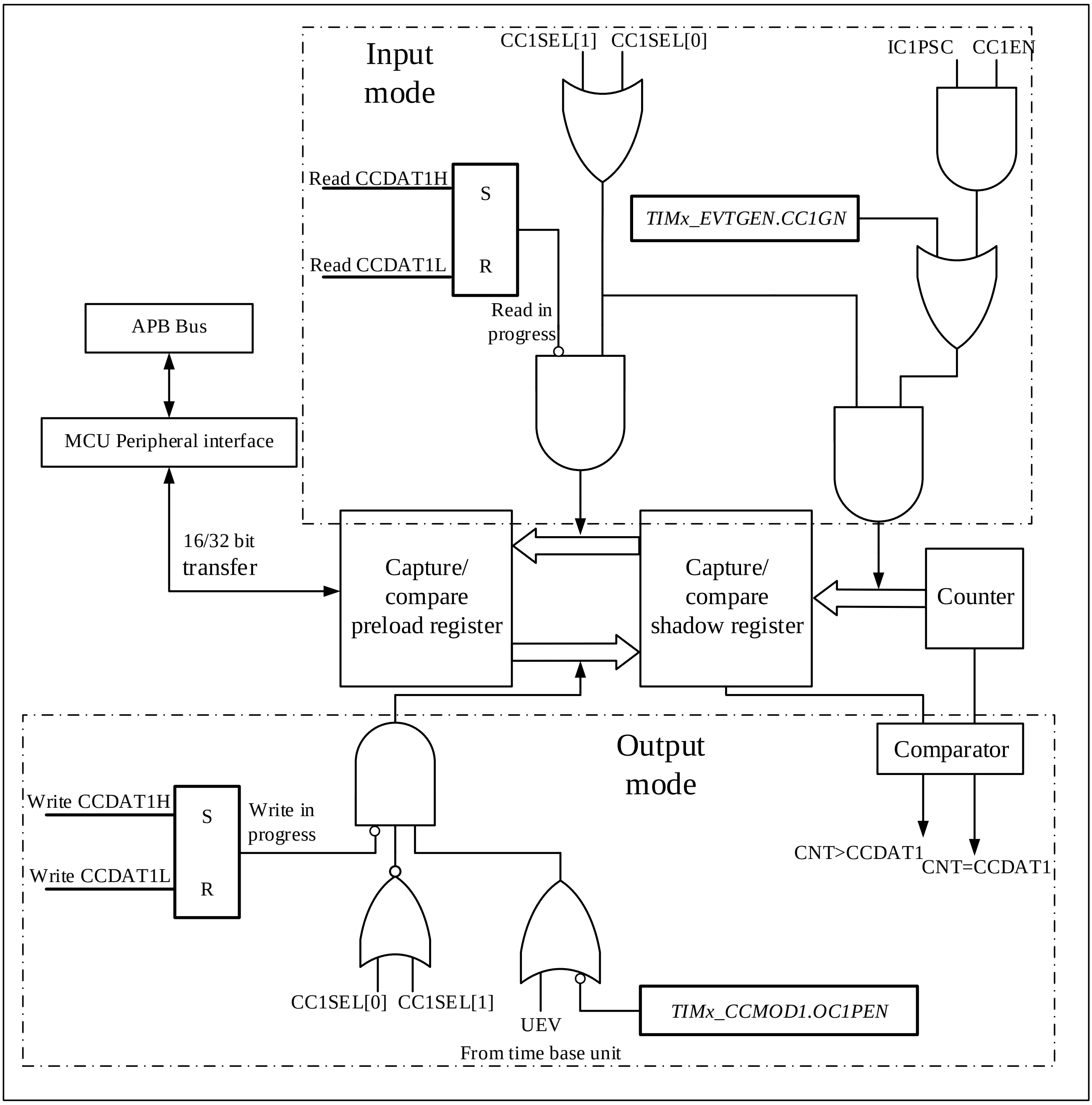
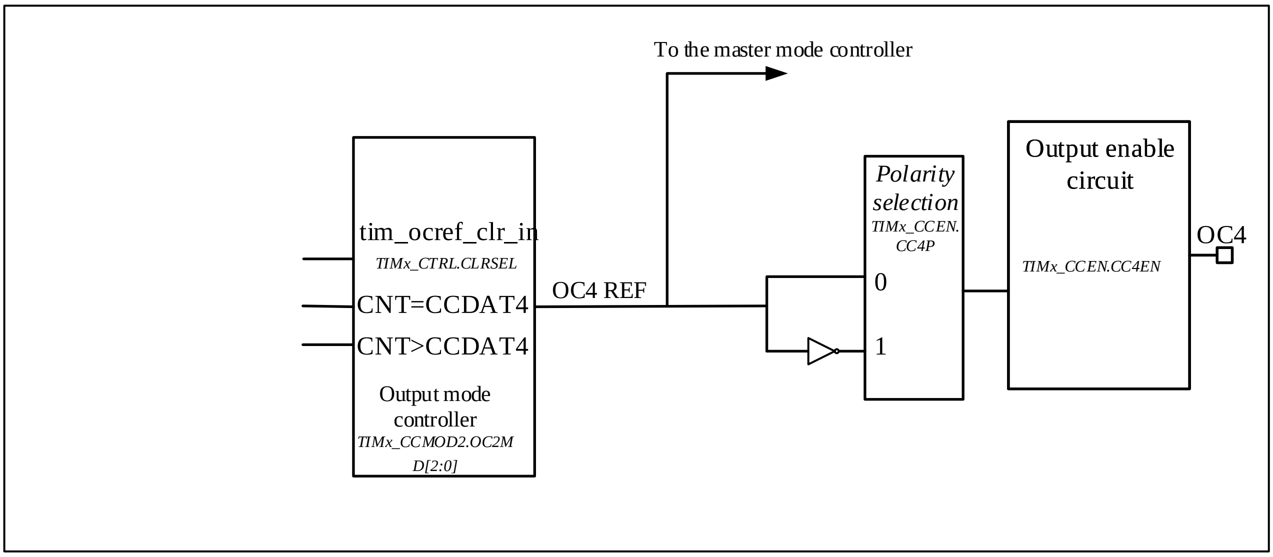


Figure 11-15 Output Part Of Channelx (x=1,2,3,4; Take Channel 4 As An Example)



Reads and writes always access preloaded registers when capturing/comparing. The two specific working processes are as follows:

In capture mode, the capture is actually done in the shadow register, and then the value in the shadow register is copied into the preload register.

In compare mode, as opposed to capture mode, the value of the preload register is copied into the shadow register, which is compared with the counter.

11.5.5 Input Capture Mode

In capture mode, the TIMx_CCDATx registers are used to latch the counter value after the ICx signal detects.

There is a capture interrupt flag TIMx_STS.CCxITF, which can issue an interrupt or DMA request if the corresponding interrupt enable is pulled high.

The TIMx_STS. CCxITF bit is set by hardware when a capture event occurs and is cleared by software or by reading the TIMx_CCDATx register.

The overcapture flag TIMx_STS.CCxOCF is set equal to 1 when the counter value is captured in the TIMx_CCDATx register and TIMx_STS.CC1ITF is already pulled high. Unlike the former, TIMx_STS.CCxOCF is cleared by writing 0 to it.

To achieve a rising edge of the TI1 input to capture the counter value into the TIMx_CCDAT1 register, the configuration flow is as follows:

- To select a valid input:

  Configure TIMx_CCMOD1.CC1SEL to ‘01’. At this time, the input is the CC1 channel, and IC1 is mapped to TI1.

- The duration of the input filter required for programming：

  Define the sampling frequency of the TI1 input and the length of the digital filter by configuring the

TIMx_CCMODx.ICxF bits. Example: If the input signal jitters up to 5 internal clock cycles, we must choose a filter duration longer than these 5 clock cycles. When 8 consecutive samples (sampled at f_{DTS} frequency) with the new level are detected, we can validate the transition on TI1. Then configure TIMx_CCMOD1.IC1F to '0011'.

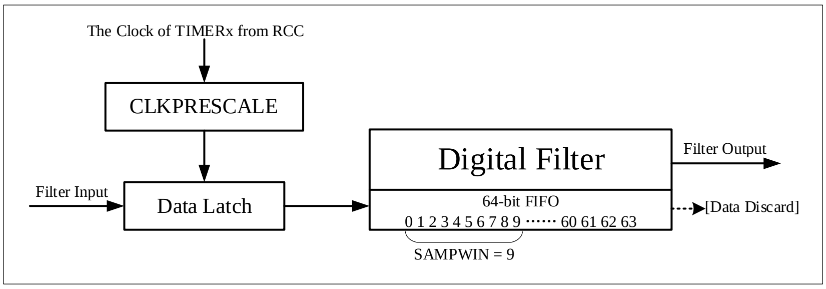
- By configuring TIMx_CCEN.CC1P=0, select the rising edge as the valid transition polarity on the TI1 channel.
- Configure the input prescaler. In this example, configure TIMx_CCMOD1.IC1PSC='00' to disable the prescaler because we want to capture every valid transition.
- Enable capture by configuring TIMx_CCEN.CC1EN = '1'.

If you want to enable DMA request, you can configure TIMx_DINTEN.CC1DEN=1.If you want enable related interrupt request, you can configureTIMx_DINTEN.CC1IEN bit=1

11.5.5.1 Channel Input Filtering

The register TIMx_CxFILT(x=1,2,3,4) is described as follows:

Figure 11-16 Sliding Filtering



- Digital filters sample the channel input signal using the RCC's TIMx clock and accumulate the samples in a 64-bit FIFO. Only data sampled within the window size defined in TIMx_CxFILT.WSIZE [5:0] is considered, with a maximum window size of 64.
- The filter outputs the majority value within the sampling window, which is defined by the threshold in TIMx_CxFILT.THRESH [5:0], with a maximum threshold of 63. This value should be equal to or greater than half the window size. If the counts of logic 1 and logic 0 within the sampling window are not greater than the threshold, the digital filter maintains the previous output value.
- TIMx_SLIDFPSC.PSC register determines the sampling rate of the corresponding digital filter. The filter FIFO captures a sample value from the input at each sampling clock.
- If the digital filter is disabled, the filter input is passed directly to the output.

11.5.6 PWM Input Mode

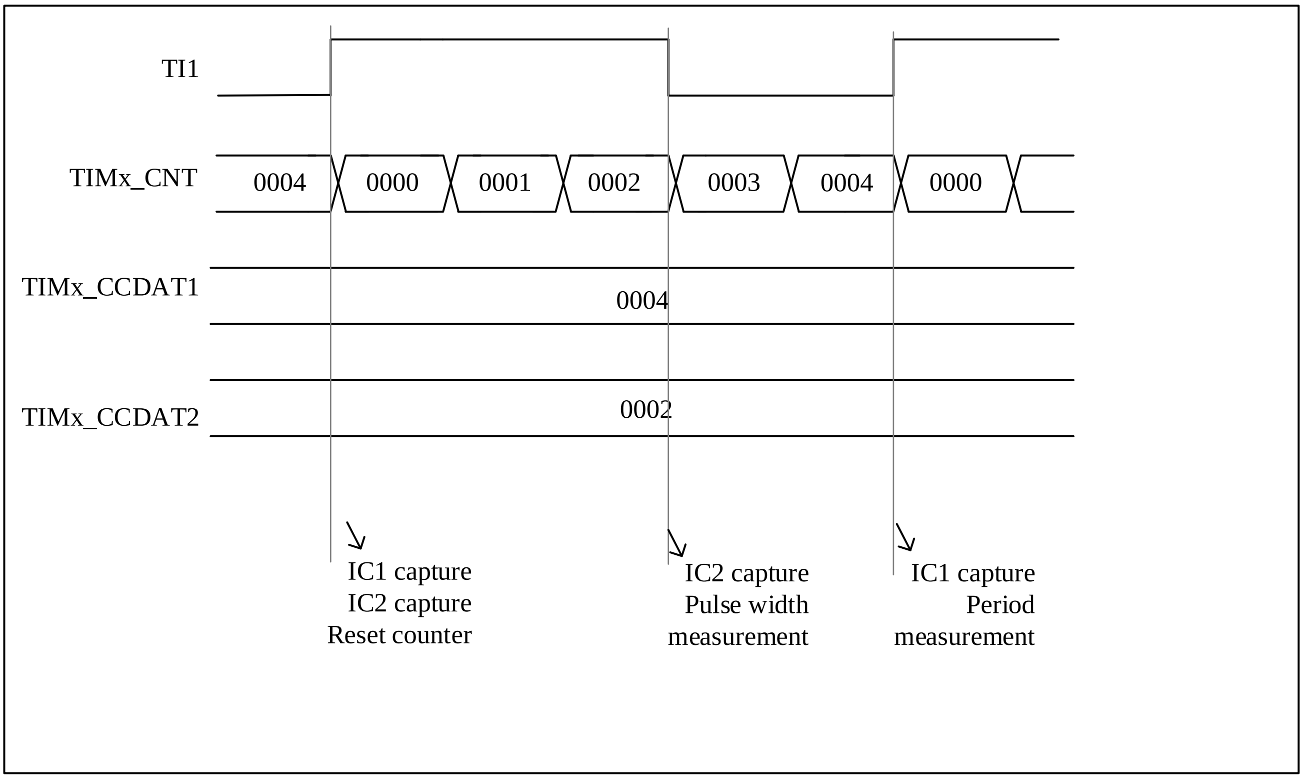
There are some differences between PWM input mode and normal input capture mode, including:

- Two ICx signals are mapped to the same TIx input.
- The two ICx signals are active on edges of opposite polarity.
- Select one of two TIxFP signals as trigger input.
- The slave mode controller is configured in reset mode.

For example, the following configuration flow can be used to know the period and duty cycle of the PWM signal on TI1 (It depends on the frequency of CK_INT and the value of the prescaler).

- Configure TIMx_CCMOD1.CC1SEL equal to '01' to select TI1 as valid input for TIMx_CCDAT1
- Configure TIMx_CCEN.CC1P equal to '0' to select the active polarity of filtered timer input 1(TI1FP1), active at the rising edge.
- Configure TIMx_CCMOD1.CC2SEL equal to '10' select TI1 as valid input for TIMx_CCDAT2.
- Configure TIMx_CCEN.CC2P equal to 1 to select the valid polarity of filtered timer input 2(TI1FP2), active at the falling edge.
- Configure TIMx_SMCTRL.TSEL=101 to select Filtered timer input 1 (TI1FP1) as valid trigger input.
- Configure TIMx_SMCTRL.SMSEL=100 to configure the slave mode controller to reset mode.
- Configure TIMx_CCEN. CC1EN=1 and TIMx_CCEN.CC2EN=1 to enable capture.

Figure 11-17 PWM Input Mode Timing



Because of only filter timer input 1 (TI1FP1) and filter timer input 2 (TI2FP2) are connected to the slave mode controller, the PWM input mode can only be used with the TIMx_CH1/TIMx_CH2 signals.

11.5.7 Forced Output Mode

Software can force output compare signals to active or inactive level directly, in output mode (TIMx_CCMODx.CCxSEL=00).

User can set TIMx_CCMODx. OCxMD=101 to force the output compare signal to active level. And the OCxREF will be forced high, OCx get opposite value to CCxP polarity bit. On the other hand, user can set TIMx_CCMODx.

OCxMD=100 to force the output compare signal to inactive level, the OCxREF will be forced low.

The values of the TIMx_CCDATx shadow register and the counter still comparing with each other in this mode.

The comparison between the output compare register TIMx_CCDATx and the counter TIMx_CNT has no effect on OCxREF. And the flag still can be set. Therefore, the interrupt and DMA requests still can be sent.

11.5.8 Output Compare Mode

User can use this mode to control the output waveform, or to indicate that a period of time has elapsed。

When the capture/compare register and the counter have the same value, the output compare function's operations are as follow：

- TIMx_CCMODx.OCxMD is for output compare mode, and TIMx_CCEN.CCxP is for output polarity. When the compare matches, if set TIMx_CCMODx.OCxMD=000, the output pin will keep its level;if set TIMx_CCMODx.OCxMD=001, the output pin will be set active;if set TIMx_CCMODx.OCxMD=010, the output pin will be set inactive;if set TIMx_CCMODx.OCxMD=011, the output pin will be set to toggle.
- Set TIMx_STS.CCxITF.
- If user set TIMx_DINTEN.CCxIEN, a corresponding interrupt will be generated
- If user set TIMx_DINTEN.CCxDEN and set TIMx_CTRL2.CCDSEL to select DMA request, and DMA request will be sent

User can set TIMx_CCMODx.OCxPEN to choose capture/compare shawdow regisete using capture/compare preload registers(TIMx_CCDATx) or not

The time resolution is one counting period of the counter.

In one pulse mode, the output compare mode can also be used to output a single pulse

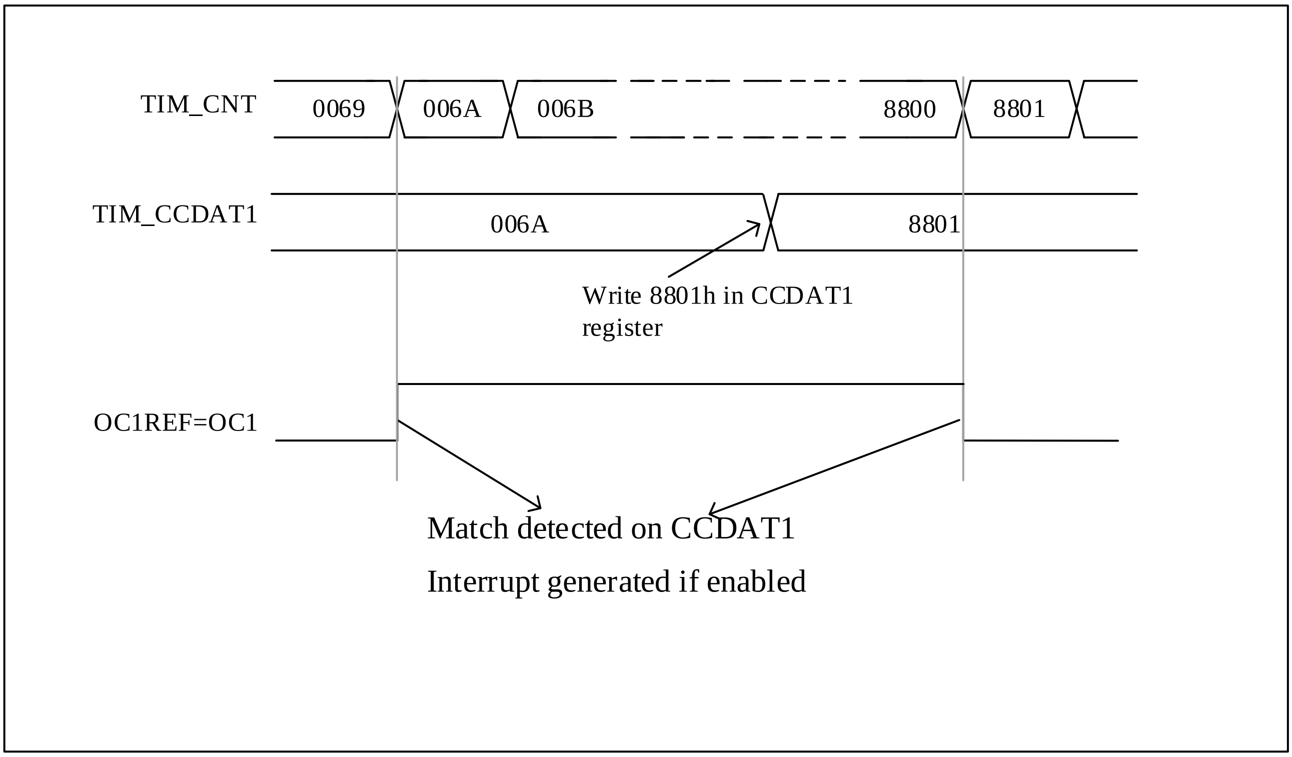
Here are the configuration steps for output compare mode:

- First of all, user should select the counter clock.
- Secondly, set TIMx_AR and TIMx_CCDATx with required data.
- If user need to generate an interrupt, set TIMx_DINTEN.CCxIEN.
- Then select the output mode by set TIMx_CCEN.CCxP, TIMx_CCMODx.OCxMD, TIMx_CCEN.CCxEN, etc.
- At last, set TIMx_CTRL1.CNTEN to enable the counter.

User can update the output waveform by setting TIMx_CCDATx at any time, as long as the preload register is not enabled. Otherwise the TIMx_CCDATx shadow register will be updated at the next update event

Here is an example.

Figure 11-18 Output Compare Mode, Toggle On OC1



11.5.9 PWM Mode

User can use PWM mode to generate a signal whose duty cycle is determined by the value of the TIMx_CCDATx register and whose frequency is determined by the value of the TIMx_AR register. And depends on the value of TIMx_CTRL1.CAMSEL, the TIM can generate PWM signal in edge-aligned mode or center-aligned mode.

User can set PWM mode 1 or PWM mode 2 by setting TIMx_CCMODx. OCxMD=110 or setting TIMx_CCMODx. OCxMD=111. To enable preload register, user must set corresponding TIMx_CCMODx.OCxPEN. And then set TIMx_CTRL1.ARPEN to auto-reload preload register eventually.

User can set polarity of OCx by setting TIMx_CCEN.CCxP.

The values of TIMx_CNT and TIMx_CCDATx are always compared with each other when the TIM is under PWM mode.

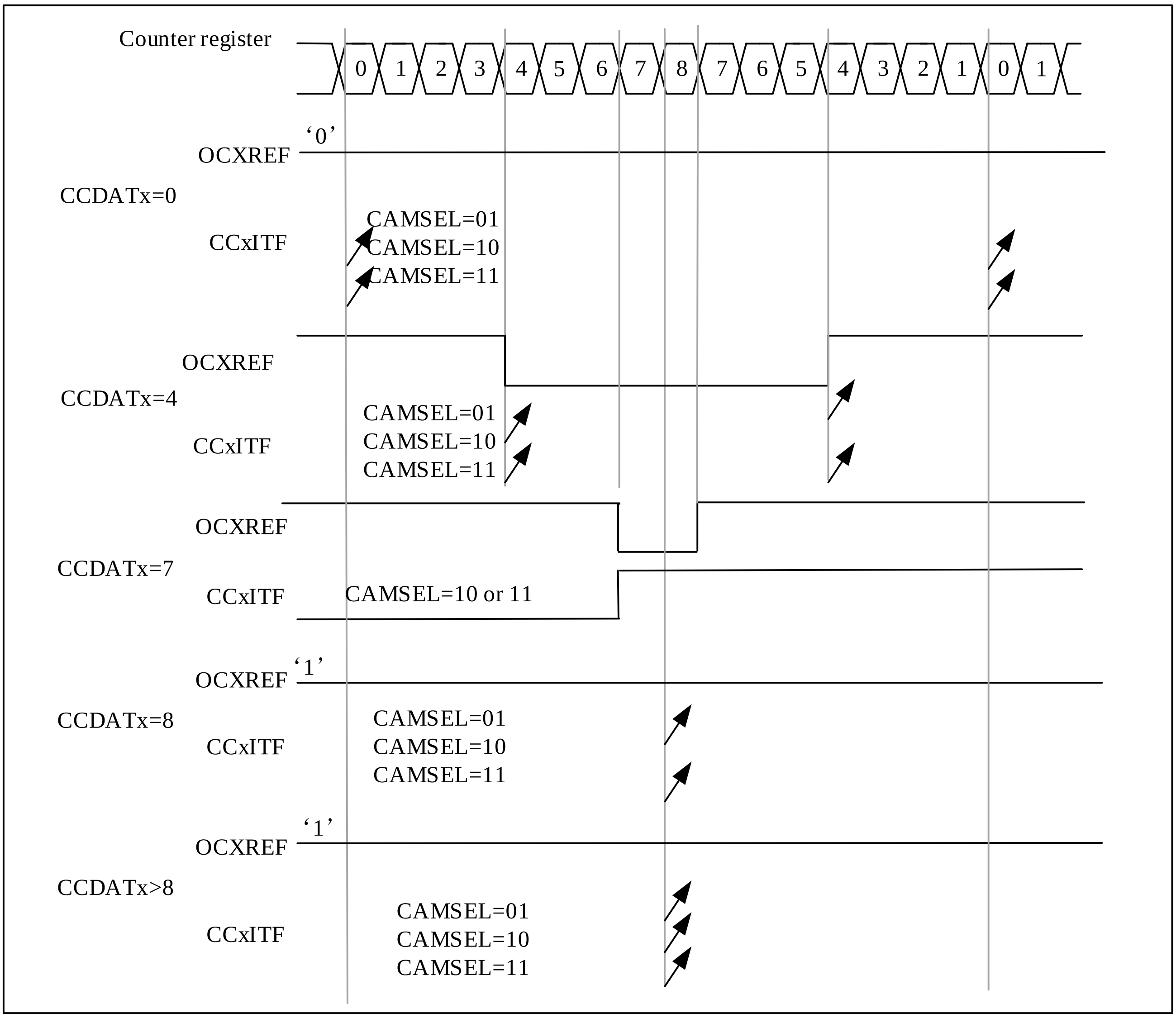
Only if an update event occurs, the preload register will transfer to the shadow register. Therefore user must reset all the registers by setting TIMx_EVTGEN.UDGN before the counter starts counting.

11.5.9.1 PWM center-aligned mode

If user set TIMx_CTRL1.CAMSEL equal 01, 10 or 11, the PWM center-aligned mode will be active. The setting of the compare flag depends on the value of TIMx_CTRL1.CAMSEL. There are three kinds of situation that the compare flag is set, only when the counter counts up, only when the counter counts down, or when the counter counts up and counts down. User should not modified TIMx_CTRL1.DIR by software, it is updated by hardware.

Examples of center-aligned PWM waveforms is as follow, and the setting of the waveform are: TIMx_AR=8, PWM mode 1, the compare flag is set when the counter counts down corresponding to TIMx_CTRL1.CAMSEL=01.

Figure 11-19 Center-Aligned PWM Waveform (AR=8)



When using center-aligned mode, users should pay attention to the following considerations:

It depends on the value of TIMx_CTRL1.DIR that the counter counts up or down. Cautions that the DIR and CAMSEL bits should not be changed at the same time.

- User should not write the counter while running in center-aligned mode, otherwise it will cause unexpected results. Here are some examples:
  - If the value written into the counter is 0 or is the value of TIMx_AR, the direction will be updated but the update event will not be generated.
  - If the value written into the counter is greater than the value of auto-reload, the direction will not be updated
- To be on the safe side, user is suggested setting TIMx_EVTGEN.UDGN to generate an update by software before starting the counter, and not writing the counter while it is running.

11.5.9.2 PWM edge-aligned mode

There are two kinds of configuration in edge-aligned mode, up-counting and down-counting.

- Up-counting

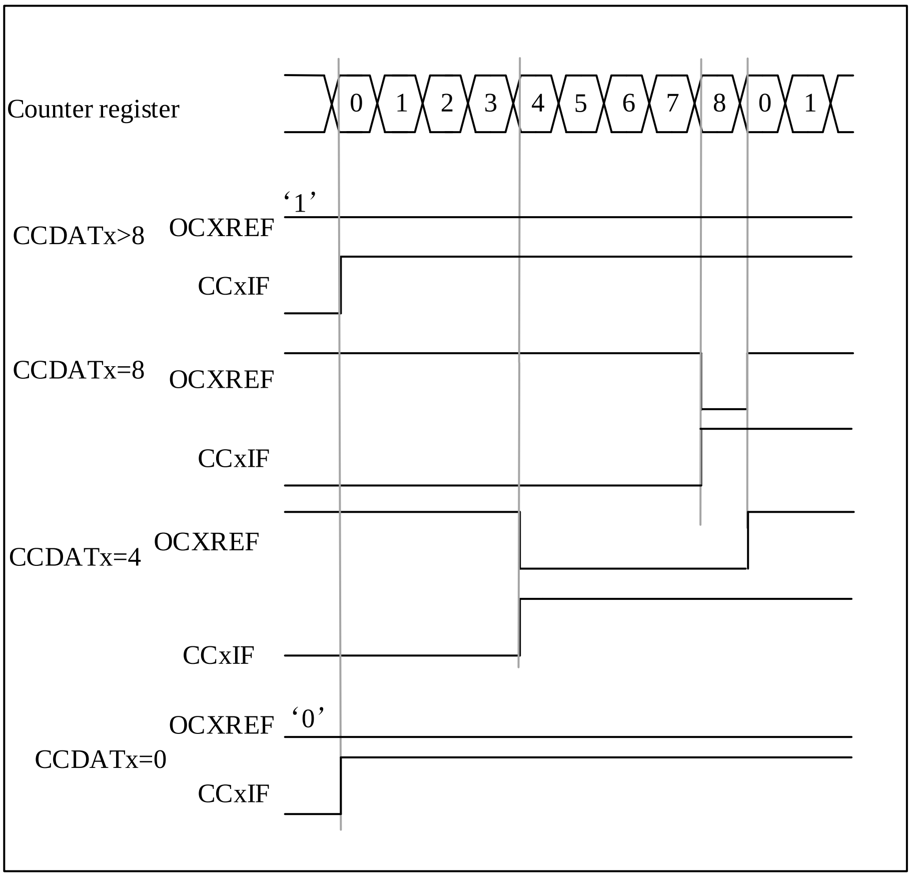
User can set TIMx_CTRL1.DIR=0 to make counter counts up.

Example for PWM mode1：

When TIMx_CNT < TIMx_CCDATx, the reference PWM signal OCxREF is high. Otherwise it will be low. If the compare value in TIMx_CCDATx is greater than the auto-reload value, the OCxREF will remains 1. Conversely, if the compare value is 0, the OCxREF will remains 0.

When TIMx_AR=8, the PWM waveforms are as follow:

Figure 11-20 Edge-Aligned PWM waveform (AR=8)



- Down-counting

User can set TIMx_CTRL1.DIR=1 to make counter counts down.
Example for PWM mode1：

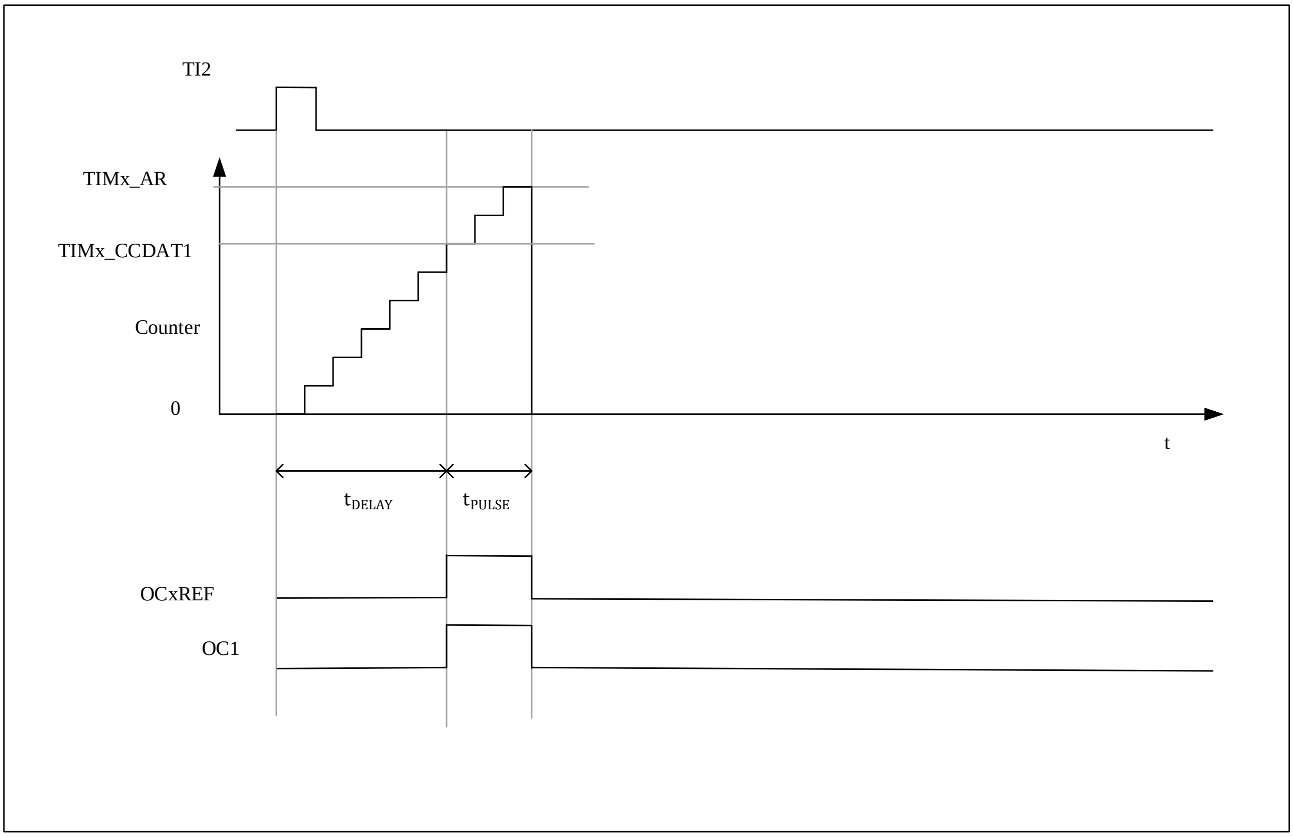
When TIMx_CNT > TIMx_CCDATx, the reference PWM signal OCxREF is low. Otherwise it will be high. If the compare value in TIMx_CCDATx is greater than the auto-reload value, the OCxREF will remains 1.

Note: If the nth PWM cycle CCDATx shadow register >= AR value, the shadow register value of CCDATx in the (n+1)th PWM cycle is 0. At the moment when the counter is 0 in the (n+1)th PWM cycle, although the value of the counter = CCDATx shadow register = 0 and OCxREF = '0', no compare event will be generated.

11.5.10 One-Pulse Mode

In the one-pulse mode (ONEPM), a trigger signal is received, and a pulse t_{PULSE} with a controllable pulse width is generated after a controllable delay t_{DELAY}. The output mode needs to be configured as output compare mode or PWM mode. After selecting one-pulse mode, the counter will stop counting after the update event UEV is generated

Figure 11-21 Example Of One-pulse Mode



The following is an example of a one-pulse mode：

A rising edge trigger is detected from the TI2 input, and a pulse with a width of t_{PULSE} is generated on OC1 after a delay of t_{DELAY}.

1. Counter configuration: count up, counter TIMx_CNT < TIMx_CCDAT1 ≤ TIMx_AR;
2. TI2FP2 is mapped to TI2, TIMx_CCMOD1.CC2SEL=‘01’; TI2FP2 is configured for rising edge detection, TIMx_CCEN.CC2P=‘0’;
3. TI2FP2 acts as the trigger (TRGI) of the slave mode controller and starts the counter, TIMx_SMCTRL.TSEL= ‘110’, TIMx_SMCTRL.SMSEL=‘110’ (trigger mode);
4. TIMx_CCDAT1 writes the count value to be delayed (t_{DELAY}), TIMx_AR - TIMx_CCDAT1 is the count value of the pulse width t_{PULSE};
5. Configure TIMx_CTRL1.ONEPM=1 to enable single pulse mode, configure TIMx_CCMOD1.OC1MD=‘111’ to select PWM2 mode;

6. Wait for an external trigger event on TI2, and a one pulse waveform will be output on OC1;

11.5.10.1 Special case: Ocx fast enable:

In one-pulse mode, an edge is detected through the TIx input, and triggers the start of the counter to count to the comparison value and then output a pulse. These operations limit the minimum delay t_{DELAY} that can be achieved.

You can set TIMx_CCMODx.OCxFEN=1 to turn on OCx fast enable, after triggering the rising edge, the OCxREF signal will be forced to be converted to the same level as the comparison match occurs immediately, regardless of the comparison result. OCxFEN fast enable only takes effect when the channel mode is configured for PWM1 and PWM2 modes.

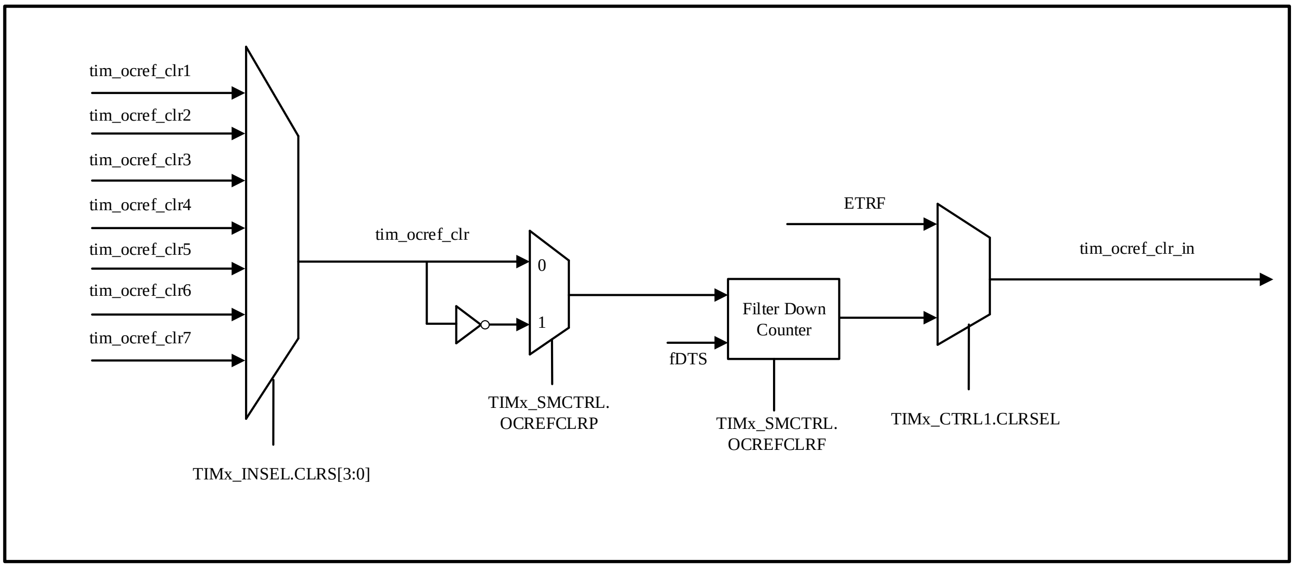
11.5.11 Clearing The OCxref Signal On An External Event

If user set TIMx_CCMODx.OCxCEN=1, high level of tim_ocref_clr_in input can be used to driven the OCxREF signal to low, and the OCxREF signal will remains low, until the next UEV happens. Only output compare and PWM modes can use this function. This cannot be used when it is in forced mode.

The input clear signal tim_ocref_clr_in can be selected as tim_ocref_clr or ETRF through the CLRSEL bit in the TIMx_CTRL1 register.

The tim_ocref_clr signal can be selected through the CLRS[3:0] in the TIMx_INSEL register, as shown in the following figure.

Figure 11-22 Clearing OCxREF Signal By External Event

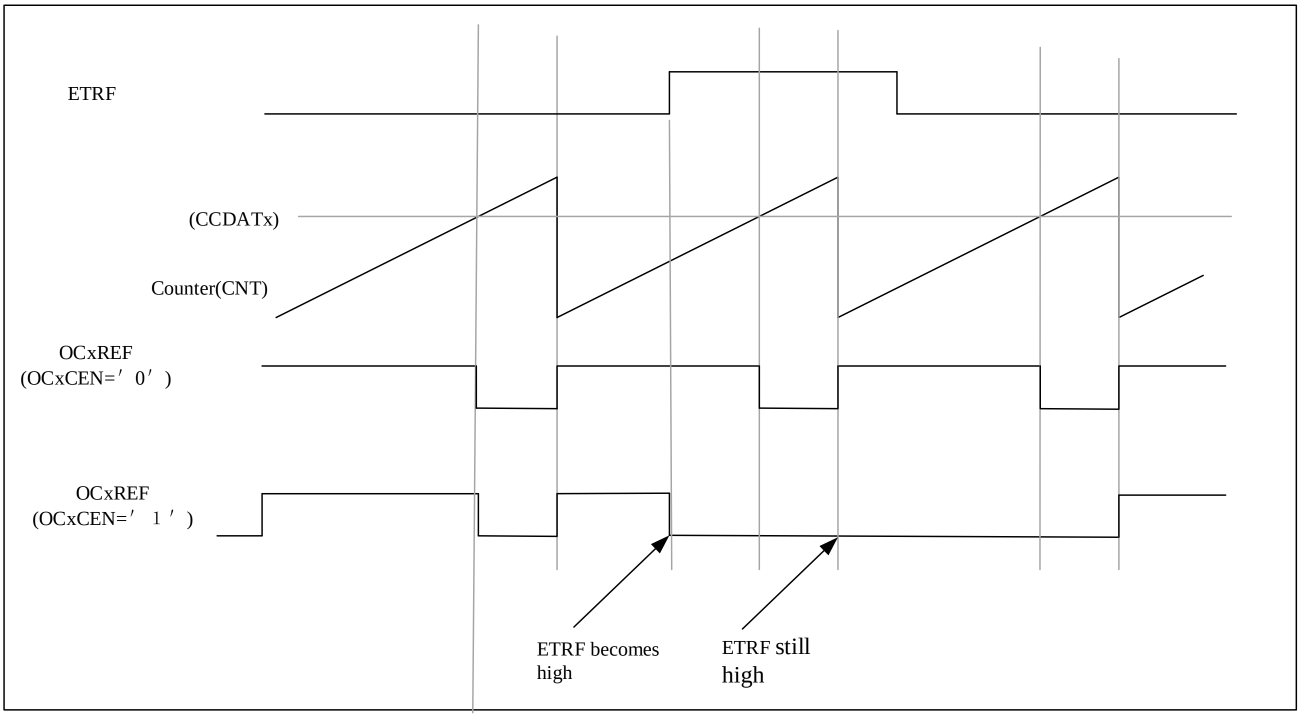


For example, when the tim_ocref_clr_in signal selects ETRF, the tim_etr_in configuration is as follows:

- Set TIMx_SMCTRL.EXTPS=00 to disable the external trigger prescaler.
- Set TIMx_SMCTRL.EXCEN=0 to disable the external clock mode 2.
- Set TIMx_SMCTRL.EXTP and TIMx_SMCTRL.EXTF to configure the external trigger polarity and external trigger filter as needed.
- Here is an example for the case that when ETRF input becomes high, the behavior of OCxREF signal for

different value of OCxCEN. Timer is set to be in PWM mode in this case.

Figure 11-23 Clearing OCxREF of TIMx



11.5.12 Debug Mode

When the microcontroller is in debug mode (the Cortex^{TM}-M4F core halted), depending on the DBG_CTRL.GTIMx_STOP configuration, the TIMx counter can either continue to work normally or stop. For more details,see Section37.3.2.

11.5.13 GTIMx and External Trigger Synchronization

The timer can be synchronized through triggers in the slave modes (reset, trigger, and gated).

11.5.13.1 Slave Mode: Reset Mode

In reset mode, the trigger event can reset the counter and prescaler. Update the preload register TIMx_AR, TIMx_CCDATx, and generate an update event UEV (TIMx_CTRL1.UPRS=0).

Here is an example of the reset mode:

1. Channel 1 is configured to detect the rising edge of input TI1 (TIMx_CCMOD1.CC1SEL=01, TIMx_CCEN.CC1P=0).

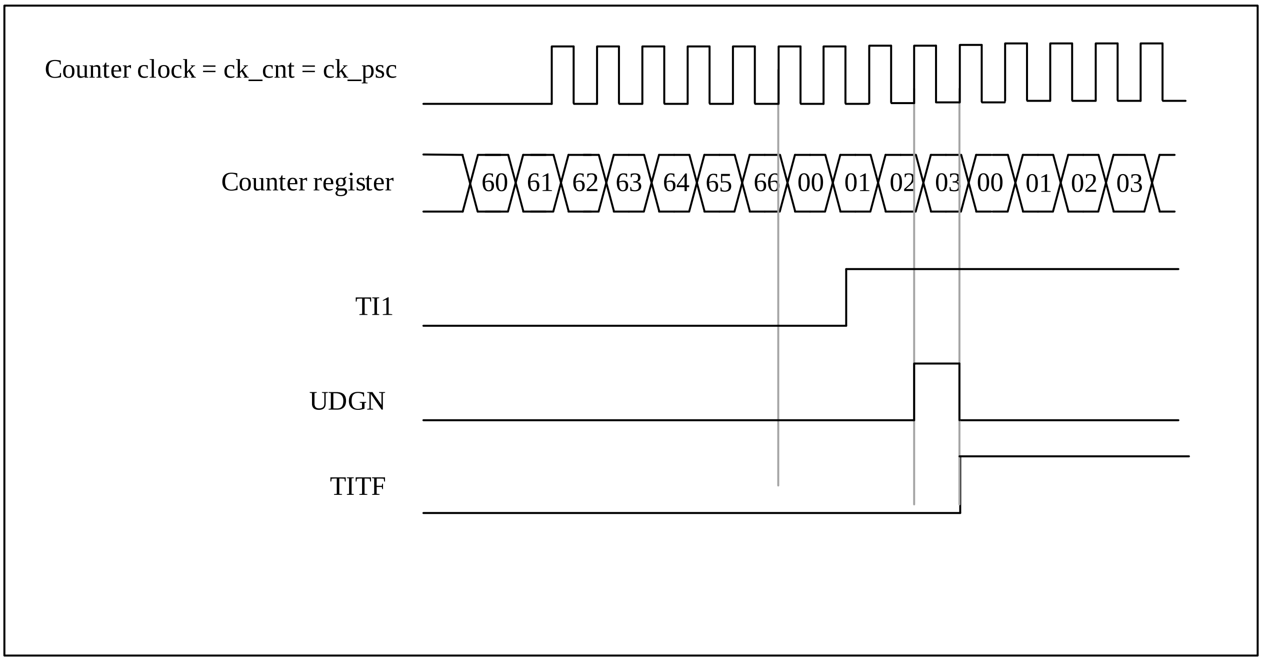
2. Select the mode as reset mode (TIMx_SMCTRL.SMSEL=0100), and trigger input selection as TI1 (TIMx_SMCTRL.TSEL=101).

3. Start the counter (TIMx_CTRL1.CNTEN = 1).

After starting the timer, when TI1 detects a rising edge, the counter resets and restarts counting, and sets the trigger flag (TIMx_STS.TITF=1).

The delay between the rising edge of TI1 and the actual counter reset is caused by the resynchronization circuit of the TI1 input.

Figure 11-24 Control Circuit In Reset Mode



11.5.13.2 Slave Mode : Trigger Mode

In trigger mode, the trigger event (rising edge/falling edge) at the input port can start the counter counting

Here is an example of the trigger mode:

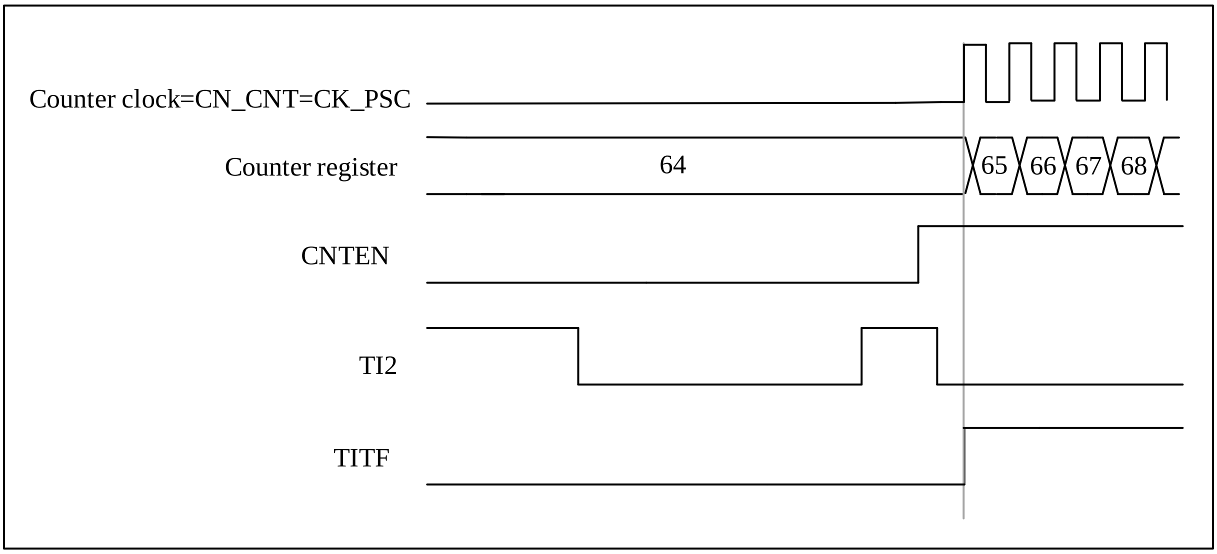
1. Channel 2 is configured as an input to detect the rising edge of TI2 (TIMx_CCMOD1.CC2SEL=01，TIMx_CCEN.CC2P=0)

2. Select the mode as trigger mode (TIMx_SMCTRL.SMSEL=0110), and trigger input selection as TI2 (TIMx_SMCTRL.TSEL=110).

When TI2 detects a rising edge, the counter starts counting, and the trigger flag is set (TIMx_STS.TITF=1);

The delay between the rising edge of TI2 and the actual start of the counter is caused by the resynchronization circuit of the TI2 input.

Figure 11-25 Control Circuit In Trigger Mode



11.5.13.3 Slave Mode: Gated Mode

In gated mode, the level polarity of the input port can control whether the counter counts.

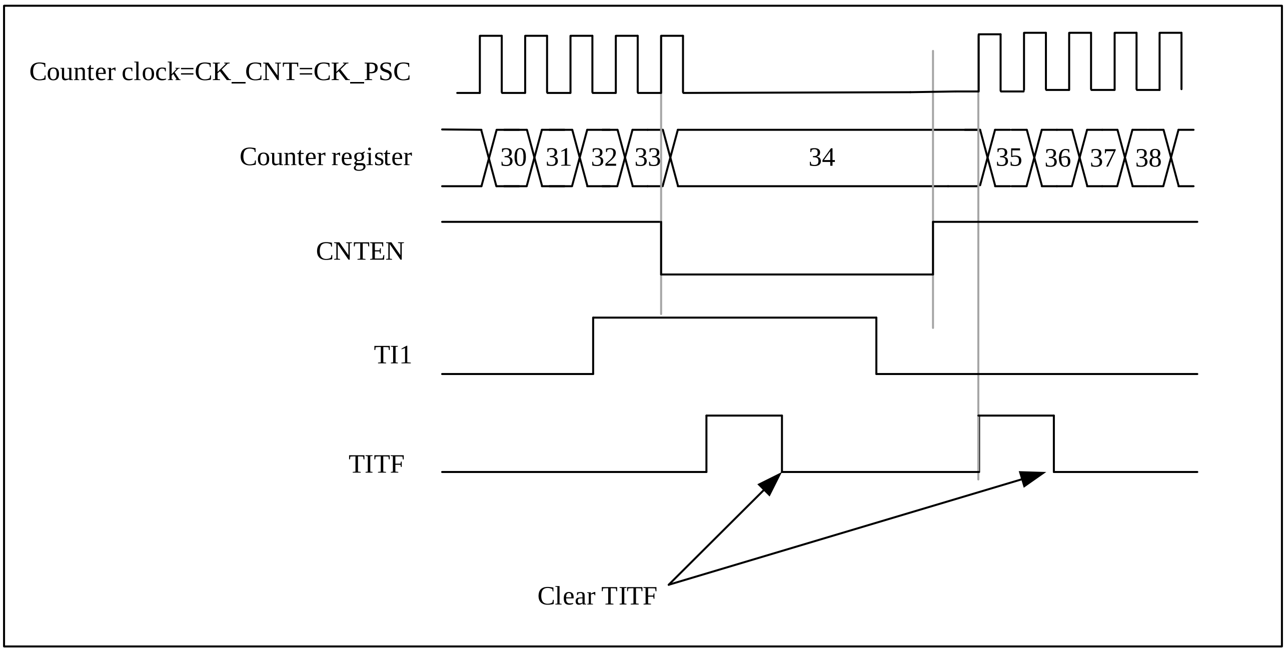
Here is an example of the gated mode:

1. Channel 1 is configured to detect low-level effective input on TI1 (TIMx_CCMOD1.CC1SEL=01, TIMx_CCEN.CC1P=1);
2. Select the mode as gated mode (TIMx_SMCTRL.SMSEL=0101), and choose TI1 as TRGI (TIMx_SMCTRL.TSEL=101);
3. Start the counter（TIMx_CTRL1.CNTEN = 1）;

When TI1 detects a transition from low to high level, the counter stops counting. When TI1 detects a transition from high to low level, the counter starts counting. The trigger flag is set when counting starts or stops (TIMx_STS.TITF=1).

The delay between the rising edge of TI1 and the actual stop of the counter is caused by the resynchronization circuit of the TI1 input.

Figure 11-26 Control Circuit In Gated Mode



11.5.13.4 SlaveMode: Trigger Mode + External Clock Mode 2

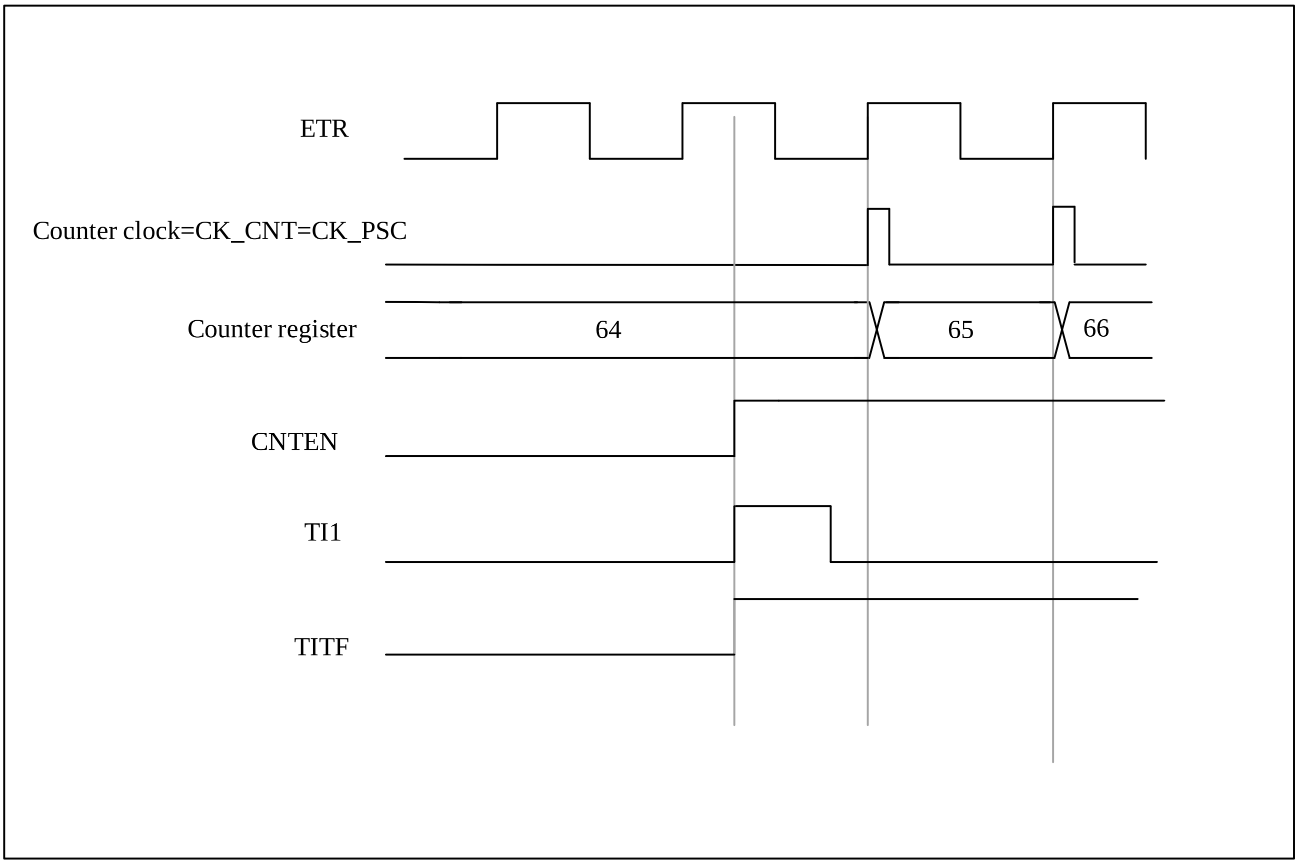
In reset mode, trigger mode, and gated mode, the counter clock can be selected as External Clock Mode 2, with the ETR signal as the external clock source input. In this case, the trigger selection needs to choose non-ETRF (TIMx_SMCTRL.TSEL=111).

For example:

1. Channel 1 is configured as an input to detect the rising edge of TI1 (TIMx_CCMOD1.CC1SEL=01, TIMx_CCEN.CC1P=0);
2. Enable External Clock Mode 2 (TIMx_SMCTRL.EXCEN=1), select rising edge for external trigger polarity (TIMx_SMCTRL.EXTP=0), choose trigger mode as slave mode (TIMx_SMCTRL.SMSEL=0110), and select TRGI as TI1 (TIMx_SMCTRL.TSEL=101);

When TI1 detects a rising edge, the counter starts counting on the rising edge of ETR, and sets the trigger flag (TIMx_STS.TITF=1).

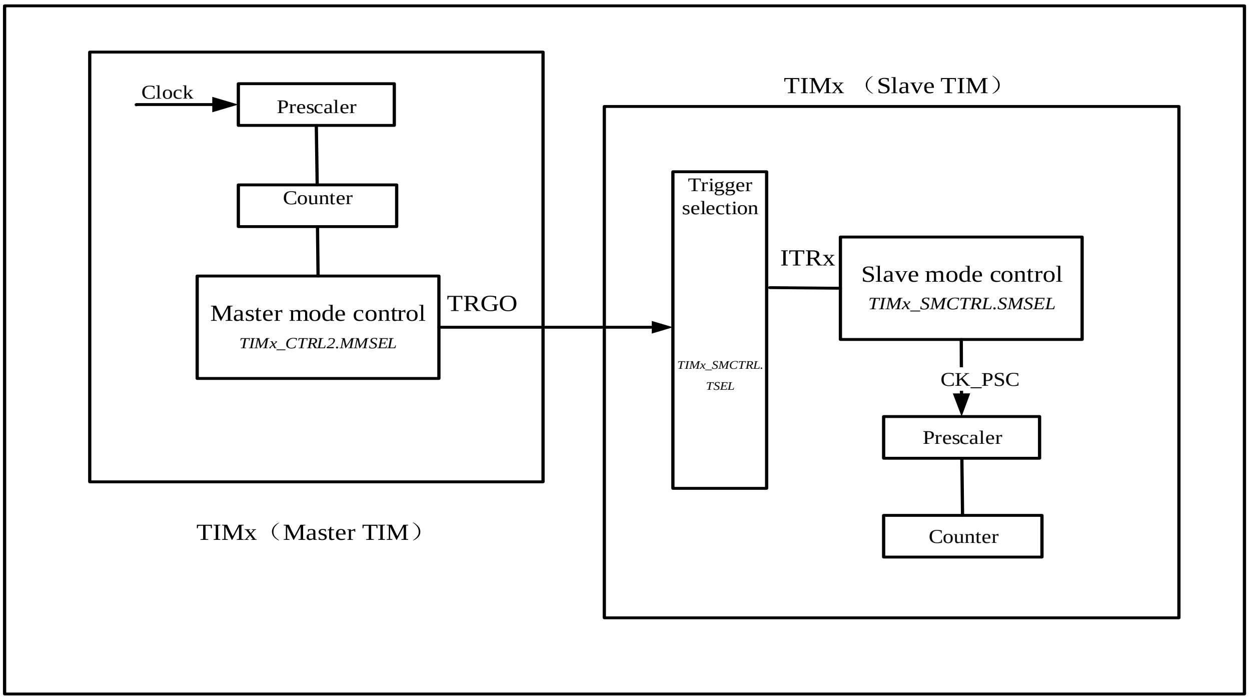
Figure 11-27 Control circuit in External Clock Mode 2 + Trigger Mode.



11.5.14 Timer Synchronization

All TIMx timers are internally interconnected to each other. This implementation allows a master timer to provide trigger to reset, start, stop or provide a clock for the other slave timers. The master clock is used for internal counter and can be prescaled. Below figure shows a block diagram of timer interconnection. The synchronization function does not support dynamic change of the connection. User should configure and enable the slave timer before enabling the master timer’s trigger or clock.

Figure 11-28 Master/Slave Timer Connection



11.5.14.1 Master timer as a prescaler for another timer

ATIM1 acts as the prescaler for GTIM1. ATIM1 is master timer, and GTIM 1 is slave timer.

User needs to do the following steps for this configuration.

- Set ATIM1_CTRL2.MMSEL=’0010’ to use the update event of ATIM1 as trigger output.
- Configure GTIM2_SMCTRL.TSEL= ‘000’、GTIM2_INSEK.ITRS=’000’ to connect the TRGO of ATIM1 to GTIM2.
- Configure GTIM2_SMCTRL.SMSEL = ‘0111’ so that the slave mode controller will be configured in external clock mode 1.
- Start GTIM2 by setting GTIM2_CTRL1. CNTEN = ‘1’.
- Start ATIM1 by setting ATIM1_CTRL1. CNTEN = ‘1’.

Note: If user select OCx as the trigger output of ATIM1 by configuring MMSEL = ‘01xx’, OCx rising edge will be used to drive TIM2.

11.5.14.2 Master timer to enable another timer

In this example, GTIM2 is enabled by the output compare of ATIM1. GTIM2 counter will start to count after the OC1REF output from ATIM1 is high. Both counters are clocked based on CK_INT via a prescaler divide by 3 is performed ($f_{CK\_CNT} = f_{CK\_INT}/3$).

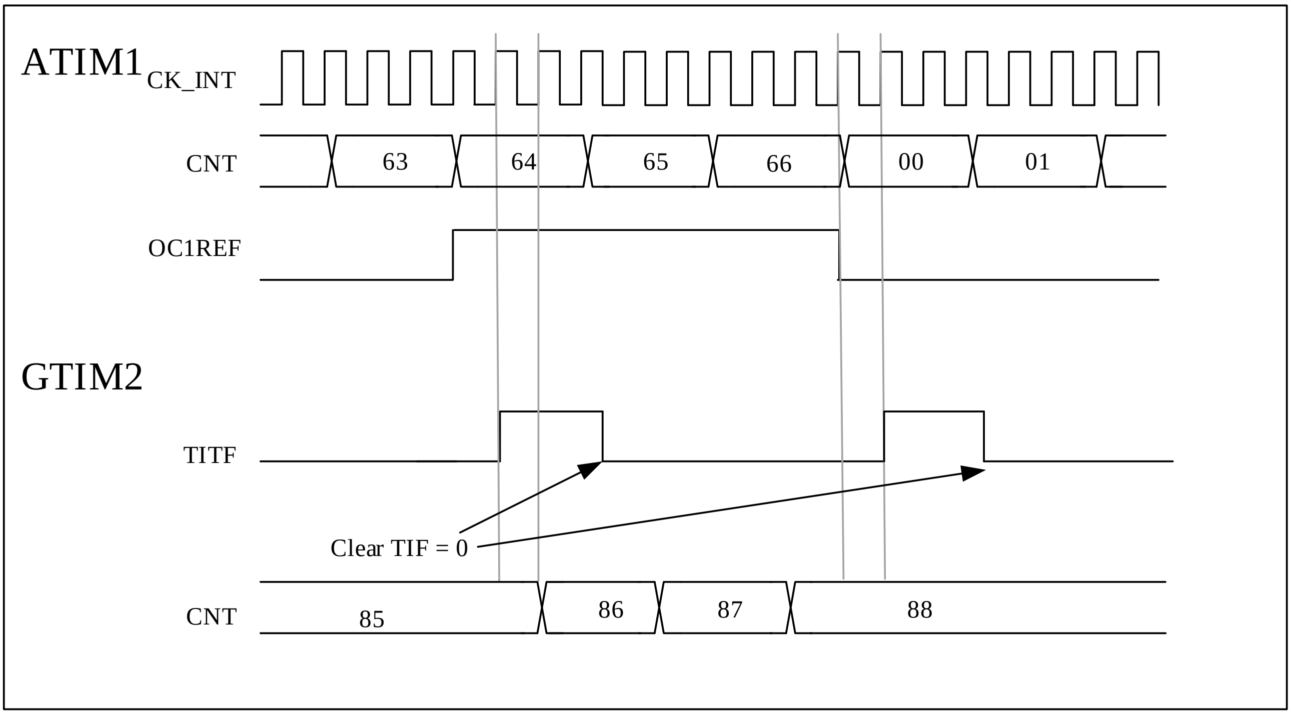
The configuration steps are shown as below。

- Set ATIM1_CTRL2.MMSEL=’0100’ to use the OC1REF of ATIM1 as trigger output.
- Configure ATIM1_CCMOD1 register to configure the OC1REF output waveform.

- Set GTIM2_SMCTRL.TSEL = ‘000’、GTIM2_INSEL.ITRS=‘000’ to connect ATIM1 trigger output to GTIM2.
- Set GTIM2_SMCTRL.SMSEL= ‘0101’ to set GTIM2 to gated mode.
- Set GTIM2_CTRL1.CNTEN= ‘1’ to start GTIM2.
- Set ATIM1_CTRL1.CNTEN= ‘1’ to start ATIM1.

Note: The GTIM2 clock is not synchronized with the ATIM1 clock, this mode only affects the GTIM2 counter enable signal.

Figure 11-29 GTIM2 Gated By OC1REF Of ATIM1

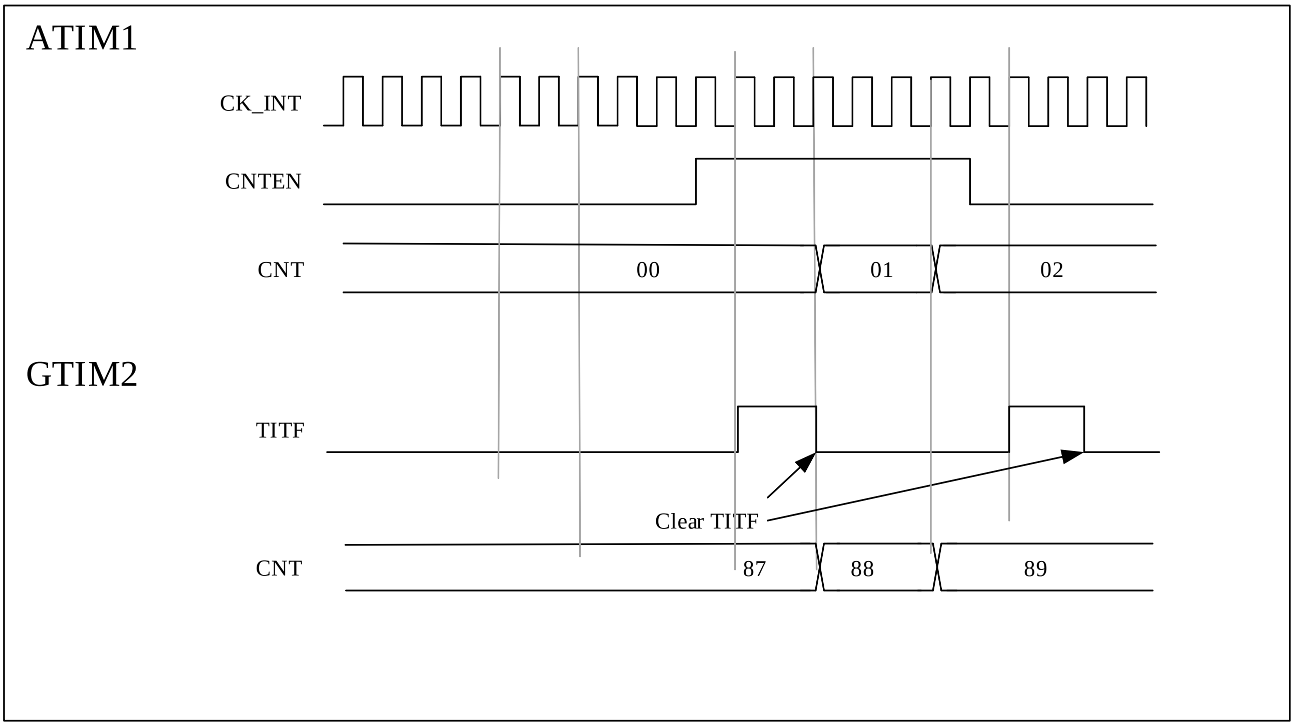


In the next example, it sets the enable of GTIM2 with enable signal of ATIM1. Set ATIM1.CTRL1.CNTEN = ‘0’ to stop ATIM1. GTIM2 counts on the divided internal clock only when ATIM1 is enable. Both counters are clocked based on CK_INT via a prescaler divide by 3 is performed ($f_{CK\_CNT} = f_{CK\_INT}/3$).

The configuration steps are shown as below:

- Set ATIM1_CTRL2.MMSEL=’0001’ to use the enable signal of ATIM1 as trigger output.
- Set GTIM2_SMCTRL.TSEL = ‘000’、 GTIM2__INSEL. ITRS=‘000’ to configure GTIM2 to get the trigger input from ATIM1.
- Set GTIM2_SMCTRL.SMSEL = ‘0101’ to configure GTIM2 in gated mode.
- Set GTIM2_CTRL1.CNTEN= ‘1’ to start GTIM2.
- Set ATIM1_CTRL1.CNTEN= ‘1’ to start ATIM1.
- Set ATIM1_CTRL1.CNTEN= ‘0’ to stop ATIM1.

Figure 11-30 GTIM2 Gated By Enable Signal Of TIM1



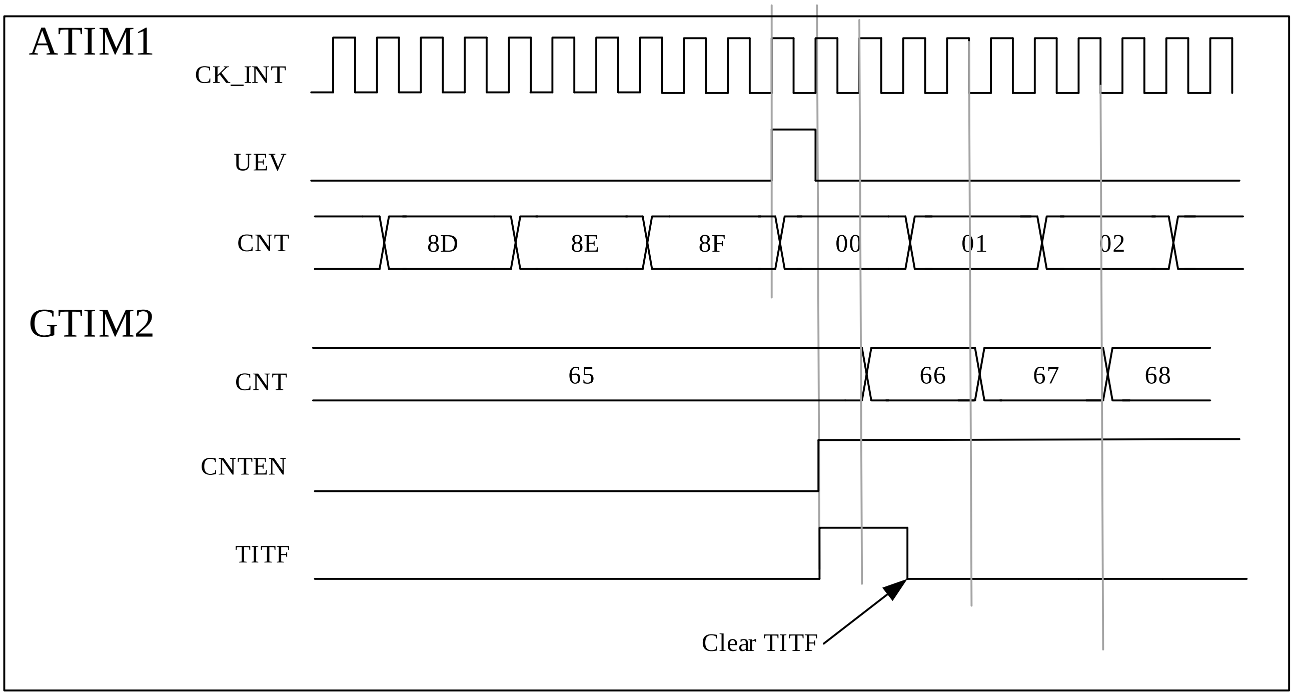
11.5.14.3 Master timer to start another timer

In this example, we can use update event as trigger source.ATIM1 is master, GTIM2 is slave.

The configuration steps are shown as below:

- Set ATIM1_CTRL2.MMSEL=' 0010' to use the update event of ATIM1 as trigger output.
- Configure ATIM1_AR register to set the output period.
- Set GTIM2_SMCTRL .TSEL= '000'、 GTIM2_INSEL. ITRS='000' to connect TIM1 trigger output to GTIM2.
- Set GTIM2_SMCTRL. SMSEL = '110' to set GTIM2 to trigger mode.
- Set ATIM1_CTRL1.CNTEN=1 to start ATIM1.

Figure 11-31 Trigger GTIM2 with An Update Of ATIM1



11.5.14.4 Start 2 timer synchronously using an external trigger

In this example, ATIM1 is enabled when ATIM1's TI1 input rises, and GTIM2 is enabled when ATIM1 is enabled. To ensure the alignment of counters, ATIM1 must be configured in master/slave mode. For TI1, ATIM1 is the slave; for GTIM2, ATIM1 is the master.

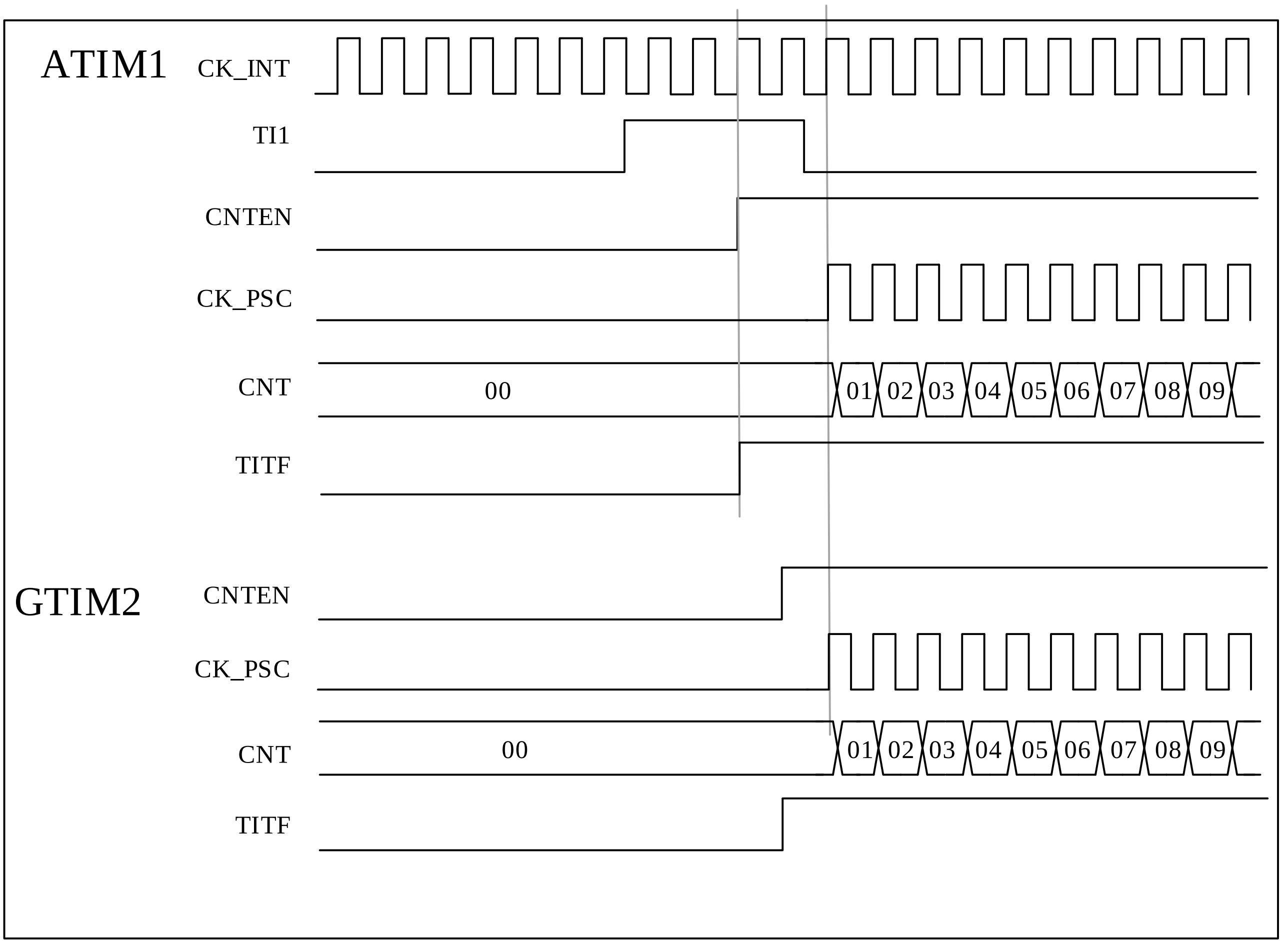
The configuration steps are shown as below:

- Set ATIM1.MMSEL = '0001' to use the enable signal as trigger output.
- Set ATIM1_SMCTRL.TSEL = '100' to configure the ATIM1 to slave mode and receive the trigger input of TI1.
- Set ATIM1_SMCTRL.SMSEL = '0110' to configure ATIM1 to trigger mode.
- Set ATIM1_SMCTRL.MSMD = '1' to configure ATIM1 to master/slave mode.
- Set GTIM2_SMCTRL.TSEL = '000'、GTIM2_INSEL. ITRS='000' to connect ATIM1 trigger output to GTIM2.
- Set GTIM2_SMCTRL.SMSEL = '0110' to configure GTIM2 to trigger mode.

When TI1 rising edge arrives, both timers start counting synchronously according to the internal clock, and both TITF flags are set simultaneously.

Note：The following figure shows a delay between CNTEN and CK_PSC of ATIM1 in master/slave mode.

Figure 11-32 Triggers ATIM1 And GTIM2 Using The TI1 Input Of ATIM1



11.5.15 Trigger ADC

The timer can generate ADC trigger events through various internal signals, such as reset, enable, or comparison events, and can also generate pulse triggers issued by the internal edge detector. The triggering signal issued on the internal line redirected to the ADC can be selected through the MMSEL[3:0] bits in the TIMx_CTRL2 register.

11.5.16 Encode interface mode

11.5.16.1 Quadrature Coding Mode

The encoder uses two inputs TI1 and TI2 as an interface. And the counter counts on every edge change on TI1FP1 or TI2FP2. The counting direction is automatically controlled by hardware TIMx_CTRL1.DIR. There are five types of quadrature encoder counting modes:

- Encode Mode 1: The counter only counts on the edge of TI1, TIMx_SMCTRL.SMSEL = ‘0001’;
- Encode Mode 2: The counter only counts on the edge of TI2, TIMx_SMCTRL.SMSEL =’0010’;
- Encode Mode 3: The counter counts on both TI1 and TI2 edges, TIMx_SMCTRL.SMSEL = ‘0011’;
- Encode Mode 4: T2 in high level, the counter counts on TI1 edges, TIMx_SMCTRL.SMSEL = ‘1001’;

- Encode Mode 5: T1 in high level, the counter counts on TI2 edges, TIMx_SMCTRL.SMSEL = '1010'.

The encoder interface is equivalent to using an external clock with direction selection, and the counter only counts continuously between 0 and the auto-reload value (TIMx_AR.AR [15:0]). Therefore, it is necessary to configure the auto-reload register TIMx_AR in advance.

Note: Encoder mode and external clock mode 2 are not compatible and must not be selected together.

The relationship between the counting direction and the encoder signal is shown in following table:

Table 11-10 The Relationship Between The Counting Direction And The Encoder Signal (CC1P=CC2P=0)

Active Edge	**SMSEL[3:0]**	**Relative Signal Level** **（TI1FP1 for TI2,** **TI2FP2 for TI1）**	**TI1FP1 Signal**		**TI2FP2 Signal**	
			Rising	**Falling**	**Rising**	**Falling**
Counting only at TI1	0001	High	Counting down	Counting up	Don't count	Don't count
		Low	Counting up	Counting down	Don't count	Don't count
Counting only at TI2	0010	High	Don't count	Don't count	Counting up	Counting down
		Low	Don't count	Don't count	Counting down	Counting up
Counting on TI1 and TI2	0011	High	Counting down	Counting up	Counting up	Counting down
		Low	Counting up	Counting down	Counting down	Counting up
Counting only at TI1 and T2 in high level	1001	High	Counting down	Counting up	Don't count	Don't count
		Low	Don't count	Don't count	Don't count	Don't count
Counting only at T2 and T1 in high level	1010	High	Don't count	Don't count	Counting up	Counting down
		Low	Don't	Don't	Don't	Don't

Active Edge	SMSEL[3:0]	Relative Signal Level （TI1FP1 for TI2, TI2FP2 for TI1）	TI1FP1 Signal		TI2FP2 Signal	
			Rising	Falling	Rising	Falling
			count	count	count	count

The changes in the counter value of the counter in each mode are as follows:

Figure 11-33 The Encode Counts Only on TI1

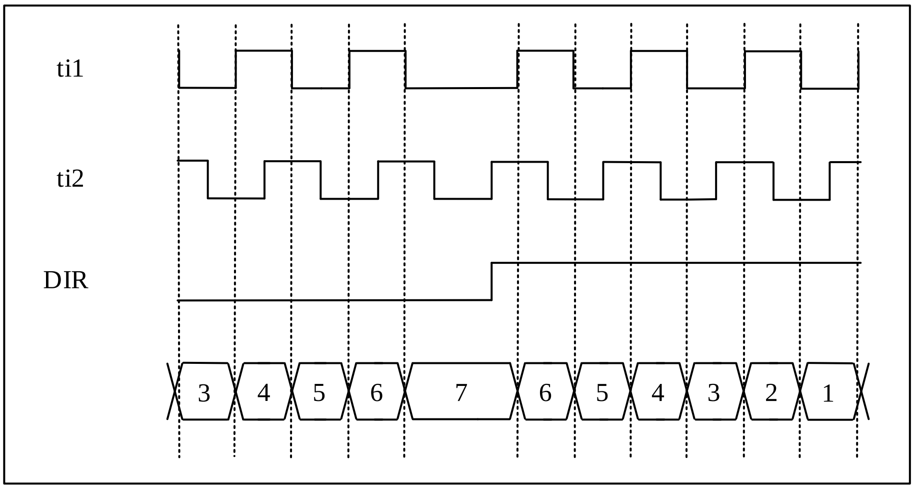


Figure 11-34 The Encode Counts Only on TI2

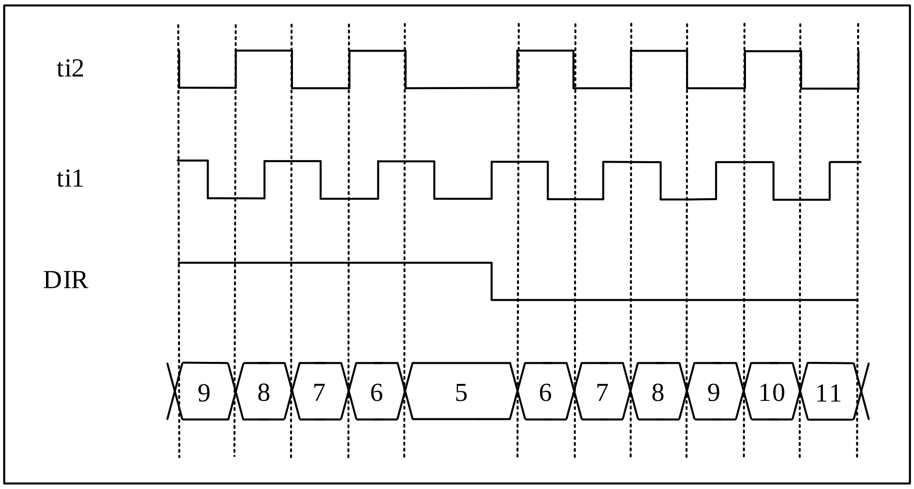


Figure 11-35 The Encoder Counts on Both TI1 and TI2

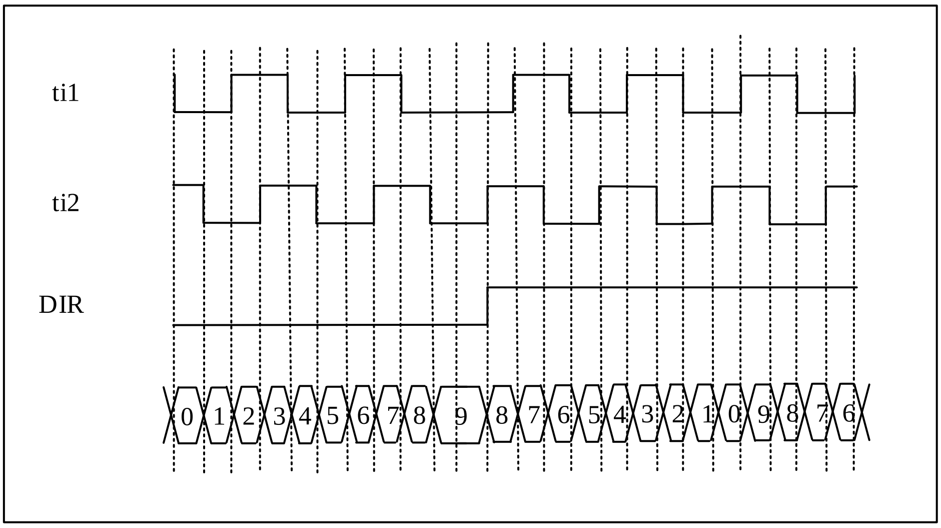


Figure 11-36 When T2 is high level, the counter only counts on TI1

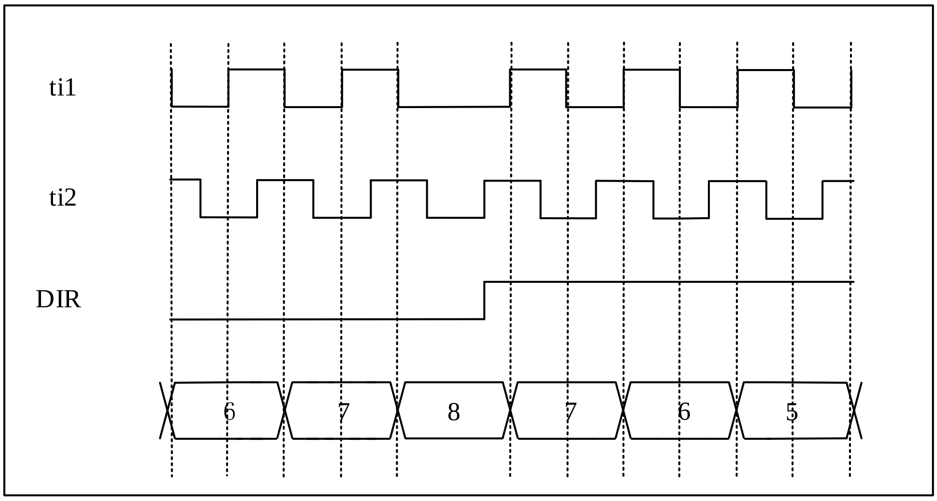
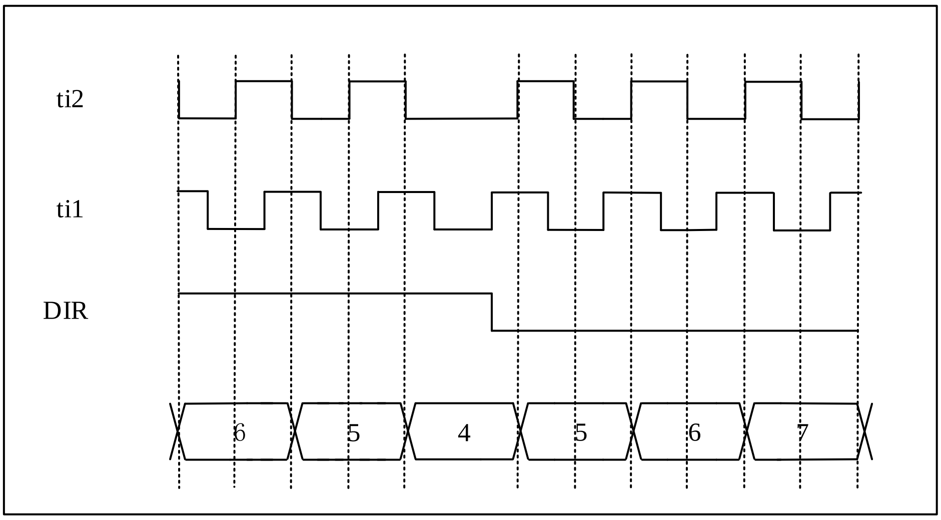


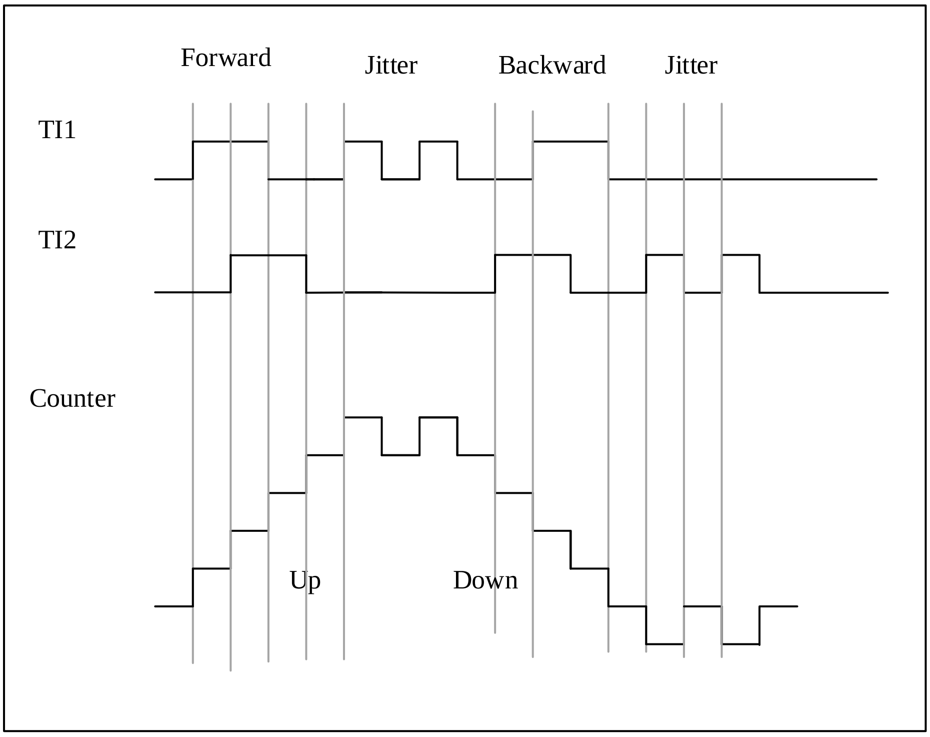
Figure 11-37 When T1 is high level, the counter only counts on TI2



Here is an example of an encoder with dual edges triggering selected to suppress input jitter：

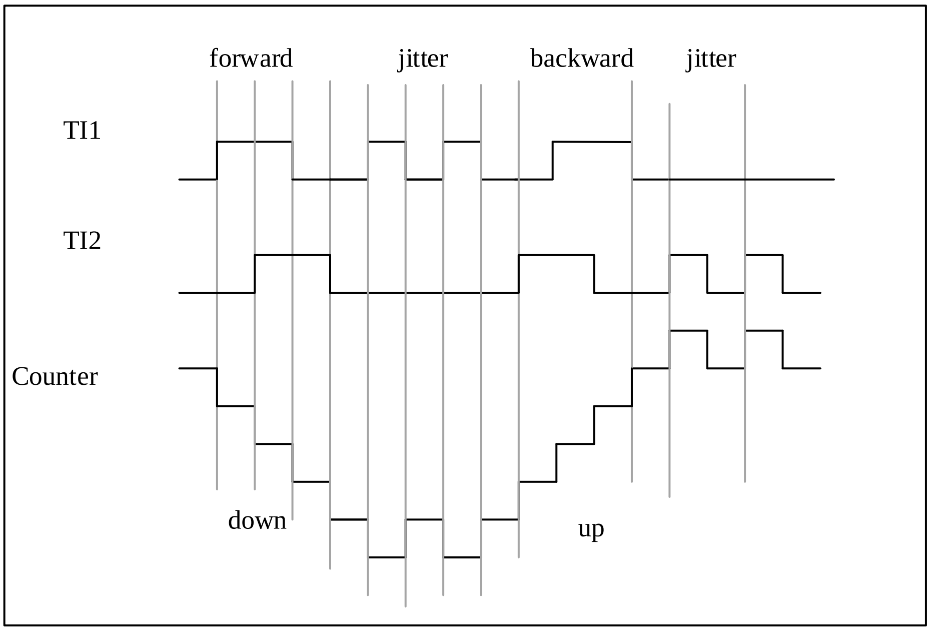
1. IC1FP1 is mapped to TI1(TIMx_CCMOD1.CC1SEL=‘01’), IC1FP1 is not inverted(TIMx_CCEN.CC1P=‘0’);
2. IC1FP2 is mapped to TI2(TIMx_CCMOD2.CC2SEL=‘01’), IC2FP2 is not inverted(TIMx_CCEN.CC2P=‘0’);
3. The input is valid on both rising and falling edges(TIMx_SMCTRL.SMSEL=‘0011’);
4. Enable counter TIMx_CTRL1.CNTEN=‘1’;

Figure 11-38 Example Of Counter Operation In Encoder Interface Mode



The following figure shows the example of counter behavior when IC1FP1 polarity is inverted (CC1P='1', other configurations are the same as above)

Figure 11-39 Encoder Interface Mode Example With IC1FP1 Polarity Inverted



11.5.16.2 Pulse Level Encoding Mode

In the pulse level encoding mode, the clock is provided on a single line on TI2, while the counting direction is provided by the TI1 input.

This mode is enabled through the SMSEL[3:0] in the TIMx_SMCTRL register, as follows。

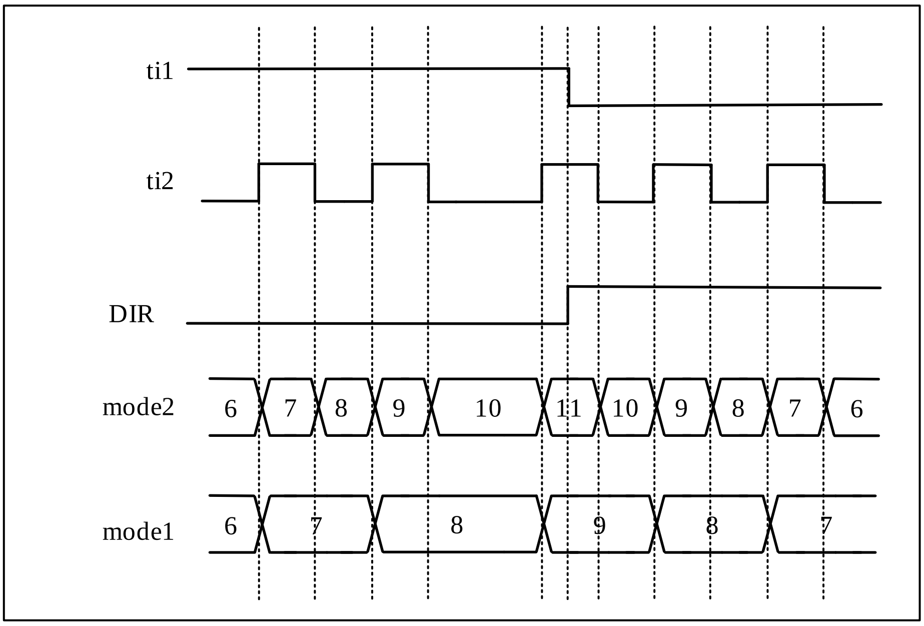
1011：In pulse level encoding mode 2, the counter is updated on both the rising and falling edges of the clock.

1100：In pulse level encoding mode 1, the counter is updated on a single clock edge based on the value of CC2P. CC2P=0 corresponds to counting on the rising edge, and CC2P=1 corresponds to counting on the falling edge.

The polarity of the direction signal of TI1 is set through the CC1P bit. When CC2P=0, counting is up when TI1 is high, counting down when TI1 is low; when CC1P=1, counting is up when TI1 is low, counting down when TI1 is high.

The following diagram takes CC1P=CC2P=0 as an example：

Figure 11-40 Pulse Level Encoding Mode (CC1P=CC2P=0)



11.5.16.3 Dual Pulse Encoding Mode

In the dual pulse encoding mode, the clock is provided on two lines, and only one line is provided at a time based on different direction. This results in one clock line for counting up and one clock line for counting down.

This mode is enabled through the SMSEL[3:0] field in the TIMx_SMCTRL register, as follows:

- 1000: In Dual Pulse Encoding Mode 2, the counter is updated on both the rising and falling edges of either of the two clock lines. The CC1P and CC2P bits encode the idle state of the clock. CCxP=0 corresponds to a high level idle state, and CCxP=1 corresponds to a low level idle state.
- 1111: In Dual Pulse Encoding Mode 1, the counter is updated on a single clock edge based on the values of the CC1P and CC2P bits. CCxP=0 corresponds to the falling edge and high level state, while CCxP=1 corresponds to the rising edge and low level state.

The table below describes the relationship between counting direction and encoder signal and polarity settings.

Table 11-11 Relationship Between Counting Direction And Encoder Signal And Polarity Settings

Dual pulse encoding mode	SMSEL[3:0]	Relative Signal Level (TI1FP1 for TI2, TI2FP2 for TI1)	TI1FP1 Signal		TI2FP2 Signal	
			Rising	Falling	Rising	Falling
Mode 2	1000	high	Counting down	Counting down	Counting up	Counting up

CCxP=0		low	Don’t count	Don’t count	Don’t count	Don’t count
Mode 2 CCxP=1	1000	high	Don’t count	Don’t count	Don’t count	Don’t count
		low	Counting down	Counting down	Counting up	Counting up
Mode 1 CCxP=0	1111	high	Don’t count	Counting down	Don’t count	Counting up
		low	Don’t count	Don’t count	Don’t count	Don’t count
Mode 1 CCxP=1	1111	high	Don’t count	Don’t count	Don’t count	Don’t count
		low	Counting down	Don’t count	Counting up	Don’t count

The following diagram shows the counting method of the dual pulse encoding mode counter.

Figure 11-41 Dual Pulse Encoding Mode (CC1P = CC2P = 0)

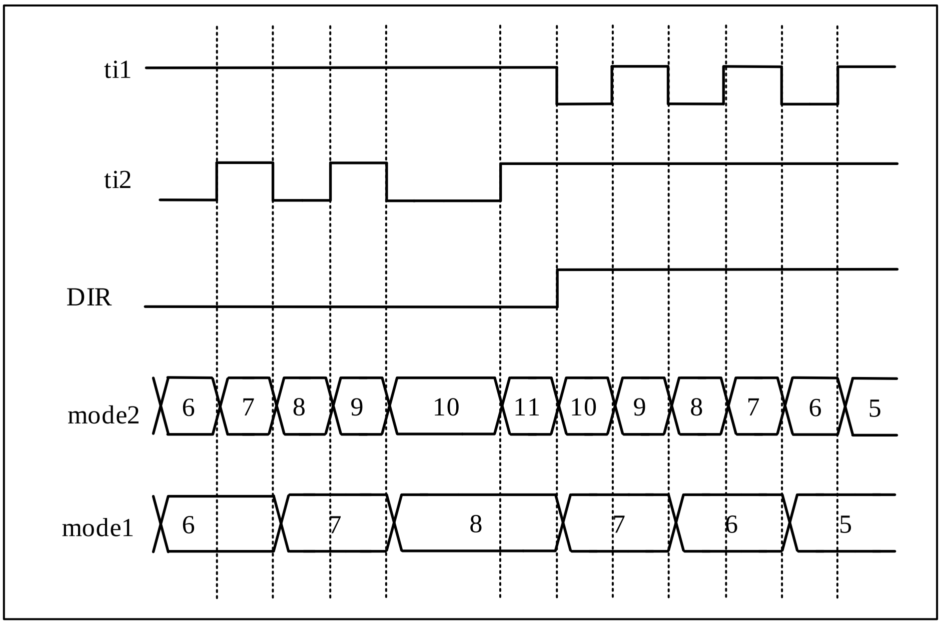
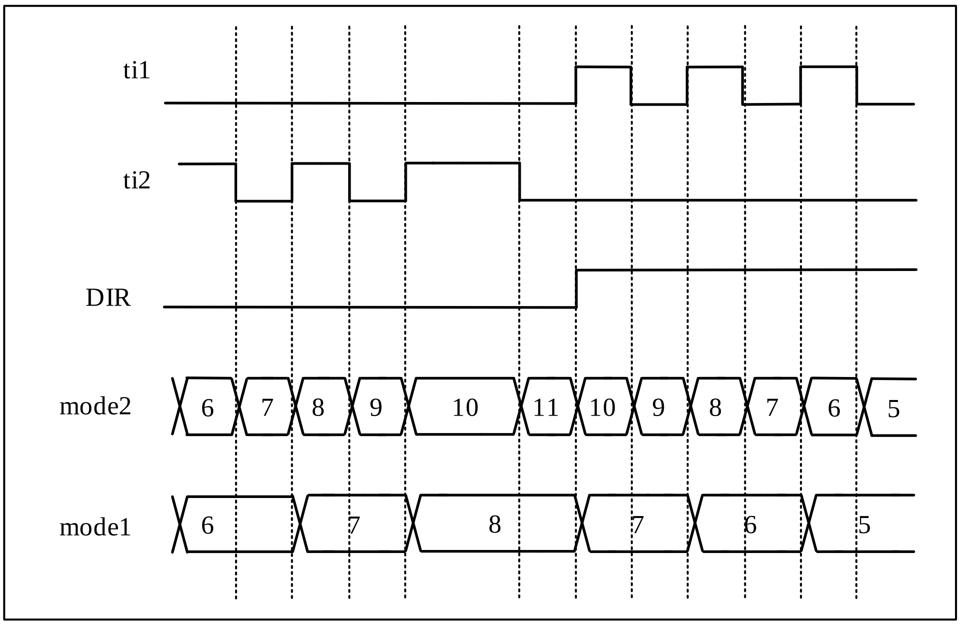


Figure 11-42 Dual Pulse Encoding Mode (CC1P = CC2P = 1)



11.5.17 Interface with Hall Sensors

See Section10.5.22.

11.6 GTIMx（x=1-7）Register Description

For abbreviations used in the register description, refer to Section1.1.

These peripheral registers can be accessed in half-word (16-bit) or word (32-bit) mode.

11.6.1 Control Register 1（TIMx_CTRL1）

Offset address：0x00

Reset value：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved												C4SEL	C3SEL	C2SEL	C1SEL
												rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		CLRSEL	Reserved			ARPEN	ONEPM	CLKD[1:0]		UPDIS	UPRS	CAMSEL[1:0]		DIR	CNTEN
		rw				rw	rw	rw		rw	rw	rw		rw	rw

Bit Field	Name	Description
31:24	Reserved	Reserved, the reset value must be maintained
19	C4SEL	Channel 4 Selection 0: Select CH4 signal (Specific selection can be seen in TIMx_INSEL.TI4S) 1: For GTIM1, select the HSE/128 input, select the CH4 signal (from HSE/128)
18	C3SEL	Channel 3 Selection 0: Select CH3 signal（Specific selection can be seen in TIMx_INSEL.TI3S） 1: For GTIM1, select the LSI input, select the CH3 signal (from LSI)
17	C2SEL	Channel 2 Selection 0: Select CH2 signal（Specific selection can be seen in TIMx_INSEL.TI2S） 1: For GTIM1, select the LSE input, select the CH2 signal (from LSE)
16	C1SEL	Channel 1 Selection 0: Choose the CH1 signal (specific selection can be seen in TIMx_INSEL.TI1S) 1: Reserved
15:14	Reserved	Reserved, the reset value must be maintained
13	CLRSEL	OcxRef Clear Selection 0: Select the external Ocxclr (TIMx_ETR) signal, specific selection can be seen in TIMx_INSEL.ETRS 1: Choose the internal Ocxclr (tim_ocref_clr) signal, specific selection can be seen in TIMx_INSEL.CLRS
12:10	Reserved	Reserved, the reset value must be maintained
9	ARPEN	Auto-reload preload enable 0: Shadow register disable for TIMx_AR register 1: Shadow register enable for TIMx_AR register
8	ONEPM	One-pulse mode 0: Disable one-pulse mode, the counter counts are not affected when an update event occurs。 1: Enable one-pulse mode, the counter stops counting when the next update event occurs
7:6	CLKD[1:0]	Clock division CLKD[1:0] indicates the division ratio between CK_INT (timer clock) and t_{DTS} (clock used for dead-time generator and digital filters (ETR, TIx)) 00: $t_{DTS} = t_{CK\_INT}$.

Bit Field	Name	Description
		01: $t_{DTS} = 2 \times t_{CK\_INT}$ 10: $t_{DTS} = 4 \times t_{CK\_INT}$ 11: Reserved, do not use this configuration
5	UPDIS	Update disable This bit is used to enable/disable the Update event (UEV) events generation by software. 0: Enable UEV. And UEV will be generated if one of following condition been fulfilled: ─ Counter overflow/underflow ─ The TIMx_EVTGEN.UDGN bit is set ─ Update generation from the slave mode controller Shadow registers will update with preload value. 1: UEV disabled. No update event is generated, and the shadow registers (AR, PSC, and CCDATx) keep their values. If the TIMx_EVTGEN.UDGN bit is set or a hardware reset is issued by the slave mode controller, the counter and prescaler are reinitialized.
4	UPRS	Update request source This bit is used to select the UEV event sources by software. 0: If update interrupt or DMA request is enabled, any of the following events will generate an update interrupt or DMA request ─ Counter overflow/underflow – TIMx_EVTGEN.UDGN bit is set ─ Update generation from the slave mode controller 1: If update interrupt or DMA request is enabled, only counter overflow/underflow will generate update interrupt or DMA request
3:2	CAMSEL[1:0]	Center-aligned mode selection 00: Edge-aligned mode. TIMx_CTRL1.DIR specifies up-counting or down-counting。 01: Center-aligned mode 1. The counter counts in center-aligned mode, and the output compare interrupt flag bit is set to 1 when down-counting。 10: Center-aligned mode 2. The counter counts in center-aligned mode, and the output compare interrupt flag bit is set to 1 when up-counting。 11: Center-aligned mode 3. The counter counts in center-aligned mode, and the output compare interrupt flag bit is set to 1 when up-counting or down-counting。 *Note: Switching from edge-aligned mode to center-aligned mode is not allowed when the counter is still enabled (TIMx_CTRL1.CNTEN = 1)*
1	DIR	Direction 0: Up-counting 1: Down-counting *Note: This bit is read-only when the counter is configured in center-aligned mode or encoder mode.*
0	CNTEN	Counter enable 0: Disable counter 1: Enable counter *Note: external clock, gating mode and encoder mode can only work after TIMx_CTRL1.CNTEN bit is set in the software. Trigger mode can automatically set*

Bit Field	Name	Description
		TIMx_CTRL1.CNTEN bit by hardware.

11.6.2 Control Register 2（TIMx_CTRL2）

Offset address: 0x04

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved												TI1SEL	Reserved	CCDSEL	Reserved
												rw		rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MMSEL[3:0]				Reserved											
rw															

Bit Field	Name	Description
31:20	Reserved	Reserved, the reset value must be maintained
19	TI1SEL	TI1 selection 0: TIMx_CH1 pin connected to TI1 input; 1: TIMx_CH1, TIMx_CH2, and TIMx_CH3 pins are XOR connected to the TI1 input.
18	Reserved	Reserved, the reset value must be maintained
17	CCDSEL	Capture/compare DMA selection 0: When a CCx event occurs, a DMA request for CCx is sent； 1: When an update event occurs, a DMA request for CCx is sent.
16	Reserved	Reserved, the reset value must be maintained
15:12	MMSEL[3:0]	Master Mode Selection These 4 bits (TIMx_CTRL2. MMSEL [3:0]) are used to select the synchronization information (TRGO) sent to the slave timer in the master mode. Possible combinations are as follows： x000: Reset –When the TIMx_EVTGEN.UDGN is set or a reset is generated by the slave mode controller, a TRGO pulse occurs. And in the later case, the signal on TRGO is delayed compared to the actual reset. x001: Enable - The TIMx_CTRL1.CNTEN bit is used as the trigger output (TRGO). Sometimes you need to start multiple timers at the same time or enable slave timer for a period of time. The counter enable signal is set when TIMx_CTRL1.CNTEN bit is set or the trigger input in gated mode is high. When the counter enable signal is controlled by the trigger input, there is a delay on TRGO except if the master/slave mode is selected (see the description of the TIMx_SMCTRL.MSMD bit). x010: Update - The update event is selected as the trigger output (TRGO). For example, a master timer clock can be used as a slave timer prescaler.

Bit Field	Name	Description
		x011: Compare pulse - Triggers the output to send a positive pulse (TRGO) when the TIMx_STS.CC1ITF is to be set (even if it is already high), when a capture or a comparison succeeds. x100: Compare - OC1REF signal is used as the trigger output (TRGO). x101: Compare - OC2REF signal is used as the trigger output (TRGO). x110: Compare - OC3REF signal is used as the trigger output (TRGO). x111: Compare - OC4REF signal is used as the trigger output (TRGO)
11:0	Reserved	Reserved, the reset value must be maintained

11.6.3 Status Registers（TIMx_STS）

Offset address: 0x08

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved													TITF	Reserved	UDITF
													rc_w0		rc_w0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				CC4OCF	CC3OCF	CC2OCF	CC1OCF	Reserved				CC4ITF	CC3ITF	CC2ITF	CC1ITF
				rc_w0	rc_w0	rc_w0	rc_w0					rc_w0	rc_w0	rc_w0	rc_w0

Bit Field	Name	Description
31:19	Reserved	Reserved, the reset value must be maintained
18	TITF	Trigger interrupt flag This bit is set by hardware when an active edge is detected on the TRGI input when the slave mode controller is in a mode other than gated. This bit is set by hardware when any edge in gated mode is detected. This bit is cleared by software. 0: No trigger event occurred； 1: Trigger interrupt occurred.
17	Reserved	Reserved, the reset value must be maintained
16	UDITF	Update interrupt flag This bit is set by hardware when an update event occurs under the following conditions： –When TIMx_CTRL1.UPDIS = 0, and repetition counter value overflow or underflow (when repetition counter equal to 0 an update event is generated). – When TIMx_CTRL1.UPRS = 0, TIMx_CTRL1.UPDIS = 0, and set the TIMx_EVTGEN.UDGN bit by software to reinitialize the CNT. –When TIMx_CTRL1.UPRS = 0, TIMx_CTRL1.UPDIS = 0, and the counter CNT is reinitialized by the trigger event. (See TIMx_SMCTRL Register description) This bit is cleared by software. 0: No update event occurred 1: Update interrupt occurred
15:12	Reserved	Reserved, the reset value must be maintained

Bit Field	Name	Description
11	CC4OCF	Capture/Compare 4 overcapture flag See TIMx_STS.CC1OCF description.
10	CC3OCF	Capture/Compare 3 overcapture flag See TIMx_STS.CC1OCF description.
9	CC2OCF	Capture/Compare 2 overcapture flag See TIMx_STS.CC1OCF description.
8	CC1OCF	Capture/Compare 1 overcapture flag This bit is set by hardware only when the corresponding channel is configured in input capture mode. Cleared by software writing 0. 0: No overcapture occurred; 1: TIMx_STS.CC1ITF was already set when the value of the counter has been captured in the TIMx_CCDAT1 register.
7:4	Reserved	Reserved, the reset value must be maintained.
3	CC4ITF	Capture/Compare 4 interrupt flag See TIMx_STS.CC1ITF description.
2	CC3ITF	Capture/Compare 3 interrupt flag See TIMx_STS.CC1ITF description.
1	CC2ITF	Capture/Compare 2 interrupt flag See TIMx_STS.CC1ITF description.
0	CC1ITF	Capture/Compare 1 interrupt flag **When the corresponding channel of CC1 is in output mode:** Except in center-aligned mode, this bit is set by hardware when the counter value is the same as the compare value (see TIMx_CTRL1.CAMSEL bit description). This bit is cleared by software. 0: No match occurred. 1: The value of TIMx_CNT is the same as the value of TIMx_CCDAT1. When the value of TIMx_CCDAT1 is greater than the value of TIMx_AR, the TIMx_STS.CC1ITF bit will go high if the counter overflows (in up-counting and up/down-counting modes) and underflows in down-counting mode. **When the corresponding channel of CC1 is in input mode：** This bit is set by hardware when the capture event occurs. This bit is cleared by software or by reading TIMx_CCDAT1. 0: No input capture occurred. 1: Input capture occurred. Counter value has captured in the TIMx_CCDAT1. An edge with the same polarity as selected has been detected on IC1.

11.6.4 Event Generation Registers（TIMx_EVTGEN）

Offset address: 0x0C

Reset values: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved					TGN	Reserved	UDGN	Reserved				CC4GN	CC3GN	CC2GN	CC1GN
					w		w					w	w	w	w

Bit Field	Name	Description
31:11	Reserved	Reserved, the reset value must be maintained
10	TGN	Trigger generation This bit can generate a trigger event when set by software. And at this time TIMx_STS.TITF = 1, if the corresponding interrupt and DMA are enabled, the corresponding interrupt and DMA will be generated. This bit is automatically cleared by hardware. 0: No action No action 1: Generated a trigger event
9	Reserved	Reserved, the reset value must be maintained
8	UDGN	Update generation This bit can generate an update event when set by software. And at this time the counter will be reinitialized, the prescaler counter will be cleared, the counter will be cleared in center-aligned or up-counting mode, but take TIMx_AR in down-counting mode the value of the register. This bit is automatically cleared by hardware. 0: No action 1: Generated an update event
7:4	Reserved	Reserved, the reset value must be maintained
3	CC4GN	Capture/Compare 4 generation See TIMx_EVTGEN.CC1GN description.
2	CC3GN	Capture/Compare 3 generation See TIMx_EVTGEN.CC1GN description.
1	CC2GN	Capture/Compare 2 generation See TIMx_EVTGEN.CC1GN description.
0	CC1GN	Capture/Compare 1 generation This bit can generate a capture/compare event when set by software. This bit is automatically cleared by hardware. **When the corresponding channel of CC1 is in output mode:** The TIMx_STS.CC1ITF flag will be pulled high, if the corresponding interrupt and DMA are enabled, the corresponding interrupt and DMA will be generated. **When the corresponding channel of CC1 is in input mode:** TIMx_CCDAT1 will capture the current counter value, and the TIMx_STS.CC1ITF flag will be pulled high, if the corresponding interrupt and DMA are enabled, the corresponding interrupt and DMA will be generated. If The IMx_STS.CC1ITF is already pulled high, pull TIMx_STS.CC1OCF high. 0: No action 1: Generated a CC1 capture/compare event

11.6.5 Slave Mode Control Register（TIMx_SMCTRL）

Offset address: 0x10

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								OCREFCLRF[3:0]				OCREF CLRP	Reserved		MSMD
								rw				rw			rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTF[3:0]				EXTP	EXCEN	EXTPS		SMSEL[3:0]				Reserved	TSEL[2:0]		
rw				rw	rw	rw		rw					rw		

Bit Field	Name	Description
31:24	Reserved	Reserved, the reset value must be maintained
23:20	OCREFCLRF	tim_ocref_clr signal filter These bits are used to define the frequency at which the tim_ocref_clk signal is sampled and the bandwidth of the tim_ocref_clr digital filtering. In effect, the digital filter is an event counter that generates a validate output after consecutive N events are recorded. 0000: No filter, sampling 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: $f_{SAMPLING}=f_{DTS/4}$， N=6 1110: $f_{SAMPLING}=f_{DTS/32}$， N=6 0111: $f_{SAMPLING}=f_{DTS/4}$， N=81111: $f_{SAMPLING}=f_{DTS/32}$， N=8
19	OCREFCLRP	tim_ocref_clr signal polarity This bit is used to select whether the trigger operation is to use tim_ocref_clr or the inversion of tim_ocref_clr. 0: tim_ocref_clr active at high level or rising edge. 1: tim_ocref_clr active at a low level or falling edge.
18:17	Reserved	Reserved, the reset value must be maintained
16	MSMD	Master/slave mode 0: No effect; 1: The event on the trigger input (TRGI) is delayed to allow perfect synchronization between the current timer (via TRGO) and its slave timers. This is very useful for cases where multiple timers need to be synchronized to a single external event.
15:12	EXTF[3:0]	External trigger filter These bits are used to define the frequency at which the ETRP signal is sampled and the bandwidth of the ETRP digital filtering. In effect, the digital filter is an event counter that generates a validate output after consecutive N events are recorded. 0000: No filter, sampling 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: $f_{SAMPLING}$=$f_{DTS/4}$，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
11	EXTP	External trigger polarity This bit is used to select whether the trigger operation is to use tim_etr_in or the inversion of tim_etr_in. 0: tim_etr_in active at high level or rising edge. 1: tim_etr_in active at low level or falling edge.
10	EXCEN	External clock enable This bit is used to enable external clock mode 2, and the counter is driven by any active edge on the ETRF signal in this mode. 0：External clock mode 2 disable. 1：External clock mode 2 enable. *Note1: When external clock mode 1 and external clock mode 2 are enabled at the same time, the input of the external clock is ETRF.* *Note2: The following slave modes can be used simultaneously with external clock mode 2: reset mode, gated mode and trigger mode; However, TRGI cannot connect to ETRF (TIMx_SMCTRL.TSEL ≠ '111').* *Note3: Setting the TIMx_SMCTRL.EXCEN bit has the same effect as selecting external clock mode 1 and connecting TRGI to ETRF (TIMx_SMCTRL.SMSEL = 111 and TIMx_SMCTRL.TSEL = 111).*
9:8	EXTPS[1:0]	External trigger prescaler The frequency of the external trigger signal ETRP must be at most 1/4 of TIMxCLK frequency. When a faster external clock is input, a prescaler can be used to reduce the frequency of ETRP. 00: Prescaler disable 01: ETRP frequency divided by 2 10: ETRP frequency divided by 4 11: ETRP frequency divided by 8
7:4	SMSEL[3:0]	Slave mode selection When an external signal is selected, the active edge of the trigger signal (TRGI) is linked to the selected external input polarity (see input control register and control register description) 0000 : Disable slave mode. If TIMx_CTRL1.CNTEN = 1, the prescaler is driven directly by the internal clock.。 0001 : Encoder mode 1. According to the level of TI2FP2, the counter up-counting or down-counting on the edge of TI1FP1。 0010 : Encoder mode 2. According to the level of TI1FP1, the counter up-counting or down-counting on the edge of TI2FP2. 0011 : Encoder mode 3. According to the input level of another signal, the counter up-counting or down-counting on the edges of TI2FP1 and TI2FP2 0100 : Reset mode. On the rising edge of the selected trigger input (TRGI), the counter is reinitialized and the shadow register is updated

		0101 : Gated mode. When the trigger input (TRGI) is high, the clock of the counter is enabled. Once the trigger input becomes low, the counter stops counting, but is not reset. In this mode, the start and stop of the counter are controlled 0110 : Trigger mode. When a rising edge occurs on the trigger input (TRGI), the counter is started but not reset. In this mode, only the start of the counter is controlled. 0111 : External clock mode 1. The counter is clocked by the rising edge of the selected trigger input (TRGI). 1000 : Dual input mode2. 1001 : Quadrature encoder mode 4 - The counter counts up/down on the edge of TI1FP1 according to the level of TI2FP2. Select the counting edge with CC1P. 1010 : Quadrature encoder mode 5 - The counter counts up/down on the edge of TI2FP2 according to the level of TI1FP1. Select the counting edge with CC2P 1011 : Pulse level encoding mode 2. 1100 : Pulse level encoding mode 1. Set the counting edge of TI2FP2 with CC2P. 1101 : Combined gated + reset mode - When the trigger input (TRGI) is high, the counter's clock is enabled. Once the trigger input becomes low, the counter stops (and resets). The start and stop of the counter are both controlled. 1110 : Combined reset + trigger mode - The counter starts (and resets) on the rising edge of the trigger input TRGI, with only the start of the counter being controlled 1111 : Dual input mode 1. Set the counting sensitive edges of TI1FP1 and TI2FP2 with CC1P and CC2P. *Note: Do not use gated mode if TI1F_ED is selected as the trigger input (TIMx_SMCTRL.TSEL=100). This is because TI1F_ED outputs a pulse for each TI1F transition, whereas gated mode checks the level of the triggered input.*
3	Reserved	Reserved, the reset value must be maintained
2:0	TSEL[2:0]	Trigger selection These 3 bits are used to select the trigger input of the synchronous counter. 0xx：Internal trigger x (ITRx), Select the ITR signal source based on TIMx_INSEL. ITRS 100：Edge detector for TI1 (TI1F_ED) 101：Filtered timer input 1 (TI1FP1) 110：Filtered timer input 2 (TI2FP2) 111：External trigger input (ETRF) *Note: These bits must be changed only when not in use (e. g. TIMx_SMCTRL.SMSEL=000) to avoid false edge detection at the transition.*

11.6.6 DMA/Interrupt Enable Register（TIMx_DINTEN）

Offset address: 0x14

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

Reserved										TDEN	Reserved	UDEN	Reserved	TIEN	UIEN
										rw		rw		rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				CC4DEN	CC3DEN	CC2DEN	CC1DEN	Reserved				CC4IEN	CC3IEN	CC2IEN	CC1IEN
				rw	rw	rw	rw					rw	rw	rw	rw

Bit Field	Name	Description
31:22	Reserved	Reserved, the reset value must be maintained
21	TDEN	Trigger DMA request enable 0: Disable trigger DMA request. 1: Enable trigger DMA request
20	Reserved	Reserved, the reset value must be maintained
19	UDEN	Update DMA request enable 0: Disable update DMA request. 1: Enable update DMA request
18	Reserved	Reserved, the reset value must be maintained
17	TIEN	Trigger interrupt enable 0: Disable trigger interrupt 1: Enable trigger interrupt
16	UIEN	Update interrupt enable 0: Disable update interrupt 1: Enables update interrupt
15:12	Reserved	Reserved, the reset value must be maintained
11	CC4DEN	Capture/Compare 4 DMA request enable 0: Disable capture/compare 4 DMA request 1: Enable capture/compare 4 DMA request
10	CC3DEN	Capture/Compare 3 DMA request enable 0: Disable capture/compare 3 DMA request 1: Enable capture/compare 3 DMA request
9	CC2DEN	Capture/Compare 2 DMA request enable 0: Disable capture/compare 2 DMA request 1: Enable capture/compare 2 DMA request
8	CC1DEN	Capture/Compare 1 DMA request enable 0: Disable capture/compare 1 DMA request 1: Enable capture/compare 1 DMA request
7:4	Reserved	Reserved, the reset value must be maintained
3	CC4IEN	Capture/Compare 4 interrupt enable 0: Disable capture/compare 4 interrupt 1: Enable capture/compare 4 interrupt
2	CC3IEN	Capture/Compare 3 interrupt enable 0: Disable capture/compare 3 interrupt 1: Enable capture/compare 3 interrupts

Bit Field	Name	Description
1	CC2IEN	Capture/Compare 2 interrupt enable 0: Disable capture/compare 2 interrupt 1: Enables capture/compare 2 interrupts
0	CC1IEN	Capture/Compare 1 interrupt enable 0: Disable capture/compare 1 interrupt 1: Enables capture/comparing 1 interrupt

11.6.7 Capture/Compare Mode Register 1（TIMx_CCMOD1）

Offset address: 0x18

Reset value: 0x0000 0000

Channels can be used for input (capture mode) or output (compare mode), and the direction of the channel is defined by the corresponding CCxSEL bit. The other bits of the register act differently in input and output modes. OCx describes the function of a channel in output mode, ICx describes the function of a channel in input mode. Hence, please note that the same bit can have different meanings for output mode and for input mode.

Output compare mode：

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC2MD[2:0]			OC2CEN	OC2FEN	OC2PEN	CC2SEL[1:0]		OC1MD[2:0]			OC1CEN	OC1FEN	OC1PEN	CC1SEL[1:0]	
rw			rw	rw	rw	rw		rw			rw	rw	rw	rw	

Bit Field	Name	Description
31:16	Reserved	Reserved, the reset value must be maintained
15:13	OC2MD[2:0]	Output Compare 2 mode
12	OC2CEN	Output Compare 2 clear enable
11	OC2FEN	Output Compare 2 fast enable
10	OC2PEN	Output Compare 2 preload enable
9:8	CC2SEL[1:0]	Capture/Compare 2 selection These bits are used to select the input/output and input mapping of the channel 00: CC2 channel is configured as output. 01: CC2 channel is configured as input, IC2 is mapped on TI2 10: CC2 channel is configured as input, IC2 is mapped on TI1 11: CC2 channel is configured as input, IC2 is mapped on TRC. This mode is only active when the internal trigger input is selected by TIMx_SMCTRL.TSEL. *Note: CC2SEL is writable only when the channel is off (TIMx_CCEN.CC2EN = 0).*
7:5	OC1MD[2:0]	Output Compare 1 mode These bits are used to manage the output reference signal OC1REF, which determines the

Bit Field	Name	Description
		values of OC1 and OC1N, and is valid at high levels, while the active levels of OC1 and OC1N depend on the TIMx_CCEN.CC1P and TIMx_CCEN.CC1NP bits. 000: Frozen. Comparison between TIMx_CCDAT1 register and counter TIMx_CNT has no effect on OC1REF signal. 001: Set channel 1 to the active level on match. When TIMx_CCDAT1 = TIMx_CNT, OC1REF signal will be forced high. 010: Set channel 1 as inactive level on match. When TIMx_CCDAT1 = TIMx_CNT, OC1REF signal will be forced low. 011: Toggle. When TIMx_CCDAT1 = TIMx_CNT, OC1REF signal will be toggled. 100: Force to inactive level. OC1REF signal is forced low. 101: Force to active level. OC1REF signal is forced high. 110: PWM mode 1 - In up-counting mode, if TIMx_CNT < TIMx_CCDAT1, OC1REF signal of channel 1 is high, otherwise it is low. In down-counting mode, if TIMx_CNT > TIMx_CCDAT1, OC1REF signal of channel 1 is low, otherwise it is high. 111: PWM mode 2 - In up-counting mode, if TIMx_CNT < TIMx_CCDAT1, OC1REF signal of channel 1 is low, otherwise it is high. In down-counting mode, if TIMx_CNT > TIMx_CCDAT1, OC1REF signal of channel 1 is high, otherwise it is low. *Note 1: In PWM mode 1 or PWM mode 2, the OC1REF level changes only when the comparison result changes or when the output compare mode is switched from frozen mode to PWM mode.*
4	OC1CEN	Output Compare 1 clear enable 0: OC1REF is not affected by tim_ocref_clr_in input level. 1: OC1REF is cleared immediately when the tim_ocref_clr_in input level is detected as high(the source of tim_ocref_clr_in is controlled by TIMx_CTRL1.CLRSEL).
3	OC1FEN	Output Compare 1 fast enable This bit is used to speed up the response of the CC output to the trigger input event. 0: CC1 behaves normally depending on the counter and CCDAT1 values, even if the trigger is ON. The minimum delay for activating CC1 output when an edge occurs on the trigger input is 5 clock cycles 1: An active edge of the trigger input acts like a comparison match on CC1 output. Therefore, OC is set to the comparison level regardless of the comparison result. The delay time for sampling the trigger input and activating the CC1 output is reduced to 3 clock cycles. OCxFEN only works if the channel is configured in PWM1 or PWM2 mode.
2	OC1PEN	Output Compare 1 preload enable 0: Disable preload function of TIMx_CCDAT1 register. Supports write operations to TIMx_CCDAT1 register at any time, and the written value is effective immediately. 1: Enable preload function of TIMx_CCDAT1 register. Only read and write operations to preload registers. When an update event occurs, the value of TIMx_CCDAT1 is loaded into the active register. *Note 1: Only when TIMx_CTRL1.ONEPM = 1(In one-pulse mode), PWM mode can be used without verifying the preload register, otherwise no other behavior can be predicted.*
1:0	CC1SEL[1:0]	Capture/Compare 1 selection

Bit Field	Name	Description
		These bits are used to select the input/output and input mapping of the channel 00: CC1 channel is configured as output. 01: CC1 channel is configured as input, IC1 is mapped on TI1. 10: CC1 channel is configured as input, IC1 is mapped on TI2 11: CC1 channels are configured as inputs and IC1 is mapped to TRC. This mode is only active when the internal trigger input is selected by TIMx_SMCTRL.TSEL. *Note: CC1SEL is writable only when the channel is off (TIMx_CCEN.CC1EN = 0).*

Input capture mode:

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IC2F[3:0]				IC2PSC[1:0]		CC2SEL[1:0]		IC1F[3:0]				IC1PSC[1:0]		CC1SEL[1:0]	
rw				rw		rw		rw				rw		rw	

Bit Field	Name	Description
31:16	Reserved	Reserved, the reset value must be maintained
15:12	IC2F[3:0]	Input capture 2 filter
11:10	IC2PSC[1:0]	Input capture 2 prescaler
9:8	CC2SEL[1:0]	Capture/Compare 2 selection These bits are used to select the input/output and input mapping of the channel: 00: CC2 channel is configured as output 01: CC2 channel is configured as input, IC2 is mapped on TI2 10: CC2 channel is configured as input, IC2 is mapped on TI1 11: CC2 channel is configured as input, IC2 is mapped on TRC. This mode is only active when the internal trigger input is selected by TIMx_SMCTRL.TSEL. *Note: CC2SEL is writable only when the channel is off (TIMx_CCEN.CC2EN = 0).*
7:4	IC1F[3:0]	Input capture 1 filter These bits are used to define sampling frequency of TI1 input and the length of digital filter. The digital filter is an event counter that generates an output transition after N events are recorded. 0000: No filter, sampling 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: $f_{SAMPLING}$=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
3:2	IC1PSC[1:0]	Input capture 1 prescaler These bits are used to select the ratio of the prescaler for IC1 (CC1 input). When TIMx_CCEN.CC1EN = 0, the prescaler will be reset. 00: No prescaler, capture is done each time an edge is detected on the capture input 01: Capture is done once every 2 events 10: Capture is done once every 4 events 11: Capture is done once every 8 events

1:0	CC1SEL[1:0]	Capture/Compare 1 Selection These bits are used to select the input/output and input mapping of the channel 00: CC1 channel is configured as output 01: CC1 channel is configured as input, IC1 is mapped on TI1 10: CC1 channel is configured as input, IC1 is mapped on TI2 11: CC1 channel is configured as input, IC1 is mapped to TRC. This mode is only active when the internal trigger input is selected by TIMx_SMCTRL.TSEL. *Note: CC1SEL is writable only when the channel is off (TIMx_CCEN.CC1EN = 0).*

11.6.8 Capture/Compare Mode Register 2（TIMx_CCMOD2）

Offset address: 0x1C

Reset value: 0x0000 0000

See the description of the CCMOD1 register above

Output comparison mode:

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC4MD[2:0]			OC4CEN	OC4FEN	OC4PEN	CC4SEL[1:0]		OC3MD[2:0]			OC3CEN	OC3FEN	OC3PEN	CC3SEL[1:0]	
rw			rw	rw	rw	rw		rw			rw	rw	rw	rw	

Bit Field	Name	Description
31:16	Reserved	Reserved, the reset value must be maintained
15:13	OC4MD[2:0]	Output compare 4 mode
12	OC4CEN	Output compare 4 clear enable
11	OC4FEN	Output compare 4 fast enable
10	OC4PEN	Output compare 4 preload enable
9:8	CC4SEL[1:0]	Capture/Compare 4 selection These bits are used to select the input/output and input mapping of the channel 00: CC4 channel is configured as output 01: CC4 channel is configured as input, IC4 is mapped on TI4 10: CC4 channel is configured as input, IC4 is mapped on TI3 11: CC4 channel is configured as input, IC4 is mapped on TRC. This mode is only active when the internal trigger input is selected by TIMx_SMCTRL.TSEL. *Note: CC4SEL is writable only when the channel is off (TIMx_CCEN.CC4EN = 0).*
7:5	OC3MD[2:0]	Output compare 3 mode
4	OC3CEN	Output compare 3 clear enable
3	OC3FEN	Output compare 3 fast enable
2	OC3PEN	Output compare 3 preload enable
1:0	CC3SEL[1:0]	Capture/Compare 3 selection

		These bits are used to select the input/output and input mapping of the channel 00: CC3 channel is configured as output 01: CC3 channel is configured as input, IC3 is mapped to TI3 10: CC3 channel is configured as input, IC3 is mapped on TI4 11: CC3 channel is configured as input, IC3 is mapped to TRC. This mode is only active when the internal trigger input is selected by TIMx_SMCTRL.TSEL. *Note: CC3SEL is writable only when the channel is off (TIMx_CCEN.CC3EN = 0).*

Input capture mode:

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IC4F[3:0]				IC4PSC[1:0]		CC4SEL[1:0]		IC3F[3:0]				IC3PSC[1:0]		CC3SEL[1:0]	
rw				rw		rw		rw				rw		rw	

Bit Field	Name	Description
31:16	Reserved	Reserved, the reset value must be maintained
15:12	IC4F[3:0]	Input capture 4 filter
11:10	IC4PSC[1:0]	Input capture 4 prescaler
9:8	CC4SEL[1:0]	Capture/Compare 4 selection These bits are used to select the input/output and input mapping of the channel 00: CC4 channel is configured as output 01: CC4 channel is configured as input, IC4 is mapped on TI4 10: CC4 channel is configured as input, IC4 is mapped on TI3 11: CC4 channel is configured as input, IC4 is mapped on TRC. This mode is only active when the internal trigger input is selected by TIMx_SMCTRL.TSEL. *Note: CC4SEL is writable only when the channel is off (TIMx_CCEN.CC4EN = 0).*
7:4	IC3F[3:0]	Input capture 3 filter
3:2	IC3PSC[1:0]	Input capture 3 prescaler
1:0	CC3SEL[1:0]	Capture/compare 3 selection These bits are used to select the input/output and input mapping of the channel 00: CC3 channel is configured as output 01: CC3 channel is configured as input, IC3 is mapped to TI3 10: CC3 channel is configured as input, IC3 is mapped on TI4 11: CC3 channel is configured as input, IC3 is mapped to TRC. This mode is only active when the internal trigger input is selected by TIMx_SMCTRL.TSEL. *Note: CC3SEL is writable only when the channel is off (TIMx_CCEN.CC3EN = 0).*

11.6.9 Capture/Compare Enable Registers（TIMx_CCEN）

Offset address: 0x24

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CC4P	CC4EN	Reserved		CC3P	CC3EN	Reserved		CC2P	CC2EN	Reserved		CC1P	CC1EN	Reserved	
rw	rw			rw	rw			rw	rw			rw	rw		

Bit Field	Name	Description
31:16	Reserved	Reserved, the reset value must be maintained.
15	CC4P	Capture/Compare 4 output polarity See TIMx_CCEN.CC1P description.
14	CC4EN	Capture/Compare 4 output enable See TIMx_CCEN.CC1EN description.
13:12	Reserved	Reserved, the reset value must be maintained.
11	CC3P	Capture/Compare 3 output polarity See TIMx_CCEN.CC1P description.
10	CC3EN	Capture/Compare 3 output enabl） See TIMx_CCEN.CC1EN description.
9:8	Reserved	Reserved, the reset value must be maintained.
7	CC2P	Capture/Compare 2 output polarity See TIMx_CCEN.CC1P description.
6	CC2EN	Capture/Compare 2 output enable See TIMx_CCEN.CC1EN description.
5:4	Reserved	Reserved, the reset value must be maintained.
3	CC1P	Capture/Compare 1 output polarity **When the corresponding channel of CC1 is in output mode:** 0: OC1 active high 1: OC1 active low **When the corresponding channel of CC1 is in input mode:** At this time, this bit is used to select whether IC1 or the inverse signal of IC1 is used as the trigger or capture signal. 0: non-inverted: Capture action occurs when IC1 generates a rising edge. When used as external trigger, IC1 is non-inverted. 1: inverted: Capture action occurs when IC1 generates a falling edge. When used as external trigger, IC1 is inverted.
2	CC1EN	Capture/Compare 1 output enable **When the corresponding channel of CC1 is in output mode:** 0: Disable - Disable output OC1 signal. Therefore, the output level of OC1 depends on the values of MOEN, OSSI, OSSR, OI1, OI1N, and CC1NEN bits. 1: Enable - Enable output OC1 signal. The OC1 signal is output to the corresponding output pin, and its output level depends on the values of the MOEN, OSSI, OSSR, OI1, OI1N, and CC1NEN bits.

Bit Field	Name	Description
		When the corresponding channel of CC1 is in input mode: At this time, this bit is used to disable/enable the capture function. 0: Disable capture 1: Enable capture
1:0	Reserved	Reserved, the reset value must be maintained.

Table 11-12 Output Control Bits Of Standard Ocx Channel

CCxEN	OCx output status
0	Disable output (OCx=0)
1	OCx = OCxREF + polarity

Note: The state of external I/O pins connected to standard OCx channels depends on the OCx channel state and GPIO and AFIO registers.

11.6.10 Capture/Compare Register 1（TIMx_CCDAT1）

Offset address: 0x28

Reset value: 0x0000 0000

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="16">Reserved</td></tr>
</table>

<table>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="16">CCDAT1[15:0]</td></tr>
<tr><td colspan="16">rw</td></tr>
</table>

Bit field	Name	Description
31:16	Reserved	Reserved, the reset value must be maintained.
15:0	CCDAT1[15:0]	Capture/Compare 1 value **CC1 channel is configured as output:** CCDAT1 contains the value to be compared to the counter TIMx_CNT, signaling on the OC1 output. If the preload feature is not selected in TIMx_CCMOD1.OC1PEN bit, the written value is immediately transferred to the active register. Otherwise, this preloaded value is transferred to the active register only when an update event occurs. **CC1 channel is configured as input:** CCDAT1 contains the counter value transferred by the last input capture 1 event (IC1). When configured as input mode, register CCDAT1 is only readable. When configured as output mode, register CCDAT1 is readable and writable.

11.6.11 Capture/Compare Register 2（TIMx_CCDAT2）

Offset address: 0x2C

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCDAT2[15:0]															
rw															

Bit field	Name	Description
31:16	Reserved	Reserved, the reset value must be maintained.
15:0	CCDAT2[15:0]	Capture/Compare 2 value **CC2 channel is configured as output:** CCDAT2 contains the value to be compared to the counter TIMx_CNT, signaling on the OC2 output. If the preload feature is not selected in TIMx_CCMOD1.OC2PEN bit, the written value is immediately transferred to the active register. Otherwise, this preloaded value is transferred to the active register only when an update event occurs. **CC2 channel is configured as input:** CCDAT2 contains the counter value transferred by the last input capture 2 event (IC2). When configured as input mode, register CCDAT2 is only readable. When configured as output mode, register CCDAT2 is readable and writable.

11.6.12 Capture/Compare Register 3（TIMx_CCDAT3）

Offset address: 0x30

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCDAT3[15:0]															
rw															

Bit field	Name	Description
31:16	Reserved	Reserved, the reset value must be maintained.
15:0	CCDAT3[15:0]	Capture/Compare 3 value **CC3 channel is configured as output:** CCDAT3 contains the value to be compared to the counter TIMx_CNT, signaling on the OC3 output. If the preload feature is not selected in TIMx_CCMOD2.OC3PEN bit, the written value is immediately transferred to the active register. Otherwise, this preloaded value is transferred to the active register only when an update event occurs. **CC3 channel is configured as input:**

Bit field	Name	Description
31:16	Reserved	Reserved, the reset value must be maintained.
		CCDAT3 contains the counter value transferred by the last input capture 3 event (IC3). When configured as input mode, register CCDAT3 is only readable. When configured as output mode, register CCDAT3 is readable and writable.

11.6.13 Capture/Compare Register 4（TIMx_CCDAT4）

Offset address: 0x34

Reset value: 0x0000 0000

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="16">Reserved</td></tr>
</table>

<table>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="16">CCDAT4[15:0]</td></tr>
<tr><td colspan="16">rw</td></tr>
</table>

Bit field	Name	Description
31:16	Reserved	Reserved, the reset value must be maintained.
15:0	CCDAT4[15:0]	Capture/Compare 4 value **CC4 channel is configured as output:** CCDAT4 contains the value to be compared to the counter TIMx_CNT, signaling on the OC4 output. If the preload feature is not selected in TIMx_CCMOD2.OC4PEN bit, the written value is immediately transferred to the active register. Otherwise, this preloaded value is transferred to the active register only when an update event occurs. **CC4 channel is configured as input:** CCDAT4 contains the counter value transferred by the last input capture 4 event (IC4). When configured as input mode, register CCDAT4 is only readable. When configured as output mode, register CCDAT4 is readable and writable.

11.6.14 Prescaler（TIMx_PSC）

Offset address: 0x40

Reset value: 0x0000 0000

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="16">Reserved</td></tr>
</table>

<table>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="16">PSC[15:0]</td></tr>
<tr><td colspan="16">rw</td></tr>
</table>

Bit field	Name	Description
31:16	Reserved	Reserved, the reset value must be maintained.
15:0	PSC[15:0]	Prescaler value Counter clock $f_{CK\_CNT} = f_{CK\_PSC}/$ (PSC [15:0] +1). Each time an update event occurs, the PSC value is loaded into the active prescaler register.

11.6.15 Auto-Reload Register（TIMx_AR）

Offset address: 0x44

Reset value: 0x0000FFFF

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="16">Reserved</td></tr>
</table>

<table>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="16">AR[15:0]</td></tr>
<tr><td colspan="16">rw</td></tr>
</table>

Bit field	Name	Description
31:16	Reserved	Reserved, the reset value must be maintained.
15:0	AR[15:0]	Auto-reload value These bits define the value that will be loaded into the actual auto-reload register。See Section 10.5.1 for more details. When the TIMx_AR.AR [15:0] value is null, the counter does not work.

11.6.16 Counter（TIMx_CNT）

Offset address: 0x48

Reset value: 0x0000 0000

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="16">Reserved</td></tr>
</table>

<table>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="16">CNT[15:0]</td></tr>
<tr><td colspan="16">rw</td></tr>
</table>

Bit field	Name	Description
31:16	Reserved	Reserved, the reset value must be maintained.
15:0	CNT[15:0]	Counter value

11.6.17 Channel 1 Filter Register（TIMx_C1FILT）

Offset address: 0x64

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		THRESH[5:0]						Reserved	WSIZE[5:0]						FILTEN
		rw							rw						rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															

Bit field	**Name**	**Description**
31:30	Reserved	Reserved, the reset value must be maintained.
29:24	THRESH[5:0]	Threshold number of valid logic level samples, maximum 63: the threshold for valid logic levels. Within the sampling window, if the quantity of logic highs is greater than or equal to the threshold, the next logic level will be a logic high. The same rule applies to logic lows. If the quantity of high and low logic levels within the window is less than the threshold, the filter output remains unchanged. The threshold should be set to a value greater than or equal to half of the Window value Recommended threshold range: **Minimum value:** one additional pre-scaler clock period beyond the maximum glitch size limit (pre-scaler clock period), and must be greater than half of the window size. For example, if the glitch size is 3.2 * (the pre-scaler clock period), then the threshold should be ┌3.2┐ = 4 + 1 =5 **Maximum value:** the lower limit of the minimum size of the valid signal (in pre-scaler clock periods), and must be less than the window size. For example, if the minimum signal size is 3.2 * (the pre-scaler clock period), then the threshold should be the lower limit (3.2) = 3.
23	Reserved	Reserved, the reset value must be maintained.
22:17	WSIZE[5:0]	Window size value for logic level checking, maximum 63: The window size determines how many sample values will be considered when obtaining the next logic level. The built-in FIFO is 64 bits, with a maximum index of 63, so the window size can only be set to 63.
16	FILTEN	Filter enable: 0: Filter disable 1: Filter enable
15:0	Reserved	Reserved, the reset value must be maintained.

11.6.18 Channel 2 Filter Register（TIMx_C2FILT）

Offset address: 0x68

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		THRESH[5:0]						Reserved	WSIZE[5:0]						FILTEN

				rw							rw				rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															

Bit field	Name	Description
31:30	Reserved	Reserved, the reset value must be maintained.
29:24	THRESH[5:0]	Threshold number of valid logic level samples, maximum 63: the threshold for valid logic levels. Within the sampling window, if the quantity of logic highs is greater than or equal to the threshold, the next logic level will be a logic high. The same rule applies to logic lows. If the quantity of high and low logic levels within the window is less than the threshold, the filter output remains unchanged. The threshold should be set to a value greater than or equal to half of the Window value Recommended threshold range: **Minimum value:** one additional pre-scaler clock period beyond the maximum glitch size limit (pre-scaler clock period), and must be greater than half of the window size. For example, if the glitch size is 3.2 * (the pre-scaler clock period), then the threshold should be ┌3.2┐= 4 + 1 =5 **Maximum value:** the lower limit of the minimum size of the valid signal (in pre-scaler clock periods), and must be less than the window size. For example, if the minimum signal size is 3.2 * (the pre-scaler clock period), then the threshold should be the lower limit (3.2) = 3.
23	Reserved	Reserved, the reset value must be maintained.
22:17	WSIZE[5:0]	Window size value for logic level checking, maximum 63: The window size determines how many sample values will be considered when obtaining the next logic level. The built-in FIFO is 64 bits, with a maximum index of 63, so the window size can only be set to 63.
16	FILTEN	Filter enable: 0: Filter disable 1: Filter enable
15:0	Reserved	Reserved, the reset value must be maintained.

11.6.19 Channel 3 Filter Register（TIMx_C3FILT）

Offset address: 0x6C

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		THRESH[5:0]						Reserved	WSIZE[5:0]						FILTEN
				rw							rw				rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															

Bit field	Name	Description
31:30	Reserved	Reserved, the reset value must be maintained.
29:24	THRESH[5:0]	Threshold number of valid logic level samples, maximum 63: the threshold for valid logic levels. Within the sampling window, if the quantity of logic highs is greater than or equal to the threshold, the next logic level will be a logic high. The same rule applies to logic lows. If the quantity of high and low logic levels within the window is less than the threshold, the filter output remains unchanged. The threshold should be set to a value greater than or equal to half of the Window value Recommended threshold range: **Minimum value:** one additional pre-scaler clock period beyond the maximum glitch size limit (pre-scaler clock period), and must be greater than half of the window size. For example, if the glitch size is 3.2 * (the pre-scaler clock period), then the threshold should be ┌3.2┐ = 4 + 1 =5 **Maximum value:** the lower limit of the minimum size of the valid signal (in pre-scaler clock periods), and must be less than the window size. For example, if the minimum signal size is 3.2 * (the pre-scaler clock period), then the threshold should be the lower limit (3.2) = 3.
23	Reserved	Reserved, the reset value must be maintained.
22:17	WSIZE[5:0]	Window size value for logic level checking, maximum 63: The window size determines how many sample values will be considered when obtaining the next logic level. The built-in FIFO is 64 bits, with a maximum index of 63, so the window size can only be set to 63.
16	FILTEN	Filter enable: 0: Filter disable 1: Filter enable
15:0	Reserved	Reserved, the reset value must be maintained.

11.6.20 Channel 4 Filter Register（TIMx_C4FILT）

Offset address: 0x70

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		THRESH[5:0]						Reserved	WSIZE[5:0]						FILTEN
		rw							rw						rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															

Bit field	Name	Description
31:30	Reserved	Reserved, the reset value must be maintained.
29:24	THRESH[5:0]	Threshold number of valid logic level samples, maximum 63: the threshold for valid logic

		levels. Within the sampling window, if the quantity of logic highs is greater than or equal to the threshold, the next logic level will be a logic high. The same rule applies to logic lows. If the quantity of high and low logic levels within the window is less than the threshold, the filter output remains unchanged. The threshold should be set to a value greater than or equal to half of the Window value Recommended threshold range: **Minimum value:** one additional pre-scaler clock period beyond the maximum glitch size limit (pre-scaler clock period), and must be greater than half of the window size. For example, if the glitch size is 3.2 * (the pre-scaler clock period), then the threshold should be ┌3.2┐ = 4 + 1 =5 **Maximum value:** the lower limit of the minimum size of the valid signal (in pre-scaler clock periods), and must be less than the window size. For example, if the minimum signal size is 3.2 * (the pre-scaler clock period), then the threshold should be the lower limit (3.2) = 3.
23	Reserved	Reserved, the reset value must be maintained.
22:17	WSIZE[5:0]	Window size value for logic level checking, maximum 63: The window size determines how many sample values will be considered when obtaining the next logic level. The built-in FIFO is 64 bits, with a maximum index of 63, so the window size can only be set to 63.
16	FILTEN	Filter enable: 0: Filter disable 1: Filter enable
15:0	Reserved	Reserved, the reset value must be maintained.

11.6.21 Input Channel Filter Output Register（TIMx_FILTO）

Offset address: 0x74

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												C4FILTO	C3FILTO	C2FILTO	C1FILTO
												r	r	r	r

Bit field	**Name**	**Description**
31:4	Reserved	Reserved, the reset value must be maintained.

3	C4FILTO	Channel 4 filter output level status
		0: Output low level;
		1: Output high level;

2	C3FILTO	Channel 3 filter output level status
		0: Output low level;
		1: Output high level;

1	C2FILTO	Channel 2 filter output level status
		0: Output low level;
		1: Output high level;

0	C1FILTO	Channel 1 filter output level status
		0: Output low level;
		1: Output high level;

11.6.22 Input Selection（TIMx_INSEL）

Note:For detailed selection of signal sources, please refer to Chapter 11.4.

Offset address: 0x78
Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				CLRS[3:0]				ITRS[3:0]				ETRS[3:0]			
				rw				rw				rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TI4S[3:0]				TI3S[3:0]				TI2S[3:0]				TI1S[3:0]			
rw				rw				rw				rw			

Bit field	**Name**	**Description**
31:28	Reserved	Reserved, the reset value must be maintained.
27:24	CLRS[3:0]	Selects tim_ocref_clr input signal 0000: tim_ocref_clr0 0001: tim_ocref_clr1 … 1111: tim_ocref_clr15 *Note: Once the LOCK level (LCKCFG bit in the TIMx_BKDT register) is set to '1', this bit cannot be modified.*
23:20	ITRS[3:0]	Selects tim_itr input signal 0000: tim_itr0 0001: tim_itr1 … 1111: tim_itr15
19:16	ETRS[3:0]	Selects tim_etr input signal 0000: tim_etr0 0001: tim_etr1 … 1111: tim_etr15 *Note: Once the LOCK level (LCKCFG bit in the TIMx_BKDT register) is set to '1', this bit cannot be modified.*
15:12	TI4S[3:0]	Selects tim_ti4[15:0] input signal 0000: tim_ti4_in0

		0001: tim_ti4_in1 … 1111: tim_ti4_in15
11:8	TI3S[3:0]	Selects tim_ti3[15:0] input signal 0000: tim_ti3_in0 0001: tim_ti3_in1 … 1111: tim_ti3_in15
7:4	TI2S[3:0]	Selects tim_ti2[15:0] input signal 0000: tim_ti2_in0 0001: tim_ti2_in1 … 1111: tim_ti2_in15
3:0	TI1S[3:0]	Selects tim_ti1[15:0] input signal 0000: tim_ti1_in0 0001: tim_ti1_in1 … 1111: tim_ti1_in15

11.6.23 Sliding Filter Prescaler Register（TIMx_SLIDFPSC）

Offset address: 0x88

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SLIDFPSC [15:0]															
rw															

Bit field	Name	Description
31:16	Reserved	Reserved, the reset value must be maintained.
15:0	SLIDFPSC [15:0]	Prescaler value for the sliding filter sampling clock: For this filter, it supports a 65535 division (16 bits). The clock prescaler scales the system clock to the sampling clock. The sampling clock determines the distance between two sampling points. Only the values of the sampling points are considered for logic level calculations. These bits can be configured to determine the sampling clock division for the sliding filter of channels 1/2/3/4, used in conjunction with TIMx_CH1FILT, TIMx_CH2FILT, TIMx_CH3FILT, TIMx_CH4FILT.

11.6.24 DMAControl Register（TIMx_DCTRL）

Offset address：0x94

Reset value：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		DBADDR[5:0]						Reserved		DBLEN[5:0]					
		rw								rw					

Bit field	Name	Description
31:14	Reserved	Reserved, the reset value must be maintained.
13:8	DBADDR[5:0]	DMA base address This Bit Field defines the first address where the DMA accesses the TIMx_DADDR register. When access is done through the TIMx_DADDR first time, this bit-field specifies the address you just access. And then the second access to the TIMx_DADDR, you will access the address of “DMA Base Address + 4” 00000: TIMx_CTRL1, 00001: TIMx_CTRL2, 00010: TIMx_SMCTRL, ... 10001: TIMx_BKDT, 10010: TIMx_DCTRL,
7:6	Reserved	Reserved, the reset value must be maintained.
5:0	DBLEN[5:0]	DMA burst length This Bit Field defines the number DMA will accesses (write/read) TIMx_DADDR register. 000000: 1 time transfer 000001: 2 times transfer 000010: 3 times transfer … 010001: 18 times transfer ...... 100010: 35 times transfer Example: We consider the following transfer: DBLEN=7, DBADDR=TIMx_CTRL1 If DBLEN=7 and DBADDR=TIMx_CTRL1 represent the address of the data to be transferred, then the address of the transfer is given by the following equation ((Address of TIMx_CTRL1) + DBADDR + (DMA index)), where DMA index = DBLEN. Adding 7 to ((Address of TIMx_CTRL1) + DBADDR) gives the address where data will be written to or read from, resulting in data transfer occurring in 7 registers starting from the address ((Address of TIMx_CTRL1) + DBADDR). If the data is set as half-word (16 bits), then the data will be transferred to all 7 registers If the data is set as bytes, the data will still be transferred to all 7 registers: the first register contains the first MSB byte, the second register contains the first LSB byte, and so on.

Bit field	Name	Description
		Therefore, for the timer, the user must specify the data width to be transferred by DMA.

11.6.25 DMA Transfer Buffer Register（TIMx_DADDR）

Offset address: 0x98

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BURST[31:0]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BURST[31:0]															
rw															

Bit field	Name	Description
31:0	BURST[31:0]	DMA accessing buffer When a read or write operation is assigned to this register, the register located at the address range (DMA base address + DMA burst length × 4) will be accessed. DMA base address = The address of TIM_CTRL1 + TIMx_DCTRL. DBADDR * 4; DMA burst len = TIMx_DCTRL.DBLEN + 1. Example: If TIMx_DCTRL.DBLEN = 0x3(4 transfers), TIMx_DCTRL.DBADDR = 0xD (TIMx_CCDAT1), DMA data length = half word, DMA memory address = buffer address in SRAM, DMA peripheral address = TIMx_DADDR address. When an event occurs, TIMx will send requests to the DMA, and transfer data 4 times. For the first time, DMA access to the TIMx_ DADDR register will be mapped to access TIMx_CCDAT1 register; For the second time, DMA access to the TIMx_ DADDR register will be mapped to access TIMx_CCDAT2 register; … … For the fourth time, DMA access to the TIMx_ DADDR register will be mapped to access TIMx_CCDAT4 register;

12 General-Purpose Timers（GTIM8/ GTIM9/ GTIM10）

12.1 GTIMx（x=8-10）Introduction

The general-purpose timers (GTIM8/ GTIM9/ GTIM10) is mainly used in the following scenarios: counting the input signal, measuring the pulse width of the input signal and generating the output waveform, etc.

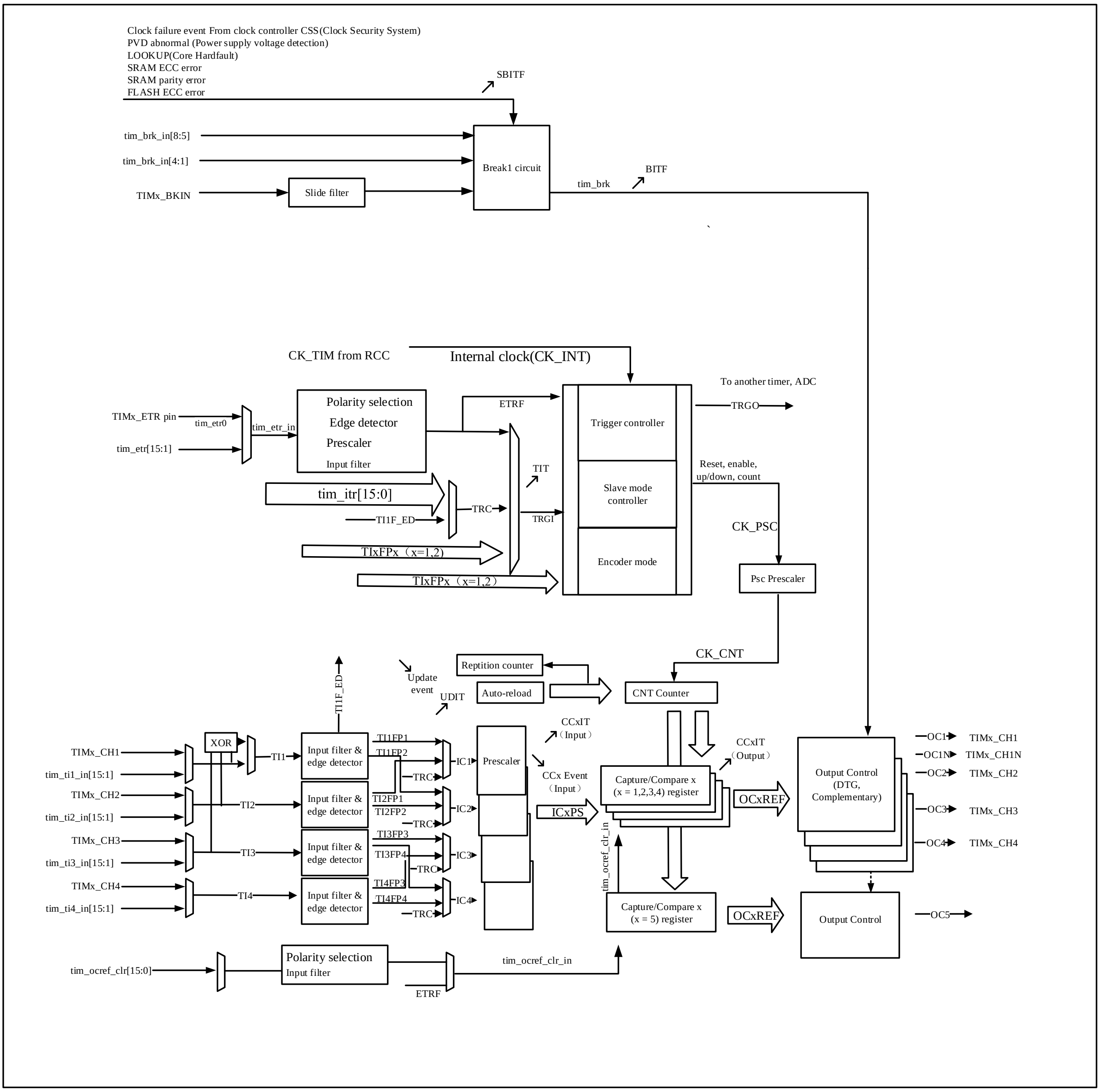
General timers feature complementary output, dead-time insertion, and break functions. They are suitable for motor control applications.

12.2 Main Features Of GTIMx（x=8-10）

- 16-bit auto-reload counters. (It can realize up-counting, down-counting, up/down counting)
- 16-bit programmable prescaler. (The prescaler factor can be configured with any value between 1 and 65536)
- Programmable repetition counter
- GTIMx up to 5 channels
- 4 capture/compare channels, working modes: PWM output, ouput compare, one-pulse mode output, input capture
- 1 break input signal with digital filtering support, used to place the timer's output signal in a safe user-selectable configuration.
- The events that generate the interrupt/DMA are as follows：
  - Update event
  - Trigger event
  - Input capture
  - Output compare
  - Break signal input
- Programmable dead-time for complementary outputs
  - For GTIMx, Channel 1 supports this function
- Timer can be controlled by external signal
- Timers can be linked together internally for timer synchronization or chaining
- Incremental (quadrature) encoder interface: used for tracking motion and resolving rotation direction and position
- Hall sensor interface: used to do three-phase motor control
- Trigger input as an external clock or for per-cycle current management

12.3 Block Diagram Of GTIMx (x=8-10)

Figure 12-1 Block Diagram Of GTIMx



12.4 Pin and Internal Signals of GTIMx (x=8-10)

The following table describes the input and output pins and signals of GTIMx.

Table 12-1 GTIMx input/output pins

Pins	Type	Description
TIMx_CH1	Input/Output	Timer multi-purpose channel.

TIMx_CH2 TIMx_CH3 TIMx_CH4		Each channel can be used for capture, compare or PWM generation. TIM_CH1 and TIM_CH2 can also be used as external clock (less than 1/4 of the internal clock frequency), external trigger, and quadrature encoder inputs. TIM_CH1, TIM_CH2, and TIM_CH3 can also be used for interfacing with Hall effect sensors.
TIM_CH1N	Output	Timer complementary output channels with dead-time insertion.
TIMx_ETR	Input	External trigger input. This input can serve as an external trigger or external clock source. If a prescaler is used, the input signal TIMx_ETR can be a signal with a frequency higher than the system clock frequency
TIMx_BKIN	Input/Output	Break input. Can be configured as bidirectional mode.

Table 12-2 GTIMx Internal Input/Output Signal

Internal Signal	Type	Description
tim_ti1_in[15:0] tim_ti2_in[15:0] tim_ti3_in[15:0] tim_ti4_in[15:0]	Input	Timer channels 1/2/3/4 input signals. The tim_ti1_in[15:0] and tim_ti2_in[15:0] inputs can be used for capture or as an external clock (less than 1/4 of the system clock frequency) and for quadrature encoder signals.
tim_etr[15:0]	Input	External trigger channel input signals. These inputs can be used as triggers, external clocks, or for hardware per-cycle pulse width control. If a prescaler is used, the input signal TIMx_ETR can be a signal with a frequency higher than the system clock frequency.
tim_itr[15:0]	Input	Internal trigger input signals. These inputs can be used in mode controllers or as input clocks (less than 1/4 of the system clock frequency).
tim_trgo	Output	Internal trigger signal output. These trigger signals can be used by other timers and/or other peripheral devices.

12.4.1 Source Of signals tim_ti1/ tim_ti2/ tim_ti3/ tim_ti4 for GTIMx

Table 12-3 Source of input signal tim_ti1

tim_ti1 inputs	Signal Source		
	GTIM8	GTIM9	GTIM10

tim_ti1_in0	GTIM8_CH1	GTIM9_CH1	GTIM10_CH1
tim_ti1_in1	COMP1_OUT	COMP1_OUT	COMP1_OUT
tim_ti1_in2	COMP2_OUT	COMP2_OUT	COMP2_OUT
tim_ti1_in3	COMP3_OUT	COMP3_OUT	COMP3_OUT
tim_ti1_in4	COMP4_OUT	COMP4_OUT	COMP4_OUT
tim_ti1_in[15:5]	Reserved		

Table 12-4 Source of input signal tim_ti2

tim_ti2 inputs	Signal Source		
	GTIM8	GTIM9	GTIM10
tim_ti2_in0	GTIM8_CH2	GTIM9_CH2	GTIM10_CH2
tim_ti2_in1	COMP5_OUT	COMP5_OUT	COMP5_OUT
tim_ti2_in2	COMP6_OUT	COMP6_OUT	COMP6_OUT
tim_ti2_in3	COMP7_OUT	COMP7_OUT	COMP7_OUT
tim_ti2_in4	GTIM7_CC3	GTIM7_CC4	GTIM7_CC3
tim_ti2_in[15:5]	Reserved		

Table 12-5 Source of input signal tim_ti3

tim_ti3 inputs	Signal Source		
	GTIM8	GTIM9	GTIM10
tim_ti3_in0	GTIM8_CH3	GTIM9_CH3	GTIM10_CH3
tim_ti3_in1	COMP5_OUT	COMP6_OUT	COMP7_OUT
tim_ti3_in[15:2]	Reserved		

Table 12-6 Source of input signal tim_ti4

tim_ti4 inputs	Signal Source		
	GTIM8	GTIM9	GTIM10
tim_ti4_in0	GTIM8_CH4	GTIM9_CH4	GTIM10_CH4
tim_ti3_in1	COMP5_OUT	COMP6_OUT	COMP7_OUT
tim_ti4_in[15:2]	Reserved		

12.4.2 Source of signal tim_itr for GTIMx

Table 12-7 Source of input signal tim_itr

GTIMx	GTIM8	GTIM9	GTIM10
tim_itr0	atim1_trgo	atim1_trgo	atim1_trgo
tim_itr1	gtim1_trgo	gtim1_trgo	gtim1_trgo
tim_itr2	gtim2_trgo	gtim2_trgo	gtim2_trgo
tim_itr3	gtim3_trgo	gtim3_trgo	gtim3_trgo
tim_itr4	gtim4_trgo	gtim4_trgo	gtim4_trgo
tim_itr5	atim2_trgo	atim2_trgo	atim2_trgo
tim_itr6	Reserved	gtim8_trgo	gtim8_trgo
tim_itr7	gtim9_trgo	Reserved	gtim9_trgo
tim_itr8	gtim10_trgo	gtim10_trgo	Reserved
tim_itr9	atim3_trgo	atim3_trgo	Reserved
tim_itr10	shrtim1_out_sync2	shrtim1_out_sync2	shrtim1_out_sync2
tim_itr11	Reserve	Reserve	Reserve
tim_itr12	gtim5_trgo	gtim5_trgo	gtim5_trgo
tim_itr13	gtim6_trgo	gtim6_trgo	gtim6_trgo

tim_itr14	gtim7_trgo	gtim7_trgo	gtim7_trgo
tim_itr15	Reserve	Reserve	Reserve

12.4.3 Source of signal tim_etr for GTIMx

Table 12-8 Source of input signal tim_etr

GTIMx	GTIM8	GTIM9	GTIM10
tim_etr0	GTIM8_ETR	GTIM9_ETR	GTIM10_ETR
tim_etr1	COMP1_OUT	COMP1_OUT	COMP1_OUT
tim_etr2	COMP2_OUT	COMP2_OUT	COMP2_OUT
tim_etr3	COMP3_OUT	COMP3_OUT	COMP3_OUT
tim_etr4	COMP4_OUT	COMP4_OUT	COMP4_OUT
tim_etr5	COMP5_OUT	COMP5_OUT	COMP5_OUT
tim_etr6	COMP6_OUT	COMP6_OUT	COMP6_OUT
tim_etr7	COMP7_OUT	COMP7_OUT	COMP7_OUT
tim_etr8	GTIM9_ETR	GTIM8_ETR	GTIM8_ETR
tim_etr9	GTIM10_ETR	GTIM10_ETR	GTIM9_ETR
tim_etr10	GTIM1_ETR	GTIM2_ETR	GTIM3_ETR
tim_etr11	ADC4_AWD1	ADC1_AWD1	ADC2_AWD1
tim_etr12	ADC4_AWD2	ADC1_AWD2	ADC2_AWD2
tim_etr13	ADC4_AWD3	ADC1_AWD3	ADC2_AWD3
tim_etr_in[15:14]	Reserved		

12.4.4 Break 1 input signal source of GTIMx

Table 12-9 Break 1 input signal source of GTIMx

Break input	GTIM8	GTIM9	GTIM10
TIM_BKIN	GTIM8_BKIN pin	GTIM9_BKIN pin	GTIM10_BKIN pin
tim_brk_in1	COMP1_OUT	COMP1_OUT	COMP1_OUT
tim_brk_in2	COMP2_OUT	COMP2_OUT	COMP2_OUT
tim_brk_in3	COMP3_OUT	COMP3_OUT	COMP3_OUT
tim_brk_in4	COMP4_OUT	COMP4_OUT	COMP4_OUT
tim_brk_in5	COMP5_OUT	COMP5_OUT	COMP5_OUT
tim_brk_in6	COMP6_OUT	COMP6_OUT	COMP6_OUT
tim_brk_in7	COMP7_OUT	COMP7_OUT	COMP7_OUT
tim_brk_ in8	Reserved		

12.4.5 tim_ocref_clr input signal source of GTIMx

Table 12-10 tim_ocref_clr input signal source of GTIMx

OCREF clear signal	GTIMx OCREF clear signal assignment		
	GTIM8	GTIM9	GTIM10
tim_ocref_clr0	COMP1_OUT	COMP1_OUT	COMP1_OUT
tim_ocref_clr1	COMP2_OUT	COMP2_OUT	COMP2_OUT
tim_ocref_clr2	COMP3_OUT	COMP3_OUT	COMP3_OUT
tim_ocref_clr3	COMP4_OUT	COMP4_OUT	COMP4_OUT
tim_ocref_clr4	COMP5_OUT	COMP5_OUT	COMP5_OUT
tim_ocref_clr5	COMP6_OUT	COMP6_OUT	COMP6_OUT

tim_ocref_clr6	COMP7_OUT	COMP7_OUT	COMP7_OUT
tim_ocref_in[15:7]	Reserved		

12.5 GTIMx(x=8-10) Description

12.5.1 Time-base Unit

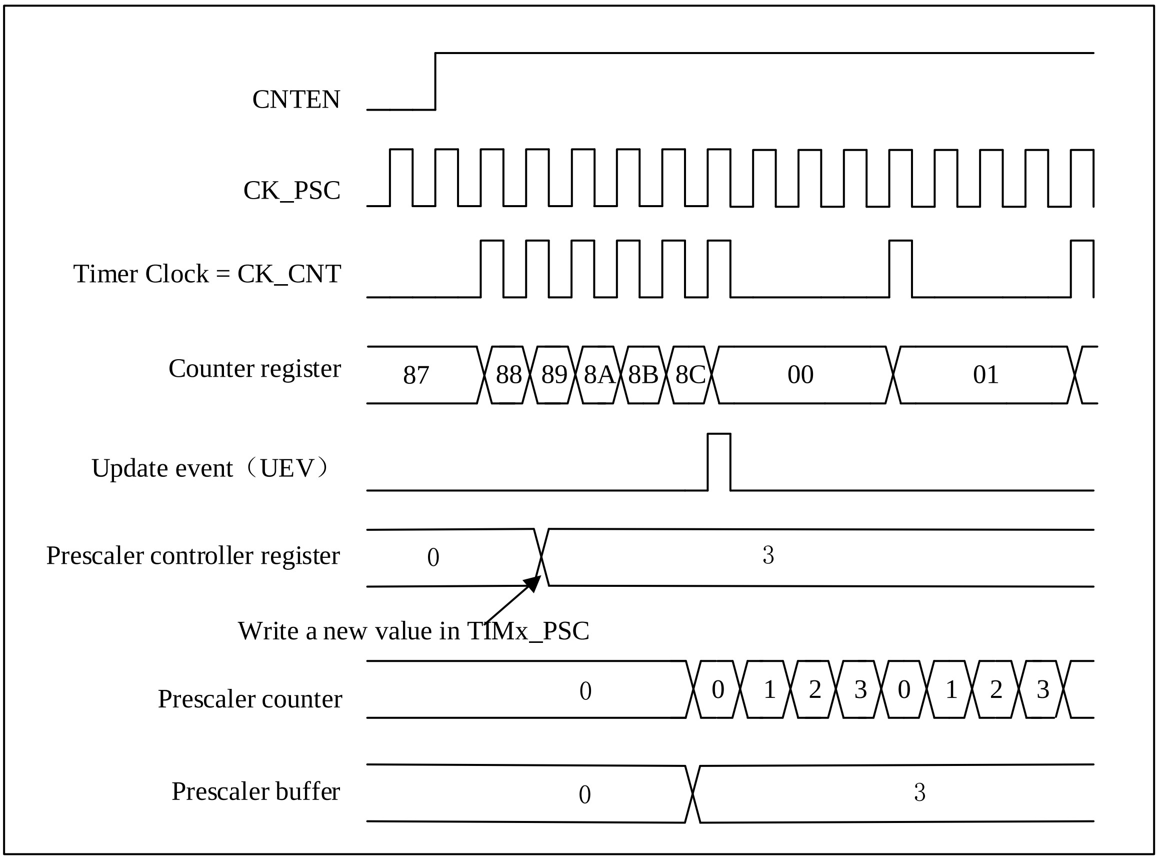
The time-base unit mainly includes: prescaler, counter, auto-reload register and repetition counter. When the time base unit is working, the software can read and write the corresponding registers (TIMx_PSC, TIMx_CNT, TIMx_AR and TIMx_REPCNT) at any time.

Depending on the setting of the auto-reload preload enable bit (TIMx_CTRL1.ARPEN), the value of the preload register is transferred to the shadow register immediately or at each update event UEV. An update event is generated when the counter reaches the overflow/underflow condition and it can be generated by software set TIMx_EVTGEN.UDGN when TIMx_CTRL1.UPDIS=0. The counter CK_CNT is valid only when the TIMx_CTRL1.CNTEN bit is set. The counter starts counting one clock cycle after the TIMx_CTRL1.CNTEN bit is set.

12.5.1.1 Prescaler Description

The TIMx_PSC register consists of a 16-bit counter that can be used to divide the counter clock frequency by any factor between 1 and 65536. It can be changed on the fly as it is buffered. The prescaler value is only taken into account at the next update event.

Figure 12-2 Counter Timing Diagram With Prescaler Division Change From 1 To 4



12.5.2 Counter Mode

12.5.2.1 Up-counting mode

In up-counting mode, the counter will count from 0 to the value of the register TIMx_AR, then it resets to 0. And a counter overflow event is generated.

If the TIMx_CTRL1.UPRS bit (select update request) and the TIMx_EVTGEN.UDGN bit are set, an update event (UEV) will generate And TIMx_STS.UDITF will not be set by hardware, therefore, no update interrupts or update DMA requests are generated. This setting is used in scenarios where you want to clear the counter but do not want to generate an update interrupt.

Depending on the update request source is configured in the TIMx_CTRL1.UPRS. When an update event occurs, TIMx_STS.UDITF is set, all registers are updated:

- The repetition counter is reloaded with the content of TIMx_REPCNT.
- Update auto-reload shadow registers with preload value(TIMx_AR), when TIMx_CTRL1.ARPEN = 1
- The prescaler shadow register is reloaded with the preload value(TIMx_PSC)

To avoid updating the shadow registers when new values are written to the preload registers, you can disable the update by setting TIMx_CTRL1.UPDIS=1

When an update event occurs, the counter will still be cleared and the prescaler counter will also be set to 0 (but the prescaler value will remain unchanged)。

The figure below shows some examples of the counter behavior and the update flags for different division factors in the up-counting mode

Figure 12-3 Timing Diagram Of Up-Counting. The Internal Clock Divider Factor = 2/N

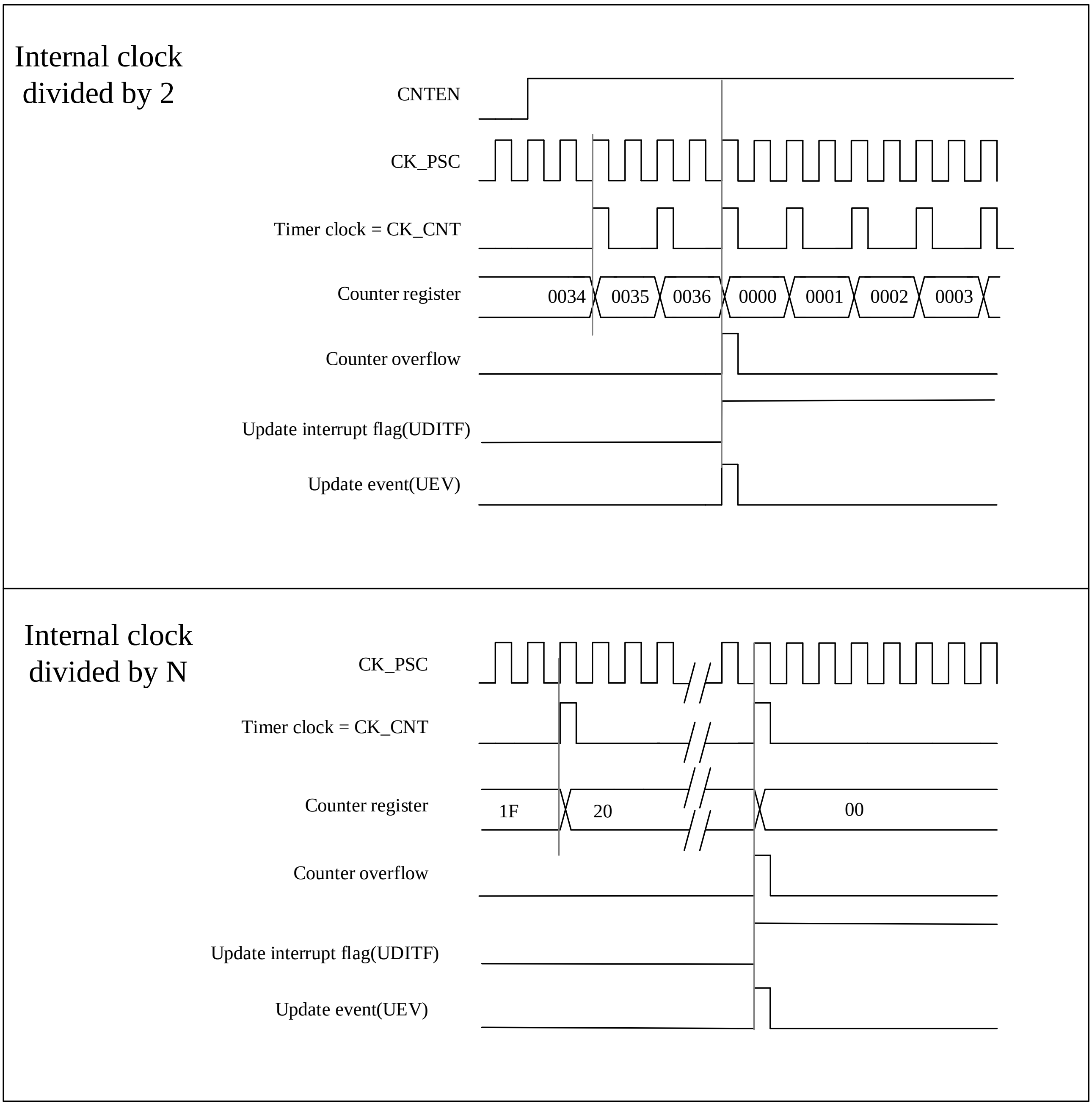
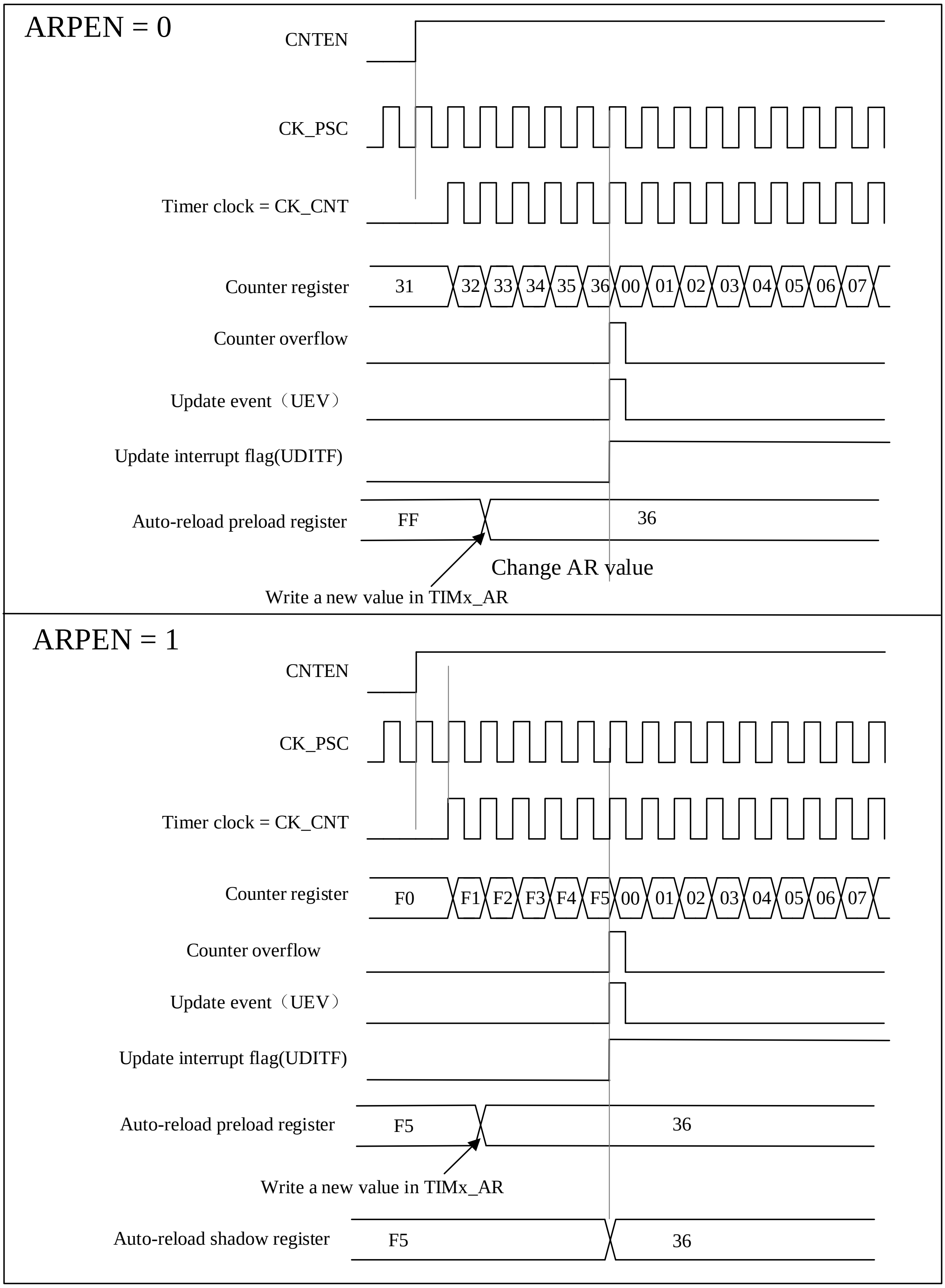


Figure 12-4 Timing Diagram Of The Up-Counting With Update Event When ARPEN=0/1



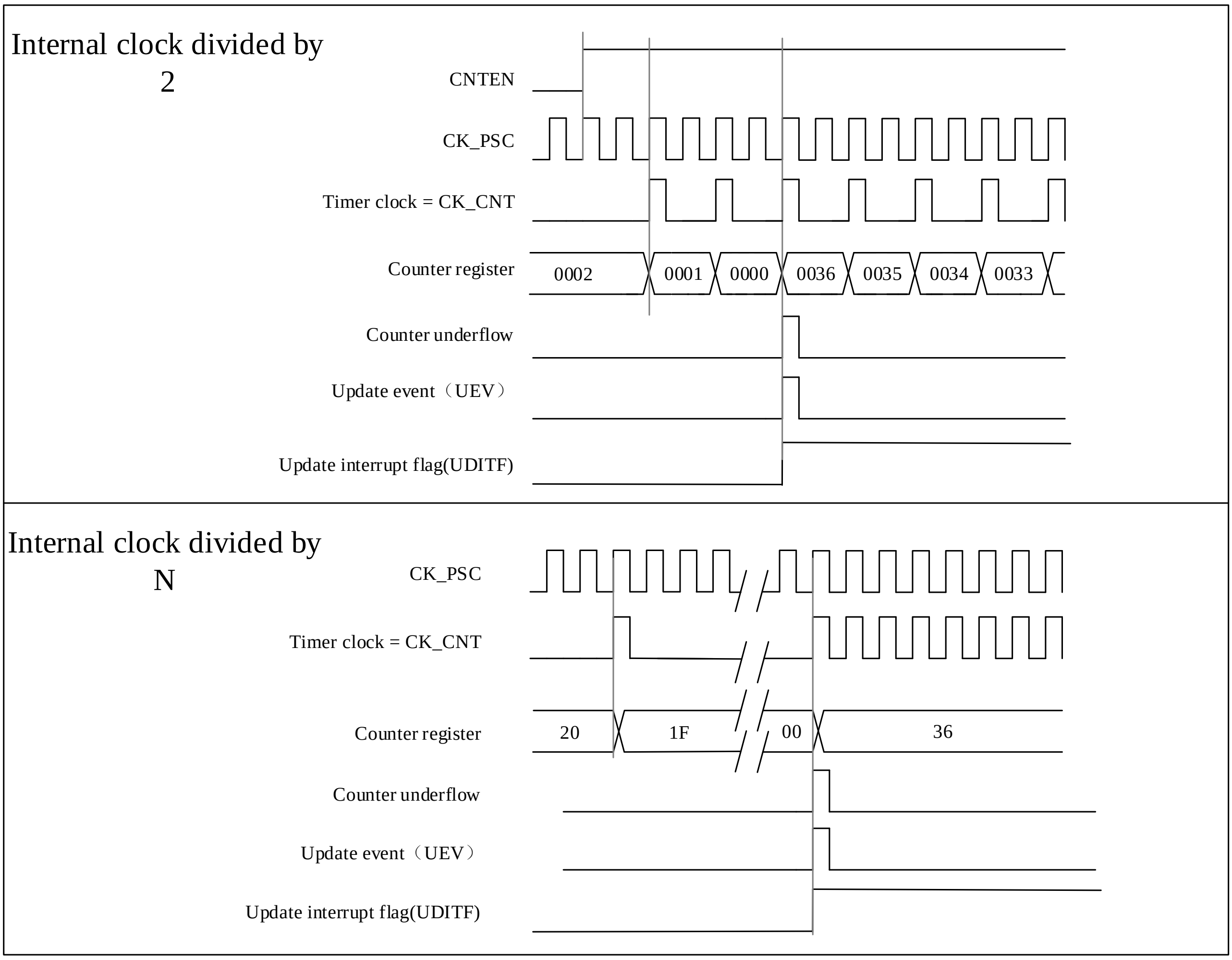
12.5.2.2 Down-counting mode

In down-counting mode, the counter will decrement from the value of the register TIMx_AR to 0, then restart from the auto-reload value and generate a counter underflow event.

The process of configuring update events and updating registers in down-counting mode is the same as in up-counting mode, see Section10.5.2.1.

The figure below shows some examples of the counter behavior and the update flags for different division factors in the down-counting mode.

Figure 12-5 Timing Diagram Of The Down-Counting With Internal Clock Divided Factor = 2/N



12.5.2.3 Center-aligned mode

12.5.2.3.1 Center-aligned symmetric mode

In center-aligned mode, the counter increments from 0 to the value (TIMx_AR) – 1, a counter overflow event is generated. It then counts down from the auto-reload value (TIMx_AR) to 1 and generates a counter underflow event. Then the counter resets to 0 and starts counting up again。

In this mode, the TIMx_CTRL1.DIR direction bits have no effect and the count direction is updated and specified by hardware. Center-aligned mode is valid when the TIMx_CTRL1. CAMSEL bit is not equal to "00"。

The update events can be generated each time the counter overflows and each time the counter underflows.

Alternatively, an update event can also be generated by setting the TIMx_EVTGEN. UDGN bit (either by software or using a slave mode controller). In this case, the counter restarts from 0, as does the prescaler's counter.

Note: if the update source is a counter overflow, auto-reload update before reloading the counter。

Figure 12-6 Timing Diagram Of The Center-Aligned, Internal Clock Divided Factor =2/N

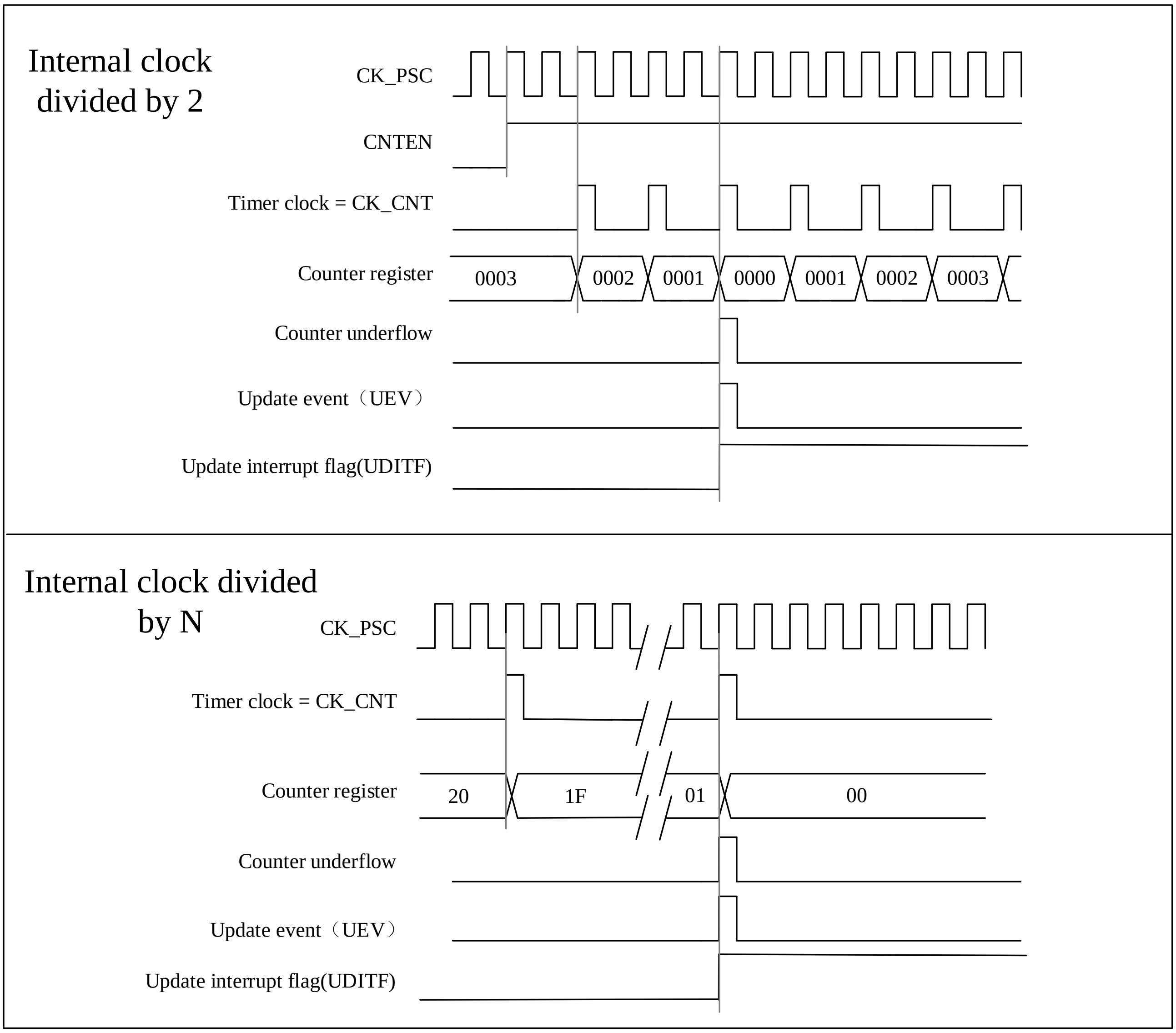
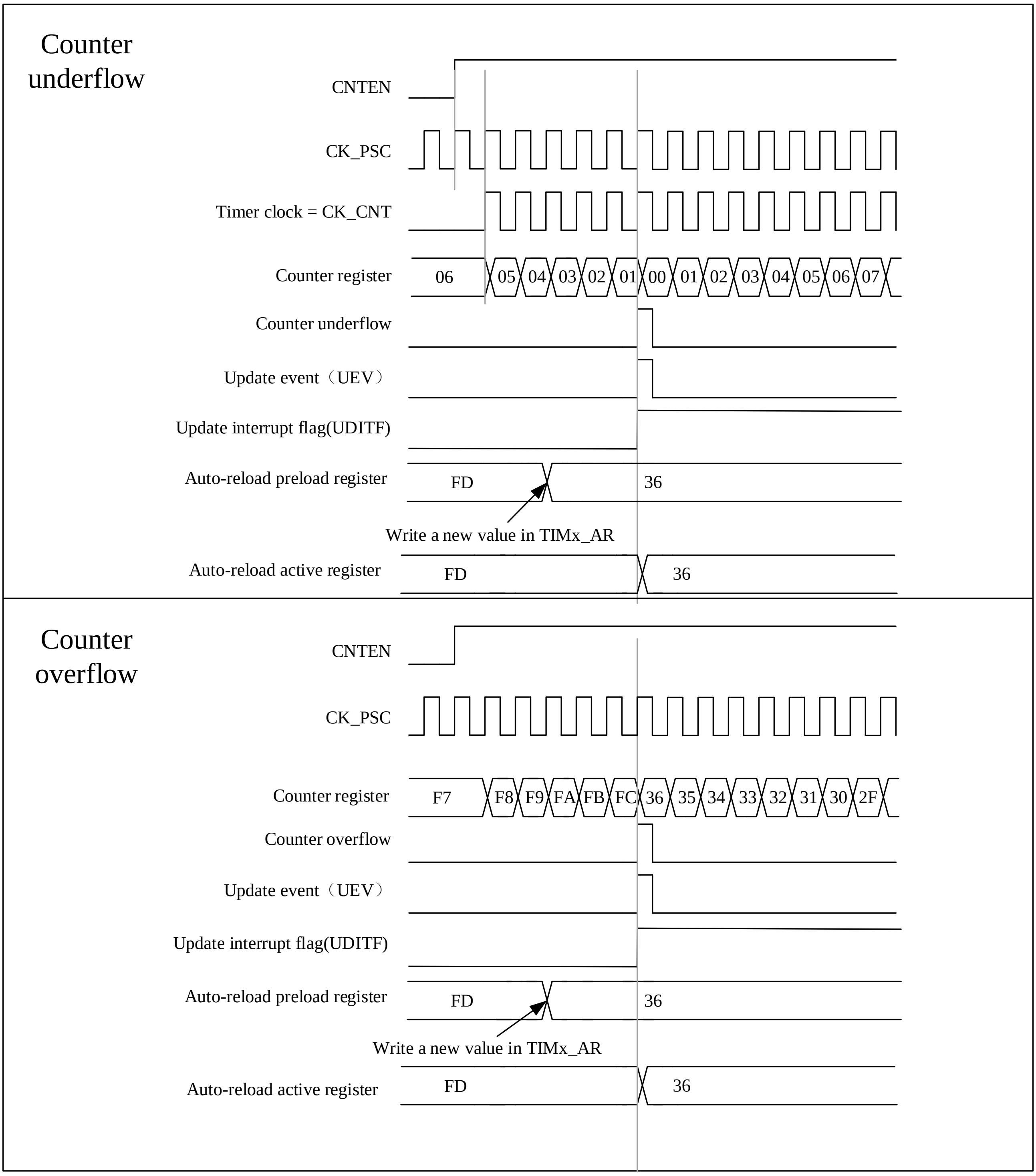


Figure 12-7 A Center-Aligned Sequence Diagram That Includes Counter Overflows And Underflows (ARPEN = 1)



12.5.2.3.2 Central Aligned Asymmetric Mode

In the central aligned asymmetric mode (TIMx_CTRL1. ASYMMETRIC is 1, TIMx_CTRL1.CAMSEL[1:0] is non-zero), the counter counts from 0 to the auto-reload value (TIMx_AR) - 1, generates a counter overflow event, then counts from the auto-reload value to 1, and generates a counter underflow event, then restarts counting from 0

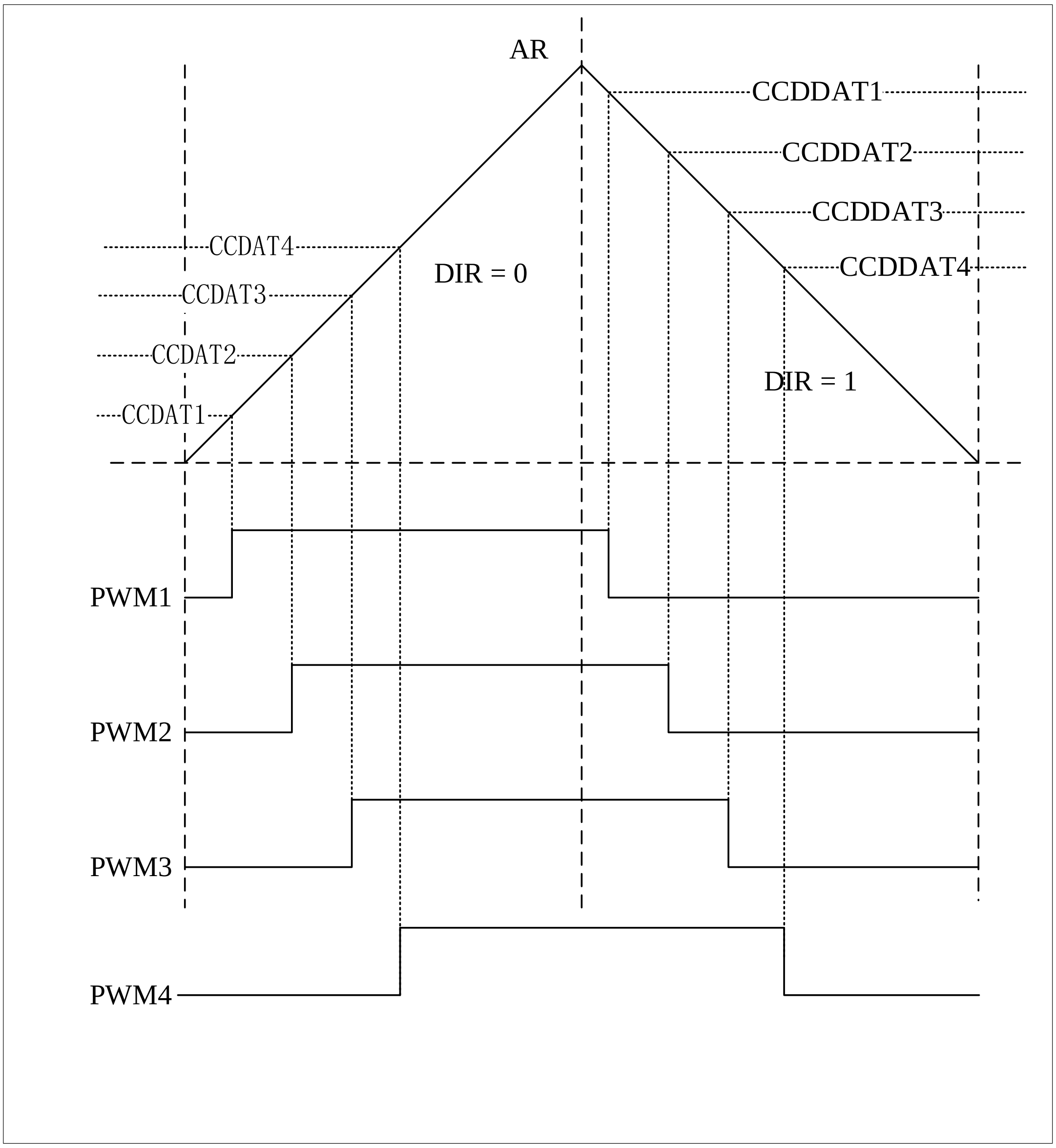
The value of TIMx_CTRL1.DIR cannot be written in this mode; it is updated by hardware and specifies the current counting direction.

When the channel is not 1, 2, 3, or 4, the comparison value is CCDATx. When the dead-time generator is enabled, please note that when DIR=0, the dead-time insertion point is when the counter value is equal to CCDATx (x=1,2,3,4), and when DIR=1, the dead-time insertion point is when the counter value is equal to CCDDATx (x=1,2,3,4).

An update event occurs each time there is a counter overflow or underflow. Alternatively, an update event can also be generated by setting the TIMx_EVTGEN.UDGN bit (either through software or by using the mode controller). In this case, the counter restarts counting from 0, and the prescaler counter also restarts counting from 0.

Note: If an update is generated due to a counter overflow, the auto-reload will be updated before the counter is reloaded.

Figure 12-8 Output WaveformCorrespo Asymmetric Mode



12.5.3 Repetition Counter

The basic unit of Section 12.5.1 describes the conditions for generating Update Events (UEV). Update Events (UEV) are actually generated only when the repetition counter reaches zero, which is very useful for generating PWM signals

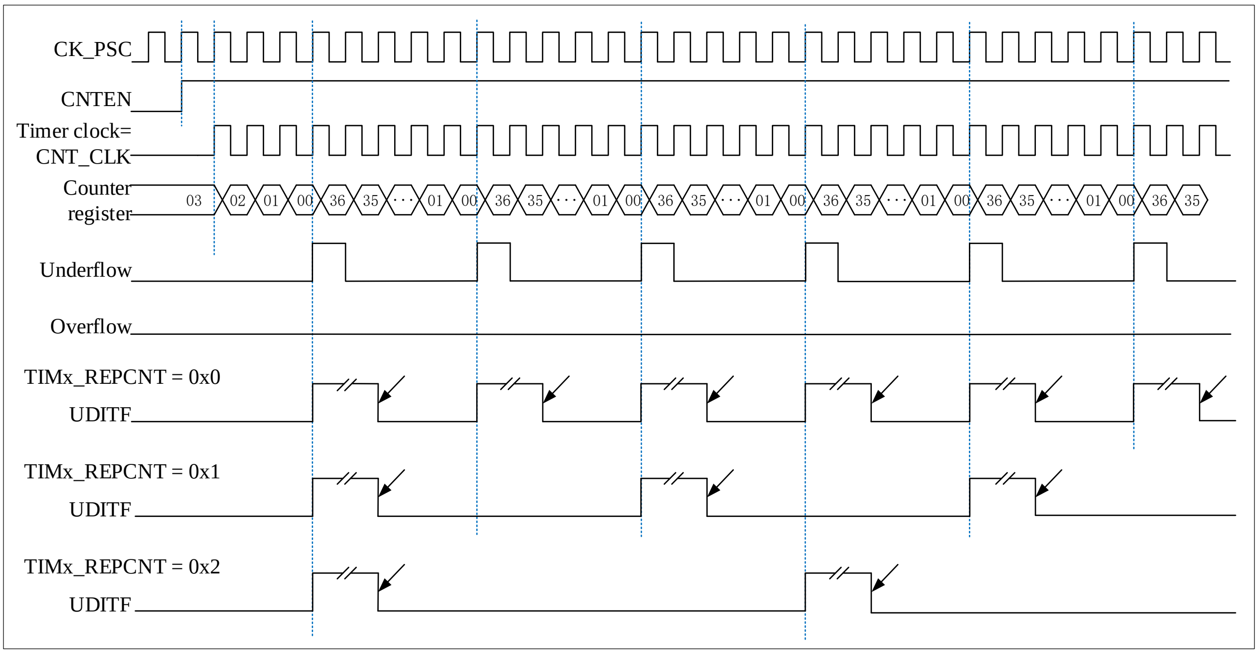
This means that every time the N+1 counter overflows or underflows, data is transferred from the preload register to the shadow register, where N is the value in TIMx_REPCNT.

Decrementing repetition counter:

- In up-count mode, overflow occurs each time the counter reaches its maximum value.
- In down-count mode, underflow occurs each time the counter reaches its minimum value.
- In center-aligned mode, the repetition rate is determined by the value of the TIMx_REPCNT register each time the counter overflows or underflow.

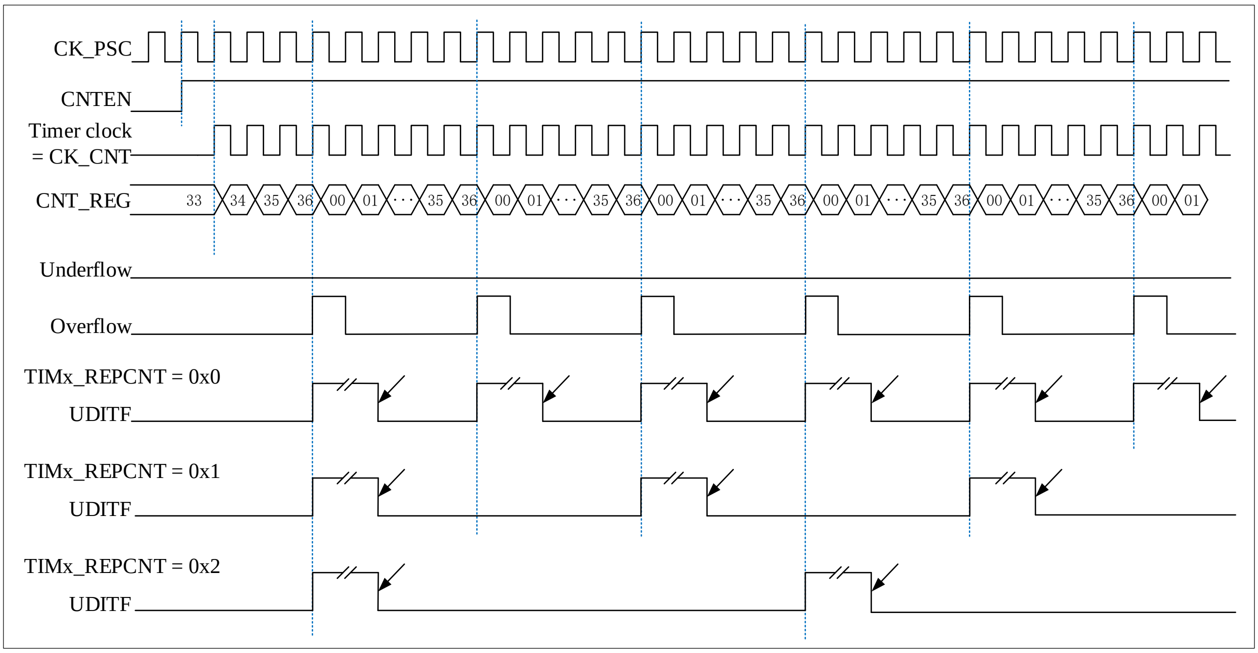
The repetition counter has an automatic reload function. Regardless of the value of the repetition counter, the update event (triggered by setting TIMx_EVTGEN.UDGN in the mode controller or generated by hardware) occurs immediately.

Figure 12-9 Timing diagram of the repetition counter in down-count mode



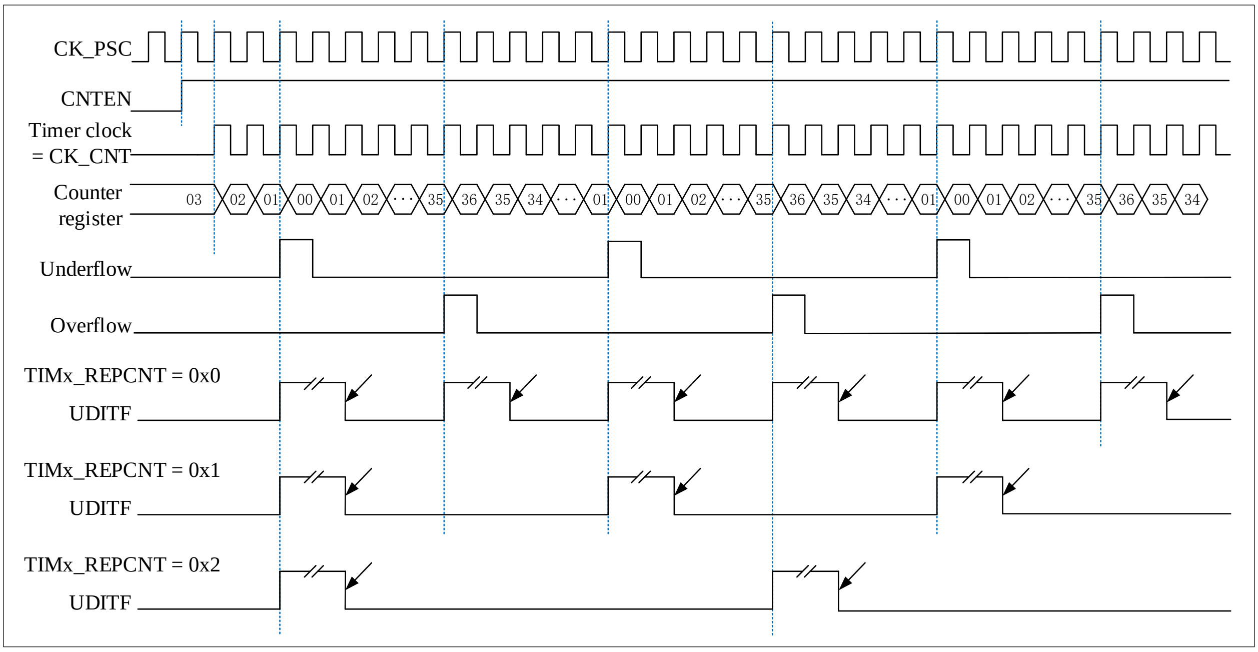
↙ *software clear*

Figure 12-10 Timing diagram of the repetition counter in up-count mode



software clear

Figure 12-11 Timing diagram of the repetition counter in center-aligned mode



software clear

12.5.4 Clock Selection

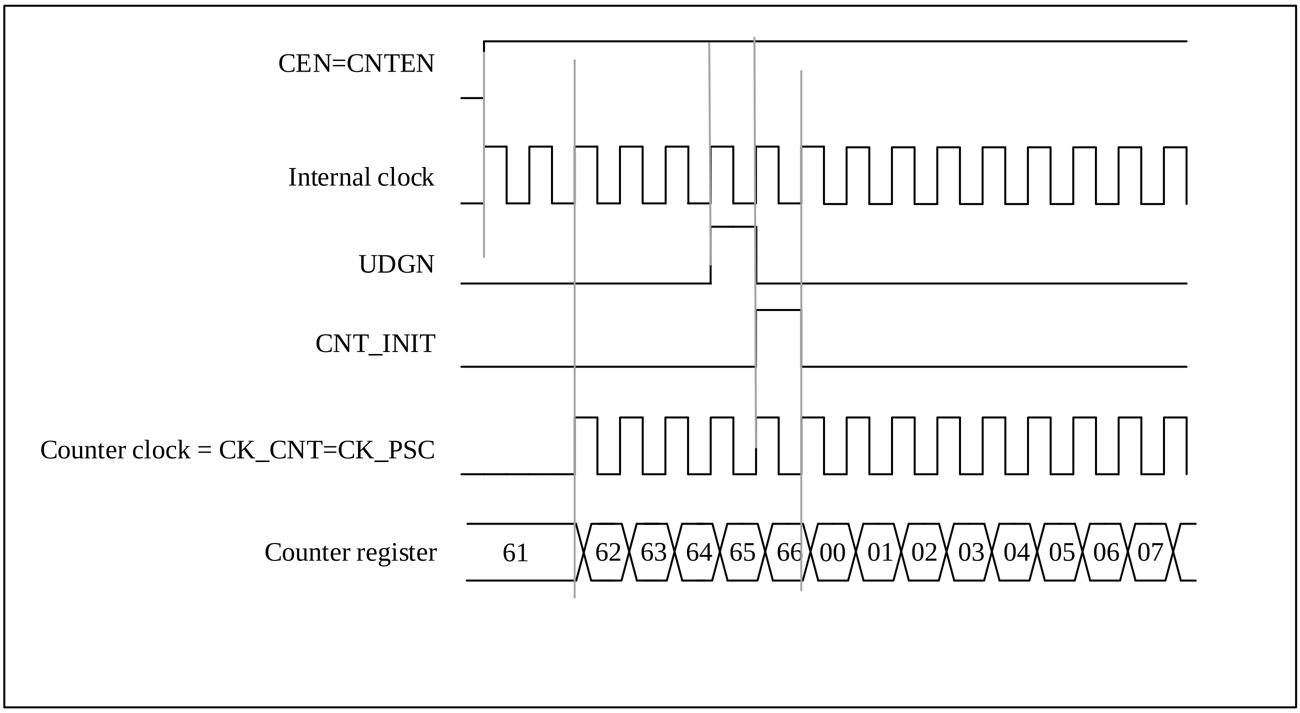
- The internal clock of timers: CK_INT

- Two kinds of external clcock mode:
  - external input pin.
  - external trigger input ETR.
- Internal trigger input(ITRx): one timer is used as a prescaler for another timer.

12.5.4.1 Internal Clock Source(CK_INT)

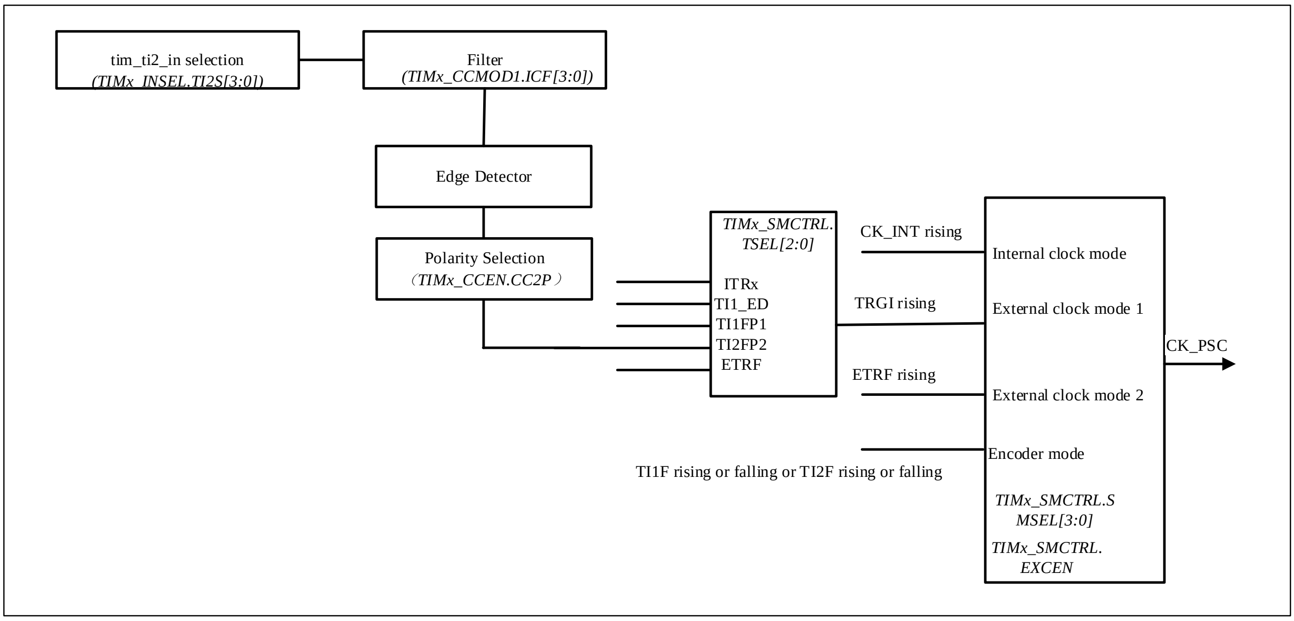
When the TIMx_SMCTRL.SMSEL is equal to “0000”, the slave mode controller is disabled. The three control bits (TIMx_CTRL1.CNTEN、TIMx_CTRL1. DIR、TIMx_EVTGEN. UDGN) can only be changed by software (except TIMx_EVTGEN. UDGN, which remains cleared automatically). It is provided that the TIMx_CTRL1.CNTEN bit is written as' 1 ' by software, the clock source of the prescaler is provided by the internal clock CK_INT.

Figure 12-12 Control Circuit In Normal Mode With Internal Clock Divided By 1



12.5.4.2 External clock source mode 1

Figure 12-13 TI2 External Clock Connection Example



This mode is selected by configuring TIMx_SMCTRL.SMSEL=‘0111’. The counter can be configured to count on the rising or falling edge of the clock at the selected input.

For example, to configure up-counting mode to count on the rising edge of the clock at the T12 input, the configuration steps are as follows：

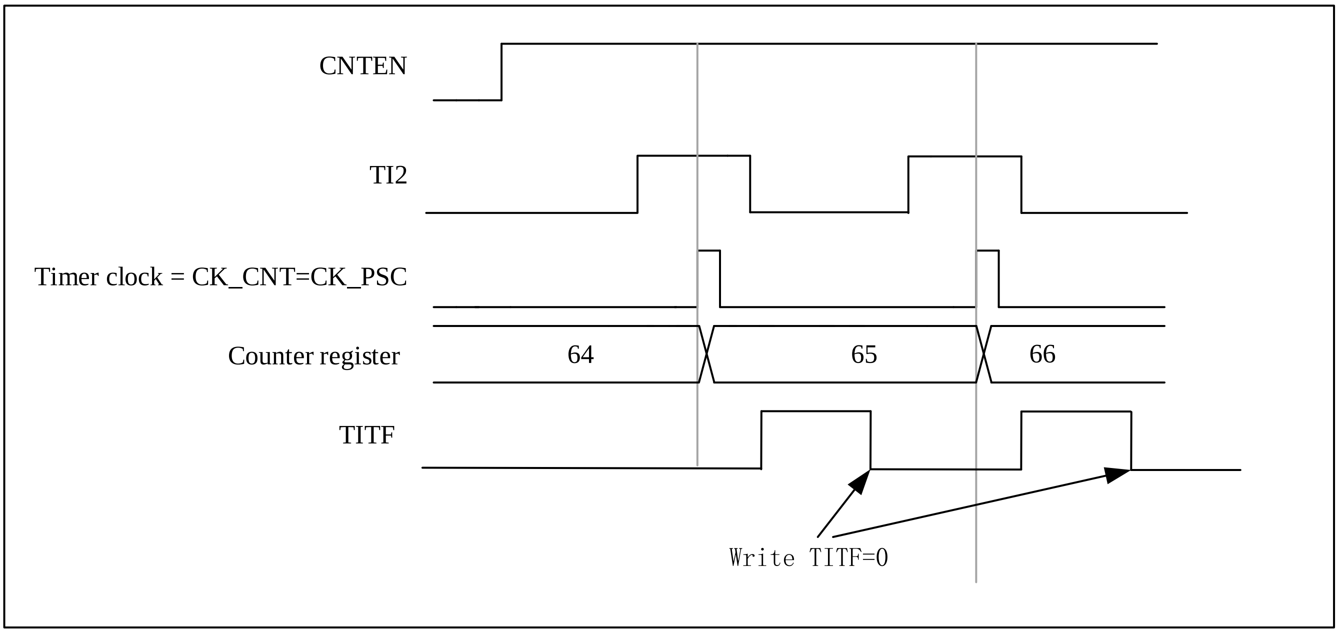
- Configure TIMx_CCMOD1.CC2SEL equal to ‘01’, CC2 channel is configured as input, IC2 is mapped to TI2
- Configure TIMx_CCEN.CC2P equal to ‘0’, select clock rising edge polarity
- To select input filter bandwidth by configuring TIMx_CCMOD1.IC2F[3:0] (if filter is not needed, keep IC2F bit at ‘0000’)
- Configure TIMx_SMCTRL.SMSEL equal to ‘0111’, select timer external clock mode 1
- Configure TIMx_INSEL.TI2S[3:0] to ‘0000’, select TIM_CH2 as the TI2 input
- Configure TIMx_SMCTRL.TSEL equal to ‘110’, select TI2 as the trigger input source
- Configure TIMx_CTRL1.CNTEN equal to ‘1’ to start the counter

Note: The capture prescaler is not used for triggering, so it does not need to be configured

When the rising edge of the timer clock occurs at TI2=1, the counter counts once and the TIMx_STS .TITF flag is pulled high.

The delay between the rising edge of TI2 and the actual clock of the counter depends on the resynchronization circuit at the input of TI2.

Figure 12-14 Control Circuit In External Clock Mode 1

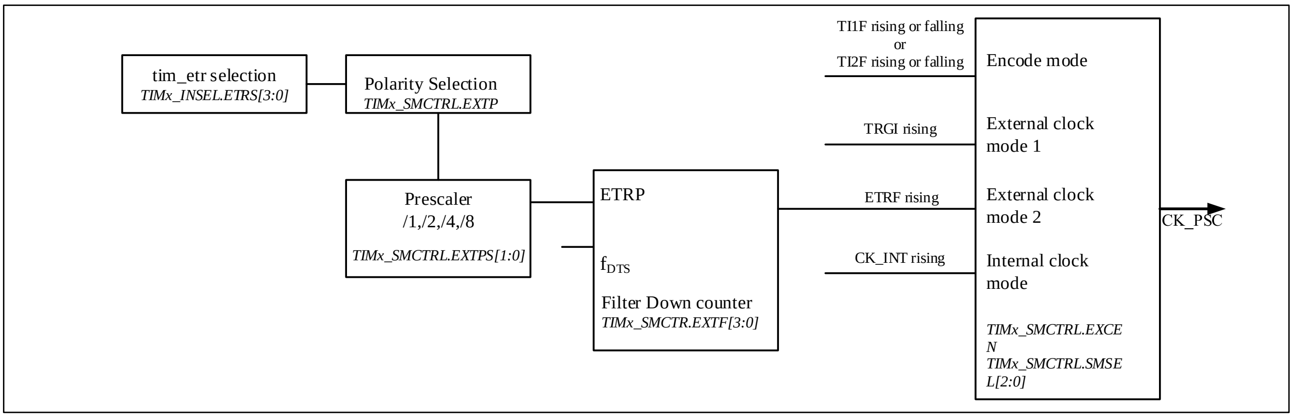


12.5.4.3 External clock source mode 2

This mode is selected by TIMx_SMCTRL .EXCEN equal to 1. The counter can count on every rising or falling edge of the external trigger input ETR.

The following figure is a schematic diagram of the external trigger input module in External clock source mode 2

Figure 12-15 External Trigger Input Block Diagram

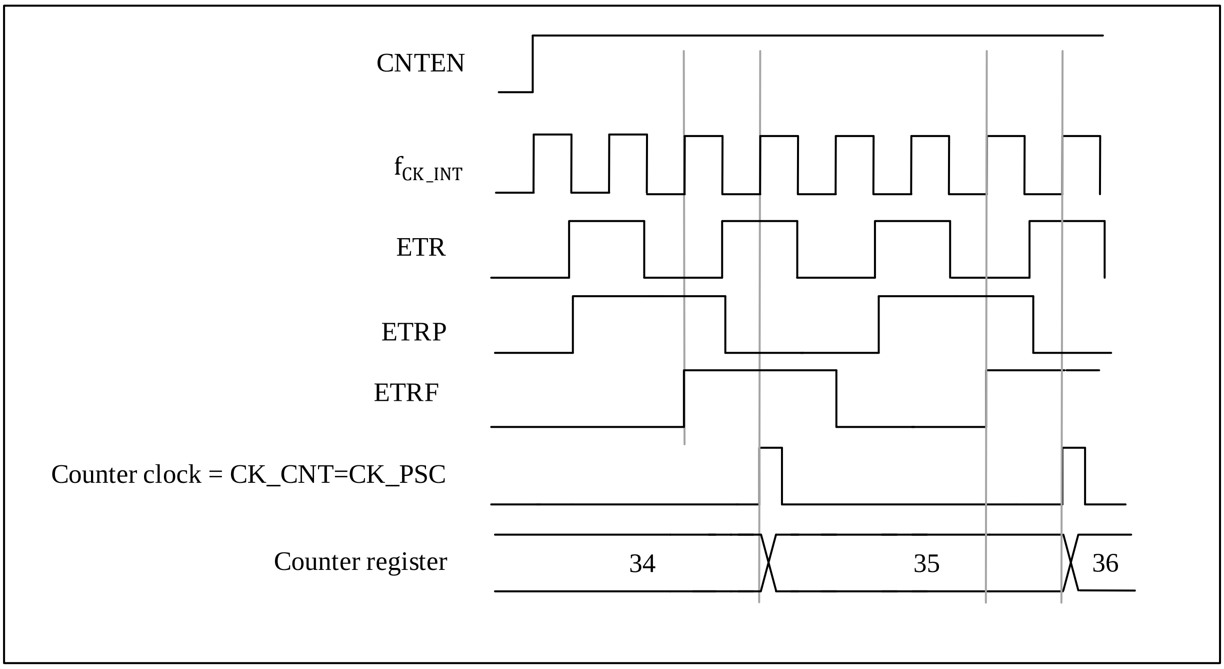


For example, use the following configuration steps to make the up counter count every 2 rising edges on ETR。

- Since no filter is needed in this case, make TIMx_SMCTRL .EXTF[3:0] equal to ‘0000’
- Configure the prescaler by setting TIMx_SMCTRL.EXTPS[1:0] equal to ‘01’
- Select the polarity on ETR pin by setting TIMx_SMCTRL.EXTP equal to ‘0’, The rising edge of ETR is valid
- External clock mode 2 is selected by setting TIMx_SMCTRL .EXCEN equal to ‘1’
- Turn on the counter by setting TIMx_CTRL1. CNTEN equal to ‘1’

The counter counts every 2 rising edges of ETR. The delay between the rising edge of ETR and the actual clock to the counter is due to a resynchronization circuit on the ETRP signal.

Figure 12-16 Control Circuit In External Clock Mode 2

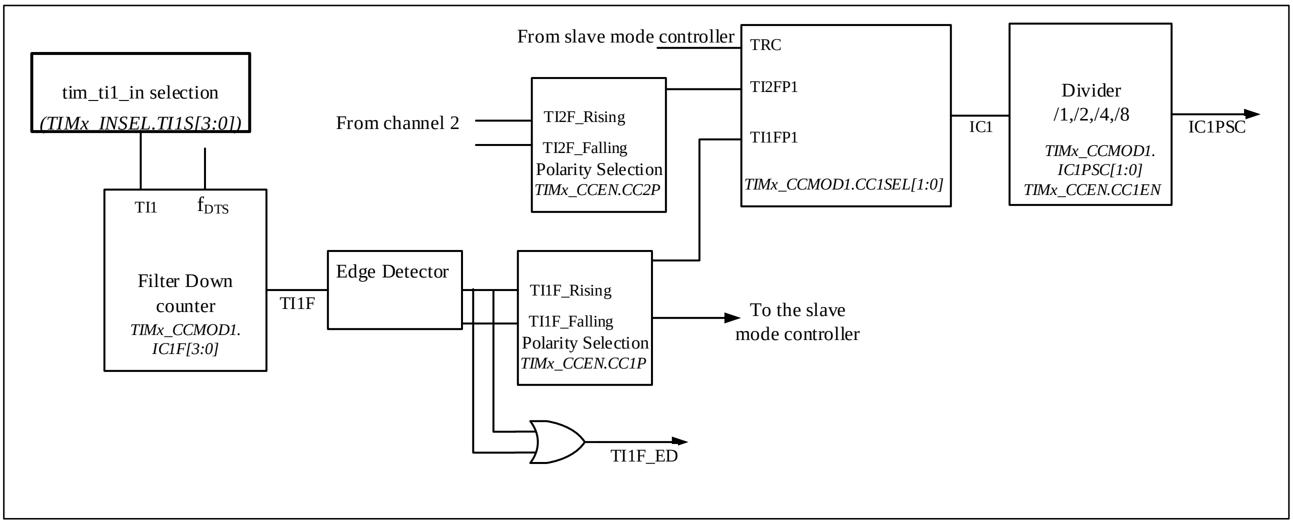


12.5.5 Capture/Compare Channels

Capture/compare channels include capture/compare registers and shadow registers. The input section consists of digital filters, multiplexers and prescalers. The output section includes comparators and output controls。

The input signal TIx is sampled and filtered to generate the signal TIxF. A signal (TIxF_rising or TIxF_falling) is then generated by the edge detector of the polarity select function, the polarity of which is selected by the TIMx_CCEN.CCxP bits. This signal can be used as a trigger input for the slave mode controller. At the same time, the signal ICx is sent to the capture register after prescale. The following figure shows a block diagram of a capture/compare channel.

Figure 12-17 Capture/Compare Channel (Example: Channel 1 Input Stage)



The output part generates an intermediate waveform OCxRef (active high) as reference. The polarity acts at the end of the chain。

Figure 12-18 Capture/Compare Channel 1 Main Circuit

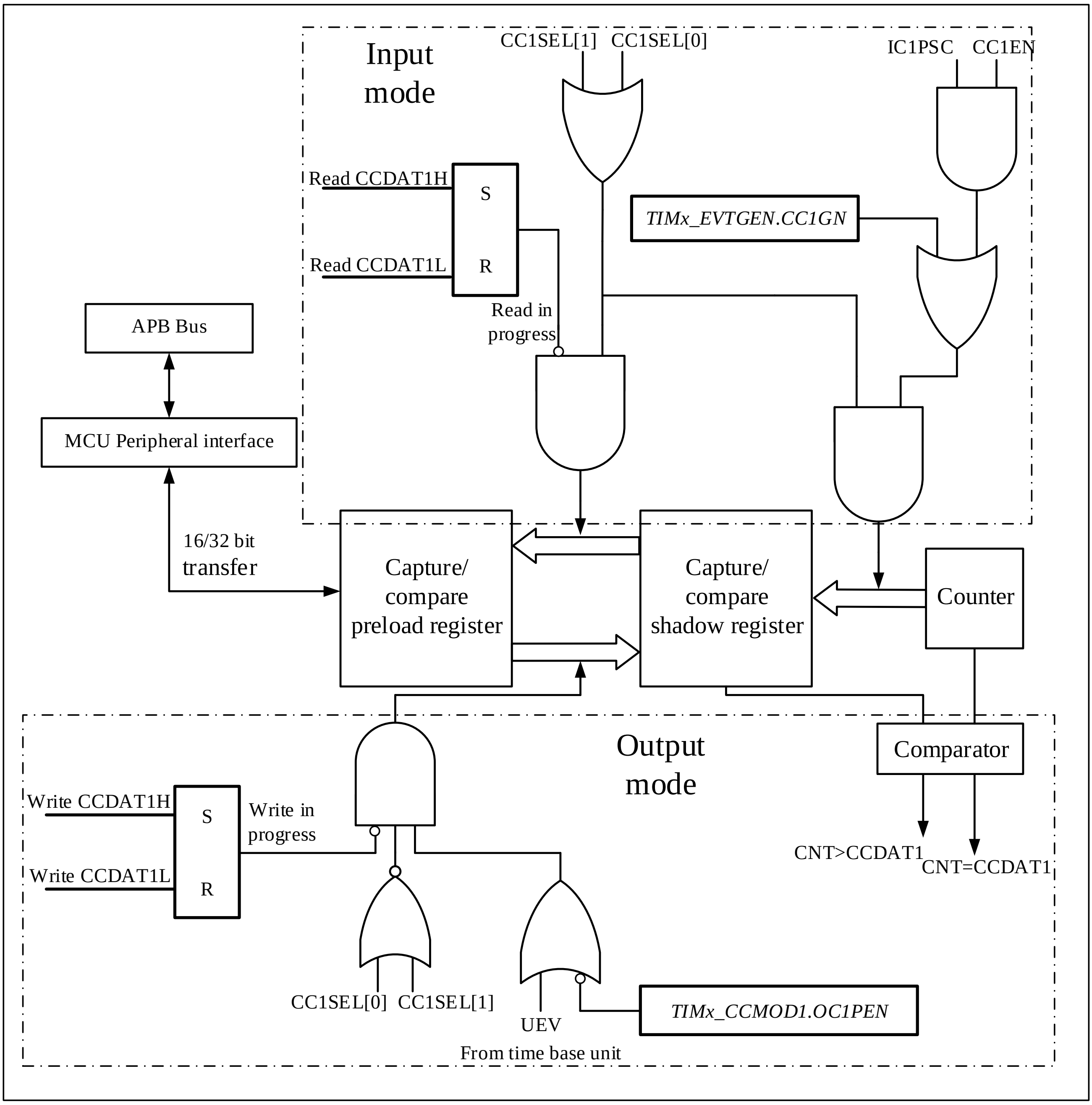


Figure 12-19 Output Part Of Channelx (x= 1; Take Channel 1 As An Example)

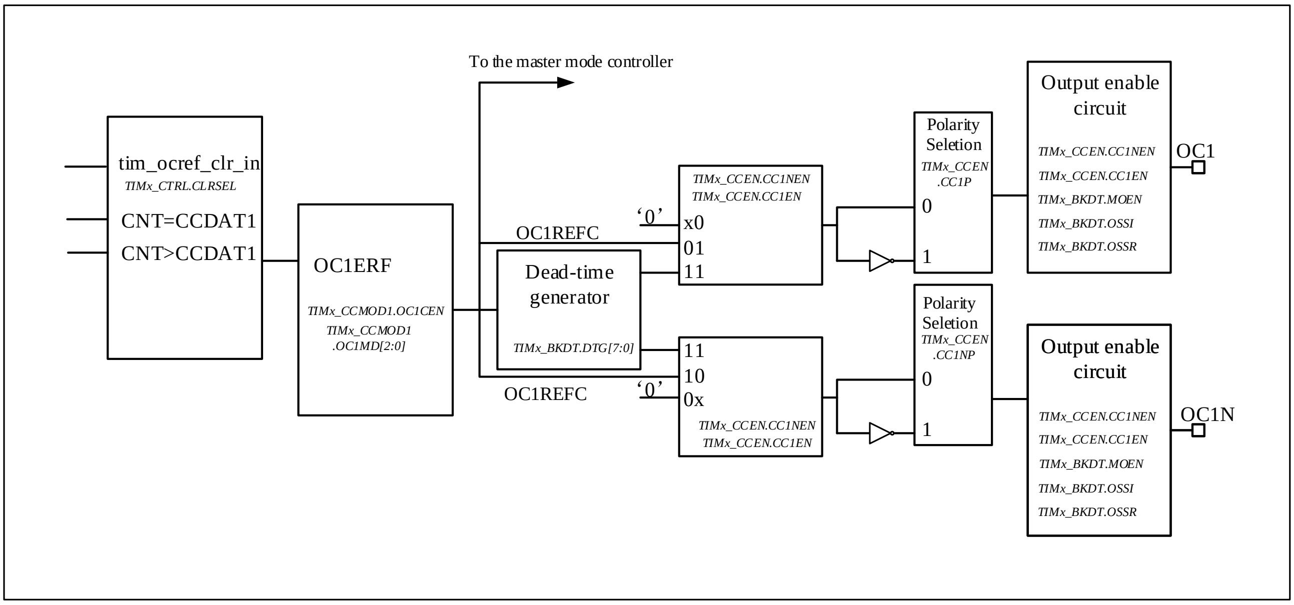
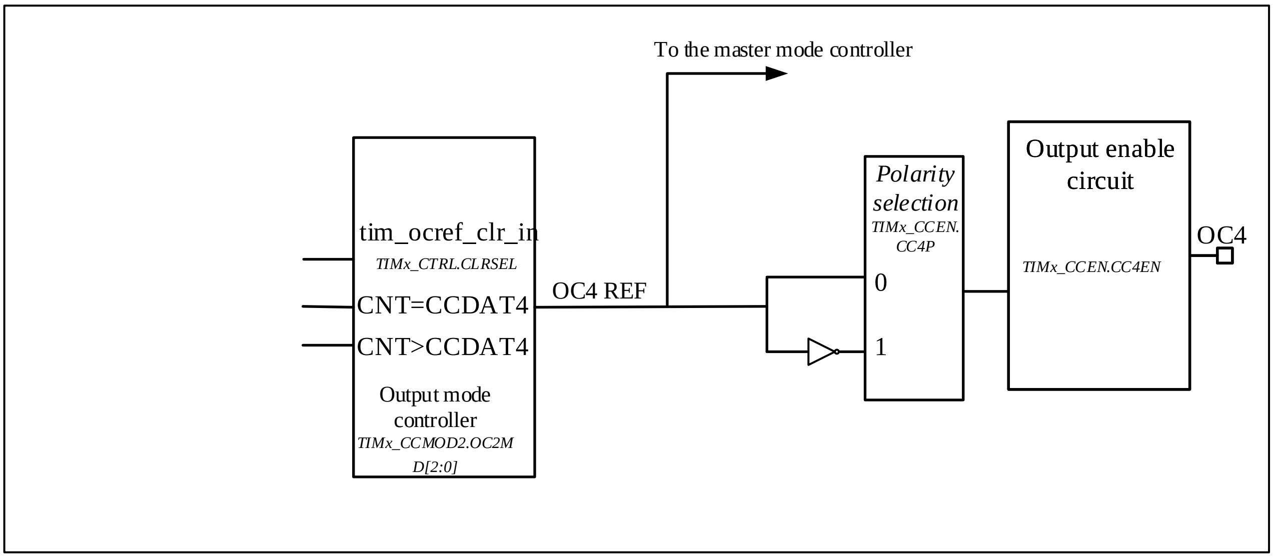


Figure 12-20 Output Part Of Channelx （X = 1,2,3,4;Take Channel 4 As An Example）



Reads and writes always access preloaded registers when capturing/comparing. The two specific working processes are as follows:

In capture mode, the capture is actually done in the shadow register, and then the value in the shadow register is copied into the preload register.

In compare mode, as opposed to capture mode, the value of the preload register is copied into the shadow register, which is compared with the counter.

12.5.6 Input Capture Mode

In capture mode, the TIMx_CCDATx registers are used to latch the counter value after the ICx signal detects.

There is a capture interrupt flag TIMx_STS.CCxITF, which can issue an interrupt or DMA request if the corresponding interrupt enable is pulled high.

The TIMx_STS.CCxITF bit is set by hardware when a capture event occurs and is cleared by software or by reading the TIMx_CCDATx register。

The overcapture flag TIMx_STS.CCxOCF is set equal to 1 when the counter value is captured in the TIMx_CCDATx register and TIMx_STS.CC1ITF is already pulled high. Unlike the former, TIMx_STS.CCxOCF is cleared by writing 0 to it.

To achieve a rising edge of the TI1 input to capture the counter value into the TIMx_CCDAT1 register, the configuration flow is as follows:

- To select a valid input:

Configure TIMx_CCMOD1.CC1SEL to '01'. At this time, the input is the CC1 channel, and IC1 is mapped to TI1.

- The duration of the input filter required for programming:

  Define the sampling frequency of the TI1 input and the length of the digital filter by configuring the TIMx_CCMODx.ICxF bits. Example: If the input signal jitters up to 5 internal clock cycles, we must choose a filter duration longer than these 5 clock cycles. When 8 consecutive samples (sampled at f_{DTS} frequency) with the new level are detected, we can validate the transition on TI1. Then configure TIMx_CCMOD1.IC1F to '0011'.

- By configuring TIMx_CCEN.CC1P=0, select the rising edge as the valid transition polarity on the TI1 channel
- Configure the input prescaler. In this example, configure TIMx_CCMOD1.IC1PSC='00' to disable the prescaler because we want to capture every valid transition.
- Enable capture by configuring TIMx_CCEN.CC1EN = '1'.

If you want to enable DMA request, you can configure TIMx_DINTEN.CC1DEN=1.If you want enable related interrupt request, you can configure TIMx_DINTEN.CC1IEN=1.

12.5.7 PWM Input Mode

There are some differences between PWM input mode and normal input capture mode, including:

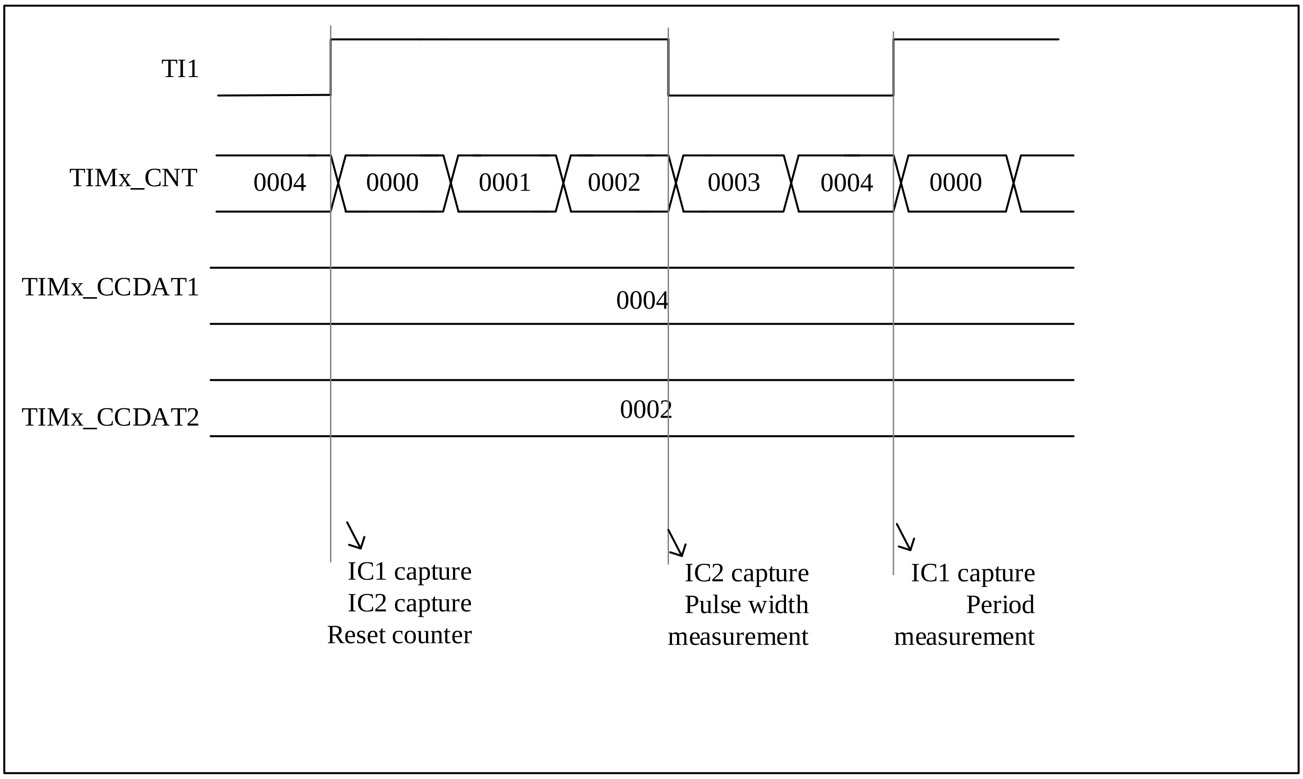
- Two ICx signals are mapped to the same TIx input.
- The two ICx signals are active on edges of opposite polarity.
- Select one of two TIxFP signals as trigger input.
- The slave mode controller is configured in reset mode.

For example, the following configuration flow can be used to know the period and duty cycle of the PWM signal on TI1 (It depends on the frequency of CK_INT and the value of the prescaler).

- Configure TIMx_CCMOD1.CC1SEL equal to '01', select TI1 as valid input for TIMx_CCDAT1.
- Configure TIMx_CCEN.CC1P equal to '0' to select the active polarity of filtered timer input 1(TI1FP1), active at the rising edge.

- Configure TIMx_CCMOD1.CC2SEL equal to ‘10’, select TI1 as valid input for TIMx_CCDAT2.
- Configure TIMx_CCEN.CC2P equal to 1 to select the valid polarity of filtered timer input 2(TI1FP2), active at the falling edge.
- Configure TIMx_SMCTRL.TSEL=101 to select Filtered timer input 1 (TI1FP1) as valid trigger input.
- Configure TIMx_SMCTRL.SMSEL=‘0100’ to configure the slave mode controller to reset mode.
- Configure TIMx_CCEN. CC1EN=1 and TIMx_CCEN.CC2EN=1 to enable capture.

Figure 12-21 PWM Input Mode Timing



Because of only filter timer input 1 (TI1FP1) and filter timer input 2 (TI2FP2) are connected to the slave mode controller, the PWM input mode can only be used with the TIMx_CH1/TIMx_CH2 signals.

12.5.8 Forced Output Mode

Software can force output compare signals to active or inactive level directly, in output mode (TIMx_CCMODx.CCxSEL=‘00’).

User can set TIMx_CCMODx. OCxMD=‘101’ to force the output compare signal to active level. And the OCxREF will be forced high, OCx get opposite value to CCxP polarity bit. On the other hand, user can set TIMx_CCMODx. OCxMD=‘100’ to force the output compare signal to inactive level, the OCxREF will be forced low.

The values of the TIMx_CCDATx shadow register and the counter still comparing with each other in this mode.

The comparison between the output compare register TIMx_CCDATx and the counter TIMx_CNT has no effect on OCxREF. And the flag still can be set. Therefore, the interrupt and DMA requests still can be sent.

12.5.9 Output Compare Mode

User can use this mode to control the output waveform, or to indicate that a period of time has elapsed.

When the capture/compare register and the counter have the same value, the output compare function’s operations are as follow：

- TIMx_CCMODx.OCxMD is for output compare mode, and TIMx_CCEN.CCxP is for output polarity. When the compare matches, if set TIMx_CCMODx.OCxMD=‘000’, the output pin will keep its level;if set TIMx_CCMODx.OCxMD=‘001’, the output pin will be set active;if set TIMx_CCMODx.OCxMD=‘010’, the output pin will be set inactive;if set TIMx_CCMODx.OCxMD=‘011’, the output pin will be set to toggle.
- Set TIMx_STS.CCxITF
- If user set TIMx_DINTEN.CCxIEN, a corresponding interrupt will be generated
- If user set TIMx_DINTEN.CCxDEN and set TIMx_CTRL2.CCDSEL to select DMA request, and DMA request will be sent.

User can set TIMx_CCMODx.OCxPEN to choose capture/compare shawdow regisete using capture/compare preload registers(TIMx_CCDATx) or not.

The time resolution is one counting period of the counter.

In one pulse mode, the output compare mode can also be used to output a single pulse.

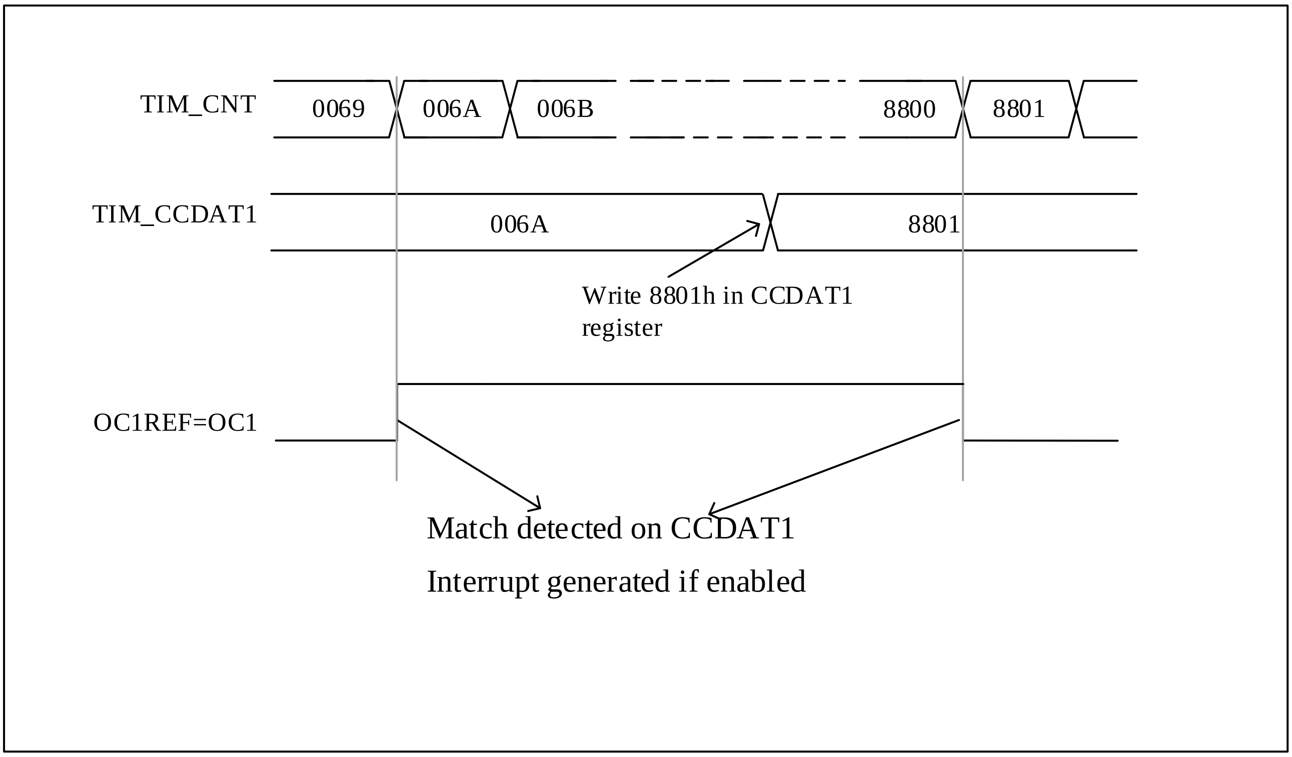
Here are the configuration steps for output compare mode:

- First of all, user should select the counter clock.
- Secondly, set TIMx_AR and TIMx_CCDATx with required data
- If user need to generate an interrupt, set TIMx_DINTEN.CCxIEN
- Then select the output mode by set TIMx_CCEN.CCxP, TIMx_CCMODx.OCxMD, TIMx_CCEN.CCxEN, etc
- At last, set TIMx_CTRL1.CNTEN to enable the counter

User can update the output waveform by setting TIMx_CCDATx at any time, as long as the preload register is not enabled. Otherwise the TIMx_CCDATx shadow register will be updated at the next update event。

Here is an example：

Figure 12-22 Output Compare Mode, Toggle On OC1



12.5.10 PWM Mode

User can use PWM mode to generate a signal whose duty cycle is determined by the value of the TIMx_CCDATx register and whose frequency is determined by the value of the TIMx_AR register. And depends on the value of TIMx_CTRL1.CAMSEL, the TIM can generate PWM signal in edge-aligned mode or center-aligned mode.

User can set PWM mode 1 or PWM mode 2 by setting TIMx_CCMODx. OCxMD=‘110’ or setting TIMx_CCMODx. OCxMD=‘111’. To enable preload register, user must set corresponding TIMx_CCMODx.OCxPEN. And then set TIMx_CTRL1.ARPEN to auto-reload preload register eventually.

User can set polarity of OCx by setting TIMx_CCEN.CCxP. To enable the output of OCx, user need to set the combination of the value of CCxEN,CCxNEN,MOEN,OSSI and OSSR in TIMx_CCEN and TIMx_BKDT.

The values of TIMx_CNT and TIMx_CCDATx are always compared with each other when the TIM is under PWM mode.

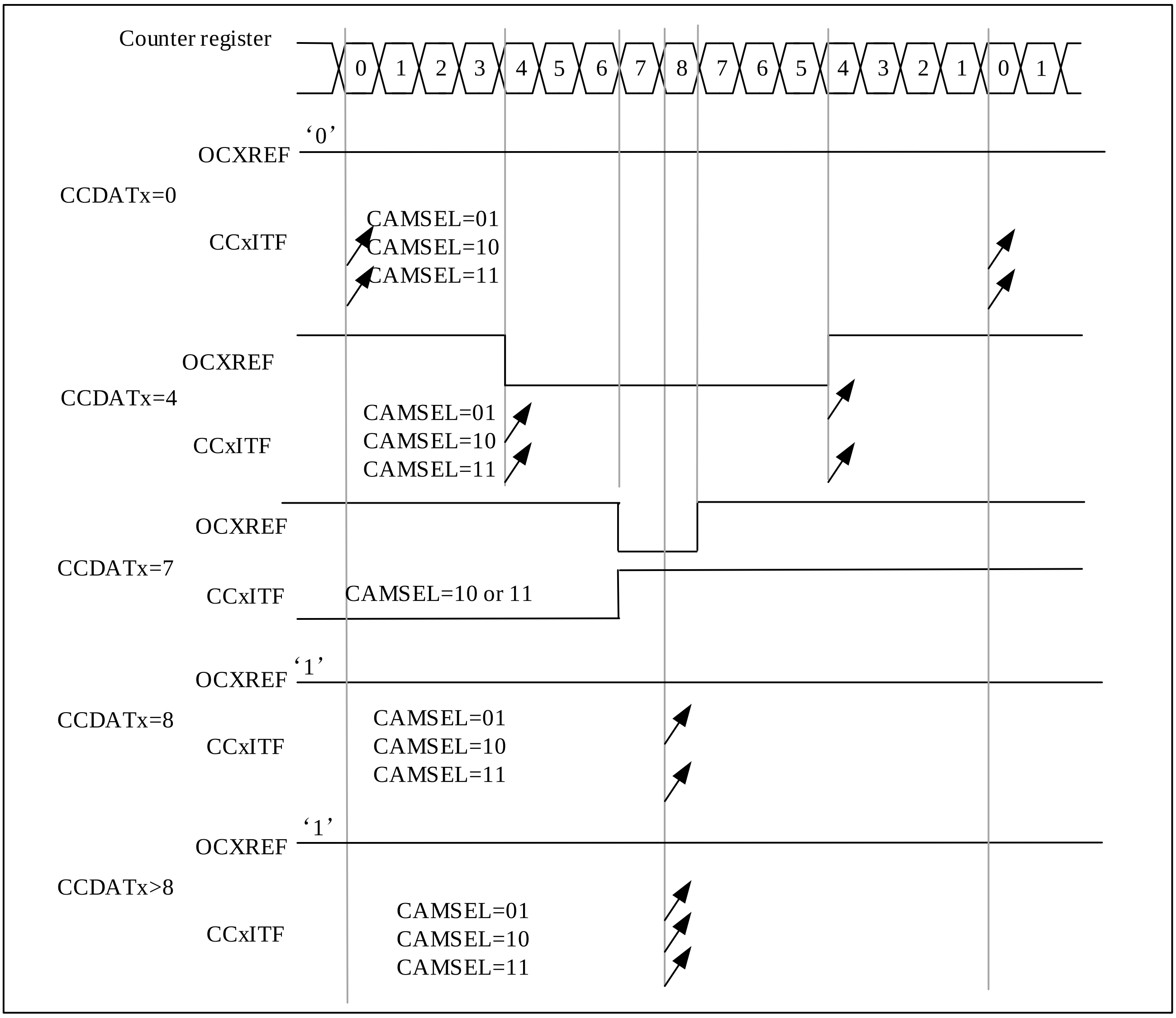
Only if an update event occurs, the preload register will transfer to the shadow register. Therefore user must reset all the registers by setting TIMx_EVTGEN.UDGN before the counter starts counting.

12.5.10.1 PWM center-aligned mode

If user set TIMx_CTRL1.CAMSEL equal ‘01’, ‘10’ or ‘11’, the PWM center-aligned mode will be active. The setting of the compare flag depends on the value of TIMx_CTRL1.CAMSEL. There are three kinds of situation that the compare flag is set, only when the counter counts up, only when the counter counts down, or when the counter counts up and counts down. User should not modified TIMx_CTRL1.DIR by software, it is updated by hardware.

Examples of center-aligned PWM waveforms is as follow, and the setting of the waveform are: TIMx_AR=8, PWM mode 1, the compare flag is set when the counter counts down corresponding to TIMx_CTRL1. CAMSEL='01'.

Figure 12-23 Center-Aligned PWM Waveform (AR=8)



When using center-aligned mode, users should pay attention to the following considerations:

- It depends on the value of TIMx_CTRL1.DIR that the counter counts up or down. Cautions that the DIR and CAMSEL bits should not be changed at the same time.
- User should not write the counter while running in center-aligned mode, otherwise it will cause unexpected results. Here are some examples:
  - If the value written into the counter is 0 or is the value of TIMx_AR, the direction will be updated but the update event will not be generated.
  - If the value written into the counter is greater than the value of auto-reload, the direction will not be updated.
- To be on the safe side, user is suggested setting TIMx_EVTGEN.UDGN to generate an update by software before starting the counter, and not writing the counter while it is running.

12.5.10.2 PWM Center-Aligned Asymmetric Mode

Please refer to Section 12.5.2.3.2 for information on PWM Center-Aligned Asymmetric Mode.

12.5.10.3 PWM edge-aligned mode

There are two kinds of configuration in edge-aligned mode, up-counting and down-counting.

- Up-counting

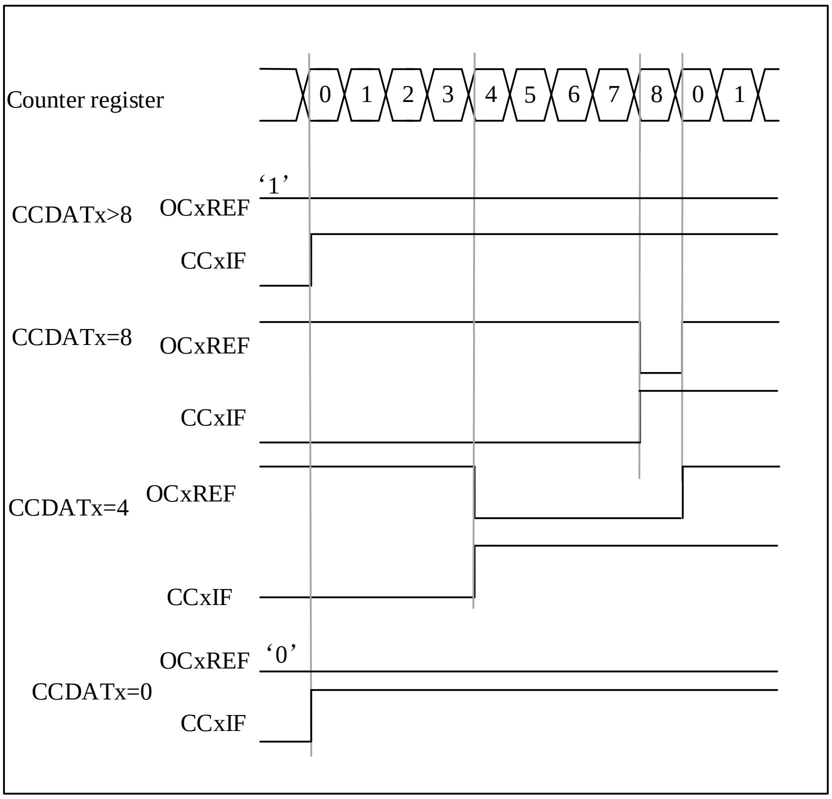
User can set TIMx_CTRL1.DIR=0 to make counter counts up.

Example for PWM mode1:

When TIMx_CNT < TIMx_CCDATx, the reference PWM signal OCxREF is high. Otherwise it will be low. If the compare value in TIMx_CCDATx is greater than the auto-reload value, the OCxREF will remains 1. Conversely, if the compare value is 0, the OCxREF will remains 0.

When TIMx_AR=8, the PWM waveforms are as follow:

Figure 12-24 Edge-Aligned PWM Waveform (AR=8)



- Down-counting

User can set TIMx_CTRL1.DIR=1 to make counter counts down.
Example for PWM mode1:

When TIMx_CNT > TIMx_CCDATx, the reference PWM signal OCxREF is low. Otherwise it will be high. If the compare value in TIMx_CCDATx is greater than the auto-reload value, the OCxREF will remains 1.

Note: If the nth PWM cycle CCDATx shadow register >= AR value, the shadow register value of CCDATx in the *(n+1)th PWM cycle is 0. At the moment when the counter is 0 in the (n+1)th PWM cycle, although the value of the counter = CCDATx shadow register = 0 and OCxREF = '0', no compare event will be generated。*

12.5.11 Combined PWM Mode

In the combined PWM mode, programmable delays and phase shifts are allowed between the pulses of the two generated edge-aligned or center-aligned PWM signals. The frequency is determined by the value of the TIMx_AR register, while the duty cycle and delays are determined by two TIMx_CCMODx registers. The generated signal OCxREFC is composed of a logical OR or logical AND combination of the two reference PWM signals.

- OC1REFC (or OC2REFC) is controlled by TIMx_CCDAT1 and TIMx_CCDAT2.
- OC3REFC (or OC4REFC) is controlled by TIMx_CCDAT3 and TIMx_CCDAT4.

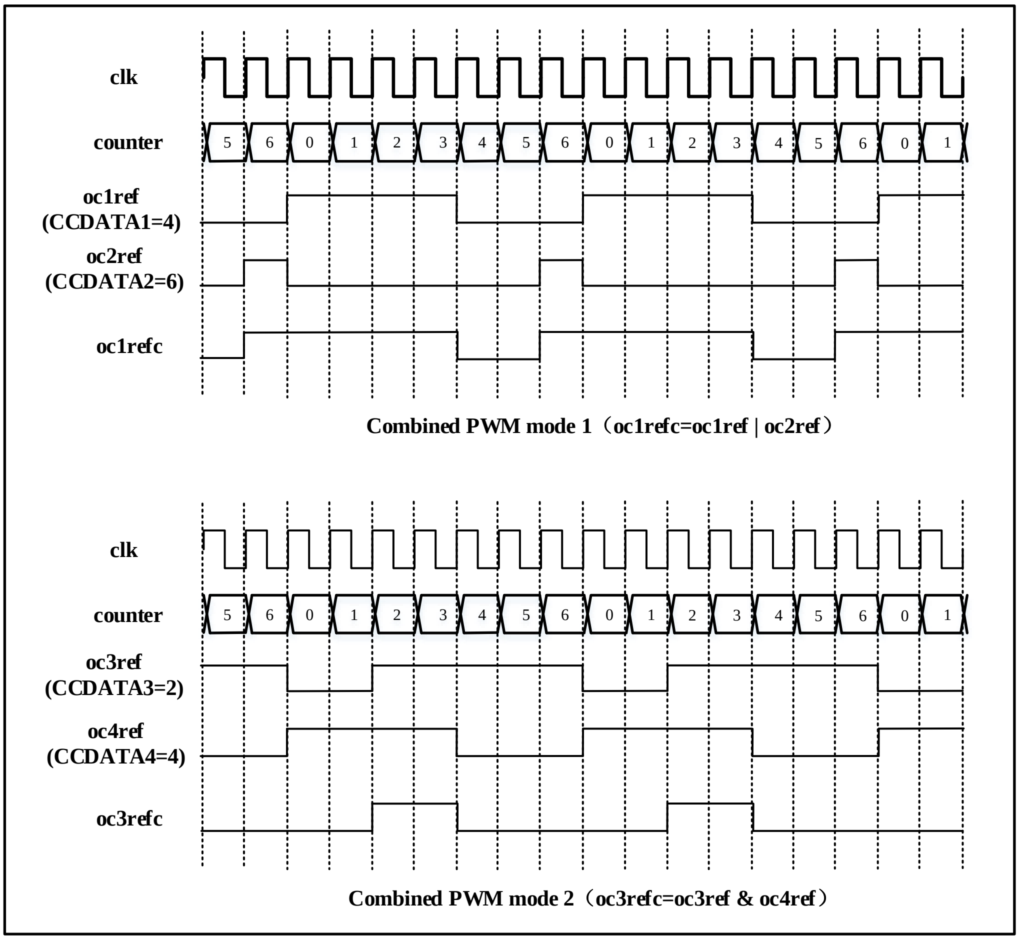
Two channels can independently select the combined PWM mode (one OCx output for each pair of TIMx_CCDATx registers) by writing ’1’ to the OCxMD3 bit of the TIMx_CCMODx register, writing ’110’ to the OCxMD bit (Combined PWM mode 1), or writing ’1’ to the OCxMD3 bit and ’111’ to the OCxMD bit (Combined PWM mode 2).

When a given channel is used as a combined PWM channel, its complementary channel must be configured in the opposite PWM mode (for example, one channel configured in Combined PWM mode 1 and the other channel configured in Combined PWM mode 2).

The following diagram illustrates signal examples that can be generated in combined PWM mode, and these signals can be obtained through the following configurations:

- Channel 1 configured in Combined PWM mode 1.
- Channel 2 configured in PWM mode 2.
- Channel 3 configured in Combined PWM mode 2.
- Channel 4 configured in PWM mode 1.

Figure 12-25 Combined PWM mode on channels 1 and 3



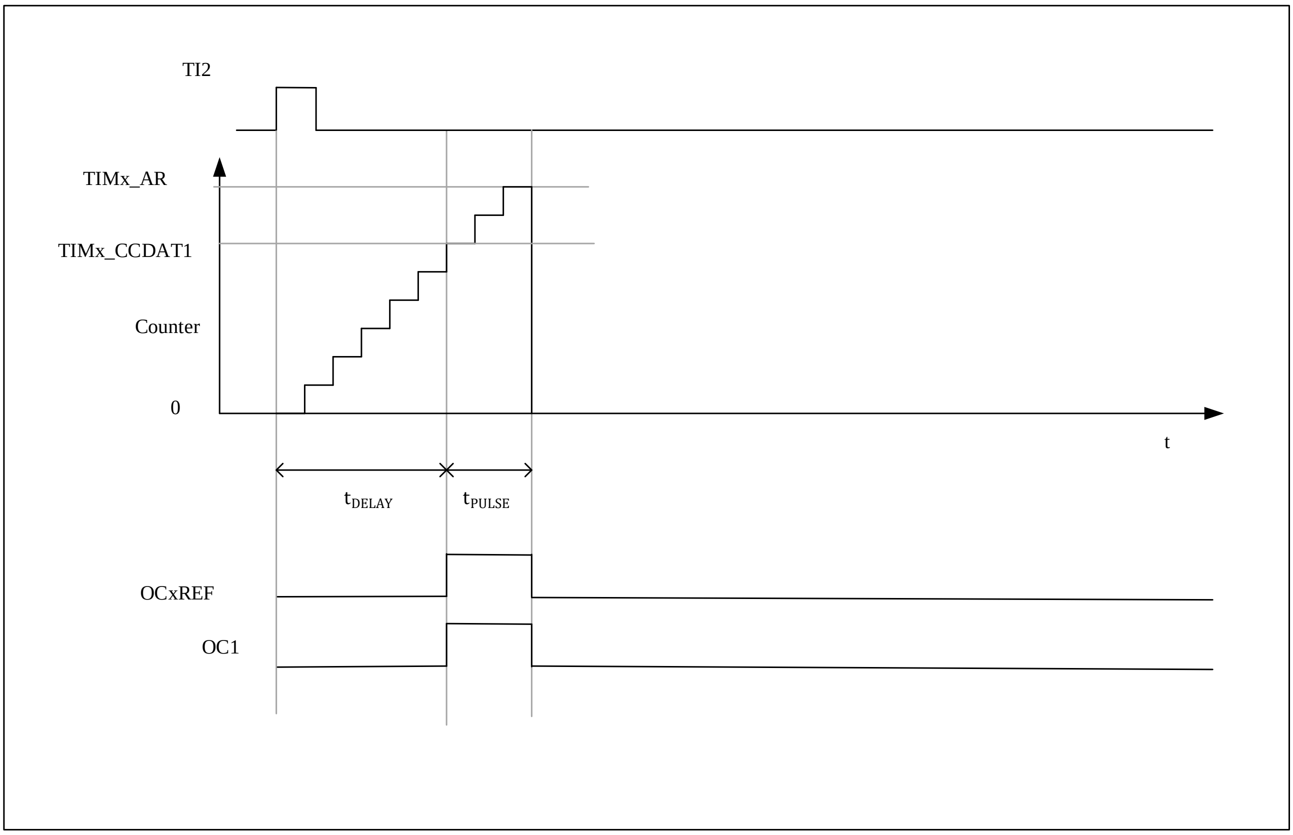
Combined PWM mode 1（oc1refc=oc1ref | oc2ref）

Combined PWM mode 2（oc3refc=oc3ref & oc4ref）

12.5.12 One-Pulse Mode

In the one-pulse mode (ONEPM), a trigger signal is received, and a pulse t_{PULSE} with a controllable pulse width is generated after a controllable delay t_{DELAY}. The output mode needs to be configured as output compare mode or PWM mode. After selecting one-pulse mode, the counter will stop counting after the update event UEV is generated.

Figure 12-26 Example Of One-Pulse Mode



The following is an example of a one-pulse mode:

A rising edge trigger is detected from the TI2 input, and a pulse with a width of t_{PULSE} is generated on OC1 after a delay of t_{DELAY}.

1. Counter configuration: count up, counter TIMx_CNT < TIMx_CCDAT1 ≤ TIMx_AR;
2. TI2FP2 is mapped to TI2, TIMx_CCMOD1.CC2SEL= '01'; TI2FP2 is configured for rising edge detection, TIMx_CCEN.CC2P= '0';
3. TI2FP2 acts as the trigger (TRGI) of the slave mode controller and starts the counter, TIMx_SMCTRL.TSEL= '110', TIMx_SMCTRL.SMSEL= '0110' (trigger mode);
4. TIMx_CCDAT1 writes the count value to be delayed (t_{DELAY}), TIMx_AR - TIMx_CCDAT1 is the count value of the pulse width t_{PULSE};
5. Configure TIMx_CTRL1.ONEPM='01' to enable single pulse mode, configure TIMx_CCMOD1.OC1MD = '111' to select PWM2 mode;
6. Wait for an external trigger event on TI2, and a one pulse waveform will be output on OC1.

12.5.12.1 Special case: OCx fast enable

In one-pulse mode, an edge is detected through the TIx input, and triggers the start of the counter to count to the comparison value and then output a pulse. These operations limit the minimum delay t_{DELAY} that can be achieved.

You can set TIMx_CCMODx.OCxFEN='1' to turn on OCx fast enable, after triggering the rising edge, the OCxREF signal will be forced to be converted to the same level as the comparison match occurs immediately, regardless of the comparison result. OCxFEN fast enable only takes effect when the channel mode is configured for PWM1 and PWM2 modes。

12.5.13 Retriggerable Single Pulse Mode

This mode allows the counter to start on a trigger signal and generate a programmable-length pulse, but there are differences between this and the non-retriggerable single pulse mode:

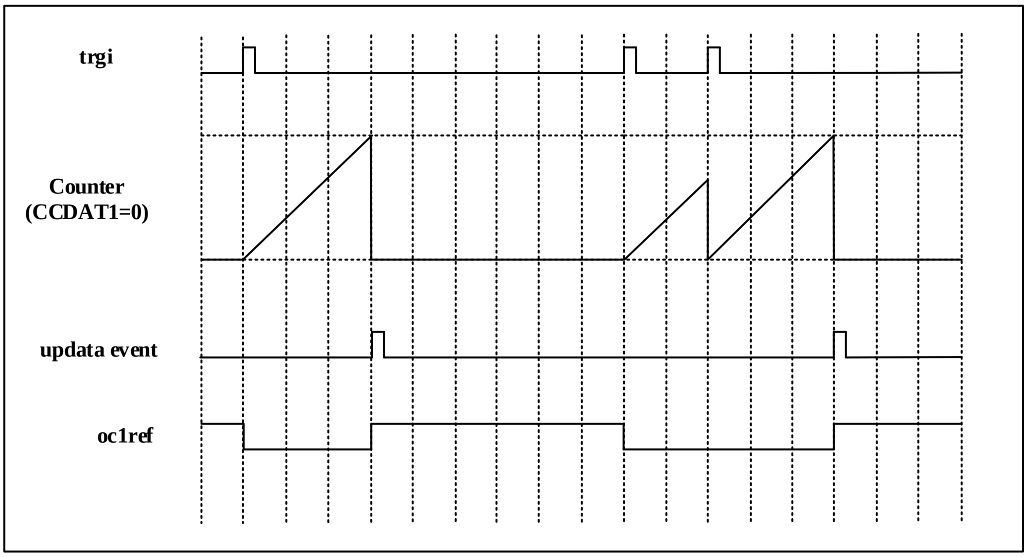
- When triggered, the pulse is generated immediately.
- If a new trigger occurs before the previous one is completed, the pulse will be extended.

The timer must be in slave mode with the TIMx_SMCTRL register bit SMSEL[3:0] = ‘1110’ (Combined reset + trigger mode). For retriggerable single pulse mode 1 or mode 2, write ‘1’ to the OCxMD3 bit and ‘000’ or ‘001’ to the OCxMD bit.

When the timer is configured in up-count mode, the corresponding TIMx_CCDATx must be set to 0 (AR register sets the pulse length). If the timer is configured in down-count mode, CCDATx must be greater than or equal to AR.

The following diagram show the retriggerable single pulse mode 1 as an example:

Figure 12-27 Retriggerable Single Pulse Mode 1



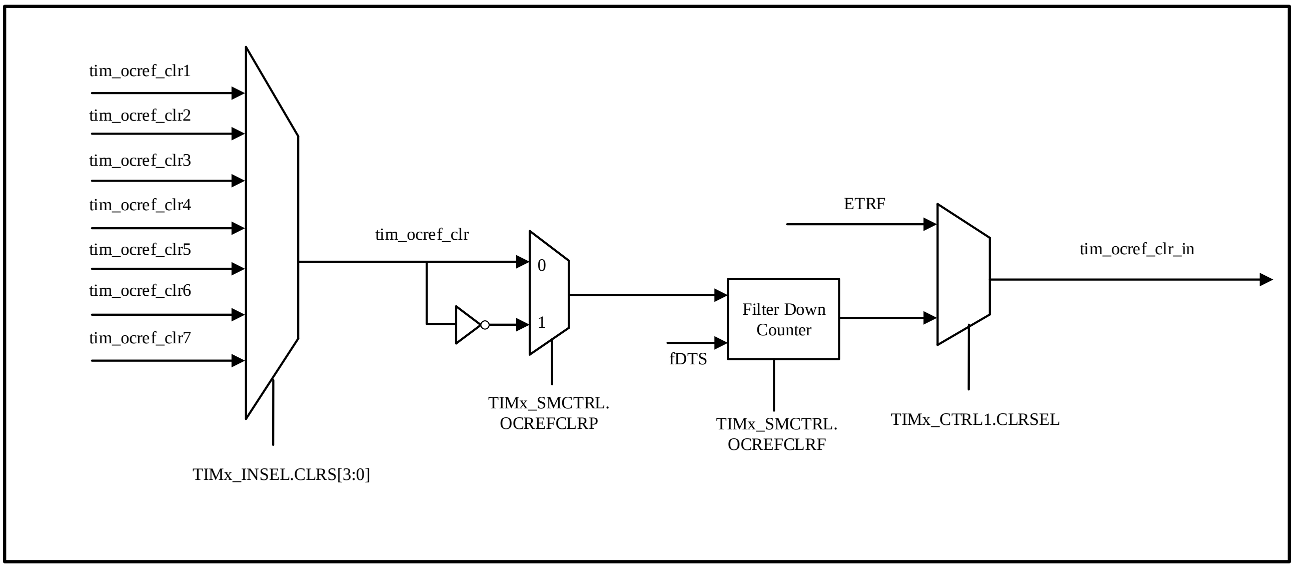
12.5.14 Clearing the OCxREF Signal On An External Event

If user set TIMx_CCMODx.OCxCEN=1, tim_ocref_clr_in input high level can be used to driven the OCxREF signal to low level, and the OCxREF signal will remains low level, until the next UEV happens. Only output compare and PWM modes can use this function. This cannot be used when it is in forced mode.

The clear signal input tim_ocref_clr_in can be selected as tim_ocref_clr or ETRF through the CLRSEL bit in the TIMx_CTRL1 register.

The tim_ocref_clr signal can be selected through the CLRS[3:0] in the TIMx_INSEL register, as shown in the following diagram

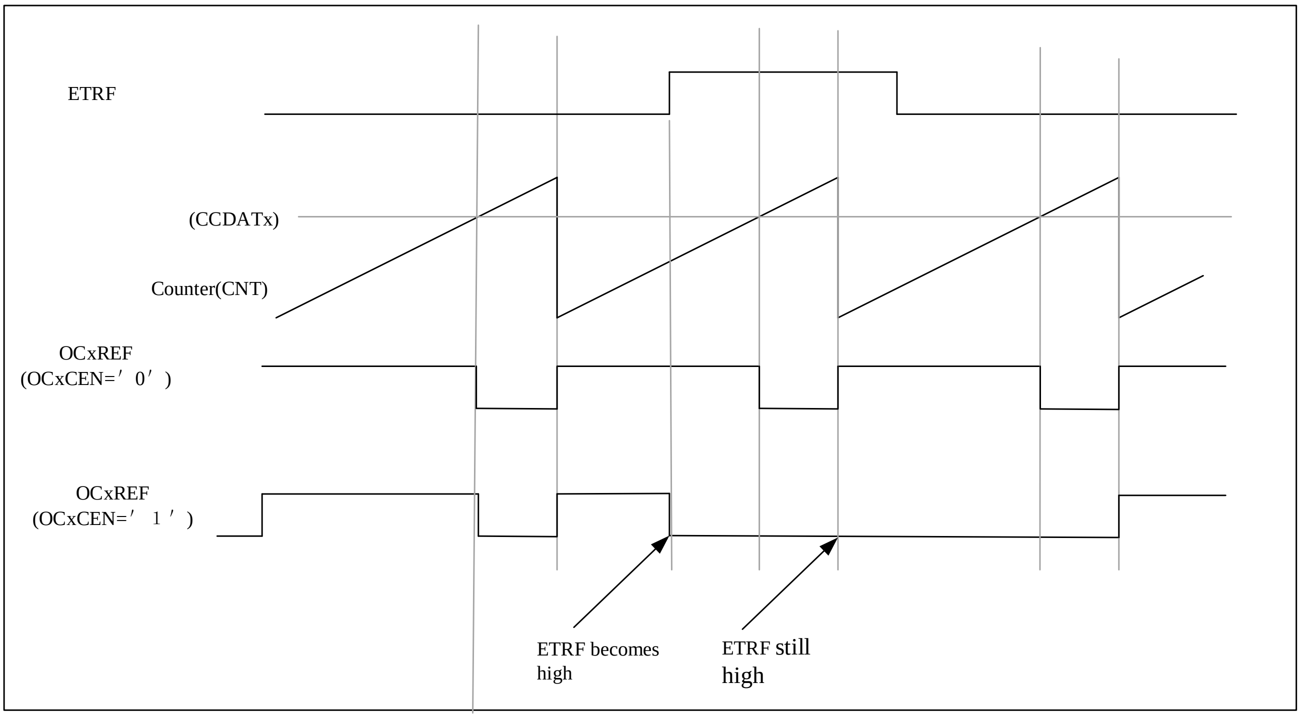
Figure 12-28 External event clearing OCxREF signal



For example, when the tim_ocref_clr_in signal is selected as ETRF, the tim_etr_in configuration is as follows：

- Set TIMx_SMCTRL.EXTPS=‘00’, disable the external trigger prescaler.
- Set TIMx_SMCTRL.EXCEN=‘0’, disable the external clock mode 2.
- Set TIMx_SMCTRL.EXTP and TIMx_SMCTRL.EXTF to configure the external trigger polarity and external trigger filter according to the need.
- When the ETRF input goes high, the behavior of the OCxREF signal varies for different OCxCEN values. In this case, the timer is set to PWM mode.

Figure 12-29 Clearing the OCxREF of TIMx



12.5.15 Complementary Output and Dead-Time Insertion

GTIMx (x=8-10) can output two complementary signals (CH1 and CH1N) and manage the closing and opening of the output, which involvting dead time. Users adjest the dead time on the characteristics of the connected device to ensure proper operation and avoid potential issues.

Users can select the polarity of the output by setting TIMx_CCEN.CCxP and TIMx_CCEN.CCxNP, and this selection is independent for each output.

Users can control the complementary signals OCx and OCxN by setting a combination of control bits, namely TIMx_CCEN.CCxEN, TIMx_CCEN.CCxNEN, TIMx_BKDT.MOEN, TIMx_CTRL2.OIx, TIMx_CTRL2.OIxN, TIMx_BKDT.OSSI, and TIMx_BKDT.OSSR. The dead time will be activated when switching to the idle state.

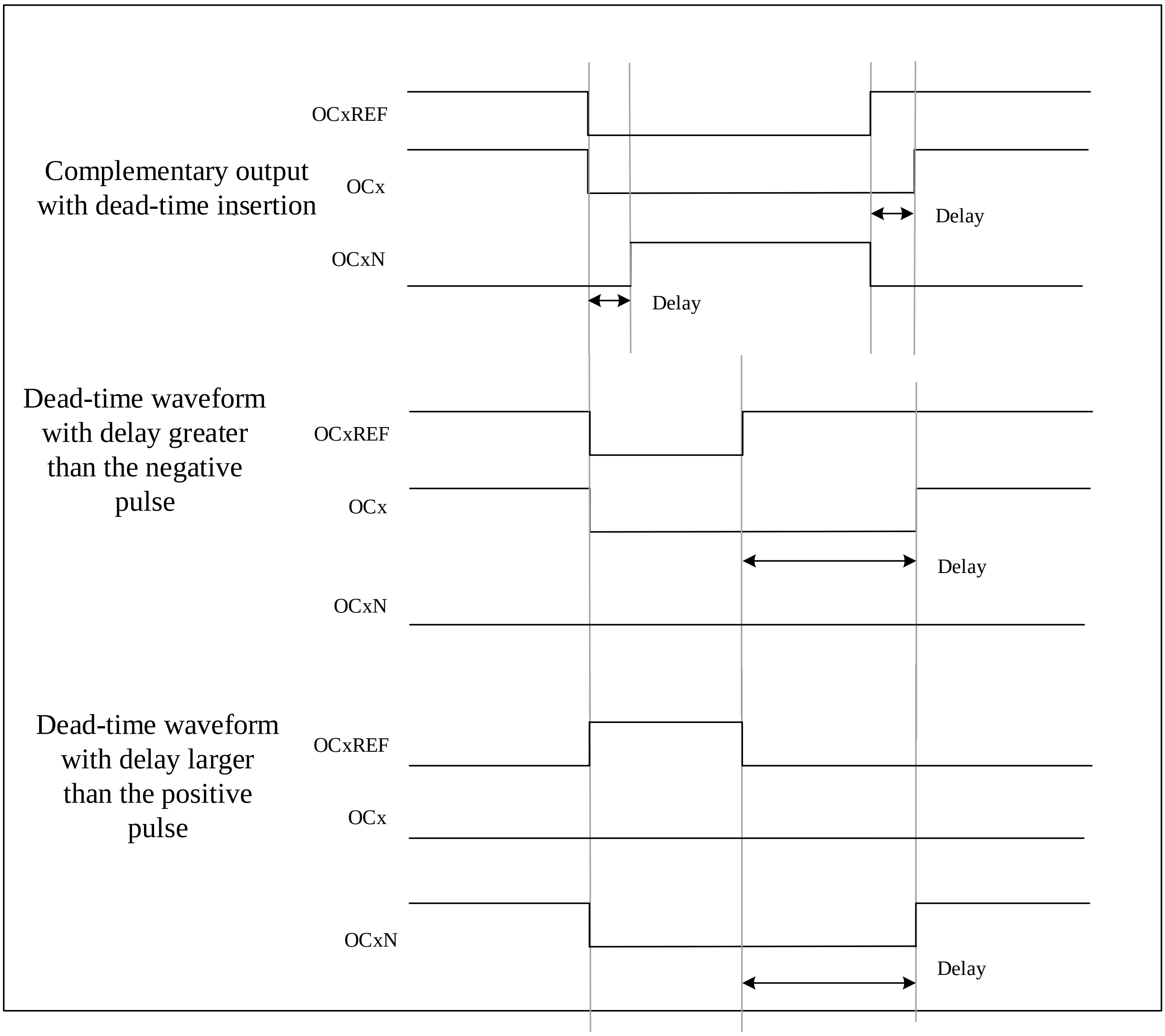
If the user sets both TIMx_CCEN.CCxEN and TIMx_CCEN.CCxNEN, dead time will be inserted. If there is breaking, TIMx_BKDT.MOEN should also be set. Each channel has a 10-bit dead-time generator.

The reference waveform OCxREF can generate two outputs, OCx and OCxN. If OCx and OCxN are active high, the OCx output signal will be the same as the reference signal, while the OCxN output signal will be the inverse of the reference signal. However, the OCx output signal will have a delay relative to the reference rising edge, and the OCxN output signal will have a delay relative to the reference falling edge. If the delay is greater than the width of the effective OCx or OCxN output, the corresponding pulse will not be generated.

The relationship between the output signal of the dead-time generator and the reference signal OCxREF is as follows:

If TIMx_CCEN.CCxP=0， TIMx_CCEN.CCxNP=0， TIMx_BKDT.MOEN=1， TIMx_CCEN.CCxEN=1， TIMx_CCEN.CCxNEN=1.

Figure 12-30 Complementary Output with Dead-Time Insertion


Complementary output with dead-time insertion
OCxREF
OCx
OCxN
Delay
Delay
Dead-time waveform with delay greater than the negative pulse
OCxREF
OCx
Delay
OCxN
Dead-time waveform with delay larger than the positive pulse
OCxREF
OCx
OCxN
Delay

Users can set TIMx_BKDT.DTGN to program the dead-time delay for each channel.

12.5.15.1 Redirecting OCxREF to OCx or OCxN

In output mode, users can set TIMx_CCEN.CCxEN and TIMx_CCEN.CCxNEN to redirect OCxREF to the OCx output or OCxN output.

There are two ways to use this feature. When the complementary is held at its inactive level, users can use this feature to send specific waveforms, such as PWM or static active levels. Users can also use this feature to set both outputs to inactive levels, or set both outputs to active levels, complementary and with dead time.

If the user sets TIMx_CCEN.CCxEN=0 and TIMx_CCEN.CCxNEN=1, they are not complementary. When OCxREF is high, OCxN will become active. On the other hand, if the user sets TIMx_CCEN.CCxEN=1 and TIMx_CCEN.CCxNEN=1, OCx will become active when OCxREF is high. Conversely, when OCxREF is low, OCxN will become active.

12.5.16 Breaking Function

When using the breaking function, setting the corresponding control bits will modify the output enable signal and inactive level. However, at any time, the outputs of OCx and OCxN cannot be at active levels simultaneously, meaning that the condition (CCxP ^ OIx) ^ (CCxNP ^ OIxN) != 0 must be satisfied.

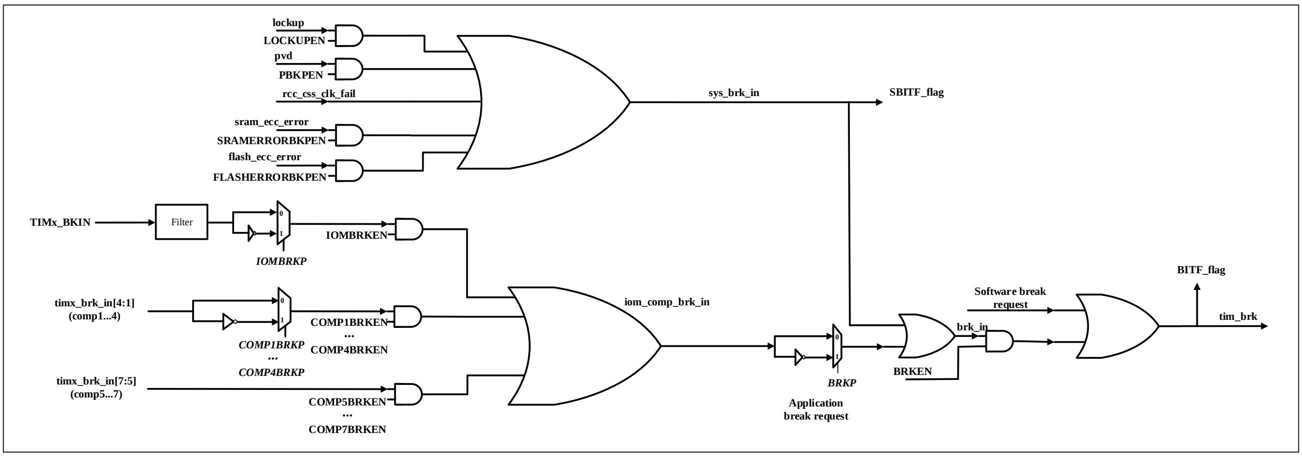
When multiple breaking signals are enabled, each breaking signal forms an OR logic. There are several signals that may act as sources of breaking.

Break 1：

- Break 1 input pin.
- Clock failure event, generated by the Clock Security System(CSS) in the RCC clock.
- PVD event.
- Core Hardfault event.
- SRAM ECC error.
- SRAM parity error.
- FLASH ECC error.
- Comparator output signal.
- Software set TIMx_EVTGEN.BGN.

Perform OR operation on all sources before entering the timer tim_brk input, as shown in the diagram.

Figure 12-31 Break input



Note: Asynchronous (clockless) operation can only be guaranteed when the programmable filters are disabled. If the programmable filters are enabled, the fault-safe clock mode (e.g., using internal PLL and/or CSS) must be used to ensure handling of open circuit events.

After reset, the break circuit will be disabled. The MOEN bit will be at a low level. Users can set TIMx_BKDT.BKEN to enable the break function. By setting TIMx_BKDT.BKP, you can select the polarity of the break input signal. Users can simultaneously modify TIMx_BKDT.BKEN and TIMx_BKDT.BKP. After users set TIMx_BKDT.BKEN and TIMx_BKDT.BKP, there is a 1 APB clock cycle delay before it takes effect. Therefore, users need to wait for 1

APB clock cycle before reading back the written value.

The falling edge of MOEN can be asynchronous, so a synchronization circuit is placed between the actual signal and the synchronous control bit. This circuit will introduce a delay between the asynchronous and synchronous signals. When the user sets TIMx_BKDT.MOEN to a low level, a delay needs to be inserted before reading this value. This is because an asynchronous signal is written, but the user is reading a synchronous signal.

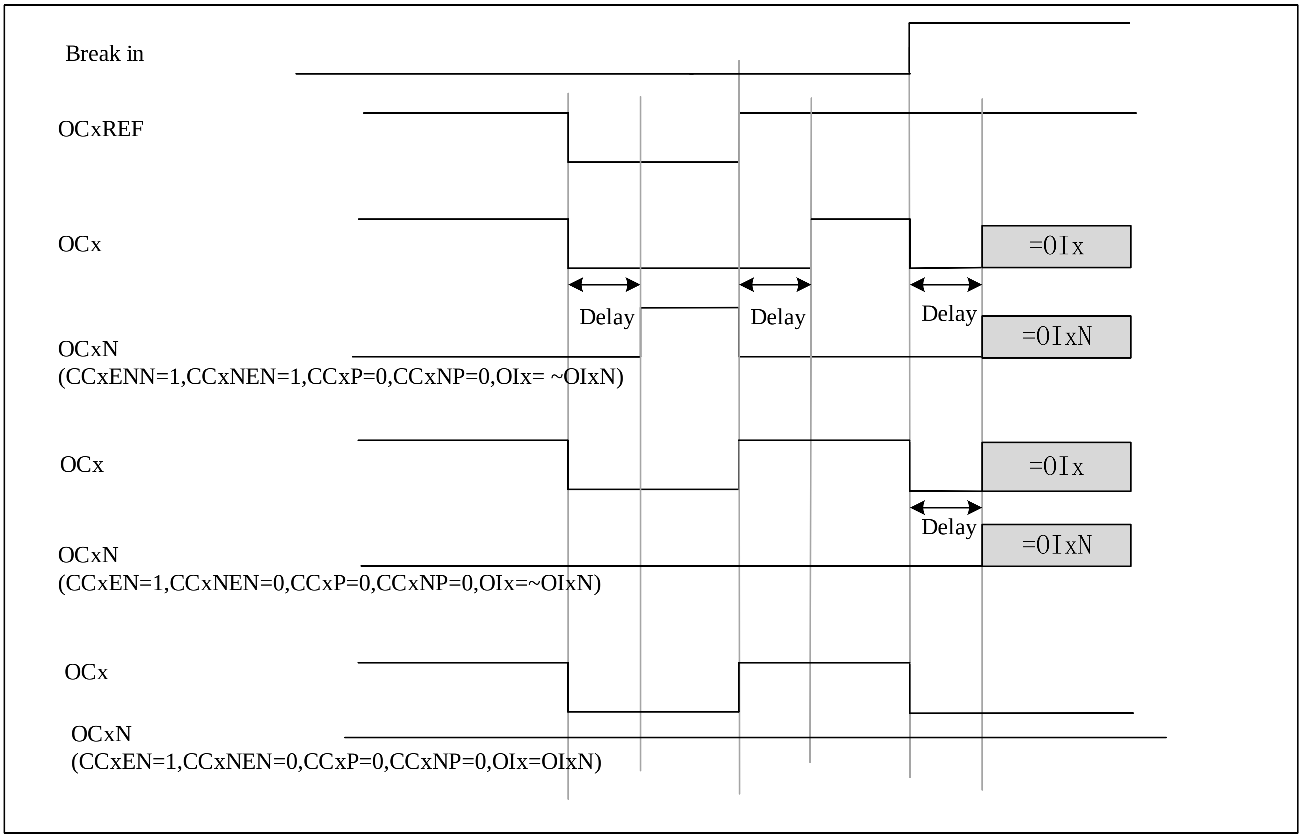
The behavior after the break occurs is as follows:

- TIMx_BKDT.MOEN will be asynchronously cleared, and then the output will enter an invalid state, idle state, or reset state. The output state can be selected by setting TIMx_BKDT.OSSI. This will take effect even if the MCU oscillator is turned off.
- Once TIMx_BKDT.MOEN=0,the output of each output channel will be driven by the level programmed in TIMx_CTRL2.OIx. If TIMx_BKDT.OSSI=0, the timer will release the enable output (taken over by the GPIO controller), otherwise, it will remain at a high level.
- If the user chooses to use complementary outputs, the behavior of TIM is as follows:
  - Depending on the polarity, the output will first be set to the reset state. This is an asynchronous option, so it can still working even if no clock is provided to the timer.
  - If the timer clock is still provided, the dead-time generator will be reactivated when (CCxP ^ OIx) ^ (CCxNP^OIxN) != 0, meaning that OCx and OCxN still cannot be simultaneously driven to valid levels. After the dead time, the outputs will be driven based on the values of TIMx_CTRL2.OIx and TIMx_CTRL2.OIxN. Please note that due to the synchronization circuit on MOEN (approximately 2 ck_tim periods), the dead time will be longer than usual.
  - If TIMx_BKDT.OSSI=0, the timer will release output control. Otherwise, if the enable output is at a high level, it will remain high. If it is at a low level, it will become high when TIMx_CCEN.CCxEN or TIMx_CCEN.CCxNEN is at a high level.
- If TIMx_DINTEN.BIEN=1, an interrupt will be generated when TIMx_STS.BITF=1.
- If the user sets TIMx_BKDT.AOEN, TIMx_BKDT.MOEN will be automatically set on the next UEV occurrence. The user can adjust it using this feature. If the user does not set TIMx_BKDT.AOEN, TIMx_BKDT.MOEN will remain at a low level until set to 1 again. In this case, the user can use it to ensure safety. The user can connect the break input to a thermal sensor, power driver alarm, or other safety components.
- When the break input is active, TIMx_BKDT.MOEN cannot be automatically set or set simultaneously by software, and TIMx_STS.BITF cannot be cleared. This is because the break input is in an active state at the level.

To ensure application safety, the break circuit has write protection functionality and break input-output management. It allows users to freeze certain parameters, such as dead-time duration, OCx/OCxN polarity and disabled state, OCxMD configuration, break enable and polarity. Users can choose one of three protection levels by setting TIMx_BKDT.LCKCFG. However, TIMx_BKDT.LCKCFG can only be written once after MCU reset.

The output behavior in response to the break is shown as follows:

Figure 12-32 The output behavior in response to the break

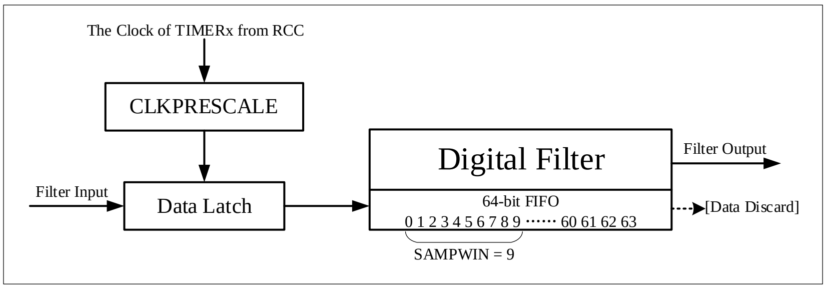

Break in
OCxREF
OCx
=OIx
Delay
Delay
Delay
=OIxN
OCxN
(CCxENN=1,CCxNEN=1,CCxP=0,CCxNP=0,OIx= ~OIxN)
OCx
=OIx
Delay
=OIxN
OCxN
(CCxEN=1,CCxNEN=0,CCxP=0,CCxNP=0,OIx=~OIxN)
OCx
OCxN
(CCxEN=1,CCxNEN=0,CCxP=0,CCxNP=0,OIx=OIxN)

- The tim_brk input can disable (invalid state) the PWM output, or force the PWM output to a predefined safe state.

12.5.16.1 Break Filter

The register TIMx_BKFR is described as follows:

Figure 12-33 Sliding filter



- The digital filter samples the brake signal using the RCC's TIMx clock and accumulates the samples in a 64-bit FIFO. It samples data only within the window size defined in TIMx_BKFR.WSIZE [5:0], with a maximum size of 64.
- The filter output samples the majority value within the sampling window, which is defined by the threshold in TIMx_BKFR.THRESH [5:0], with a maximum threshold of 63. This value should be equal to or greater than half the window size. If the count of logic 1s and logic 0s within the sampling window is not greater than the

threshold, the digital filter will maintain the previous output value.

- The TIM1_SLIDFPSC.SLIDFPSC[15:0] register determines the sampling rate of the corresponding digital filter. The filter FIFO captures a sample value from the input at each sampling clock.
- If the digital filter is disabled, the filter input is directly output.

12.5.17 Bidirectional Break

GTIMx(x=8-10) has bidirectional break I/O function.

Application support :

● A board-level global break signal, which can send a fault signal to an external MCU or gate driver through a unique IO pin that is both an input and output.

● When multiple internal and external break sources need to be merged, they are connected together through "OR" operation to generate a single break event.

The tim_brk input can be configured as bidirectional mode by controlling the BRKBID in the TIMx_BDTR register. The BRKBID can be locked in read-only mode (at LOCK level 1 or above) using the LOCK bit in the TIMx_BKDT register.

Bidirectional mode is available for the tim_brk input, requiring the I/O configuration to be in open-drain mode with a low-level polarity (through TIMx_AF1.IOMBRKP, TIMx_BKDT.BKP bits). Any break request from the system (such as CSS), on-chip peripherals, or break input will force the break input to a low level to indicate a fault event. For safety reasons, if the polarity bit is not set correctly, the bidirectional mode will be suppressed (for example, if set to high-level effective, bidirectional mode will not take effect).

A software break event (TIMx_EVTGEN.BGN) will also force the break IO to "0" to indicate to external devices that the timer has entered the break state. However, this is only effective when the break is enabled (TIMx_BKDT.BKEN=1). When a software break event occurs (TIMx_BKDT.BKEN=0), the output is placed in a safe state, and the break flag is set. However, this has no effect on the TIMx_BKIN I/O.

The safety release mechanism prevents the system from being completely locked (breaking triggered by a low level on the break input, thereby forcing a low level on the same input).

When the TIMx_BKDT.BRKDSRM bit is set to 1, the brake output will be released to clear a fault signal and provide the possibility for system restart.

At any time, the brake protection circuit cannot be disabled：

●The break input path is always active: even if the TIMx_BKDT.BRKDSRM bit is set and the open-drain control is released, the break event remains active. This prevents the PWM output from restarting when break conditions are present.

●The TIMx_BKDT.BRKDSRM bit cannot release the brake protection as long as the output is enabled (TIMx_BKDT.MOEN bit is set).

Table 12-11 Conditions For Releasing The Break Protection State

MOEN	BRKBID	BRKDSRM	Break Protection State

0	0	X	Protection
0	1	0	Protection
0	1	1	Release Protection
1	X	X	Protection

Enable and Re-enable the break circuit :

By default (in input or bidirectional mode), the break circuit is in standby mode (peripheral reset configuration).

After a break event, the following procedure must be followed to re-enable the protection:

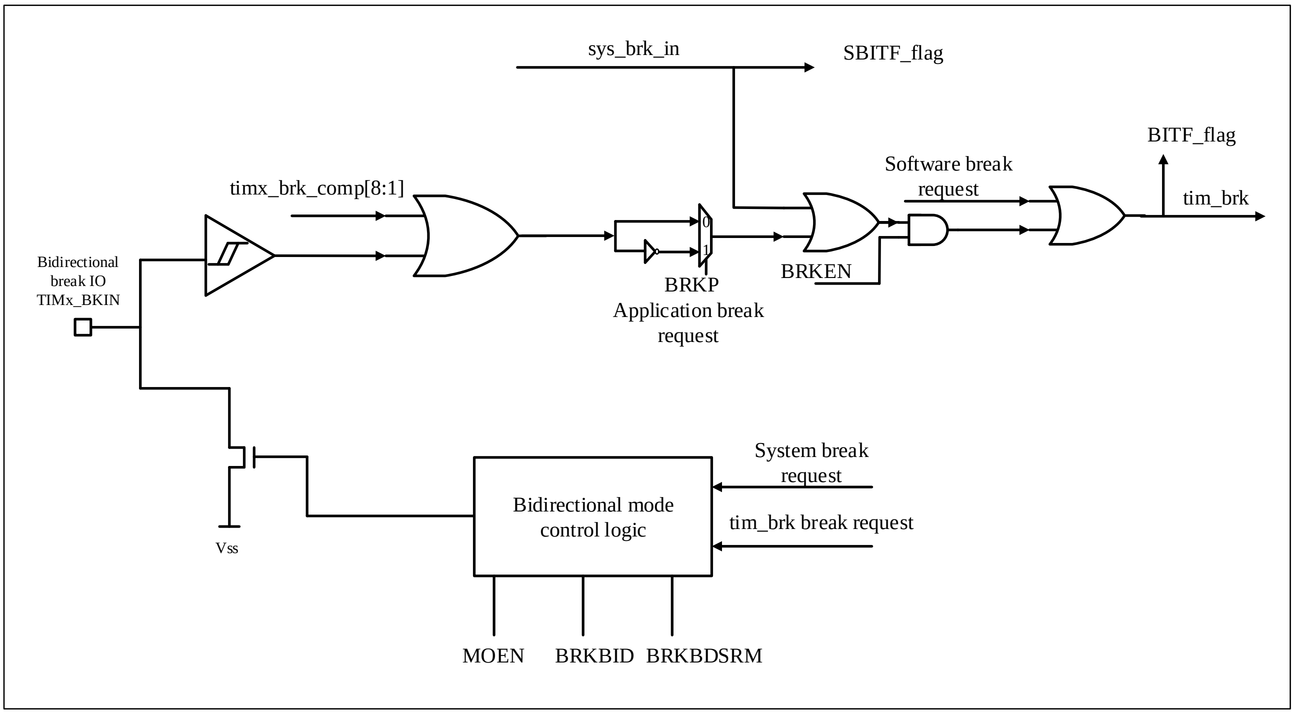
● The TIMx_BKDT.BRKDSRM bit must be set to release the output control

● The software must wait for the system break condition to disappear and clear the TIMx_STS.SBIF status flag (or have the system clear it before re-enabling)

● The software must poll the TIMx_BKDT.BRKDSRM bit until it is cleared by hardware (when the application's break condition disappears)

From this point, the break circuit is in standby and active state, and the TIMx_BKDT.MOEN bit can be set to re-enable the PWM output.

Figure 12-34 Output Redirection



12.5.18 Debug Mode

When the microcontroller is in debug mode (the Cortex^{TM}-M4F core halted), depending on the

DBG_CTRL.TIMx_STOP configuration in the DBG module, the TIMx counter can either continue to work normally or stop. For more details, see Section37.3.2.

12.5.19 Synchronization of GTIMx and External Trigger

The timer can be synchronized through triggers in the slave mode(reset、trigger and gated).

12.5.19.1 Slave Mode: Reset Mode

In the reset mode, the trigger event can reset the counter and prescaler. It updates the preload registers TIMx_AR, TIMx_CCDATx, and generates an update event UEV (TIMx_CTRL1.UPRS=0).

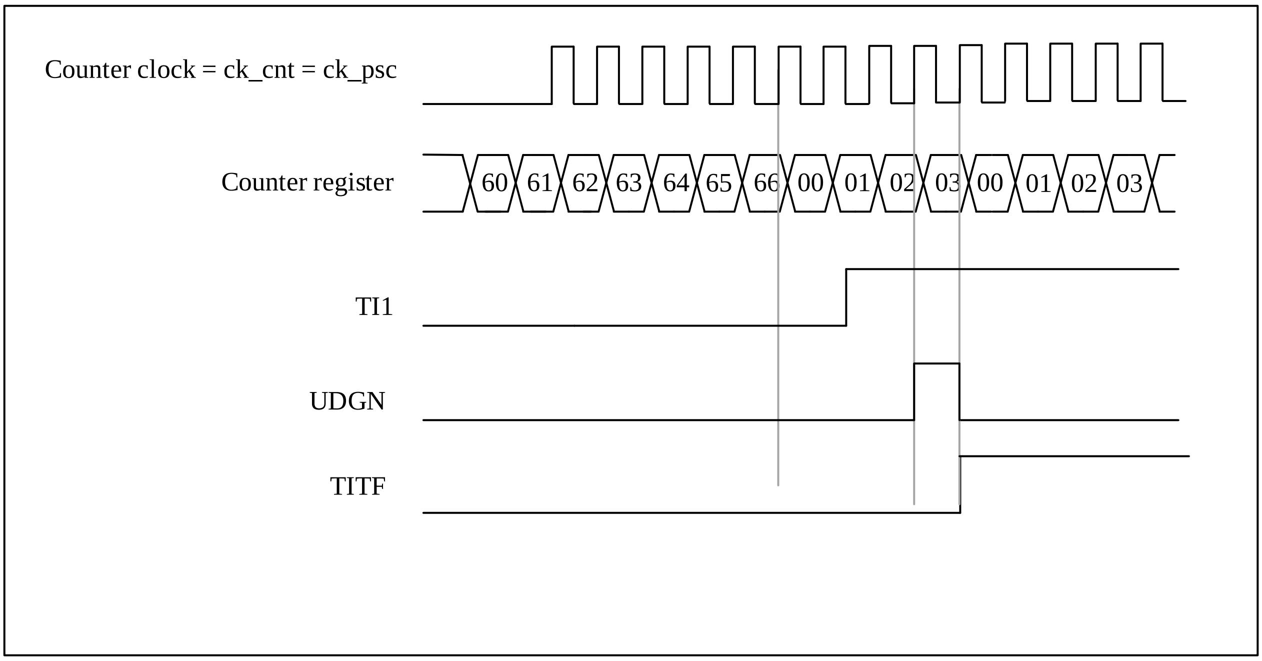
Here is an example of the reset mode:

4. Channel 1 is configured to detect the rising edge of TI1 input (TIMx_CCMOD1.CC1SEL=01, TIMx_CCEN.CC1P=0) ;
5. Select the slave mode as reset mode (TIMx_SMCTRL.SMSEL=0100), and choose TI1 as the trigger input selection (TIMx_SMCTRL.TSEL=101) ;
6. Start the counter (TIMx_CTRL1.CNTEN=1).

After starting the timer, when a rising edge is detected by TI1, the counter resets and restarts counting, and the trigger flag is set (TIMx_STS.TITF=1) ;

The delay between the rising edge of TI1 and the actual counter reset is caused by the synchronization circuit of the TI1 input.

Figure 12-35 Control Circuit in the Reset Mode



12.5.19.2 Slave Mode: Trigger Mode

In trigger mode, the trigger event (rising edge/falling edge) on the input port can trigger the counter to start counting.

Here is an example of the trigger mode:

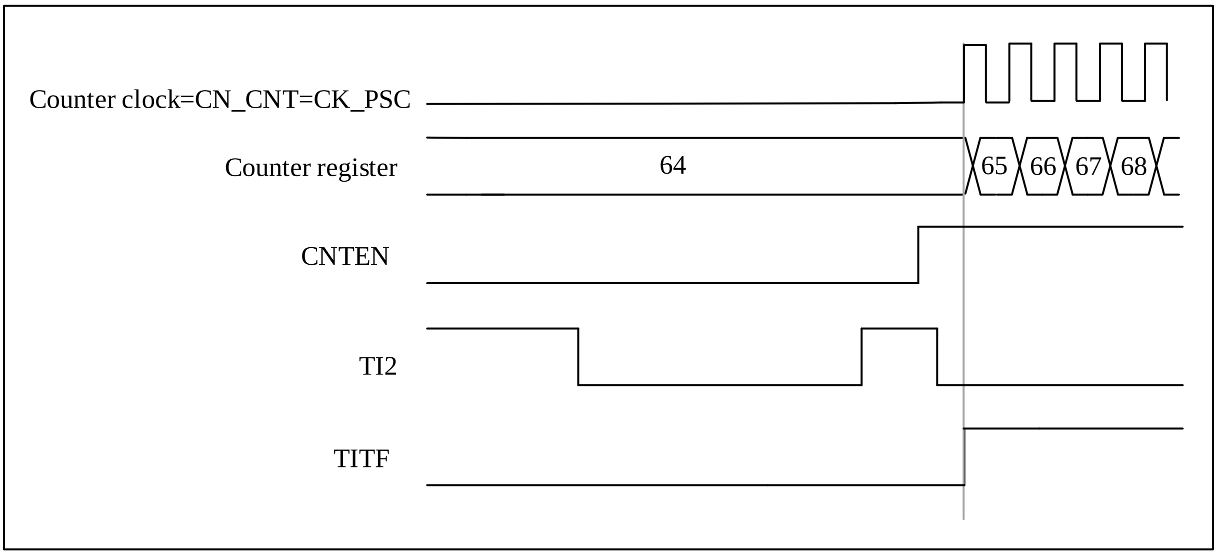
3. Channel 2 is configured as input, detect the rising edge of TI2 (TIMx_CCMOD1.CC2SEL=01, TIMx_CCEN.CC2P=0) ;

4. Select the slave mode as trigger mode (TIMx_SMCTRL.SMSEL=0110), and select TI2 as the trigger input (TIMx_SMCTRL.TSEL=110) ;

When a rising edge is detected by TI2, the counter starts counting, and the trigger flag is set (TIMx_STS.TITF=1) ;

The delay between the rising edge of TI2 and the actual start of the counter is caused by the synchronization circuit of the TI2 input.

Figure 12-36 Control Circuit in Trigger Mode



12.5.19.3 Slave Mode: Gated Mode

In gated mode, the logic level polarity of the input port can determine whether the counter counts.

Here is an example of gated mode :

4. Channel 1 is configured to detect low level input on TI1 (TIMx_CCMOD1.CC1SEL=01, TIMx_CCEN.CC1P=1) ;

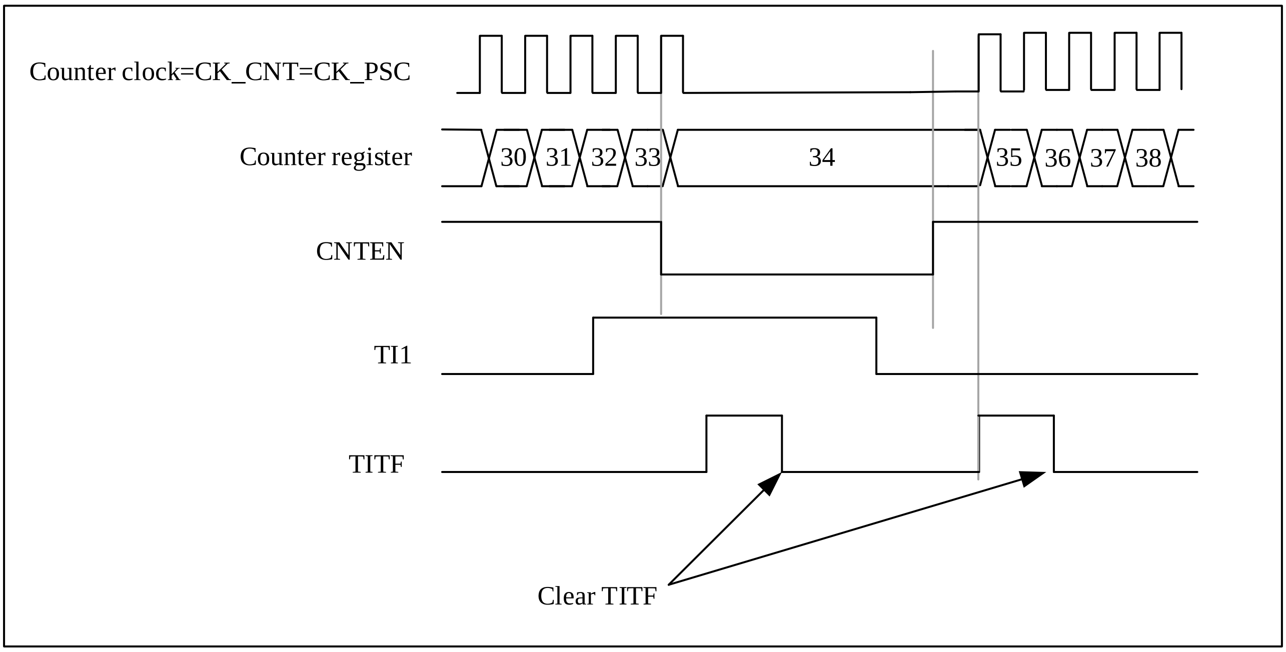
5. Select the slave mode as gated mode (TIMx_SMCTRL.SMSEL=0101), and choose TI1 as the TRGI (TIMx_SMCTRL.TSEL=101) ;

6. Start the counter (TIMx_CTRL1.CNTEN=1) ;

When TI1 detects a transition from low to high level, the counter stops counting. When TI1 detects a transition from high to low level, the counter starts counting. The trigger flag is set when counting starts or stops (TIMx_STS.TITF=1).

The delay between the rising edge of TI1 and the actual stop of the counter is caused by the synchronization circuit of the TI1 input.

Figure 12-37 Control Circuit in Gated Mode



12.5.19.4 Slave Mode: Trigger mode + External Clock Mode 2

In reset mode, trigger mode, and gated mode, the counter clock can be selected as External Clock Mode 2, with the ETR signal as the external clock source input. In this case, the trigger selection should choose a non-ETRF option (TIMx_SMCTRL.TSEL=111).

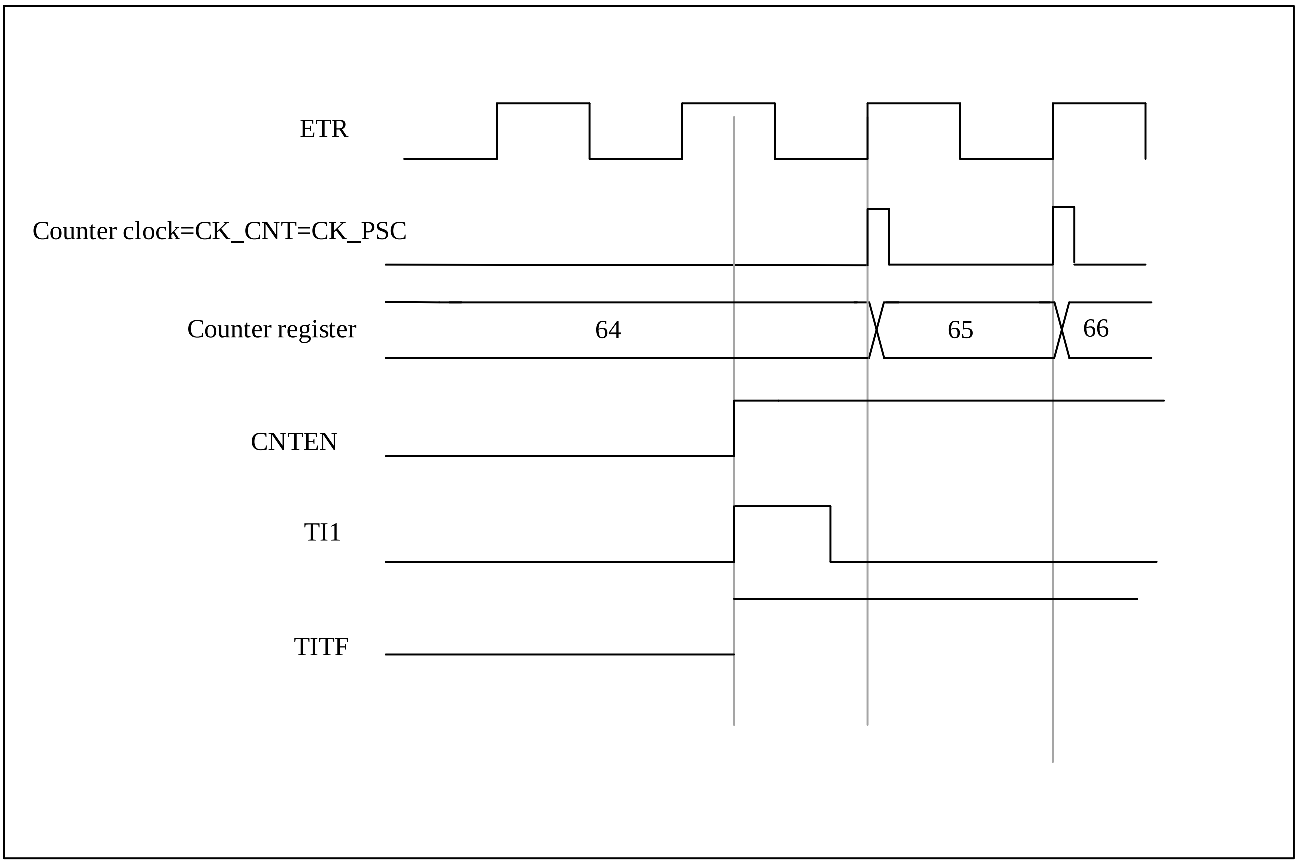
This is an example：

3. Channel 1 is configured to detect the rising edge of TI1 as input (TIMx_CCMOD1.CC1SEL=01, TIMx_CCEN.CC1P=0) ;

4. Enable External Clock Mode 2 (TIMx_SMCTRL.EXCEN=1), select rising edge for external trigger polarity (TIMx_SMCTRL.EXTP=0), set trigger mode as slave mode (TIMx_SMCTRL.SMSEL=0110), and choose TI1 as the TRGI (TIMx_SMCTRL.TSEL=101) ;

When TI1 detects a rising edge, the counter starts counting on the rising edge of ETR and sets the trigger flag (TIMx_STS.TITF=1) ;

Figure 12-38 Control circuit in External Clock Mode 2 + Trigger Mode



12.5.19.5 Slave Mode: Combined Reset + Trigger Mode

In this case, the rising edge of the selected trigger input (TRGI) will reinitialize the counter, update the generation of registers, and start the counter.

This mode is used for one pulse mode.

The selected event on the input port resets and enables the counter.

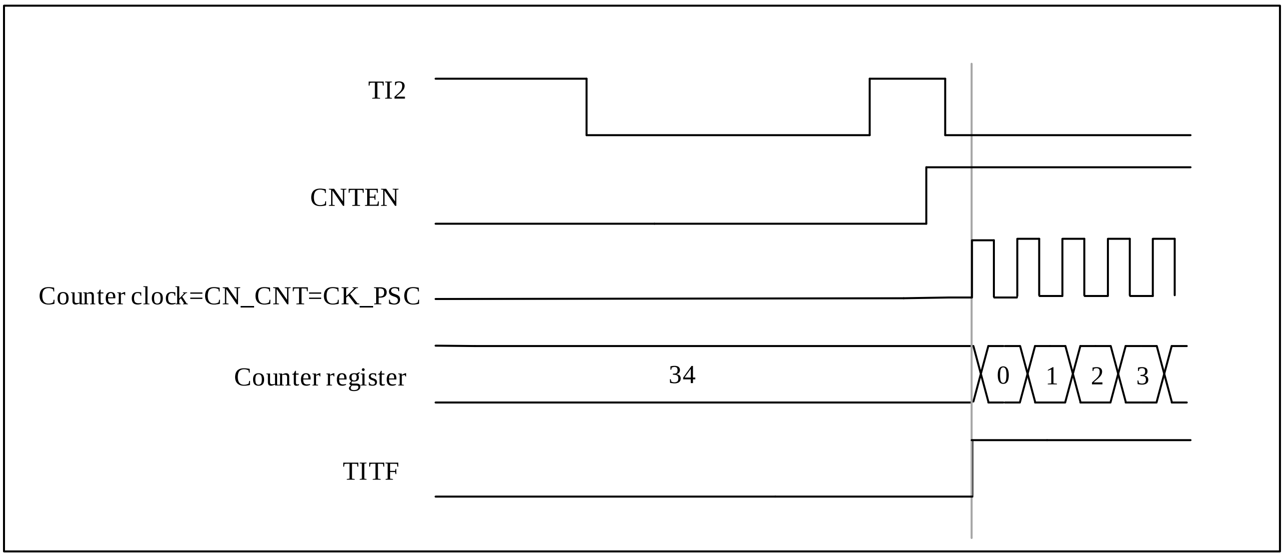
In the following example, the counter resets on the rising edge of the TI2 input and starts counting up.

- Configure channel 2 to detect the rising edge of TI2. Set the input filter bandwidth (in this case, no filter is needed, keep TIMx_CCMOD1.IC2F=0000). Capture prescaler is not used in the trigger operation, so no configuration is needed. The TIMx_CCMOD1.CC2SEL bit is only used to select the input capture source, set TIMx_CCMOD1.CC2SEL=01. Set TIMx_CCEN.CC2P=1 to determine the polarity (detect only low level).
- Set TIMx_SMCTRL.SMSEL=1110 to configure the timer for combined reset + trigger mode； Set TIMx_SMCTRL.TSEL=110 to select TI2 as the input source.

When a rising edge occurs on TI2, the counter starts counting under the internal clock drive and sets the TITF flag。

The delay between the rising edge of TI2 and the start of counting by the counter depends on the synchronization circuit of the TI2 input terminal。

Figure 12-39 Control Circuit in Combined Reset + Trigger Mode



12.5.19.6 Slave Mode : Combined Gated + Reset Mode

When the trigger input (TRGI) is high level, the counter clock is enabled. Once the trigger goes low level, the counter will stop and be reset. The start and stop of the counter are both controlled.

This mode can detect out-of-range PWM signals (duty cycle exceeding the maximum expected value).

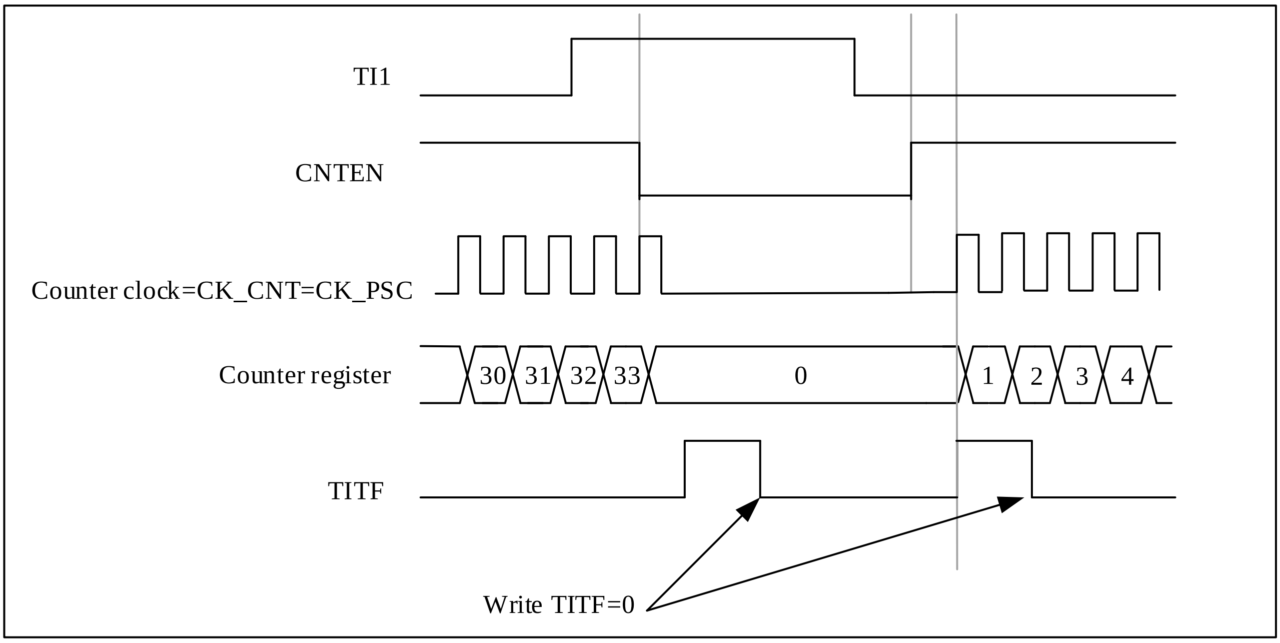
In the following example, the counter only counts up when TI1 is low level, and stops and resets when TI1 becomes high level :

- Configure channel 1 to detect low level on TI1. Set the input filter bandwidth (in this case, no filter is needed, so keep TIMx_CCMOD1.IC1F=0000). Capture prescaler is not used in the trigger operation, so no configuration is needed. The TIMx_CCMOD1.CC1SEL bit is used to select the input capture source, set TIMx_CCMOD1.CC1SEL=01. Set TIMx_CCEN.CC1P=1 to determine the polarity (detect only low level).
- Set TIMx_SMCTRL.SMSEL=1101 to configure the timer for gated + reset mode; Set TIMx_SMCTRL.TSEL=101 to select TI1 as the input source.
- Set CNTEN=1 in the TIMx_CTRL1 register to start the counter. In gated + reset mode, if CNTEN=0, the counter cannot start, regardless of the trigger input level.

As long as TI1 is low, the counter starts counting based on the internal clock; once TI1 goes high, the counting stops. The TITF flag in TIMx_STS is set whenever the counter starts or stops.

The delay between the rising edge of TI1 and the actual stop of the counter depends on the synchronization circuit of the TI1 input terminal.

Figure 12-40 Control Circuit in Combined Gated + Reset Mode



12.5.20 Timer Synchronization

All TIM timers are internally connected for timer synchronization or chaining. Refer to Section 11.5.14.

12.5.21 Trigger ADC

The timer can generate ADC trigger events through various internal signals, such as reset, enable, or compare events, and can also generate pulse triggers issued by internal edge detectors. The selection of issuing trigger signals on the internal line redirected to ADC TRGO can be done by selecting the MMSEL[3:0] bits in the TIMx_CTRL2 register.

12.5.22 Generating Six-Step PWM Output

To simultaneously modify the configuration of all channels, you can pre-set the configuration for the next step (preload bits for OCxMD, CCxEN, and CCxNEN). When a COM commutation event occurs, the preload bits for OCxMD, CCxEN, and CCxNEN are transferred to the shadow register bits.

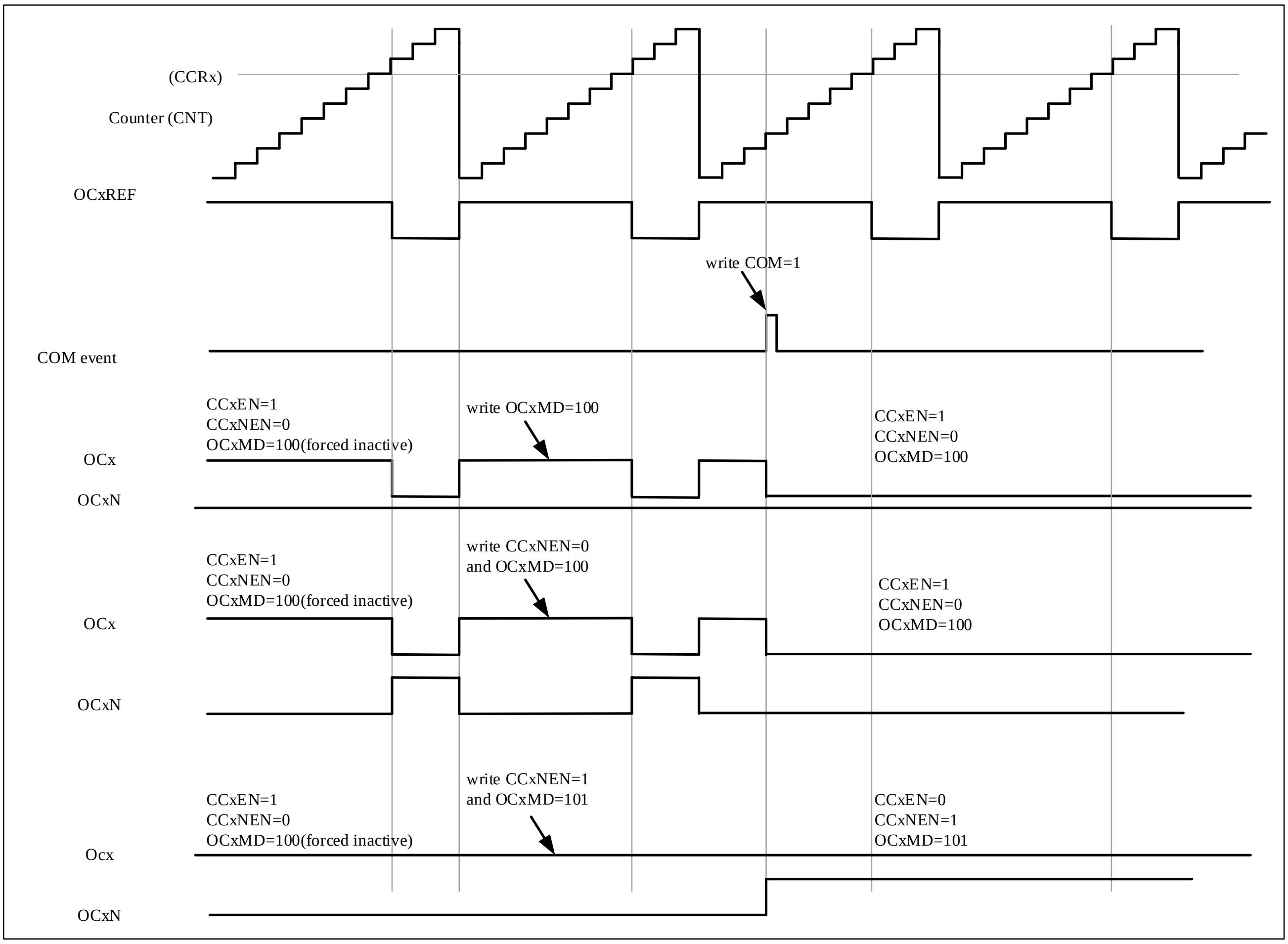
Method of generating COM commutation events :

1. Software set TIMx_EVTGEN.CCUDGN ;
2. Generated by hardware on the rising edge of TRGI ;

When a COM commutation event occurs, the TIMx_STS.COMITF flag will be set, enabling the interrupt (TIMx_DINTEN.COMIEN) to generate an interrupt, and enabling the DMA request (TIMx_DINTEN.COMDEN) will generate a DMA request.

The following diagram shows the timing diagram of the output of OCx and OCxN when a COM commutation event occurs under three different configurations :

Figure 12-41 Generating Six-Step PWM Using COM as Example (OSSR=1)



12.5.23 Encoder Interface Mode

12.5.23.1 Quadrature Encoder Mode

The encoder uses two inputs, TI1 and TI2 as interfaces, and the counter counts each edge change on TI1FP1 or TI2FP2. The counting direction is automatically controlled by the hardware TIMx_CTRL1.DIR. There are a total of five modes for quadrature encoder counting :

- Encoder Mode 1: The counter counts only on the edge of TI1, TIMx_SMCTRL.SMSEL = ‘0001’;
- Encoder Mode 2: The counter counts only on the edge of TI2, TIMx_SMCTRL.SMSEL = ‘0010’;
- Encoder Mode 3: The counter counts on the edges of both TI1 and TI2 simultaneously, TIMx_SMCTRL.SMSEL = ‘0011’;
- Encoder Mode 4: When T2 is high level, the counter only counts on the edge of TI1, TIMx_SMCTRL.SMSEL = ‘1001’;
- Encoder Mode 5: When T1 is high, the counter only counts on the edge of TI2, TIMx_SMCTRL.SMSEL = ‘1010’;

The encoder interface is equivalent to using an external clock with direction selection, where the counter continuously counts between 0 and the automatic reload value (TIMx_AR.AR [15:0]). Therefore, the automatic reload register

TIMx_AR needs to be configured in advance.

Note: Encoder mode and External Clock Mode 2 are not compatible and cannot be selected simultaneously.

The relationship between counting direction and encoder signals is shown in the following table：

Table 12-12 The Relationship Between Counting Direction And Encoder Signals (CC1P=CC2P=0)

Valid Edges	**SMSEL[3:0]**	**The level of the relative signal (TI1FP1 for TI2, TI2FP2 for TI1)**	**TI1FP1 Signal**		**TI2FP2 Signal**	
			Rising	**Falling**	**Rising**	**Falling**
Counting only on TI1	0001	high	Counting down	Counting up	Don’t count	Don’t count
		low	Counting up	Counting down	Don’t count	Don’t count
Counting only on TI2	0010	high	Don’t count	Don’t count	Counting up	Counting down
		low	Don’t count	Don’t count	Counting down	Counting up
Counting on both TI1 and TI2	0011	high	Counting down	Counting up	Counting up	Counting down
		low	Counting up	Counting down	Counting down	Counting up
Counting only on TI1 and T2 is high	1001	high	Counting down	Counting up	Don’t count	Don’t count
		low	Don’t count	Don’t count	Don’t count	Don’t count
Counting only on TI2 and T1 is high	1010	high	Don’t count	Don’t count	Counting up	Counting down
		low	Don’t count	Don’t count	Don’t count	Don’t count

The change in the counter value when the counter is in each mode is as follows：

Figure 12-42 The Encoder Counts Only On TI1

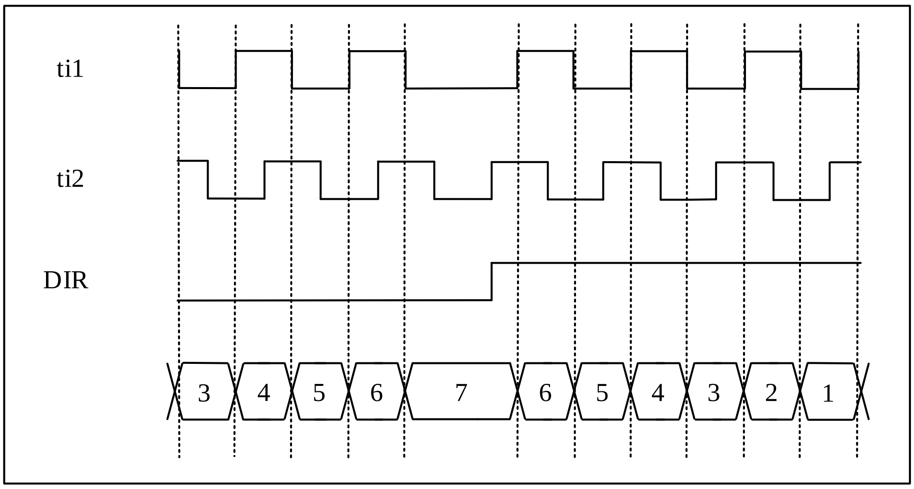


Figure 12-43 The Encoder Counts Only On TI2

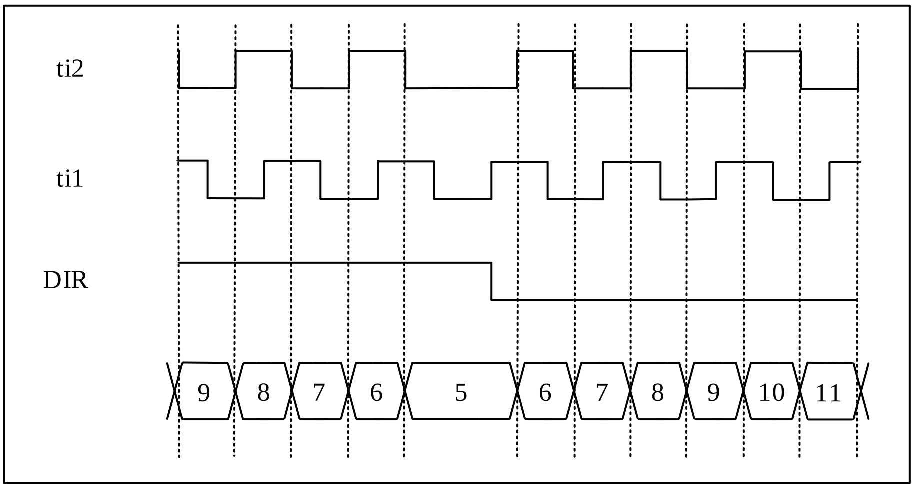


Figure 12-44 The Encoder Counts On Both TI1 and TI2

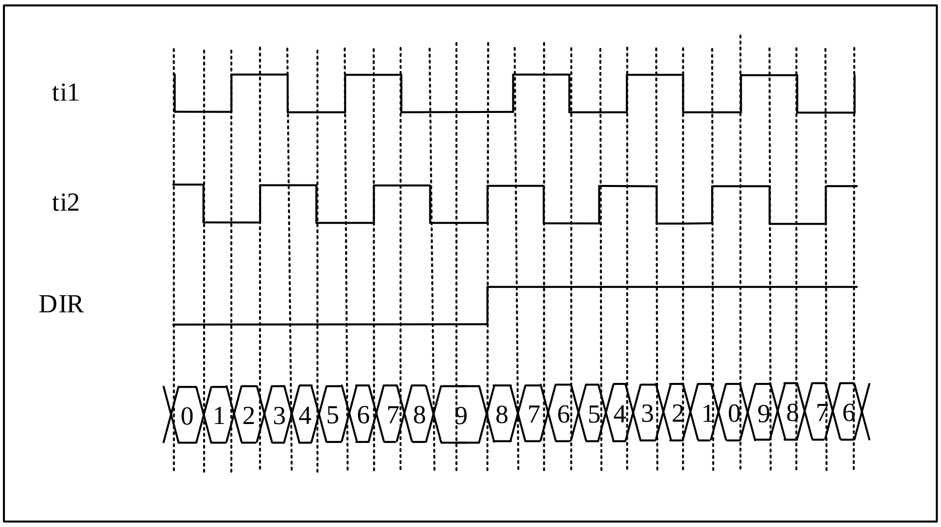


Figure 12-45 When T2 is high level, the counter counts only on TI1

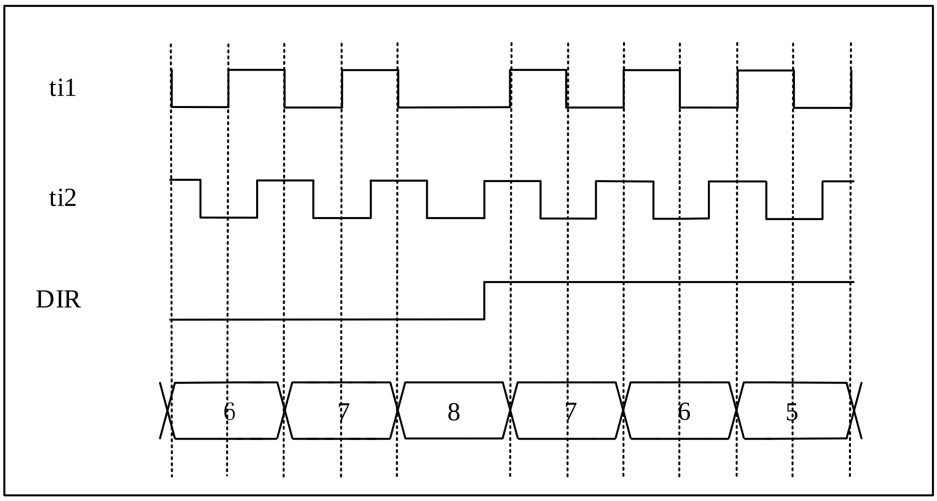
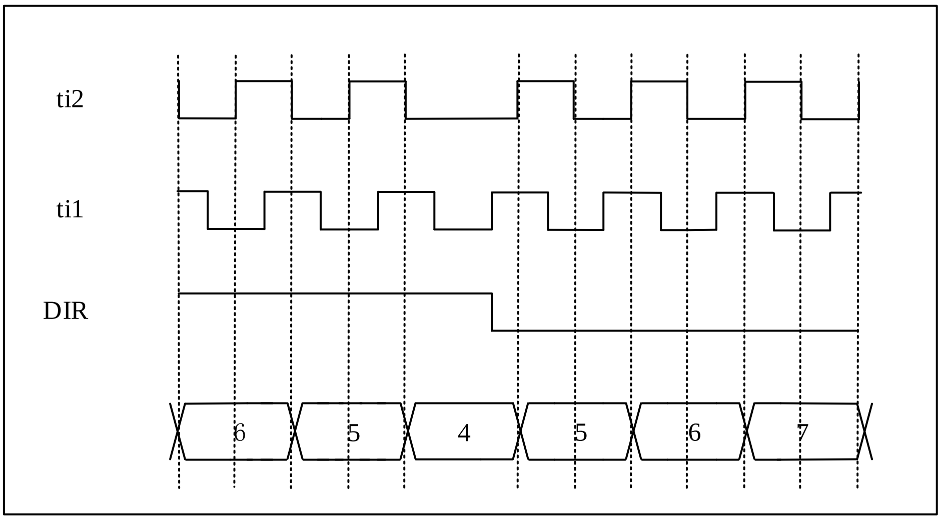


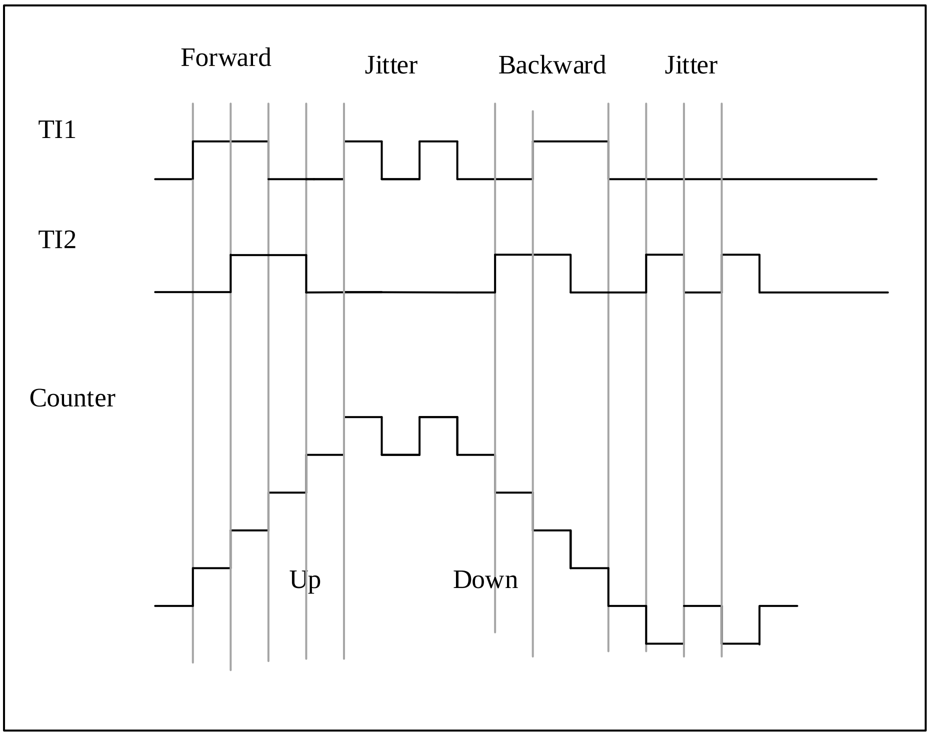
Figure 12-46 When T1 is high level, the counter counts only on TI2



Here is an example of an encoder that has selected double-edge triggering to suppress input jitter：

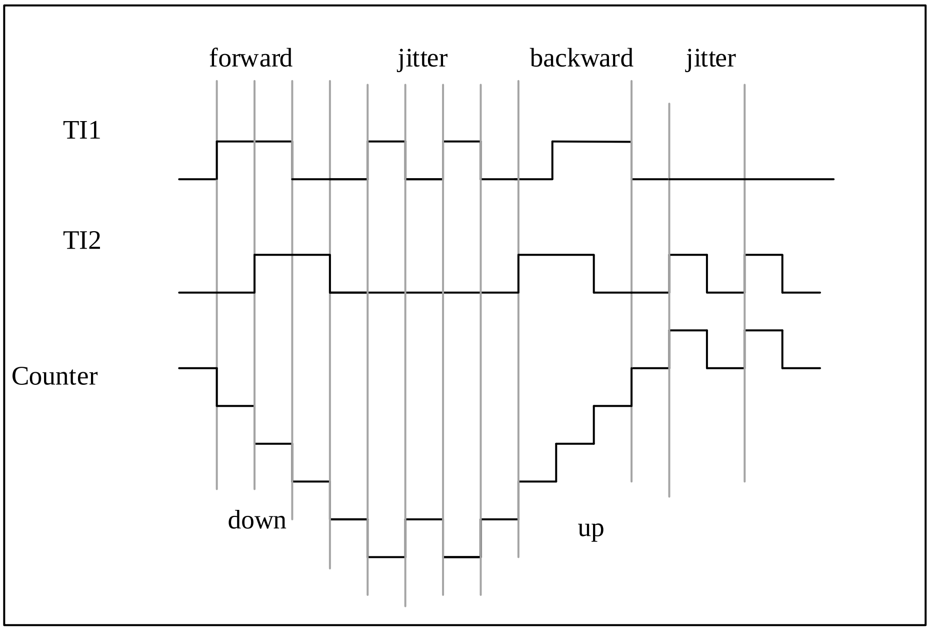
1. IC1FP1 is mapped to TI1 (TIMx_CCMOD1.CC1SEL=‘01’), IC1FP1 is not inverted (TIMx_CCEN.CC1P=‘0’)；
2. IC1FP2 is mapped to TI2 (TIMx_CCMOD2.CC2SEL=‘01’), IC2FP2 is not inverted (TIMx_CCEN.CC2P=‘0’)；
3. Inputs are valid both rising and falling edges (TIMx_SMCTRL.SMSEL=‘0011’) ;
4. Start the counter TIMx_CTRL1.CNTEN=‘1’；

Figure 12-47 Example of counter operation in encoder mode



The following diagram shows an example of counter behavior when IC1FP1 polarity is inverted (CC1P = '1', other configurations are the same as above)

Figure 12-48 Example of an inverted encoder interface mode with IC1FP1 inverted



12.5.23.2 Pulse Level Encoding Mode

In the pulse level encoding mode, the clock is provided on a single line on TI2, while the counting direction is provided by the TI1 input.

This mode is enabled through the SMSEL[3:0] in the TIMx_SMCTRL register, as detailed below.

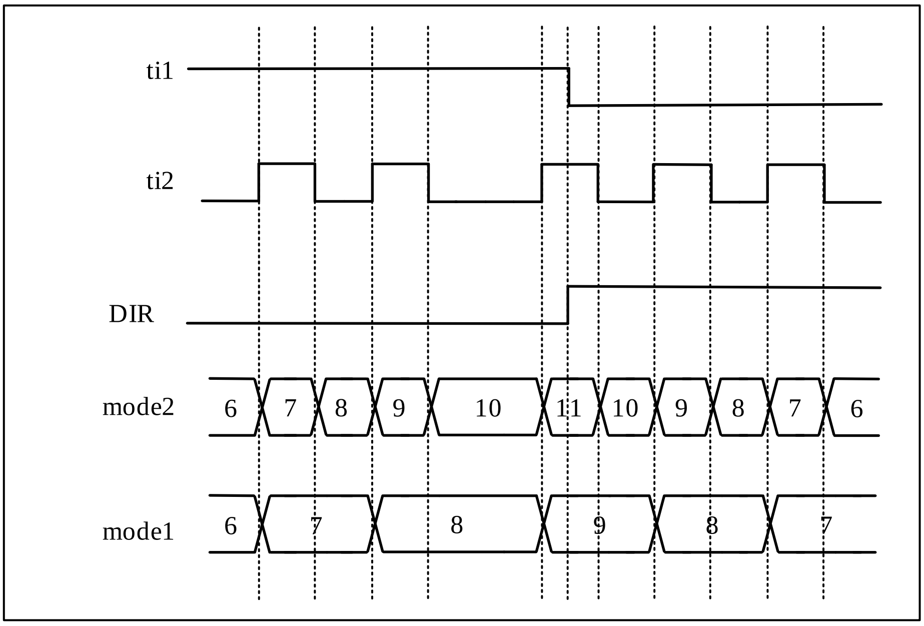
1011：In Pulse Level Encoding Mode 2, the counter is updated on both the rising and falling edges of the clock.

1100：In Pulse Level Encoding Mode 1, the counter is updated on a single clock edge based on the value of CC2P. CC2P = 0 corresponds to counting on the rising edge, CC2P = 1 corresponds to counting on the falling edge.

The polarity of the direction signal of TI1 is set by the CC1P bit. When CC2P = 0, the counter counts up when TI1 is high and counts down when TI1 is low; when CC1P = 1, the counter counts up when TI1 is low and counts down when TI1 is high.

The following diagram is an example with CC1P=CC2P=0 :

Figure 12-49 Pulse Level Encoding Mode (CC1P=CC2P=0)



12.5.23.3 Dual Pulse Encoding Mode

In the Dual Pulse Encoding Mode, the clock is provided on two lines, with only one line provided at a time depending on the direction. This results in one clock line for counting up and one clock line for counting down.

This mode is enabled through the SMSEL[3:0] field in the TIMx_SMCTRL register, as detailed below.

- 1000 : Dual Pulse Encoding Mode 2, the counter is updated on both the rising and falling edges of either of the two clock lines. The CC1P and CC2P bits encode the clock idle state. CCxP=0 corresponds to a high-level idle state, and CCxP=1 corresponds to a low-level idle state.
- 1111：Dual Pulse Encoding Mode 1, the counter updates on a single clock edge based on the values of the CC1P and CC2P bits. CCxP=0 corresponds to the falling edge and high-level state, CCxP=1 corresponds to the rising edge and low-level state.

The table below describes the relationship between counting direction and encoder signals and polarity settings.

Table 12-13 The relationship between counting direction and encoder signals and polarity settings

Dual Pulse Encoding Mode	SMSEL[3:0]	The level of the relative signal(TI1FP1 for TI2, TI2FP2 for TI1)	TI1FP1 Signal		TI2FP2 Signal	
			Rising	falling	Rising	falling
Mode 2 CCxP=0	1000	High	Counting Down	Counting Down	Counting Up	Counting Up
		Low	Don’t Count	Don’t Count	Don’t Count	Don’t Count
Mode 2 CCxP=1	1000	High	Don’t Count	Don’t Count	Don’t Count	Don’t Count
		Low	Counting Down	Counting Down	Counting Up	Counting Up
Mode 1 CCxP=0	1111	High	Don’t Count	Counting Down	Don’t Count	Counting Up
		Low	Don’t Count	Don’t Count	Don’t Count	Don’t Count
Mode 1 CCxP=1	1111	High	Don’t Count	Don’t Count	Don’t Count	Don’t Count
		Low	Counting Down	Don’t Count	Counting Up	Don’t Count

The diagram below shows the counting method of the counter in the Dual Pulse Encoding Mode

Figure 12-50 Dual Pulse Encoding Mode (CC1P=CC2P=0)

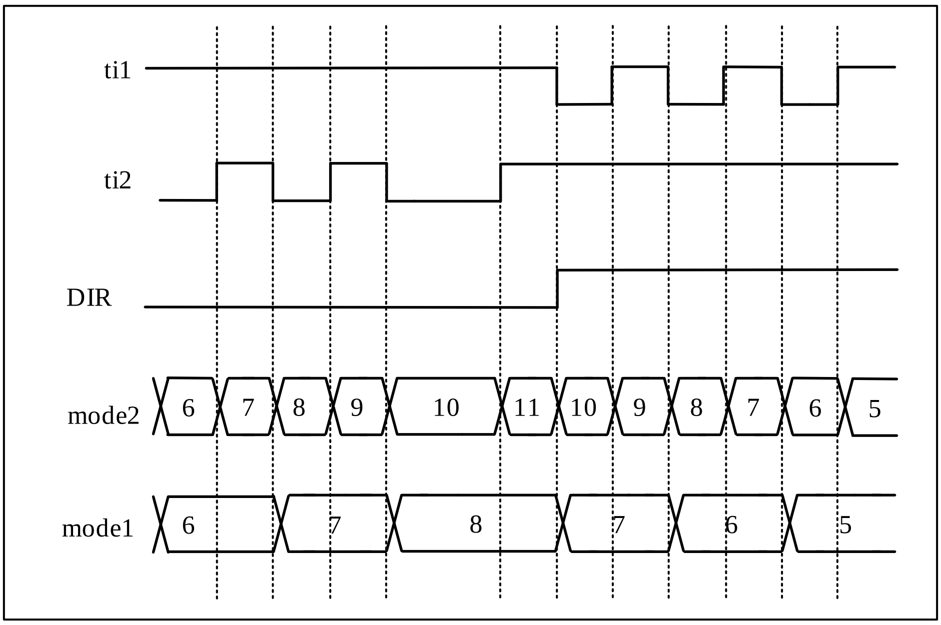
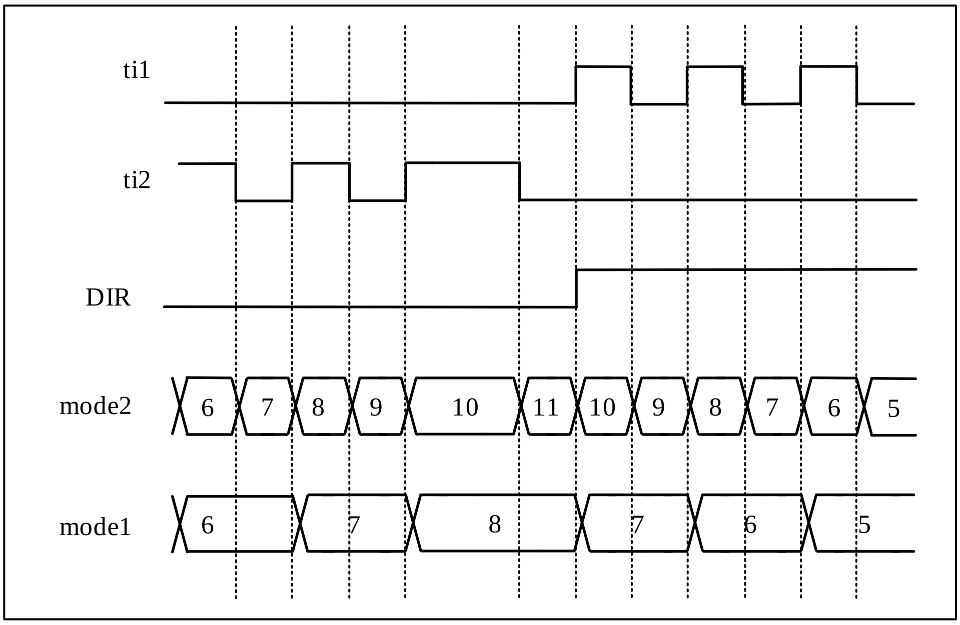


Figure 12-51 Dual Pulse Encoding Mode (CC1P = CC2P = 1)



12.5.24 Interface with Hall Sensors

See Section10.5.22.

12.5.25 UDITF Bit Remapping

The UDITFREMAP bit in the TIMx_CTRL1 register forces the update interrupt flag UDITF to be continuously copied to bit 31 (TIMx_CNT[31]) of the timer counter register, i.e., the UDITFCPY bit. This allows for automatic reading of the counter value and the flip of the UDITFCPY flag. In specific scenarios, this can avoid race conditions between background tasks (counter reads) and interrupts (update interrupts), simplifying calculations.

There is no delay between the enablement of the UDITF and UDITFCPY flags.

12.6 GTIMx(x=8,9,10) Register Description

For abbreviations used in the register descriptions, please refer to Section1.1.

These peripheral registers can be operated as half-word (16-bit) or one word (32-bit).

12.6.1 Control Register 1（TIMx_CTRL1）

Offset address : 0x00

Reset value : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved							UDITF REMAP	ASYMMETRIC	Reserved						
							rw	rw							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SRAMECCERREN	FLASHECCERREN	CLRSEL	SRAMPARERREN	PBKPEN	LBKPEN	ARPEN	ONEPM	CLKD[1:0]		UPDIS	UPRS	CAMSEL[1:0]		DIR	CNTEN
rw	rw	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw		rw	rw

Bit Field	Name	Description
31:25	Reserved	Reserved, the reset value must be maintained.
24	UDITFREMAP	UDITF status bit remapping 0: No remapping. The UDITF status bit is not copied to bit 31 of the TIMx_CNT register. 1: Enable remapping. Copy the status bit UDITF to bit 31 of the TIMx_CNT register.
23	ASYMMETRIC	Asymmetric mode enable in center-aligned mode 0: Disable 1: Enable (When TIMx_CTRL1.CAMSEL[1:0] is non-zero, each channel is compared with CCDATx when counting up and with CCDDATx when counting down)
22:16	Reserved	Reserved, the reset value must be maintained.
15	SMECCERREN	SRAM ECC error as brk Enable 0: Disable 1: Enable *Note: This bit is set to 1 by software and cleared to 0 by system reset*
14	FLECCERREN	FLASH ECC error as brk Enable 0: Disable 1: Enable *Note: This bit can be cleared to 0 by system reset and power-on reset, but not by module reset.*
13	CLRSEL	OcxRef clear selection 0: Select the external Ocxclr (TIMx_ETR) signal, specific selection see TIMx_INSEL.ETRS. 1: Select the internal Ocxclr (tim_ocref_clr) signal, specific selection see TIMx_INSEL.CLRS
12	SMPARERREN	SRAM parity error as brk Enable

Bit Field	Name	Description
		0: Disable 1: Enable *Note: This bit can be cleared to 0 by system reset and power-on reset, but not by module reset.*
11	PBKPEN	PVD as brk Enable 0: Disable 1: Enable *Note: This bit can be cleared to 0 by system reset and power-on reset, but not by module reset.*
10	LBKPEN	LockUp as brk Enable (Core Hardfault) 0: Disable 1: Enable *Note: This bit can be cleared to 0 by system reset and power-on reset, but not by module reset.*
9	ARPEN	Auto-reload preload enable 0: Disable the shadow register of the TIMx_AR register 1: Enabling the shadow register of the TIMx_AR register
8	ONEPM	One pulse mode 0: Disable one-pulse mode, the counter counts are not affected when an update event occurs. 1: Enable one-pulse mode, the counter stops counting when the next update event occurs
7:6	CLKD[1:0]	Clock division CLKD[1:0] indicates the division ratio between CK_INT (timer clock) and t_{DTS} (clock used for dead-time generator and digital filters (ETR, TIx)) 00: $t_{DTS} = t_{CK\_INT}$. 01: $t_{DTS} = 2 \times t_{CK\_INT}$ 10: $t_{DTS} = 4 \times t_{CK\_INT}$ 11: Reserved, do not use this configuration
5	UPDIS	Update disable This bit is used to enable/disable the Update event (UEV) events generation by software. 0: Enable. UEV will be generated if one of following condition been fulfilled ─ Counter overflow/underflow – The TIMx_EVTGEN.UDGN bit is set ─ Update generation from the slave mode controller Shadow registers will update with preload value. 1: Disabled. No update event is generated, and the shadow registers (AR, PSC, and CCDATx) keep their values. If the TIMx_EVTGEN.UDGN bit is set or a hardware reset is issued by the slave mode controller, the counter and prescaler are reinitialized.
4	UPRS	Update request source This bit is used to select the UEV event sources by software. 0: If update interrupt or DMA request is enabled, any of the following events will generate an update interrupt or DMA request： ─ Counter overflow/underflow – TIMx_EVTGEN.UDGN bit is set – Update generation from the slave mode controller 1: If update interrupt or DMA request is enabled, only counter overflow/underflow will

Bit Field	Name	Description
		generate update interrupt or DMA request
3:2	CAMSEL[1:0]	Center-aligned mode selection 00: Edge-aligned mode. TIMx_CTRL1.DIR specifies up-counting or down-counting。 01: Center-aligned mode 1. The counter counts in center-aligned mode, and the output compare interrupt flag bit is set to 1 when down-counting。 10: Center-aligned mode 2. The counter counts in center-aligned mode, and the output compare interrupt flag bit is set to 1 when up-counting 11: Center-aligned mode 3. The counter counts in center-aligned mode, and the output compare interrupt flag bit is set to 1 when up-counting or down-counting。 *Note: Switching from edge-aligned mode to center-aligned mode is not allowed when the counter is still enabled (TIMx_CTRL1.CNTEN = 1).*
1	DIR	Direction 0: Up-counting 1: Down-counting *Note: This bit is read-only when the counter is configured in center-aligned mode or encoder mode.*
0	CNTEN	Counter enable 0: Disable counter; 1: Enable counter *Note: external clock, gating mode and encoder mode can only work after TIMx_CTRL1.CNTEN bit is set in the software. Trigger mode can automatically set TIMx_CTRL1.CNTEN bit by hardware.*

12.6.2 Control Register 2（TIMx_CTRL2）

Offset address: 0x04

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved												TI1SEL	CCPCTL	CCDSEL	CCUSEL
												rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MMSEL[3:0]				Reserved			OI5	Reserved	OI4	Reserved	OI3	Reserved	OI2	OI1N	OI1
rw							rw		rw		rw		rw	rw	rw

Bit Field	Name	Description
31:20	Reserved	Reserved, the reset value must be maintained
19	TI1SEL	TI1 selection 0: TIMx_CH1 pin connected to TI1 input; 1: TIMx_CH1、TIMx_CH2 and TIMx_CH3 pins are XOR connected to the TI1 input.
18	CCPCTL	Capture/compare preloaded control 0: CCxEN, CCxNEN and OCxMD are not preload; 1: CCxEN, CCxNEN and OCxMD are preload; After setting this bit, they are only updated when the CCUDGN bit is set. *Note: This function only works for channels with complementary outputs*
17	CCDSEL	Capture/Compare DMA selection 0: When a CCx event occurs, a DMA request for CCx is sent； 1: When an update event occurs, a DMA request for CCx is sent.
16	CCUSEL	Capture/Compare control update selection 0: If the Capture/Compare control bits are preloaded (CCPCTL=1), they can only be updated by setting the CCUDGN bit ; 1: If the Capture/compare control bits are preloaded (CCPCTL=1), they can be updated by setting the CCUDGN bit or by a rising edge on TRGI. *Note: This bit only applies to channels with complementary outputs*
15:12	MMSEL[3:0]	Master Mode Selection These 4 bits are used to select the synchronization information (TRGO) sent to the slave timer in the master mode. Possible combinations are as follows: 0000: Reset –When the TIMx_EVTGEN.UDGN is set or a reset is generated by the slave mode controller, a TRGO pulse occurs. And in the latter case, the signal on TRGO is delayed compared to the actual reset. 0001: Enable - The TIMx_CTRL1.CNTEN bit is used as the trigger output (TRGO). Sometimes you need to start multiple timers at the same time or enable slave timer for a period of time. The counter enable signal is set when TIMx_CTRL1.CNTEN bit is set or the trigger input in gated mode is high。 When the counter enable signal is controlled by the trigger input, there is a delay on TRGO except if the master/slave mode is selected (see the description of the TIMx_SMCTRL.MSMD bit). 0010: Update - The update event is selected as the trigger output (TRGO). For example, a master timer clock can be used as a slave timer prescaler 0011: Compare pulse - Triggers the output to send a positive pulse (TRGO) when the TIMx_STS.CC1ITF is to be set (even if it is already high), when a capture or a comparison succeeds. 0100: Compare - OC1REF signal is used as the trigger output (TRGO). 0101: Compare – OC2REF signal is used as the trigger output (TRGO). 0110: Compare – OC3REF signal is used as the trigger output (TRGO). 0111: Compare – OC4REF signal is used as the trigger output (TRGO). 1xxx：Reserved.
11:9	Reserved	Reserved, the reset value must be maintained

Bit Field	Name	Description
8	OI5	Output Idle State 5 (OC5 output). See bit OI1.
7	Reserved	Reserved, the reset value must be maintained
6	OI4	Output Idle State 4 (OC4 output). See bit OI1.
5	Reserved	Reserved, the reset value must be maintained
4	OI3	Output Idle State 3 (OC3 output). See bit OI1.
3	Reserved	Reserved, the reset value must be maintained
2	OI2	Output Idle State 2 (OC2 output). See bit OI1.
1	OI1N	Output Idle state 1N 0: when MOEN=0, OC1N=0 after the dead time; 1: when MOEN=0, OC1N=1 after the dead time; *Note: Once the TIMx_BKDT.LCKCFG level 1, 2, or 3 has been set, this bit cannot be modified.*
0	OI1	Output Idle state 1 0: when MOEN=0, If OC1N is implemented, then OC1=0 after the dead time; 1: when MOEN=0, If OC1N is implemented, then OC1=1 after the dead time; *Note: Once the TIMx_BKDT.LCKCFG level 1, 2, or 3 has been set, this bit cannot be modified.*

12.6.3 Status Register（TIMx_STS）

Offset address: 0x08

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved										SBITF	Reserved	BITF	TITF	COMITF	UDITF
										rc_w0		rc_w0	rc_w0	rc_w0	rc_w0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				CC4OCF	CC3OCF	CC2OCF	CC1OCF	Reserved			CC5ITF	CC4ITF	CC3ITF	CC2ITF	CC1ITF
				rc_w0	rc_w0	rc_w0	rc_w0				rc_w0	rc_w0	rc_w0	rc_w0	rc_w0

Bit Field	Name	Description
31:22	Reserved	Reserved, the reset value must be maintained
21	SBITF	System break interrupt flag Once the system break input is active, it is set to '1' by hardware. If the system break input is inactive, this bit can be cleared to '0' by software. 0: No break event generated; 1: Valid level detected on system break input;
20	Reserved	Reserved, the reset value must be maintained
19	BITF	Break 1 interrupt flag Once break 1 input is active, it is set to '1' by hardware. If break 1 input is inactive, this bit can be cleared to '0' by software. 0: No break 1 event generated; 1: Valid level detected on break 1 input;

Bit Field	Name	Description
18	TITF	Trigger interrupt flag When a trigger event occurs (valid edge detected at the TRGI input terminal when the slave mode controller is in a mode other than gated mode, or any edge in gated mode), it is set to '1' by hardware. It can be cleared to '0' by software. 0: No trigger event generated; 1: Interrupt triggered waiting for response;
17	COMITF	COM interrupt flag Once a COM event occurs (when the capture/compare control bits: CCxEN, CCxNEN, OCxMD have been updated), this bit is set to '1' by hardware. It is cleared to '0' by software. 0: No COM event generated; 1: COM interrupt waiting for response;
16	UDITF	Update interrupt flag This bit is set by hardware when an update event occurs under the following conditions： –When TIMx_CTRL1.UPDIS = 0, And the repetition counter value overflow or underflow (generating an update event UEV when the repetition counter equals 0) –When TIMx_CTRL1.UPRS = 0, TIMx_CTRL1.UPDIS = 0, and set the TIMx_EVTGEN.UDGN bit by software to reinitialize the CNT。 –When TIMx_CTRL1.UPRS = 0, TIMx_CTRL1.UPDIS = 0, and the counter CNT is reinitialized by the trigger event. (Refer to the TIMx_SMCTRL register description) This bit is cleare by software。 0: No update event occurred; 1: Update interrupt occurred;
15:12	Reserved	Reserved, the reset value must be maintained
11	CC4OCF	Capture/Compare 4 overcapture flag Refer to CC1OCF description.
10	CC3OCF	Capture/Compare 3 overcapture flag Refer to CC1OCF description.
9	CC2OCF	Capture/Compare 2 overcapture flag Refer to CC1OCF description.
8	CC1OCF	Capture/Compare 1 overcapture flag This flag can only be set to 1 by hardware when the corresponding channel is configured for input capture. Writing 0 can clear this bit 0: No repeated capture generated. 1: When the value of the counter is captured into the TIMx_CCDAT1 register, the status of CC1ITF is already '1'
7:5	Reserved	Reserved, the reset value must be maintained
4	CC5ITF	Capture/Compare 5 interrupt flag Refer to CC1ITF description.
3	CC4ITF	Capture/Compare 4 interrupt flag Refer to CC1ITF description.
2	CC3ITF	Capture/Compare 3 interrupt flag Refer to CC1ITF description.

Bit Field	Name	Description
1	CC2ITF	Capture/Compare 2 interrupt flag Refer to CC1ITF description.
0	CC1ITF	Capture/Compare 1 interrupt flag **If channel CC1 is configured as an output mode:** Except in center-aligned mode, when the counter value is equal to the comparison value, this bit is set by hardware (refer to the TIMx_CTRL1.CAMSEL bit description). This bit is cleared by software. 0: No match occurred; 1: The value of TIMx_CNT is the same as the value of TIMx_CCDAT1 When the value of TIMx_CCDAT1 is greater than the value of TIMx_AR, if the counter overflows (in upcounting and up/downcounting modes) and underflows in downcounting mode, the TIMx_STS.CC1ITF bit will be set to '1'. **If channel CC1 is configured in input mode :** When a capture event occurs, this bit is set by hardware. This bit is cleared by software or by reading TIMx_CCDAT1. 0: No input capture occurred; 1: Input capture occurred; The counter value has been captured in TIMx_CCDAT1. An edge with the same selected polarity has been detected on IC1.

12.6.4 Event Generation Registers (TIMx_EVTGEN)

Offset address : 0x0C

Reset value : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				BGN	TGN	CCUDGN	UDGN	Reserved				CC4GN	CC3GN	CC2GN	CC1GN
				w	w	w	w					w	w	w	w

Bit Field	Name	Description
31:12	Reserved	Reserved, the reset value must be maintained.
11	BGN	Break 1 generation When set by the software, this bit can generate a break 1 event. At this time, TIMx_BKDT.MOEN = 0 and TIMx_STS.BITF = 1. If the corresponding interrupt and DMA are enabled, the corresponding interrupt and DMA will be generated. This bit is automatically cleared by hardware. 0: No action 1: Generated Break 1 event

Bit Field	Name	Description
10	TGN	Trigger generation This bit can generate a trigger event when set by software. And at this time TIMx_STS.TITF = 1, if the corresponding interrupt and DMA are enabled, the corresponding interrupt and DMA will be generated. This bit is automatically cleared by hardware. 0: No action 1: Generated a trigger event
9	CCUDGN	Capture/Compare control update generation This bit is set by software. If TIMx_CTRL2.CCPCTL=1 at this time, it allows updating the CCxEN, CCxNEN, and OCxMD bits. This bit is automatically cleared by hardware. 0: No action 1: Generated a COM event *Note: This bit is only valid for channels with complementary outputs*
8	UDGN	Update generation This bit is set by software and automatically cleared by hardware. This bit can generate an update event when set by software. And at this time the counter will be reinitialized, the prescaler counter will be cleared, the counter will be cleared in center-aligned or up-counting mode, but take TIMx_AR in down-counting mode the value of the register. This bit is automatically cleared by hardware. 0: No action 1: Generated an update event
7:4	Reserved	Reserved, the reset value must be maintained.
3	CC4GN	Capture/Compare 4 generation Refer to CC1GN description
2	CC3GN	Capture/Compare 3 generation Refer to CC1GN description
1	CC2GN	Capture/Compare 2 generation Refer to CC1GN description
0	CC1GN	Capture/Compare 1 generation This bit can generate a capture/compare event when set by software. This bit is automatically cleared by hardware. **When the corresponding channel of CC1 is in output mode:** The TIMx_STS.CC1ITF flag will be pulled high, if the corresponding interrupt and DMA are enabled, the corresponding interrupt and DMA will be generated. **When the corresponding channel of CC1 is in input mode:** TIMx_CCDAT1 will capture the current counter value, and the TIMx_STS.CC1ITF flag will be pulled high, if the corresponding interrupt and DMA are enabled, the corresponding interrupt and DMA will be generated. If The IMx_STS.CC1ITF is already pulled high, pull TIMx_STS.CC1OCF high. 0: No action 1: Generated a CC1 capture/compare event

12.6.5 Slave Mode Control（TIMx_SMCTRL）

Offset address：0x10

Reset value：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								OCREFCLRF[3:0]				OCREF CLRP	Reserved		MSMD
								rw				rw			rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTF[3:0]				EXTP	EXCEN	EXTPS		SMSEL[3:0]				Reserved	TSEL[2:0]		
rw				rw	rw	rw		rw					rw		

Bit Field	Name	Description
31:24	Reserved	Reserved, the reset value must be maintained.
23:20	OCREFCLRF	tim_ocref_clr signal filter These bits are used to define the sampling frequency of the tim_ocref_clr signal and the bandwidth of the tim_ocref_clr digital filter. In fact, the digital filter is an event counter that generates a validated output after recording a continuous N events. 0000: No filter, sampled 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: $f_{SAMPLING}=f_{DTS/4}$, 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
19	OCREFCLRP	tim_ocref_clr signal polarity This bit selects whether to use tim_ocref_clr or the invert of tim_ocref_clr as the trigger operation 0: tim_ocref_clr is active high or on the rising edge; 1: tim_ocref_clr is active low or on the falling edge;
18:17	Reserved	Reserved, the reset value must be maintained.
16	MSMD	Master/slave mode 0: No action; 1: The event on the trigger input (TRGI) is delayed to allow perfect synchronization between the current timer (via TRGO) and its slave timers. This is very useful when multiple timers need to be synchronized to a single external event;
15:12	EXTF[3:0]	External trigger filter These bits are used to define the sampling frequency of the ETRP signal and the bandwidth of the ETRP digital filter. In fact, the digital filter is an event counter that generates a validated output after recording a continuous N events. 0000: No filter, sampled 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: $f_{SAMPLING}=f_{DTS/4}$, 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
11	EXTP	External trigger polarity This bit selects whether to use tim_etr_in or the invert of tim_etr_in as the trigger operation 0：tim_etr_in is active high or on the rising edge; 1：tim_etr_in is active low or on the falling edge;
10	EXCEN	External clock enable This bit enables External Clock Mode 2. When enabled, the counter is driven by any valid edge on the ETRF signal. 0: Disable External Clock Mode 2; 1: Enable External Clock Mode 2; *Note 1: When External Clock Mode 1 and External Clock Mode 2 are enabled simultaneously, the input for external clock is ETRF.* *Note 2: The following slave modes can be used simultaneously with External Clock Mode 2: Reset Mode, Gated Mode, and Trigger Mode; however, TRGI cannot be connected to ETRF (TIMx_SMCTRL.TSEL≠'111').* *Note 3: Setting the TIMx_SMCTRL.EXCEN bit to select External Clock Mode 1 and connect TRGI to ETRF (TIMx_SMCTRL.SMSEL=111 and TIMx_SMCTRL.TSEL=111) has the same effect.*

9:8	EXTPS[1:0]	External trigger prescaler
		The frequency of the external trigger signal ETRP must be at most 1/4 of the TIMxCLK frequency. When using a faster external clock input, a prescaler can be used to reduce the frequency of ETRP.
		00: Disable prescaling;
		01: ETRP frequency divided by 2;
		10: ETRP frequency divided by 4;
		11: ETRP frequency divided by 8;

7:4	SMSEL[3:0]	Slave mode selection When an external signal is selected, the effective edge of the trigger signal (TRGI) is related to the selected external input polarity (refer to the description of the input control register and control register) 0000: Disable Slave Mode - If CNTEN = 1, the prescaler is directly driven by the internal clock。 0001: Encoder Mode 1 - The counter counts up/down on the edges of TI1FP1 based on the level of TI2FP2. 0010: Encoder Mode 2 - The counter counts up/down on the edges of TI2FP2 based on the level of TI1FP1. 0011: Encoder Mode 3 - The counter counts up/down on the edges of TI1FP1 and TI2FP2 based on the input level of another signal. 0100: Reset Mode - On the rising edge of the selected trigger input (TRGI), the counter is reinitialized and the shadow registers are updated. 0101: Gated Mode - When the trigger input (TRGI) is high, the counter clock is enabled. Once the trigger input becomes low, the counter stops (but does not reset). The start and stop of the counter

		are controlled. 0110: Trigger Mode - The counter starts on the rising edge of the trigger input TRGI (but does not reset), only the start of the counter is controlled. 0111: External Clock Mode 1 - The rising edge of the selected trigger input (TRGI) drives the counter. 1000: Dual input pulse encoder mode 2. 1001：Quadrature Encoder Mode 4 - The counter counts up/down on the edges of TI1FP1 based on the level of TI2FP2. The counting edge is selected via CC1P. 1010：Quadrature Encoder Mode 5 - The counter counts up/down on the edges of TI2FP2 based on the level of TI1FP1. The counting edge is selected via CC2P 1011: Pulse Width Encoder Mode 2. 1100: Pulse Width Encoder Mode 1. The counting edge of TI2FP2 is set via CC2P 1101: Combined Gated + Reset Mode - When the trigger input (TRGI) is high, the counter clock is enabled. Once the trigger input becomes low, the counter stops (and resets). The start and stop of the counter are both controlled。 1110: Combined Reset + Trigger Mode - The counter starts and resets on the rising edge of the trigger input TRGI, with only the start of the counter being controlled 1111: Dual Input Pulse Encoder Mode 1. The counting sensitive edges of TI1FP1 and TI2FP2 are set via CC1P and CC2P。 Note: Do not use Gated Mode when TI1F_ED is selected as the trigger input (TSEL=100). This is because TI1F_ED outputs a pulse every time TI1F changes, whereas Gated Mode checks the level of the trigger input。
3	Reserved	Reserved, the reset value must be maintained.
2:0	TSEL[2:0]	Trigger selection These 3 bits are used to select the trigger input for synchronizing the counter 0xx: Internal Trigger (ITRx) selects the ITR signal source based on TIMx_INSEL 100: Edge Detector for TI1 (TI1F_ED) 101: Filtered Timer Input 1 (TI1FP1) 110: Filtered Timer Input 2 (TI2FP2) 111: External Trigger input（ETRF） *Note: These bits can only be changed when not in use (e.g., SMSEL=000) to avoid incorrect edge detection when changed*

12.6.6 DMA/Interrupt Enable Register（TIMx_DINTEN）

Offset address: 0x14

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved									COMIEN	TDEN	COMDEN	UDEN	BIEN	TIEN	UIEN

									rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				CC4DEN	CC3DEN	CC2DEN	CC1DEN	Reserved				CC4IEN	CC3IEN	CC2IEN	CC1IEN
				rw	rw	rw	rw					rw	rw	rw	rw

Bit Field	Name	Description
31:23	Reserved	Reserved, the reset value must be maintained

22	COMIEN	COM interrupt enable
		0: Disable COM interrupt;
		1: Enable COM interrupt;

21	TDEN	Trigger DMA request enable 0: Disable trigger DMA request; 1: Enable trigger DMA request.
20	COMDEN	COM DMA request enable 0: Disable COM DMA request; 1: Enable COM DMA request.
19	UDEN	Update DMA request enable 0: Disable update DMA request; 1: Enable update DMA request.
18	BIEN	Break interrupt enable 0: Disable break interrupt; 1: Enable break interrupt.
17	TIEN	Trigger interrupt enable 0: Disable trigger interrupt; 1: Enable trigger interrupt.
16	UIEN	Update interrupt enable 0: Disable update interrupt; 1: Enable update interrupt.
15:12	Reserved	Reserved, the reset value must be maintained
11	CC4DEN	Capture/Compare 4 DMA request enable 0: Disable Capture/Compare 4 DMA request; 1: Enable Capture/Compare 4 DMA request
10	CC3DEN	Capture/Compare 3 DMA request enable 0: Disable Capture/Compare 3 DMA request; 1: Enable Capture/Compare 3 DMA request
9	CC2DEN	Capture/Compare 2 DMA request enable 0: Disable Capture/Compare 2 DMA request; 1: Enable Capture/Compare 2 DMA request
8	CC1DEN	Capture/Compare 1 DMA request enable 0: Disable Capture/Compare 1 DMA request; 1: Enable Capture/Compare 1 DMA request
7:4	Reserved	Reserved, the reset value must be maintained

Bit Field	Name	Description
3	CC4IEN	Capture/Compare 4 interrupt enable 0: Disable capture/compare 4 interrupt; 1: Enable capture/compare 4 interrupt.
2	CC3IEN	Capture/Compare 3 interrupt enable 0: Disable capture/compare 3 interrupt; 1: Enable capture/compare 3 interrupt.
1	CC2IEN	Capture/Compare 2 interrupt enable 0: Disable capture/compare 2 interrupt; 1: Enable capture/compare 2 interrupt.
0	CC1IEN	Capture/Compare 1 interrupt enable 0: Disable capture/compare 1 interrupt; 1: Enable capture/compare 1 interrupt.

12.6.7 Capture/Compare Mode Register 1（TIMx_CCMOD1）

Offset address: 0x18

Reset value: 0x0000 0000

The channel can be used for input (capture mode) or output (compare mode), and the direction of the channel is defined by the corresponding CCxSEL bit. The functions of the other bits in this register are different in input and output modes. OCx describes the function of the channel in output mode, while ICx describes the function of the channel in input mode. Therefore, it is important to note that the functionality of the same bit is different in output and input modes。

Output Compare Mode:

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved													OC2MD3	OC1MD3	Reserved
													rw	rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC2MD[2:0]			OC2CEN	OC2FEN	OC2PEN	CC2SEL[1:0]		OC1MD[2:0]			OC1CEN	OC1FEN	OC1PEN	CC1SEL[1:0]	
rw			rw	rw	rw	rw		rw			rw	rw	rw	rw	

Bit Field	Name	Description
31:19	Reserved	Reserved, the reset value must be maintained
18	OC2MD3	Used with OC2MD, see the description of OC2MD for details.
17	OC1MD3	Used with OC1MD, see the description of OC1MD for details.
16	Reserved	Reserved, the reset value must be maintained
15:13	OC2MD[2:0]	Output Compare 2 mode OC2MD3 is used with OC2MD to manage the output reference signal OC2REF, which determines the values of OC2 and OC2N. The effective levels of OC2 and OC2N depend on the TIMx_CCEN.CC2P and TIMx_CCEN.CC2NP bits when it is high level effective. The 4-bit signals {OC2MD_3, OC2MD} correspond to the following Output Compare 1 modes: 0000: Frozen. The comparison between the TIMx_CCDAT2 register and the counter

Bit Field	Name	Description
		TIMx_CNT has no effect on the OC2REF signal. 0001: Set channel 2 to the active level when a match occurs. When TIMx_CCDAT2 = TIMx_CNT, the OC2REF signal will be forced to a high level. 0010: Set channel 2 to the inactive level when a match occurs. When TIMx_CCDAT2 = TIMx_CNT, the OC2REF signal will be forced to a low level. 0011: Toggle. When TIMx_CCDAT2 = TIMx_CNT, the OC2REF signal will be toggled. 0100: Force Inactive Level. The OC2REF signal is forced to a low level. 0101: Force Active Level. The OC2REF signal is forced to a high level. 0110: PWM Mode 1 - In up-counting mode, if TIMx_CNT < TIMx_CCDAT2, channel 2's OC2REF signal is high; otherwise, it is low. In down-counting mode, if TIMx_CNT > TIMx_CCDAT2, channel 2's OC2REF signal is low; otherwise, it is high. 0111: PWM Mode 2 - In up-counting mode, if TIMx_CNT < TIMx_CCDAT2, channel 2's OC2REF signal is low; otherwise, it is high. In down-counting mode, if TIMx_CNT > TIMx_CCDAT2, channel 2's OC2REF signal is high; otherwise, it is low. 1000: Retriggerable Single Pulse Mode 1 - In the up-counting mode, the channel is in the active state until a trigger event is detected (on the TRGI signal). Then, a comparison is made in PWM mode 1, and the channel will become active again on the next update. In the down-counting mode, the channel is in the inactive state until a trigger event is detected (on the TRGI signal). Then, a comparison is made in PWM mode 1, and the channel will become inactive again on the next update. 1001: Retriggerable Single Pulse Mode 2 - In the up-counting mode, the channel is in the inactive state until a trigger event is detected (on the TRGI signal). Then, a comparison is made in PWM mode 2, and the channel will become inactive again on the next update. In the down-counting mode, the channel is in the active state until a trigger event is detected (on the TRGI signal). Then, a comparison is made in PWM mode 2, and the channel will become active again on the next update. 1010-1101：Reserved 1110: Combined PWM Mode 1 - OC2REF behaves the same as in PWM Mode 1. OC2REFC is the logical OR of OC1REF and OC2REF. 1111: Combined PWM Mode 2 - OC2REF behaves the same as in PWM Mode 2. OC2REFC is the logical AND of OC1REF and OC2REF。 *Note: In PWM Mode 1 or PWM Mode 2, the OC2REF level changes only when the comparison result changes or when the output compare mode switches from frozen mode to PWM mode.*
12	OC2CEN	Output Compare 2 clear enable
11	OC2FEN	Output Compare 2 fast enable
10	OC2PEN	Output Compare 2 preload enable
9:8	CC2SEL[1:0]	Capture/Compare 2 selection This bit defines the direction (input/output) of the channel and the selection of the input pin： 00: CC2 channel is configured as an output; 01: CC2 channel is configured as an input, IC2 is mapped to TI2; 10: CC2 channel is configured as an input, IC2 is mapped to TI1;

Bit Field	Name	Description
		11: CC2 channel is configured as an input, IC2 is mapped to TRC. This mode only works when the internal trigger input is selected (chosen by the TSEL bit of the TIMx_SMCTRL register). *Note: CC2SEL is writable only when the channel is disabled (CC2EN=0 in the TIMx_CCEN register).*
7:5	OC1MD[2:0]	Output Compare 1 mode OC1MD3 and OC1MD are used together to control the output reference signal OC1REF, which determines the values of OC1 and OC1N, when high-level effectiveness. The effective level of OC1 and OC1N depends on the TIMx_CCEN.CC1P and TIMx_CCEN.CC1NP bits. The 4-bit signal {OC1M_3, OC1M} corresponds to the following output comparison mode 1: 0000: Freeze. The comparison between the TIMx_CCDAT1 register and the TIMx_CNT counter has no effect on the OC1REF signal. 0001: Setting channel 1 to the effective level when matched. When TIMx_CCDAT1=TIMx_CNT, the OC1REF signal will be forced to a high level. 0010: Setting channel 1 to the ineffective level when matched. When TIMx_CCDAT1=TIMx_CNT, the OC1REF signal will be forced to a low level. 0011: Toggle. When TIMx_CCDAT1=TIMx_CNT, the OC1REF signal will be toggled. 0100: Force inactive level. The OC1REF signal is forced to a low level. 0101: Force active level. The OC1REF signal is forced to a high level. 0110: PWM mode 1 - In up-counting mode, if TIMx_CNT < TIMx_CCDAT1, the OC1REF signal of channel 1 is high, otherwise low. In down-counting mode, if TIMx_CNT > TIMx_CCDAT1, the OC1REF signal of channel 1 is low, otherwise high. 0111: PWM mode 2 - In up-counting mode, if TIMx_CNT < TIMx_CCDAT1, the OC1REF signal of channel 1 is low, otherwise high. In down-counting mode, if TIMx_CNT > TIMx_CCDAT1, the OC1REF signal of channel 1 is high, otherwise low. 1000: Retriggerable single pulse mode 1 - In the upcounting mode, the channel is in the active state until a trigger event is detected (on the TRGI signal). Then, in PWM mode 1 comparison is done, the channel will return to the active state at the next update. In the downcounting mode, the channel is in the inactive state until a trigger event is detected (on the TRGI signal). Then, in PWM mode 1 comparison is done, the channel will return to the inactive state at the next update. 1001: Retriggerable single pulse mode 2 - In the upcounting mode, the channel is in the inactive state until a trigger event is detected (on the TRGI signal). Then, in PWM mode 2 comparison is done, the channel will return to the inactive state at the next update. In the downcounting mode, the channel is in the active state until a trigger event is detected (on the TRGI signal). Then, in PWM mode 2 comparison is done, the channel will return to the active state at the next update. 1010-1101：Reserved 1110: Combined PWM mode 1 - OC1REF behaves the same as in PWM mode 1. OC1REFC is the logical OR operation result of OC1REF and OC2REF. 1111: Combined PWM mode 2 - OC1REF behaves the same as in PWM mode 2. OC1REFC is the logical AND operation result of OC1REF and OC2REF.

Bit Field	Name	Description
		Note: In PWM mode 1 or PWM mode 2, the OC1REF level only changes when the comparison result changes or when the output comparison mode switches from freeze mode to PWM mode.
4	OC1CEN	Output Compare 1 clear enable 0: OC1REF is not affected by the tim_ocref_clr_in input; 1: Once a high level is detected on the tim_ocref_clr_in input (The tim_ocref_clr_in is controlled by the TIMx_CTRL1.CLRSEL source), OC1REF will be set to 0.
3	OC1FEN	Output Compare 1 fast enable This bit is used to speed up the response of the CC output to trigger input events. 0: Based on the values of the counter and CCDAT1, CC1 operates normally even if the trigger is enabled. When the trigger input has a valid edge, the minimum delay for activating the CC1 output is 5 clock cycles. 1: The effect of a valid edge input to the trigger is as if a comparison match occurred. Therefore, OC1 is set to the comparison level regardless of the comparison result. The delay between the valid edge of the sampling trigger and the CC1 output is shortened to 3 clock cycles. OCxFEN only works when the channel is configured in PWM1 or PWM2 mode
2	OC1PEN	Output Compare 1 preload enable 0: Disable the preload function of the TIMx_CCDAT1 register. Support writing to the TIMx_CCDAT1 register at any time, and the written value takes effect immediately. 1: Enable the preload function of the TIMx_CCDAT1 register. Only read and write operations are performed on the preload register. When an update event occurs, the value of TIMx_CCDAT1 is loaded into the shadow register. *Note: Only when TIMx_CTRL1.ONEPM=1 (in single pulse mode) can the PWM mode be used without validating the preload register; otherwise, other behaviors cannot be predicted.*
1:0	CC1SEL[1:0]	Capture/Compare 1 selection These 2 bits define the direction (input/output) of the channel and the selection of the input pin： 00: CC1 channel is configured as an output； 01: CC1 channel is configured as an input, and IC1 is mapped to TI1； 10: CC1 channel is configured as an input, and IC1 is mapped to TI2； 11: CC1 channel is configured as an input, and IC1 is mapped to TRC. This mode only works when the internal trigger input is selected (chosen by the TSEL bit of the TIMx_SMCTRL register). *Note: CC1SEL is only writable when the channel is disabled (CC1EN = 0 in the TIMx_CCEN register)*

Input capture mode :

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IC2F[3:0]				IC2PSC[1:0]		CC2SEL[1:0]		IC1F[3:0]				IC1PSC[1:0]		CC1SEL[1:0]	
rw				rw		rw		rw				rw		rw	

Bit Field	Name	Description
31:16	Reserved	Reserved, the reset value must be maintained
15:12	IC2F[3:0]	Input capture 2 filter
11:10	IC2PSC[1:0]	Input capture 2 prescaler
9:8	CC2SEL[1:0]	Capture/Compare 2 selection These 2 bits define the direction of the channel (input/output) and the selection of the input pin： 00: CC2 channel is configured as an output； 01: CC2 channel is configured as an input, and IC2 is mapped to TI2； 10: CC2 channel is configured as an input, and IC2 is mapped to TI1； 11: CC2 channel is configured as an input, and IC2 is mapped to TRC. This mode only works when the internal trigger input is selected (chosen by the TSEL bit of the TIMx_SMCTRL register). *Note: CC2SEL is only writable when the channel is disabled (CC2EN = 0 in the TIMx_CCEN register).*
7:4	IC1F[3:0]	Input capture 1 filter These bits define the sampling frequency of the TI1 input and the length of the digital filter. The digital filter consists of an event counter, which generates an output change after counting N events： 0000: No filter, sampled 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: $f_{SAMPLING}=f_{DTS}/4$, 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
3:2	IC1PSC[1:0]	Input capture 1 prescaler These 2 bits define the prescaler factor for the CC1 input (IC1). Once TIMx_CCEN.CC1EN=0, the prescaler is reset. 00: No prescaler, each edge detected on the capture input triggers a capture; 01: Capture is triggered every 2 events; 10: Capture is triggered every 4 events; 11: Capture is triggered every 8 events.

1:0	CC1SEL[1:0]	Capture/Compare 1 Selection These 2 bits define the direction of the channel (input/output) and the selection of the input pin: 00: CC1 channel is configured as an output; 01: CC1 channel is configured as an input, and IC1 is mapped to TI1; 10: CC1 channel is configured as an input, and IC1 is mapped to TI2; 11: CC1 channel is configured as an input, and IC1 is mapped to TRC. This mode only works when the internal trigger input is selected (chosen by the TSEL bit of the TIMx_SMCTRL register). *Note: CC1SEL is only writable when the channel is disabled (CC1EN = 0 in the TIMx_CCEN register).*

12.6.8 Capture/Compare Mode Register 2（TIMx_CCMOD2）

Offset address：0x1C

Reset value：0x0000 0000

Refer to the description of the CCMOD1 register above

output compare mode:

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved													OC4MD3	OC3MD3	Reserved
													rw	rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC4MD[2:0]			OC4CEN	OC4FEN	OC4PEN	CC4SEL[1:0]		OC3MD[2:0]			OC3CEN	OC3FEN	OC3PEN	CC3SEL[1:0]	
rw			rw	rw	rw	rw		rw			rw	rw	rw	rw	

Bit Field	Name	Description
31:19	Reserved	Reserved, the reset value must be maintained
18	OC4MD3	Used with OC4MD, refer to the description of OC4MD for details
17	OC3MD3	Used with OC3MD, refer to the description of OC3MD for details
16	Reserved	Reserved, the reset value must be maintained
15:13	OC4MD[2:0]	Output compare 4 mode OC4MD3 and OC4MD define the action of the output reference signal OC4REF, which determines the value of OC4. OC4REF is active at high level, and the effective level of OC4 depends on the CC4P bit. The 4-bit signal {OC4MD3, OC4MD} corresponds to the following output compare 4 modes： 0000: Freeze. The comparison between the output compare register TIMx_CCDAT4 and the counter TIMx_CNT does not affect OC4REF; 0001: Set channel 4 to active level when a match occurs. When the value of counter TIMx_CNT is the same as the capture/compare register 4 (TIMx_CCDAT4), force OC4REF to high. 0010: Set channel 4 to an inactive level when a match occurs. When the value of counter TIMx_CNT is the same as the capture/compare register 4 (TIMx_CCDAT4), force OC4REF to low. 0011: Toggle. When TIMx_CCDAT4 = TIMx_CNT, toggle the level of OC4REF. 0100: Force to an inactive level. Force OC4REF to low.

Bit Field	Name	Description
31:19	Reserved	Reserved, the reset value must be maintained
		0101: Force to an active level. Force OC4REF to high. 0110: PWM mode 1 - During up-counting, channel 4 is at an active level when TIMx_CNT < TIMx_CCDAT3, otherwise it is at an inactive level. During down-counting, channel 4 is at an inactive level (OC4REF=0) when TIMx_CNT > TIMx_CCDAT4, otherwise it is at an active level (OC4REF=1). 0111: PWM mode 2 - During up-counting, channel 4 is at an inactive level when TIMx_CNT < TIMx_CCDAT4, otherwise it is at an active level. During down-counting, channel 4 is at an active level when TIMx_CNT > TIMx_CCDAT4, otherwise it is at an inactive level. 1000: Retriggerable single pulse mode 1 - In the up-counting mode, channel 4 is in the active state until a trigger event is detected (on the TRGI signal). Then, in PWM mode 1, channel 4 will return to the active state on the next update. In the down-counting mode, channel 4 is in the inactive state until a trigger event is detected (on the TRGI signal). Then, in PWM mode 1, channel 4 will return to the inactive state on the next update。 1001: Retriggerable single pulse mode 2 - In the up-counting mode, channel 4 is inactive until a trigger event is detected (on the TRGI signal). Then, in PWM mode 2 comparison, channel 4 will return to the inactive state on the next update. In the down-counting mode, channel 4 is active until a trigger event is detected (on the TRGI signal). Then, in PWM mode 2 comparison, channel 4 will return to the active state on the next update. 1010-1101：Reserved 1110: Combined PWM mode 1 - OC4REF behaves the same as in PWM mode 1. OC4REFC is the logical OR operation result of OC4REF and OC3REF. 1111: Combined PWM mode 2 - OC4REF behaves the same as in PWM mode 2. OC4REFC is the logical AND operation result of OC4REF and OC3REF. *Note: In PWM mode 1 or PWM mode 2, the OC4REF level changes only when the comparison result changes or when switching from freeze mode to PWM mode in the output compare mode.*
12	OC4CEN	Output compare 4 clear enable
11	OC4FEN	Output compare 4 fast enable
10	OC4PEN	Output compare 4 preload enable
9:8	CC4SEL[1:0]	Capture/Compare 4 selection These 2 bits define the direction (input/output) of the channel and the selection of the input pin： 00: CC4 channel is configured as an output; 01: CC4 channel is configured as an input, with IC4 mapped on TI4; 10: CC4 channel is configured as an input, with IC4 mapped on TI3； 11: CC4 channel is configured as an input, with IC4 mapped on TRC. This mode only works when the internal trigger input is selected (chosen by the TSEL bit of the TIMx_SMCTRL register). Note: CC4SEL is writable only when the channel is disabled (CC4EN=0 in the TIMx_CCEN register).
7:5	OC3MD[2:0]	Output compare 3 mode OC3MD3 and OC3MD define the action of the output reference signal OC3REF, which

Bit Field	Name	Description
31:19	Reserved	Reserved, the reset value must be maintained
		determines the value of OC3. OC3REF is active at high level, and the effective level of OC3 depends on the CC3P bit. The 4-bit signal {OC3MD3, OC3MD} corresponds to the following output compare 3 modes： 0000: Freeze. The comparison between the output compare register TIMx_CCDAT3 and the counter TIMx_CNT does not affect OC3REF; 0001: Set channel 3 to an active level when a match occurs. When the value of counter TIMx_CNT is the same as the capture/compare register 3 (TIMx_CCDAT3), force OC3REF to high. 0010: Set channel 3 to an inactive level when a match occurs. When the value of counter TIMx_CNT is the same as the capture/compare register 3 (TIMx_CCDAT3), force OC3REF to low. 0011: Toggle. When TIMx_CCDAT3 = TIMx_CNT, toggle the level of OC3REF. 0100: Force to an inactive level. Force OC3REF to low. 0101: Force to an active level. Force OC3REF to high. 0110: PWM mode 1 - During up-counting, channel 3 is at an active level when TIMx_CNT < TIMx_CCDAT3, otherwise it is at an inactive level. During down-counting, channel 3 is at an inactive level (OC3REF=0) when TIMx_CNT > TIMx_CCDAT3, otherwise it is at an active level (OC3REF=1). 0111: PWM mode 2 - During up-counting, channel 3 is at an inactive level when TIMx_CNT < TIMx_CCDAT3, otherwise it is at an active level. During down-counting, channel 3 is at an active level when TIMx_CNT > TIMx_CCDAT3, otherwise it is at an inactive level. 1000: Retriggerable single pulse mode 1 - In the up-counting mode, channel 3 is in the active state until a trigger event is detected (on the TRGI signal). Then, in PWM mode 1 comparison, channel 3 will return to the active state on the next update. In the down-counting mode, channel 3 is in the inactive state until a trigger event is detected (on the TRGI signal). Then, in PWM mode 1 comparison, channel 3 will return to the inactive state on the next update. 1001: Retriggerable single pulse mode 2 - In the up-counting mode, channel 3 is in the inactive state until a trigger event is detected (on the TRGI signal). Then, in PWM mode 2 comparison, channel 3 will return to the inactive state on the next update. In the down-counting mode, channel 3 is in the active state until a trigger event is detected (on the TRGI signal). Then, in PWM mode 2 comparison, channel 3 will return to the active state on the next update. 1010-1101：Reserved 1110: Combined PWM mode 1 - OC3REF behaves the same as in PWM mode 1. OC3REFC is the logical OR operation result of OC3REF and OC4REF. 1111: Combined PWM mode 2 - OC3REF behaves the same as in PWM mode 2. OC3REFC is the logical AND operation result of OC3REF and OC4REF. *Note: In PWM mode 1 or PWM mode 2, the OC3REF level changes only when the comparison result changes or when switching from freeze mode to PWM mode in the output compare mode.*
4	OC3CEN	Output compare 3 clear enable
3	OC3FEN	Output compare 3 fast enable
2	OC3PEN	Output compare 3 preload enable

Bit Field	Name	Description
31:19	Reserved	Reserved, the reset value must be maintained
1:0	CC3SEL[1:0]	Capture/Compare 3 selection These 2 bits define the direction (input/output) of the channel and the selection of the input pin: 00: CC3 channel is configured as an output; 01: CC3 channel is configured as an input, with IC3 mapped on TI3; 10: CC3 channel is configured as an input, with IC3 mapped on TI4; 11: CC3 channel is configured as an input, with IC3 mapped on TRC. This mode only works when the internal trigger input is selected (chosen by the TSEL bit of the TIMx_SMCTRL register). *Note: CC3SEL is writable only when the channel is disabled (CC3EN=0 in the TIMx_CCEN register).*

Input capture mode:

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IC4F[3:0]				IC4PSC[1:0]		CC4SEL[1:0]		IC3F[3:0]				IC3PSC[1:0]		CC3SEL[1:0]	
rw				rw		rw		rw				rw		rw	

Bit Field	Name	Description
31:16	Reserved	Reserved, the reset value must be maintained
15:12	IC4F[3:0]	Input capture 4 filter
11:10	IC4PSC[1:0]	Input capture 4 prescaler
9:8	CC4SEL[1:0]	Capture/Compare 4 selection Note: These 2 bits define the direction (input/output) of the channel and the selection of the input pin: 00: CC4 channel is configured as an output； 01: CC4 channel is configured as an input, with IC4 mapped on TI4； 10: CC4 channel is configured as an input, with IC4 mapped on TI3； 11: CC4 channel is configured as an input, with IC4 mapped on TRC. This mode only works when the internal trigger input is selected (chosen by the TSEL bit of the TIMx_SMCTRL register). Note: CC4SEL is writable only when the channel is disabled (CC4EN=0 in the TIMx_CCEN register).
7:4	IC3F[3:0]	Input capture 3 filter
3:2	IC3PSC[1:0]	Input capture 3 prescaler

Bit Field	Name	Description
31:16	Reserved	Reserved, the reset value must be maintained
1:0	CC3SEL[1:0]	Capture/Compare 3 selection These 2 bits define the direction (input/output) of the channel and the selection of the input pin： 00: CC3 channel is configured as an output； 01: CC3 channel is configured as an input, with IC3 mapped on TI3； 10: CC3 channel is configured as an input, with IC3 mapped on TI4； 11: CC3 channel is configured as an input, with IC3 mapped on TRC. This mode only works when the internal trigger input is selected (chosen by the TSEL bit of the TIMx_SMCTRL register). Note: CC3SEL is writable only when the channel is disabled (CC3EN=0 in the TIMx_CCEN register).

12.6.9 Capture/Compare Mode Register 3（TIMx_CCMOD3）

Offset address：0x20

Reset value：0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								OC5MD[2:0]			OC5CEN	OC5FEN	OC5PEN	Reserved	
									rw		rw	rw	rw		

Bit Field	Name	Description
31:8	Reserved	Reserved, the reset value must be maintained
7:5	OC5MD[2:0]	Output compare 5 mode
4	OC5CEN	Output compare 5 clear enable
3	OC5FEN	Output compare 5 fast enable
2	OC5PEN	Output compare 5 preload enable
1:0	Reserved	Reserved, the reset value must be maintained

12.6.10 Capture/Compare Enable Register（TIMx_CCEN）

Offset address：0x24

Reset value：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved												CC5P	CC5EN	Reserved	
												rw	rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

CC4P	CC4EN	Reserved	CC3P	CC3EN	Reserved	CC2P	CC2EN	Reserved	CC1P	CC1EN	CC1NP	CC1NEN
rw	rw		rw	rw		rw	rw		rw	rw	rw	rw

Bit Field	Name	Description
31:20	Reserved	Reserved, the reset value must be maintained
19	CC5P	Capture/Compare 5 output polarity Refer to the description of TIMx_CCEN.CC1P.
18	CC5EN	Capture/Compare 5 output enable Refer to the description of TIMx_CCEN.CC1EN.
17：16	Reserved	Reserved, the reset value must be maintained
15	CC4P	Capture/Compare 4 output polarity Refer to the description of TIMx_CCEN.CC1P.
14	CC4EN	Capture/Compare 4 output enable Refer to the description of TIMx_CCEN.CC1EN.
13:12	Reserved	Reserved, the reset value must be maintained
11	CC3P	Capture/Compare 3 output polarity Refer to the description of TIMx_CCEN.CC1P.
10	CC3EN	Capture/Compare 3 output enable Refer to the description of TIMx_CCEN.CC1EN.
9:8	Reserved	Reserved, the reset value must be maintained
7	CC2P	Capture/Compare 2 output polarity Refer to the description of TIMx_CCEN.CC1P.
6	CC2EN	Capture/Compare 2 output enable Refer to the description of TIMx_CCEN.CC1EN.
5:4	Reserved	Reserved, the reset value must be maintained
3	CC1P	Capture/Compare 1 output polarity **When CC1 corresponds to the output mode:** 0: OC1 is active high 1: OC1 is active low **When CC1 corresponds to the input mode:** In this case, this bit is used to select whether to use IC1 or the inverted signal of IC1 as the trigger or capture signal. 0: Non-inverted: A capture action occurs when IC1 generates a rising edge. When used as an external trigger, IC1 is non-inverted. 1: Inverted: A capture action occurs when IC1 generates a falling edge. When used as an external trigger, IC1 is inverted.
2	CC1EN	Capture/Compare 1 output enable **CC1 channel configured as output：** 0: Disabled - OC1 output is disabled, so the output level of OC1 depends on the values of MOEN, OSSI, OSSR, OI1, OI1N, and CC1NEN bits. 1: Enabled - OC1 signal is output to the corresponding output pin, and its output level depends on the values of the MOEN, OSSI, OSSR, OI1, OI1N, and CC1NEN bits. **CC1 channel configured as input：**

Bit Field	Name	Description

		This bit determines whether the value of the counter can be captured into the TIMx_CCDAT1 register.
		0: Capture disable;
		1: Capture enable;

1	CC1NP	Capture/Compare 1 complementary output polarity
		0: OC1N high level active;
		1: OC1N low level active;

0	CC1NEN	Capture/Compare 1 complementary output enable 0: Disabled - Disables the output of the OC1N signal. The level of OC1N depends on the values of TIMx_BKDT.MOEN, TIMx_BKDT.OSSI, TIMx_BKDT.OSSR, TIMx_CTRL2.OI1, TIMx_CTRL2.OI1N, and TIMx_CCEN.CC1EN. 1: Enabled - Enables the output of the OC1N signal. The level of OC1N depends on the values of TIMx_BKDT.MOEN, TIMx_BKDT.OSSI, TIMx_BKDT.OSSR, TIMx_CTRL2.OI1, TIMx_CTRL2.OI1N, and TIMx_CCEN.CC1EN.

Table 12-14 Control bits for complementary output channels OCx and OCxN with brake function

Control Bit					Output State[1]	
MOEN	**OSSI**	**OSSR**	**CCxEN**	**CCxNEN**	**OCx Output State**	**OCxN Output State**
1	X	0	0	0	Output disabled (disconnected from the timer) OCx=0，OCx_EN=0	Output disabled (disconnected from the timer) OCxN=0，OCxN_EN=0
		0	0	1	Output disabled (disconnected from the timer) OCx=0，OCx_EN=0	OCxREF + Polarity， OCxN= OCxREF xor CCxNP， OCxN_EN=1
		0	1	0	OCxREF + Polarity， OCx= OCxREF xor CCxP， OCx_EN=1	Output disabled (disconnected from the timer) OCxN=0，OCxN_EN=0
		0	1	1	OCxREF + Polarity + Deadtime，OCx_EN=1	OCxREF Invert + Polarity + Deadtime， OCxN_EN=1
		1	0	0	Output disabled (disconnected from the timer) OCx=CCxP，OCx_EN=0	Output disabled (disconnected from the timer) OCxN=CCxNP，OCxN_EN=0
		1	0	1	Disabled state (output enabled but at an inactive level) OCx=CCxP，OCx_EN=1	OCxREF + Polarity， OCxN= OCxREF xor CCxNP， OCxN_EN=1

		1	1	0	OCxREF + Polarity， OCx= OCxREF xor CCxP， OCx_EN=1	Disabled state (output enabled but at an inactive level) OCxN=CCxNP，OCxN_EN=1
		1	1	1	OCxREF + Polarity + Deadtime， OCx_EN=1	OCxREF Invert + Polarity + Deadtime， OCxN_EN=1
0	0	X	0	0	Output disabled (disconnected from the timer) Asynchronous: OCx=CCxP， OCx_EN=0， OCxN=CCxNP，OCxN_EN=0； If clock exists: Assume (CCxP ^ OIx) ^ (CCxNP^OIxN)! = 0， After a dead time OCx=OIx，OCxN=OIxN	
	0		0	1		
	0		1	0		
	0		1	1		
	1		0	0	Disabled state (output enabled but at an inactive level) Asynchronous: OCx=CCxP, OCx_EN=1, OCxN=CCxNP, OCxN_EN=1； If clock exists: Assume (CCxP ^ OIx) ^ (CCxNP^OIxN)! = 0， After a dead time OCx=OIx, OCxN=OIxN，	
	1		0	1		
	1		1	0		
	1		1	1		

1. If both outputs of a channel are not used (CCxEN = CCxNEN = 0), then OIx, OIxN, CCxP, and CCxNP must all be cleared.

Note: The state of the pins connected to the external I/O pins of the complementary OCx and OCxN channels depends on the states of the OCx and OCxN channels and the GPIO and AFIO registers.

12.6.11 Capture/Compare Register 1（TIMx_CCDAT1）

Offset address：0x28

Reset value：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CCDDAT1[15:0]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCDAT1[15:0]															
rw															

Bit Field	Name	Descriptions
31:16	CCDDAT1[15:0]	Capture/Compare 1 down-counting value Dedicated to center-aligned asymmetric mode • **CC1 channel can only be configured as an output：** CCDDAT1 contains the value to be compared with the counter TIMx_CNT (only when TIMx_CTRL1.DIR = 1 and in asymmetric mode), to generate a signal on the OC1 output. If the preload function is not selected in the TIMx_CCMOD1.OC1PEN bit, the written value will be immediately transferred to the active register. Otherwise, the preload value will only be transferred to the active register when an update event occurs.
15:0	CCDAT1[15:0]	Capture/Compare 1 value • **CC1 Channel configured as output:** CCDAT1 contains the value to be compared with the counter TIMx_CNT, to generate a signal on the OC1 output. If the preload function is not selected in the TIMx_CCMOD1.OC1PEN bit, the written value will be immediately transferred to the effective register. Otherwise, this preload value will only be transferred to the active register when an update event occurs. • **CC1 channel configured as input:** CCDAT1 contains the counter value transferred by the last input capture 1 event (IC1). When configured in input mode, registers CCDAT1 and CCDDAT1 can only be read. When configured in output mode, registers CCDAT1 and CCDDAT1 are readable and writable.

12.6.12 Capture/Compare Register 2（TIMx_CCDAT2）

Offset address：0x2C

Reset value：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CCDDAT2[15:0]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCDAT2[15:0]															
rw															

Bit Field	Name	Description
31:16	CCDDAT2[15:0]	Capture/Compare 2 down-counting value Dedicated to center-aligned asymmetric mode. • **CC2 channel can only be configured as an output：** CCDDAT2 contains the value to be compared with the counter TIMx_CNT (only when TIMx_CTRL1.DIR = 1 and in asymmetric mode), to generate a signal on the OC2 output. If the preload function is not selected in the TIMx_CCMOD1.OC2PEN bit, the written value will be immediately transferred to the effective register. Otherwise, this preload value will only be transferred to the active register when an update event occurs.
15:0	CCDAT2[15:0]	Capture/Compare 2 value • **CC2 channel configured as output:** CCDAT2 contains the value to be compared with the counter TIMx_CNT, to generate a signal on the OC2 output. If the preload function is not selected in the TIMx_CCMOD1.OC2PEN bit, the written value will be immediately transferred to the effective register. Otherwise, this preload value will only be transferred to the active register when an update event occurs. • **CC2 channel configured as input:** CCDAT2 contains the counter value transferred by the last input capture 2 event (IC2). When configured in input mode, registers CCDAT2 and CCDDAT2 can only be read. When configured in output mode, registers CCDAT2 and CCDDAT2 are readable and writable.

12.6.13 Capture/Compare Register 3（TIMx_CCDAT3）

Offset address：0x30

Reset value：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CCDDAT3[15:0]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCDAT3[15:0]															
rw															

Bit Field	Name	Description
31:16	CCDDAT3[15:0]	Capture/Compare 3 down-counting value **Dedicated to center-aligned asymmetric mode**. • CC3 channel can only be configured as an output： CCDDAT3 contains the value to be compared with the counter TIMx_CNT (only when TIMx_CTRL1.DIR = 1 and in asymmetric mode), to generate a signal on the OC3 output. If the preload function is not selected in the TIMx_CCMOD2.OC3PEN bit, the written value will be immediately transferred to the effective register. Otherwise, this preload value will only be transferred to the active register when an update event occurs.
15:0	CCDAT3[15:0]	Capture/Compare 3 value • **CC3 channel configured as output:** CCDAT3 contains the value to be compared with the counter TIMx_CNT, to generate a signal on the OC3 output. If the preload function is not selected in the TIMx_CCMOD2.OC3PEN bit, the written value will be immediately transferred to the effective register. Otherwise, this preload value will only be transferred to the active register when an update event occurs. • **CC3 channel configured as input:** CCDAT3 contains the counter value transferred by the last input capture 3 event (IC3). When configured in input mode, registers CCDAT3 and CCDDAT3 can only be read. When configured in output mode, registers CCDAT3 and CCDDAT3 are readable and writable.

12.6.14 Capture/Compare Register 4（TIMx_CCDAT4）

Offset address：0x34

Reset value：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CCDDAT4[15:0]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCDAT4[15:0]															
rw															

Bit Field	Name	Description
31:16	CCDDAT4[15:0]	Capture/Compare 4 down-counting value Dedicated to center-aligned asymmetric mode. • **CC4 channel can only be configured as an output:** CCDDAT4 contains the value to be compared with the counter TIMx_CNT (only when TIMx_CTRL1.DIR = 1 and in asymmetric mode), to generate a signal on the OC4 output. If the preload function is not selected in the TIMx_CCMOD2.OC4PEN bit, the written value will be immediately transferred to the effective register. Otherwise, this preload value will only be transferred to the active register when an update event occurs.

15:0	CCDAT4[15:0]	Capture/Compare 4 value • **CC4 channel configured as output:** CCDAT4 contains the value to be compared with the counter TIMx_CNT, to generate a signal on the OC4 output. If the preload function is not selected in the TIMx_CCMOD2.OC4PEN bit, the written value will be immediately transferred to the effective register. Otherwise, this preload value will only be transferred to the active register when an update event occurs. • **CC4 channel configured as input:** CCDAT4 contains the counter value transferred by the last input capture 4 event (IC4). When configured in input mode, registers CCDAT4 and CCDDAT4 can only be read. When configured in output mode, registers CCDAT4 and CCDDAT4 are readable and writable.

12.6.15 Capture/Compare Register 5（TIMx_CCDAT5）

Offset address：0x38

Reset value：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCDAT5[15:0]															
rw															

Bit Field	Name	Description
31:16	Reserved	Reserved, the reset value must be maintained
15:0	CCDAT5[15:0]	Capture/Compare 5 value • **CC5 channel can only be configured as an output:** CCDAT5 contains the value to be compared with the counter TIMx_CNT, and signals on the OC5 output. If the preload function is not selected in the TIMx_CCMOD3.OC5PEN bit, the written value will be immediately transferred to the active register. Otherwise, the preload value will only be transferred to the active register when an update event occurs. CC5 is used for comparator blanking.

12.6.16 Prescaler（TIMx_PSC）

Offset address：0x40

Reset value：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSC[15:0]															

rw

Bit Field	Name	Description
31:16	Reserved	Reserved, the reset value must be maintained
15:0	PSC[15:0]	Prescaler value Counter clock f_{CK_CNT} = f_{CK_PSC}/ (PSC [15:0] +1)。 Each time an update event occurs, the PSC value is loaded into the shadow register of the prescaler.

12.6.17 Auto-Reload Register（TIMx_AR）

Offset address:0x44

Reset value: 0x0000 FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AR[15:0]															

rw

Bit Field	Name	Description
31:16	Reserved	Reserved, the reset value must be maintained
15:0	AR[15:0]	Auto-reload value AR contains the value that will be loaded into the actual auto-reload register. Refer to section 10.5.1 for details on the update and actions of AR. When the value of auto-reload is zero, the counter does not work.

12.6.18 Counter（TIMx_CNT）

Offset address：0x48

Reset value：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UDITF CPY	Reserved														
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT[15:0]															

rw

Bit Field	Name	Description
31	UDITFCPY	UDITF copy This bit is a read-only copy of the UDITF bit in the TIMx_STS register. If the TIMx_CTRL1.UDITFREMAP bit is reset, bit 31 is reserved and reads as 0.

Bit Field	Name	Description
30:16	Reserved	Reserved, the reset value must be maintained
15:0	CNT[15:0]	Counter value

12.6.19 Repetition Counter Register（TIMx_REPCNT）

Offset address：0x4C

Reset value：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								REPCNT[7:0]							
								rw							

Bit Field	Name	Description
31:8	Reserved	Reserved, the reset value must be maintained
7:0	REPCNT[7:0]	Repetition counter value The repetition counter is used to generate an update event or update the timer registers after a given number (N+1) of counter periods, where N is the value of TIMx_REPCNT.REPCNT. In up-counting mode, the repetition counter decrements on each counter overflow, in down-counting mode it decrements on each counter underflow, and in center-aligned mode it decrements on each counter overflow and underflow. Setting the TIMx_EVTGEN.UDGN bit reloads the content of TIMx_REPCNT.REPCNT and generates an update event.

12.6.20 Break and Deadtime Register（TIMx_BKDT）

Offset address：0x50

Reset value：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved											BRKBID	Reserved	BRK DSRM	Reserved	
											rw		rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LCKCFG[1:0]		OSSR	OSSI	BKEN	BKP	AOEN	MOEN	DTGN[7:0]							
rw		rw	rw	rw	rw	rw	rw	rw							

Note: Depending on the lock settings, the BRK2BID, BRKBID, BK2EN, BK2P, AOEN, BKP, BKEN, OSSI, OSSR, and DTGN[7:0] bits can be write-protected. It is necessary to configure them when writing to the TIMx_BKDT register for the first time.

Bit Field	Name	Description
31:21	Reserved	Reserved, the reset value must be maintained

20	BRKBID	Break 1 bidirectional enable 0: Break 1 is the input mode 1: Break 1 is the bidirectional mode In bidirectional mode, Break 1 input is configured to operate in both input and open-drain output modes. Any Break 1 event will generate a low level on the Break 1 input IO, indicating to external devices that an internal Break 1 event has occurred. *Note: Once the LOCK level (LCKCFG bit in the TIMx_BKDT register) is set to '1', this bit cannot be modified.* *Note: Any write operation to this bit will take effect after a delay of an APB clock cycle.*
18	BRKDSRM	Break 1 disarm 0: Break 1 input preparation 1: Break 1 input disarm When the Break 1 input is invalid, this bit will be automatically cleared by the hardware. BRKDSRM is set by software to release Break 1 bidirectional output control (open-drain output in high-impedance state), then software polls this bit until it is reset by hardware, indicating that the Break 1 event has disappeared. *Note: Any write operation to this bit will take effect after a delay of an APB clock cycle.*
15:14	LCKCFG[1:0]	Lock configuration These bits provide write protection against software errors. 00:– No write protection. 01:– LOCK Level 1 TIMx_BKDT.DTGN、TIMx_BKDT.BKEN、TIMx_BKDT.BKP、TIMx_BKDT.AOEN、TIMx_CTRL2.OIx、TIMx_CTRL2.OIxN enable write protection. 10: – LOCK Level 2 In addition to register write protection in LOCK Level 1 mode, write protection is also enabled for TIMx_CCEN.CCxP and TIMx_CCEN.CCxNP (if the corresponding channel is configured in output mode), as well as for the TIMx_BKDT.OSSR and TIMx_BKDT.OSSI bits. 11:– LOCK Level 3 In addition to register write protection in LOCK Level 2, write protection is also enabled for TIMx_CCMODx.OCxMD and TIMx_CCMODx.OCxPEN bits (if the corresponding channel is configured in output mode). *Note: After system reset, the LCKCFG bit can only be written once. Once written to the TIMx_BKDT register, LCKCFG will be protected until the next reset.*
13	OSSR	This bit is used when TIMx_BKDT.MOEN=1 and the channel is configured as complementary output. There is no OSSR bit in timers without complementary output. 0: When the timer is not operating, disable OC/OCN output (OC/OCN enable output signal = 0)； 1: When the timer is not operating, if CCxEN=1 or CCxNEN=1, OC/OCN will first be enabled to output an invalid level, and then set OC/OCN enable output signal to 1. For more detailed information, please refer to Section10.6.9, Capture/Compare Enable Register (TIMx_CCEN).
12	OSSI	Off-state selection for Idle mode This bit is used when TIMx_BKDT.MOEN=0 and the channel is configured as an output.

		0: When the timer is not operating, disable OC/OCN output (OC/OCN enable output signal = 0); 1: When the timer is not operating, if CCxEN=1 or CCxNEN=1, OC/OCN first outputs its idle level, and then OC/OCN enables the output signal to 1. For more detailed information, please refer to Section10.6.9, Capture/Compare Enable Register (TIMx_CCEN).

11	BKEN	Break 1 enable
		0: Disable Break 1 input;
		1: Enable Break 1 input;
		Note: Any write operation to this bit will take effect after a delay of an APB clock cycle

10	BKP	Break 1 polarity 0: Break 1 input low level effective; 1: Break 1 input high level effective; *Note: Any write operation to this bit will take effect after a delay of one APB clock cycle*
9	AOEN	Automatic output enable 0: Only software can set TIMx_BKDT.MOEN; 1: Software sets TIMx_BKDT.MOEN; or if the break input is not activated, hardware automatically sets TIMx_BKDT.MOEN at the next update event.
8	MOEN	Main output enable This bit can be set by software or hardware based on the TIMx_BKDT.AOEN bit. Once the break input is valid, this bit is asynchronously cleared by hardware. It is only effective for channels configured as outputs. 0: OC and OCN outputs are disabled or forced into the idle state。 1: If the TIMx_CCEN.CCxEN or TIMx_CCEN.CCxNEN bits are set, the OC and OCN outputs are enabled. For more detailed information, please refer to Section 10.6.9, Capture/Compare Enable Register (TIMx_CCEN).
7:0	DTGN[7:0]	Dead-time generator setup These bits define the dead-time duration between the inserted complementary outputs. The relationship between the DTGN value and the dead-time is as follows: DTGN[7:5]=0xx => DT=DTGN[7:0] × T_{dtgn}， $T_{dtgn} = T_{DTS}$; DTGN[7:5]=10x => DT=(64+DTGN[5:0]) × T_{dtgn}， $T_{dtgn} = 2 \times T_{DTS}$; DTGN[7:5]=110 => DT=(32+DTGN[4:0]) × T_{dtgn}， $T_{dtgn} = 8 \times T_{DTS}$; DTGN[7:5]=111 => DT=(32+DTGN[4:0])× T_{dtgn}， $T_{dtgn} = 16 \times T_{DTS}$;

12.6.21 Break 1 Filter Register（TIMx_BKFR）

Offset address：0x60

Reset value：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		THRESH[5:0]						Reserved	WSIZE[5:0]						FILTEN
		rw							rw						rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Reserved

Bit Field	Name	Description
31:30	Reserved	Reserved, the reset value must be maintained
29:24	THRESH[5:0]	Threshold: The number of valid logic levels in the sampling logic, with a maximum of 63. Within the sampling window, if the number of logic highs is greater than or equal to the threshold, the next logic level will be high. The same rule applies to logic lows. If the number of 1s and 0s within the window is less than the threshold, the filter output remains unchanged. The threshold should be set to be greater than or equal to half of the Window value. The recommended threshold range is: **Minimum value:** It is one additional prescaler clock cycle beyond the upper limit of the maximum glitch size (prescaler clock cycles) and must be greater than half of the window size. For example, if the glitch size is 3.2 * (prescaler clock cycles), the threshold should be ┌3.2┐=4+1=5. **Maximum value:** It is the lower limit of the minimum valid signal size (in prescaler clock cycles) and must be less than the window size. For example, if the minimum signal size is 3.2 * (prescaler clock cycles), the threshold should be the floor value of 3.2, which is 3.
23	Reserved	Reserved, the reset value must be maintained
22:17	WSIZE[5:0]	The window size value for logic level checking, with a maximum of 63: The window size determines how many samples will be considered when obtaining the next logic level. The built-in FIFO is 64 bits with a maximum index of 63, so the window size can only be set to 63.
16	FILTEN	Filter enable 0: Filter disable 1: Filter enable
15:0	Reserved	Reserved, the reset value must be maintained

12.6.22 Channel 1 Filter Register（TIMx_C1FILT）

Offset address：0x64

Reset value：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		THRESH[5:0]						Reserved	WSIZE[5:0]						FILTEN
		rw							rw						rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															

Bit Field	Name	Description
31:30	Reserved	Reserved, the reset value must be maintained
29:24	THRESH[5:0]	Threshold: The number of valid logic levels in the sampling logic, with a maximum of 63. Within the sampling window, if the number of logic highs is greater than or equal to the

		threshold, the next logic level will be high. The same rule applies to logic lows. If the number of 1s and 0s within the window is less than the threshold, the filter output remains unchanged. The threshold should be set to be greater than or equal to half of the Window value. The recommended threshold range is: **Minimum value:** It is one additional prescaler clock cycle beyond the upper limit of the maximum glitch size (prescaler clock cycles) and must be greater than half of the window size. For example, if the glitch size is 3.2 * (prescaler clock cycles), the threshold should be ┌3.2┐ =4+1=5. **Maximum value:** It is the lower limit of the minimum valid signal size (in prescaler clock cycles) and must be less than the window size. For example, if the minimum signal size is 3.2 * (prescaler clock cycles), the threshold should be the floor value of 3.2, which is 3.
23	Reserved	Reserved, the reset value must be maintained
22:17	WSIZE[5:0]	The window size value for logic level checking, with a maximum of 63: The window size determines how many samples will be considered when obtaining the next logic level. The built-in FIFO is 64 bits with a maximum index of 63, so the window size can only be set to 63.
16	FILTEN	Filter enable 0: Filter disable 1: Filter enable
15:0	Reserved	Reserved, the reset value must be maintained

12.6.23 Channel 2 Filter Register（TIMx_C2FILT）

Offset address：0x68

Reset value：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		THRESH[5:0]						Reserved	WSIZE[5:0]						FILTEN
		rw							rw						rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															

Bit Field	Name	Description
31:30	Reserved	Reserved, the reset value must be maintained
29:24	THRESH[5:0]	Threshold: The number of valid logic levels in the sampling logic, with a maximum of 63. Within the sampling window, if the number of logic highs is greater than or equal to the threshold, the next logic level will be high. The same rule applies to logic lows. If the number of 1s and 0s within the window is less than the threshold, the filter output remains unchanged. The threshold should be set to be greater than or equal to half of the Window value. The recommended threshold range is: **Minimum value:** It is one additional prescaler clock cycle beyond the upper limit of the maximum glitch size (prescaler clock cycles) and must be greater than half of the window size.

		For example, if the glitch size is 3.2 * (prescaler clock cycles), the threshold should be ┌3.2┐ =4+1=5. **Maximum value:** It is the lower limit of the minimum valid signal size (in prescaler clock cycles) and must be less than the window size. For example, if the minimum signal size is 3.2 * (prescaler clock cycles), the threshold should be the floor value of 3.2, which is 3.
23	Reserved	Reserved, the reset value must be maintained
22:17	WSIZE[5:0]	The window size value for logic level checking, with a maximum of 63: The window size determines how many samples will be considered when obtaining the next logic level. The built-in FIFO is 64 bits with a maximum index of 63, so the window size can only be set to 63.
16	FILTEN	Filter enable 0: Filter disable 1: Filter enable
15:0	Reserved	Reserved, the reset value must be maintained

12.6.24 Channel 3 Filter Register（TIMx_C3FILT）

Offset address：0x6C

Reset value：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		THRESH[5:0]						Reserved	WSIZE[5:0]						FILTEN
		rw							rw						rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															

Bit Field	Name	Description
31:30	Reserved	Reserved, the reset value must be maintained
29:24	THRESH[5:0]	Threshold: The number of valid logic levels in the sampling logic, with a maximum of 63. Within the sampling window, if the number of logic highs is greater than or equal to the threshold, the next logic level will be high. The same rule applies to logic lows. If the number of 1s and 0s within the window is less than the threshold, the filter output remains unchanged. The threshold should be set to be greater than or equal to half of the Window value. The recommended threshold range is: **Minimum value:** It is one additional prescaler clock cycle beyond the upper limit of the maximum glitch size (prescaler clock cycles) and must be greater than half of the window size. For example, if the glitch size is 3.2 * (prescaler clock cycles), the threshold should be ┌3.2┐ =4+1=5. **Maximum value:** It is the lower limit of the minimum valid signal size (in prescaler clock cycles) and must be less than the window size. For example, if the minimum signal size is 3.2 * (prescaler clock cycles), the threshold should be the floor value of 3.2, which is 3.
23	Reserved	Reserved, the reset value must be maintained

22:17	WSIZE[5:0]	The window size value for logic level checking, with a maximum of 63: The window size determines how many samples will be considered when obtaining the next logic level. The built-in FIFO is 64 bits with a maximum index of 63, so the window size can only be set to 63.
16	FILTEN	Filter enable 0: Filter disable 1: Filter enable
15:0	Reserved	Reserved, the reset value must be maintained

12.6.25 Channel 4 Filter Register（TIMx_C4FILT）

Offset address：0x70

Reset value：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		THRESH[5:0]						Reserved	WSIZE[5:0]						FILTEN
		rw							rw						rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															

Bit Field	Name	Description
31:30	Reserved	Reserved, the reset value must be maintained
29:24	THRESH[5:0]	Threshold: The number of valid logic levels in the sampling logic, with a maximum of 63. Within the sampling window, if the number of logic highs is greater than or equal to the threshold, the next logic level will be high. The same rule applies to logic lows. If the number of 1s and 0s within the window is less than the threshold, the filter output remains unchanged. The threshold should be set to be greater than or equal to half of the Window value. The recommended threshold range is: **Minimum value:** It is one additional prescaler clock cycle beyond the upper limit of the maximum glitch size (prescaler clock cycles) and must be greater than half of the window size. For example, if the glitch size is 3.2 * (prescaler clock cycles), the threshold should be ┌3.2┐ =4+1=5. **Maximum value:** It is the lower limit of the minimum valid signal size (in prescaler clock cycles) and must be less than the window size. For example, if the minimum signal size is 3.2 * (prescaler clock cycles), the threshold should be the floor value of 3.2, which is 3.
23	Reserved	Reserved, the reset value must be maintained
22:17	WSIZE[5:0]	The window size value for logic level checking, with a maximum of 63: The window size determines how many samples will be considered when obtaining the next logic level. The built-in FIFO is 64 bits with a maximum index of 63, so the window size can only be set to 63.
16	FILTEN	Filter enable 0: Filter disable 1: Filter enable

15:0	Reserved	Reserved, the reset value must be maintained

12.6.26 Input Channel Filter Output Register（TIMx_FILTO）

Offset address：0x74

Reset value：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												C4FILTO	C3FILTO	C2FILTO	C1FILTO
												r	r	r	r

Bit Field	Name	Description
31:4	Reserved	Reserved, the reset value must be maintained
3	C4FILTO	Channel 4 filter output level status 0: Output low level 1: Output high level
2	C3FILTO	Channel 3 filter output level status 0: Output low level 1: Output high level
1	C2FILTO	Channel 2 filter output level status 0: Output low level 1: Output high level
0	C1FILTO	Channel 1 filter output level status 0: Output low level 1: Output high level

12.6.27 Input Selection Register（TIMx_INSEL）

Note:For detailed selection of signal sources, please refer to Chapter 12.4.

Offset address：0x78

Reset value：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				CLRS[3:0]				ITRS[3:0]				ETRS[3:0]			
				rw				rw				rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TI4S[3:0]				TI3S[3:0]				TI2S[3:0]				TI1S[3:0]			
rw				rw				rw				rw			

Bit Field	Name	Description
31:28	Reserved	Reserved, the reset value must be maintained

27:24	CLRS[3:0]	Selects tim_ocref_clr[0:15] input signal 0000: tim_ocref_clr0 0001: tim_ocref_clr1 … 1111: tim_ocref_clr15 *Note: Once the LOCK level (LCKCFG bit in the TIMx_BKDT register) is set to '1', this bit cannot be modified.*
23:20	ITRS[3:0]	Selects tim_itr[0:15] input signal 0000: tim_itr0 0001: tim_itr1 … 1111: tim_itr15
19:16	ETRS[3:0]	Selects tim_etr[0:15] input signal 0000: tim_etr0 0001: tim_etr1 … 1111: tim_etr15 *Note: Once the LOCK level (LCKCFG bit in the TIMx_BKDT register) is set to '1', this bit cannot be modified.*
15:12	TI4S[3:0]	Selects tim_ti4[0:15] input signal 0000: tim_ti4_in0 0001: tim_ti4_in1 … 1111: tim_ti4_in15
11:8	TI3S[3:0]	Selects tim_ti3[0:15] input signal 0000: tim_ti3_in0 0001: tim_ti3_in1 … 1111: tim_ti3_in15
7:4	TI2S[3:0]	Selects tim_ti2[0:15] input signal 0000: tim_ti2_in0 0001: tim_ti2_in1 … 1111: tim_ti2_in15
3:0	TI1S[3:0]	Select tim_ti1[0:15] input signal 0000: tim_ti1_in0 0001: tim_ti1_in1 … 1111: tim_ti1_in15

12.6.28 Alternate Function Register 1（TIMx_AF1）

Offset address：0x7C

Reset value：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		COMP4 BRKP	COMP3 BRKP	COMP2 BRKP	COMP1 BRKP	IOM BRKP	Reserved	COMP7 BRKEN	COMP6 BRKEN	COMP5 BRKEN	COMP4 BRKEN	COMP3 BRKEN	COMP2 BRKEN	COMP1 BRKEN	IOM BRKEN
		rw	rw	rw	rw	rw		rw	rw	rw	rw	rw	rw	rw	rw

Bit Field	**Name**	**Description**
31:14	Reserved	Reserved, the reset value must be maintained
13	COMP4BRKP	tim_brk_comp4 Break Input Polarity Selection 0: tim_brk_comp4 Break Input Polarity Non-Inverted (Low level effective if BKP=0; High level effective if BKP=1) 1: tim_brk_comp4 Brake Input Polarity Inverted (High level effective if BKP=0; Low level effective if BKP=1)
12	COMP3BRKP	tim_brk_comp3 Break Input Polarity Selection 0: tim_brk_comp3 Break Input Polarity Non-Inverted (Low level effective if BKP=0; High level effective if BKP=1) 1: tim_brk_comp3 Brake Input Polarity Inverted (High level effective if BKP=0; Low level effective if BKP=1)
11	COMP2BRKP	tim_brk_comp2 Break Input Polarity Selection 0: tim_brk_comp2 Break Input Polarity Non-Inverted (Low level effective if BKP=0; High level effective if BKP=1) 1: tim_brk_comp2 Brake Input Polarity Inverted (High level effective if BKP=0; Low level effective if BKP=1)
10	COMP1BRKP	tim_brk_comp1 Break Input Polarity Selection 0: tim_brk_comp1 Break Input Polarity Non-Inverted (Low level effective if BKP=0; High level effective if BKP=1) 1: tim_brk_comp2 Brake Input Polarity Inverted (High level effective if BKP=0; Low level effective if BKP=1)
9	IOMBRK	TIMx_BKIN Brake Input Polarity Selection 0: TIMx_BKIN Break Input Polarity Non-Inverted (Low level effective if BKP=0; High level effective if BKP=1) 1: TIMx_BKIN Break Input Polarity Inverted (High level effective if BKP=0; Low level effective if BKP=1)
8	Reserved	Reserved, the reset value must be maintained
7	COMP7BRKEN	tim_brk_comp7 break input enable 0: tim_brk_comp7 break input disable 1: tim_brk_comp7 break input enable
6	COMP6BRKEN	tim_brk_comp6 break input enable 0: tim_brk_comp6 break input disable 1: tim_brk_comp6 break input enable

5	COMP5BRKEN	tim_brk_comp5 break input enable 0: tim_brk_comp5 break input disable 1: tim_brk_comp5 break input enable
4	COMP4BRKEN	tim_brk_comp4 break input enable 0: tim_brk_comp4 break input disable 1: tim_brk_comp4 break input enable
3	COMP3BRKEN	tim_brk_comp3 break input enable 0: tim_brk_comp3 break input disable 1: tim_brk_comp3 break input enable
2	COMP2BRKEN	tim_brk_comp2 break input enable 0: tim_brk_comp2 break input disable 1: tim_brk_comp2 break input enable
1	COMP1BRKEN	tim_brk_comp1 break input enable 0: tim_brk_comp1 break input disable 1: tim_brk_comp1 break input enable
0	IOMBRKEN	TIMx_BKIN break input enable 0: TIMx_BKIN break input disable 1: TIMx_BKIN break input enable

12.6.29 Sliding Filter Prescaler Register（TIMx_SLIDFPSC）

Offset address：0x88

Reset value：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SLIDFPSC [15:0]															
rw															

Bit Field	Name	Description
31:16	Reserved	Reserved, the reset value must be maintained
15:0	SLIDFPSC [15:0]	Sliding Filter Sampling Clock Prescaler Register (Prescaler): For this filter, it supports a division of 65535 (16 bits). The clock prescaler scales the system clock to the sampling clock. The sampling clock determines the distance between two sampling points. Only the values of the sampling points are considered for logic level calculation. Configure these bits to determine the sampling clock division for the channel 1/2/3/4 sliding filter and the break input sliding filter, in conjunction with TIMx_CH1FILT, TIMx_CH2FILT, TIMx_CH3FILT, TIMx_CH4FILT, and TIMx_BKFR.

12.6.30 DMA Control Register（TIMx_DCTRL）

Offset address：0x94

Reset value：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		DBADDR[5:0]						Reserved		DBLEN[5:0]					
		rw								rw					

Bit Field	Name	Description
31:14	Reserved	Reserved, the reset value must be maintained
13:8	DBADDR[5:0]	DMA base address This field defines the first address for DMA access to the TIMx_DADDR register. When accessing TIMx_DADDR for the first time, this field specifies the address that was just accessed. Subsequent accesses to TIMx_DADDR will access the address of "DMA Base Address + 4" 00000: TIMx_CTRL1 00001: TIMx_CTRL2 00010: TIMx_SMCTRL ...... 10001: TIMx_BKDT 10010: TIMx_DCTRL
7:6	Reserved	Reserved, the reset value must be maintained
5:0	DBLEN[5:0]	DMA burst length This field defines the number of times DMA will access (write/read) the TIMx_DADDR register. 000000: One time transfer 000001: Two times transfer 000010: Three times transfer ...... 010001: Eighteen times transfer ...... 100010: Thirty-five times transfers For example, we consider the following transfer: DBLEN=7, DBADDR=TIMx_CTRL1 If DBLEN=7 and DBADDR=TIMx_CTRL1 represents the address of the data to be transferred, then the address of the transfer is given by the following formula: ((Address of TIMx_CTRL1) + DBADDR + (DMA index)), where DMA index = DBLEN. Adding 7 to ((Address of TIMx_CTRL1) + DBADDR) gives the address where data will be written to or read from. This means the data transfer will occur in the 7 registers starting from the address ((Address of TIMx_CTRL1) + DBADDR). If the data is set to half-word (16 bits), then the data will be transferred to all 7 registers. If the data is set to byte, the data will still be transferred to all 7 registers: the first register contains the first MSB byte, the second register contains the first LSB byte, and so on.

Bit Field	Name	Description
		Therefore, for the timer, the user must specify the data width to be transferred by DMA.

12.6.31 Continuous Mode DMA Address（TIMx_DADDR）

Offset address：0x98

Reset value：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BURST[31:0]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BURST[31:0]															
rw															

Bit Field	Name	Description
31:0	BURST[31:0]	DMA access buffer When a read or write operation is assigned to this register, the registers located at the address range (DMA base address + DMA burst length × 4) will be accessed. DMA base address = The address of TIM_CTRL1 + TIMx_DCTRL. DBADDR * 4； DMA burst len = TIMx_DCTRL.DBLEN + 1. Example： If TIMx_DCTRL.DBLEN = 0x3 (4 transfers), TIMx_DCTRL.DBADDR = 0xD (TIMx_CCDAT1), DMA data length = half-word, DMA memory address = buffer address in SRAM, and DMA peripheral address = TIMx_DADDR address. When the event occurs, TIMx will send a request to DMA and transfer data 4 times. The first time, DMA access to the TIMx_DADDR register will be mapped to access the TIMx_CCDAT1 register; The second time, DMA access to the TIMx_DADDR register will be mapped to access the TIMx_CCDAT2 register; …… The fourth time, DMA access to the TIMx_DADDR register will be mapped to access the TIMx_CCDAT4 register;

13 Basic Timers (BTIM1/BTIM2)

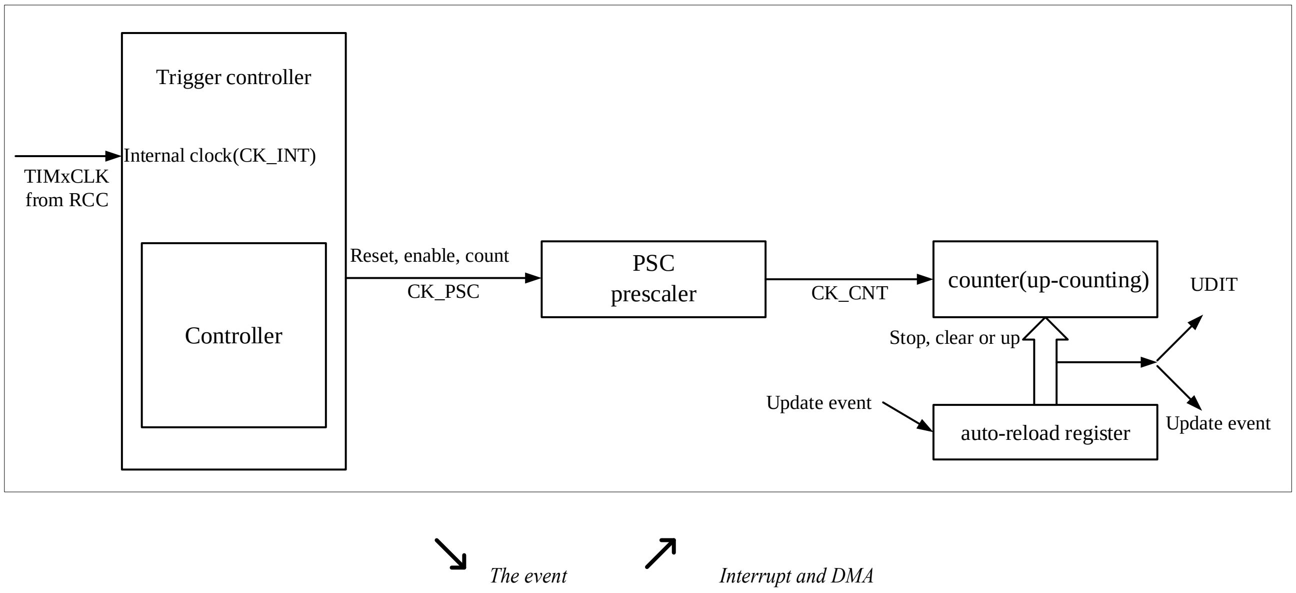
13.1 BTIMx（x=1-2）Introduction

Basic timers BTIM1 and BTIM2 each contain a 32-bit auto-reload counter.

13.2 BTIMx（x=1-2）Main Features

- 32-bit auto-reload up-counting counter.
- 16-bit programmable prescaler (The prescaler factor can be configured with any value between 1 and 65536).
- The events that generate the interrupt/DMA are as follows:
  - Update event

Figure 13-1 Block Diagram Of BTIMx (x = 1,2)



13.3 BTIMx (x=1-2) Description

13.3.1 Time-base Unit

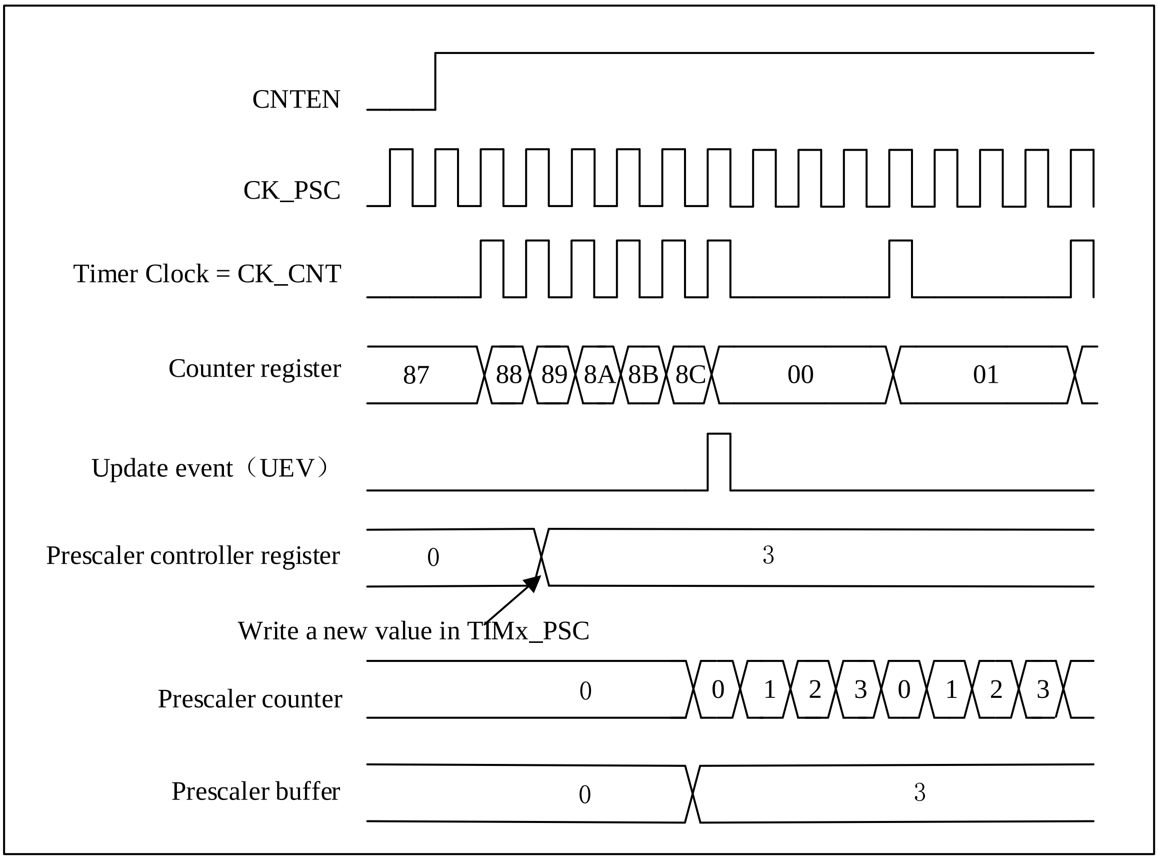
The time-base unit mainly includes: prescaler, counter and auto-reload register. When the time base unit is working, the software can read and write the corresponding registers (TIMx_PSC, TIMx_CNT and TIMx_AR) at any time.

Depending on the setting of the auto-reload preload enable bit (TIMx_CTRL1.ARPEN), the value of the preload register is transferred to the shadow register immediately or at each update event UEV. When TIMx_CTRL1.UPDIS=0, an update event is generated when the counter reaches the overflow condition, or when the TIMx_EVTGEN.UDGN bit is set by software. The counter CK_CNT is valid only when the TIMx_CTRL1.CNTEN bit is set. The counter starts counting one clock cycle after the TIMx_CTRL1.CNTEN bit is set.

13.3.1.1 Prescaler Description

The TIMx_PSC register consists of a 16-bit counter that can be used to divide the counter clock frequency by any factor between 1 and 65536. It can be changed on the fly as it is buffered. The prescaler value is only taken into account at the next update event.

Figure 13-2 Counter Timing Diagram With Prescaler Division Change From 1 To 4



13.3.2 Counting Mode

13.3.2.1 Up-counting mode

In up-counting mode, the counter will count from 0 to the value of the register TIMx_AR, then it resets to 0. And a counter overflow event is generated.

If the TIMx_CTRL1.UPRS bit (select update request) and the TIMx_EVTGEN.UDGN bit are set, an update event (UEV) will generate, but TIMx_STS.UDITF will not be set by hardware. Therefore, no update interrupts or DMA update requests are generated. This setting is used in scenarios where you want to clear the counter but do not want to generate an update interrupt.

Depending on the update request source is configured in TIMx_CTRL1.UPRS, when an update event occurs, TIMx_STS.UDITF is set and all registers are updated：

- Update auto-reload shadow registers with preload value(TIMx_AR), when TIMx_CTRL1.ARPEN = 1.
- The prescaler shadow register is reloaded with the preload value(TIMx_PSC).

To avoid updating the shadow registers when new values are written to the preload registers, you can disable the update by setting TIMx_CTRL1.UPDIS=1.

In this way,when an update event occurs, the counter and the prescaler counter will also be reset to 0 (but the prescaler rate will remain unchanged).

The figure below shows some examples of the counter behavior and the update flags for different clock frequencies in the up-counting mode.

Figure 13-3 Timing Diagram Of Up-Counting. The Internal Clock Divider Factor = 2/N

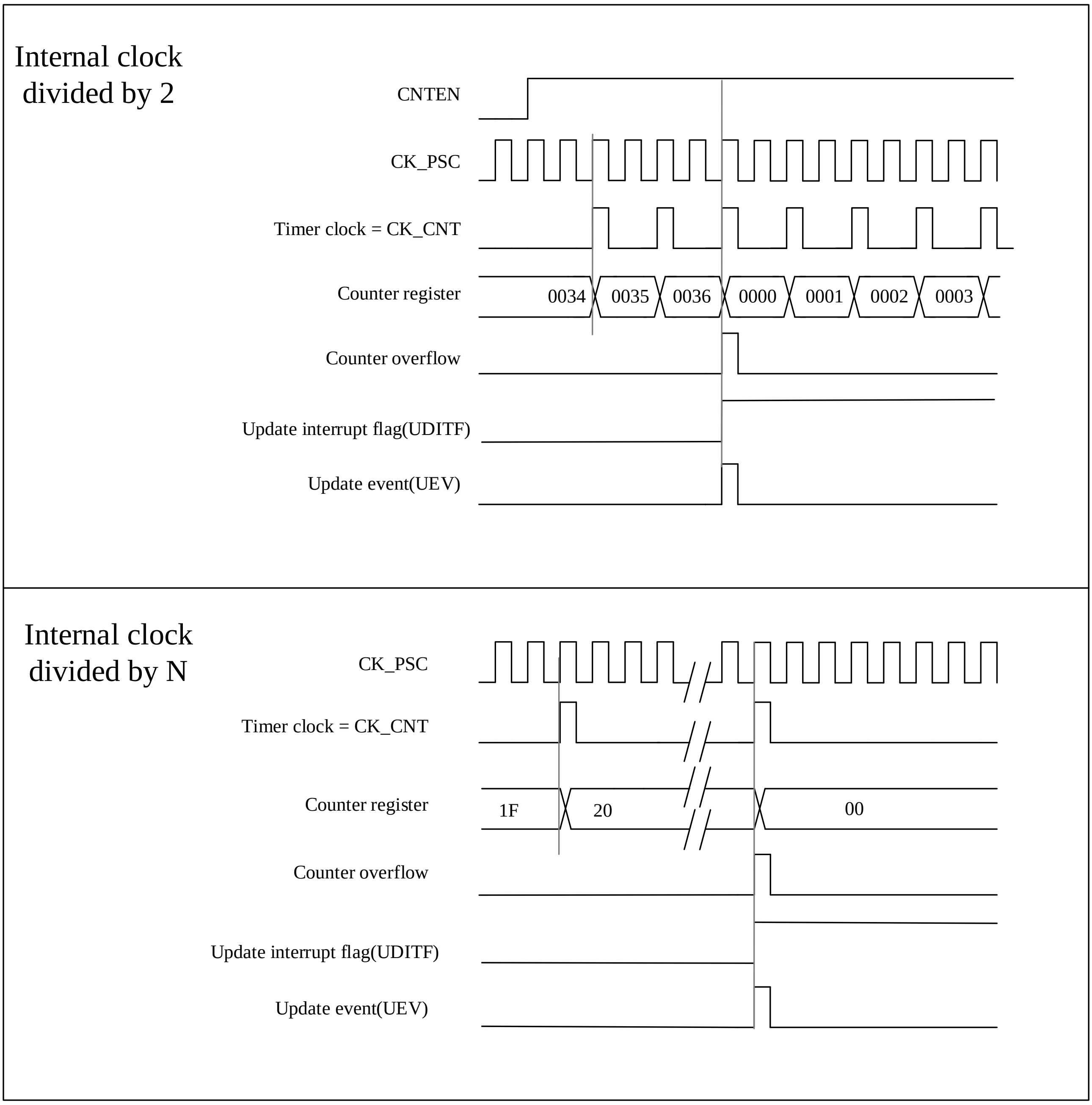
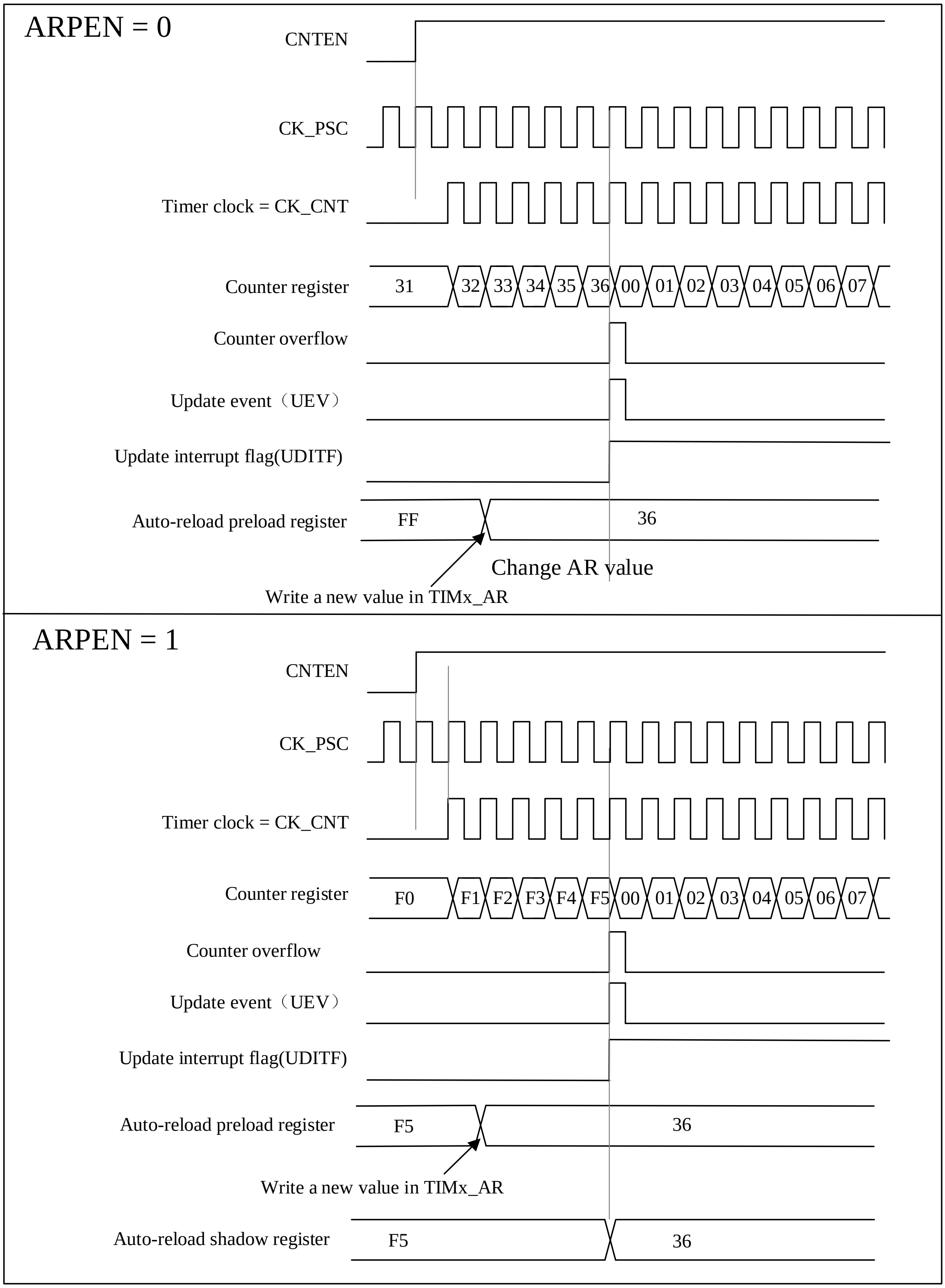


Figure 13-4 Timing diagram of up-counting and update event when ARPEN=0/1



13.3.3 Clock Selection

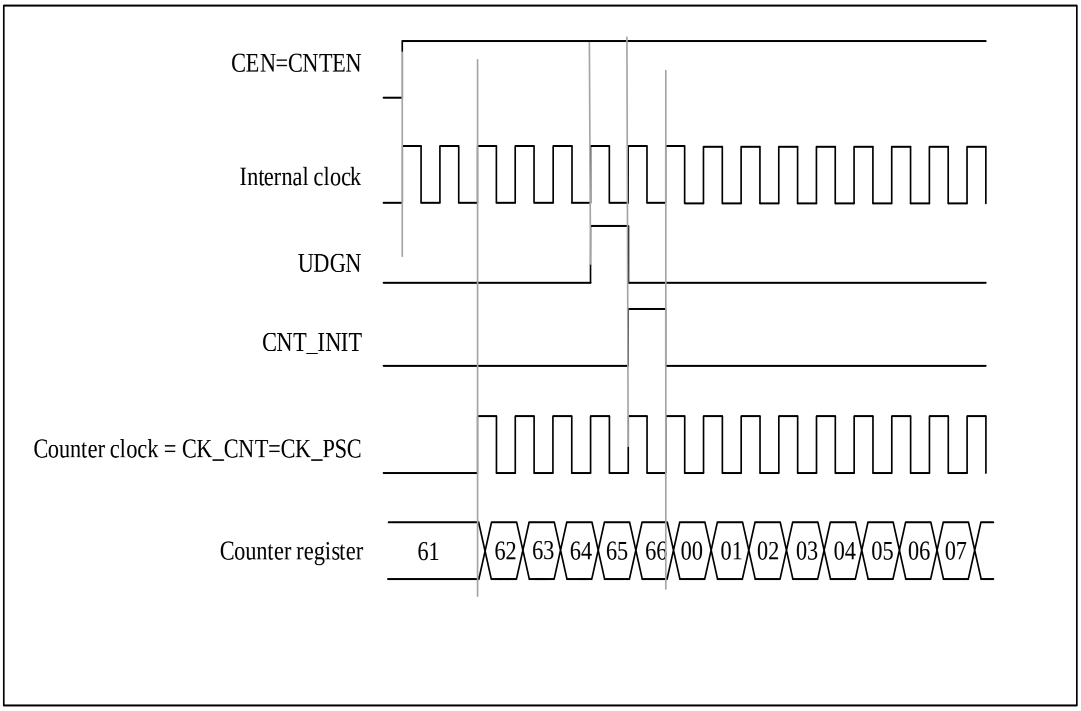
- The internal clock of timers : CK_INT

Note: The maximum operating clock for BTIM1-2 is 180MHz. Therefore, if the AHB is greater than 180MHz and BTIM1-2 is needed, the APB1 cannot be divided by 1 or 2. For example, with an AHB frequency of 240MHz and the need to use BTIM1, the APB1 can only be divided by 4 to achieve 60MHz, resulting in the BTIM1 operating at 120MHz.

13.3.3.1 Internal clock source (CK_INT)

It is provided that the TIMx_CTRL1.CNTEN bit is written as '1' by software, the clock source of the prescaler is provided by the internal clock CK_INT.

Figure 13-5 Control Circuit In Normal Mode, Internal Clock Divided By 1



13.3.4 Debug Mode

When the microcontroller is in debug mode (the Cortex^{TM}-M4F core halted), depending on the DBG_CTRL.TIMx_STOP configuration in the DBG module, the TIMx counter can either continue to work normally or stop. For more details, see Section37.3.2.

13.4 BTIMx(x=1-2) Register Description

For abbreviations used in registers, see Section 1.1

These peripheral registers can be operated as half word (16-bits) or one word (32-bits).

13.4.1 Register Overview

13.4.2 Control Register 1 (TIMx_CTRL1)

Offset address: 0x00

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						ARPEN	ONEPM	Reserved		UPDIS	UPRS	Reserved			CNTEN
						rw	rw			rw	rw				rw

Bit Field	Name	Description
31:10	Reserved	Reserved, the reset value must be maintained
9	ARPEN	Auto-reload preload enable 0: Shadow register disable for TIMx_AR register 1: Shadow register enable for TIMx_AR register
8	ONEPM	One pulse mode 0: Disable one-pulse mode, the counter counts are not affected when an update event occurs. 1: Enable one-pulse mode, the counter stops counting when the next update event occurs (clearing TIMx_CTRL1.CNTEN bit).
7:6	Reserved	Reserved, the reset value must be maintained
5	UPDIS	Update disable This bit is used to enable/disable the Update event (UEV) events generation by software. 0: Enable UEV. UEV will be generated if one of following condition been fulfilled： – Counter overflow – The TIMx_EVTGEN.UDGN bit is set Shadow registers will update with preload value。 1: Disabled UEV. No update event is generated, and the shadow registers (AR, PSC) keep their values. If the TIMx_EVTGEN.UDGN bit is set, the counter and prescaler are reinitialized.
4	UPRS	Update request source This bit is used to select the UEV event sources by software. 0: If update interrupt or DMA request is enabled, any of the following events will generate an update interrupt or DMA request： – Counter overflow – The TIMx_EVTGEN.UDGN bit is set 1: If update interrupt or DMA request is enabled, only counter overflow will generate update interrupt or DMA request

3:1	Reserved	Reserved, the reset value must be maintained
0	CNTEN	Counter enable 0: Disable counter 1: Enable counter

13.4.3 Control Register 2（TIMx_CTRL2）

Offset address：0x04

Reset value：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MMSEL[3:0]				Reserved											
rw															

Bit Field	Name	Description
31:16	Reserved	Reserved, the reset value must be maintained
15:12	MMSEL[3:0]	Master Mode Selection These 4 bits are used to select the synchronization information (TRGO) sent to the slave timer in the master mode. Possible combinations are as follows： x000: Reset –When the TIMx_EVTGEN.UDGN is set or a reset is generated by the slave mode controller, a TRGO pulse occurs. And in the latter case, the signal on TRGO is delayed compared to the actual reset. x001: Enable - The TIMx_CTRL1.CNTEN bit is used as the trigger output (TRGO). Sometimes you need to start multiple timers at the same time or enable slave timer for a period of time. The counter enable signal is set when TIMx_CTRL1.CNTEN bit is set or the trigger input in gated mode is high. When the counter enable signal is controlled by the trigger input, there is a delay on TRGO unless the master/slave mode is selected (refer to the description of the TIMx_SMCTRL.MSMD bit). x010: Update - The update event is selected as the trigger output (TRGO). For example, a master timer clock can be used as a slave timer prescaler. other：Reserved.
11:0	Reserved	Reserved, the reset value must be maintained

13.4.4 Status Registers (TIMx_STS)

Offset address: 0x08

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

Reserved	UDITF
	rc_w0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															

Bit Field	Name	Description
31:17	Reserved	Reserved, the reset value must be maintained
16	UDITF	Update interrupt flag This bit is set by hardware when an update event occurs under the following conditions： –When TIMx_CTRL1.UPDIS = 0, and counter value overflow。 –When TIMx_CTRL1.UPRS = 0, TIMx_CTRL1.UPDIS = 0, and set the TIMx_EVTGEN.UDGN bit by software to reinitialize the CNT。 This bit is cleared by software. 0: No update event occurred 1: Update interrupt occurred
15:0	Reserved	Reserved, the reset value must be maintained

13.4.5 Event Generation Register (TIMx_EVTGEN)

Offset address: 0x0C

Reset value: 0 x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved							UDGN	Reserved							
							w								

Bit Field	Name	Description
31:9	Reserved	Reserved, the reset value must be maintained.
8	UDGN	Update generation Software can set this bit to update configuration register value and hardware will clear it automatically. 0: No effect. 1: Timer will restart and all shadow register will be updated. It will restart prescaler counter also.
7:0	Reserved	Reserved, the reset value must be maintained.

13.4.6 DMA/Interrupt Enable Register (TIMx_DINTEN)

Offset address: 0x14

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved												UDEN	Reserved		UIEN
												rw			rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															

Bit Field	Name	Description
31:20	Reserved	Reserved, the reset value must be maintained
19	UDEN	Update DMA request enable 0: Disable update DMA request 1: Enable update DMA request
18:17	Reserved	Reserved, the reset value must be maintained
16	UIEN	Update interrupt enable 0: Disable update interrupt 1: Enable update interrupt
15:0	Reserved	Reserved, the reset value must be maintained

13.4.7 Prescaler (TIMx_PSC)

Offset address: 0x40

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSC[15:0]															
rw															

Bit field	Name	Description
31:16	Reserved	Reserved, the reset value must be maintained
15:0	PSC[15:0]	Prescaler value PSC register value will be updated to prescaler register at update event. Counter clock frequency is input clock frequency divide PSC + 1。

13.4.8 Automatic Reload Register (TIMx_AR)

Offset address: 0x44

Reset value: 0xFFFF FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AR[31:0]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AR[31:0]															
rw															

Bit field	Name	Description
31:0	AR[31:0]	Auto-reload value These bits define the value that will be loaded into the actual auto-reload register. See 13.3.1 for more details。 When the TIMx_AR.AR [15:0] value is null, the counter does not work.

13.4.9 Counter (TIMx_CNT)

Offset address: 0x48

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CNT[31:0]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT[31:0]															
rw															

Bit Field	Name	Description
31:0	CNT[31:0]	Counter value

14 Low Power Timer (LPTIM)

14.1 Introduction

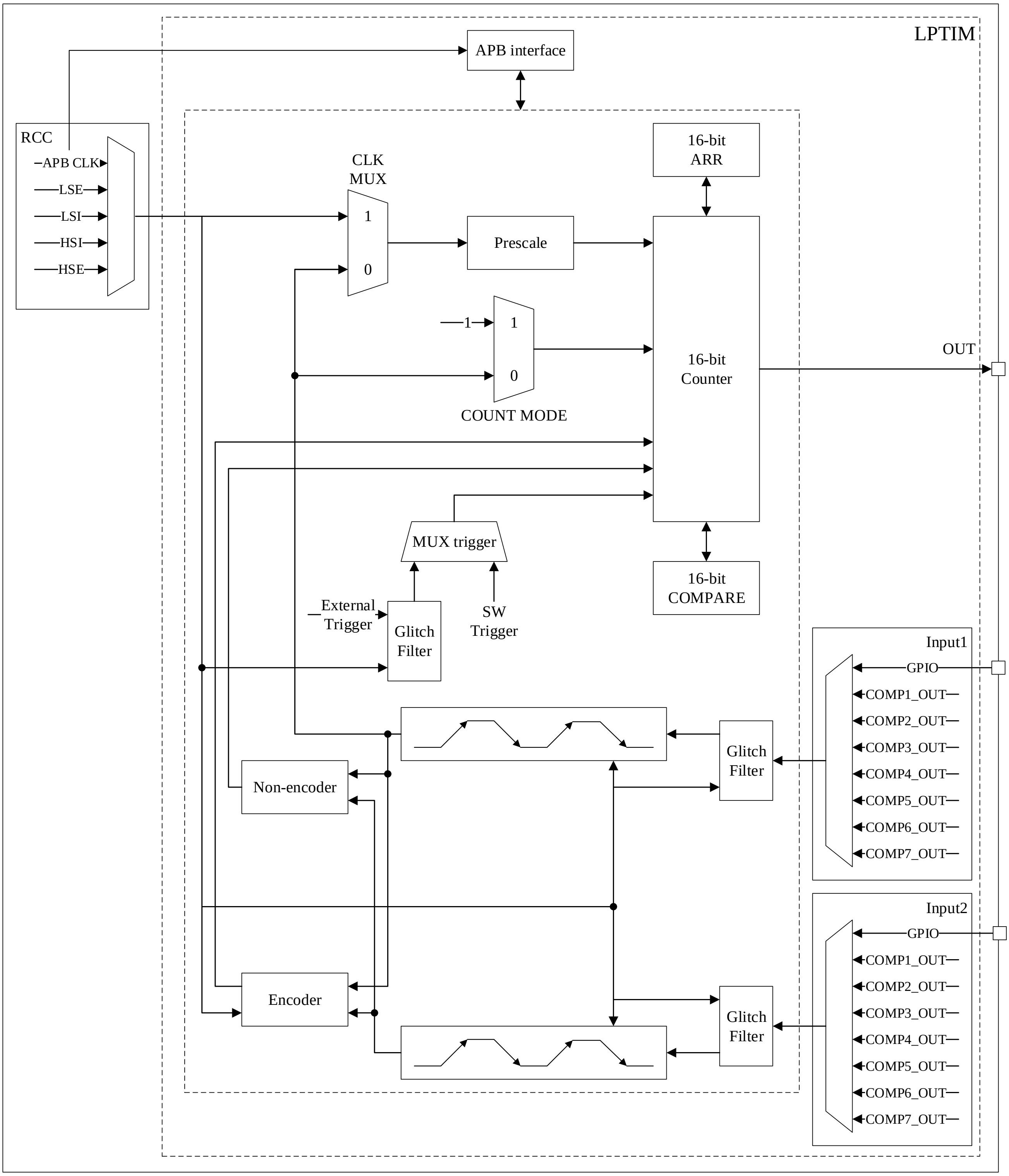
Supports two low-power timers: LPTIM1 and LPTIM2.The LPTIM is a 16-bit timer with multiple clock sources, it can keep running in all power modes. LPTIM can run without internal clock source, it can be used as a “Pulse Counter”. Also, the LPTIM can wake up the system from low-power modes, to realize “Timeout functions” with extremely low power consumption.

14.2 Main Features

- 16-bit up counter.
- 3-bit prescaler, 8 kinds of prescaler factors (1, 2, 4,8, 16, 32, 64, 128).
- Multiple clock sources:
  - Internal clock source: LSE, LSI, HSI, HSE or APB1 clock.
  - External clock source: External clock source through LPTIM Input1 (operating without LP oscillator, for pulse counter applications).
- 16-bit auto-load register (LPTIM_ARR).
- 16-bit compare register (LPTIM_CMP).
- Continuous or one-shot mode counting mode.
- Programmable software or hardware input trigger.
- Programmable digital filter for glitch filtering.
- Configurable output (PWM).
- Configurable IO polarity.
- Encoder mode.
- Pulse counting mode, support single pulse counting, double pulse counting (quadrature and non-quadrature).

14.3 Block Diagram

Figure 14-1 LPTIM Block Diagram



14.4 Function Description

14.4.1 Clocks and Reset

The LPTIM can use an internal clock source or an external clock source. The internal clock source can configure by RCC_RDCTRL.LPTIMSEL[2:0] bits. The external clock source can be selected from GPIO or Comparator 1, 2, 3, 4, 5, 6, 7. For external clock source, LPTIM has two configurations:

- The LPTIM uses both external clock and internal clock.
- The LPTIM only use external clock from comparator or external Input1. This configuration is suitable for low power application.

LPTIM_CFG.CLKSEL and LPTIM_CFG.CNTMEN bits are used for the clock source configuration. The active clock edge is configured through LPTIM_CFG.CLKPOL[1:0] bits.

When the LPTIM only uses external clock source. It can only select one active clock edge. LPTIM can select both active clock edges only when it is using internal clock source or both external and internal clock sources.

Note: when using both active edges for external clock, LPTIM needs to use an internal clock to oversample the external clock. The internal clock frequency should be at least 4 times higher than the external clock frequency.

14.4.2 Prescaler

The LPTIM counter is preceded by a configurable power-of-2 prescaler. The prescaler ratio is controlled by LPTIM_CFG.CLKPRE[2:0]. The table below lists all possible division ratios:

Table 14-1 Pre-scaler Ddivision Ratios

Control Bits	The Corresponding Frequency Division factor
000	/1
001	/2
010	/4
011	/8
100	/16
101	/32
110	/64
111	/128

14.4.3 Glitch Filter

LPTIM has glitch filters for inputs to remove glitches and prevent unexpected counts or triggers.Glitch filter needs an internal clock source to operate. And the clock source should be provided before the glitch filter is enabled. This is necessary to guarantee the proper operation of the filters.

The glitch filters has two major purposes:

- For the external inputs: The filter sensitivity is configured through the LPTIM_CFG.CLKFLT[1:0] bits.

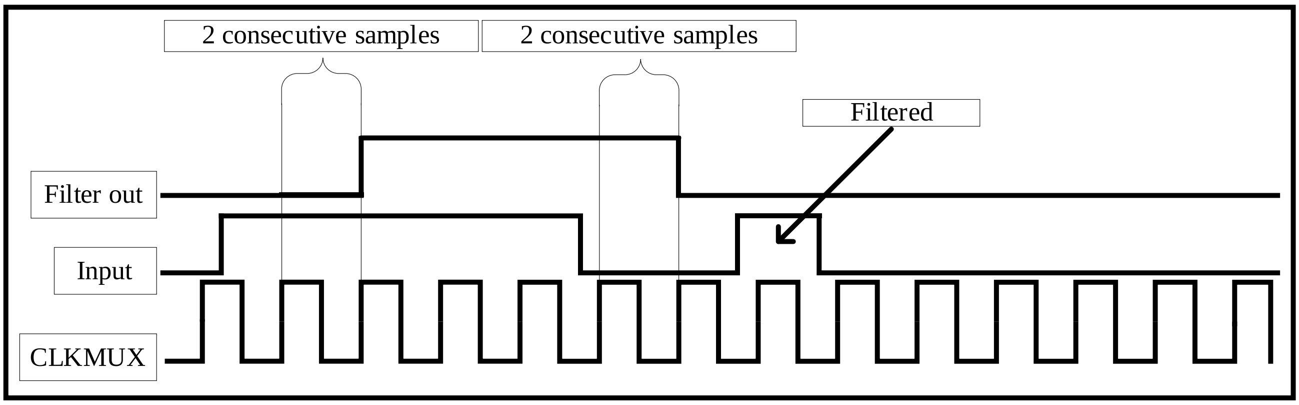
- For the internal trigger inputs: The filter sensitivity is configured through the LPTIM_CFG.RIGFLT[1:0] bits.

Note: The detection configuration is only applicable for its corresponding inputs.

The filter sensitivity acts on the number of consecutive equal samples that should be detected on one of the LPTIM inputs to consider a signal level change as a valid transition.

Figure 14-2 shows an example of glitch filter behavior when detected a 2 consecutive samples

Figure 14-2 Glitch Ffilter Timing Diagram



Note: If no internal clock is used, the glitch filter needs to be turned off by clearing LPTIM_CFG.CLKFLT[1:0] and LPTIM_CFG.TRIGFLT[1:0] bits. If glitch filter is not used, the user can use a digital filter in the comparator or an external analog filter to remove the glitch.

14.4.4 Timer Enable

The LPTIM_CTRL.LPTIMEN bit is used to enable/disable the LPTIM kernel logic. After setting the LPTIM_CTRL.LPTIMEN bit, a delay of two counter clock is needed before the LPTIM is turned on.

The LPTIM_CFG and LPTIM_INTEN registers must be modified only when the LPTIM is turned off.

14.4.5 Multiple Triggers

The LPTIM counter can be triggered either by software or by an effective edge on one of the 13 trigger inputs. The trigger source is configured through LPTIM_CFG.TRGEN[1:0] bits. LPTIM_CFG.TRGEN[1:0] = ‘00’, the trigger is selected as LPTIM_CTRL.TSTCM or LPTIM_CTRL.SNGMST bit, which can be set by software. The other values of LPTIM_CFG.TRGEN[1:0] are for the effective edge configuration of the trigger. The internal counter will start once an effective edge is detected.

LPTIM_CFG.TRGSEL[3:0] is used to select one of the 13 trigger inputs only when LPTIM_CFG.TRGEN[1:0] is not equal to ‘00’.

If LPTIM is using external trigger, which will be considered as asynchronous triggers. For asynchronous triggers, the LPTIM needs two counter clock cycles latency for synchronization.

If timeout function is disabled, new trigger event will be ignored if the LPTIM is already started.

Note: Any write to the LPTIM_CTRL.SNGMST/ LPTIM_CTRL.TSTCM bit will be discarded if the LPTIM is not enabled.

Table 14-2 Trigger Inputs Corresponding To LPTIM_CFG.TRGSEL[3:0] Bits

Control Bits	Corresponding Trigger Input
0000	LPTIM_ETR PIN
0001	RTC alarm A
0010	RTC alarm B
0011	RTC_TAMP1
0100	RTC_TAMP2
0101	RTC_TAMP3
0110	COMP1_OUT
0111	COMP2_OUT
1000	COMP3_OUT
1001	COMP4_OUT
1010	COMP5_OUT
1011	COMP6_OUT
1100	COMP7_OUT

14.4.6 Operating Mode

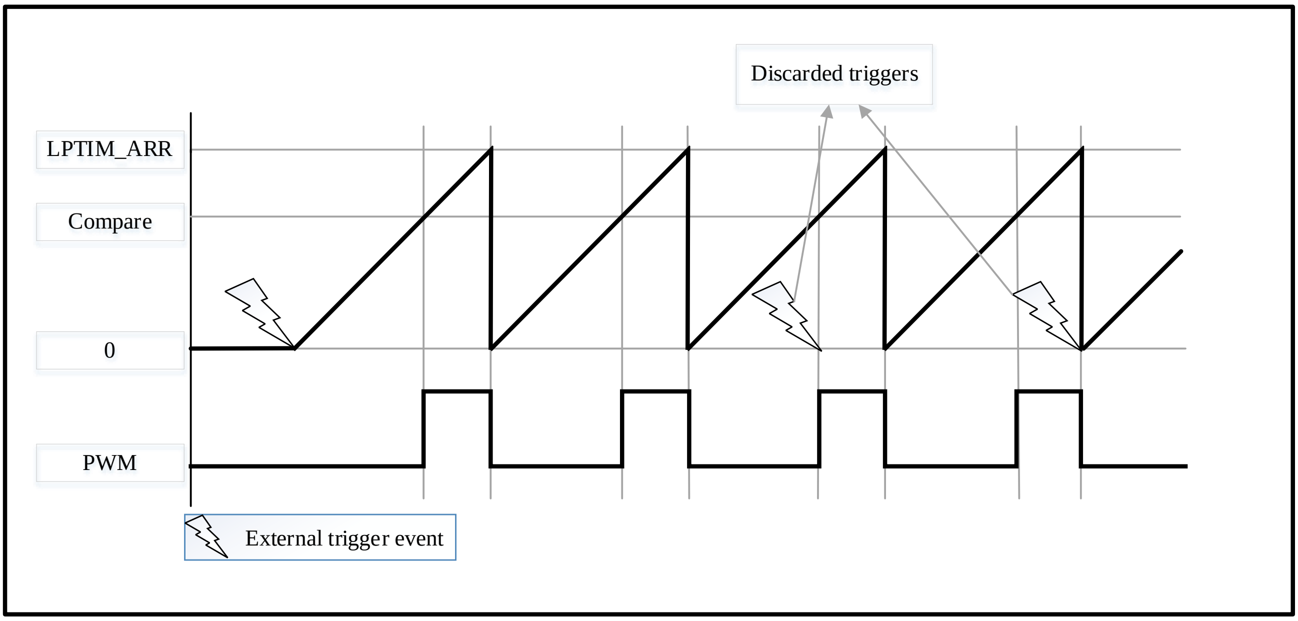
The LPTIM has two operating modes:

- Continuous mode: A trigger event will start the LPTIM and it will continue running until the user switched off the LPTIM.
- One-shot mode: A trigger event will start the LPTIM and it will stop when the counter value reaches LPTIM_ARR.ARRVAL[15:0].

Continuous mode:

LPTIM_CTRL.TSTCM bit must be set to enable the continuous mode. If LPTIM uses external trigger, the internal counter will start when an external trigger event arrives after LPTIM_CTRL.TSTCM bit is set. After the continuous mode starts, hardware will discard any subsequent external trigger event. Figure 14-3.

Figure 14-3 LPTIM Output Waveform, Continuous Counting Mode Configuration



PTIM_CTRL.SNGMST and LPTIM_CTRL.TSTCM bits can only be set when the LPTIM is enabled (The LPTIM_CTRL. LPTIMEN bit is set to ‘1’).

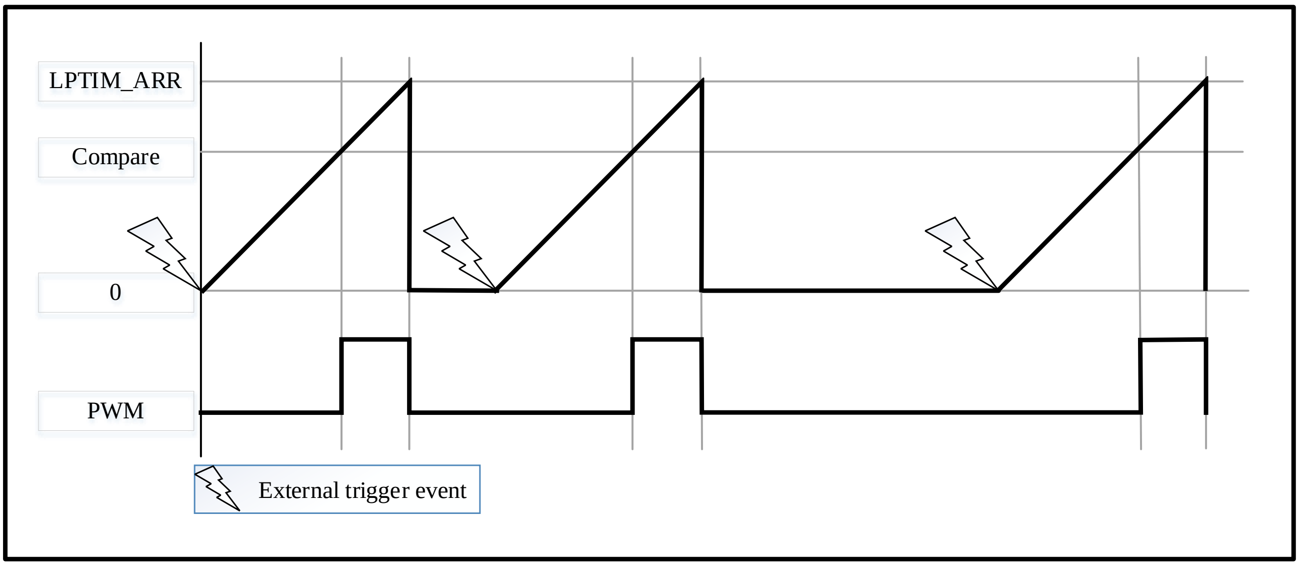
It is possible to switch from one-shot mode to continuous mode. Setting LPTIM_CTRL.SNGMST bit will switch the LPTIM to one-shot counting mode if continuous counting mode was previously selected. The counter stops as soon as it reaches the LPTIM_ARR register value. If the one-shot counting mode was previously selected, setting LPTIM_CTRL.TSTCM bit to 1 will switch the LPTIM to continuous counting mode. Counter will restart as soon as LPTIM_ARR register value is reached.

One-shot mode:

LPTIM_CTRL.SNGMST bit must be set to enable the one-shot mode. A trigger event will re-start the LPTIM. Hardware will abandon all the trigger events after the internal counter starts and before the counter value equal to LPTIM_ARR.ARRVAL[15:0] value.

If an external trigger is selected, each external trigger event arriving after the LPTIM_CTRL.SNGMST bit is set, and after the timer register is stopped (containing a zero value), the timer is restarted for a new count cycle, as shown in Figure Figure 14-4.

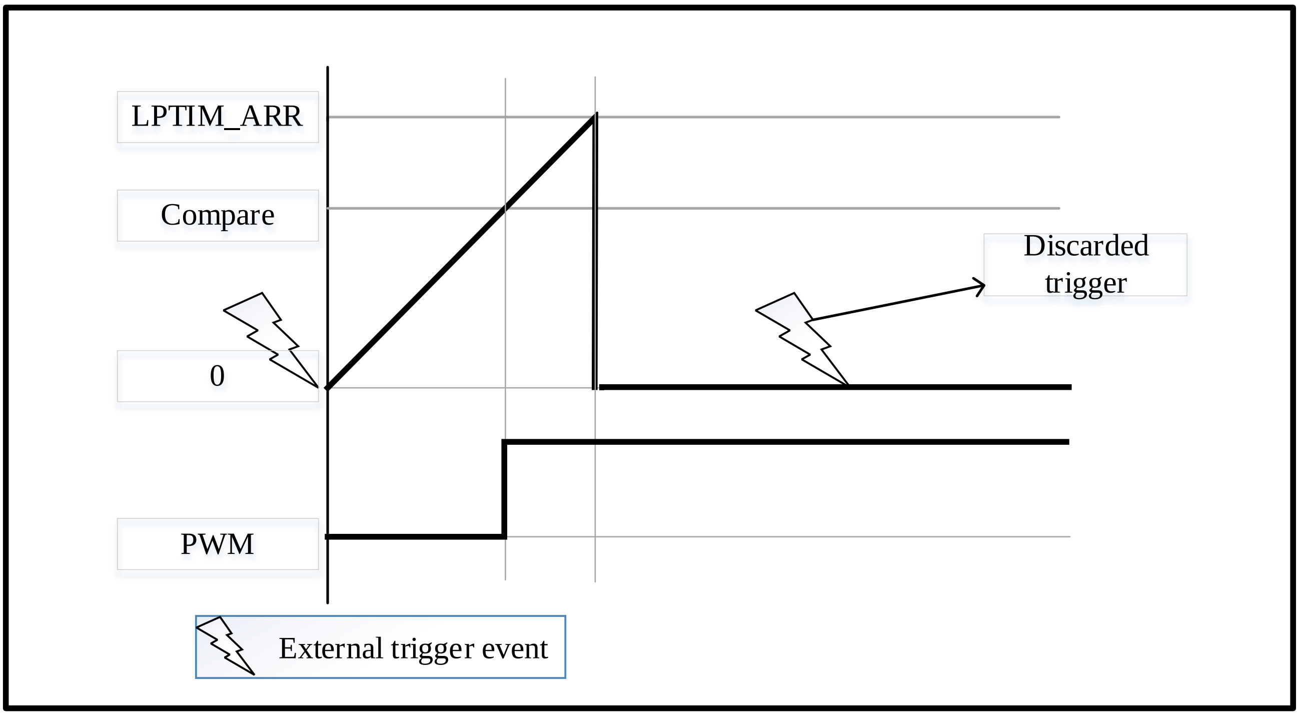
Figure 14-4 LPTIM Output Waveform, One-shot Counting Mode Configuration



Set-once mode:

The Set-once mode is used when the LPTIM_CFG.WAVE bit is set. In Set-once mode, the counter is started once when the first trigger event happens, the hardware will discard any subsequent trigger event, as shown Figure Figure 14-5.

Figure 14-5 LPTIM Output Waveform, Single Counting Mode configCration And Set-Once Mode Activated



In case of software start (LPTIM_CFG.TRGEN[1:0] = ‘00’), the LPTIM_CTRL.SNGMST setting will start the counter for one-shot counting.

14.4.7 Waveform Generation

The LPTIM auto-reload register (LPTIM_ARR) and compare register (LPTIM_CMP) are used for generating LPTIM output waveforms.

The waveforms supported by LPTIM are shown as below:

- PWM mode: LPTIM output is set when a CMP match event happens. (I.E. the LPTIM_CNT register value matched the LPTIM_CMP register value.) The LPTIM output is reset when a ARR match happens. (I.E. the LPTIM_CNT register value matched the LPTIM_ARR register value).
- One-pulse mode: The first pulse is triggered same as PWM mode, then the output is permanently reset when the ARR match happens.
- Set-once mode: the output waveform is similar to the One-pulse mode except that the output is kept to the last signal level (depends on the output configured polarity).

Above waveform configuration require that LPTIM_ARR register value must be configured bigger than the LPTIM_CMP register value.

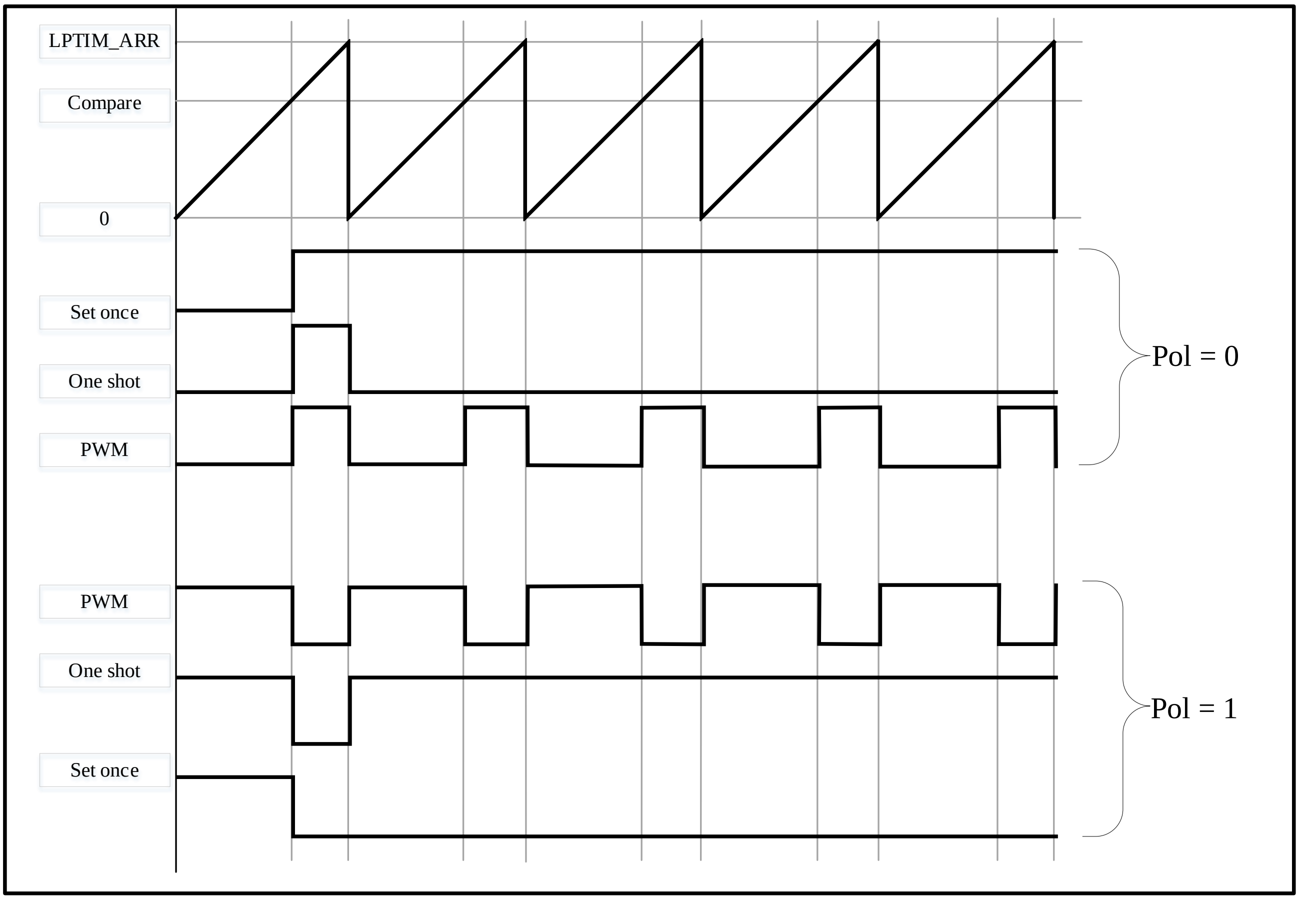
The LPTIM output waveform can be configured through the LPTIM_CFG.WAVE bit as follow:

- Clearing the LPTIM_CFG.WAVE bit will force the LPTIM to generate a PWM waveform or a One-pulse waveform depending on the set bit (LPTIM_CTRL.TSTCM or LPTIM_CTRL.SNGMST).
- LPTIM_CTRL.WAVE bit equals to '1' forces the LPTIM to generate a Set-once mode waveform.

The LPTIM_CFG.WAVEPOL bit controls LPTIM output polarity. The output default value will change immediately after the user configured the polarity, even when the timer is disabled.

Signals with frequencies up to the LPTIM clock frequency divided by 3 can be generated. Figure 14-6 below shows the three possible waveforms that can be generated on the LPTIM output. Also, it shows the effect of the polarity change using the LPTIM_CFG.WAVEPOL bit.

Figure 14-6 Waveform generation



14.4.8 Register Update

The LPTIM_ARR register and LPTIM_CMP register can be updated immediately after a software write operation. If the LPTIM is started, the LPTIM_ARR register and LPTIM_CMP register can be updated when counter overflow.

The LPTIM APB interface and the LPTIM kernel logic use different clocks, so there is some latency between the software write through APB bus and the moment when these values are available to the kernel logic. During this latency period, any additional write into these registers must be avoided.

The update method of LPTIM_ARR and LPTIM_CMP registers is determined by the LPTIM_CFG.RELOAD bit:

- LPTIM_CFG.RELOAD bit equals to '1': LPTIM_ARR and LPTIM_CMP registers are updated when counter overflow, if the LPTIM already started. When counter overflow, latency = 2~3 APB clock period.
- LPTIM_CFG.RELOAD bit equals to '0': LPTIM_ARR and LPTIM_CMP registers are updated after any software write access. Latency = 2~3 APB clock period + 2~3 LPTIM internal prescaled clock period.

The LPTIM_INTSTS.ARRUPD flag and the LPTIM_INTSTS.CMPUPD flag indicate when the write operation is completed to respectively the LPTIM_ARR register and the LPTIM_COMP register.

After a write to the LPTIM_ARR register or the LPTIM_CMP register, any successive write before respectively the

LPTIM_INTSTS.ARRUPD flag or the LPTIM_INTSTS.CMPUPD flag be set, will lead to unpredictable results. So a new write operation to the same register can only be performed when the previous write operation is completed.

14.4.9 Counter Mode

The internal counter can count external trigger events from LPTIM Input1 or internal clock cycles. This can be configured through LPTIM_CFG.CLKSEL and LPTIM_CFG.CNTMEN bits.

If LPTIM is counting external triggers, user can configure LPTIM_CFG.CLKPOL[1:0] bits to select the active edge from rising edge, falling edge or both edges. The count modes below can be selected, depending on LPTIM_CFG.CLKSEL and LPTIM_CFG.CNTMEN bits values:

- LPTIM_CFG.CLKSEL = 0: the LPTIM use an internal clock source to clock.
  - LPTIM_CFG.CNTMEN=0, The LPTIM is configured to be clocked by an internal clock source and the LPTIM counter is configured to be updated following each internal clock pulse.
  - LPTIM_CFG.CNTMEN=1, The LPTIM external Input1 is sampled with the internal clock provided to the LPTIM. In order to not miss any event, the frequency of the changes on the external Input1 signal should never exceed the frequency of the internal clock provided to the LPTIM. Also, the internal clock provided to the LPTIM must not be pre-scaled (LPTIM_CFG.CLKPRE[2:0] = 000).
- LPTIM_CFG.CLKSEL = 1: the LPTIM use an external clock source to clock.
  - LPTIM_CFG.CNTMEN bit value is don’t care. In this configuration, the LPTIM has no need for an internal clock source (except if the glitch filters are enabled). The signal injected on the LPTIM external Input1 is used as system clock for the LPTIM. This configuration is suitable for operation modes where no embedded oscillator is enabled.
  - For this configuration, the LPTIM counter can be updated either on rising edges or falling edges of the input1 clock signal but not on both rising and falling edges.
  - Since the signal injected on the LPTIM external Input1 is also used to clock the LPTIM kernel logic, there is some initial latency (after the LPTIM is enabled) before the counter is incremented. More precisely, the first five active edges on the LPTIM external Input1 (after LPTIM is enable) are lost.

14.4.10 Encode Mode

The Encoder mode can handle signals from quadrature encoders which used to detect angular position of rotary elements. The encoder mode allows the counter counts the events within 0 and LPTIM_ARR.ARRVAL[15:0] value. (0 up to LPTIM_ARR.ARRVAL[15:0] or LPTIM_ARR.ARRVAL[15:0] to 0). In this case, user must configure LPTIM_ARR.ARRVAL[15:0] before enable the counter. From external Input1 and Input2, a clock is generated for the counter. The counting direction depends on the phase between these two input signals.

The Encoder mode is only available when the LPTIM use an internal clock source to clock. The signals frequency on both Input1 and Input2 inputs must not exceed the LPTIM internal clock frequency divided by 4. This is mandatory in order to guarantee a proper operation of the LPTIM.

The change of counting direction is updated by the two Down and Up flags in the LPTIM_INTSTS register. Also, an interrupt can be generated for both direction change events if enabled through the LPTIM_INTEN.DOWNIE and LPTIM_INTEN.UPIE register.

User can enable Encoder mode by setting LPTIM_CFG.ENC bit. And the LPTIM need to be configured in continuous mode first.

When Encoder mode is active, the LPTIM counter is modified automatically following the speed and the direction of the incremental encoder. Therefore, its content always represents the encoder's position. The count direction, signaled by the Up and Down flags, correspond to the rotation direction of the encoder rotor.

Different counting scenarios may occur based on the different trigger edges configured by the LPTIM_CFG.CLKPOL[1:0] bits. The following table summarizes the possible combinations, assuming that Input1 and Input2 do not switch at the same time.

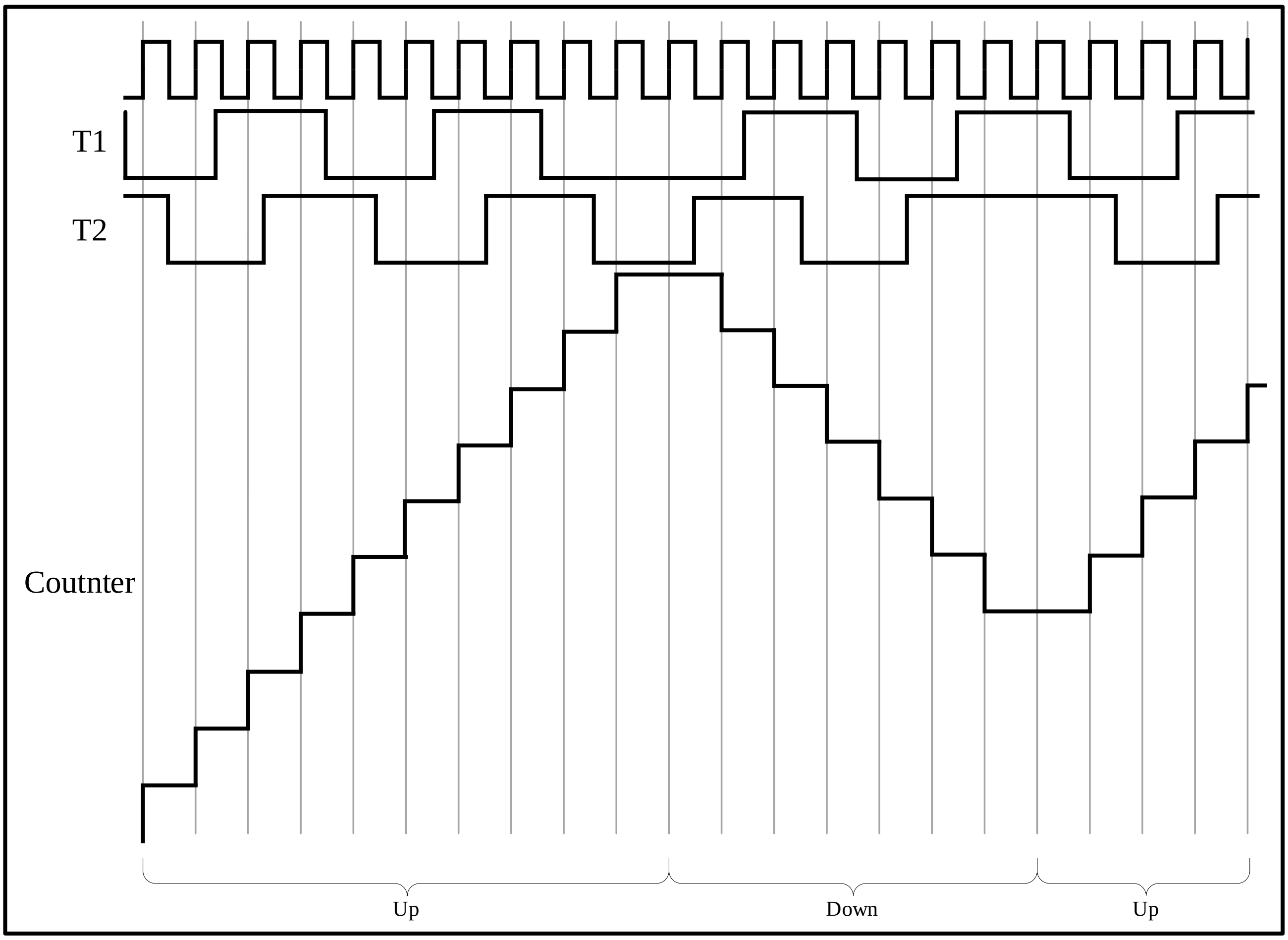
Table 14-3 Encoder Counting Scenarios

Trigger edge	**The signals is opposite (Input1 For Input2, Input2 For Input1)**	**Input1 Signal**		**Input2 Signal**	
		Rising	**Falling**	**Rising**	**Falling**
Rising Edge	High	Down	No count	Up	No count
	Low	Up	No count	Down	No count
Falling Edge	High	No count	Up	No count	Down
	Low	No count	Down	No count	Up
Both Edges	High	Down	Up	Up	Down
	Low	Up	Down	Down	Up

The following figure shows a counting sequence for Encoder mode where both-edge polarity is configured.

Note: *In this mode the LPTIM must be clocked by an internal clock source, so the LPTIM_CFG.CLKSEL bit must be maintained to its reset value which is equal to '0'. Also, the prescaler division ratio must be equal to its reset value which is 1 (LPTIM_CFG.CLKPRE[2:0] bits must be '000').*

Figure 14-7 Encoder Mode Counting Sequence



14.4.11 Non-Quadrature Encoder Mode

This mode allows handling signals from non-quadrature encoders, which is used to detect sub-sequent positive pulses from external interface. Non-Encoder interface mode acts simply as an external clock with direction selection. This means that the counter just counts continuously between 0 and the auto-reload value programmed into the LPTIM_ARR register (0 up to LPTIM_ARR.ARRVAL[15:0] or LPTIM_ARR.ARRVAL[15:0] down to 0 depending on the direction). Therefore you must configure LPTIM_ARR before starting. From the two external input signals, Input1 and Input2, a clock signal is generated to clock the LPTIM counter. The order between those two signals determines the counting direction.

The Non-Encoder mode is only available when the LPTIM is clocked by an internal clock source. The signals frequency on both Input1 and Input2 inputs must not exceed the LPTIM internal clock frequency divided by 4. This is mandatory in order to guarantee a proper operation of the LPTIM.

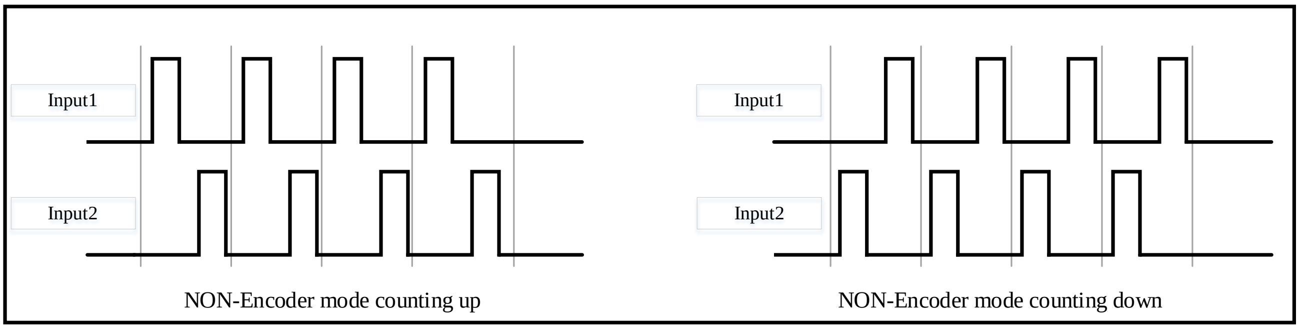
Direction change is signalized by the two Down and Up flags in the LPTIM_INTSTS register. Also, an interrupt can be generated for both direction change events if enabled through the LPTIM_INTEN.DOWNIE and LPTIM_INTEN.UPIE register.

To activate the Non-Encoder mode the LPTIM_CFG.NENC bit has to be set to ‘1’. The LPTIM must first be configured in Continuous mode.

When Non-Encoder mode is active, the LPTIM counter is modified automatically following the speed and the direction of the incremental encoder. Therefore, its content always represents the encoder's position. The count direction, signaled by the Up and Down flags, correspond to the rotation direction of the encoder rotor.

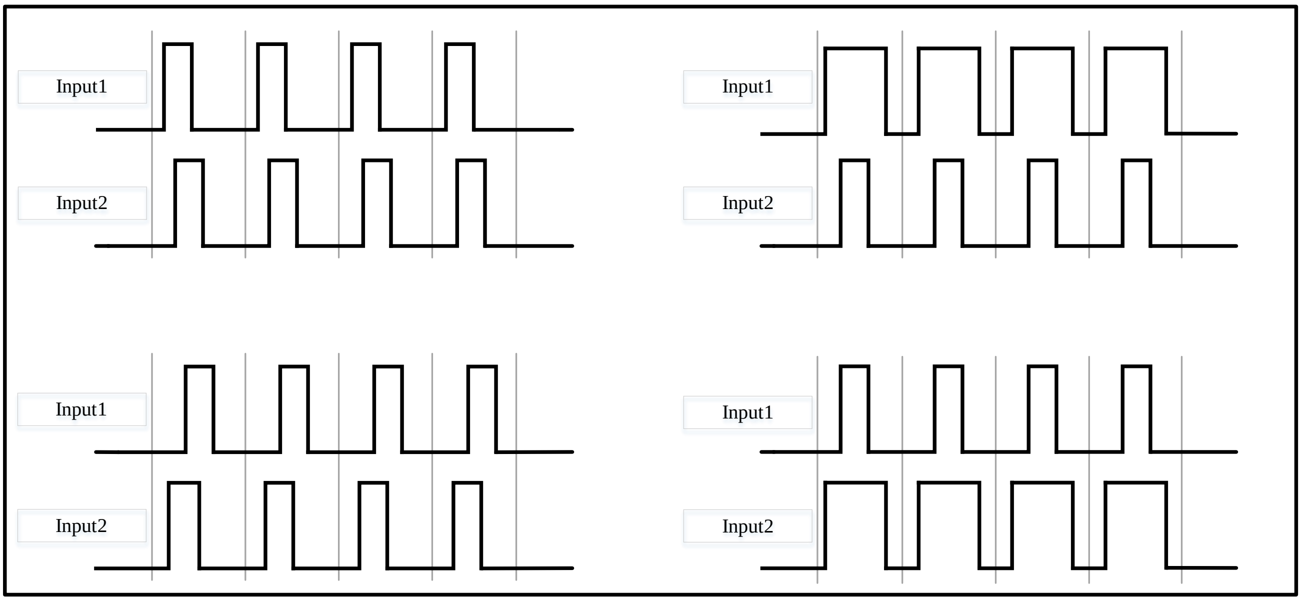
Figure 14-8 shown, the decoder module can work properly, when there is no case that both Input1 and Input2 are high

Figure 14-8 Input waveforms of Input1 and Input2 when the decoder module is working normally



If the Input1 and Input2 waveform is as Figure 14-9, the decoder module can't work properly. The counter will ignore these waveforms and keep the previous value.

Figure 14-9 Input1 and Input2 input waveforms when decoder module is not working



14.4.12 Timeout Function

When the LPTIM_CFG.TIMOUTEN bit is enable, the LPTIM counter will be reset by an active edge from one selected trigger input.

When timeout function is used, the LPTIM counter will be reset and re-start by a selected trigger input event. If no trigger occurs within the configured time, the compare match event will happen. The waiting time is configured through the timeout value.

14.4.13 LPTIM Interrupts

The following events generate an interrupt/wake-up event, if they are enabled through the LPTIM_INTEN register:

- Compare match.
- Auto-reload match (whatever the direction if encoder mode).
- External trigger event.
- Autoreload register write completed.
- Compare register write completed.
- Direction change (encoder mode), programmable (up / down / both).

Note: If any bit in the LPTIM_INTEN register (Interrupt Enable Register) is set after that its corresponding flag in the LPTIM_INTSTS register (Status Register) is set, the interrupt is not asserted.

Table 14-4 Interruption Events

Interrupt Event	Description
Compare match	Interrupt flag LPTIM_INSTS.CMPM is set when LPTIM_CNT (counter register value) = LPTIM_COMP (compare register value).
Auto reload match	Interrupt flag LPTIM_INSTS.ARRM is set when LPTIM_CNT (counter register value) = LPTIM_ARR (auto-reload register value)
External trigger event	Interrupt flag LPTIM_INSTS.EXTRIG is set when an external trigger event is detected
Autoreload register write completed	Interrupt flag LPTIM_INSTS.CMPUPD is set when the write operation to the LPTIM_ARR register is complete
Compare register write completed	Interrupt flag LPTIM_INSTS.ARRUPD is set when the write operation to the LPTIM_COMP register is complete
Direction change	Used in Encoder mode. Two interrupt flags are embedded to signal direction change: - LPTIM_INSTS.Up flag indicated that the count direction is changed to count up - LPTIM_INSTS.Down flag indicated that the count direction is changed to count down.

14.5 LPTIM Registers

14.5.1 LPTIM Registers Overview

Table 14-5 LPTIM Register Overview

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
000h	LPTIM_INTSTS	Reserved																									DOWN	UP	ARRUPD	CMPUPD	EXTRIG	ARRM	CMPM
	Reset Value																										0	0	0	0	0	0	0
004h	LPTIM_INTCLR	Reserved																									DOWNCF	UPCF	ARRUPDCF	CMPUPDCF	EXTRIGCF	ARRMCF	CMPMCF
	Reset Value																										0	0	0	0	0	0	0
008h	LPTIM_INTEN	Reserved																									DOWNIE	UPIE	ARRUPDIE	CMPUPDIE	EXTRIGIE	ARRMIE	CMPMIE
	Reset Value																										0	0	0	0	0	0	0
00Ch	LPTIM_CFG	Reserved						NENC	ENC	CNTMEN	RELOAD	WAVEPOL	WAVE	TIMOUTEN	TRGEN[1:0]		TRGSEL[3:0]				Reserved	CLKPRE[2:0]			Reserved	TRIGFLT[1:0]		Reserved	CLKFLT[1:0]		CLKPOL[1:0]		CLKSEL
	Reset Value							0	0	0	0	0	0	0	0	0	0	0	0	0		0	0	0		0	0		0	0	0	0	0
010h	LPTIM_CTRL	Reserved																													TSTCM	SNGMST	LPTIMEN
	Reset Value																														0	0	0
014h	LPTIM_CMP	Reserved																CMPVAL[15:0]															
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
018h	LPTIM_ARR	Reserved																ARRVAL[15:0]															
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
01Ch	LPTIM_CNT	Reserved																CNTVAL[15:0]															
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
020h	LPTIM_OPT	Reserved																										OPT2			OPT1		
	Reset Value																											0	0	0	0	0	0

14.5.2 LPTIM Interrupt And Status Register (LPTIM_INTSTS)

Offset address: 0x00

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									DOWN	UP	ARRUPD	CMPUPD	EXTRIG	ARRM	CMPM
									r	r	r	r	r	r	r

Bit Field	Name	Description
31:7	Reserved	Reserved, the reset value must be maintained.
6	DOWN	Change counter direction to down. In Encoder mode, hardware will set DOWN bit to inform the application the counter direction.
5	UP	Change counter direction up. In Encoder mode, hardware will set UP bit to inform the application the counter direction.
4	ARRUPD	Auto-reload value updated to register. Hardware sets ARRUPD to inform application that LPTIM_ARR register has been written by the APB bus successfully. For more details, see 14.4.8.
3	CMPUPD	Compare value updated to register. Hardware sets COMPUPD to inform application that LPTIM_COMP register has been written by the APB bus successfully. For more details, see 14.4.8.
2	EXTRIG	External trigger valid event. Hardware sets EXTRIG to inform application that a valid external trigger edge has occurred. If the trigger is discarded when timer has already started, then this flag is not set.
1	ARRM	Auto-reload match. Hardware set this to inform application that LPTIM_CNT register value reached the LPTIM_ARR register's value.
0	CMPM	Compare match. Hardware set this to inform application that LPTIM_CNT register value reached the LPTIM_COMP register's value.

14.5.3 LPTIM Interrupt Clear Register (LPTIM_INTCLR)

Offset address: 0x04

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									DOWNCF	UPCF	ARRUPDCF	CMPUPDCF	EXTRIGCF	ARRMCF	CMPMCF
									w	w	w	w	w	w	w

Bit Field	Name	Description
31: 7	Reserved	Reserved, the reset value must be maintained.
6	DOWNCF	Direction change to down Clear Flag Writing 1 to this bit clear the DOWN flag in the LPTIM_INTSTS register
5	UPCF	Direction change to UP Clear Flag Writing 1 to this bit clear the UP flag in the LPTIM_INTSTS register
4	ARRUPDCF	Autoreload register update completed Clear Flag Writing 1 to this bit clears the ARRUPD flag in the LPTIM_INTSTS register
3	CMPUPDCF	Compare register update completed Clear Flag Writing 1 to this bit clears the CMPUPD flag in the LPTIM_INTSTS register
2	EXTRIGCF	External trigger valid edge Clear Flag Writing 1 to this bit clears the EXTRIG flag in the LPTIM_INTSTS register
1	ARRMCF	Autoreload match Clear Flag Writing 1 to this bit clears the ARRM flag in the LPTIM_INTSTS register
0	CMPMCF	compare match Clear Flag Writing 1 to this bit clears the CMPM flag in the LPTIM_INTSTS register

14.5.4 LPTIM Interrupt Enable Register (LPTIM_INTEN)

Offset address: 0x08

Reset value: 0x0000 0000

Note: The LPTIM_INTEN register must only be modified when the LPTIM is disabled (LPTIM_CTRL.LPTIMEN bit reset to ‘0’).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									DOWNIE	UPIE	ARRUPDIE	CMPUPDIE	EXTRIGIE	ARRMIE	CMPMIE
									rw	rw	rw	rw	rw	rw	rw

Bit Field	Name	Description
31:7	Reserved	Reserved, the reset value must be maintained.
6	DOWNIE	Direction change to down interrupt enable bit. 0: DOWN interrupt disabled

Bit Field	Name	Description
		1: DOWN interrupt enabled
5	UPIE	Direction change to up interrupt enable bit. 0: UP interrupt disabled 1: UP interrupt enabled
4	ARRUPDIE	Auto reload register update succeeded interrupt enable bit. 0: ARRUPD interrupt disable 1: ARRUPD interrupt enable
3	CMPUPDIE	Compare register update succeeded interrupt enable bit. 0: CMPUPD interrupt disabled 1: CMPUPD interrupt enabled
2	EXTRIGIE	External trigger valid edge interrupt enable bit. 0: EXTRIG interrupt disabled 1: EXTRIG interrupt enabled
1	ARRMIE	Auto reload match interrupt enable bit. 0: ARRM interrupt disabled 1: ARRM interrupt enabled
0	CMPMIE	Compare match interrupt enable bit. 0: CMPM interrupt disabled 1: CMPM interrupt enabled

14.5.5 LPTIM Configuration Register (LPTIM_CFG)

Offset address: 0x0C

Reset value: 0x0000 0000

Note: The LPTIM_CFG register must only be modified when the LPTIM is disabled (LPTIM_CTRL.LPTIMEN bit reset to '0').

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved						NENC	ENC	CNTMEN	RELOAD	WAVEPOL	WAVE	TIMOUTEN	TRGEN[1:0]		TRGSEL[3]
						rw	rw	rw	rw	rw	rw	rw	rw		rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRGSEL[2:0]			Reserved	CLKPRE[2:0]			Reserved	TRIGFLT[1:0]		Reserved	CLKFLT[1:0]		CLKPOL[1:0]		CLKSEL
rw				rw				rw			rw		rw		rw

Bit Field	Name	Description
31:26	Reserved	Reserved, the reset value must be maintained.
25	NENC	Non-orthogonal mode enable 0: Non-orthogonal Encoder mode disabled 1: Non-orthogonal Encoder mode enabled
24	ENC	Encoder mode enable 0: Encoder mode disabled 1: Encoder mode enabled
23	CNTMEN	Counter mode enabled

Bit Field	Name	Description
		The CNTMEN bit selects clock source for the LPTIM counter: 0: Counter is incremented following each internal clock pulse 1: Counter is incremented following each valid clock pulse on the LPTIM external Input1
22	RELOAD	Registers update mode The RELOAD bit controls the LPTIM_ARR and the LPTIM_COMP registers update mode 0: Registers are updated after each APB bus write access 1: Registers are updated at the end of the current LPTIM period
21	WAVEPOL	Waveform shape polarity The WAVEPOL bit controls the output polarity 0: The LPTIM output reflects the compare results between LPTIM_ARR and LPTIM_CMP registers 1: The LPTIM output reflects the inverse of the compare results between LPTIM_ARR and LPTIM_CMP registers
20	WAVE	Waveform shape The WAVE bit controls the output shape 0: Deactivate Set-once mode, PWM / One Pulse waveform (depending on LPTIM_CTRL.TSTCM or LPTIM_CTRL.SNGMST bit) 1: Activate the Set-once mode
19	TIMOUTEN	Timeout enable 0: A trigger event arriving when the timer is already started will be ignored 1: A trigger event arriving when the timer is already started will reset and restart the counter
18:17	TRGEN[1:0]	Trigger enable and polarity The TRGEN bits controls whether the LPTIM counter is started by an external trigger or not. If the external trigger option is selected, three configurations are possible for the trigger active edge: 00: Software trigger (counting start is initiated by software) 01: Falling edge is the active edge 10: rising edge is the active edge 11: Both edges are active edges
16:13	TRGSEL[3:0]	Trigger selector The TRGSEL bits select the trigger source that will serve as a trigger event for the LPTIM among the below 13 available sources: 0000: LPTIM_ETR PIN 0001: RTC alarm A 0010: RTC alarm B 0011: RTC_TAMP1 0100: RTC_TAMP2 0101: RTC_TAMP3 0110: COMP1_OUT

Bit Field	Name	Description
		0111: COMP2_OUT 1000: COMP3_OUT 1001: COMP4_OUT 1010: COMP5_OUT 1011: COMP6_OUT 1100: COMP7_OUT Other: reserved
12	Reserved	Reserved, the reset value must be maintained.
11:9	CLKPRE[2:0]	Clock division factor bit. 000: /1 001: /2 010: /4 011: /8 100: /16 101: /32 110: /64 111: /128
8	Reserved	Reserved, the reset value must be maintained.
7:6	TRIGFLT[1:0]	Configure the data filter trigger bit. The TRIGFLT value sets the number of consecutive equal samples that should be detected when a level change occurs on an internal trigger before it is considered as a valid level transition. An internal clock source must be present to use this feature 00: Any trigger active level change is considered as a valid trigger. 01: Trigger active level change must be stable for at least 2 clock periods before it is considered as valid trigger. 10: Trigger active level change must be stable for at least 4 clock periods before it is considered as valid trigger. 11: Trigger active level change must be stable for at least 8 clock periods before it is considered as valid trigger.
5	Reserved	Reserved, the reset value must be maintained.
4:3	CLKFLT[1:0]	Digital filter external clock input configuration The CLKFLT value sets the number of consecutive equal samples that should be detected when a level change occurs on an external clock signal before it is considered as a valid level transition. An internal clock source must be present to use this feature 00: Any external clock signal level change is considered as a valid transition. 01: External clock signal level change must be stable for at least 2 clock periods before it is considered as valid transition. 10: External clock signal level change must be stable for at least 4 clock periods before it is considered as valid transition. 11: External clock signal level change must be stable for at least 8 clock periods before it is considered as valid transition.
2:1	CLKPOL[1:0]	Clock Polarity

Bit Field	Name	Description
		If LPTIM is clocked by an external clock source: When the LPTIM is clocked by an external clock source, CLKPOL bits is used to configure the active edge or edges used by the counter: 00: The rising edge is the active edge used for counting 01: The falling edge is the active edge used for counting 10: Both edges are active edges. 11: Not allowed *Note: When both external clock signal edges are considered active ones, the LPTIM must also be clocked by an internal clock source with a frequency equal to at least four time the external clock frequency.* If the LPTIM is configured in Encoder mode (LPTIM_CFG.ENC bit is set): 00: The encoder rising edgecounting mode. 01: The encoder falling edge counting mode. 10: The encoder both edges counting mode.
0	CLKSEL	Clock selector The CKSEL bit selects which clock source the LPTIM will use: 0: LPTIM is clocked by internal clock source (APB clock or any of the embedded oscillators) 1: LPTIM is clocked by an external clock source through the LPTIM external Input1

14.5.6 LPTIM Control Register (LPTIM_CTRL)

Offset address: 0x10

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													TSTCM	SNGMST	LPTIMEN
													rw	rw	rw

Bit Field	Name	Description
31:3	Reserved	Reserved, the reset value must be maintained.
2	TSTCM	Timer start in Continuous mode This bit is set by software and cleared by hardware. In case of software start (LPTIM_CFG.TRGEN[1:0] = ‘00’), setting this bit starts the LPTIM in Continuous mode. If the software start is disabled (TRGEN[1:0] ≠ ‘00’), setting this bit starts the timer in Continuous mode as soon as an external trigger is detected. If this bit is set when a single pulse mode counting is ongoing, then the timer will not stop at the next match between the LPTIM_ARR and LPTIM_CNT registers and the

Bit Field	Name	Description
		LPTIM counter keeps counting in Continuous mode. This bit can be set only when the LPTIM is enabled. It will be automatically reset by hardware.
1	SNGMST	LPTIM start in Single pulse mode This bit is set by software and cleared by hardware. In case of software start (LPTIM_CFG.TRGEN[1:0] = ‘00’), setting this bit starts the LPTIM in single pulse mode. If the software start is disabled (LPTIM_CFG.TRGEN[1:0] ≠ ‘00’), setting this bit starts the LPTIM in single pulse mode as soon as an external trigger is detected. If this bit is set when the LPTIM is in continuous counting mode, then the LPTIM will stop at the following match between LPTIM_ARR and LPTIM_CNT registers. This bit can only be set when the LPTIM is enabled. It will be automatically reset by hardware.
0	LPTIMEN	LPTIM enable The LPTIMEN bit is set and cleared by software. 0: LPTIM is disabled 1: LPTIM is enabled

14.5.7 LPTIM Compare Register (LPTIM_CMP)

Offset address: 0x14

Reset value: 0x0000 0000

Note: The LPTIM_CMP register must only be modified when the LPTIM is enabled (LPTIM_CTRL.LPTIMEN bit reset to ‘1’).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMPVAL[15:0]															
rw															

Bit Field	Name	Description
31:16	Reserved	Reserved, the reset value must be maintained.
15:0	CMPVAL[15:0]	Compare value CMPVAL is the compare value used by the LPTIM.

14.5.8 LPTIM Auto-Reload Register (LPTIM_ARR)

Offset address: 0x18

Reset value: 0x0000 0001

Note: The LPTIM_ARR register must only be modified when the LPTIM is enabled (LPTIM_CTRL.LPTIMEN bit reset to ‘1’)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARRVAL[15:0]															
rw															

Bit Field	Name	Description
31:16	Reserved	Reserved, the reset value must be maintained.
15:0	ARRVAL[15:0]	Auto reload value ARRVAL is the autoreload value for the LPTIM. This value must be strictly greater than the LPTIM_COMP.CMPVAL[15:0] value.

14.5.9 LPTIM Counter Register (LPTIM_CNT)

Offset address: 0x1C

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNTVAL[15:0]															
r															

Bit Field	Name	Description
31:16	Reserved	Reserved, the reset value must be maintained.
15:0	CNTVAL[15:0]	Counter value When the LPTIM is running with an asynchronous clock, reading the LPTIM_CNT register may return unreliable values. So in this case it is necessary to perform two consecutive read accesses and verify that the two returned values are identical. If identical, the reading is reliable.

14.5.10 LPTIM Option Register (LPTIM_OPT)

Offset address: 0x20

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										OPT2[2:0]			OPT1[2:0]		
										rw			rw		

Bit Field	Name	Description
31:6	Reserved	Reserved, the reset value must be maintained.
5:3	OPT2[2:0]	LPTIM Input2 Connection selection bit 000: LPTIM Input 2 connection to I/O 001: LPTIM Input 2 connection to COMP1_OUT 010: LPTIM Input 2 connection to COMP2_OUT … 111: LPTIM Input 2 connection to COMP7_OUT
2:0	OPT1[2:0]	LPTIM Input1 Connection selection bit 000: LPTIM Input 1 connection to I/O 001: LPTIM Input 1 connection to COMP1_OUT 010: LPTIM Input 1 connection to COMP2_OUT … 111: LPTIM Input 1 connection to COMP7_OUT

15 Independent Watchdog（IWDG）

15.1 Introduction

The Chip has embedded independent watchdog (IWDG) and window watchdog (WWDG) timers to solve the problems caused by software errors. The watchdog timer is very flexible to use, which improves the security of the system and the accuracy of timing control

The independent Watchdog (IWDG) is driving by Low-speed internal clock (LSI clock) running at 32 KHz, which will still running event dead loop or MCU stuck is happening. This can provide higher safety level, timing accuracy and flexibility of watchdog. It can reset and resolve system malfunctions due to software failure. The IWDG is best suited for applications that require the watchdog to run as a totally independent process outside the main application, but have lower timing accuracy constraints.

When the power control register PWR_CTRL2.IWDGRSTEN bit is '1' and the IWDG counter reaches 0, a system reset will be generated (if this bit is '0', the IWDG will count but not reset).

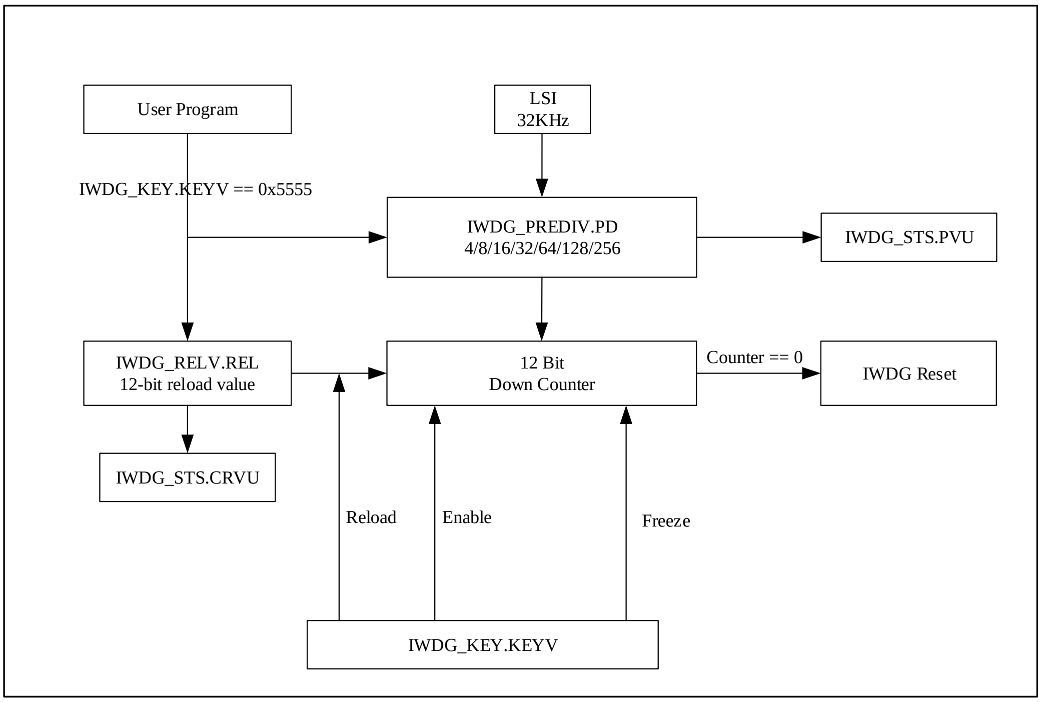
Note: This chapter is based on the system default IWDGRSTEN=1 discussion..

15.2 Main Feature

- Independent 12-bit down-counter
- RC oscillator provides independent clock source, which can also operate in SLEEP, STOP0 and STANDBY mode
- Reset and low-power wake-up can be matched
- A system reset occurs when the down counter reaches 0x0000 (if watchdog activated)

15.3 Functional Description

Figure 15-1 Functional Block Diagram Of The Independent Watchdog Module



Note: Watchdog function is in V_{DD} power supply area, and it can still work normally in SLEEP, STOP0 and STANDBY modes.

To enable IWDG, we need to write 0xCCCC to IWDG_KEY.KEYV[15:0] bits. Counter starts counting down from reset value (0xFFF). When counter count to 0x000, it generates a reset signal (IWDG_RESET) to MCU. Other than that, as long as 0xAAAA (reload request) is write to IWDG_KEY.KEYV[15:0] bits before reset, the counter value is set to the reload value in the IWDG_RELV.REL[11:0] bits and prevents the watchdog from resetting the entire device.

If the "hardware watchdog timer" function is enabled through the selection byte, the watchdog will automatically start running after the system is powered on and will generate a system reset, unless the software reloads the counter before it reaches ‘0’.

15.3.1 Register Access Protection

IWDG_PREDIV and IWDG_RELV register are write protected. To modify the value of those two register, user needs to write 0x5555 to IWDG_KEY.KEYV[15:0] bits. Writing other value enables write protections again. IWDG_STS.PVU indicates whether the pre-scaler value update is on going. IWDG_STS.CRVU indicates whether the IWDG is updating the reload value. The hardware sets the IWDG_STS.PVU bit and/or IWDG_STS.CRVU bit when the pre-scaler value and/or reload value is updating. After the pre-scaler value and/or reload value update is complete, the hardware clears the IWDG_STS.PVU bit and/or IWDG_STS.CRVU bit.

The reload operation (IWDG_KEY.KEYV[15:0] configured with value of 0xAAAA) will also cause the registers to become write protected again.

15.3.2 Debug Mode

In debug mode (Cortex-M4 core stops), IWDG counter will either continue to work normally or stops, depending on DBG_CTRL.IWDG_STOP bit in debug module. If this bit is set to '1', the counter stops. The counter works normally when the bit is '0'. See the Section 37.3.2 on debugging module for details.

15.3.3 IWDG Freeze

Once the IWDG is enabled, either by hardware or software, the IWDG will not stop counting unless a system reset is generated or runtime freeze is configured by writing 0x4567 to IWDG_KEY.KEYV[15:0] bits. Users can also configure IWDG freeze in certain operating modes. IWDG provides freeze options in SLEEP, STOP0, and STANDBY modes. When the IWDG is enabled, it forces the LSI clock to turn on.

15.4 User Interface

IWDG module user interface contains 4 registers: Key Register (IWDG_KEY), Pre-scale Register (IWDG_PREDIV), Reload Register (IWDG_RELV) and Status Register (IWDG_STS).

15.4.1 Operate Process

When IWDG is reset enabled by software (write 0xAAAA to IWDG_KEY.KEYV[15:0] bits) or hardware (clear WDG_SW bit), it starts counting down from 0xFFF. Down counting gap is determined by pre-scale LSI clock. Once the counter is reloaded, each new round will start from the value in IWDG_RELV.REL[11:0] instead of 0xFFF.

When program is running normally, software needs to feed IWDG before counter reaches 0 and start a new round of down counting. When counter reaches 0, this indicates program malfunction. IWDG generates reset signal under this circumstance.

If user wants to configure IWDG prescaler and reload value register, it needs to write 0x5555 to IWDG_KEY.KEYV[15:0] first, then confirm IWDG_STS.CRVU bit and IWDG_STS.PVU bit. IWDG_STS.CRVU bit indicates reload value update is ongoing, IWDG_STS.PVU indicates prescale divider ratio is updating. Only when those two bit are 0 then user can update corresponding value. When update is on-going, hardware sets corresponding bit to 1. At this time, reading IWDG_PREDIV.PD[2:0] or IWDG_RELV.REL[11:0] is invalid since data needs to be synchronized to LSI clock domain. The value read from IWDG_PREDIV.PD[2:0] or IWDG_RELV.REL[11:0] will be valid after hardware clears the IWDG_STS.PVU bit or IWDG_STS.CRVU bit.

If the application uses more than one reload value or pre-scaler value, it must wait until the IWDG_STS.CRVU bit is reset before changing the reload value, the same as changing the pre-scaler value. However, after updating the pre-scale and/or the reload value, it is not necessary to wait until IWDG_STS.CRVU bit or IWDG_STS.PVU bit are reset before continuing code execution (even in case of low-power mode entry, the write operation is taken into account and will complete).

Prescale register and reload register controls the time that generates reset, as shown in Table 15-1.

Table 15-1 IWDG Counting Maximum And Minimum Reset Time

Pre-Scale Factor	PD[2:0]	Minmum（ms） RL[11:0]=0	Maximum（ms） RL[11:0]=0xFFF
/4	000	0.125	512
/8	001	0.25	1024
/16	010	0.5	2048
/32	011	1.0	4096
/64	100	2.0	8192
/128	101	4.0	16384
/256	11x	8.0	32768

15.4.2 IWDG Configuration Flow

Software flow：

1. Write 0x5555 to IWDG_KEY.KEYV[15:0] bits to enable write access of IWDG_PREDIV and IWDG_RELV registers；
2. Check IWDG_STS.PVU bit or IWDG_STS.CRVU bit, if they are 0, continue next step；
3. Configure IWDG_PREDIV.PD[2:0] bits to select pre-scale value；
4. Configure IWDG_RELV.REL[11:0] bits reload value；
5. Writing 0xAAAA to IWDG_KEY.KEYV[15:0] bits to upload counter with reload value；
6. Enable watchdog by software or hardware writing 0xCCCC to IWDG_KEY.KEYV[15:0] bits。

If user wants change pre-scale and reload value, repeat step 1~5. If not, just feed the dog with step 5.

15.5 IWDG Register

15.5.1 IWDG Register Overview

Table 15-2 IWDG Register Overview

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
000h	IWDG_KEY	Reserved																KEYV[15:0]															
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
004h	IWDG_STS	Reserved																													FRZF	CRVU	PVU
	Reset Value																														0	0	0
008h	IWDG_PREDIV	Reserved																													PD[2:0]		
	Reset Value																														0	0	0

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
00Ch	IWDG_RELV	Reserved																				REL[11:0]											
	Reset Value																					1	1	1	1	1	1	1	1	1	1	1	1

15.5.2 IWDG Key Register（IWDG_KEY）

Offset address：0x00

Reset value：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEYV[15:0]															

rc_w0

Bit Field	Name	Description
31:16	Reserved	Reserved, the reset value must be maintained.
15:0	KEYV[15:0]	Key value register: only certain value will serve particular function 0xCCCC: Start watch dog counter, does not have any effect if hardware watchdog is enable, (if hardware watchdog is selected, it is not limited by this command word) 0xAAAA: Reload counter with REL value in IWDG_RELV register to prevent reset. 0x5555: Disable write protection of IWDG_PREDIV and IWDG_RELV register 0x4567: IWDG freeze, stop the IWDG counter in RUN mode 0x89AB: IWDG unfreeze, the counter resumes counting

15.5.3 IWDG Status Register（IWDG_STS）

Offset address：0x04

Reset value：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													FRZF	CRVU	PVU
													r	r	r

Bit Field	Name	Description
31:3	Reserved	Reserved, the reset value must be maintained.
2	FRZF	IWDG freeze flag： 0: No freeze occurred 1: Freeze occurred

Bit Field	Name	Description
1	CRVU	Watchdog reload value update Reload value update: This bit indicates that the reload value is being updated. It is set by hardware and cleared by hardware. Software can only attempt to change the value of IWDG_RELV.REL[11:0] when the value of IWDG_KEY.KEYV[15:0] is 0x5555 and this bit is 0.
0	PVU	Watchdog prescaler value update Prescaler value update: This bit indicates that the prescaler value is being updated. It is set by hardware and cleared by hardware. Software can only attempt to change the value of IWDG_PREDIV.PD[2:0] when the value of IWDG_KEY.KEYV[15:0] is 0x5555 and this bit is 0.

15.5.4 IWDG Pre-Scaler Register（IWDG_PREDIV）

Offset address：0x08

Reset value：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													PD[2:0]		
													rw		

Bit Field	Name	Description
31:3	Reserved	Reserved, the reset value must be maintained.
2:0	PD[2:0]	Prescaler- factor Prescaler divider: with write access protection when IWDG_KEY.KEYV[15:0] is not 0x5555. The IWDG_STS.PVU bit must be 0 otherwise PD [2:0] value cannot be changed. Divide number is as follow: 000：divider /4 001：divider /8 010：divider /16 011：divider /32 100：divider /64 101：divider /128 Other : divider /256 *Note: Reading this register will return the pre-divided value from the VDD voltage domain. If a write operation is in progress, the read-back value may be invalid. Therefore, the read value is valid only when the IWDG_STS.PVU bit is ‘0’.*

15.5.5 IWDG Reload Register（IWDG_RELV）

Offset address：0x0C

Reset value：0x00000FFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				REL[11:0]											
				rw											

Bit Field	Name	Description
31:12	Reserved	Reserved, the reset value must be maintained.
11:0	REL[11:0]	Watchdog counter reload value. With write protection. Defines the reload value of the watchdog counter, which is loaded to the counter every time 0xAAAA is written to IWDG_KEY.KEYV[15:0] bits. The counter then starts to count down from this value. The watchdog timeout period can be calculated from this reloading value and the clock pre-scaler value, refer to Table 15-1. This register can only be modified when the IWDG_STS.CRVU bit is ‘0’. *Note: Reading this register will return the reload value from the VDD voltage domain. If a write operation is in progress, the read-back value may be invalid. Therefore, the read value is valid only when the IWDG_STS.CRVU bit is ‘0’.*

16 Window Watchdog（WWDG）

16.1 Introduction

The clock of the window watchdog (WWDG) is obtained by dividing the APB1 clock frequency by 4096, and it is used to detect abnormal program operation through the configuration of the time window. Therefore, WWDG is suitable for precise timing, and is often used to monitor software failures caused by external disturbances or unforeseen logic conditions, which cause an application to deviate from its normal operating sequence. A system reset occurs when the WWDG down counter is refreshed before reaching the window register value or after the WWDG_CTRL.T6 bit becomes 0.

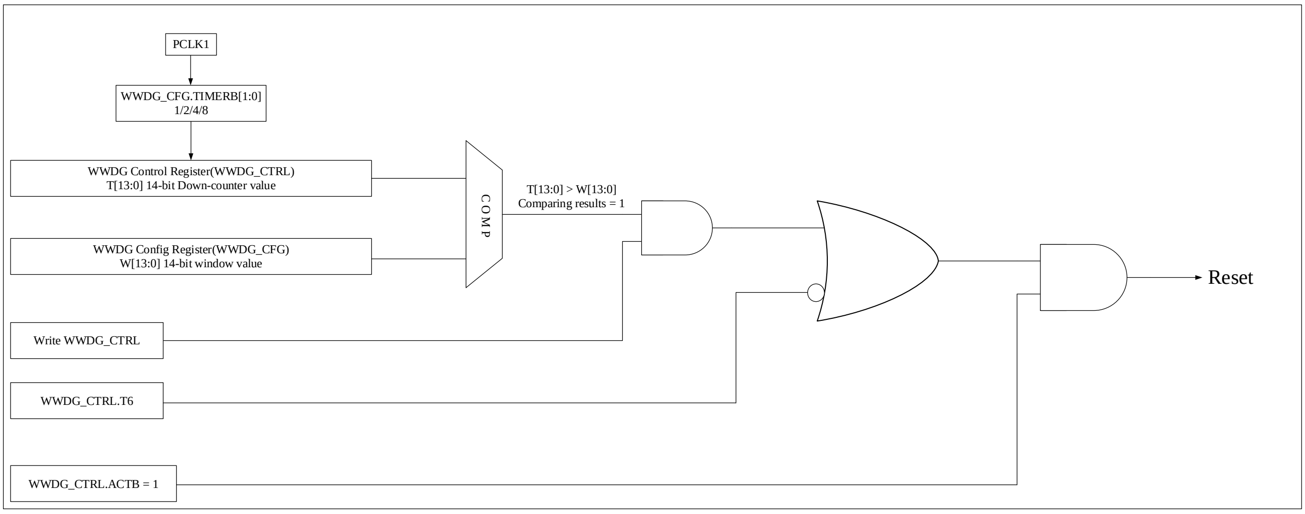
16.2 Main Features

- 14-bit programmable running down counter
- After WWDG is enabled, a reset occurs under the following conditions
  - The value of the down counter is less than 0x40.
  - When the down counter value is greater than the value of the window register, it is reloaded.
- Early wake-up interrupt: If the watchdog is started and the interrupt is enabled, wake-up interrupt (WWDG_CFG.EWINT) will be generated when the count value reaches 0x40.

16.3 Function Description

If the watchdog is activated (the WWDG_CTRL.ACTB bit), when the 14-bit (WWDG_CTRL.T[13:0]) down-counter reaches 0x3F(WWDG_CTRL.T6 bit is cleared), or the software reloads the counter when the counter value is greater than the value of the window register, a system reset will be generated. In order to avoid system reset, the software must periodically refresh the counter value in the window during normal operation.

Figure 16-1 Window Watchdog Block Diagram



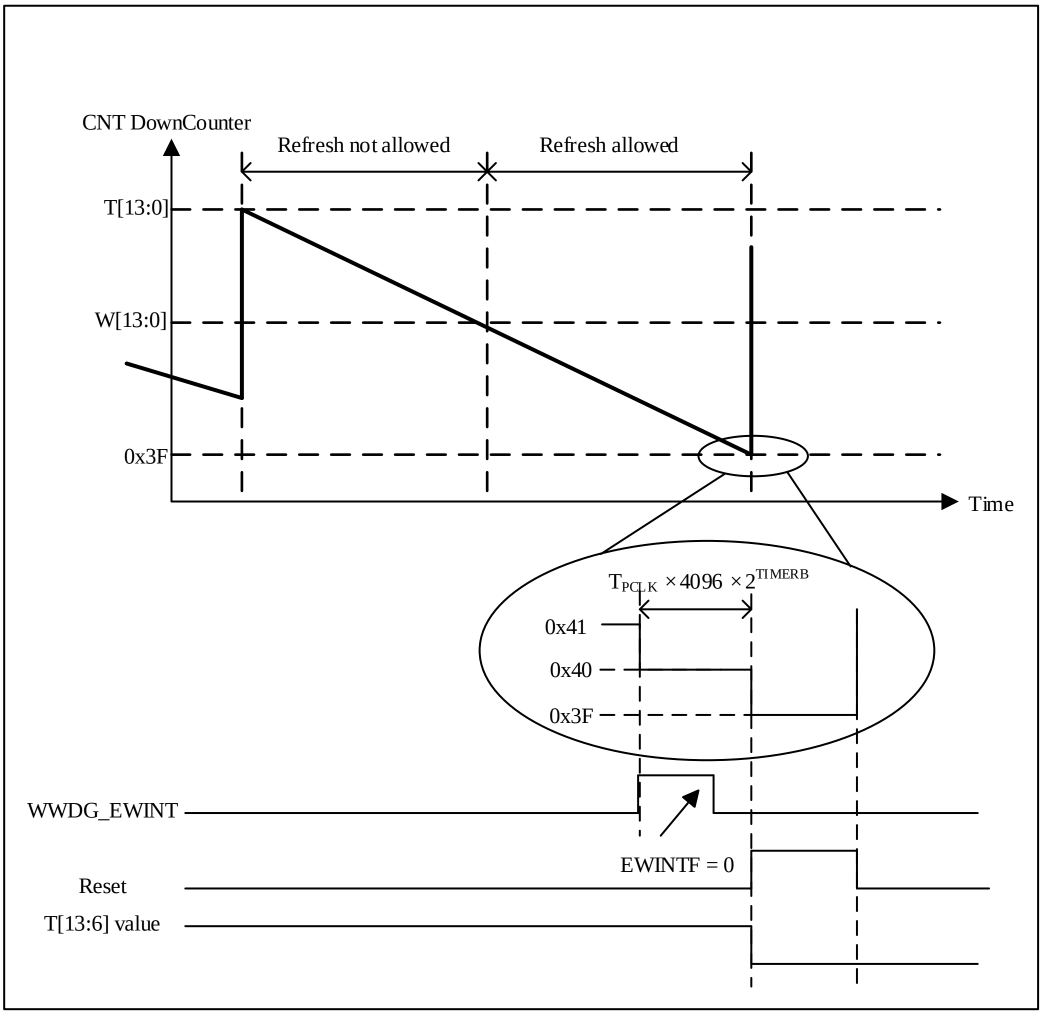
Set the WWDG_CTRL.ACTB bit to enable the watchdog, and thereafter, the WWDG will remain on until reset occurs. The 14-bit down-counter runs independently, and the counter keeps counting down whether WWDG is enabled or not. Therefore, before enabling the watchdog, Need to set one of the high 8 bits MSB (WWDG_CTRL.T[13:6]) to 1, preventing reset immediately after enable. The prescaler value set by the clock APB1 and WWDG_CFG.TIMERB[1:0] bits determine the decrement speed of the counter. WWDG_CFG.W[13:0] bits set the upper limit of the window.

When the down-counter is refreshed before reaching the window register value or after WWDG_CTRL.T6 bit becomes 0, a system reset will be generated. Figure 16-2 describes the operating process of the window registers.

Set the WWDG_CFG.EWINT bit to enable early wake-up interrupt. When the count-down counter reaches 0x40, an interrupt will be generated. You can analyze the cause of software failure or save important data in the corresponding interrupt service routine (ISR), and reload the counter to prevent WWDG from resetting. Write '0' to the WWDG_STS.EWINTF bit to clear the interrupt.

16.4 Timing For Refresh Watchdog And Interrupt Generation

Figure 16-2 Refresh Window And Interrupt Timing Of WWDG



Watchdog refreshing window is between WWDG_CFG.W[13:0] value (maximum value 0x3FFF) and 0x3F, refresh outside this window will generates reset request to MCU. Counter count down from 0x3FFF to 0x3F using scaled

APB clock, the maximum counting time and minimum counting time is shown in following Table（assuming APB clock 120 MHz）, with calculate equation：

$$\mathrm{T}_{WWDG} = T_{PCLK1} \times 4096 \times 2^{TIMERB} \times (T[13:0] - 0X3F + 1)$$

In which：

$\mathrm{T}_{\mathrm{WWDG}}$:WWDG timeout

$\mathrm{T}_{\mathrm{PCLK1}}$:APB1 clock interval in ms

Minimum-maximum timeout value at PCLK1=120MHz

Table 16-1 Maximum And Minimum Counting Time Of WWDG

TIMERB	Maximum counting（ms）	Minimum counting（ms）
0	556.92	0.0341
1	1113.84	0.0682
2	2227.68	0.136
3	4455.36	0.273

16.5 Debug Mode

In debug mode (Cortex$^{\mathrm{TM}}$-M4F core stops), WWDG counter will either continue to work normally or stop, depending on DBG_CTRL.WWDG_STOP bit in debug module. If this bit is set to ‘1’, the counter stops. The counter works normally when the bit is ‘0’. See the chapter 37.3.2 on debugging module for details.

16.6 User Interface

16.6.1 WWDG Configuration Flow

1. Configure RCC_APB1PCLKEN.WWDGEN[11] bit to enable the clock of WWDG module.
2. Software setting WWDG_CFG.TIMERB[15:14] bits to configure pre-scale factor for WWDG.
3. Software configure WWDG_CTRL.T[13:0] bits, setting starting value of counter. Need to set one of the high 8 bits MSB (WWDG_CTRL.T[13:6]) to 1, preventing reset right after enable.
4. Configure WWDG_CFG.W[13:0] bits to configure upper boundary window value.
5. Setting WWDG_CTRL.ACTB[14] bit to enable WWDG.
6. Software operates WWDG_STS.EWINTF[0] bit to clear wake-up interrupt flag.
7. Configure WWDG_CFG.EWINT[16] bit to enable early wake-up interrupt.

16.7 WWDG Register

16.7.1 WWDG Register Overview

Table 16-2 WWDG Register Overview

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	WWDG_CFG	Reserved															EWINT	TIMERB[1:0]		W[13:0]													
	Reset value																0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x04	WWDG_CTRL	Reserved																	ACTB	T[13:0]													
	Reset value																		0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x08	WWDG_STS	Reserved																															EWINTF
	Reset value																																0

16.7.2 WWDG Config Register（WWDG_CFG）

Offset address：0x00

Reset value：0x00003FFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															EWINT
															rs

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIMERB[1:0]		W[13:0]													
rw		rw													

Bit Field	Name	Description
31:17	Reserved	Reserved, the reset value must be maintained.
16	EWINT	Early wake-up interrupt When set, an interrupt occurs whenever the counter reaches the value 0x40. This interrupt is only cleared by hardware after a reset.
15:14	TIMERB[1:0]	Timer base The time base of the pre-scaler can be modified as follows： 00：CK Counter Clock（PCLK1 div 4096）div 1 01：CK Counter Clock（PCLK1 div 4096）div 2 10：CK Counter Clock（PCLK1 div 4096）div 4 11：CK Counter Clock（PCLK1 div 4096）div 8

Bit Field	Name	Description
13:0	W[13:0]	14-bit window value These bits contain the window value to be compared to the down counter

16.7.3 WWDG Control Register（WWDG_CTRL）

Offset address：0x04

Reset value：0x00003FFF

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="16">Reserved</td></tr>
</table>

<table>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>Reserved</td><td>ACTB</td><td colspan="14">T[13:0]</td></tr>
<tr><td></td><td>rs</td><td colspan="14">rw</td></tr>
</table>

Bit Field	Name	Description
31:15	Reserved	Reserved, the reset value must be maintained.
14	ACTB	Activation bit When ACTB=1, the watchdog can generate a reset. This bit is set by software and only cleared by hardware after a reset. When ACTB = 1, the watchdog can generate a reset. 0: Disable watchdog 1: Enable warchdong
13:0	T[13:0]	These bits contain the value of the watchdog counter. It is decremented every (4096x2^{TIMERB}) PCLK1 cycles. A reset is produced when it rolls over from 0x40 to 0x3F (T6 becomes cleared).

16.7.4 WWDG Status Register（WWDG_STS）

Offset address：0x08

Reset value：0x0000

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="16">Reserved</td></tr>
</table>

<table>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="15">Reserved</td><td>EWINTF</td></tr>
<tr><td colspan="15"></td><td>rc_w0</td></tr>
</table>

Bit Field	Name	Description
31:1	Reserved	Reserved, the reset value must be maintained.
0	EWINTF	Early wake-up interrupt flag This bit is set by hardware when the counter has reached the value 0x40. It must be cleared by software by writing ‘0’. A write of ‘1’ has no effect. This bit is also set if the interrupt is not enabled.

17 Analog To Digital Conversion（ADC）

17.1 Introduction

The 12-bit ADC is a high-speed analog-to-digital converter using successive approximation. There are four ADCs, (ADC1/ADC2, ADC3/ADC4) can be combined as dual ADC; ADC1/ADC2/ADC3 can be combined as triple ADC. Each ADC can have up to 19 multiplexed channels, The A/D conversion of each channel has four execution modes: single, continuous, scan or discontinuous. ADC measurements are stored (left-aligned/ right-aligned) in 16-bit data registers. The application can detect that the input voltage is within user-defined high/low thresholds by analog watchdog1/2/3, and the maximum frequency of the input clock to the ADC is 80MHz.

17.2 ADC Main Features

- Supports 4 ADC, supports single-ended and differential inputs
  - ADC1 supports 16 external channels, 3 internal channels, where 2 external channels can be connected to the PGA1 output;
  - ADC2 supports 18 external channels, 1 internal channel, where 2 external channels can be connected to the PGA2 output;
  - ADC3 supports 19 external channels, where 2 external channels can be connected to the PGA3 output;
  - ADC4 supports 19 external channels, where 2 external channels can be connected to the PGA4 output;
- Support 12-bit, 10-bit, 8-bit, 6-bit configurable resolution
  - The highest sampling rate 4.7MSPS under 12bit resolution
  - The highest sampling rate 6MSPS under 10bit resolution
  - The highest sampling rate 7.2MSPS under 8bit resolution
  - The highest sampling rate 9MSPS under 6bit resolution
- ADC clock source is divided into working clock source, sampling clock source and timing clock source
  - Only AHB_CLK can be configured as the working clock source,up to 240MHz.
  - PLL can be configured as a sampling clock source, up to 80MHz, supporting prescaler 1,2,4,6,8,10,12.
  - The AHB_CLK can be configured as the sampling clock source, up to 80MHz, and supporting prescaler 1,2,4,6,8,10,12,16,32.
  - The timing clock is used for internal timing functions and the frequency must be configured to 1MHz.
- Support trigger sampling, including EXTI/TIMER
- Programmable channel sampling interval
- Each ADC has 3 analog watchdogs
- When the ADC is ready, sampling is completed, conversion is finished or Analog Watchdog 1/2/3 events occur, an interrupt can be triggered.

- Support 4 conversion modes
  - Single conversion
  - Continuous conversion
  - Discontinuous conversion
  - Scan conversion
- Support self-calibration mode
- Support FIFO mode
- Data alignment with embedded data consistency
- Start conversion mode
  - Start regular and injected conversions by software
  - Start regular and injected conversions using externally configurable polarity triggers (GPIO input events or internal timer events)
- Oversampling
  - Adjustable oversampling ratios x2, x4, x8, x16, x32, x64, x128, x256
  - Data right shift of 0-8 bits configurable
  - 16-bit data result register
- Data preprocessing
  - Support gain compensation
  - Support offset compensation
- Multiple ADC modes
  - Dual ADC mode: Combination of ADC1 and ADC2, combination of ADC3 and ADC4
  - Triple ADC mode: Combination of ADC1, ADC2, and ADC3
- ADC power requirements: 1.8V to 3. 6V
- ADC input voltage range: VREF- ≤ VIN ≤ VREF+

17.3 ADC Function Description

The following diagram shows the block diagram of an ADC module, Table 17-1 is the description of the ADC pins.

Figure 17-1 ADC Block Diagram

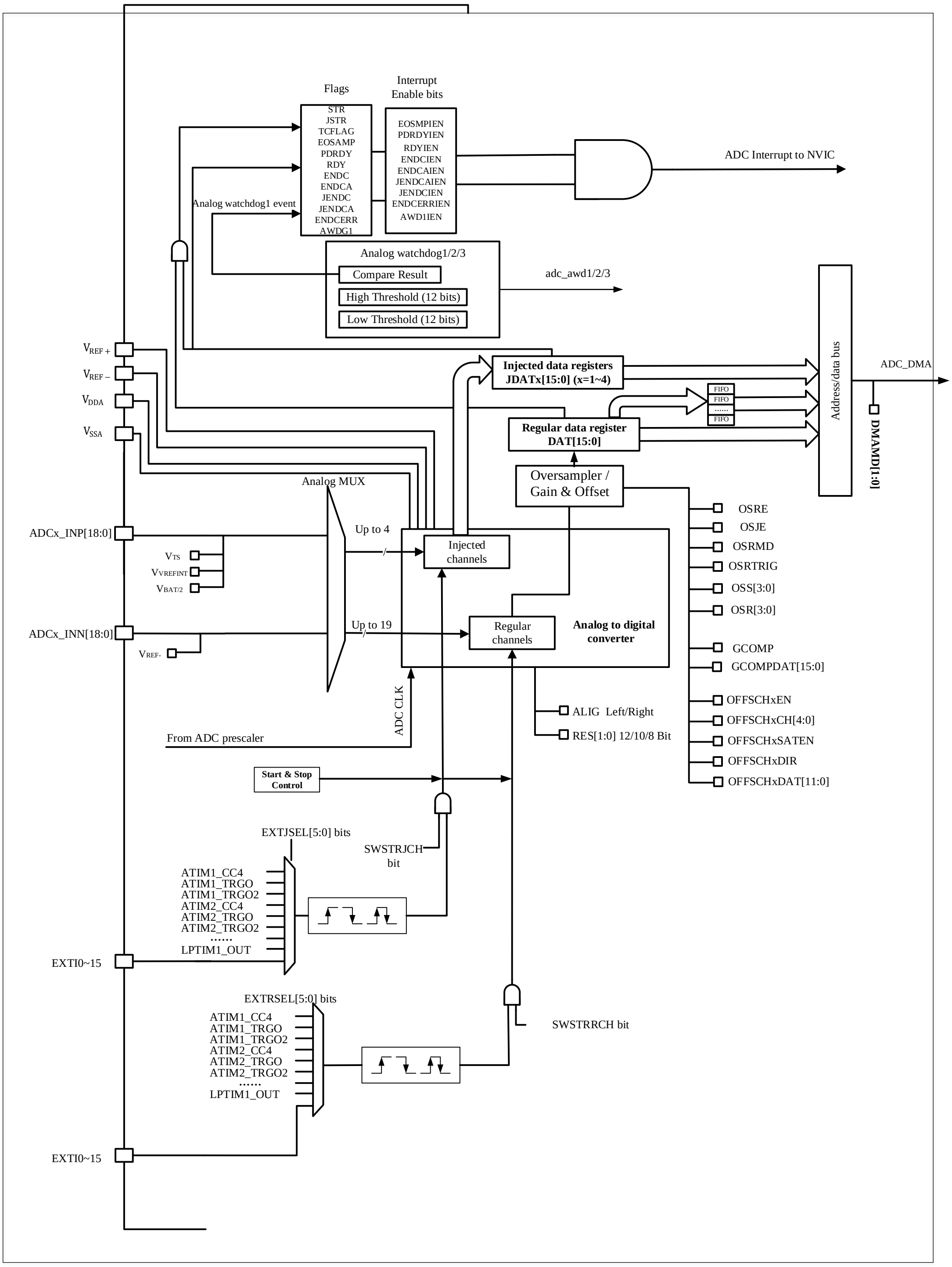


Table 17-1 ADC Pins

Name	Signal Types	Description
V_{REF+}	The positive reference voltage used by the DAC $1.8V \leq V_{REF+} \leq V_{DDA}$（3.6V）	Input, positive analog reference voltage
V_{DDA}	Input, analog power supply	Analog power supply equivalent to VDD: $1.8V \leq V_{DD} \leq 3.6V$
V_{SSA}	Input, analog power ground	Analog ground equivalent to VSS
ADCx_INP[i]	Positive analog input signal	Support for up to 19 external analog input channels
ADCx_INN[i]	Negative analog input signal	Support for up to 19 external analog input channels
V_{TS}	Input signal	Output voltage of the internal temperature sensor
V_{REFINT}	Input signal	Output voltage of the internal reference voltage
V_{BAT}	Input signal	External battery voltage
adc_awdx	Output signal	Internal analog watchdog output signal, connected to on-chip timer. (x = analog watchdog number 1, 2, 3)

17.3.1 ADC Clock

ADC requires three clocks，ADC_CLK、HCLK、ADC_1MCLK.

- HCLK is used for the register access.
- ADC_CLK is the working clock of ADC. ADC_CLK has two sources (divided from HCLK or divided from PLL). The HCLK divided clock and system clock are synchronous clock, while the PLL divided clock and system clock are asynchronous clock. The advantage of using a synchronous clock is that there is no uncertainty when triggering the ADC to respond to the trigger.The advantage of using PLL's divider clock is that the ADC's working clock can be handled independently without affecting other modules attached to the HCLK.
- ADC_1MCLK is used for internal timing function, configured in RCC, the frequency must be configured to 1MHz.

Notes:

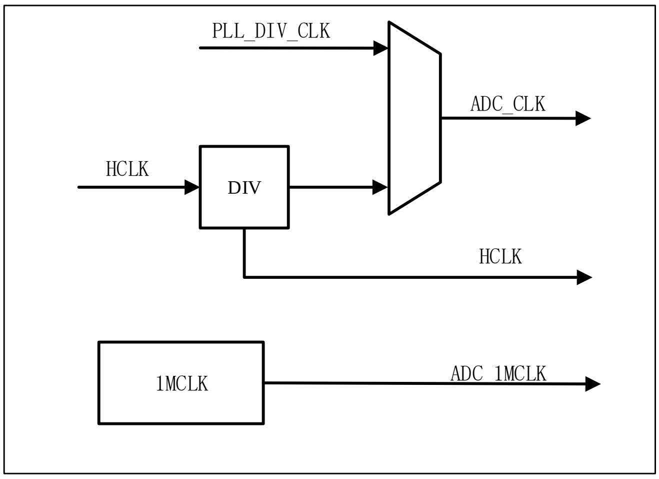
(1) When configuration PLL as a clock source, the maximum frequency can reach 80 MHz, supporting prescaler *factors of 1,2,4,6,8,10,12.*

(2) The AHB_CLK prescaler *can be configured as a working clock up to 80MHz. The AHB_CLK* prescaler *can be 1,2,4,6,8,10,12,16,32.*

(3) When switching the ADC_1M clock source, you need to ensure that the HSI clock is turned on.

(4) After ADC reset, at least 4 ADC_CLK (maximum frequency of 80M) cycles of delay are required before reconfiguring the ADC registers

Figure 17-2 ADC Clock



17.3.2 ADC Switch Control

You can proceed to the next step only after the power-up process is complete. You can set the ADC_CTRL2.ON bit to turn on the ADC. When the ADC_CTRL2.ON bit is set, it wakes up the ADC from the power-off state. After a power-on delay of ADC (t_{STAB}), You can check if the power-up is complete by polling the ADC_STS.RDY bit.

The conversion can be stopped by clearing the ADC_CTRL2.ON bit and placing the ADC in power-off mode. In this mode, the ADC consumes almost no power consumption (just a few μA). Power-down can be checked by polling the ADC_STS.PDRDY bit.

When the ADC is disabled, the default mode is power-down.

17.3.3 Channel Selection

Each channel can be configured as a regular sequence and an injection sequence.

The injection sequence consists of multiple conversions, up to a maximum of 4. The ADC_JSEQ register specifies the injection channel and the conversion sequence of the injection channel. The ADC_JSEQ.JLEN[1:0] bits specified injection sequence length.

The regular sequence consists of multiple conversions, up to a maximum of 16. The ADC_RSEQx registers specify the regular channels and the conversion sequence of the regular channels. The ADC_RSEQ1.LEN[3:0] bits specified regular channel sequence length.

Note: During conversion, changes to the ADC_RSEQx or ADC_JSEQ registers are prohibited; the ADC_RSEQx or ADC_JSEQ registers can only be changed when the ADC is idle.

Taking ADC1 as an example:

Users can select the positive terminal of ADC1_CH1 to be connected to the positive terminal output of PGA1 or external IO (i.e., ADC1_INP1) through the register ADC_CTRL3.INP1SEL;

Users can select the negative terminal of ADC1_CH1 to be connected to the negative terminal output of PGA1 or external IO (i.e., ADC1_INN1) through the register ADC_CTRL3.INN1SEL;

Users can select the positive terminal of ADC1_CH2 to be connected to the negative terminal output of PGA1 or external IO (i.e., ADC1_INP2) through the register ADC_CTRL3.INP2SEL.

Figure 17-3 Connection between ADC1 and ADC2 channels

- VREFBUF: Internal ADC reference voltage, 2.048V/2.5V/2.9V
- VREFINT: Internal 1.2V BG Buffer
- VTS: Internal temperature sensor
- VPGA1OUT_P/VPGA1OUT_N：Output of PGA1, supports one differential output and two independent single-ended outputs.
  The two single-ended PGAs can be independently controlled to output or not.
- VPGA2OUT_P/VPGA2OUT_N: Output of PGA2, supports one differential output and two independent single-ended outputs.
  The two single-ended PGAs can be independently controlled to output or not.
- The channels of ADC1 and ADC2 can be selected to be either from the chip's ADC pins or from the internal VPGA1OUT_P/VPGA1OUT_N and VPGA2OUT_P/ VPGA2OUT_N.

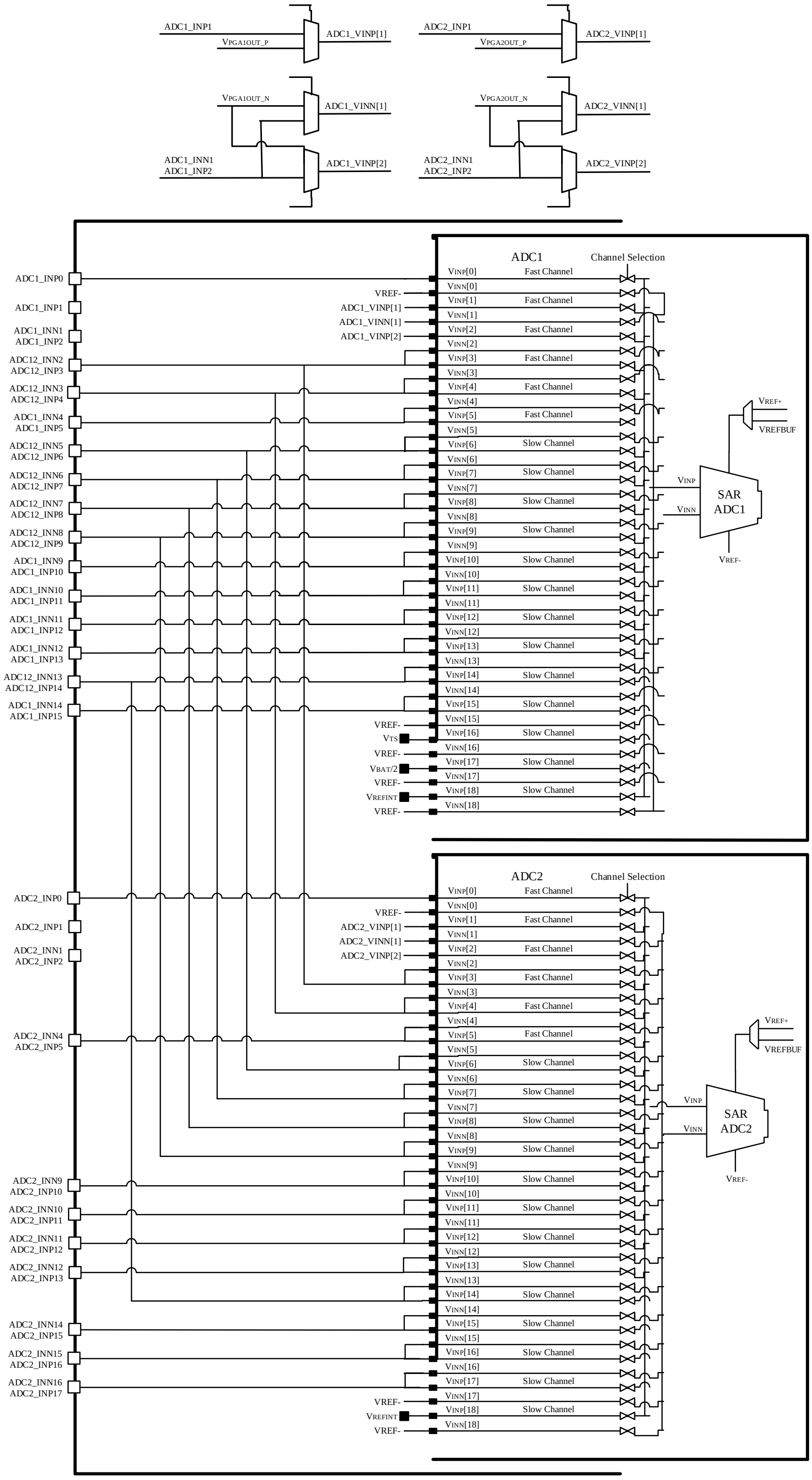
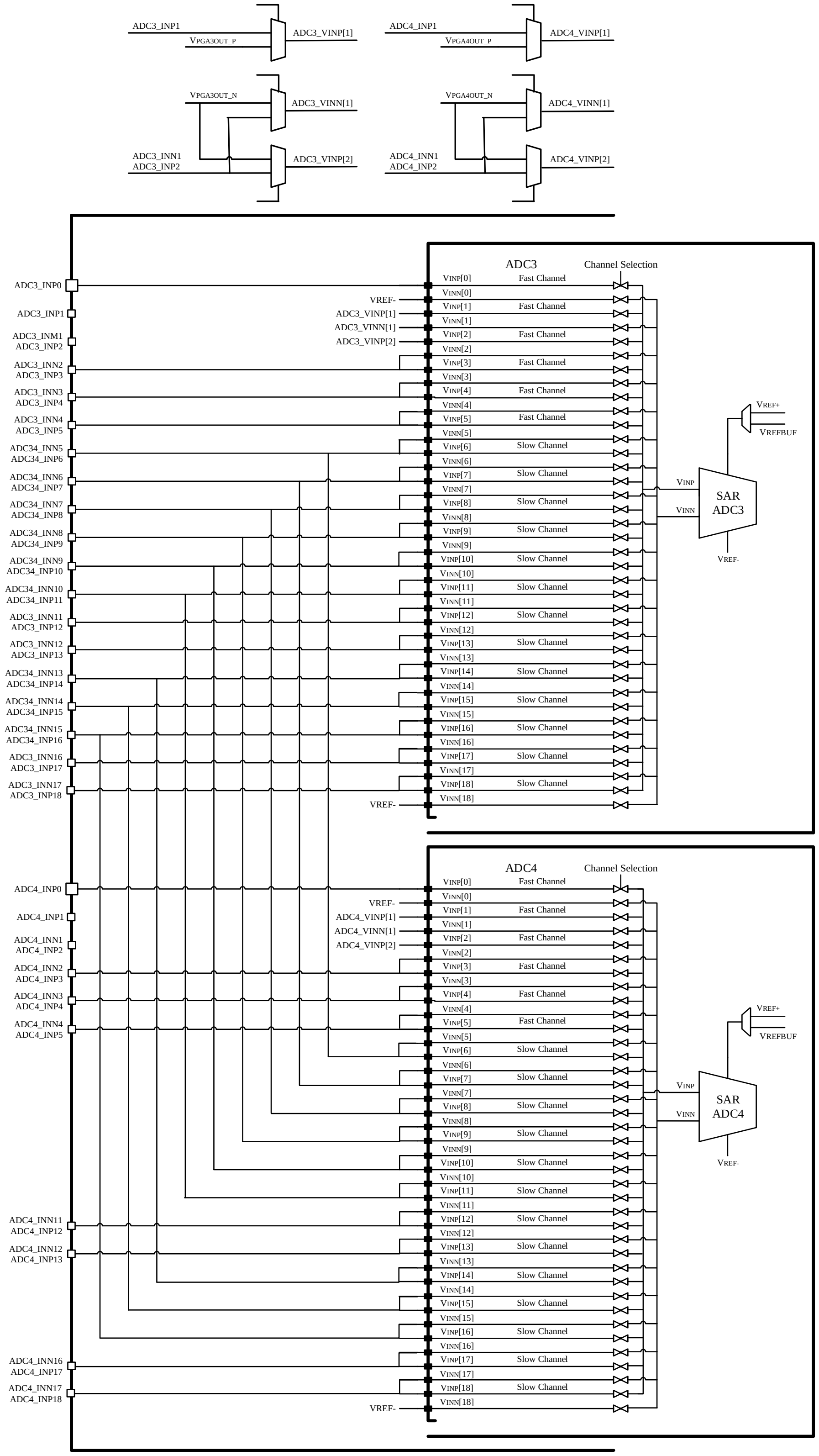


Figure 17-4 Connection between ADC3 and ADC4 channels

- VREFBUF: Internal ADC reference voltage, 2.048V/2.5V/2.9V
- VREFINT: Internal 1.2V BG Buffer
- VTS: Internal temperature sensor
- $V_{PGA3OUT\_P}$/$V_{PGA3OUT\_N}$: Output of PGA3, supports one differential output and two independent single-ended outputs. The two single-ended PGAs can be independently controlled to output or not.
- $V_{PGA4OUT\_P}$/$V_{PGA4OUT\_N}$: Output of PGA4, supports one differential output and two independent single-ended outputs. The two single-ended PGAs can be independently controlled to output or not.
- The channels of ADC3 and ADC4 can be selected to be either from the chip's ADC pins or from the internal VPGA3OUT_P/VPGA3OUT_N and VPGA4OUT_P/VPGA4OUT_N.



17.3.4 Internal Channel

- The temperature sensor is connected to channel ADC1_IN16;
- $V_{BAT/2}$ is connected to channel ADC1_IN17;
- V_{REFINT} is connected to ADC1_IN18 and ADC2_IN18;

ADC internal channels can be converted by injection or regular channels.

17.3.5 Single-Ended And Differential Input Channels

Users can configure channels as single-ended or differential input by writing to the ADC_DIFSEL.DIFSEL[17:0] bits.

In single-ended input mode, the analog voltage to be converted for channel 'i' is the difference between the external voltage VINP[i] (positive input) and VREF- (negative input).

In differential input mode, the analog voltage to be converted for channel 'i' is the difference between the external voltage VINP[i] (positive input) and VINN[i] (negative input).

The output data in differential mode is unsigned data. When VINP[i] is VREF- and VINN[i] is VREF+, the output data is 0x000 (in 12-bit resolution mode); when VINP[i] is VREF+ and VINN[i] is VREF-, the output data is 0xFFF. The conversion formula is as follows：

$$OUT \quad = \frac{Full\_Scale}{2} * (1 + \frac{VINP - VINN}{VREF +})$$

OUT: represents the value converted to the ADC result register;

Full_Scale: represents the full-scale value of the ADC. In the case of 12 bits, it ranges from 0 to 0xFFF;

VREF+: represents the reference voltage of the ADC;

VINP: represents the positive input voltage of the ADC;

VINN: Represents the negative input voltage of the ADC;

17.3.6 ADC Calibration

To reduce the errors, the ADC will have a embedded self-calibration mechanism. Before the A/D conversion, this self-calibration mechanism is used to calculate a calibration factor on each capacitor. Errors due to changes in the internal capacitor bank during conversion are eliminated by this calibration factor. Calibration must be performed before any ADC operation. Calibration can eliminate system errors between chips.

Each time the ADC loses power, the internal analog calibration coefficients are lost (eg:when the chip enters standby mode or VBAT mode). In this case, to avoid wasting time recalibrating the ADC, the calibration coefficients can be rewritten to the ADC_CALFACT.CALFACTS[6:0] and ADC_CALFACT.CALFACTD[6:0] registers. However, this is conditional on the software storing the calibration coefficients obtained during the previous calibration in power down retention area (eg:Flash). Then, enable the automatic calibration loading function (ie:ADC_CTRL3.CALALD=1), which can save the time required for recalibration.

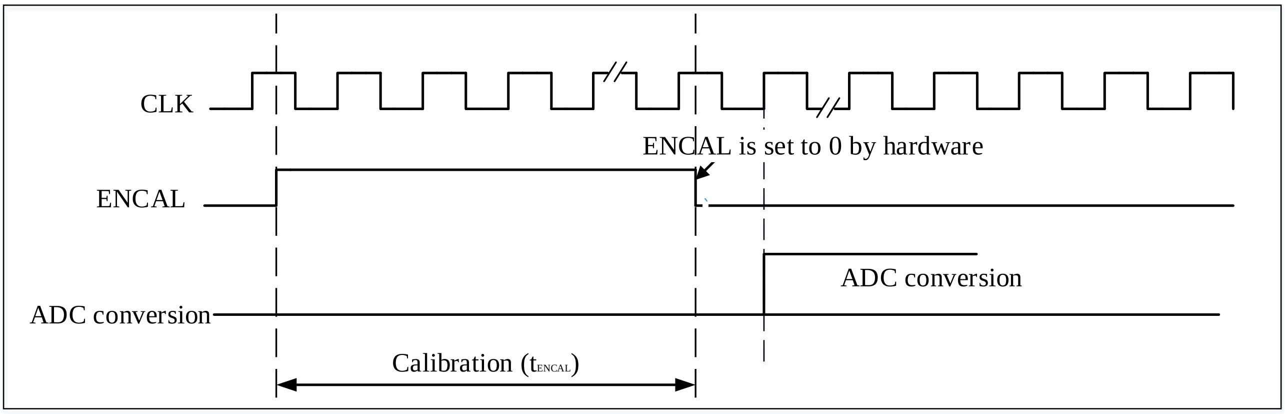
When ADC_CTRL2.ON=0, the internal calibration coefficients are not lost. However, if the ADC is not used for a long time, it is still recommended that users recalibrate the ADC before enabling its operation. If you want to reset the calibration coefficients, you can set ADC_CTRL2.RSTCAL=1.

ADC calibration process:

1. Ensure that the ADC is in the enabled state, ADC_CTRL2.ON=1;
2. When enabling single-ended mode calibration, set ADC_CTRL3.CALDIF=0; otherwise, set ADC_CTRL3.CALDIF=1;
3. Set ADC_CTRL2.ENCAL=1 to start calibration;
4. Wait for ADC_CTRL2.ENCAL to be cleared to 0 by hardware, indicating the completion of calibration. After calibration is completed, the calibration coefficients will be written to the ADC_CALFACT register (if single-ended input mode is enabled, the calibration coefficients will be written to ADC_CALFACT.CALFACTS[6:0]; if differential input mode is enabled, the calibration coefficients will be written to ADC_CALFACT.CALFACTD[6:0]);
5. Users can choose to read the calibration coefficients from the ADC_CALFACT register.

Note: During the calibration process, the automatic calibration loading function (ADC_CTRL3.CALALD = 1) should not be enabled, as it may affect the calibration accuracy.

Figure 17-5 Calibration Timing Diagram



17.3.7 Single Conversion Mode

The ADC can enter the single conversion mode by configuring ADC_CTRL2.CTU to 0.

In this mode, conversions can be started in the following ways：

1. Software trigger for regular channel: ADC_CTRL2.SWSTRRCH=1, external trigger source ADC_CTRL2.EXTRSEL[5:0] configured as SWSTRRCH;

2.: Software trigger for injected channel: ADC_CTRL2.SWSTRJCH=1, external trigger source ADC_CTRL2.EXTJSEL[5:0] configured as SWSTRJCH;

3. External event trigger (regular channel or injected channel);

When converting a regular channel, when the conversion is completed, the end of conversion flag for any regular channel (ADC_STS.ENDCA) will be set to 1. If the end of conversion interrupt enable for any regular channel (ADC_INTEN.ENDCAIEN) is set to 1, an interrupt will be generated, and the converted data will be stored in the ADC_DAT register.

When converting an injected channel, when the conversion is completed, the end of conversion flag for any injected channel (ADC_STS.JENDCA) will be set to 1. If the end of conversion interrupt enable for any injected channel (ADC_INTEN.JENDCAIEN) is set to 1, an interrupt will be generated, and the converted data will be stored in the ADC_JDATx register.

When converting a sequence of regular channels, when the conversion is completed, the end of conversion flag for the regular channel sequence (ADC_STS.ENDC) will be set to 1. If the end of conversion interrupt enable for regular channels (ADC_INTEN.ENDCIEN) is set to 1, an interrupt will be generated, and the converted data will be stored in the ADC_DAT register.

When converting a sequence of injected channels, when the conversion is completed, the end of conversion flag for the injected channel sequence (ADC_STS.JENDC) will be set to 1. If the end of conversion interrupt enable for injected channels (ADC_INTEN.JENDCIEN) is set to 1, an interrupt will be generated, and the converted data will be stored in the ADC_JDATx register.

After a single conversion, the ADC will stop.

17.3.8 Continuous Conversion Mode

The ADC can enter continuous conversion mode by setting ADC_CTRL2.CTU to 1. In this mode, both software for regular channel and external events can trigger the ADC to start conversions, and the ADC will continuously convert the selected channels.

After the conversion starts, when a sequence of regular channel conversions is completed, the end of conversion flag for the regular channel sequence (ADC_STS.ENDC) will be set to 1. If the end of conversion interrupt enable for regular channels (ADC_INTEN.ENDCIEN) is set to 1, an interrupt will be generated. The data after each conversion will be stored in the ADC_DAT register.

17.3.9 Scan Mode

By setting ADC_CTRL1.SCAMD to 1, the scan conversion mode can be enabled. By configuring the four registers ADC_RSEQ1, ADC_RSEQ2, ADC_RSEQ3, ADC_JSEQ, channel sequences for conversion can be selected. The ADC will perform scan conversions on all selected regular or injected channels. After the conversion starts, the channels will be converted one by one. If ADC_CTRL2.CTU is set to 1 at this time, after the completion of conversions on all selected regular channels, the conversion will restart from the first channel in the conversion sequence. The injected channels do not support continuous conversion. The DMA function can be enabled by setting ADC_CTRL2.DMAMD[1:0] to a value other than '00'. DMA will transfer data to SRAM after regular channel conversions are completed. The data from injected channel conversions is always stored in the ADC_JDATx register.

Note: In dual ADC mode, the DMA function on the regular channels of ADC2 needs to be completed through the DMA of ADC1, and the DMA function on the regular channels of ADC4 needs to be completed through the DMA of ADC3. In triple ADC mode, the DMA function on the regular channels of ADC2 and ADC3 needs to be completed through the DMA of ADC1, while ADC4 is idle.

17.3.10 Injected Channel Management

17.3.10.1 Automatic injection

If the ADC_CTRL1.AUTOJC bit is set, the injected channels selected by ADC_JSEQ will automatically convert after the regular channels selected by ADC_RSEQx have completed conversion, with a maximum of 16+4 channels. Setting ADC_CTRL2.CTU will continuously convert the conversion sequence.

When this function is turned on, the external trigger of the injection channel needs to be turned off.

This function cannot be used with the discontinuous mode at the same time.

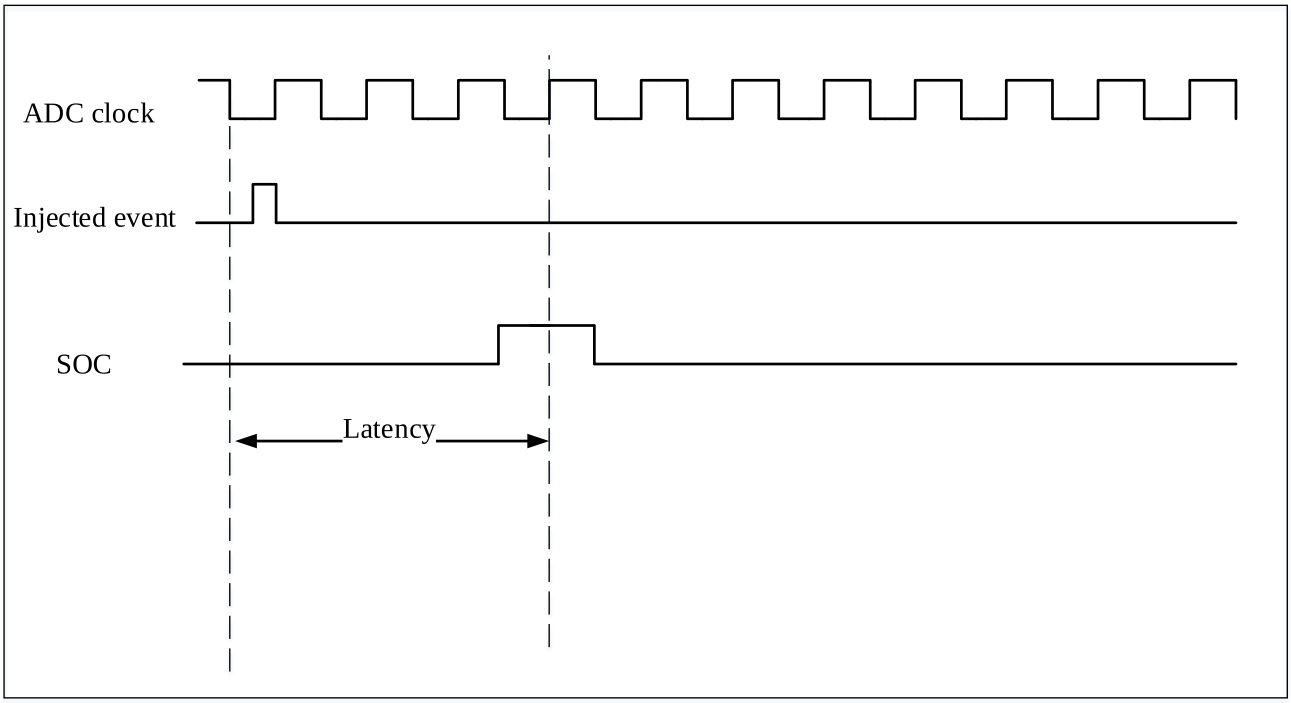
When the ADC clock prescale factor is 2, there is a delay of two ADC clock intervals when the conversion sequence changes from regular to injection or injection to regular. When the ADC clock prescale factor is 4 to 8, there is a delay of one ADC clock intervals when the conversion sequence changes from regular to injection or injection to regular.

17.3.10.2 Triggered injection

Set ADC_CTRL1.AUTOJC to 0 to enable the triggered injection function. In this mode, initiate the continuous conversion of regular channels by setting ADC_CTRL2.SWSTRRCH or by external triggering. During the conversion of regular channels, if an external injection trigger occurs, the current conversion is paused, and the injection sequence channels start their conversion. Once the injection sequence channels have completed conversion, the interrupted regular sequence channel conversion resumes. If a regular event occurs during the injection conversion, the regular sequence channels will start their conversion after the injection sequence channels have completed conversion.

When using this feature, the time interval between the injected channel triggers needs to be greater than the time required for the injected sequence to complete the conversion.

Figure 17-6 Injected Conversion Delay



Note:For the maximum delay value, please refer to the electrical characteristics section in the data manual.

17.3.11 Discontinuous Mode

17.3.11.1 Regular channels

Configure ADC_CTRL1.DREGCH to 1 to enable the discontinuous mode on the regular channel, obtain the regular sequence by configuring ADC_RSEQ1, ADC_RSEQ2, ADC_RSEQ3, and configure ADC_CTRL1.DCTU[2:0] to control the conversion of n channels each time a trigger signal is generated.

When the trigger signal is generated, it will convert n channels of the regular sequence and then stop, until the next trigger signal is generated. Next trigger will continue to convert n channels from the point where the previous conversion stopped, until all channels of the regular sequence are converted (If the last trigger occurs and the remaining channels in the conversion sequence are less than n, only the remaining channels will be converted and the conversion will be stopped), and the end of conversion flag bit will also be set to 1. When the conversion of all channels in the conversion sequence is completed, and the next trigger signal occurs ,the conversion starts from the first channel of the regular sequence again.

17.3.11.2 Injected channels

Configure ADC_CTRL1.DJCH to 1 to enable the discontinuous mode on the injection channel, obtain the injection sequence by configuring ADC_JSEQ.

When the trigger signal is generated, it will convert 1 channel of the injection sequence and then stop,until the next trigger signal is generated. Next trigger will continue to convert 1 channel from the point where the previous conversion stopped until all channels of the injection sequence are converted, and the end of conversion flag bit will also be set to 1. When the conversion of all channels in the conversion sequence is completed, and the next trigger signal occurs ,the conversion starts from the first channel of the injection sequence again.

Only one of injection conversion and regular conversion can be set to discontinuous mode at the same time, and the automatic injection function and discontinuous mode cannot be set at the same time.

17.3.12 Stop Ongoing Conversions

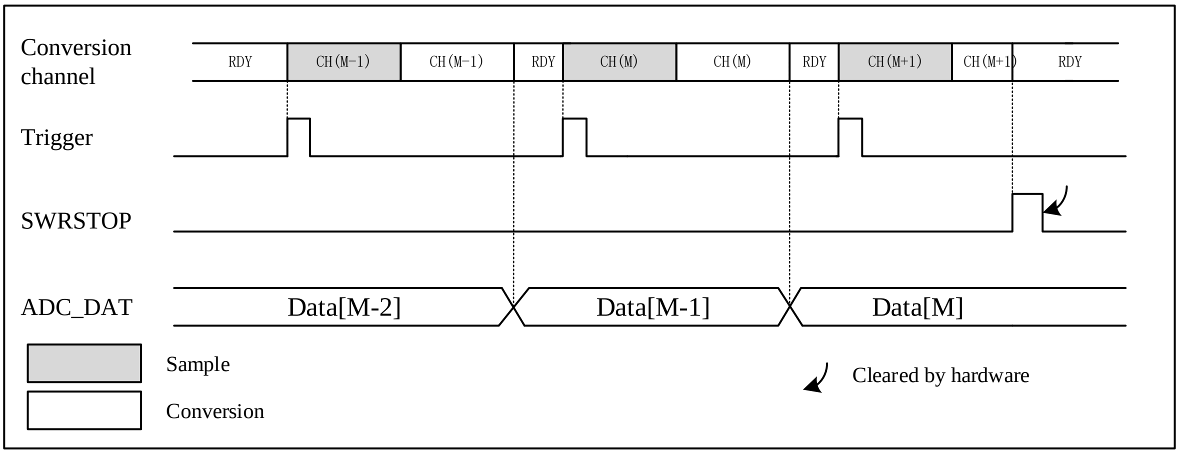
Stopping the conversion will reset the ongoing ADC operation. Subsequently, the ADC can be reconfigured (for example: changing channel selection or trigger source) to prepare for a new operation. If ADC_CTRL2.SWSTOP is set to 1, it will stop any ongoing regular conversions and discard partial conversion results (ADC_DAT register will not be updated with the current conversion result). If ADC_CTRL2.JSWSTOP is set to 1, it will stop any ongoing injected conversions and discard partial conversion results (ADC_JDATx register will not be updated with the current conversion result). The scan sequence will also be aborted and reset.

After the program execution is completed, the ADC_CTRL2.SWSTOP bit (for regular conversions) or the ADC_CTRL2.JSWSTOP bit (for injected conversions) will be cleared by hardware. Software must poll the ADC_STS.STR (or ADC_STS.JSTR) until it is reset before determining that the ADC has completely stopped running.

Note：

1. *Users can stop injected conversions while regular conversions are still ongoing, or stop regular conversions while injected conversions are still ongoing. For example, users can reconfigure automatic injection enable, trigger source, and injection channel conversion sequence while regular conversions are in progress. They can also reconfigure interval mode enable, interval length, trigger source, and regular channel conversion sequence while injected conversions are in progress.*
2. *In automatic injection mode, setting ADC_CTRL2.SWSTOP to 1 will abort both regular conversions and injected conversions.*

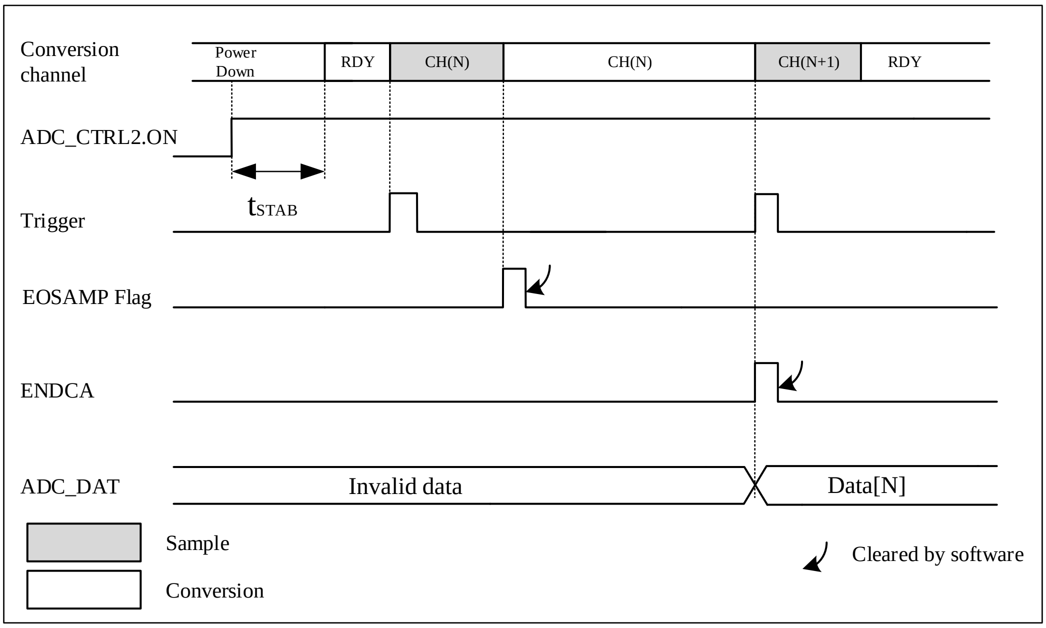
Figure 17-7 Stop Ongoing Regular Channel Conversions



17.3.13 Timing Diagram

When ADC_CTRL2.ON is first set to 1, the ADC powers on. After power-on, the ADC requires a certain time t_{STAB} to ensure its stability.

Figure 17-8 Timing Diagram



17.3.14 End of Sampling, End of Conversion Flags

Every time a regular conversion end (ENDCA) event and an injected conversion (JENDCA) event occur, the ADC notifies the application, and new data from regular channel conversions will be updated in the ADC_DAT register. The ADC immediately sets the ADC_STS.ENDCA flag to 1. If ADC_INTEN.ENDCAIEN is set to 1, an interrupt can be generated. The ADC_STS.ENDCA flag can be cleared by software by writing a 1 to it.

New data from injected channel conversions will be updated in the ADC_JDATx register. The ADC immediately sets the ADC_STS.JENDCA flag to 1. If ADC_INTEN.JENDCAIEN is set to 1, an interrupt can be generated. The ADC_STS.JENDCA flag can be cleared by software by writing a 1 to it.

The ADC also indicates the end of the sampling phase by setting the ADC_STS.EOSAMP status to 1 (only for regular conversion channels). If ADC_INTEN.EOSMPIEN is set to 1, an interrupt can be generated. The ADC_STS.EOSAMP flag can be cleared by software by writing a 1 to it.

17.3.15 Analog Watchdog

17.3.15.1 Analog Watchdog 1

Analog Watchdog 1 can be enabled on regular channels by setting ADC_CTRL1.AWDGERCH to 1.The high threshold of Analog Watchdog 1 can be set by configuring ADC_AWD1HIGH.HTH[11:0], and the low threshold of Analog Watchdog 1 can be set by ADC_AWD1LOW.LTH[11:0]. The alignment of the analog watchdog thresholds is independent of data alignment because the comparison between the ADC conversion value and the thresholds is done before alignment. When the ADC conversion value is higher than the high threshold or lower than the low threshold of the analog watchdog, the analog watchdog flag (ADC_STS.AWDG1) will be set to 1, and if ADC_INTEN.AWD1IEN is configured as 1, an interrupt will be generated. By configuring ADC_CTRL1.AWD1SGLEN and ADC_CTRL1.AWD1CH[4:0], the analog watchdog can be applied to one or multiple channels, shown inTable 17-2.

Table 17-2 Analog Watchdog Channel Selection

Channels for the analog watchdog	ADC_CTRL1 Register Control Bit		
	AWD1SGLEN	AWD1ERCH	AWD1EJCH
There is none	Any value	0	0
All injection channels	0	0	1
All regular channels	0	1	0
All injection and regular channels	0	1	1
A single injection channel	1	0	1
A single regulars of the channel	1	1	0
A single injection or regular channels	1	1	1

If the analog voltage of the ADC conversion is below the lower threshold or above the upper threshold, the status bit of Analog Watchdog 1 will be set to 1. If the resolution of the converted data is less than 12 bits (depending on the ADC_CTRL3.RES[2:0] bits), the LSB of the programmed threshold must remain cleared, and the comparison will be done internally with the complete 12-bit converted data (left-aligned to the half-word boundary).

Table 17-3 Analog Watchdog 1/2/3 comparison

Resolution	Alignment	Comparison of raw data with threshold		Annotation
		Raw Data	threshold	
12-bit	Left align	DATA[11:0]	HTH[11:0] and LTH[11:0]	-
	Right align			
10-bit	Left align	DATA[11:2] + 00	HTH[11:0] and LTH[11:0]	HTH[1:0] and LTH[1:0] must be 0
	Right align	00 + DATA[9:0]		-
8-bit	Left align	DATA[11:4] + 0000	HTH[11:0] and LTH[11:0]	HTH[3:0] and LTH[3:0] must be 0
	Right align	0000 + DATA[7:0]		-
6-bit	Left align	DATA[11:6] + 000000	HTH[11:0] and LTH[11:0]	HTH[5:0] and LTH[5:0] must be 0
	Right align	000000 + DATA[6:0]		-

17.3.15.1.1 Analog watchdog 1 filtering

The watchdog 1 filtering function is only used for watchdog on a single channel (regular channel or injected channel) (i.e., ADC_CTRL3.AWD1SGLEN = 1). If the conversion result is out of the threshold range and the number of exceedances exceeds ADC_AWD1LOW.AWDFIL[2:0], the ADC_STS.AWDG1 flag will be set, and if ADC_INTEN.AWD1IEN is set to 1, a corresponding interrupt event will also be generated.

17.3.15.2 Analog watchdog 2/3

Analog watchdogs 2 and 3 are more flexible and can protect multiple selected channels by programming the corresponding bits in ADC_AWDxEN.AWDxEN[18:0](x=2,3). Setting any bit to 1 in

ADC_AWDxEN.AWDxEN[18:0](x=2,3) enables the corresponding watchdog. If the resolution of the converted data is less than 12 bits (depending on the ADC_CTRL3.RES[2:0] bits), the LSB of the programmed threshold must remain cleared, and the comparison will be made internally against the full 12-bit converted data (left-aligned to the half-word boundary). shown in Table 17-3.

17.3.15.3 Generation of ADCy_AWDx_OUT Signal Output

Each analog watchdog is associated with an internal hardware signal ADCx_AWDy_OUT (x = ADC number, y = watchdog number), which is directly connected to the ETR input of some on-chip timers (external trigger). When the associated analog watchdog is enabled, ADCx_AWDy_OUT will be activated:

- When the protected conversion exceeds the programmed threshold, ADCx_AWDy_OUT will be set to 1.
- After the next protected conversion within the programmed threshold range is completed, ADCx_AWDy_OUT will be reset (if the next protected conversion still exceeds the programmed threshold range, this bit will remain set to 1).
- When the ADC is disabled (by setting ADC_CTRL2.ON to 0), ADCx_AWDy_OUT remains in the reset state. Please note that stopping regular channel conversions or injected channel conversions (by setting ADC_CTRL2.SWRSTOP or ADC_CTRL2.SWJSTOP to 1) has no effect on the generation of ADCy_AWDx_OUT.

Note: The analog watchdog flag has no effect on the generation of ADCx_AWDy_OUT (e.g., if the software does not clear the analog watchdog flag, i.e., the analog watchdog flag remains at 1, ADCy_AWDy_OUT can still toggle).

Figure 17-9 Generation of ADCx_AWDy_OUT (on a single regular channel)

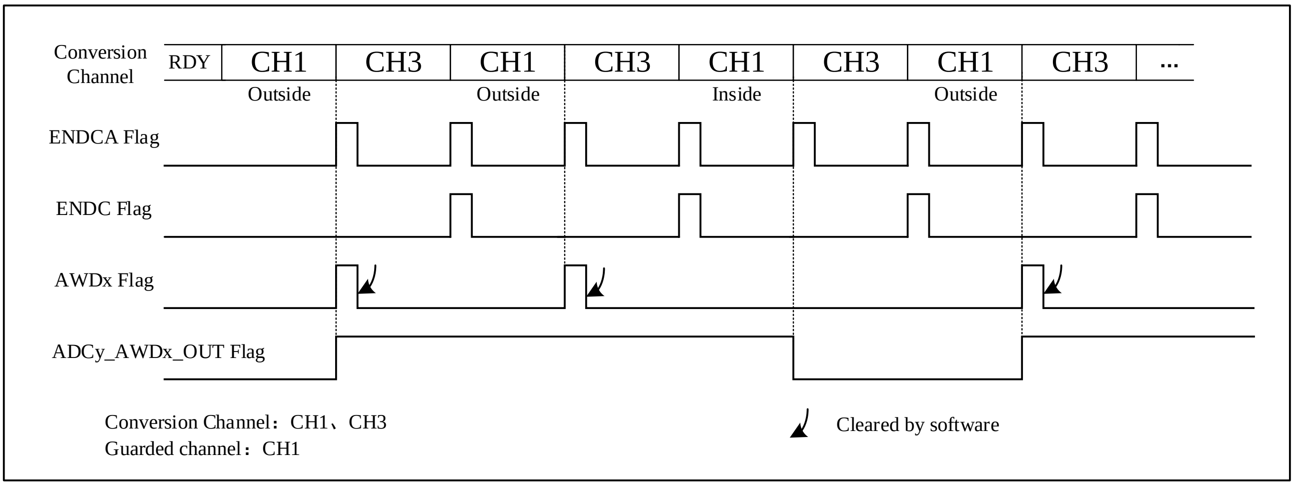


Figure 17-10 Generation of ADCx_AWDy_OUT (on multiple regular channels)

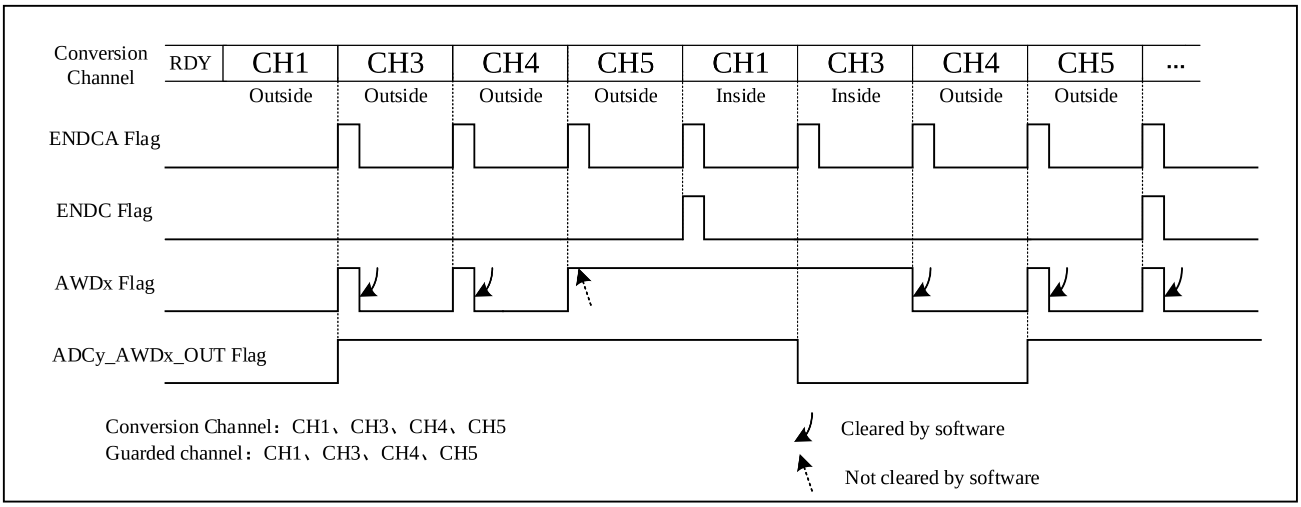
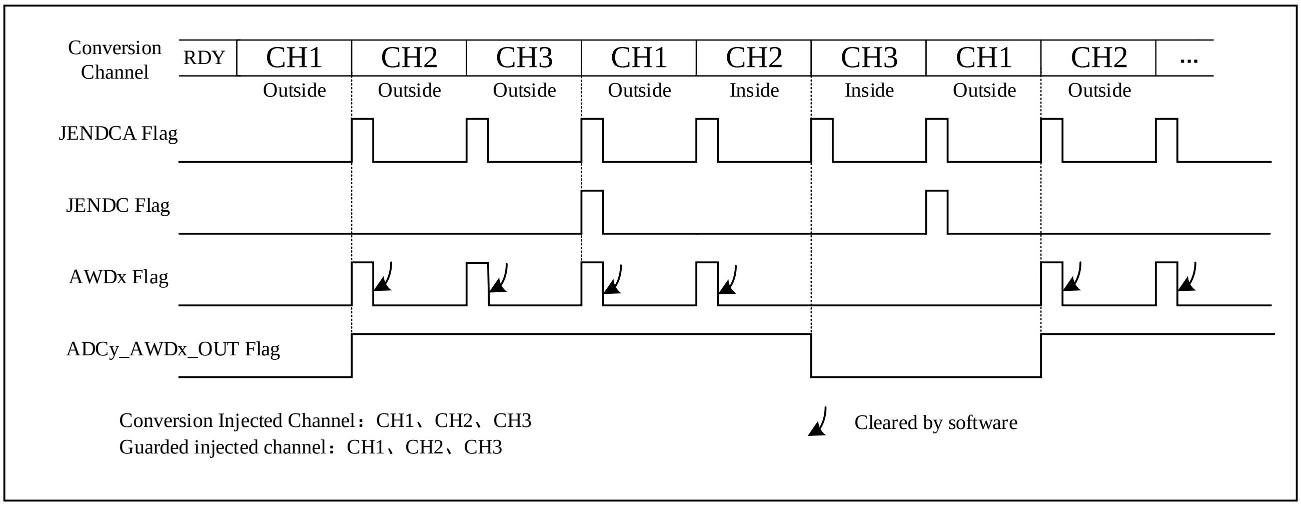


Figure 17-11 Generation of ADCx_AWDy_OUT (on multiple injected channels)



17.3.15.4 Watchdog with Gain Compensation and Offset Compensation

When gain and offset compensation are enabled, the analog watchdog compares the threshold with the compensated data.

Note: When offset compensation is enabled (ADC_OFFSETx.OFFSCHxEN = 1), the watchdog comparison results may be incorrect due to potential data overflow or underflow after offset compensation. The watchdog results are accurate only when saturation function is enabled (ADC_OFFSETx.OFFSCHxSATEN = 1). Therefore, when offset compensation is enabled, users should try to avoid the compensated result being unsigned data to prevent incorrect watchdog comparison results

17.3.16 Gain Compensation

ADC conversion supports gain compensation. When the user enables gain compensation (ADC_CTRL2.GCOMPEN = 1), the calculation is performed after each data conversion according to the following formula：

OUT (Final result) = [DATA(Conversion result) * ADC_GCOMP.GCOMPDAT[13:0]] / 4096

Since the gain compensation coefficient ADC_GCOMP.GCOMPDAT[13:0] is configured in the range of 0 to 16383, the actual range of multiples for gain compensation is 0 to 3.9997556. The gain compensation will round the calculated data and store it in ADC_DAT or ADC_JDATx.

Gain compensation is also effective when oversampling is enabled. When oversampling is enabled, the ADC will first perform the oversampling process (including accumulation and right-shifting), and then apply gain compensation to the final oversampled result.

17.3.17 Offset Compensation

By setting the bit ADC_OFFSETx.OFFSCHxEN to 1, an offset can be applied to channel x.

Users can select the channel for offset by configuring the ADC_OFFSETx.OFFSCHxCH[4:0] register; choose the offset value by configuring ADC_OFFSETx.OFFSCHxDAT[11:0]; and select either positive or negative offset by configuring ADC_OFFSETx.OFFSCHxDIR. You can configure ADC_OFFSETx.OFFSCHxSATEN to choose whether to enable the saturation function. For example, when the user selects a certain channel Y for negative offset, the value obtained after the conversion of channel Y will be subtracted by the user-defined offset value, and the data will be stored in the result register. As long as offset compensation is enabled (ADC_OFFSETx.OFFSCHxEN = 1), the read data is signed, and the SEXT bit indicates the extended sign value.

When saturation function is enabled, taking 12-bit resolution as an example, with ADC_OFFSETx.OFFSCHxSATEN=1, the compensated data is unsigned, ranging from a minimum of 0x000 to a maximum of 0xFFF.

Note：

1, Oversampling does not support offset compensation. When oversampling is enabled, the ADC_OFFSETx.OFFSCHxEN bit will be automatically ignored.

2, If both gain compensation and offset compensation are enabled, gain compensation is performed first, followed by offset compensation.

Table 17-4 Processing Data With Different Resolutions

Resolution	Alignment	Subtracting raw data from offset data		Offset result	Annotation
		Raw data	Offset data		
12-bit	Left align	DATA[11:0]	OFFSCHxDAT [11:0]	Signed 12-bit data	-
	Right align				
10-bit	Left align	DATA[11:2]+ 00	OFFSCHxDAT [11:0]	Signed 10-bit data	OFFSCHxDAT [1:0] must be 0
	Right align	00 + DATA[9:0]			-
8-bit	Left align	DATA[11:4] + 0000	OFFSCHxDAT [11:0]	Signed 8-bit data	OFFSCHxDAT [3:0] must be 0
	Right align	0000 + DATA[7:0]			-
6-bit	Left align	DATA[11:6] + 000000	OFFSCHxDAT [11:0]	Signed 6-bit data	OFFSCHxDAT [5:0] must be 0
	Right align	000000+ DATA[6:0]			-

17.3.18 FIFO Function

The ADC can operate in FIFO mode, which can be enabled by setting ADC_FIFO.EN to enable FIFO mode.

When the ADC conversion data is completed, it can be stored in the ADC's FIFO, which has a size of 12*32 bits. FIFO mode is often used in conjunction with DMA burst transfers, for more details please refer to the DMA section.

When the amount of data in the ADC FIFO reaches the threshold programmed in the ADC_FIFOCFG.WL[3:0] field, the ADC_FIFOSTS.HFFLAG will be set. If the user has enabled the half-full interrupt (ADC_FIFOCFG.HFINTEN=1), an interrupt will be generated. The data from the FIFO can be read from the ADC_DAT register, and ADC_FIFOSTS.DATCNT[4:0] will indicate the current number of remaining valid data in the FIFO.

To increase the flexibility of data control, the design includes ADC_FIFO empty, ADC_FIFO half-full, ADC_FIFO full, and ADC_FIFO not empty flags, and supports users to query the current number of valid data in the FIFO with ADC_FIFOSTS.DATCNT, to meet the needs of more compatible scenarios.

Note: In multi-ADC mode, if the FIFO function is enabled, it is recommended to actively clear the cached data in the FIFO after each multi-ADC mode switch (by setting ADC_FIFOCFG.CLR=1) to avoid reading invalid data from the previous mode in the new mode.

17.4 Data Aligned

here are two alignment methods for data memory after conversion: left-aligned and right-aligned. The alignment can be set by the ADC_CTRL2.ALIG bit. ADC_CTRL2.ALIG = 0 is right-aligned, ADC_CTRL2.ALIG = 1 is left-aligned.

Due to gain compensation, the data width may increase by 2 bits, and after offset compensation, the data width may increase by 1 bit. To prevent loss of effective bits, the data format will add an additional 3 bits on top of the original resolution. The following is the specific data format.

Figure 17-12 Right-Aligned Data

12-bit： OFFSCHxEN = 1

Bit15								Bit7							Bit0
SEXT	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

12-bit： OFFSCHxEN = 0

Bit15								Bit7							Bit0
0	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

10-bit： OFFSCHxEN = 1

Bit15								Bit7							Bit0
SEXT	SEXT	SEXT	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

10-bit： OFFSCHxEN = 0

Bit15								Bit7							Bit0
0	0	0	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

8-bit： OFFSCHxEN = 1

Bit15								Bit7							Bit0
SEXT	SEXT	SEXT	SEXT	SEXT	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

8-bit： OFFSCHxEN = 0

Bit15								Bit7							Bit0
0	0	0	0	0	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

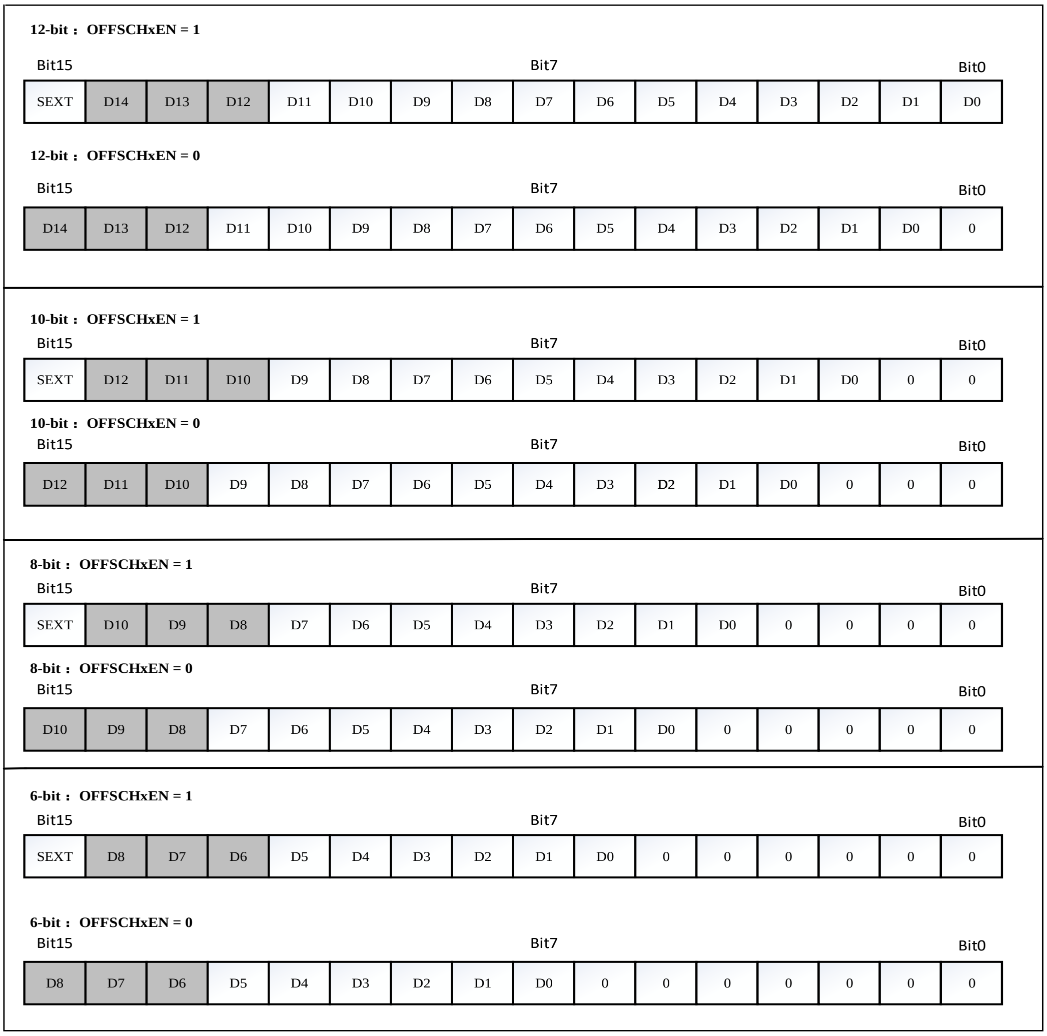
6-bit： OFFSCHxEN = 1

Bit15								Bit7							Bit0
SEXT	SEXT	SEXT	SEXT	SEXT	SEXT	SEXT	D8	D7	D6	D5	D4	D3	D2	D1	D0

6-bit： OFFSCHxEN = 0

Bit15								Bit7							Bit0
0	0	0	0	0	0	0	D8	D7	D6	D5	D4	D3	D2	D1	D0

Figure 17-13 Left-Aligned Data



12-bit：OFFSCHxEN = 1

Bit15								Bit7							Bit0
SEXT	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

12-bit：OFFSCHxEN = 0

Bit15								Bit7							Bit0
D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	0

10-bit：OFFSCHxEN = 1

Bit15								Bit7							Bit0
SEXT	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	0	0

10-bit：OFFSCHxEN = 0

Bit15								Bit7							Bit0
D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	0	0	0

8-bit：OFFSCHxEN = 1

Bit15								Bit7							Bit0
SEXT	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	0	0	0	0

8-bit：OFFSCHxEN = 0

Bit15								Bit7							Bit0
D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	0	0	0	0	0

6-bit：OFFSCHxEN = 1

Bit15								Bit7							Bit0
SEXT	D8	D7	D6	D5	D4	D3	D2	D1	D0	0	0	0	0	0	0

6-bit：OFFSCHxEN = 0

Bit15								Bit7							Bit0
D8	D7	D6	D5	D4	D3	D2	D1	D0	0	0	0	0	0	0	0

17.5 Programmable Channel Sampling Time

The ADC samples the input voltage over a certain number of ADC_CLK cycles, and the number of sampling periods can be changed through ADC_SAMPTx.SAMPx[3:0]. For different channels, the sampling intervals can be independently programmed. The total conversion time is calculated as follows:

$$T_{CONV} = \text{Sampling time} + 12.5 \text{ cycle}$$

Sampling time = ADC_SAMPTx.SAMPx[3:0]；

Note: Before starting the conversion, ensure that the configured sampling time is sufficient for the input voltage source to charge the ADC sampling capacitor to the input voltage level.

Example：

ADCCLK=30MHz，the sampling time 2.5 cycle and resolution is 12bit，the total conversion time is “2.5 +12.5”，ADC_CLK cycles，that is：

$$T_{CONV} = 2.5 + 12.5 = 15 \text{ cycle} = 0.5\mu s$$

17.6 External Trigger and Trigger Polarity Selection

17.6.1 External Polarity Selection

For regular sequences, when ADC_CTRL2.EXTRSEL[5:0] is not equal to SWSTRRCH, the regular channel can start the conversion triggered by the edge of an external event, where the edge polarity of the external event can be configured. Users can select the triggering polarity as rising edge (2b'01), falling edge (2b'10), or both rising and falling edges (2b'11) through ADC_CTRL2.EXTPRSEL[1:0].

For the injection sequence, when ADC_CTRL2.EXTJSEL[5:0] is not equal to SWSTRJCH, the injection channel can start the conversion triggered by the edge of an external event, where the edge polarity of the external event can be configured. Users can select the triggering polarity as rising edge (2b'01), falling edge (2b'10), or both rising and falling edges (2b'11) through ADC_CTRL2.EXTPJSEL[1:0].

Table 17-5 Configure polarity for external triggering of regular or injection channels

EXTPRSEL[1:0] / EXTPJSEL[1:0]	Valid edge
00	Disable hardware trigger detection, enable software trigger detection
01	Perform hardware trigger detection on the rising edge
10	Perform hardware trigger detection on the falling edge
11	Perform hardware trigger detection on both the rising and falling edges

17.6.2 External Trigger Source

Table 17-6 Selection of External Trigger Source for Regular/Injection Channels

EXTRSEL[5:0]	Trigger Source	Type
000000	ATIM1_CC4 Event	Internal signal from on-chip timer
000001	ATIM1_TRGO Event	
000010	ATIM1_TRGO2 Event	
000011	ATIM2_CC4 Event	
000100	ATIM2_TRGO Event	
000101	ATIM2_TRGO2 Event	
000110	ATIM3_CC4 Event	
000111	ATIM3_TRGO Event	
001000	ATIM3_TRGO2 Event	
001001	GTIM1_ CC4 Event	
001010	GTIM2_ CC4 Event	
001011	GTIM3_ CC4 Event	
001100	GTIM4_ CC4 Event	
001101	GTIM5_ CC4 Event	
001110	GTIM6_ CC4 Event	
001111	GTIM7_ CC4 Event	
010000	GTIM8_ CC4 Event	

EXTRSEL[5:0]	Trigger Source	Type
010001	GTIM9_ CC4 Event	
010010	GTIM10_ CC4 Event	
010011	GTIM1_TRGO Event	
010100	GTIM2_TRGO Event	
010101	GTIM3_TRGO Event	
010110	GTIM4_TRGO Event	
010111	GTIM5_ TRGO Event	
011000	GTIM6_TRGO Event	
011001	GTIM7_TRGO Event	
011010	GTIM8_TRGO Event	
011011	GTIM9_TRGO Event	
011100	GTIM10_TRGO Event	
011101	shrtim1_adc_trg1 Event	
011110	shrtim1_adc_trg2 Event	
011111	shrtim1_adc_trg3 Event	
100000	shrtim1_adc_trg4 Event	
100001	shrtim1_adc_trg5 Event	
100010	shrtim1_adc_trg6 Event	
100011	shrtim1_adc_trg7 Event	
100100	shrtim1_adc_trg8 Event	
100101	shrtim1_adc_trg9 Event	
100110	shrtim1_adc_trg10 Event	
100111	LPTIM1_OUT Event	
101000	LPTIM2_OUT Event	
101001	EXTI line 0~15 Event	External Pin
101010	SWSTRRCH/ SWSTRJCH	Software Control Bit

17.7 Oversampling

The oversampling unit performs data preprocessing to reduce the burden of CPU data processing. It can handle multiple conversions and average them into a single data with increased data width. Oversampling can increase the resolution support up to 16 bits.

It provides results in the following form, where N and M can be adjusted:

$$Result = \frac{1}{M} \sum_{n=0}^{n=N-1} \mathrm{Conversion}(t_n)$$

The oversampling ratio N is defined using the ADC_CTRL3.OSR[3:0] bits, which can range from 2x to 256x. The division factor M consists of a right shift of up to 8 bits, defined using the ADC_CTRL3.OSS[3:0] bits.

The summation unit can generate results up to 20 bits (256*0xFFF result), which are first right-shifted. It is then truncated to the lowest significant 16 bits, with the minimum difference between the least significant bits caused by the bit shift rounded to the nearest value. Finally, it is transferred to the ADC_DAT data register. Please refer to the diagram below:

Note: If the intermediate result after shifting exceeds 16 bits, the result will be truncated without saturation processing.

Figure 17-14 Diagram of truncating 20-bit data to 16 bits

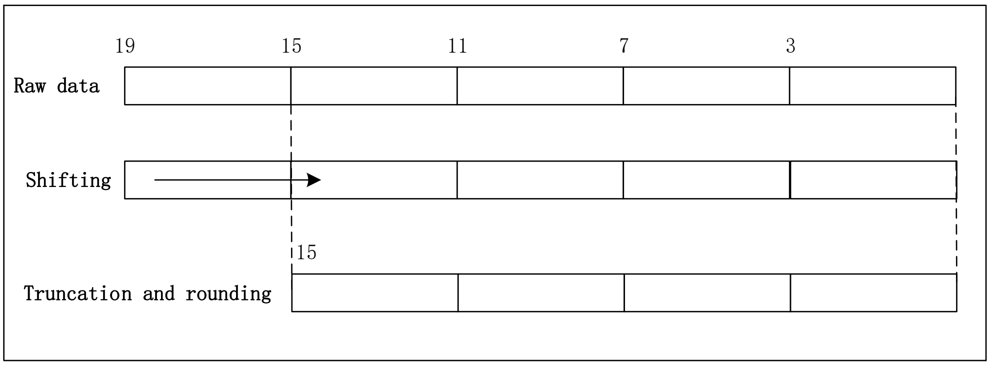


Figure 17-15 Diagram of data processing after a 5-bit right shift and rounding

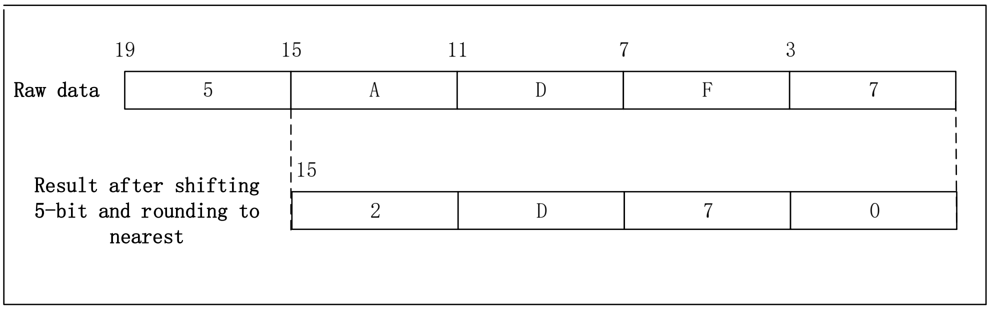


Table 17-7 Maximum output result after oversampling processing

OSR[3:0]	Oversampled source data	No right shift OSS[3:0] =0000	1-bit right shift OSS[3:0] =0001	2-bit right shift OSS[3:0] =0010	3-bit right shift OSS[3:0] =0011	4-bit right shift OSS[3:0] =0100	5-bit right shift OSS[3:0] =0101	6-bit right shift OSS[3:0] =0110	7-bit right shift OSS[3:0] =0111	8-bit right shift OSS[3:0] =1000
0000	0x0FFF	0x0FFF	0x0800	0x0400	0x0200	0x0100	0x0080	0x0040	0x0020	0x0010
0001	0x1FFE	0x1FFE	0x0FFF	0x0800	0x0400	0x0200	0x0100	0x0080	0x0040	0x0020
0010	0x3FFC	0x3FFC	0x1FFE	0x0FFF	0x0800	0x0400	0x0200	0x0100	0x0080	0x0040
0011	0x7FF8	0x7FF8	0x3FFC	0x1FFE	0x0FFF	0x0800	0x0400	0x0200	0x0100	0x0080
0100	0xFFF0	0xFFF0	0x7FF8	0x3FFC	0x1FFE	0x0FFF	0x0800	0x0400	0x0200	0x0100
0101	0x1FFE0	0xFFE0	0xFFF0	0x7FF8	0x3FFC	0x1FFE	0x0FFF	0x0800	0x0400	0x0200
0110	0x3FFC0	0xFFC0	0xFFE0	0xFFF0	0x7FF8	0x3FFC	0x1FFE	0x0FFF	0x0800	0x0400

0111	0x7FF80	0xFF80	0xFFC0	0xFFE0	0xFFF0	0x7FF8	0x3FFC	0x1FFE	0x0FFF	0x0800
1000	0xFFF00	0xFF00	0xFF80	0xFFC0	0xFFE0	0xFFF0	0x7FF8	0x3FFC	0x1FFE	0x0FFF

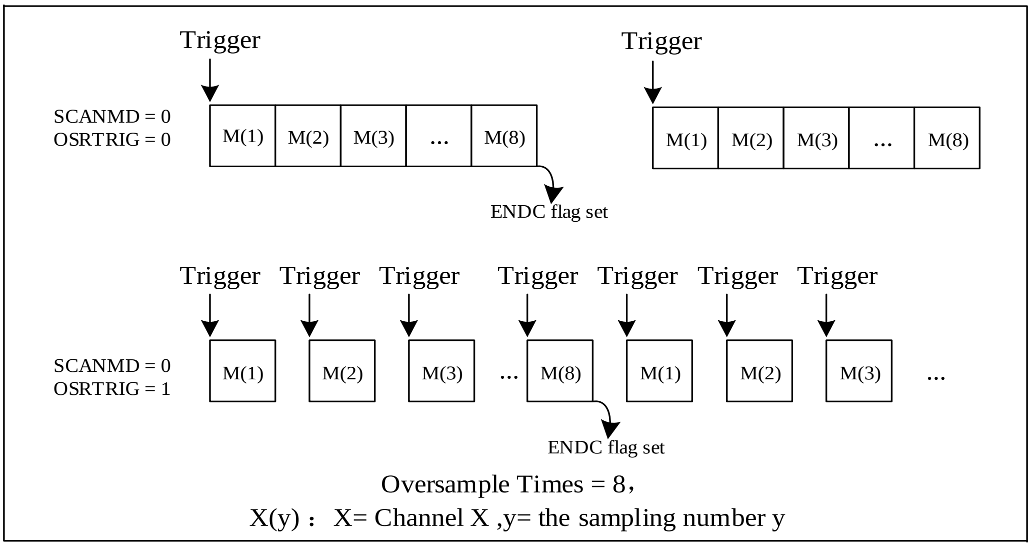
Note: Gray data cells indicate that the data has been truncated;

17.7.1 Trigger Mode

The Trigger mode for regular channels supports continuous triggering (ADC_CTRL3.OSRTRIG=0) and single triggering (ADC_CTRL3.OSRTRIG=1):

In continuous mode, the regular channel completes continuous N samples after receiving one trigger; in single trigger mode, the regular channel completes only one sample after receiving one trigger, as shown in the diagram below:

Figure 17-16 External Trigger Regular Oversampling Mode



17.7.2 Regular Channel Oversampling

The Regular Oversampling Mode bit ADC_CTRL3.OSRMD defines how the normal oversampling sequence will recover if an injection conversion interrupts the normal oversampling sequence:

In continuous mode (ADC_CTRL3.OSRMD=0), the accumulation restarts from the last valid data. This ensures that oversampling is completed regardless of the injection frequency (as long as at least one regular conversion can be completed between triggers);

In reset mode (ADC_CTRL3.OSRMD=1), the accumulation restarts from 0 (previous conversion results are ignored). This mode allows ensuring that all data used for oversampling is continuously converted within a single time period. It is important to ensure that the injection trigger period exceeds the oversampling period length; if this condition is not met, oversampling cannot be completed, and the normal sequencer will be blocked.

Figure 17-17 Regular Oversampling Continuous Mode

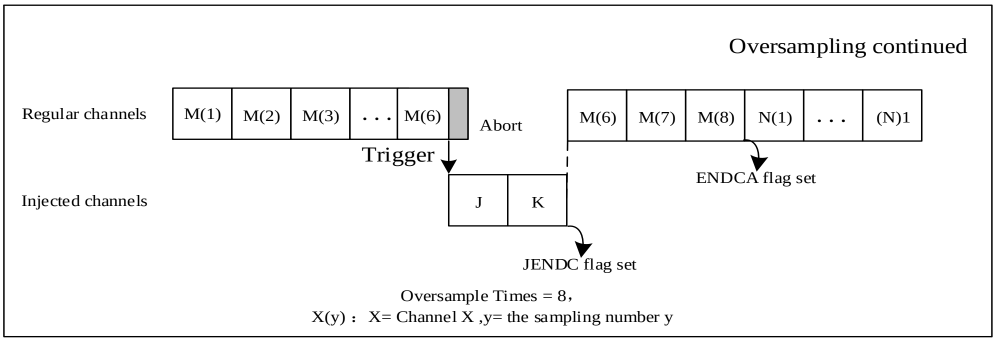
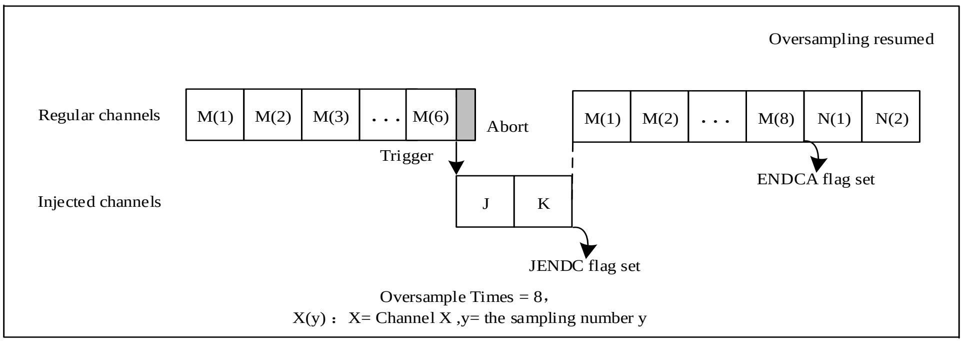


Figure 17-18 Regular Oversampling Reset Mode



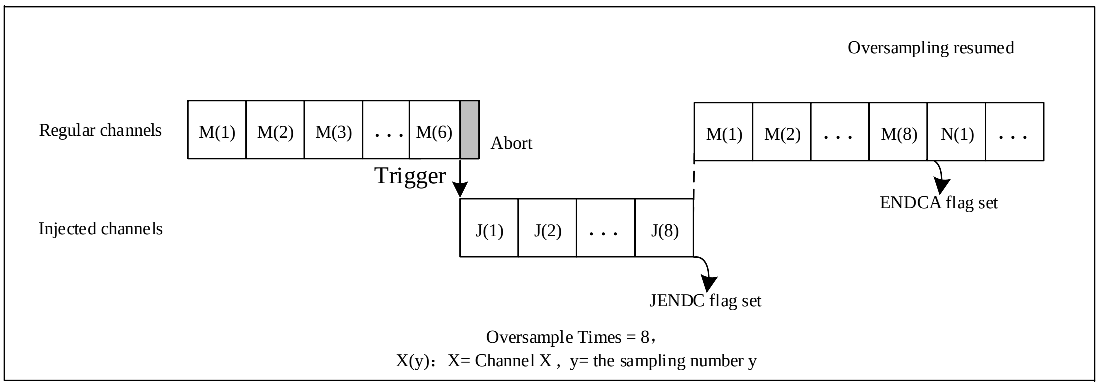
17.7.3 Injection Channel Oversampling

The Injection Oversampling Mode bit ADC_CTRL3.OSJE=1 only allows oversampling of conversions in the injection channel sequence.

17.7.4 Regular and Injection Channel Oversampling

When enabling both ADC_CTRL3.OSJE and ADC_CTRL3.OSRE at the same time, it means that oversampling is opened for both the regular and injection channels. In this case, the oversampling mode for the regular channel can still be configured as reset mode or continuous mode. As shown in the diagram below:

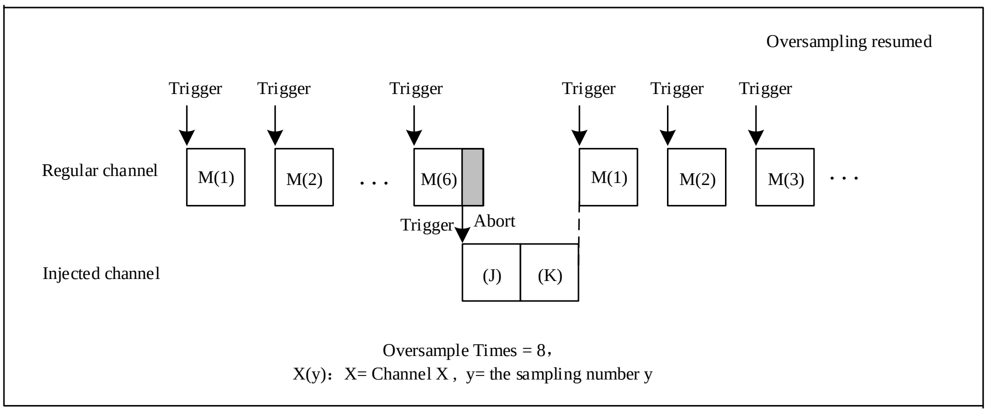
Figure 17-19 Regular and Injection Oversampling



17.7.5 Triggering Regular Channel Oversampling with Injection Conversions

In triggered regular mode (ADC_CTRL3.OSRTRIG=1), it can be used together with injection conversions. The diagram is shown below:

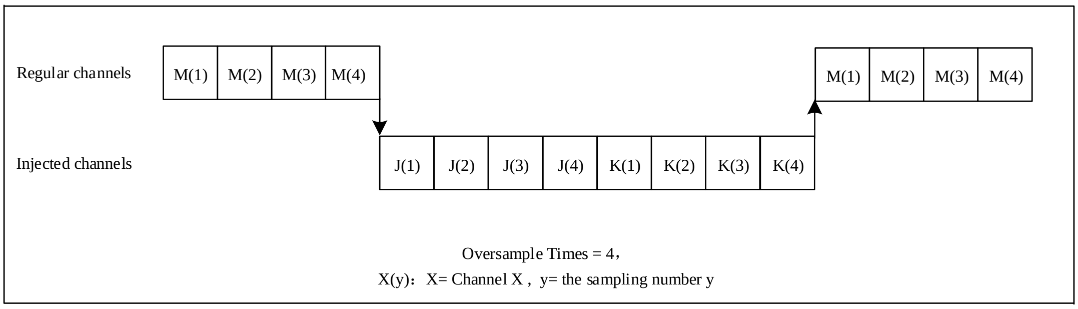
Figure 17-20 Triggered Regular Channel Oversampling with Injection Conversions



17.7.6 Automatic Injection Mode

Automatic injection sequences can also be oversampled, this mode requires enabling oversampling for both regular and injection channels, i.e., ADC_CTRL3.OSJE=1, ADC_CTRL3.OSRE=1, ADC_CTRL1.AUTOJC=1. In automatic injection mode, the oversampling mode for the regular channel is ignored. When oversampling is performed in automatic injection mode, trigger mode can also be enabled, i.e., ensuring ADC_CTRL3.OSJE=1, ADC_CTRL3.OSRE=1, ADC_CTRL1.AUTOJC=1, ADC_CTRL3.OSRTRIG=1. The diagram below shows the sequence of conversions;

Figure 17-21 Oversampling in Automatic Injection Mode



17.7.7 Multi-ADC Mode

When enabling Dual ADC or Triple ADC, oversampling can also be activated, requiring identical configurations for each ADC.

17.8 DMA Requests

To avoid data loss in the ADC_DAT register due to excessive data from multiple regular channel conversions, you can set the ADC_CTRL2.DMAMD[1:0] bits to 2b01 to use DMA. When the ADC regular channel conversion is complete, a DMA request is generated. Upon receiving the request, DMA transfers the converted data from the ADC_DAT register to the user-specified destination address.

Note: In Independent ADC mode, ADC1, ADC2, ADC3, and ADC4 have independent DMA functionality.

17.9 Multi-ADC Mode

ADC1 (Master) and ADC2 (Slave), ADC3 (Master) and ADC4 (Slave) can form a Dual ADC mode;

ADC1 (Master), ADC2 (Slave), ADC3 (Slave) can form a Triple ADC mode.

The ADC mode can be configured as independent mode, dual ADC mode or triple ADC mode, depending on the mode selected by the ADC_CTRL1.DUSEL[3:0] register. The ADC mode can be configured as the follow:

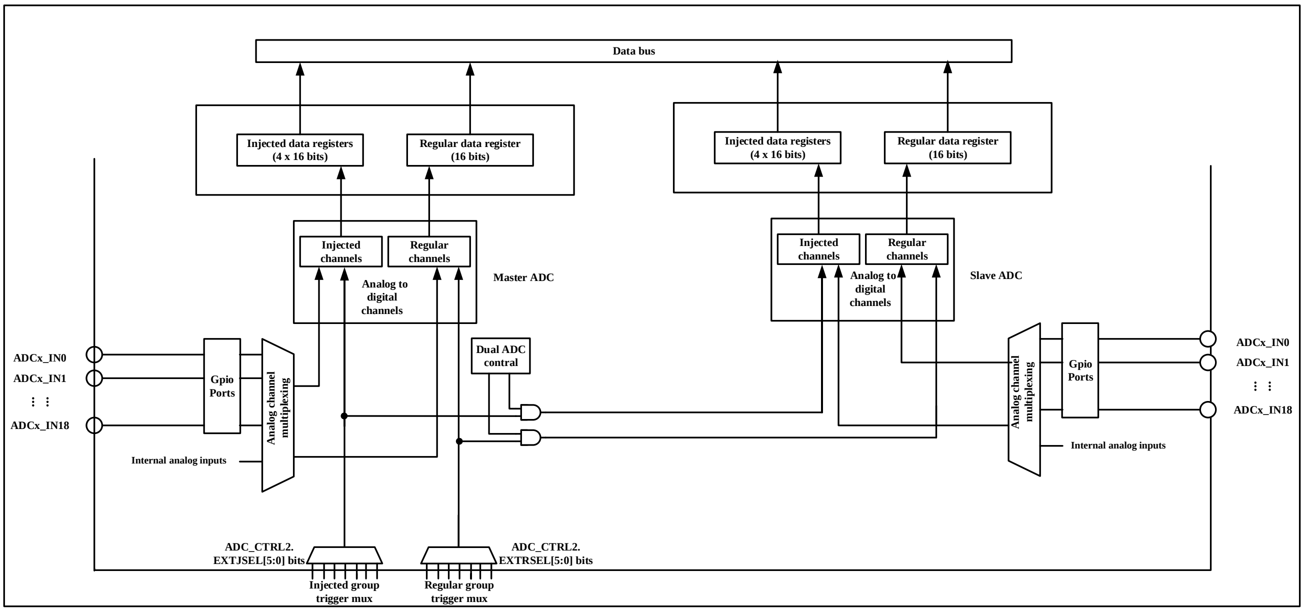
- Independent mode.
- Synchronous injection mode.
- Synchronous regular mode.
- Interleaved mode.
- Alternate trigger mode.
- Combined Synchronous regular mode + Synchronous injected mode.
- Combined Synchronous regular mode + Alternate trigger mode.
- Combined Synchronous injected mode + Interleaved mode.

Notes：

1. When configuring Dual ADC mode, users only need to configure the master ADC, including external trigger polarity selection (ADC_CTRL2.EXTPJSEL, ADC_CTRL2.EXTPRSEL), external trigger selection (ADC_CTRL2.EXTJSEL, ADC_CTRL2.EXTRSEL), and the related configurations of the slave ADCs will be ignored.

2. In Dual ADC mode operation, the conversion data from the slave ADCs can be read through the data register of the master ADC.

Figure 17-22 Dual ADC Block Diagram


Data bus
Injected data registers (4 x 16 bits)
Regular data register (16 bits)
Injected data registers (4 x 16 bits)
Regular data register (16 bits)
Injected channels
Regular channels
Analog to digital channels
Master ADC
Injected channels
Regular channels
Analog to digital channels
Slave ADC
Dual ADC contral
ADCx_IN0
ADCx_IN1
ADCx_IN18
Gpio Ports
Analog channel multiplexing
Internal analog inputs
ADCx_IN0
ADCx_IN1
ADCx_IN18
Gpio Ports
Analog channel multiplexing
Internal analog inputs
ADC_CTRL2. EXTJSEL[5:0] bits
ADC_CTRL2. EXTRSEL[5:0] bits
Injected group trigger mux
Regular group trigger mux

17.9.1 Independent Mode

In this mode, each ADC works independently.

17.9.2 Synchronous Regular Mode

In this mode, a regular sequence is converted. The external trigger comes from the multiplexer of the master ADC, determined by ADC_CTRL2.EXTRSEL[5:0], and the slave ADCs will be synchronously triggered.

In this mode, independent injection conversions are also supported. The trigger request from the injection channel will interrupt the ongoing synchronous regular conversion. After the injection conversion is completed, the regular synchronous conversion will resume.

Notes:

1. Do not convert the same channel on multiple ADCs (sampling time on the same channel for two ADCs must not overlap).

2. In synchronous regular mode, the regular sequences of the master ADC and the slave ADC need to be set for the same duration, or the interval between trigger signals should be longer than the longer conversion time of the sequences. If the interval between trigger signals is shorter than the longer conversion time of the sequences, the longer sequence may restart conversion before the shorter sequence is completed.

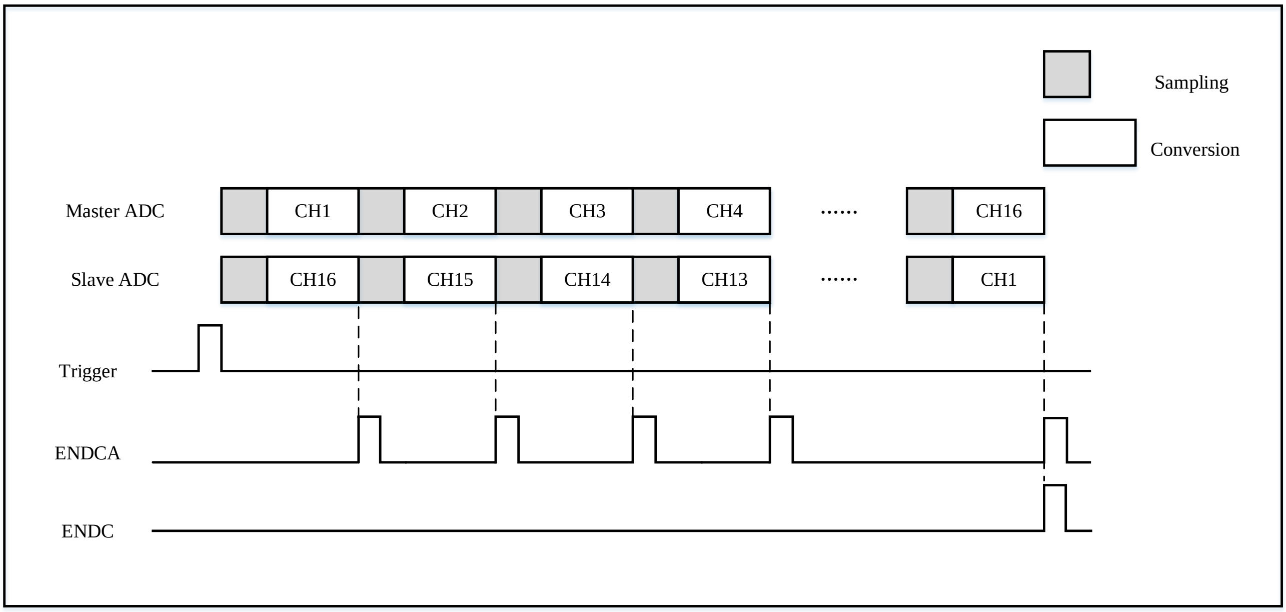
17.9.2.1 Dual ADC Mode

When the conversion event of the master ADC or slave ADC (ADC_STS.ENDCA) ends:

1. It will generate a 32-bit DMA transfer request (if ADC_CTRL2.DMAMD[1:0] bits are equal to 2b'10, and both the master and slave ADCs have enabled DMA requests). The upper half-word of ADC_DAT contains the conversion data from the slave ADC, while the lower half-word contains the conversion data from the master ADC. When DMA is enabled, the 32-bit DMA can be useds to transfer the data from ADC_DAT to SRAM.

2. If the master ADC or slave ADC sets the ADC_INTEN.ENDCAIEN, an ENDCA interrupt will be generated when the regular channel conversion of the ADC is completed, and the converted data will be stored in the ADC_DAT register.

Figure 17-23 Synchronous regular mode with 16 channels.



17.9.2.2 Triple ADC Mode

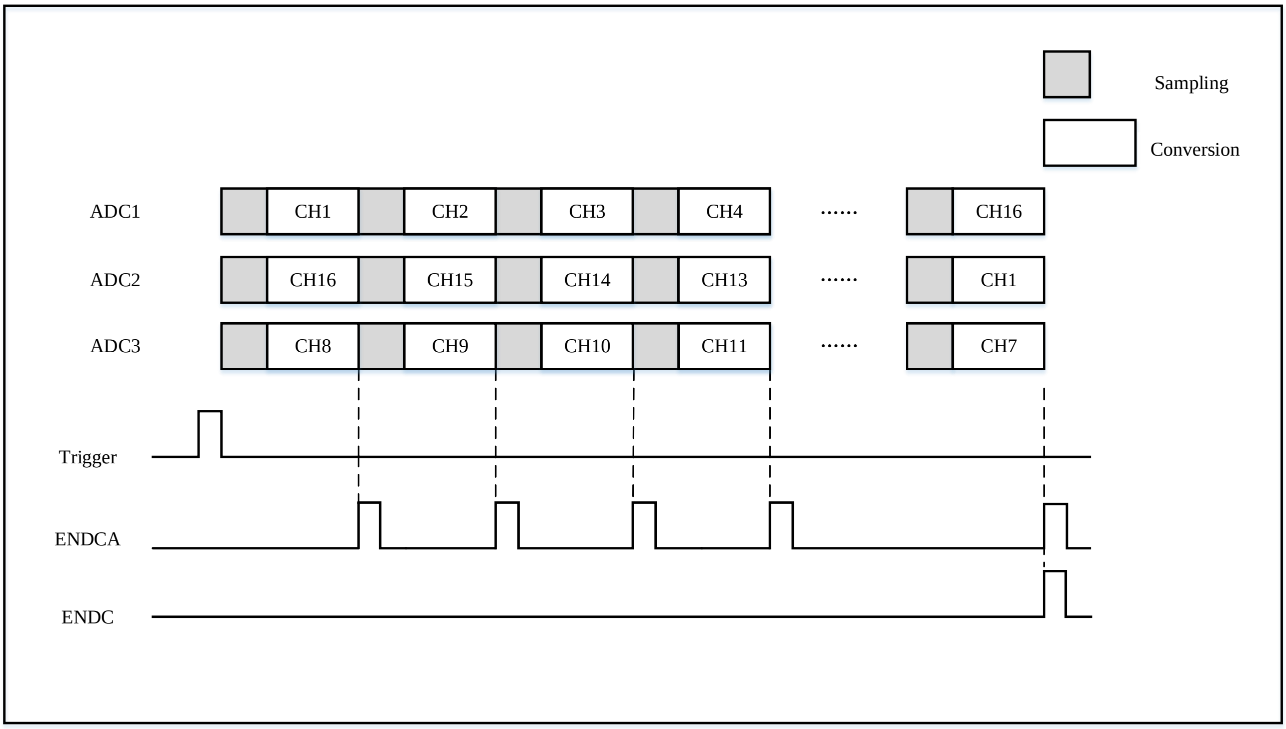
When the conversion event of ADC1, ADC2, or ADC3 ends (ADC_STS.ENDCA):

1. Three 32-bit DMA transfer requests will be generated (if ADC_CTRL2.DMAMD[1:0] bits are equal to 2b’01, and both the master and slave ADCs have enabled DMA requests). Subsequently, there will be three transfers from the ADC_DAT 32-bit register to SRAM: first, the conversion data from ADC1 will be transferred, followed by the conversion data from ADC2, and finally the conversion data from ADC3. This process will be repeated each time a new three-stage conversion occurs.

2. A 32-bit DMA transfer request will be generated (if ADC_CTRL2.DMAMD[1:0] bits are equal to 2b’10, and the master ADC has enabled DMA requests). Subsequently, there will be a transfer from the ADC_DAT 32-bit register to SRAM: in the first transfer, the upper half-word of ADC_DAT contains the conversion data from ADC2, and the lower half-word contains the conversion data from ADC1; in the second transfer, the upper half-word of ADC_DAT contains the conversion data from ADC1, and the lower half-word contains the conversion data from ADC3; in the third transfer, the upper half-word of ADC_DAT contains the conversion data from ADC3, and the lower half-word contains the conversion data from ADC3. This process will be repeated each time a new three-stage conversion occurs.

3. If ADC1/ADC2/ADC3 sets the ADC_INTEN.ENDCAIEN, an ENDCA interrupt will be generated when the regular channel conversion of the ADC is completed, and the converted data will be stored in the ADC_DAT register.

Figure 17-24 Synchronous regular mode with 16 channels



17.9.3 Synchronous Injection Mode

In this mode, converting an injection sequence, the external trigger is sourced from the multi-channel switch of the master ADC, determined by ADC_CTRL2.EXTJSEL[5:0], and the slave ADC will be synchronously triggered.

17.9.3.1 Dual ADC Mode

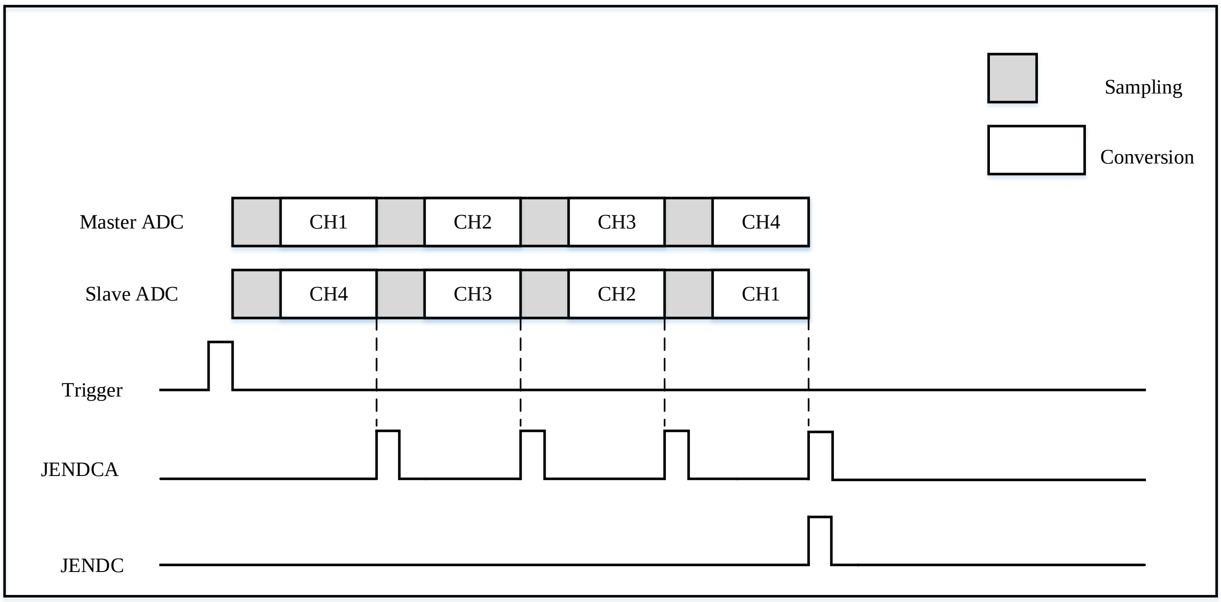
If the master ADC or slave ADC sets the ADC_INTEN.JENDCIEN, an JENDC interrupt will be generated when the injection sequence conversion of the master ADC or slave ADC is completed, and the converted data will be stored in their respective ADC_JDATx registers.

Notes :

(1) Do not convert the same channel on two ADCs (the sampling time of the two ADCs on the same channel must not overlap).

(2) In synchronous injection mode, the injection sequences for simultaneous conversion by the master ADC and slave ADC need to be set to the same time or have a trigger signal interval greater than the longer conversion time sequence. If the trigger signal interval is shorter than the longer conversion time sequence, the shorter sequence may restart conversion before the longer sequence is completed.

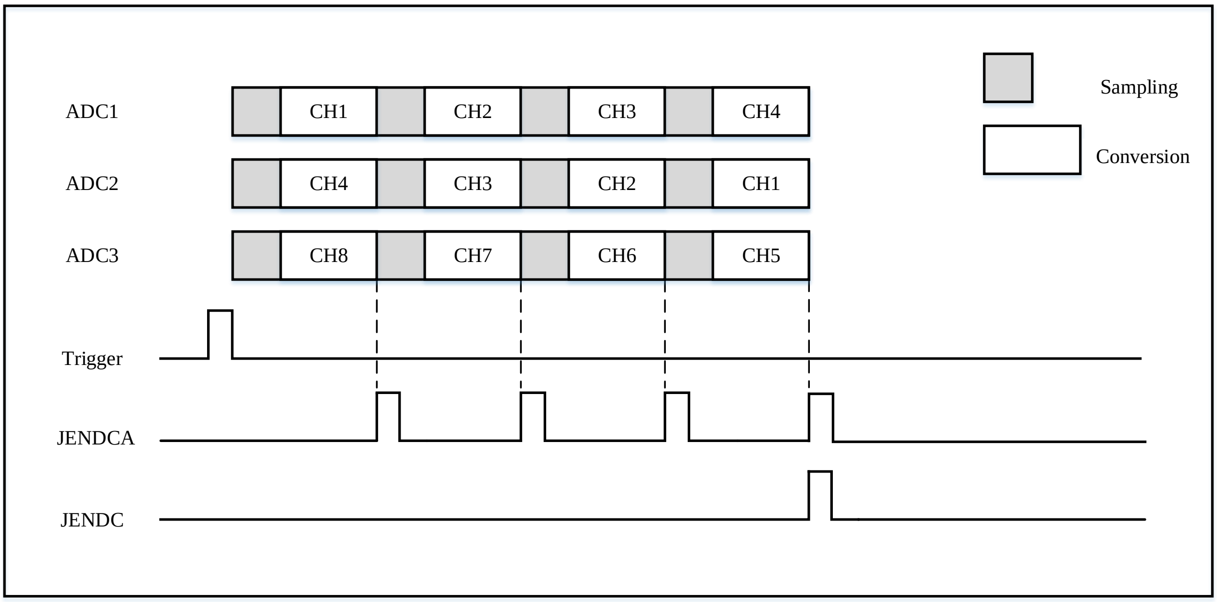
Figure 17-25 Synchronous Injection Mode with 4 channels for dual ADCs



17.9.3.2 Triple ADC Mode

If ADC1, ADC2, or ADC3 sets the ADC_INTEN.JENDCIEN, an JENDC interrupt will be generated when the injection sequence conversion of ADC1, ADC2, or ADC3 is completed, and the converted data will be stored in their respective ADC_JDATx registers.

Figure 17-26 Synchronous injection mode with 4 channels for three ADCs.



17.9.4 Interleaved Mode

This mode is suitable for regular sequences (typically one channel). The external trigger is sourced from the multi-channel switch of the master ADC, determined by ADC_CTRL2.EXTRSEL[5:0].

17.9.4.1 Dual ADC Mode

When a trigger is generated, the master ADC converts immediately, while the slave ADC starts conversion after a few ADC clock cycles. The minimum delay between two conversions in cross mode is configured through the ADC_CTRL1.DELAY[3:0] bits.

If both the master ADC and slave ADC's ADC_CTRL2.CTU are set, the selected regular sequence will be continuously converted.

When the slave ADC completes the conversion, a 32-bit DMA transfer request will be generated (if the ADC_CTRL2.DMAMD[1:0] bits are equal to 2b'10), and the converted data will be stored in the master ADC_DAT register. The upper half-word of ADC_DAT contains the conversion data from the slave ADC, while the lower half-word contains the conversion data from the master ADC. If the slave ADC sets the ADC_INTEN.ENDCAIEN, an ENDCA interrupt will be generated when the slave ADC's regular sequence conversion is completed.

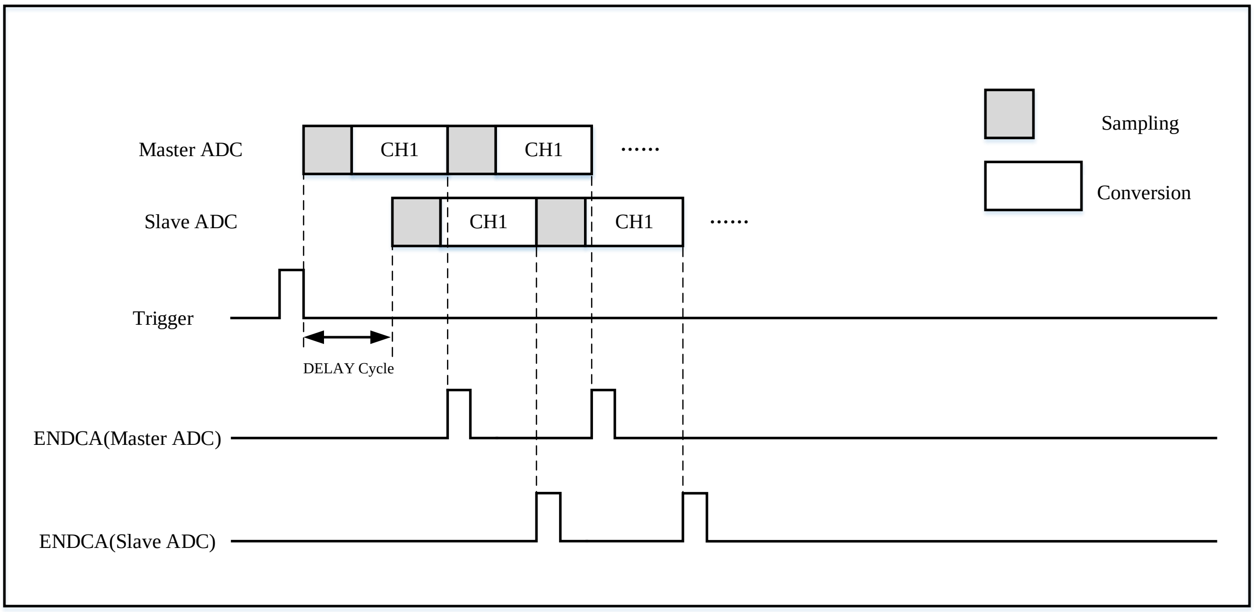
Notes :

(1) When using interleaved mode, ensure that no injection channel is externally triggered.

(2) The sampling time must be less than ADC_CTRL1.DELAY[3:0] clock cycles to avoid overlapping sampling periods when the master ADC and slave ADC convert the same channel.

(3) When using interleaved mode and DMA mode is 2 (ADC_CTRL2.DMAMD[1:0]=0b'10), it is necessary to enable FIFO.

Figure 17-27 Schematic diagram of fast interleaved mode conversion with continuous conversion of one channel.



17.9.4.2 Triple ADC Mode

When a trigger occurs, ADC1 converts immediately, ADC2 starts conversion after a few ADC clock cycles, and when ADC2 initiates conversion, ADC3 starts conversion after a few ADC clock cycles.

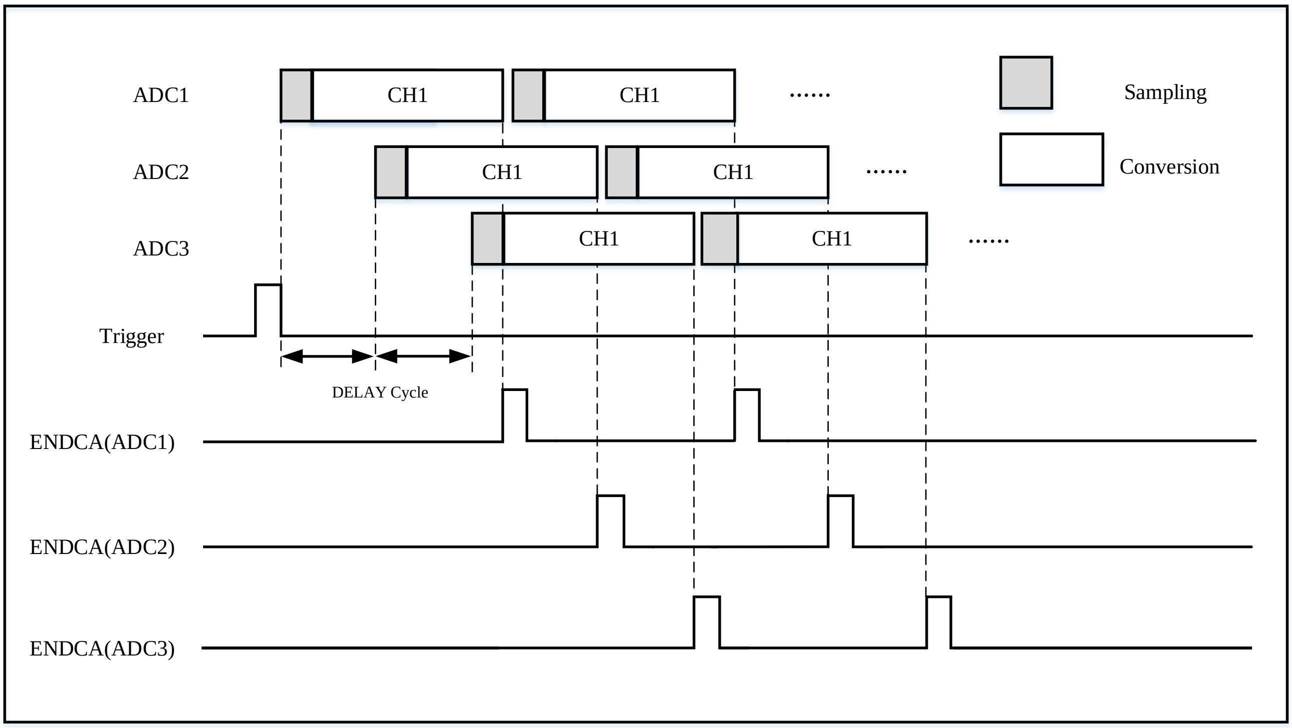
If the CTU bit is set for ADC1, ADC2, and ADC3 simultaneously, the three ADC regular channels will be converted continuously.

At the completion of the conversion for every two ADCs, a 32-bit DMA transfer request is generated (if the

ADC_CTRL2.DMAMD[1:0] bits are equal to 2b’10). The contents of the 32-bit ADC1_DAT register are transferred to SRAM, where the upper half-word contains the conversion data from the ADC started later, and the lower half-word contains the conversion data from the ADC started earlier. For example：

- The 1 request：ADC1_DAT [31:0] = ADC2_DAT[15:0] | ADC1_ DAT [15:0]
- The 2 request：ADC1_DAT [31:0] = ADC1_ DAT[15:0] | ADC3_ DAT[15:0]
- The 3 request：ADC1_DAT [31:0] = ADC3_ DAT[15:0] | ADC2_ DAT[15:0]
- The 4 request：ADC1_DAT [31:0] = ADC2_ DAT[15:0] | ADC1_DAT[15:0]……

Figure 17-28 Schematic diagram of fast Interleave mode conversion with continuous conversion of one channel



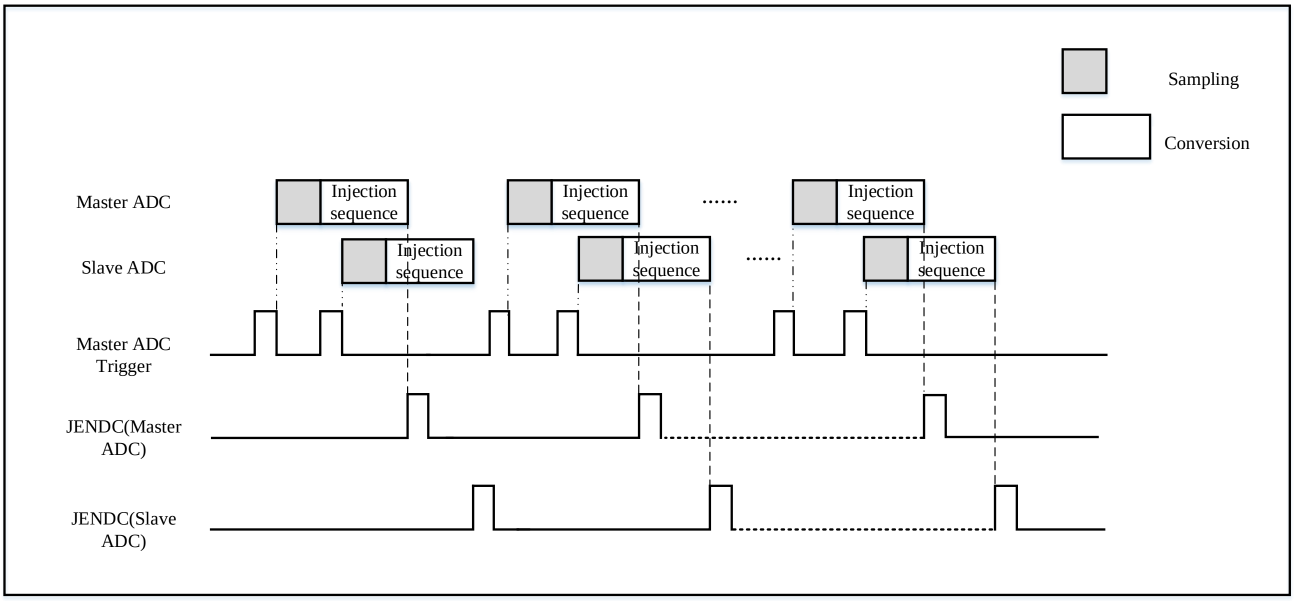
17.9.5 Alternate Trigger Mode

This mode is suitable for injection sequences.

17.9.5.1 Dual ADC Mode

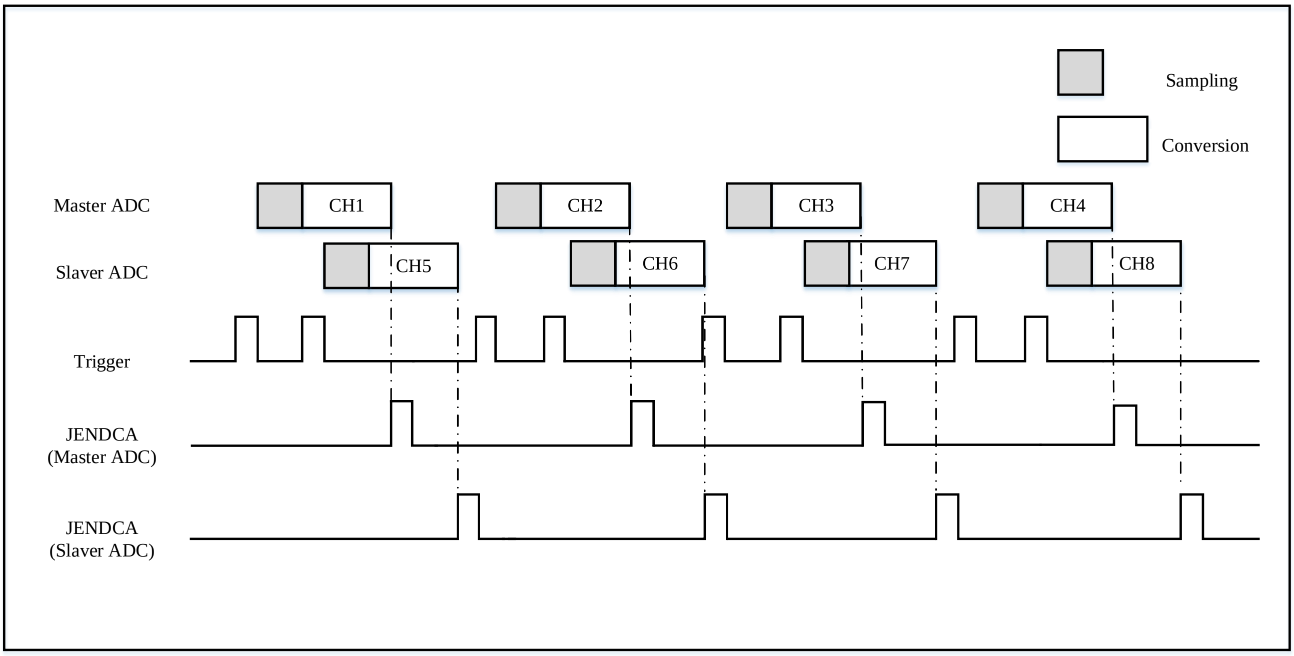
In dual ADC mode, the external trigger is sourced from the multi-channel switch of the master ADC, determined by ADC_CTRL2.EXTJSEL[5:0]. When the first trigger occurs, all injection channels of the master ADC are converted, and when the second trigger occurs, all injection channels of the slave ADC are converted, and this cycle repeats. If the master ADC or slave ADC sets the ADC_INTEN.JENDCIEN, an JENDC interrupt will be generated when the injection sequence conversion of the master ADC or slave ADC is completed. After all injection sequences have been converted, another external trigger occurs, then the alternating trigger will restart.

Figure 17-29 Alternate trigger: injection channel group



If the injection discontinuous mode is simultaneously used on the master ADC and slave ADC, when the first trigger occurs, the first group of injection channels of the master ADC is converted. When the second trigger occurs, the first group of injection channels of the slave ADC is converted. When the third trigger occurs, the second group of injection channels of the master ADC is converted, and when the fourth trigger occurs, the second group of injection channels of the slave ADC is converted, and this cycle repeats. If the master ADC or slave ADC sets the ADC_INTEN.JENDCIEN, an JENDC interrupt will be generated when the injection sequence conversion of the master ADC or slave ADC is completed. After all injection sequences have been converted, another external trigger occurs, then the alternating trigger will restart.

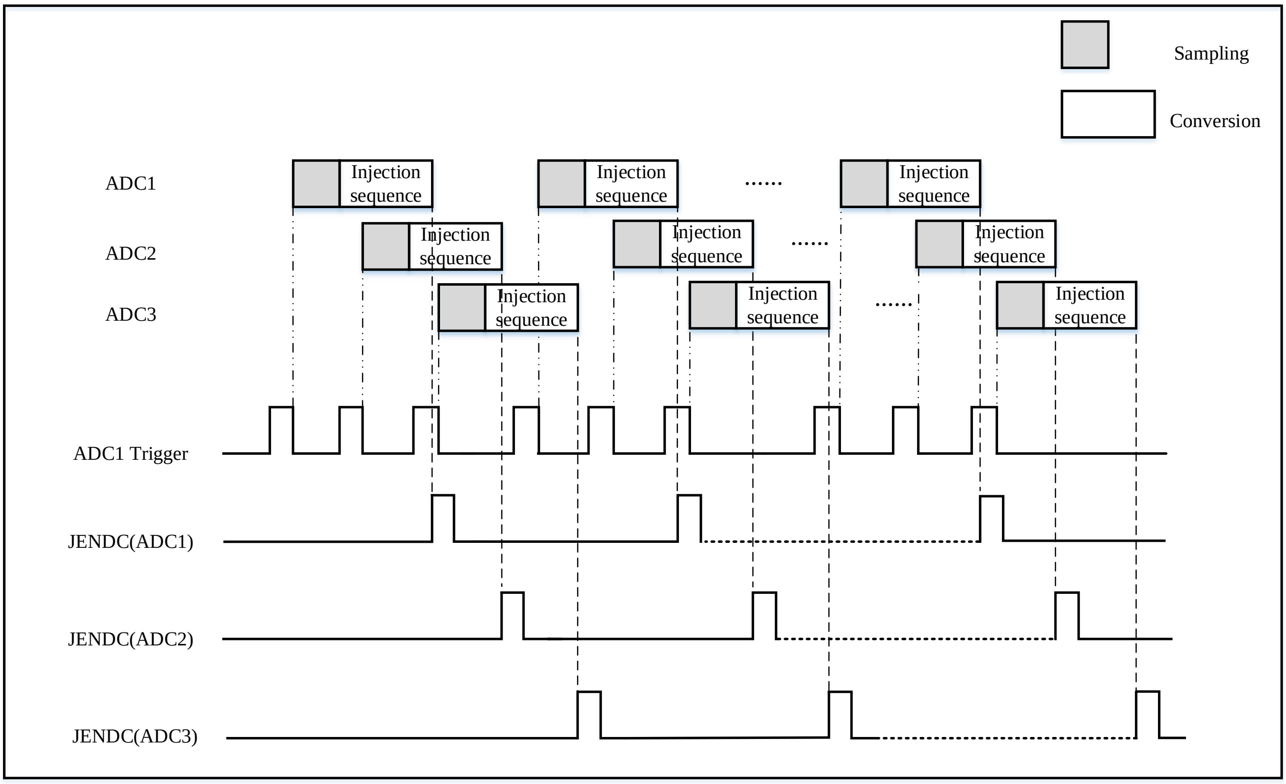
Figure 17-30 Alternate trigger: Injection channel group in Discontinuous mode



17.9.5.2 Triple ADC Mode

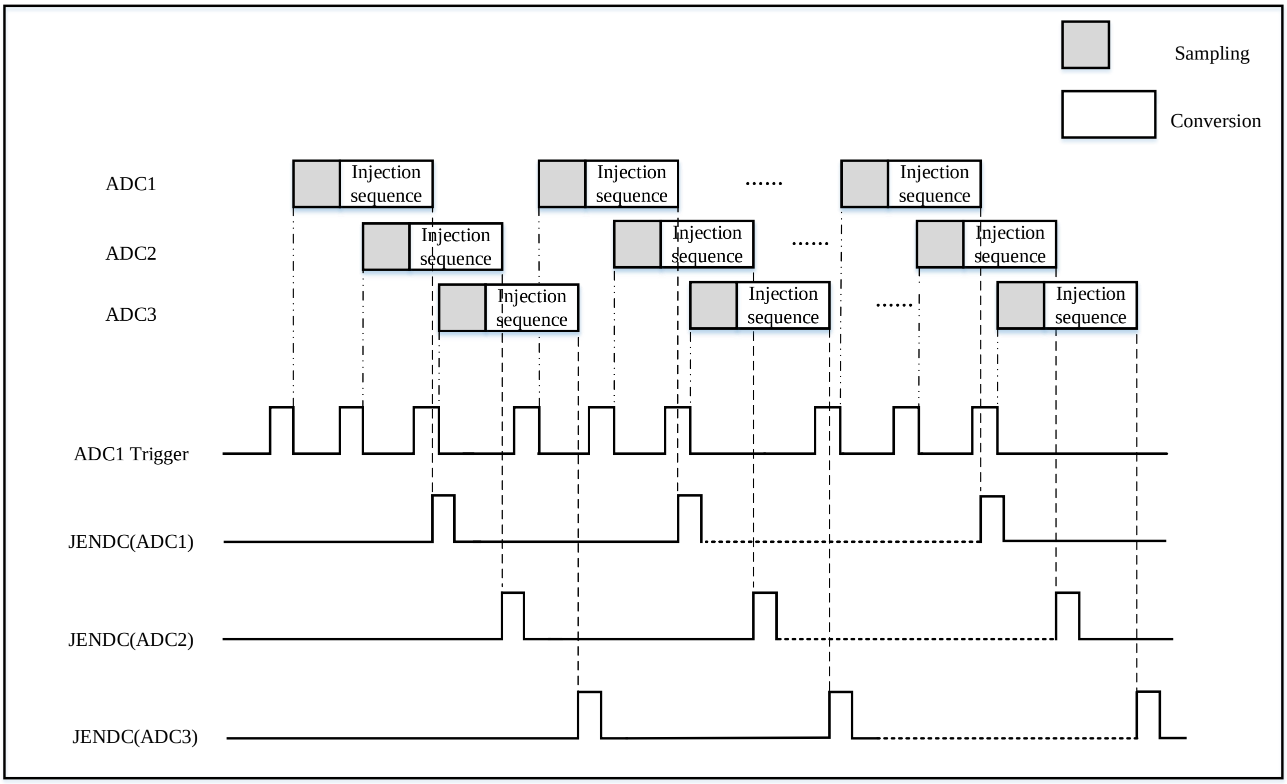
In the triple ADC mode, the external trigger is sourced from the multi-channel switch of ADC1, determined by ADC_CTRL2.EXTJSEL[5:0]. When the first trigger occurs, all injection channels of ADC1 are converted. When the second trigger occurs, all injection channels of ADC2 are converted. When the third trigger occurs, all injection channels of ADC3 are converted, and this cycle repeats. If ADC1 or ADC2 or ADC3 sets the ADC_INTEN.JENDCIEN, an JENDC interrupt will be generated when the injection sequence conversion of ADC1 or ADC2 or ADC3 is completed. After all injection sequences have been converted, another external trigger occurs, then the alternating trigger will restart.

Figure 17-31 Alternate trigger: Injection channel group



If the injection discontinuous mode is simultaneously used on ADC1, ADC2, and ADC3, when the first trigger occurs, the first group of injection channels of ADC1 is converted. When the second trigger occurs, the first group of injection channels of ADC2 is converted. When the third trigger occurs, the first group of injection channels of ADC3 is converted. When the fourth trigger occurs, the second group of injection channels of ADC1 is converted. When the fifth trigger occurs, the second group of injection channels of ADC2 is converted. When the sixth trigger occurs, the second group of injection channels of ADC3 is converted, and this cycle repeats. If ADC1, ADC2, or ADC3 sets the ADC_INTEN.JENDCIEN, an JENDC interrupt will be generated when the injection sequence conversion of ADC1, ADC2, or ADC3 is completed. After all injection sequences have been converted, another external trigger occurs, then the alternating trigger will restart.

Figure 17-32 Alternate Trigger: Injection Channel Group in Discontinuous Mode



17.9.6 Combined Synchronous Regular and Synchronous Injection Mode

In this mode, the conversion of synchronous injection channels can interrupt the conversion of synchronous regular channels.

Note: In this mode, the sequences for synchronous conversion by the master ADC and slave ADC need to be set to the same time, or the interval of trigger signals should be greater than the conversion time of the longer sequence. If the interval of trigger signals is shorter than the conversion time of the longer sequence, the longer sequence may restart before completion, causing the shorter sequence to restart as well.

17.9.7 Combined Synchronous Regular and Alternate Trigger Mode

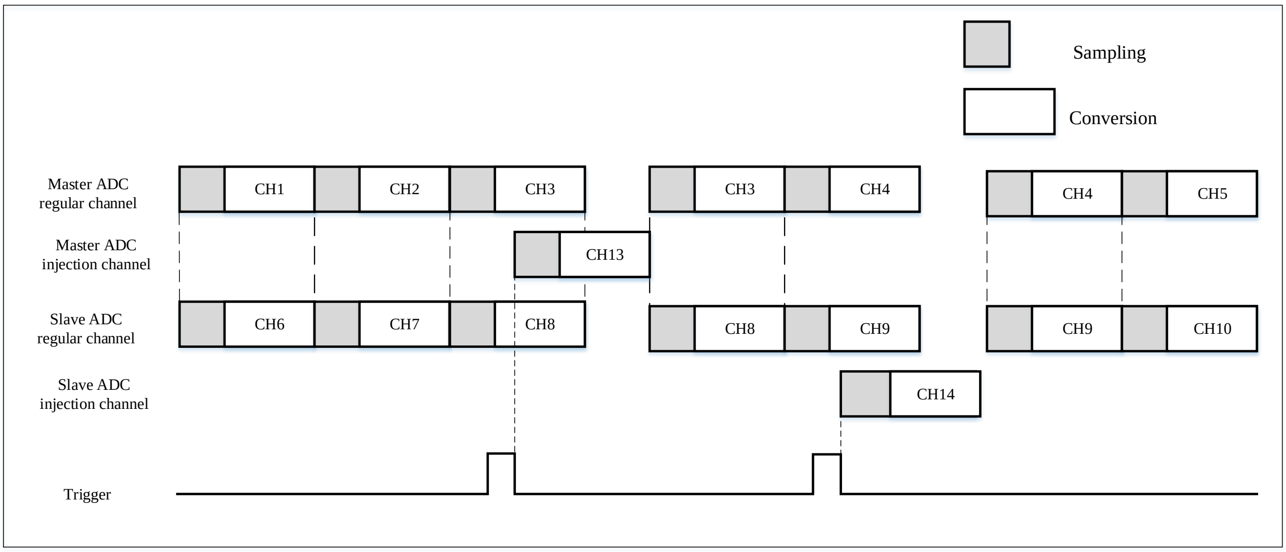
The alternate trigger conversion of injection channels can interrupt the conversion of synchronous regular channels.

When an injection channel event occurs, the alternate injection conversion starts immediately. If regular conversions are in progress, both the master ADC and slave ADC regular conversions will stop to ensure that regular conversions can resume synchronously after the injection conversions are completed.

Note: In this mode, the sequences for synchronous conversion by the master ADC and slave ADC need to be set to the same time, or the interval of trigger signals should be greater than the conversion time of the longer sequence. If the interval of trigger signals is shorter than the conversion time of the longer sequence, the longer sequence may restart before completion, causing the shorter sequence to restart as well.

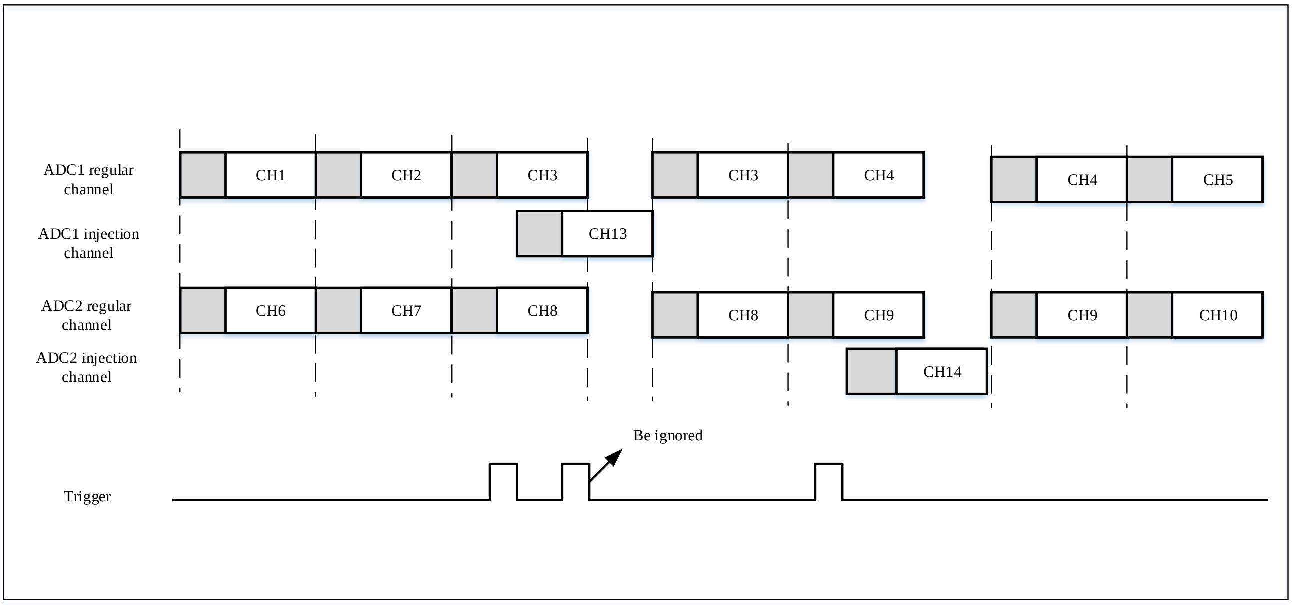
Note: In this mode, and when the reset mode (ADC_CTRL3.OSRMD=1) and oversampling are enabled, the sampling results will be incorrect. The continuous mode (ADC_CTRL3.OSRMD=0) can be used instead of the reset mode

Figure 17-33 Combination of Alternate Mode and Synchronous Regular Mode


Sampling
Conversion
Master ADC regular channel
Master ADC injection channel
Slave ADC regular channel
Slave ADC injection channel
Trigger
CH1
CH2
CH3
CH3
CH4
CH4
CH5
CH13
CH6
CH7
CH8
CH8
CH9
CH9
CH10
CH14

If another injection trigger occurs during an injection conversion, this trigger will be ignored. As shown in the diagram below:

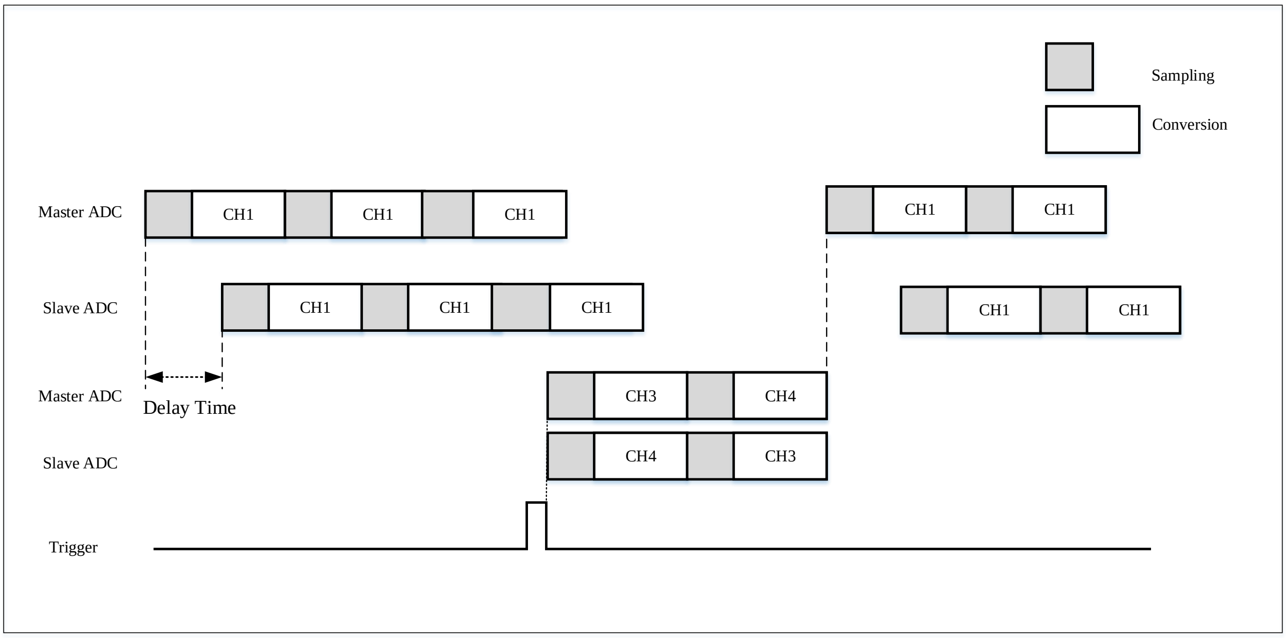
Figure 17-34 Injection trigger occurs during injection conversion



17.9.8 Combined Synchronous Injection and Interleaved Mode

In this mode, when an injection trigger occurs, the cross conversion is interrupted, and the injection conversion is initiated. After the injection conversion is completed, the cross conversion is resumed.

Figure 17-35 The interleaved single-channel conversion is interrupted by the injection sequences CH3 and CH4

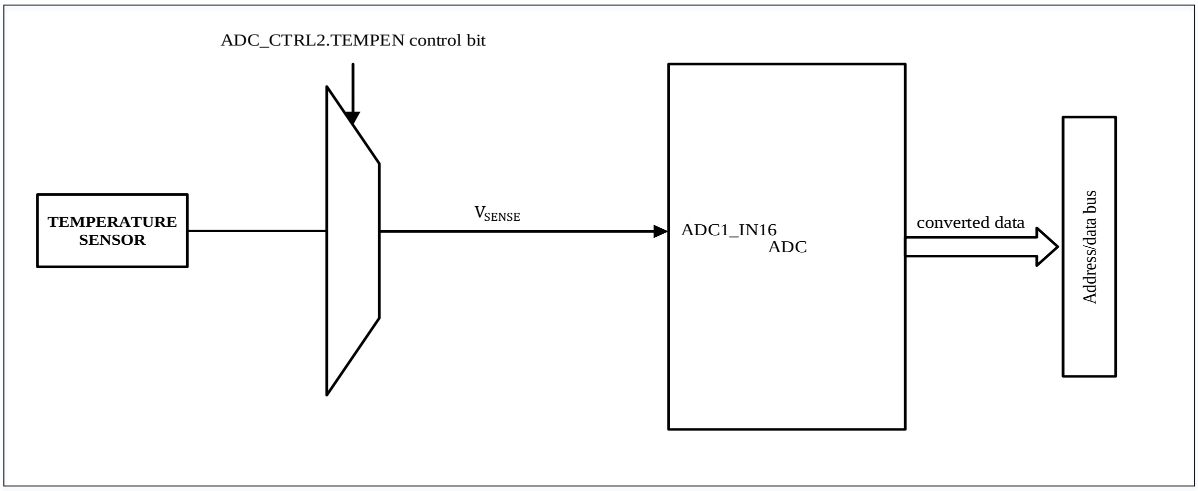


17.10 Temperature Sensor

Set the ADC_CTRL2.TEMPEN bit to 1 to enable the temperature sensor, and use the temperature sensor to detect the ambient temperature when the device is operating. The output voltage sampled by the temperature sensor is converted into a digital value by the ADC_IN16 channel. When the temperature sensor is working, the ideal sampling time is 17.1us; when the temperature sensor is not working, the ADC_CTR2.TEMPEN bit can be cleared by software to reduce power consumption. As follows Figure Figure 17-36 is a block diagram of a temperature sensor.

The output voltage of the temperature sensor changes linearly with temperature. Different chips will have different offsets in the temperature curve due to different production processes. Through testing, it is found that the maximum offset is 3°C. This characteristic makes the internal temperature sensor more suitable for detecting temperature changes. Not suitable for measuring absolute temperature. When accurate temperature measurement is required, an external temperature sensor should be used.

Figure 17-36 Temperature Sensor Channel Diagram



17.10.1 Measure Temperature Values

1. Configure the channel (ADC_IN16) and sampling time of the channel to be 17.1 us.
2. Set ADC_CTRL2.TEMPEN bit to 1 to enable temperature sensor.
3. Set ADC_CTRL2.ON bit to 1 to start ADC conversion (or through external trigger).
4. Read the temperature data in the ADC data register, and calculate the temperature value by the following formula：

$$\text{Temperature}(°C) = \{(V_{SENSE} - V_{Temperature}) / \text{Avg\_Slope}\} + \text{Temperature} - T_{offset}$$

In which：

$V_{Temperature}$ = Temperature corresponding to the V_{SENSE}

Avg_Slope ＝ temperature and Average slope of a V_{SENSE} curve (mV/°C or μV/°C)

T_{offset} = 1.25°C, Representation of empirical temperature error compensation（°C）

Temperature is the calibrated temperature.

The actual value of Avg_Slope in the electrical characteristics section of the reference data sheet.

Notes：

1. *There is a settling time from power-down mode to the correct output VSENSE of the sensor; there is also a settling time after the ADC is powered on. Therefore, to reduce the delay, both the ADC_CTRL2.TEMPEN and ADC_CTRL2.ON bits should be set simultaneously.*
2. *This temperature is the junction temperature of the chip, which can differ significantly from the ambient temperature of the chip*

17.11 ADC Interrupt

ADC interrupts can be from an end of regular or injected sequence conversion, an analog watchdog event when input voltage exceeds the threshold, any end of regular or injected channel conversion. These interrupts have independent interrupt enable bits.

There are 3 status flags in the ADC_STS register: injected sequence channel conversion started (JSTR), regular sequence channel conversion started (STR) and Truncated Conversion Flag (TCFLAG) after oversampling. But there are no interrupts associated with these three flags in the ADC.

Table 17-8 ADC Interrupt

Interrupt Event	Event Flags	Enable Control Bit
Regular sequence is complete	ENDC	ENDCIEN
Injection sequence conversion is complete	JENDC	JENDCIEN
Exceeding analog watchdog 1 threshold	AWDG1	AWD1IEN
Exceeding analog watchdog 2 threshold	AWD2FLAG[18:0]	AWD2INTEN[18:0]
Exceeding analog watchdog 3 threshold	AWD3FLAG[18:0]	AWD3INTEN [18:0]
End of any regular channel conversion	ENDCA	ENDCAIEN
End of any injection channel conversion	JENDCA	JENDCAIEN
Completion of regular channel sampling	EOSAMP	EOSMPIEN
ADC ready	RDY	RDYIEN
ADC power down ready	PDRDY	PDRDYIEN
End of conversion with error	ENDCERR	ENDCERRIEN

17.12 ADC Register

17.12.1 ADC Register Overview

Table 17-9 ADC Register Overview

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	ADC_STS	Reserved																				JSTR	STR	TCFLAG	EOSAMP	PDRDY	RDY	ENDCERR	AWDG1	JENDCA	JENDC	ENDCA	ENDC
	Reset value																					0	0	0	0	0	0	0	0	0	0	0	0
0x04	ADC_CTRL1	Reserved								DELAY[3:0]				AWD1SGLEN	AWD1EJCH	AWD1ERCH	AWD1CH[4:0]					MULTMODE[4:0]					DJCH	DREGCH	DCTU[2:0]			AUTOJC	SCANMD
	Reset value									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x08	ADC_CTRL2	Reserved	SWSTOP	JSWSTOP	SWSTRRCH	SWSTRJCH	GCOMPEN	TEMPEN	ALIG	DMAMD[1:0]		EXTRSEL[5:0]						EXTJSEL[5:0]						Reserved		EXTPJSEL[1:0]		EXTPRSEL[1:0]		RSTCAL	ENCAL	CTU	ON
	Reset value		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			0	0	0	0	0	0	0	0
0x0C	ADC_CTRL3	Reserved									INP2SEL	INN1SEL	INP1SEL	OSRMD	OSRTRIG	OSJE	OSRE	OSS[3:0]				OSR[3:0]				VABTMEN	DPWMOD	BPCAL	CKMOD	CALALD	CALDIF	RES[1:0]	
	Reset value										0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
0x10	ADC_SAMPR1	SAMP7[3:0]				SAMP6[3:0]				SAMP5[3:0]				SAMP4[3:0]				SAMP3[3:0]				SAMP2[3:0]				SAMP1[3:0]				SAMP0[3:0]			
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x14	ADC_SAMPR2	SAMP15[3:0]				SAMP14[3:0]				SAMP13[3:0]				SAMP12[3:0]				SAMP11[3:0]				SAMP10[3:0]				SAMP9[3:0]				SAMP8[3:0]			
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x18	ADC_SAMPR3	Reserved																				SAMP18[3:0]				SAMP17[3:0]				SAMP16[3:0]			
	Reset value																					0	0	0	0	0	0	0	0	0	0	0	0
0x1C	ADC_DIFSEL	Reserved													DIFSEL[17:0]																		Reserved
	Reset value														0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x20	ADC_CALFACT	Reserved									CALFACTD[6:0]							Reserved									CALFACTS[6:0]						
	Reset value										0	0	0	0	0	0	0										0	0	0	0	0	0	0
0x24	ADC_OFFSET1	OFFSCH1EN	OFFSCH1CH[4:0]					OFFSCH1SATE	OFFSCH1DIR	Reserved												OFFSCH1DAT[11:0]											
	Reset value	0	0	0	0	0	0	0	0													0	0	0	0	0	0	0	0	0	0	0	0
0x28	ADC_OFFSET2	OFFSCH2EN	OFFSCH2CH[4:0]					OFFSCH2SATE	OFFSCH2DIR	Reserved												OFFSCH2DAT[11:0]											
	Reset value	0	0	0	0	0	0	0	0													0	0	0	0	0	0	0	0	0	0	0	0
0x2C	ADC_OFFSET3	OFFSCH3EN	OFFSCH3CH[4:0]					OFFSCH3SATE	OFFSCH3DIR	Reserved												OFFSCH3DAT[11:0]											

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reset value	0	0	0	0	0	0	0	0													b	0	0	0	0	0	0	0	0	0	0	0
0x30	ADC_OFFSET4	OFFSCH4EN	OFFSCH4CH[4:0]					OFFSCH4SATE	OFFSCH4DIR	Reserved												OFFSCH4DAT[11:0]											
	Reset value	0	0	0	0	0	0	0	0													0	0	0	0	0	0	0	0	0	0	0	0
0x34	ADC_AWD1 HIGH	Reserved																				HTH1[11:0]											
	Reset value																					1	1	1	1	1	1	1	1	1	1	1	1
0x38	ADC_AWD1 LOW	Reserved																	AWD1FILT[2:0]			LTH1[11:0]											
	Reset value																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x3C	ADC_AWD2 HIGH	Reserved																				HTH2[11:0]											
	Reset value																					1	1	1	1	1	1	1	1	1	1	1	1
0x40	ADC_AWD2 LOW	Reserved																				LTH2[11:0]											
	Reset value																					0	0	0	0	0	0	0	0	0	0	0	0
0x44	ADC_AWD3 HIGH	Reserved																				HTH3[11:0]											
	Reset value																					1	1	1	1	1	1	1	1	1	1	1	1
0x48	ADC_AWD3 LOW	Reserved																				LTH3[11:0]											
	Reset value																					0	0	0	0	0	0	0	0	0	0	0	0
0x4C	ADC_AWD2EN	Reserved													AWD2EN[18:0]																		
	Reset value														0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x50	ADC_AWD3EN	Reserved													AWD3EN[18:0]																		
	Reset value														0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x54	ADC_AWD2 INTEN	Reserved													AWD2INTEN[18:0]																		
	Reset value														0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x58	ADC_AWD3 INTEN	Reserved													AWD3INTEN[18:0]																		
	Reset value														0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x5C	ADC_AWD2STS	Reserved													AWD2FLAG[18:0]																		
	Reset value														0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x60	ADC_AWD3STS	Reserved													AWD3FLAG[18:0]																		
	Reset value														0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x64	ADC_RSEQ1	Reserved		SEQ6[4:0]					SEQ5[4:0]					SEQ4[4:0]					SEQ3[4:0]					SEQ2[4:0]					SEQ1[4:0]				
	Reset value			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x68	ADC_RSEQ2	Reserved		SEQ12[4:0]					SEQ11[4:0]					SEQ10[4:0]					SEQ9[4:0]					SEQ8[4:0]					SEQ7[4:0]				
	Reset value			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x6C	ADC_RSEQ3	Reserved			LEN[3:0]				Reserved					SEQ16[4:0]					SEQ15[4:0]					SEQ14[4:0]					SEQ13[4:0]				
	Reset value				0	0	0	0						0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x70	ADC_JSEQ	Reserved					JLEN[1:0]		Reserved					JSEQ4[4:0]					JSEQ3[4:0]					JSEQ2[4:0]					JSEQ1[4:0]				
	Reset value						0	0						0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x74	ADC_JDAT1	Reserved																JDAT1[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x78	ADC_JDAT2	Reserved																JDAT2[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x7C	ADC_JDAT3	Reserved																JDAT3[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x80	ADC_JDAT4	Reserved																JDAT4[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x84	ADC_DAT	DAT2[15:0]																DAT1[15:0]															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x88	ADC_FIFOCFG	Reserved																				NEINTEN	CLR	WL[3:0]				EN	HFINTEN	EINTEN	FINTEN	WEINTEN	REINTEN
	Reset value																					0	0	0	0	0	0	0	0	0	0	0	0
0x8C	ADC_FIFOSTS	Reserved																				NEFLAG	Reserved	DATCNT[4:0]					HFFLAG	EFLAG	FFLAG	WEFLAG	REFLAG
	Reset value																					0		0	0	0	0	0	0	1	0	0	0
0x90	ADC_P CTRL	Reserved																										EXPEN	RANGESE	VREFLDO	Reserved		
	Reset value																											0	0	0			
0x94	ADC_CTRL4	Reserved								EXTRISEL[3:0]				EXTRRSEL[3:0]				Reserved															
	Reset value									0	0	1	0	1	1	1	0																

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x98	ADC_INTEN	Reserved																							EOSMPIEN	PDRDYIEN	RDYIEN	EOCERRIEN	AWD1IEN	JENDCAIEN	JENDCIEN	ENDCAIEN	ENDCIEN
	Reset value																								0	0	0	0	0	0	0	0	0
0x9C	ADC_GCOMP	Reserved																		GCOMPDAT[13:0]													
	Reset value																			0	0	0	0	0	0	0	0	0	0	0	0	0	0

17.12.2 ADC Status Register(ADC_STS)

Offset address：0x00

Reset value：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				JSTR	STR	TCFLAG	EOSAMP	PDRDY	RDY	ENDC ERR	AWDG1	JENDCA	JENDC	ENDCA	ENDC
				rc-w1	rc-w1	rc-w1	rc-w1	rc-w1	rc-w1	rc-w1	rc-w1	rc-w1	rc-w1	rc-w1	rc-w1

Bit Field	Name	Description
31:12	Reserved	Reserved, the reset value must be maintained
11	JSTR	Inject channel start flag When the injection channel conversion starts, this bit is set to 1 by hardware and cleared by software writing 1. 0: Injection channel conversion not started; 1: Injection channel conversion started.
10	STR	Regular channel start flag When the regular channel conversion starts, this bit is set to 1 by hardware and cleared by software writing 1. 0: Regular channel conversion not started; 1: Regular channel conversion started.
9	TCFLAG	Oversampling data turncate flag When oversampling is enabled, if the result after right shift exceeds 16 bits, this bit is set to 1 by hardware and cleared by software writing 1. 0: Oversampled data not truncated; 1: Oversampled data truncated.
8	EOSAMP	Regular channel sample end flag When the regular channel sampling is completed, this bit is set to 1 by hardware and cleared by software writing 1. 0: Regular channel sampling not completed；

Bit Field	Name	Description
		1: Regular channel sampling completed.
7	PDRDY	Power down ready This bit is set to 1 by hardware and cleared by software writing 1. 0: Not ready; 1: Ready.
6	RDY	ADC Ready This bit is set to 1 by hardware and cleared by software writing 1. 0: Not Ready; 1: Ready.
5	ENDCERR	End of conversion error flag When an exception occurs during regular channel conversion, this bit is set to 1 by hardware and cleared by software writing 1. 0: ENDC signal normal; 1: ENDC signal error.
4	AWDG1	Analog watchdog 1 flag When the converted voltage value exceeds the range defined by the ADC_AWD1HIGH.HTH and ADC_AWD1LOW.LTH registers, this bit is set to 1 by hardware and cleared by software writing 1. 0: No analog watchdog event occurred; 1: Analog watchdog event occurred.
3	JENDCA	Any injected channel end of conversion Set to1 by hardware when the any injection channel sequence conversion ends, cleared by software writing 1. 0: Conversion not completed; 1: Conversion completed.
2	JENDC	Injected channel end of conversion Set to 1 by hardware when the injection channel sequence conversion ends, cleared by software writing 1. 0: Conversion not completed; 1: Conversion completed.
1	ENDCA	Any end of conversion flag Set to 1 by hardware when any regular channel conversion ends, cleared by software writing 1. 0: Conversion not completed; 1: Conversion completed.
0	ENDC	End of conversion Set to 1 by hardware when the regular channel sequence conversion ends, cleared by software writing 1. 0: Conversion not completed; 1: Conversion completed.

17.12.3 ADC Control Register 1(ADC_CTRL1)

Offset address：0x04

Reset value：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								DELAY[3:0]				AWD1 SGLEN	AWD1 EJCH	AWD1 ERCH	AWD1 CH[4]
								rw				rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AWD1CH[3:0]				MULTMODE[4:0]					DJCH	DREGCH	DCTU[2:0]			AUTOJC	SCANMD
rw				rw					rw	rw	rw			rw	rw

<table>
<tr><th>Bit Field</th><th>Name</th><th>Description</th></tr>
<tr><td>31:24</td><td>Reserved</td><td>Reserved, the reset value must be maintained.</td></tr>
<tr><td>23:20</td><td>DELAY[3:0]</td><td>This bit is used for interleaved mode in dual ADC or triple ADC mode.
Please refer to the table below for the setting of ADC DELAY at different resolutions：
<table>
<tr><th>DELAY[3:0]</th><th>12-bit</th><th>10-bit</th><th>8-bit</th><th>6-bit</th></tr>
<tr><td>0000</td><td>1 * T_{ADC_CLK}</td><td>1 * T_{ADC_CLK}</td><td>1 * T_{ADC_CLK}</td><td>1 * T_{ADC_CLK}</td></tr>
<tr><td>0001</td><td>2 * T_{ADC_CLK}</td><td>2 * T_{ADC_CLK}</td><td>2 * T_{ADC_CLK}</td><td>2 * T_{ADC_CLK}</td></tr>
<tr><td>0010</td><td>3 * T_{ADC_CLK}</td><td>3 * T_{ADC_CLK}</td><td>3 * T_{ADC_CLK}</td><td>3 * T_{ADC_CLK}</td></tr>
<tr><td>0011</td><td>4 * T_{ADC_CLK}</td><td>4 * T_{ADC_CLK}</td><td>4 * T_{ADC_CLK}</td><td>4 * T_{ADC_CLK}</td></tr>
<tr><td>0100</td><td>5 * T_{ADC_CLK}</td><td>5 * T_{ADC_CLK}</td><td>5 * T_{ADC_CLK}</td><td>5 * T_{ADC_CLK}</td></tr>
<tr><td>0101</td><td>6 * T_{ADC_CLK}</td><td>6 * T_{ADC_CLK}</td><td>6 * T_{ADC_CLK}</td><td>6 * T_{ADC_CLK}</td></tr>
<tr><td>0110</td><td>7 * T_{ADC_CLK}</td><td>7 * T_{ADC_CLK}</td><td>7 * T_{ADC_CLK}</td><td>6 * T_{ADC_CLK}</td></tr>
<tr><td>0111</td><td>8 * T_{ADC_CLK}</td><td>8 * T_{ADC_CLK}</td><td>8 * T_{ADC_CLK}</td><td>6 * T_{ADC_CLK}</td></tr>
<tr><td>1000</td><td>9 * T_{ADC_CLK}</td><td>9 * T_{ADC_CLK}</td><td>8 * T_{ADC_CLK}</td><td>6 * T_{ADC_CLK}</td></tr>
<tr><td>1001</td><td>10 * T_{ADC_CLK}</td><td>10 * T_{ADC_CLK}</td><td>8 * T_{ADC_CLK}</td><td>6 * T_{ADC_CLK}</td></tr>
<tr><td>1010</td><td>11 * T_{ADC_CLK}</td><td>10 * T_{ADC_CLK}</td><td>8 * T_{ADC_CLK}</td><td>6 * T_{ADC_CLK}</td></tr>
<tr><td>1011</td><td>12 * T_{ADC_CLK}</td><td>10 * T_{ADC_CLK}</td><td>8 * T_{ADC_CLK}</td><td>6 * T_{ADC_CLK}</td></tr>
<tr><td>Other</td><td>12 * T_{ADC_CLK}</td><td>10 * T_{ADC_CLK}</td><td>8 * T_{ADC_CLK}</td><td>6 * T_{ADC_CLK}</td></tr>
</table>
</td></tr>
<tr><td>19</td><td>AWD1SGLEN</td><td>Enable the watchdog 1 on a single channel in scan mode
This bit is set and cleared by software to enable the Analog Watchdog 1 function on the channels specified by AWD1CH[4:0] or enable the Analog Watchdog 1 function on all channels.
0: Using Analog watchdog 1 on all channels;
1: Using Analog watchdog 1 on a single channel.</td></tr>
<tr><td>18</td><td>AWD1EJCH</td><td>Analog watchdog 1 enable on injected channels
This bit is set and cleared by software.
0: Disabling Analog Watchdog 1 on the injection channel;
1: Using Analog Watchdog 1 on the injection channel.</td></tr>
<tr><td>17</td><td>AWD1ERCH</td><td>Analog watchdog 1 enable on regular channels
This bit is set and cleared by software.</td></tr>
</table>

Bit Field	Name	Description
		0: Disabling Analog Watchdog 1 on the regular channel; 1: Using Analog Watchdog 1 on the regular channel.

16:12	AWD1CH[4:0]	Analog watchdog channel select bits
		These bits are set and cleared by software to select the input channels protected by Analog Watchdog 1.
		00000: ADC analog input channel 0;
		00001: ADC analog input channel 1;
		……
		10000: ADC analog input channel 16;
		10001: ADC analog input channel 17;
		10010: ADC analog input channel 18;
		Other: Reserved

11:7	MULTMODE[4:0]	Multi-ADC mode selection 00001 – 01001: Dual ADC operating mode (ADC1 and ADC2, ADC3 and ADC4) 00000: Independent mode 00001: Combined synchronous regular + synchronous injection mode 00010: Combined synchronous regular + alternate trigger mode 00011: Combined synchronous injection + interleaved mode 00100: Reserved 00101: Synchronous injection mode 00110: Synchronous regular mode 00111: Interleaved mode 01000: Reserved 01001: Alternate Trigger mode 10001 – 11001: Triple ADC operating mode(ADC1， ADC2 and ADC3) 10001: Combined synchronous regular + synchronous injection mode 10010: Combined synchronous regular + alternate trigger mode 10011: Combined synchronous injection + interleaved mode 10100: Reserved 10101: Synchronous injection mode 10110: Synchronous regular mode 10111: Interleaved mode 11000: Reserved 11001: Alternate Trigger mode *Note：* *1. In the slave ADC, these bits should be configured the same as the master ADC;* *2. In dual ADC or triple ADC mode, changing the channel configuration will result in a condition that triggers a restart, causing synchronization loss. It is recommended to disable the dual ADC or triple ADC mode before making any configuration changes.*
6	DJCH	Discontinuous mode on injected channels This bit is set and cleared by software to enable or disable discontinuous mode on the injection channel group

Bit Field	Name	Description
		0: Disable discontinuous mode on the injection channel group； 1: Use discontinuous mode on the injection channel group.
5	DREGCH	Discontinuous mode on regular channels This bit is set and cleared by software to enable or disable discontinuous mode on the regular channel group. 0: Disable discontinuous mode on the regular channel group; 1: Use discontinuous mode on the regular channel group.
4:2	DCTU[2:0]	Discontinuous mode channel count Software defines the number of regular channel conversions to be performed after receiving an external trigger in discontinuous mode using these bits. 000: 1 channel 001: 2 channels …… 111: 8 channels
1	AUTOJC	Automatic injected group conversion This bit is set and cleared by software to enable or disable the injection channel sequence conversion after the regular channel sequence conversion ends. 0: Disable automatic injection channel sequence conversion； 1: Enable automatic injection channel sequence conversion.
0	SCANMD	Scan mode This bit is set and cleared by software to enable or disable scan mode. In scan mode, conversions are performed on the channels selected by the ADC_RSEQx or ADC_JSEQ registers. 0: Disable scan mode; 1: Enable scan mode. *Note: If only the ADC_INTEN.ENDCIEN or ADC_INTEN.JENDCIEN bit is set, the ADC_STS.ENDC or ADC_STS.JENDC interrupt occurs only after the last channel conversion.*

17.12.4 ADC Control Register 2 (ADC_CTRL2)

Offset address：0x08

Reset value：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved	SWR STOP	SWJ STOP	SWSTR RCH	SWSTR JCH	GCOMP EN	TEMPEN	ALIG	DMAMD[1:0]		EXTRSEL[5:0]					
	rw	rw	rw	rw	rw	rw	rw	rw		rw					

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTJSEL[5:0]						Reserved		EXTPJSEL[1:0]		EXTPRSEL[1:0]		RSTCAL	ENCAL	CTU	ON
rw								rw		rw		rw	rw	rw	rw

Bit Field	Name	Description
31	Reserved	Reserved, the reset value must be maintained
30	SWRSTOP	Stop conversion of regular channels This bit is set by software to stop the ongoing conversion of regular channels. After stopping, the user can reconfigure the conversion sequence, trigger source, etc. This bit is cleared by hardware. 0: Reset state; 1: Stop conversion of regular channels.
29	SWJSTOP	Stop conversion of injection channels This bit is set by software to stop the ongoing conversion of injection channels. After stopping, the user can reconfigure the conversion sequence, trigger source, etc. This bit is cleared by hardware. 0: Reset state; 1: Stop conversion of injection channels.
28	SWSTRRCH	Start conversion of regular channels This bit is set by software to start the conversion, and is cleared by hardware after the conversion begins. If SWSTRRCH is selected as the trigger event in ADC_CTRL2.EXTRSEL[5:0] bits, this bit is used to start a group of regular channel conversions. 0: Reset state; 1: Start conversion of regular channels.
27	SWSTRJCH	Start conversion of injection channels This bit is set by software to start the conversion and can be cleared immediately by software or hardware after the conversion begins. If SWSTRJCH is selected as the trigger event in the ADC_CTRL2.EXTJSEL[5:0] bits, this bit is used to start a group of injection channel conversions. 0: Reset state; 1: Start conversion of injection channels.
26	GCOMPEN	Gain compensation mode This bit is set and cleared by software to enable or disable the gain compensation mode. 0: Do not enable gain compensation mode; 1: Use gain compensation mode.
25	TEMPEN	Temperature sensor enable This bit is set and cleared by software to enable or disable the temperature sensor channel. 0: Disable temperature sensor channel measurement 1: Enable temperature sensor channel measurement
24	ALIG	Data alignment This bit is set and cleared by software. 0: Right-aligned; 1: Left-aligned.
23:22	DMAMD[1:0]	DMA mode This bit is set and cleared by software to configure the DMA operating mode. 00: DMA mode 0, disable DMA

Bit Field	Name	Description
		01: DMA mode 1 (transfers half a word per DMA request. The order of transfers for multiple ADCs is ADC1, ADC2, ADC3, ADC1, ADC2......) 10: DMA mode 2 (transfers one word per DMA request. The sequence is ADC2|ADC1, ADC1& ADC3, ADC3& ADC2) 11: DMA mode 3 (transfers half a word per DMA request. The sequence is ADC2|ADC1, ADC1& ADC3, ADC3& ADC2). *Note：* *1. For single ADC, only DMA mode 0 and mode 1 are supported；* *2. For multiple ADCs, DMA mode 1, mode 2, and mode 3 are supported；* *3. For multiple ADCs, Only ADC1 and ADC3 can generate DMA requests.*
21:16	EXTRSEL[5:0]	External event select for regular group These bits select the external event used to trigger the regular sequence conversion. ADC trigger configuration: 000000: ATIM1 CC4 event 000001: ATIM1 TRGO event 000010: ATIM1 TRGO2 event 000011: ATIM2 CC4 event 000100: ATIM2 TRGO event 000101: ATIM2 TRGO2 event 000110: ATIM3 CC4 event 000111: ATIM3 TRGO event 001000: ATIM3 TRGO2 event 001001: GTIM1 CC4 event 001010: GTIM2 CC4 event 001011: GTIM3 CC4 event 001100: GTIM4 CC4 event 001101: GTIM5 CC4 event 001110: GTIM6 CC4 event 001111: GTIM7 CC4 event 010000: GTIM8 CC4 event 010001: GTIM9 CC4 event 010010: GTIM10 CC4 event 010011: GTIM1 TRGO event 010100: GTIM2 TRGO event 010101: GTIM3 TRGO event 010110: GTIM4 TRGO event 010111: GTIM5 TRGO event 011000: GTIM6 TRGO event 011001: GTIM7 TRGO event 011010: GTIM8 TRGO event 011011: GTIM9 TRGO event 011100: GTIM10 TRGO event

Bit Field	Name	Description
		011101: SHRTIM TRG1 event 011110: SHRTIM TRG2 event 011111: SHRTIM TRG3 event 100000: SHRTIM TRG4 event 100001: SHRTIM TRG5 event 100010: SHRTIM TRG6 event 100011: SHRTIM TRG7 event 100100: SHRTIM TRG8 event 100101: SHRTIM TRG9 event 100110: SHRTIM TRG10 event 100111: LPTIM1 OUT event 101000: LPTIM2 OUT event 101001: EXTI line 0~15 event 101010: SWSTRRCH
15:10	EXTJSEL[5:0]	External event select for injected group These bits select the external event used to trigger the injection sequence conversion. ADC trigger configuration: 000000: ATIM1 CC4 event 000001: ATIM1 TRGO event 000010: ATIM1 TRGO2 event 000011: ATIM2 CC4 event 000100: ATIM2 TRGO event 000101: ATIM2 TRGO2 event 000110: ATIM3 CC4 event 000111: ATIM3 TRGO event 001000: ATIM3 TRGO2 event 001001: GTIM1 CC4 event 001010: GTIM2 CC4 event 001011: GTIM3 CC4 event 001100: GTIM4 CC4 event 001101: GTIM5 CC4 event 001110: GTIM6 CC4 event 001111: GTIM7 CC4 event 010000: GTIM8 CC4 event 010001: GTIM9 CC4 event 010010: GTIM10 CC4 event 010011: GTIM1 TRGO event 010100: GTIM2 TRGO event 010101: GTIM3 TRGO event 010110: GTIM4 TRGO event 010111: GTIM5 TRGO event 011000: GTIM6 TRGO event

Bit Field	Name	Description
		011001: GTIM7 TRGO event 011010: GTIM8 TRGO event 011011: GTIM9 TRGO event 011100: GTIM10 TRGO event 011101: SHRTIM TRG1 event 011110: SHRTIM TRG2 event 011111: SHRTIM TRG3 event 100000: SHRTIM TRG4 event 100001: SHRTIM TRG5 event 100010: SHRTIM TRG6 event 100011: SHRTIM TRG7 event 100100: SHRTIM TRG8 event 100101: SHRTIM TRG9 event 100110: SHRTIM TRG10 event 100111: LPTIM1 OUT event 101000: LPTIM2 OUT event 101001: EXTI line 0~15 event 101010: SWSTRJCH
9:8	Reserved	Reserved, the reset value must be maintained.
7:6	EXTPJSEL[1:0]	External trigger enable and polarity selection for injected channels 00： External trigger detection disabled (the user can only trigger ADC conversion through software) 01: Detection of rising edge of external trigger as valid trigger 10: Detection of falling edge of external trigger as valid trigger 11: Detection of both rising and falling edges of external trigger as valid trigger
5:4	EXTPRSEL[1:0]	External trigger enable and polarity selection for regular channels 00: External trigger detection disabled (user can only trigger ADC conversion through software) 01: Detection of rising edge of external trigger as a valid trigger 10: Detection of falling edge of external trigger as a valid trigger 11: Detection of both rising and falling edges of the external trigger as valid triggers
3	RSTCAL	Reset the calibration registers This bit is set by software to reset the calibration register. After the calibration register is reset, it is cleared by hardware. 0: Calibration register maintained; 1: Calibration register reset. *Note: When this bit is set while an ADC conversion is in progress, the calibration register will be reset in the next conversion cycle.*
2	ENCAL	A/D Calibration This bit is set by software to start calibration and is cleared by hardware when calibration is completed. 0: Calibration completed; 1: Start calibration.

Bit Field	Name	Description
1	CTU	Continuous conversion This bit is set and cleared by software. If this bit is set, the conversion will continue until the bit is cleared. 0: Single conversion mode; 1: Continuous conversion mode.
0	ON	A/D converter ON/OFF This bit is set and cleared by software. When this bit is '0', writing '1' will wake up the ADC from power-down mode, enabling the ADC. When RDY is '1', it indicates that the ADC conversion is ready. 0: Turn off ADC conversion/calibration and enter power-down mode; 1: Start ADC.

17.12.5 ADC Control Register 3 (ADC_CTRL3)

Offset address：0x0C

Reset value：0x0000 0003

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved									INP2SEL	INN1SEL	INP1SEL	OSRMD	OSRTRIG	OSJE	OSRE
									rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OSS[3:0]				OSR[3:0]				VBAT MEN	DPWMOD	BPCAL	CKMOD	CALALD	CALDIF	RES[1:0]	
rw				rw				rw	rw	rw	rw	rw	rw	rw	

Bit Field	Name	Description
31:23	Reserved	Reserved, the reset value must be maintained.
22	INP2SEL	ADC channel 2 positive end select 0: ADC channel 2 positive terminal connected to external channel IO 1: ADC channel 2 positive terminal connected to internal PGAx negative terminal output
21	INN1SEL	ADC channel 1 negtive end select 0: ADC channel 1 negative terminal connected to external channel IO 1: ADC channel 1 negative terminal connected to internal PGAx negative terminal output.
20	INP1SEL	ADC channel 1 positive end select 0: ADC channel 1 positive terminal connected to external channel IO. 1: ADC channel 1 positive terminal connected to internal PGAx positive terminal output.
19	OSRMD	Regular channels oversample mode 0: Continuous mode. When the injected oversampling is triggered, oversampling temporarily stops. When triggered again, oversampling resumes. 1: Reset mode. When the injected oversampling is triggered, oversampling stops. When triggered again, oversampling restarts.
18	OSRTRIG	Regular channels oversample triagger 0: All oversampling conversions only require one trigger

Bit Field	Name	Description
		1: Each oversampling conversion requires one trigger each time
17	OSJE	Injection channels oversample enable 0: Disable 1: Enable
16	OSRE	Regular channels oversample enable 0: Disable 1: Enable
15:12	OSS[3:0]	Oversample data right shift bit width 0000: right shift 0 bit 0001: right shift 1 bit 0010: right shift 2 bit 0011: right shift 3 bit 0100: right shift 4 bit 0101: right shift 5 bit 0110: right shift 6 bit 0111: right shift 7 bit 1000: right shift 8 bit Other: right shift 0 bit
11:8	OSR[3:0]	Oversample rate times configuration 0000: The number of oversampling is 1 0001: The number of oversampling is 2 0010: The number of oversampling is 4 0011: The number of oversampling is 8 0100: The number of oversampling is 16 0101: The number of oversampling is 32 0110: The number of oversampling is 64 0111: The number of oversampling is 128 1000: The number of oversampling is 256 Other: The number of oversampling is 1
7	VBATMEN	Vbat monitor enable 0: Disable 1: Enable
6	DPWMOD	Deep Power Mode 0: When the ADC is not enabled, it enters PowerDown mode; 1: When the ADC is not enabled, it enters deep sleep mode.
5	BPCAL	Bypass calibration 0: Disable; 1: Enable.
4	CKMOD	Clock Mode 0: Select AHB as the synchronous clock; 1: PLL Select PLL as the asynchronous clock.
3	CALALD	calibration auto load

Bit Field	Name	Description
		0: Disable; 1: Enable.
2	CALDIF	Differential mode for calibration This bit is set and cleared by software to configure the calibration in single-ended or differential input mode. 0: Writing to the ENCAL bit will initiate calibration in single-ended input mode; 1: Writing to the ENCAL bit will initiate calibration in differential input mode.

1:0	RES[1:0]	Data resolution
		This bit is set and cleared by software to select the resolution of the conversion.
		00: 6 bit;
		01: 8 bit;
		10: 10 bit;
		11: 12 bit.

17.12.6 ADC Sampling Time Register 1 (ADC_SAMPT1)

Offset address：0x10

Reset value：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SAMP7[3:0]				SAMP6[3:0]				SAMP5[3:0]				SAMP4[3:0]			
rw				rw				rw				rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SAMP3[3:0]				SAMP2[3:0]				SAMP1[3:0]				SAMP0[3:0]			
rw				rw				rw				rw			

Bit Field	Name	Description
31:0	SAMPx[3:0]	Channel Sample time selection These bits are used to independently select the sampling time for each channel. The channel selection bits must remain unchanged during sampling. 0000: 1.5 cycle 0001: 2.5 cycle 0010: 4.5 cycle 0011: 7.5 cycle 0100: 13.5 cycle 0101: 19.5 cycle 0110: 28.5 cycle 0111: 41.5 cycle 1000: 55.5 cycle 1001: 61.5 cycle 1010: 71.5 cycle 1011: 181.5 cycle

Bit Field	Name	Description
		1100: 239.5 cycle 1101: 601.5 cycle

17.12.7 ADC Sampling Time Register 2 (ADC_SAMPT2)

Offset address：0x14

Reset value：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SAMP15[3:0]				SAMP14[3:0]				SAMP13[3:0]				SAMP12[3:0]			
rw				rw				rw				rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SAMP11[3:0]				SAMP10[3:0]				SAMP9[3:0]				SAMP8[3:0]			
rw				rw				rw				rw			

Bit Field	Name	Description
31:0	SAMPx[3:0]	Channel Sample time selection These bits are used to independently select the sampling time for each channel. The channel selection bits must remain unchanged during sampling. 00000: 1.5 cycle 00001: 2.5 cycle 00010: 4.5 cycle 00011: 7.5 cycle 00100: 13.5 cycle 00101: 19.5 cycle 00110: 28.5 cycle 00111: 41.5 cycle 01000: 55.5 cycle 01001: 61.5 cycle 01010: 71.5 cycle 01011: 181.5 cycle 01100: 239.5 cycle 01101: 601.5 cycle

17.12.8 ADC Sampling Time Register 3 (ADC_SAMPT3)

Offset address：0x18

Reset value：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				SAMP18[3:0]				SAMP17[3:0]				SAMP16[3:0]			
				rw				rw				rw			

Bit Field	Name	Description
31:0	SAMPx[3:0]	Channel Sample time selection These bits are used to independently select the sampling time for each channel. The channel selection bits must remain unchanged during sampling. 00000: 1.5 cycle 00001: 2.5 cycle 00010: 4.5 cycle 00011: 7.5 cycle 00100: 13.5 cycle 00101: 19.5 cycle 00110: 28.5 cycle 00111: 41.5 cycle 01000: 55.5 cycle 01001: 61.5 cycle 01010: 71.5 cycle 01011: 181.5 cycle 01100: 239.5 cycle 01101: 601.5 cycle

17.12.9 ADC Differential Mode Selection Register (ADC_DIFSEL)

Offset address：0x1C

Reset value：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved													DIFSEL[17:0]		
													rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DIFSEL[17:0]															Reserved
rw															

Bit Field	Name	Description
31:19	Reserved	Reserved, the reset value must be maintained

18:1	DIFSEL[17:0]	Differential mode for channels 18 to 1
		DIFSEL[i] = 0: Configure ADC channel input i+1 in single-ended mode;
		DIFSEL[i] = 1: Configure ADC channel input i+1 in differential mode.

0	Reserved	Reserved, the reset value must be maintained

17.12.10 ADC Calibration Factor（ADC_CALFACT）

Offset address：0x20

Reset value：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved									CALFACTD[6:0]						
									rw						

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									CALFACTS[6:0]						
									rw						

Bit Field	Name	Description
31:23	Reserved	Reserved, the reset value must be maintained
22:16	CALFACTD[6:0]	Calibration Factors in differential mode This bit can be written by hardware or software After the differential input calibration is complete, the hardware will update it according to the calibration coefficient. Software can write these bits with a new calibration factor. If the new calibration coefficient is different from the current coefficient stored in the analog ADC, the coefficient will be applied after a new differential calibration is initiated. *Note: software allows write only when ADC_CTRL2.ON=1, ADC_STS.STR =0, ADC_STS.JSTR =0 (ADC does not process conversion or start conversion)*
15:7	Reserved	Reserved, the reset value must be maintained
6:0	CALFACTS[6:0]	Calibration Factors in Single-Ended mode This bit can be written by hardware or software After the single-end input calibration is completed, the hardware will update it according to the calibration coefficient. Software can write these bits with a new calibration factor. If the new calibration coefficient is different from the current coefficient stored in the analog ADC, the coefficient will be applied after a new single-ended calibration is initiated. *Note: software allows write only when ADC_CTRL2.ON=1, ADC_STS.STR =0, ADC_STS.JSTR =0　(ADC does not process conversion or start conversion)*

17.12.11 ADC Data Offset Register x(ADC_OFFSETx)(x=1..4)

Offset address：0x24-0x30

Reset value：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OFFSCHx EN	OFFSCHxCH[4:0]					OFFSCHx SATEN	OFFSCHx DIR	Reserved							
rw	rw					rw	rw								

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				OFFSCHxDAT[11:0]											
				rw											

Bit Field	Name	Descriptions
31	OFFSCHxEN	Offset channel x enable This bit can be written by software to enable offset compensation function. 0: Disable Offset Compensation Function 1: Enable Offset Compensation Function *Note: This bit should not be operated during ADC conversion*

30:26	OFFSCHxCH[4:0]	Offset channel x selection
		This bit can be written by software to select which channel to enable offset compensation.
		00000: ADC analog input channel 0;
		00001: ADC analog input channel 1;
		……
		10000: ADC analog input channel 16;
		10001: ADC analog input channel 17;
		10010: ADC analog input channel 18;

25	OFFSCHxSATEN	Offset Saturation enable This bit can be written or cleared by software to enable saturation to 0x0000 and 0xFFF. 0: Without saturation control, the offset data may be signed 1: Enabling saturation enables the offset data to be unsigned, with a range of 0x0000-0x0FFF
24	OFFSCHxDIR	Offset direction 0: Negative offset 1: Positive offset
23:12	Reserved	Reserved, the reset value must be maintained
11:0	OFFSCHxDAT[11:0]	Data offset for channel x These bits are defined to subtract from or add to the original conversion data when converting channel x (regular channel or injected channel), depending on the value of OFFSCHxDIR. The conversion result can be read in the ADC_DAT or ADC_JDATx register. If multiple offset channels are configured for the same channel, the offset channel 1 has the highest priority, followed by offset channel 2, offset channel 3, and offset channel 4. For example, if OFFSCH1CH[4:0] = 8, OFFSCH2CH[4:0] = 8, then only OFFSCH1DAT[11:0] will be used for calculations when converting channel 8.

17.12.12 ADC Watchdog 1 High Threshold Register (ADC_AWD1HIGH)

Offset address: 0x34

Reset value: 0x0000 0FFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

Reserved

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				HLH1[11:0]											
				rw											

Bit Field	Name	Description
31:12	Reserved	Reserved, the reset value must be maintained
11:0	HTH[11:0]	Analog watchdog 1 high threshold These bits define the high thresholds for analog watchdog 1.

17.12.13 ADC Watchdog 1 Low Threshold Rgister (ADC_AWD1LOW)

Offset address: 0x38

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	AWDFIL[2:0]			LTH1[11:0]											
	rw			rw											

Bit Field	Name	Description
31:15	Reserved	Reserved, the reset value must be maintained
14:12	AWDFIL[2:0]	Analog watchdog 1 filter value, can only act on a single channel of the watchdog, this bit can be written or cleared by software. 000: Do not enable filtering. 001: Two consecutive detection results exceeding the threshold will generate an AWDx flag or interrupt. 010: Three consecutive detection results exceeding the threshold will generate an AWDx flag or interrupt. … 111: Eight consecutive detection results exceeding the threshold will generate an AWDx flag or interrupt.
11:0	LTH[11:0]	Analog watchdog 1 low threshold These bits define the low thresholds for analog watchdog 1.

17.12.14 ADC Watchdog 2 High Threshold Register (ADC_AWD2HIGH)

Offset address: 0x3C

Reset value: 0x0000 0FFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				HTH2[11:0]											
				rw											

Bit Field	Name	Description
31:12	Reserved	Reserved, the reset value must be maintained
11:0	HTH[11:0]	Analog watchdog 2 high threshold These bits define the high thresholds for analog watchdog 2.

17.12.15 ADC Watchdog 2 Low Threshold Rgister (ADC_AWD2LOW)

Offset address: 0x40

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				LTH2[11:0]											
				rw											

Bit Field	Name	Description
31:12	Reserved	Reserved, the reset value must be maintained
11:0	LTH[11:0]	Analog watchdog 2 low threshold These bits define the low thresholds for analog watchdog 2.

17.12.16 ADC Watchdog 3 High Threshold Register (ADC_AWD3HIGH)

Offset address: 0x44

Reset value: 0x0000 0FFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				HTH3[11:0]											
				rw											

Bit Field	Name	Description
31:12	Reserved	Reserved, the reset value must be maintained
11:0	HTH [11:0]	Analog watchdog 3 high threshold These bits define the high thresholds for analog watchdog 3.

17.12.17 ADC Watchdog 3 Low Threshold Rgister (ADC_AWD3LOW)

Offset address: 0x48

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				LTH3[11:0]											
				rw											

Bit Field	Name	Description
31:12	Reserved	Reserved, the reset value must be maintained
11:0	LTH[11:0]	Analog watchdog 3 low threshold These bits define the high thresholds for analog watchdog 3.

17.12.18 ADC Analog Watchdog 2 Configuration Register（ADC_AWD2EN）

Offset address: 0x4C

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved													AWD2EN[18:16]		
													rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AWD2EN[15:0]															
rw															

Bit Field	Name	Description
31:19	Reserved	Reserved, the reset value must be maintained
18:0	AWD2EN[18:0]	Analog watchdog 2 channel selection These bits define the channels monitored by Analog Watchdog 2, and can be written or cleared by software. AWD2EN[x] = 0: Analog Watchdog 2 does not monitor ADC analog input channel x; AWD2EN[x] = 1: Analog Watchdog 2 monitors ADC analog input channel x;

Bit Field	Name	Description
		AWD2EN[18:0] = 0b000..00, Analog Watchdog 2 is not operational.

17.12.19 ADC Analog Watchdog 3 Configuration Register（ADC_AWD3EN）

Offset address: 0x50

Reset value: 0x0000 0000

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="13">Reserved</td><td colspan="3">AWD3EN[18:16]</td></tr>
<tr><td colspan="13"></td><td colspan="3">rw</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="16">AWD3EN[15:0]</td></tr>
<tr><td colspan="16">rw</td></tr>
</table>

Bit Field	Name	Description
31:19	Reserved	Reserved, the reset value must be maintained
18:0	AWD3EN[18:0]	Analog watchdog 3 channel selection These bits define the channels monitored by Analog Watchdog 2, and can be written or cleared by software. AWD3EN[x] = 0: Analog Watchdog 3 does not monitor ADC analog input channel x AWD3EN[x] = 1: Analog Watchdog 3 monitors ADC analog input channel x AWD3EN[18:0] = 0b000..00, Analog Watchdog 3 is not operational.

17.12.20 ADC Analog Watchdog 2 Interrupt Enable Register（ADC_AWD2INTEN）

Offset address: 0x54

Reset value: 0x0000 0000

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="13">Reserved</td><td colspan="3">AWD2INTEN [18:16]</td></tr>
<tr><td colspan="13"></td><td colspan="3">rw</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="16">AWD2INTEN[15:0]</td></tr>
<tr><td colspan="16">rw</td></tr>
</table>

Bit Field	Name	Description
31:19	Reserved	Reserved, the reset value must be maintained

Bit Field	Name	Description
18:0	AWD2INTEN[18:0]	Analog watchdog 2 interrupt enable These bits define whether the interrupt for the channels monitored by Analog Watchdog 2 is enabled, and can be written or cleared by software. AWD2INTEN [x] = 0: The interrupt for the channel x monitored by Analog Watchdog 2 is not enabled; AWD2INTEN [x] = 1: The interrupt for the channel x monitored by Analog Watchdog 2 is enabled;

17.12.21 ADC Analog Watchdog 3 Interrupt Enable Register（ADC_AWD3INTEN）

Offset address: 0x58

Reset value: 0x0000 0000

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="13">Reserved</td><td colspan="3">AWD3INTEN [18:16]</td></tr>
<tr><td colspan="13"></td><td colspan="3">rw</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="16">AWD3INTEN[15:0]</td></tr>
<tr><td colspan="16">rw</td></tr>
</table>

Bit Field	Name	Description
31:19	Reserved	Reserved, the reset value must be maintained
18:0	AWD3INTEN[18:0]	Analog watchdog 3 interrupt enable These bits define whether the interrupt for the channels monitored by Analog Watchdog 3 is enabled, and can be written or cleared by software. AWD3INTEN [x] = 0: The interrupt for the channel x monitored by Analog Watchdog 3 is not enabled; AWD3INTEN [x] = 1: The interrupt for the channel x monitored by Analog Watchdog 3 is enabled.

17.12.22 ADC Analog Watchdog 2 Status Register（ADC_AWD2STS）

Offset address: 0x5C

Reset value: 0x0000 0000

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="13">Reserved</td><td colspan="3">AWD2FLAG[18:16]</td></tr>
<tr><td colspan="13"></td><td colspan="3">rc-w1</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="16">AWD2FLAG[15:0]</td></tr>
<tr><td colspan="16">rc-w1</td></tr>
</table>

Bit Field	Name	Description
31:19	Reserved	Reserved, the reset value must be maintained
18:0	AWD2FLAG[18:0]	Analog watchdog 2 status flag AWD2FLAG[x] = 0: Indicates that the voltage value of the channel x conversion monitored by Analog Watchdog 2 has not exceeded the range defined by the ADC_AWD2HIGH.HTH2[11:0] and ADC_AWD2LOW.LTH2[11:0] registers； AWD2FLAG[x] = 1: Indicates that when the voltage value of the channel x conversion monitored by Analog Watchdog 2 exceeds the range defined by the ADC_AWD2HIGH.HTH2[11:0] and ADC_AWD2LOW.LTH2[11:0] registers, it is set to 1 by hardware, and writing 1 clears it by software.

17.12.23 ADC Analog Watchdog 3 Status Register（ADC_AWD3STS）

Offset address: 0x60

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved													AWD3FLAG[18:16]		
													rc-w1		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AWD3FLAG[15:0]															
rc-w1															

Bit Field	Name	Description
31:19	Reserved	Reserved, the reset value must be maintained
18:0	AWD3FLAG[18:0]	Analog watchdog 3 status flag AWD3FLAG[x] = 0: Indicates that the voltage value of the channel x conversion monitored by Analog Watchdog 3 has not exceeded the range defined by the ADC_AWD3HIGH.HTH3[11:0] and ADC_AWD3LOW.LTH3[11:0] registers; AWD3FLAG[x] = 1: Indicates that when the voltage value of the channel x conversion monitored by Analog Watchdog 3 exceeds the range defined by the ADC_AWD3HIGH.HTH3[11:0] and ADC_AWD3LOW.LTH3[11:0] registers, it is set to 1 by hardware, and writing 1 clears it by software.

17.12.24 ADC Regular Sequence Register 1（ADC_RSEQ1）

Offset address: 0x64

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		SEQ6[4:0]					SEQ5[4:0]					SEQ4[4:1]			
		rw					rw					rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SEQ4[0]	SEQ3[4:0]					SEQ2[4:0]					SEQ1[4:0]				
rw	rw					rw					rw				

Bit Field	Name	Description
31:30	Reserved	Reserved, the reset value must be maintained
29:25	SEQ6[4:0]	6th conversion in regular sequence These bits are defined by software to specify the number (0-18) of the 6th conversion channel in the conversion sequence.
24:20	SEQ5[4:0]	5th conversion in regular sequence These bits are defined by software to specify the number (0-18) of the 5th conversion channel in the conversion sequence.
19:15	SEQ4[4:0]	4th conversion in regular sequence These bits are defined by software to specify the number (0-18) of the 4th conversion channel in the conversion sequence.
14:10	SEQ3[4:0]	3th conversion in regular sequence
9:5	SEQ2[4:0]	2nd conversion in regular sequence
4:0	SEQ1[4:0]	1st conversion in regular sequence

17.12.25 ADC Regular Sequence Register 2（ADC_RSEQ2）

Offset address：0x68

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		SEQ12[4:0]					SEQ11[4:0]					SEQ10[4:1]			
		rw					rw					rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REQ10[0]	SEQ9[4:0]					SEQ8[4:0]					SEQ7[4:0]				
rw	rw					rw					rw				

Bit Field	Name	Description
31:30	Reserved	Reserved, the reset value must be maintained
29:25	SEQ12[4:0]	12th conversion in regular sequence These bits are defined by software to specify the number (0-18) of the 12th conversion channel in the conversion sequence.
24:20	SEQ11[4:0]	11th conversion in regular sequence These bits are defined by software to specify the number (0-18) of the 11th conversion channel in the conversion sequence.
19:15	SEQ10[4:0]	10th conversion in regular sequence These bits are defined by software to specify the number (0-18) of the 10th conversion channel in the conversion sequence.
14:10	SEQ9[4:0]	9th conversion in regular sequence

Bit Field	Name	Description
9:5	SEQ8[4:0]	8th conversion in regular sequence
4:0	SEQ7[4:0]	7th conversion in regular sequence

17.12.26 ADC Regular Sequence Register 3（ADC_RSEQ3）

Offset address: 0x6C

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved			LEN[3:0]				Reserved					SEQ16[4:1]			
			rw				rw					rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SEQ16[0]	SEQ15[4:0]					SEQ14[4:0]					SEQ13[4:0]				
rw	rw					rw					rw				

Bit Field	Name	Description
31:29	Reserved	Reserved, the reset value must be maintained
28:25	LEN[3:0]	Regular channel sequence length These bits are defined by software to specify the number of channels in the regular sequence conversion. 0000: 1 conversion; 0001: 2 conversion; ... 1111: 16 conversion.
24:20	Reserved	Reserved, the reset value must be maintained
19:15	SEQ16[4:0]	16th conversion in regular sequence These bits are defined by software to specify the number (0-18) of the 16th conversion channel in the conversion sequence.
14:10	SEQ15[4:0]	15th conversion in regular sequence
9:5	SEQ14[4:0]	14th conversion in regular sequence
4:0	SEQ13[4:0]	13th conversion in regular sequence

17.12.27 ADC Injected Sequence Register（ADC_JSEQ）

Offset address: 0x70

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved					JLEN[1:0]		Reserved					JSEQ4[4:1]			
					rw							rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
JSEQ4[0]	JSEQ3[4:0]					JSEQ2[4:0]					JSEQ1[4:0]				
rw	rw					rw					rw				

Bit Field	Name	Description
31:27	Reserved	Reserved, the reset value must be maintained

26:25	JLEN[1:0]	Injected sequence length
		These bits are software-defined as the number of channels in the injected channel conversion sequence.
		00: 1 conversion;
		01: 2 conversions;
		10: 3 conversions;
		11: 4 conversions.

24:20	Reserved	Reserved, the reset value must be maintained
19:15	JSEQ4[4:0]	4th conversion in injected sequence These bits are defined by software as the number (0-18) of the 4th conversion channel in the conversion sequence. *Note: Different from regular conversion sequences, if the length of ADC_JSEQ.JLEN[1:0] is less than 4, the sequence of conversion starts from (4-JLEN). For example, ADC_JSEQ[21:0] = 10 00011 00011 00111 00010 means that the scan conversion will be converted in the following channel order: 7, 3, 3 instead of 2, 7, 3.*
14:10	JSEQ3[4:0]	3rd conversion in injected sequence
9:5	JSEQ2[4:0]	2nd conversion in injected sequence
4:0	JSEQ1[4:0]	1st conversion in injected sequence

17.12.28 ADC Injection Data Register x (ADC_JDATx)(x = 1..4)

Offset address: 0x74 – 0x80

Reset value：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
JDATx[15:0]															
r															

Bit Field	Name	Description
31:16	Reserved	Reserved, the reset value must be maintained
15:0	JDAT[15:0]	Injected data These bits are read-only and contain the conversion results of the injected channel. The data is left-

Bit Field	Name	Description
		aligned or right-aligned

17.12.29 ADC Regulars Data Register (ADC_DAT)

Offset address: 0x84

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADC2DAT[15:0]															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DAT[15:0]															
r															

Bit Field	Name	Description
31:16	DAT2[15:0]	2nd data item of a pair of regular conversions —In dual mode, these bits contain data from the regular ADC。 —In triple mode, these bits can also contain the regular data from ADC1, ADC2, and ADC3. These bits are read-only and contain the conversion results of the regular channels. Data can be left-aligned or right-aligned.
15:0	DAT1[15:0]	1st data item of a pair of regular conversions —In dual mode, these bits contain the regular data from the master ADC. —In triple mode, these bits can also contain the regular data from ADC1, ADC2, and ADC3. These bits are read-only and contain the conversion results of the regular channels. The data can be left-aligned or right-aligned.

17.12.30 ADC FIFO Configuration Register (ADC_FIFOCFG)

Offset address: 0x88

Reset value: 0x0000 0180

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				NE INTEN	CLR	WL[3:0]				EN	HF INTEN	EINTEN	FINTEN	WEINT EN	REINT EN
				rw	rw	rw				rw	rw	rw	rw	rw	rw

Bit Field	Name	Description
31:12	Reserved	Reserved, the reset value must be maintained
11	NEINTEN	FIFO non-empty interrupt enable 0: Disable 1: Enable
10	CLR	FIFO clear signal 0: Disable 1: Enable

9:6	WL[3:0]	FIFO water level
		0000: Reserved;
		0001: FIFO water level is 1;
		0010: FIFO water level is 2;
		0011: FIFO water level is 3;
		0100: FIFO water level is 4;
		0101: FIFO water level is 5;
		0110: FIFO water level is 6;
		0111: FIFO water level is 7;
		1000: FIFO water level is 8;
		1001: FIFO water level is 9;
		1010: FIFO water level is 10;
		Other: Reserved

5	EN	FIFO enable 0: Disable 1: Enable
4	HFINTEN	FIFO half-full interrupt enable 0: Disable 1: Enable
3	EINTEN	FIFO empty interrupt enable 0: Disable 1: Enable
2	FINTEN	FIFO full interrupt enable 0: Disable 1: Enable
1	WEINTEN	FIFO write err interrupt enable 0: Disable 1: Enable
0	REINTEN	FIFO read err interrupt enable 0: Disable 1: Enable

17.12.31 ADC FIFO Status Register (ADC_FIFOSTS)

Offset address: 0x8C

Reset value: 0x0000 0008

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				NE FLAG	Reserved	DATCNT[4:0]					HF FLAG	EFLAG	FFLAG	WE FLAG	RE FLAG
				rc_w1		r					rc_w1	rc_w1	rc_w1	rc_w1	rc_w1

Bit Field	Name	Description
31:12	Reserved	Reserved, the reset value must be maintained
11	NEFLAG	FIFO non-empty status It is set to 1 by hardware, and writing 1 clears it by software
10	Reserved	Reserved, the reset value must be maintained

9:5	DATCNT[4:0]	FIFO valid data count
		00000: Valid data is 0;
		00001: Valid data is 1;
		00010: Valid data is 2;
		00011: Valid data is 3;
		00100: Valid data is 4;
		00101: Valid data is 5;
		00110: Valid data is 6;
		00111: Valid data is 7;
		01000: Valid data is 8;
		01001: Valid data is 9;
		01010: Valid data is 10;
		01011: Valid data is 11;
		01100: Valid data is 12;
		Other: Reserved

4	HFFLAG	FIFO half-full status It is set to 1 by hardware, and writing 1 clears it by software.
3	EFLAG	FIFO empty status It is set to 1 by hardware, and writing 1 clears it by software.
2	FFLAG	FIFO full status It is set to 1 by hardware, and writing 1 clears it by software.
1	WEFLAG	FIFO write error status It is set to 1 by hardware, and writing 1 clears it by software.
0	REFLAG	FIFO read error status

Bit Field	Name	Description
		It is set to 1 by hardware, and writing 1 clears it by software.

17.12.32 ADC Power Control Register（ADC_PCTRL）

Offset address: 0x90

Reset value: 0x0000 0008

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										EXPEN	RANGE SEL	VREF LDOEN	Reserved		
										rw	rw	rw			

Bit Field	Name	Description
31:6	Reserved	Reserved, the reset value must be maintained.
5	EXPEN	0: Disable external power supply and use internal LDO to power the ADC; 1: Enable external power supply.
4	RANGESEL	0: The voltage range of the LDO is 0.81 ~ 0.99V; 1: The voltage range of the LDO is 0.99 ~ 1.21V.
3	VREFLDOEN	0: The reference voltage of the LDO is DVDD; 1: The reference voltage of the LDO is VREF_LDO.
2:0	Reserved	Reserved, the reset value must be maintained.

17.12.33 ADC Control Register 4（ADC_CTRL4）

Offset address: 0x94

Reset value: 0x002E 3DE1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								EXTRISEL[3:0]				EXTRRSEL[3:0]			
								rw				rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															

Bit Field	Name	Description
31:24	Reserved	Reserved, the reset value must be maintained.

Bit Field	Name	Description
23:20	EXTRISEL[3:0]	Select the interrupt line as the external trigger signal source for the injection channel: 0000: EXTI Line 0 0001: EXTI Line 1 0010: EXTI Line 2 0011: EXTI Line 3 0100: EXTI Line 4 0101: EXTI Line 5 0110: EXTI Line 6 0111: EXTI Line 7 1000: EXTI Line 8 1001: EXTI Line 9 1010: EXTI Line 10 1011: EXTI Line 11 1100: EXTI Line 12 1101: EXTI Line 13 1110: EXTI Line 14 1111: EXTI Line 15
19:16	EXTRRSEL[3:0]	Select the interrupt line as the external trigger signal source for the regular channel: 0000: EXTI Line 0 0001: EXTI Line 1 0010: EXTI Line 2 0011: EXTI Line 3 0100: EXTI Line 4 0101: EXTI Line 5 0110: EXTI Line 6 0111: EXTI Line 7 1000: EXTI Line 8 1001: EXTI Line 9 1010: EXTI Line 10 1011: EXTI Line 11 1100: EXTI Line 12 1101: EXTI Line 13 1110: EXTI Line 14 1111: EXTI Line 15
15:0	Reserved	Reserved, the reset value must be maintained.

17.12.34 ADC Interrupt Enable Register (ADC_INTEN)

Offset address: 0x98

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved							EOSMP IEN	PDRDY IEN	RDY IEN	ENDCER RIEN	AWD1 IEN	JENDCA IEN	JENDC IEN	ENDCA IEN	ENDC IEN
							rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit Field	Name	Description
31:9	Reserved	Reserved, the reset value must be maintained.
8	EOSMPIEN	ADC sample done interrupt enable 0: Disable ADC conversion complete interrupt; 1: Enable ADC conversion complete interrupt.
7	PDRDYIEN	ADC Power down ready interrupt enable 0: Disable ADC low power mode ready interrupt; 1: Enable ADC low power mode ready interrupt.
6	RDYIEN	ADC Ready interrupt enable 0: Disable ADC ready interrupt; 1: Enable ADC ready interrupt.
5	ENDCERRIEN	ENDC error interrupt enable 0: Disable ENDC error interrupt. 1: Enable ENDC error interrupt.

Bit Field	Name	Description
4	AWD1IEN	Analog watchdog 1 interrupt enable This bit is set and cleared by software to disable or enable the generation of an interrupt by Analog Watchdog 1. In scan mode, if Watchdog 1 detects an out-of-range value, the scan will be aborted only when this bit is set. 0: Disable Analog Watchdog 1 interrupt; 1: Enable Analog Watchdog 1 interrupt.
3	JENDCAIEN	Interrupt enable for any injection channels This bit is set and cleared by software to enable/disable the end of conversion interrupt for the injection channel. 0: ADC_STS.JENDCA interrupt disable; 1: ADC_STS.JENDCA interrupr enable.
2	JENDCIEN	Interrupt enable for injection channels This bit is set and cleared by software to disable or enable the interrupt after all injection channel conversions are completed. 0: Disable ADC_STS.JENDC interrupt; 1: Enable ADC_STS.JENDC interrupt.
1	ENDCAIEN	Interrupt enable for any regular channels This bit is set and cleared by software to enable/disable the end of conversion interrupt for any regular channel. 0: Disable ADC_STS.ENDCA interrupt; 1: Enable ADC_STS.ENDCA interrupt.
0	ENDCIEN	Interrupt enable for ENDC This bit is set and cleared by software to disable or enable the interrupt that occurs after a regular or injected conversion sequence is completed. 0: Disable ADC_STS.ENDC interrupt; 1: Enable ADC_STS.ENDC interrupt.

17.12.35 ADC Gain Compensation Register (ADC_GCOMP)

Offset address: 0x9C

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		GCOMPDAT[13:0]													
		rw													

Bit Field	Name	Description
31:14	Reserved	Reserved, the reset value must be maintained.
13:0	GCOMPDAT[13:0]	Gain compensation coefficient This bit is set and cleared by software to set the gain compensation coefficient. 00 1000 0000 0000: The gain compensation coefficient is 0.5 ... 01 0000 0000 0000: The gain compensation coefficient is 1 10 0000 0000 0000: The gain compensation coefficient is 2 11 0000 0000 0000: The gain compensation coefficient is 3 ... The range of this gain coefficient is 0 to 3.999756, and the gain coefficient is the parameter divided by 4096.

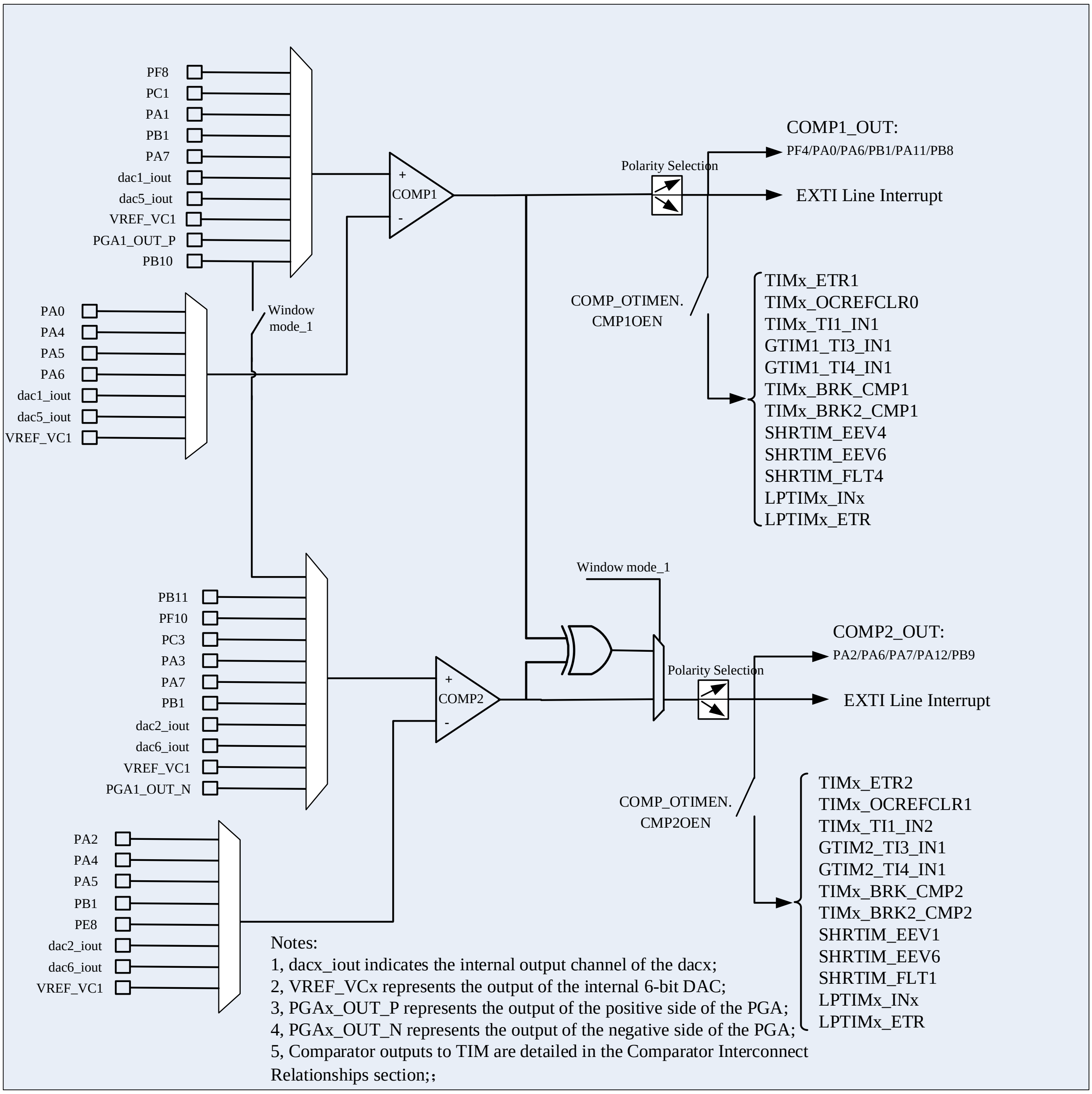
18 Comparator（COMP）

The COMP module is used to compare the analog voltages of two inputs and output high/low levels based on the comparison result. When ‘INP’ input voltage is higher than ‘INM’ input voltage, the comparator outputs are high level, when ‘INP’ input voltage is lower than ‘INM’ input voltage, the comparator outputs are low level.

18.1 COMP System Connection Block Diagram

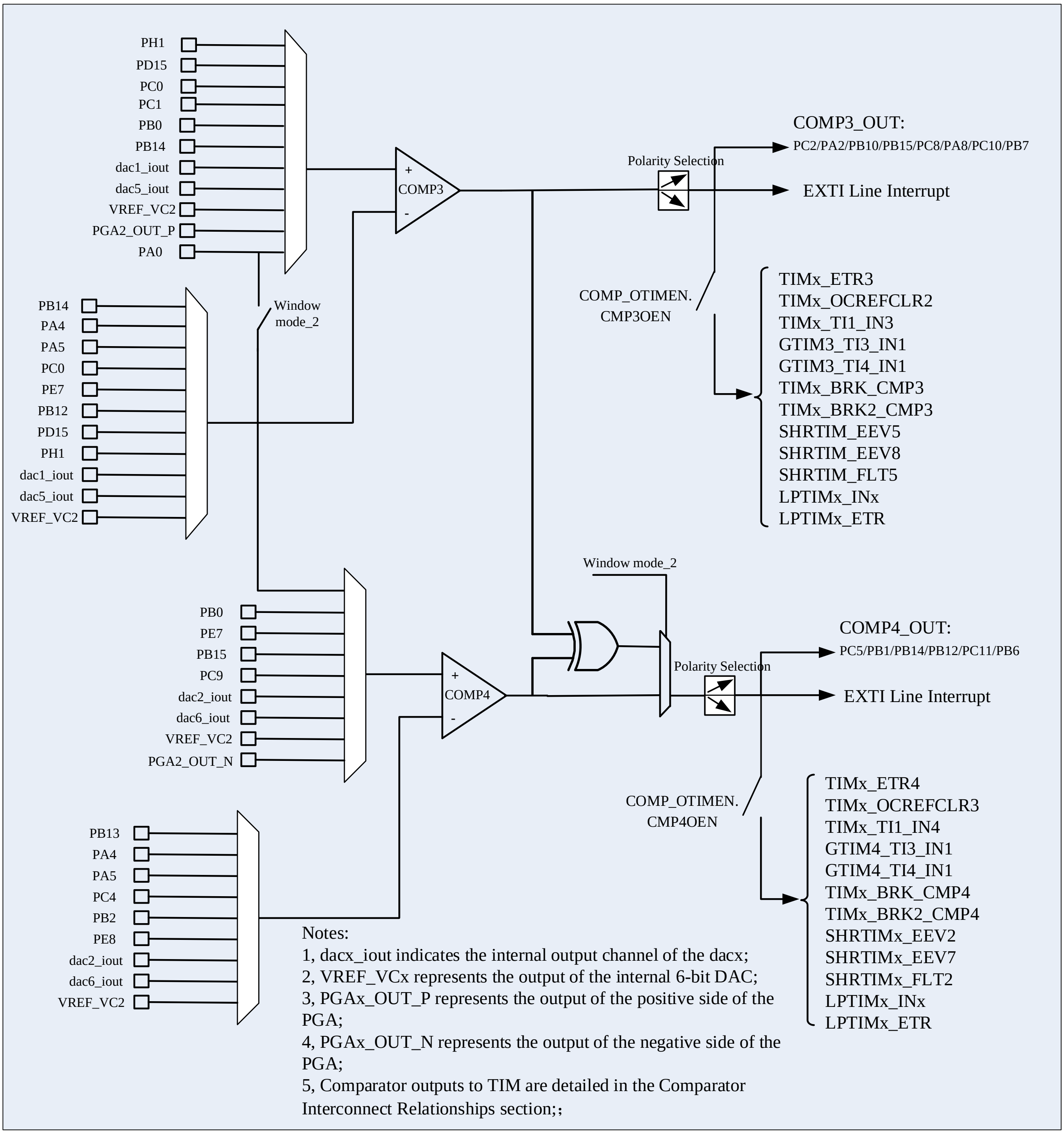
The COMP module supports 7 independent comparators, which are connected to the APB2 bus.

Figure 18-1 Comparator1 And Comparator2 Connection Diagram



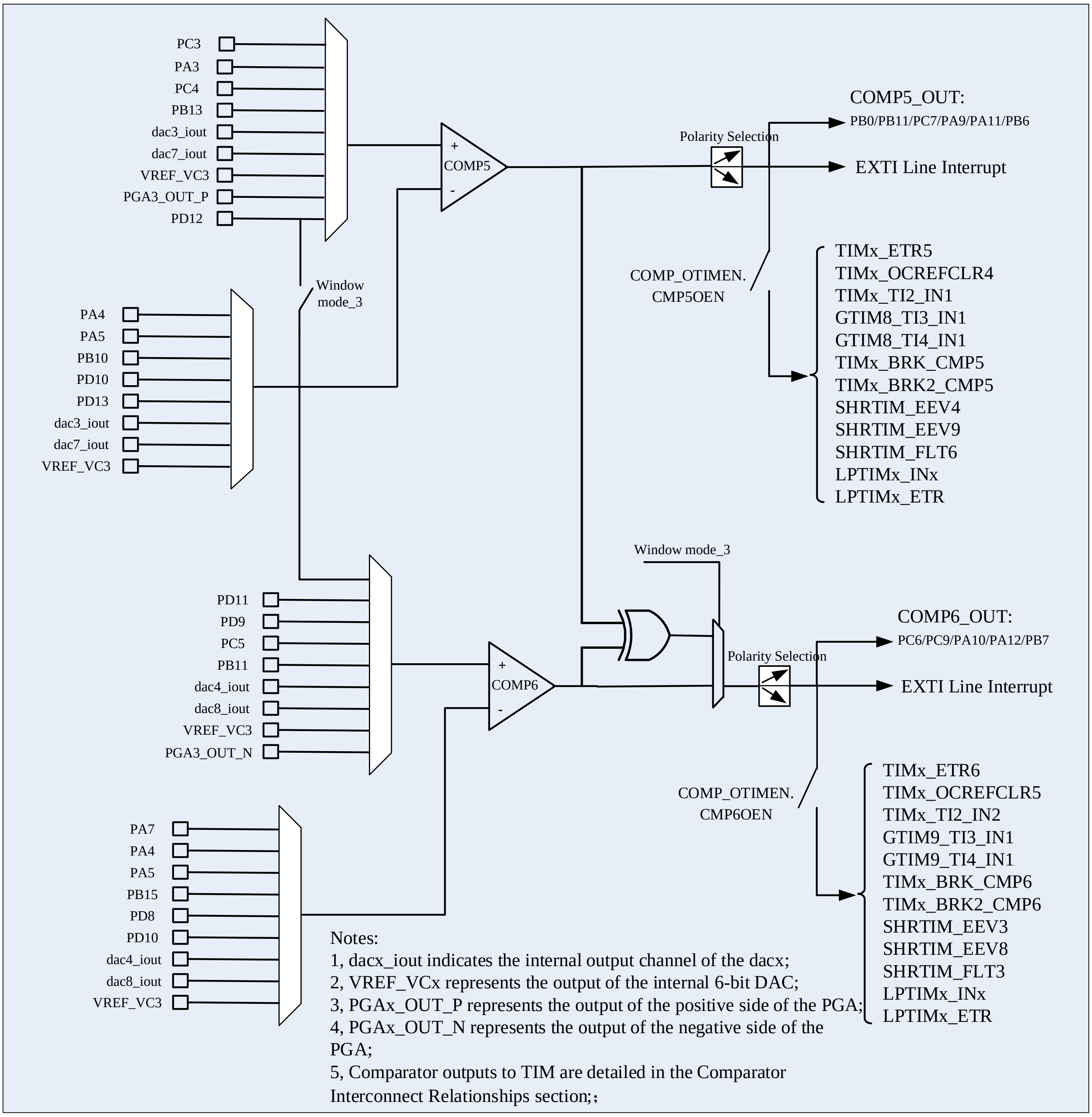
Notes:
1, dacx_iout indicates the internal output channel of the dacx;
2, VREF_VCx represents the output of the internal 6-bit DAC;
3, PGAx_OUT_P represents the output of the positive side of the PGA;
4, PGAx_OUT_N represents the output of the negative side of the PGA;
5, Comparator outputs to TIM are detailed in the Comparator Interconnect Relationships section;；

Figure 18-2 Comparator3 And Comparator4 Connection Diagram



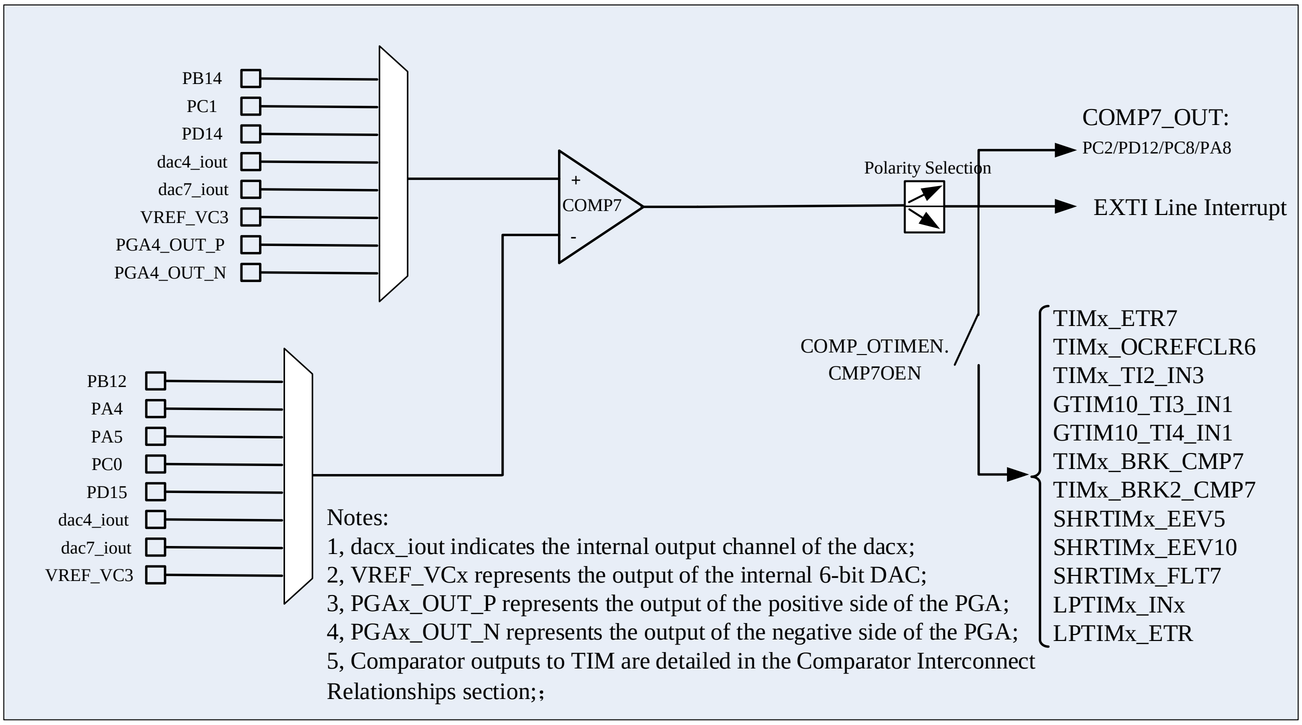
Notes:
1, dacx_iout indicates the internal output channel of the dacx;
2, VREF_VCx represents the output of the internal 6-bit DAC;
3, PGAx_OUT_P represents the output of the positive side of the PGA;
4, PGAx_OUT_N represents the output of the negative side of the PGA;
5, Comparator outputs to TIM are detailed in the Comparator Interconnect Relationships section;;

Figure 18-3 Comparator5 And Comparator6 Connection Diagram



Notes:
1, dacx_iout indicates the internal output channel of the dacx;
2, VREF_VCx represents the output of the internal 6-bit DAC;
3, PGAx_OUT_P represents the output of the positive side of the PGA;
4, PGAx_OUT_N represents the output of the negative side of the PGA;
5, Comparator outputs to TIM are detailed in the Comparator Interconnect Relationships section;;

Figure 18-4 Comparator7 Connection Diagram



Notes:
1, dacx_iout indicates the internal output channel of the dacx;
2, VREF_VCx represents the output of the internal 6-bit DAC;
3, PGAx_OUT_P represents the output of the positive side of the PGA;
4, PGAx_OUT_N represents the output of the negative side of the PGA;
5, Comparator outputs to TIM are detailed in the Comparator Interconnect Relationships section;;

18.2 COMP Main Features

- Supports up to 7 independent comparators.
- Embedded three 64 level programmable comparison voltage reference sources VREF1, VREF2, VREF3.
- Supports filter clock, filter reset.
- Output polarity can be configured high and low.
- Support 8 programmable hysteresis levels.
- The comparison result can be output to the I/O port or trigger timer, which is use to event capture, OCREF_CLR event, break event, and interrupt generation.
- The input channels can be selected from I/O ports, VREF1, VREF2, VREF3, the general 12-bit DAC, and the channel output of the internal PGA.
- It can be read-only or read-write, and can be unlocked only after a reset.
- Supports blanking, blanking source can be configurabled.
- COMP1/COMP2、COMP3/COMP4、COMP5/COMP6 can form window comparators.
- You can wake the system from Sleep mode by generating an interrupt.
- Filter window size can be configured.
- Filter threshold size can be configured.
- The sampling frequency for filtering can be configured.

18.3 COMP Configuration Precedure

The complete configuration includes the following steps If the default configuration is used, skip the corresponding configuration items.

- Configurable hysteresis level COMPx_CTRL.HYST[2:0].
- Configure the output polarity COMPx_CTRL.POL.
- Configuration input selection, Comparator positive input COMPx_CTRL.INPSEL[3:0], negative input COMPx_CTRL.INMSEL[2:0].
- Enable output selection configuration COMPx_OTIMEN.
- Configure blanking source COMPx_CTRL.BLKING[2:0].
- Configure the filter sampling window COMPx_FILC.SAMPW[4:0].
- Configure the threshold COMPx_FILC.THRESH[4:0] (threshold should be greater than COMPx_FILC.SAMPW[4:0]/2).
- Configure the filter sampling frequency (for timer applications, sampling frequency should be greater than 5MHz).
- Enable COMPx_FILC.FILEN filter.
- Enable COMPx_CTRL.EN on the comparator.

Note: For the above steps, the filter should be enabled first and then the comparator should be enabled. The comparator should be enabled after the filtering (if enabled) is configured and enabled. In addition, when the comparator control register is locked, it can only be unlocked by reset.

18.4 COMP Operating Mode

18.4.1 Window Comparators

The comparators can be combined into 3 window comparators as follows:

- Comparator 1 and comparator 2 can be combined as window comparator and share PB10.
- Comparator 3 and comparator 4 can be combined as window comparator and share PA0.
- Comparator 5 and comparator 6 can be combined as window comparator and share PD12.

Note: Without using the window comparator, PB10 cannot be directly used as the positive input of COMP2, PA0 cannot be directly used as the positive input of COMP4, and PD12 cannot be directly used as the positive input of COMP6.

18.4.2 Independent Comparators

The seven comparators can be configured independently to complete the comparator function. The output of a comparator can be output to an I/O port. Each comparator has a different remapped port, Enable the comparator to output to the TIM port by configuring the corresponding bit of COMP_OTIMEN.

The Comparator output, supports triggering events, such as configuring it as the break function of timer x or the OCREFCLEAR function of timer x.

Note: Refer to the comparator interconnection for specific configuration

18.5 Comparator Interconnection

For the interconnection of the output port of the comparator, please refer to the chapter on the multiplexing function of GPIO, which defines the value of the remapping of the comparator OUT.

The comparator OUT pin is as follows:

COMP1	COMP2	COMP3	COMP4	COMP5	COMP6	COMP7
PF4	PA2	PC2	PC5	PB0	PC6	PC2
PA0	PA6	PA2	PB1	PB11	PC9	PD12
PA6	PA7	PB10	PB14	PC7	PA10	PC8
PB1	PA12	PB15	PB12	PA9	PA12	PA8
PA11	PB9	PC8	PC11	PA11	PB7	-
PB8	-	PA8	PB6	PB6	-	-
-	-	PC10	-	-	-	-
-	-	PB7	-	-	-	-

The comparator INP pin has the following configurations:

INPSEL	COMP1	COMP2	COMP3	COMP4	COMP5	COMP6	COMP7
0000	PF8	PB10	PH1	PA0	PC3	PD12	PB14
0001	PC1	PB11	PD15	PB0	PA3	PD9	PC1
0010	PA1	PF10	PC0	PE7	PC4	PC5	PD14
0011	PB1	PC3	PC1	PB15	PB13	PB11	DAC4_out
0100	PA7	PA3	PB0	PC9	DAC3_out	DAC4_out	DAC7_out
0101	DAC1_out	PA7	PB14	DAC2_out	DAC7_out	DAC8_out	VREF_VC3
0110	DAC5_out	PB1	DAC1_out	DAC6_out	VREF_VC3	VREF_VC3	PGA4_OUT_P
0111	VREF_VC1	DAC2_out	DAC5_out	VREF_VC2	PGA3_OUT_P	PGA3_OUT_N	PGA4_OUT_N
1000	PGA1_OUT_P	DAC6_out	VREF_VC2	PGA2_OUT_N	PD12	-	-
1001	PB10	VREF_VC1	PGA2_OUT_P	-	-	-	-
1010	-	PGA1_OUT_N	PA0	-	-	-	-

The comparator INM pin has the following configurations:

INPSEL	COMP1	COMP2	COMP3	COMP4	COMP5	COMP6	COMP7
0000	PA0	PA2	PB14	PB13	PA4	PA7	PB12
0001	PA4	PA4	PA4	PA4	PA5	PA4	PA4
0010	PA5	PA5	PA5	PA5	PB10	PA5	PA5
0011	PA6	PB1	PC0	PC4	PD10	PB15	PC0
0100	DAC1_out	PE8	PE7	PB2	PD13	PD8	PD15
0101	DAC5_out	DAC2_out	PB12	PE8	DAC3_out	PD10	DAC4_out

INPSEL	COMP1	COMP2	COMP3	COMP4	COMP5	COMP6	COMP7
0110	VREF_VC1	DAC6_out	PD15	DAC2_out	DAC7_out	DAC4_out	DAC7_out
0111	-	VREF_VC1	PH1	DAC6_out	VREF_VC3	DAC8_out	VREF_VC3
1000	-	-	DAC1_out	VREF_VC2	-	VREF_VC3	-
1001	-	-	DAC5_out	-	-	-	-
1010	-	-	VREF_VC2	-	-	-	-

The output signals of comparators 1 to 4 are as follows:

COMP1_OUT	COMP2_OUT	COMP3_OUT	COMP4_OUT
ATIMx_OCrefclear0	ATIMx_OCrefclear1	ATIMx_OCrefclear2	ATIMx_OCrefclear3
GTIMx_OCrefclear0	GTIMx_OCrefclear1	GTIMx_OCrefclear2	GTIMx_OCrefclear3
ATIMx_ETR1	ATIMx_ETR2	ATIMx_ETR3	ATIMx_ETR4
GTIMx_ETR1	GTIMx_ETR2	GTIMx_ETR3	GTIMx_ETR4
ATIMx_ti1_in1	ATIMx_ti1_in2	ATIMx_ti1_in3	ATIMx_ti1_in4
GTIMx_ti1_in1	GTIMx_ti1_in2	GTIMx_ti1_in3	GTIMx_ti1_in4
GTIM1_ti3_in1	GTIM2_ti3_in1	GTIM3_ti3_in1	GTIM4_ti3_in1
GTIM1_ti4_in1	GTIM2_ti4_in1	GTIM3_ti4_in1	GTIM4_ti4_in1
ATIMx_brk_cmp1	ATIMx_brk_cmp2	ATIMx_brk_cmp3	ATIMx_brk_cmp4
GTIM8_brk_cmp1	GTIM8_brk_cmp2	GTIM8_brk_cmp3	GTIM8_brk_cmp4
GTIM9_brk_cmp1	GTIM9_brk_cmp2	GTIM9_brk_cmp3	GTIM9_brk_cmp4
GTIM10_brk_cmp1	GTIM10_brk_cmp2	GTIM10_brk_cmp3	GTIM10_brk_cmp4
ATIMx_brk2_cmp1	ATIMx_brk2_cmp2	ATIMx_brk2_cmp3	ATIMx_brk2_cmp4
GTIM8_brk2_cmp1	GTIM8_brk2_cmp2	GTIM8_brk2_cmp3	GTIM8_brk2_cmp4
GTIM9_brk2_cmp1	GTIM9_brk2_cmp2	GTIM9_brk2_cmp3	GTIM9_brk2_cmp4
GTIM10_brk2_cmp1	GTIM10_brk2_cmp2	GTIM10_brk2_cmp3	GTIM10_brk2_cmp4
SHRTIM_EEV4	SHRTIM_EEV1	SHRTIM_EEV5	SHRTIM_EEV2
SHRTIM_EEV6	SHRTIM_EEV6	SHRTIM_EEV8	SHRTIM_EEV7
SHRTIM_FALT4	SHRTIM_FALT1	SHRTIM_FALT5	SHRTIM_FALT2
LPTIMx_IN1	LPTIMx_IN1	LPTIMx_IN1	LPTIMx_IN1
LPTIMx_IN2	LPTIMx_IN2	LPTIMx_IN2	LPTIMx_IN2
LPTIMx_ETR	LPTIMx_ETR	LPTIMx_ETR	LPTIMx_ETR

The output signals of comparators 5 to 7 are as follows：

COMP5_OUT	COMP6_OUT	COMP7_OUT
ATIMx_OCrefclear4	ATIMx_OCrefclear5	ATIMx_OCrefclear6
GTIMx_OCrefclear4	GTIMx_OCrefclear5	GTIMx_OCrefclear6
ATIMx_ETR5	ATIMx_ETR6	ATIMx_ETR7
GTIMx_ETR5	GTIMx_ETR6	GTIMx_ETR7
ATIMx_ti2_in1	ATIMx_ti2_in2	ATIMx_ti2_in3
GTIMx_ti2_in1	GTIMx_ti2_in2	GTIMx_ti2_in3
GTIM8_ti3_in1	GTIM9_ti3_in1	GTIM10_ti3_in1
GTIM8_ti4_in1	GTIM9_ti4_in1	GTIM10_ti4_in1
ATIMx_brk_cmp5	ATIMx_brk_cmp6	ATIMx_brk_cmp7

GTIM8_brk_cmp5	GTIM8_brk_cmp6	GTIM8_brk_cmp7
GTIM9_brk_cmp5	GTIM9_brk_cmp6	GTIM9_brk_cmp7
GTIM10_brk_cmp5	GTIM10_brk_cmp6	GTIM10_brk_cmp7
ATIMx_brk2_cmp5	ATIMx_brk2_cmp6	ATIMx_brk2_cmp7
GTIM8_brk2_cmp5	GTIM8_brk2_cmp6	GTIM8_brk2_cmp7
GTIM9_brk2_cmp5	GTIM9_brk2_cmp6	GTIM9_brk2_cmp7
GTIM10_brk2_cmp5	GTIM10_brk2_cmp6	GTIM10_brk2_cmp7
SHRTIM_EEV4	SHRTIM_EEV3	SHRTIM_EEV5
SHRTIM_EEV9	SHRTIM_EEV8	SHRTIM_EEV10
SHRTIM_FALT6	SHRTIM_FALT3	SHRTIM_FALT7
LPTIMx_IN1	LPTIMx_IN1	LPTIMx_IN1
LPTIMx_IN2	LPTIMx_IN2	LPTIMx_IN2
LPTIMx_ETR	LPTIMx_ETR	LPTIMx_ETR

18.6 Comparator Output

The COMP can output the value before filtering of COMPx_CTRL.OUT and the value after filtering of COMPx_CTRL.FLTOUT.

- When filtering is turned off, the output result of COMPx_CTRL.OUT is valid, and the output result of COMPx_CTRL.FLTOUT is invalid;
- When filtering is enabled, COMPx_CTRL.OUT and COMPx_CTRL.FLTOUT respectively indicate the comparator output results before and after filtering.

18.7 Interrupt

COMP supports interrupt response, COMP1, COMP2 and COMP3 share one interrupt entry , COMP4, COMP5 and COMP6 share one interrupt entry, , COMP7 has one interrupt entry exclusively. There are two cases of interrupt generation as follows.

- The polarity of COMPx_CTRL.POL is not reversed, and the interrupt is enabled. When INPSEL > INMSEL, the comparator interrupt will be generated when COMPx_CTRL.OUT is set to 1 by hardware.
- The polarity of COMPx_CTRL.POL is reversed, and the interrupt is enabled. When INPSEL < INMSEL, the comparator interrupt is generated when COMPx_CTRL.OUT is set to 1 by hardware.

Note: COMP interrupt usage requires configuring the EXTI line first, refer to the NVIC section.

18.8 COMP Register

18.8.1 COMP Register Overview

Table 18-1 COMP Register Overview

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x10	COMP1_CTRL	Reserved										FLTOUT	OUT	BLKING[2:0]			HYST[2:0]			POL	Reserved				INPSEL[3:0]				Reserved	INMSEL[2:0]			EN
	Reset value											0	0	0	0	0	0	0	0	0					0	0	0	0		0	0	0	0
0x14	COMP1_FILC	Reserved																				DAC_EN	SAMPW[4:0]					THRESH[4:0]					FILEN
	Reset value																					0	0	0	0	0	0	0	0	0	0	0	0
0x18	COMP1_FILP	Reserved																CLKPSC[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x20	COMP2_CTRL	Reserved										FLTOUT	OUT	BLKING[2:0]			HYST[2:0]			POL	Reserved				INPSEL[3:0]				INMSEL[3:0]				EN
	Reset value											0	0	0	0	0	0	0	0	0					0	0	0	0	0	0	0	0	0
0x24	COMP2_FILC	Reserved																				DAC_EN	SAMPW[4:0]					THRESH[4:0]					FILEN
	Reset value																					0	0	0	0	0	0	0	0	0	0	0	0
0x28	COMP2_FILP	Reserved																CLKPSC[15:0]															
	Reset value																	b	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x30	COMP3_CTRL	Reserved										FLTOUT	OUT	BLKING[2:0]			HYST[2:0]			POL	Reserved				INPSEL[3:0]				INMSEL[3:0]				EN
	Reset value											0	0	0	0	0	0	0	0	0					0	0	0	0	0	0	0	0	0
0x34	COMP3_FILC	Reserved																				DAC_EN	SAMPW[4:0]					THRESH[4:0]					FILEN
	Reset value																					0	0	0	0	0	0	0	0	0	0	0	0

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x38	COMP3_FILP	Reserved																CLKPSC[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x40	COMP4_CTRL	Reserved										FLTOUT	OUT	BLKING[2:0]			HYST[2:0]			POL	Reserved				INPSEL[3:0]				INMSEL[3:0]				EN
	Reset value											0	0	0	0	0	0	0	0	0					0	0	0	0	0	0	0	0	0
0x44	COMP4_FILC	Reserved																				DAC_EN	SAMPW[4:0]					THRESH[4:0]					FILEN
	Reset value																					0	0	0	0	0	0	0	0	0	0	0	0
0x48	COMP4_FILP	Reserved																CLKPSC[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x50	COMP5_CTRL	Reserved										FLTOUT	OUT	BLKING[2:0]			HYST[2:0]			POL	Reserved				INPSEL[3:0]				Reserved	INMSEL[2:0]			EN
	Reset value											0	0	0	0	0	0	0	0	0					0	0	0	0		0	0	0	0
0x54	COMP5_FILC	Reserved																				DAC_EN	SAMPW[4:0]					THRESH[4:0]					FILEN
	Reset value																					0	0	0	0	0	0	0	0	0	0	0	0
0x58	COMP5_FILP	Reserved																CLKPSC[15:0]															
	Reset value																	b	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x60	COMP6_CTRL	Reserved										FLTOUT	OUT	BLKING[2:0]			HYST[2:0]			POL	Reserved				INPSEL[3:0]				INMSEL[3:0]				EN
	Reset value											0	0	0	0	0	0	0	0	0					0	0	0	0	0	0	0	0	0
0x64	COMP6_FILC	Reserved																				DAC_EN	SAMPW[4:0]					THRESH[4:0]					FILEN
	Reset value																					0	0	0	0	0	0	0	0	0	0	0	0

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x68	COMP6_FILP	Reserved																CLKPSC[15:0]															
	Reset value	Reserved																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x70	COMP7_CTRL	Reserved										FLTOUT	OUT	BLKING[2:0]			HYST[2:0]			POL	Reserved					INPSEL[2:0]			Reserved	INMSEL[2:0]			EN
	Reset value	Reserved										0	0	0	0	0	0	0	0	0	Reserved					0	0	0	Reserved	0	0	0	0
0x74	COMP7_FILC	Reserved																				DAC_EN	SAMPW[4:0]					THRESH[4:0]					FILEN
	Reset value	Reserved																				0	0	0	0	0	0	0	0	0	0	0	0
0x78	COMP7_FILP	Reserved																CLKPSC[15:0]															
	Reset value	Reserved																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x88	COMP_WINMODE	Reserved																													CMP56MD	CMP34MD	CMP12MD
	Reset value	Reserved																													0	0	0
0x8C	COMP_LOCK	Reserved																									CMP7LK	CMP6LK	CMP5LK	CMP4LK	CMP3LK	CMP2LK	CMP1LK
	Reset value	Reserved																									0	0	0	0	0	0	0
0x90	COMP_INTEN	Reserved																									CMP7IEN	CMP6IEN	CMP5IEN	CMP4IEN	CMP3IEN	CMP2IEN	CMP1IEN
	Reset value	Reserved																									0	0	0	0	0	0	0
0x94	COMP_INTSTS	Reserved																									CMP7IS	CMP6IS	CMP5IS	CMP4IS	CMP3IS	CMP2IS	CMP1IS
	Reset value	Reserved																									0	0	0	0	0	0	0
0x98	COMP_INVREF	Reserved											VV3TRM[5:0]						VV3EN	VV2TRM[5:0]						VV2EN	VV1TRM[5:0]						VV1EN
	Reset value	Reserved											0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x9C	COMP_OTIMEN	Reserved																									CMP7OEN	CMP6OEN	CMP5OEN	CMP4OEN	CMP3OEN	CMP2OEN	CMP1OEN

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reset value																										0	0	0	0	0	0	0

18.8.2 COMP1 Control Register（COMP1_CTRL）

Offset address: 0x10

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved										FLTOUT	OUT	BLKING[2:0]			HYST[2]
										r	r	rw			rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HYST[1:0]		POL	Reserved				INPSEL[3:0]				Reserved	INMSEL[2:0]			EN
rw		rw					rw					rw			rw

Bit Field	Name	Description
31:22	Reserved	Reserved, the reset value must be maintained.
21	FLTOUT	The output state after the comparator output is filtered. 0: Output low level; 1: Output high level.
20	OUT	The output state of the comparator output. 0: Output low level; 1: Output high level.
19:17	BLKING[2:0]	These bits select which timer output controls the comparator 1 output blanking. 000: No blanking; 001: ATIM1 OC5 selected as blanking source; 010: ATIM2 OC5 selected as blanking source; 011: ATIM3 OC5 selected as blanking source; 100: ATIM8 OC5 selected as blanking source; 101: ATIM9 OC5 selected as blanking source; 110: ATIM10 OC5 selected as blanking source. Other: Reserved.

16:14	HYST[2:0]	These bits control the hysteresis level.
		000: No hysteresis;
		001: 10mV hysteresis;
		010: 20mV hysteresis;
		011: 30mV hysteresis;
		100: 40mV hysteresis;
		101: 50mV hysteresis;
		110: 60mV hysteresis;
		111: 70mV hysteresis;

13	POL	This bit is used to invert the comparator output. 0: Output is not inverted； 1: Output is inverted.
12:9	Reserved	Reserved, the reset value must be maintained.

8:5	INPSEL[3:0]	Comparator positive terminal selection bit: 0000: PF8 0001: PC1 0010: PA1 0011: PB1 0100: PA7 0101: dac1_iout 0110: dac5_iout 0111: VREF_VC1 1000: PGA1_OUT_P 1001: PB10 Other: Reserved
4	Reserved	Reserved, the reset value must be maintained.
3:1	INMSEL[2:0]	Comparator negative terminal selection bit: 000: PA0 001: PA4 010: PA5 011: PA6 100: dac1_iout 101: dac5_iout 110: VREF_VC1 Other: Reserved
0	EN	This bit switches COMP ON/OFF 0: Comparator disabled; 1: Comparator enable.

18.8.3 COMP1 Filter Control Register（COMP1_FILC）

Offset address: 0x14

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				DACEN	SAMPW[4:0]					THRESH[4:0]					FILEN
				rw	rw					rw					rw

Bit Field	Name	Description
31:12	Reserved	Reserved, the reset value must be maintained
11	DACEN	When an external DAC is used as the input of the comparator, the comparator may collect unstable values. Enabling this flag can prevent unstable values from being collected and applied by COMP1.

		0: Disable; 1: Enable.
10:6	SAMPW[4:0]	Low-pass filter sampling window size, sampling window = SAMPW + 1.
5:1	THRESH[4:0]	The low-pass filter threshold is set. At least the sampling threshold of the opposite state in the sample window can change the output state. This value is required to be greater than SAMPW / 2.
0	FILEN	Filter enable. 0: Disable; 1: Enable.

18.8.4 COMP1 Filter Prescaler Register（COMP1_FILP）

Offset address: 0x18

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CLKPSC[15:0]															
rw															

Bit Field	Name	Description
31:16	Reserved	Reserved, the reset value must be maintained.
15:0	CLKPSC[15:0]	Low filter sample clock prescaler. Systerm clock divider = CLKPSC + 1。 0: Every cycle 1: Every 2 cycle 2: Every 3 cycle … 65535: Every 65536 cycle

18.8.5 COMP2 Control Register（COMP2_CTRL）

Offset address: 0x20

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved										FLTOUT	OUT	BLKING[2:0]			HYST[2]
										r	r		rw		rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HYST[1:0]		POL	Reserved				INPSEL[3:0]				INMSEL[3:0]				EN

rw rw rw rw rw

Bit Field	Name	Description
31:22	Reserved	Reserved, the reset value must be maintained
21	FLTOUT	The output state after the comparator output is filtered. 0: Ouput low level 1: Output high level
20	OUT	The output state of the comparator output. 0: Output low level 1: Output high level
19:17	BLKING[2:0]	These bits select which Timer output controls the comparator 2 output blanking. 000: No blanking 001: ATIM1 OC5 selected as blanking source. 010: ATIM2 OC5 selected as blanking source. 011: ATIM3 OC5 selected as blanking source. 100: ATIM8 OC5 selected as blanking source. 101: ATIM9 OC5 selected as blanking source. 110: ATIM10 OC5 selected as blanking source. Other configurations: Reserved.

16:14	HYST[2:0]	These bits control the hysteresis level.
		000: No hysteresis;
		001: 10mV hysteresis;
		010: 20mV hysteresis;
		011: 30mV hysteresis;
		100: 40mV hysteresis;
		101: 50mV hysteresis;
		110: 60mV hysteresis;
		111: 70mV hysteresis;

13	POL	This bit is used to invert the comparator output. 0: Output is not inverted; 1: Output is inverted.
12:9	Reserved	Reserved, the reset value must be maintained
8:5	INPSEL[3:0]	Comparator positive terminal selection bit: 0000: PB10 0001: PB11 0010: PF10 0011: PC3 0100: PA3 0101: PA7 0110: PB1 0111: dac2_iout 1000: dac6_iout

Bit Field	Name	Description
		1001: VREF_VC1 1010: PGA1_OUT_N Other: Reserved
4:1	INMSEL[2:0]	Comparator negative terminal selection bit: 000: PA2 0001: PA4 0010: PA5 0011: PB1 0100: PE8 0101: dac2_iout 0110: dac6_iout 0111: VREF_VC1 Other: Reserved
0	EN	This bit switches COMP ON/OFF 0: Comparator disabled; 1: Comparator enabled.

18.8.6 COMP2 Filter Control Register（COMP2_FILC）

Offset address: 0x24

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				DACEN	SAMPW[4:0]					THRESH[4:0]					FILEN
				rw	rw					rw					rw

Bit Field	Name	Description
31:12	Reserved	Reserved, the reset value must be maintained.
11	DACEN	When an external DAC is used as the input of the comparator, the comparator may collect unstable values. Enabling this flag can prevent unstable values from being collected and applied by COMP2. 0: Disable; 1: Enable.
10:6	SAMPW[4:0]	Low-pass filter sampling window size, sampling window = SAMPW + 1.
5:1	THRESH[4:0]	The low-pass filter threshold is set. At least the sampling threshold of the opposite state in the sample window can change the output state. This value is required to be greater than SAMPW / 2.
0	FILEN	Filter enable 0: Disable;

		1: Enable.

18.8.7 COMP2 Filter Prescaler Register（COMP2_FILP）

Offset address: 0x28

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CLKPSC[15:0]															
rw															

Bit Field	Name	Description
31:16	Reserved	Reserved, the reset value must be maintained.

15:0	CLKPSC[15:0]	Low filter sample clock prescaler.
		System clock divider = CLKPSC + 1。
		0: Every cycle;
		1: Every 2 cycle;
		2: Every 3 cycle;
		…
		65535: Every 65536 cycle;

18.8.8 COMP3 Control Register（COMP3_CTRL）

Offset address: 0x30

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved										FLTOUT	OUT	BLKING[2:0]			HYST[2]
										r	r	rw			rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HYST[1:0]		POL	Reserved				INPSEL[3:0]				INMSEL[3:0]				EN
rw		rw					rw				rw				rw

Bit Field	Name	Description
31:22	Reserved	Reserved, the reset value must be maintained.
21	FLTOUT	The output state after the comparator output is filtered. 0: Output low level 1: Output high level
20	OUT	The output state of the comparator output. 0: Output low level

		1: Output high level
19:17	BLKING[2:0]	These bits select which Timer output controls the comparator 3 output blanking. 000: No blanking 001: ATIM1 OC5 selected as blanking source 010: ATIM2 OC5 selected as blanking source 011: ATIM3 OC5 selected as blanking source 100: ATIM8 OC5 selected as blanking source 101: ATIM9 OC5 selected as blanking source 110: ATIM10 OC5 selected as blanking source Other configurations: Reserved.

16:14	HYST[2:0]	These bits control the hysteresis level.
		000: No hysteresis;
		001: 10mV hysteresis;
		010: 20mV hysteresis;
		011: 30mV hysteresis;
		100: 40mV hysteresis;
		101: 50mV hysteresis;
		110: 60mV hysteresis;
		111: 70mV hysteresis;

13	POL	This bit is used to invert the comparator output. 0: Output is not inverted; 1: Output is inverted.
12:9	Reserved	Reserved, the reset value must be maintained.
8:5	INPSEL[3:0]	Comparator positive terminal selection bit: 0000: PH1 0001: PD15 0010: PC0 0011: PC1 0100: PB0 0101: PB14 0110: dac1_iout 0111: dac5_iout 1000: VREF_VC2 1001: PGA2_OUT_P 1010: PA0 Other: Reserved
4:1	INMSEL[2:0]	Comparator negative terminal selection bit: 0000: PB14 0001: PA4 0010: PA5 0011: PC0 0100: PE7 0101: PB12

		0110: PD15 0111: PH1 1000: dac1_iout 1001: dac5_iout 1010: VREF_VC2 Other: Reserved
0	EN	This bit switches COMP ON/OFF 0: Comparator disabled; 1: Comparator enabled。

18.8.9 COMP3 Filter Control Register（COMP3_FILC）

Offser address: 0x34

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				DACEN	SAMPW[4:0]					THRESH[4:0]					FILEN
				rw	rw					rw					rw

Bit Field	Name	Description
31:12	Reserved	Reserved, the reset value must be maintained.
11	DACEN	When an external DAC is used as the input of the comparator, the comparator may collect unstable values. Enabling this flag can prevent unstable values from being collected and applied by COMP3. 0: Disable; 1: Enable.
10:6	SAMPW[4:0]	Low-pass filter sampling window size, sampling window = SAMPW + 1.
5:1	THRESH[4:0]	The low-pass filter threshold is set. At least the sampling threshold of the opposite state in the sample window can change the output state. This value is required to be greater than SAMPW / 2.
0	FILEN	Filter enable 0: Disable; 1: Enable.

18.8.10 COMP3 Filter Prescaler Register（COMP3_FILP）

Offset address: 0x38

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

Reserved

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CLKPSC[15:0]															
rw															

Bit Field	Name	Description
31:16	Reserved	Reserved, the reset value must be maintained.

15:0	CLKPSC[15:0]	Low filter sample clock prescaler.
		System clock divider = CLKPSC + 1.
		0: Every cycle;
		1: Every 2 cycle;
		2: Every 3 cycle;
		…
		65535: Every 65536 cycle;

18.8.11 COMP4 Control Register（COMP4_CTRL）

Offset address: 0x40

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved										FLTOUT	OUT	BLKING[2:0]			HYST[2]
										r	r	rw			rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HYST[1:0]		POL	Reserved				INPSEL[3:0]				INMSEL[3:0]				EN
rw		rw					rw				rw				rw

Bit Field	Name	Description
31:22	Reserved	Reserved, the reset value must be maintained.
21	FLTOUT	The output state after the comparator output is filtered. 0: Output low level 1: Output high level
20	OUT	The output state of the comparator output. 0: Output low level 1: Output high level
19:17	BLKING[2:0]	These bits select which Timer output controls the comparator 4 output blanking. 000: No blanking 001: ATIM1 OC5 selected as blanking source 010: ATIM2 OC5 selected as blanking source 011: ATIM3 OC5 selected as blanking source 100: ATIM8 OC5 selected as blanking source 101: ATIM9 OC5 selected as blanking source

		110: ATIM10 OC5 selected as blanking source Other configurations: Reserved.

16:14	HYST[2:0]	These bits control the hysteresis level.
		000: No hysteresis;
		001: 10mV hysteresis;
		010: 20mV hysteresis;
		011: 30mV hysteresis;
		100: 40mV hysteresis;
		101: 50mV hysteresis;
		110: 60mV hysteresis;
		111: 70mV hysteresis;

13	POL	This bit is used to invert the comparator output 0: Output is not inverted; 1: Output is inverted.
12:9	Reserved	Reserved, the reset value must be maintained.
8:5	INPSEL[3:0]	Comparator positive terminal selection bit: 0000: PA0 0001: PB0 0010: PE7 0011: PB15 0100: PC9 0101: dac2_iout 0110: dac6_iout 0111: VREF_VC2 1000: PGA2_OUT_N Other: Reserved Other: Reserved
4:1	INMSEL[3:0]	Comparator negative terminal selection bit: 0000：PB13 0001：PA4 0010：PA5 0011：PC4 0100：PB2 0101：PE8 0110：dac2_iout 0111：dac6_iout 1000：VREF_VC2 Other: Reserved
0	EN	This bit switches COMP ON/OFF 0: Comparator disabled; 1: Comparator enabled.

18.8.12 COMP4 Filter Control Register（COMP4_FILC）

Offset address: 0x44

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				DACEN	SAMPW[4:0]					THRESH[4:0]					FILEN
				rw	rw					rw					rw

Bit Field	Name	Description
31:12	Reserved	Reserved, the reset value must be maintained.
11	DACEN	When an external DAC is used as the input of the comparator, the comparator may collect unstable values. Enabling this flag can prevent unstable values from being collected and applied by COMP4. 0: Disable; 1: Enable.
10:6	SAMPW[4:0]	Low-pass filter sampling window size, sampling window = SAMPW + 1.
5:1	THRESH[4:0]	The low-pass filter threshold is set. At least the sampling threshold of the opposite state in the sample window can change the output state. This value is required to be greater than SAMPW / 2.
0	FILEN	Filter enable 0: Disable; 1: Enable.

18.8.13 COMP4 Filter Prescaler Register（COMP4_FILP）

Offset address: 0x48

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CLKPSC[15:0]															
rw															

Bit Field	Name	Description
31:16	Reserved	Reserved, the reset value must be maintained.

15:0	CLKPSC[15:0]	Low filter sample clock prescaler.
		System clock divider = CLKPSC + 1.
		0: Every cycle;
		1: Every 2 cycle;
		2: Every 3 cycle;

		… 65535: Every 65536 cycle;

18.8.14 COMP5 Control Register（COMP5_CTRL）

Offset address: 0x50

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved										FLTOUT	OUT	BLKING[2:0]			HYST[2]
										r	r		rw		rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HYST[1:0]		POL	Reserved				INPSEL[3:0]				Reserved	INMSEL[2:0]			EN
rw		rw						rw					rw		rw

Bit Field	Name	Description
31:22	Reserved	Reserved, the reset value must be maintained.
21	FLTOUT	The output state after the comparator output is filtered. 0: Output low level 1: Output high level
20	OUT	The output state of the comparator output. 0: Output low level 1: Output high level
19:17	BLKING[2:0]	These bits select which Timer output controls the comparator 5 output blanking. 000: No blanking 001: ATIM1 OC5 selected as blanking source 010: ATIM2 OC5 selected as blanking source 011: ATIM3 OC5 selected as blanking source 100: ATIM8 OC5 selected as blanking source 101: ATIM9 OC5 selected as blanking source 110: ATIM10 OC5 selected as blanking source Other configurations: Reserved

16:14	HYST[2:0]	These bits control the hysteresis level.
		000: No hysteresis;
		001: 10mV hysteresis;
		010: 20mV hysteresis;
		011: 30mV hysteresis;
		100: 40mV hysteresis;
		101: 50mV hysteresis;
		110: 60mV hysteresis;
		111: 70mV hysteresis;

13	POL	This bit is used to invert the comparator output. 0: Output is not inverted;

		1: Output is inverted.
12:9	Reserved	Reserved, the reset value must be maintained.
8:5	INPSEL[3:0]	Comparator positive terminal selection bit: 0000: PC3 0001: PA3 0010: PC4 0011: PB13 0100: dac3_iout 0101: dac7_iout 0110: VREF_VC3 0111: PGA3_OUT_P 1000: PD12 Other: Reserved
4	Reserved	Reserved, the reset value must be maintained.
3:1	INMSEL[2:0]	Comparator negative terminal selection bit: 000: PA4 001: PA5 010: PB10 011: PD10 100: PD13 101: dac3_iout 110: dac7_iout 111: VREF_VC3 Other: Reserved
0	EN	This bit switches COMP ON/OFF 0: Comparator disabled; 1: Comparator enabled.

18.8.15 COMP5 Filter Control Register（COMP5_FILC）

Offset address: 0x54

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				DACEN	SAMPW[4:0]					THRESH[4:0]					FILEN
				rw			rw					rw			rw

Bit Field	Name	Description
31:12	Reserved	Reserved, the reset value must be maintained.
11	DACEN	When an external DAC is used as the input of the comparator, the comparator may

		collect unstable values. Enabling this flag can prevent unstable values from being collected and applied by COMP5. 0: Disable; 1: Enable.
10:6	SAMPW[4:0]	Low-pass filter sampling window size, sampling window = SAMPW + 1.
5:1	THRESH[4:0]	The low-pass filter threshold is set. At least the sampling threshold of the opposite state in the sample window can change the output state. This value is required to be greater than SAMPW / 2.
0	FILEN	Filter enable 0: Disable; 1: Enable.

18.8.16 COMP5 Filter Prescaler Register（COMP5_FILP）

Offset address: 0x58

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CLKPSC[15:0]															
rw															

Bit Field	Name	Description
31:16	Reserved	Reserved, the reset value must be maintained.

15:0	CLKPSC[15:0]	Low filter sample clock prescaler.
		System clock divider = CLKPSC + 1。
		0: Every cycle;
		1: Every 2 cycle;
		2: Every 3 cycle;
		…
		65535: Every 65536 cycle;

18.8.17 COMP6 Control Register（COMP6_CTRL）

Offset address: 0x60

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved										FLTOUT	OUT	BLKING[2:0]			HYST[2]
										r	r		rw		rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HYST[1:0]		POL	Reserved				INPSEL[3:0]				INMSEL[3:0]				EN
rw		rw					rw				rw				rw

Bit Field	Name	Description
31:22	Reserved	Reserved, the reset value must be maintained.
21	FLTOUT	The output state after the comparator output is filtered. 0: Output low level 1: Output high level
20	OUT	The output state of the comparator output 0: Output low level 1: Output high level
19:17	BLKING[2:0]	These bits select which Timer output controls the comparator 6 output blanking. 000: No blanking 001: ATIM1 OC5 selected as blanking source 010: ATIM2 OC5 selected as blanking source 011: ATIM3 OC5 selected as blanking source 100: ATIM8 OC5 selected as blanking source 101: ATIM9 OC5 selected as blanking source 110: ATIM10 OC5 selected as blanking source Other configurations: Reserved

16:14	HYST[2:0]	These bits control the hysteresis level.
		000: No hysteresis;
		001: 10mV hysteresis;
		010: 20mV hysteresis;
		011: 30mV hysteresis;
		100: 40mV hysteresis;
		101: 50mV hysteresis;
		110: 60mV hysteresis;
		111: 70mV hysteresis;

13	POL	This bit is used to invert the comparator output. 0: Output is not inverted; 1: Output is inverted.
12:9	Reserved	Reserved, the reset value must be maintained.
8:5	INPSEL[3:0]	Comparator positive terminal selection bit: 0000: PD12 0001: PD11 0010: PD9 0011: PC5 0100: PB11 0101: dac4_iout 0110: dac8_iout 0111: VREF_VC3

		1000: PGA3_OUT_N Other: Reserved
4:1	INMSEL[3:0]	Comparator negative terminal selection bit: 0000: PA7 0001: PA4 0010: PA5 0011: PB15 0100: PD8 0101: PD10 0110: dac4_iout 0111: dac8_iout 1000: VREF_VC3 Other : Reserved
0	EN	This bit switches COMP ON/OFF 0: Comparator disabled; 1: Comparator enabled.

18.8.18 COMP6 Filter Control Register（COMP6_FILC）

Offset address: 0x64

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				DACEN	SAMPW[4:0]					THRESH[4:0]					FILEN
				rw			rw					rw			rw

Bit Field	Name	Description
31:12	Reserved	Reserved, the reset value must be maintained.
11	DACEN	When an external DAC is used as the input of the comparator, the comparator may collect unstable values. Enabling this flag can prevent unstable values from being collected and applied by COMP6. 0: Disable; 1: Enable.
10:6	SAMPW[4:0]	Low-pass filter sampling window size, sampling window = SAMPW + 1.
5:1	THRESH[4:0]	The low-pass filter threshold is set. At least the sampling threshold of the opposite state in the sample window can change the output state. This value is required to be greater than SAMPW / 2.
0	FILEN	Filter enable 0: Disable; 1: Enable.

18.8.19 COMP6 Filter Prescaler Register（COMP6_FILP）

Offset address: 0x68

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CLKPSC[15:0]															
rw															

Bit Field	Name	Description
31:16	Reserved	Reserved, the reset value must be maintained.

15:0	CLKPSC[15:0]	Low filter sample clock prescaler，
		System clock divider = CLKPSC + 1。
		0: Every cycle;
		1: Every 2 cycle;
		2: Every 3 cycle;
		…
		65535: Every 65536 cycle;

18.8.20 COMP7 Control Register（COMP7_CTRL）

Offset address: 0x70

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved										FLTOUT	OUT	BLKING[2:0]			HYST[2]
										r	r		rw		rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HYST[1:0]		POL	Reserved					INPSEL[2:0]			Reserved	INMSEL[2:0]			EN
rw		rw							rw				rw		rw

Bit Field	Name	Description
31:22	Reserved	Reserved, the reset value must be maintained.
21	FLTOUT	The output state after the comparator output is filtered. 0: Output low level 1: Output high level
20	OUT	The output state of the comparator output. 0: Output low level 1: Output high level

19:17	BLKING[2:0]	These bits select which Timer output controls the comparator 7 output blanking. 000: No blanking 001: ATIM1 OC5 selected as blanking source 010: ATIM2 OC5 selected as blanking source 011: ATIM3 OC5 selected as blanking source 100: ATIM8 OC5 selected as blanking source 101: ATIM9 OC5 selected as blanking source 110: ATIM10 OC5 selected as blanking source Other configurations: Reserved

16:14	HYST[2:0]	These bits control the hysteresis level.
		000: No hysteresis;
		001: 10mV hysteresis;
		010: 20mV hysteresis;
		011: 30mV hysteresis;
		100: 40mV hysteresis;
		101: 50mV hysteresis;
		110: 60mV hysteresis;
		111: 70mV hysteresis;

13	POL	This bit is used to invert the comparator output. 0: Output is not inverted; 1: Output is inverted.
12:8	Reserved	Reserved, the reset value must be maintained.
7:5	INPSEL[2:0]	Comparator positive terminal selection bit: 000: PB14 001: PC1 010: PD14 011: dac4_iout 100: dac7_iout 101: VREF_VC3 110: PGA4_OUT_P 111: PGA4_OUT_N Other: Reserved
4	Reserved	Reserved, the reset value must be maintained.
3:1	INMSEL[2:0]	Comparator negative terminal selection bit: 000: PB12 001: PA4 010: PA5 011: PC0 100: PD15 101: dac4_iout 110: dac7_iout 111: VREF_VC3 Other: Reserved

0	EN	This bit switches COMP ON/OFF 0: Comparator disabled; 1: Comparator enabled.

18.8.21 COMP7 Filter Control Register（COMP7_FILC）

Offset address: 0x74

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				DACEN	SAMPW[4:0]					THRESH[4:0]					FILEN
				rw	rw					rw					rw

Bit Field	Name	Description
31:12	Reserved	Reserved, the reset value must be maintained.
11	DACEN	When an external DAC is used as the input of the comparator, the comparator may collect unstable values. Enabling this flag can prevent unstable values from being collected and applied by COMP7. 0: Disable; 1: Enable.
10:6	SAMPW[4:0]	Low-pass filter sampling window size, sampling window = SAMPW + 1
5:1	THRESH[4:0]	The low-pass filter threshold is set. At least the sampling threshold of the opposite state in the sample window can change the output state. This value is required to be greater than SAMPW / 2.
0	FILEN	Filter enable 0: Disable; 1: Enable.

18.8.22 COMP7 Filter Prescaler Register（COMP7_FILP）

Offset address: 0x78

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CLKPSC[15:0]															
rw															

Bit Field	Name	Description
31:16	Reserved	Reserved, the reset value must be maintained.

15:0	CLKPSC[15:0]	Low filter sample clock prescaler,
		System clock divider = CLKPSC + 1.
		0: Every cycle;
		1: Every 2 cycle;
		2: Every 3 cycle;
		…
		65535: Every 65536 cycle;

18.8.23 COMP Window Mode Register（COMP_WINMODE）

Offset address: 0x88

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													CMP56 MD	CMP34 MD	CMP12 MD
													rw	rw	rw

Bit Field	Name	Description
31:3	Reserved	Reserved, the reset value must be maintained.
2	CMP56MD	This bit selects window mode: the two non-inverting inputs of the comparator share the PD12 input. 0: Comparators 5 and 6 are not in window mode, the port output the result of COMP6; 1: Comparators 5 and 6 are in window mode, the port output the XOR result of COMP5 and COMP6.
1	CMP34MD	This bit selects window mode: the two non-inverting inputs of the comparator share the PA0 input. 0: Comparators 3 and 4 are not in window mode, the port output the result of COMP4; 1: Comparators 3 and 4 are in window mode, the port output the XOR result of COMP3 and COMP4.
0	CMP12MD	This bit selects window mode: the two non-inverting inputs of the comparator share the PB10 input. 0: Comparators 1 and 2 are not in window mode, the port output the result of COMP2; 1: Comparators 1 and 2 are in window mode, the port output the XOR result of COMP1 and COMP2。

18.8.24 COMP Lock Register（COMP_LOCK）

Offset address: 0x8C

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									CMP7LK	CMP6LK	CMP5LK	CMP4LK	CMP3LK	CMP2LK	CMP1LK
									rw	rw	rw	rw	rw	rw	rw

Bit Field	Name	Description
31:7	Reserved	Reserved, the reset value must be maintained.
6	CMP7LK	This bit is write-once. It is set by software. It can only be cleared by a system reset. If set it causes COMP7_CTRL, COMP7_FILC, COMP7_FILP registers to be read-only. 0: COMP7_CTRL, COMP7_FILC, COMP7_FILP is read-write. 1: COMP7_CTRL, COMP7_FILC, COMP7_FILP is read-only.
5	CMP6LK	This bit is write-once. It is set by software. It can only be cleared by a system reset. If set it causes COMP6_CTRL, COMP6_FILC, COMP6_FILP registers to be read-only. 0: COMP6_CTRL, COMP6_FILC, COMP6_FILP is read-write. 1: COMP6_CTRL, COMP6_FILC, COMP6_FILP is read-only.
4	CMP5LK	This bit is write-once. It is set by software. It can only be cleared by a system reset. If set it causes COMP5_CTRL, COMP5_FILC, COMP5_FILP registers to be read-only. 0: COMP5_CTRL, COMP5_FILC, COMP5_FILP is read-write. 1: COMP5_CTRL, COMP5_FILC, COMP5_FILP is read-only.
3	CMP4LK	This bit is write-once. It is set by software. It can only be cleared by a system reset. If set it causes COMP4_CTRL, COMP4_FILC, COMP4_FILP registers to be read-only. 0: COMP4_CTRL, COMP4_FILC, COMP4_FILP is read-write. 1: COMP4_CTRL, COMP4_FILC, COMP4_FILP is read-only.
2	CMP3LK	This bit is write-once. It is set by software. It can only be cleared by a system reset. If set it causes COMP3_CTRL, COMP3_FILC, COMP3_FILP registers to be read-only. 0: COMP3_CTRL, COMP3_FILC, COMP3_FILP is read-write. 1: COMP3_CTRL, COMP3_FILC, COMP3_FILP is read-only.
1	CMP2LK	This bit is write-once. It is set by software. It can only be cleared by a system reset. If set it causes COMP2_CTRL, COMP2_FILC, COMP2_FILP registers to be read-only.

		0: COMP2_CTRL, COMP2_FILC, COMP2_FILP is read-write. 1: COMP2_CTRL, COMP2_FILC, COMP2_FILP is read-only.
0	CMP1LK	This bit is write-once. It is set by software. It can only be cleared by a system reset. If set it causes COMP1_CTRL, COMP1_FILC, COMP1_FILP registers to be read-only. 0: COMP1_CTRL, COMP1_FILC, COMP1_FILP is read-write. 1: COMP1_CTRL, COMP1_FILC, COMP1_FILP is read-only.

18.8.25 COMP Interrupt Enable Register（COMP_INTEN）

Offset address: 0x90

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									CMP7IEN	CMP6IEN	CMP5IEN	CMP4IEN	CMP3IEN	CMP2IEN	CMP1IEN
									rw	rw	rw	rw	rw	rw	rw

Bit Field	**Name**	**Description**
31:7	Reserved	Reserved, the reset value must be maintained
6	CMP7IEN	This bit controlled Interrupt enable of COMP7 0: Disable; 1: Enable.
5	CMP6IEN	This bit controlled Interrupt enable of COMP6 0: Disable; 1: Enable.
4	CMP5IEN	This bit controlled Interrupt enable of COMP5 0: Disable; 1: Enable.
3	CMP4IEN	This bit controlled Interrupt enable of COMP4 0: Disable; 1: Enable.
2	CMP3IEN	This bit controlled Interrupt enable of COMP3 0: Disable; 1: Enable.
1	CMP2IEN	This bit controlled Interrupt enable of COMP2 0: Disable; 1: Enable.
0	CMP1IEN	This bit controlled Interrupt enable of COMP1 0: Disable; 1: Enable.

		Note: COMPx_CTRL.OUT high level triggers an interrupt.

18.8.26 COMP Interrupt Status Register（COMP_INTSTS）

Offset address: 0x94

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									CMP7IS	CMP6IS	CMP5IS	CMP4IS	CMP3IS	CMP2IS	CMP1IS
									rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0

Bit Field	Name	Description
31:7	Reserved	Reserved, the reset value must be maintained
6	CMP7IS	This bit indicate the interrupt status of COMP7,write 0 to clear.
5	CMP6IS	This bit indicate the interrupt status of COMP6,write 0 to clear.
4	CMP5IS	This bit indicate the interrupt status of COMP5,write 0 to clear.
3	CMP4IS	This bit indicate the interrupt status of COMP4,write 0 to clear.
2	CMP3IS	This bit indicate the interrupt status of COMP3,write 0 to clear.
1	CMP2IS	This bit indicate the interrupt status of COMP2,write 0 to clear.
0	CMP1IS	This bit indicate the interrupt status of COMP1,write 0 to clear.

18.8.27 COMP Reference Input Voltage Comparison Register（COMP_INVREF）

Offset address: 0x98

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved											VV3TRM[5:1]				
											rw				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VV3TRM[0]	VV3EN	VV2TRM[5:0]						VV2EN	VV1TRM[5:0]						VV1EN
rw	rw	rw						rw	rw						rw

Bit Field	Name	Description
31:21	Reserved	Reserved, the reset value must be maintained.
20:15	VV3TRM[5:0]	Internal comparator 3 reference input voltage comparison VREF level selection 0~0b'111111 corresponds to an output voltage range of 0~VREF+, with a total of 64 levels. For example, 7 represents （7）* VREF+/63 = 1/9 * VREF+
14	VV3EN	Internal comparator 3 reference input voltage comparison enable: 0: Disable;

		1: Enable.
13:8	VV2TRM[5:0]	Internal comparator 2 reference input voltage comparison VREF level selection 0~0b'111111 corresponds to an output voltage range of 0~VREF+, with a total of 64 levels. For example, 7 represents （7）* VREF+/63 = 1/9 * VREF+
7	VV2EN	Internal comparator 2 reference input voltage comparison enable: 0: Disable; 1: Enable.
6:1	VV1TRM[5:0]	Internal comparator 1 reference input voltage comparison VREF level selection 0~0b'111111 corresponds to an output voltage range of 0~VREF+, with a total of 64 levels. For example, 7 represents （7）* VREF+/63 = 1/9 * VREF+
0	VV1EN	Internal comparator 1 reference input voltage comparison enable: 0: Disable; 1: Enable.

18.8.28 COMP Output to Timer Enable Register（COMP_OTIMEN）

Offset address: 0x9C

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									CMP7 OEN	CMP6 OEN	CMP5 OEN	CMP4 OEN	CMP3 OEN	CMP2 OEN	CMP1 OEN
									rw	rw	rw	rw	rw	rw	rw

Bit Field	Name	Description
31:7	Reserved	Reserved, the reset value must be maintained.
6	CMP7OEN	This bit controls the enabling of COMP7 output to the timer function. 0: Disable 1: Enable
5	CMP6OEN	This bit controls the enabling of COMP6 output to the timer function 0: Disable 1: Enable
4	CMP5OEN	This bit controls the enabling of COMP5 output to the timer function 0: Disable 1: Enable
3	CMP4OEN	This bit controls the enabling of COMP4 output to the timer function 0: Disable 1: Enable
2	CMP3OEN	This bit controls the enabling of COMP3 output to the timer function 0: Disable 1: Enable

1	CMP2OEN	This bit controls the enabling of COMP2 output to the timer function 0: Disable 1: Enable
0	CMP1OEN	This bit controls the enabling of COMP1 output to the timer function 0: Disable 1: Enable

19 Digital To Analog Conversion (DAC)

19.1 Introduction

DAC is a digital/analog converter, mainly digital input, voltage output. DAC data can be 8-bit or 12-bit and support DMA functionality. When the DAC is configured in 12-bit mode, the DAC data can be right-aligned or left-aligned. When the DAC is configured in 8-bit mode, the DAC data can be right-aligned.Each DAC has an independent converter and can perform conversions independently. In dual DAC mode, each DAC can perform conversions independently, or two DACs (DAC1 & DAC2, DAC3 & DAC4, DAC5 & DAC6, DAC7 & DAC8 can each form a pair) can simultaneously perform conversions and updates.

VREF+ is input through the pin as the DAC reference voltage, which increases the precision of the DAC conversion data. The built-in VREFBUF can also serve as the DAC's reference voltage. For information about the voltage reference buffer (VREFBUF) section, please refer to the VREFBUF chapter.

When the DAC output is internally connected to external peripherals on the chip, the DACx_OUT pin can be used as a general-purpose input/output (GPIO). The DAC output buffer can be selectively enabled to achieve high driving output current.sion data.

19.2 Main Features

- Support for 8 DACs, each corresponding to an independent DAC converter.
- Support for 8-bit or 12-bit output, with data available in both right-aligned and left-aligned modes in 12-bit mode.
- Dual DAC supports synchronous or independent conversion.
- Each DAC supports DMA functionality and DMA underflow error detection.
- DMA double data mode can save bus bandwidth.
- Noise wave, triangular waveform, sawtooth wave generation.
- DAC output supports connection to on-chip peripherals (COMP, PGA).
- Buffer offset calibration.
- Input reference voltage supports VREF+ and built-in VREFBUF.
- External event-triggered conversion.

19.3 DAC Overview

Table 19-1 DAC Features

DAC Features	DAC1	DAC2	DAC3	DAC4	DAC5	DAC6	DAC7	DAC8
Output buffer	Support				-			
I/O connection	PA4 connected to	PA5 connected to	PA6 connected to	PA7 connected to	-			

	DAC1_OUT	DAC2_OUT	DAC3_OUT	DAC4_OUT	
External output	Support				-
Internal output	Support				Support
Maximum sampling rate	1MSPS				15MSPS

DAC structural diagram and pin description are as follows:

Figure 19-1 Block diagram of DAC

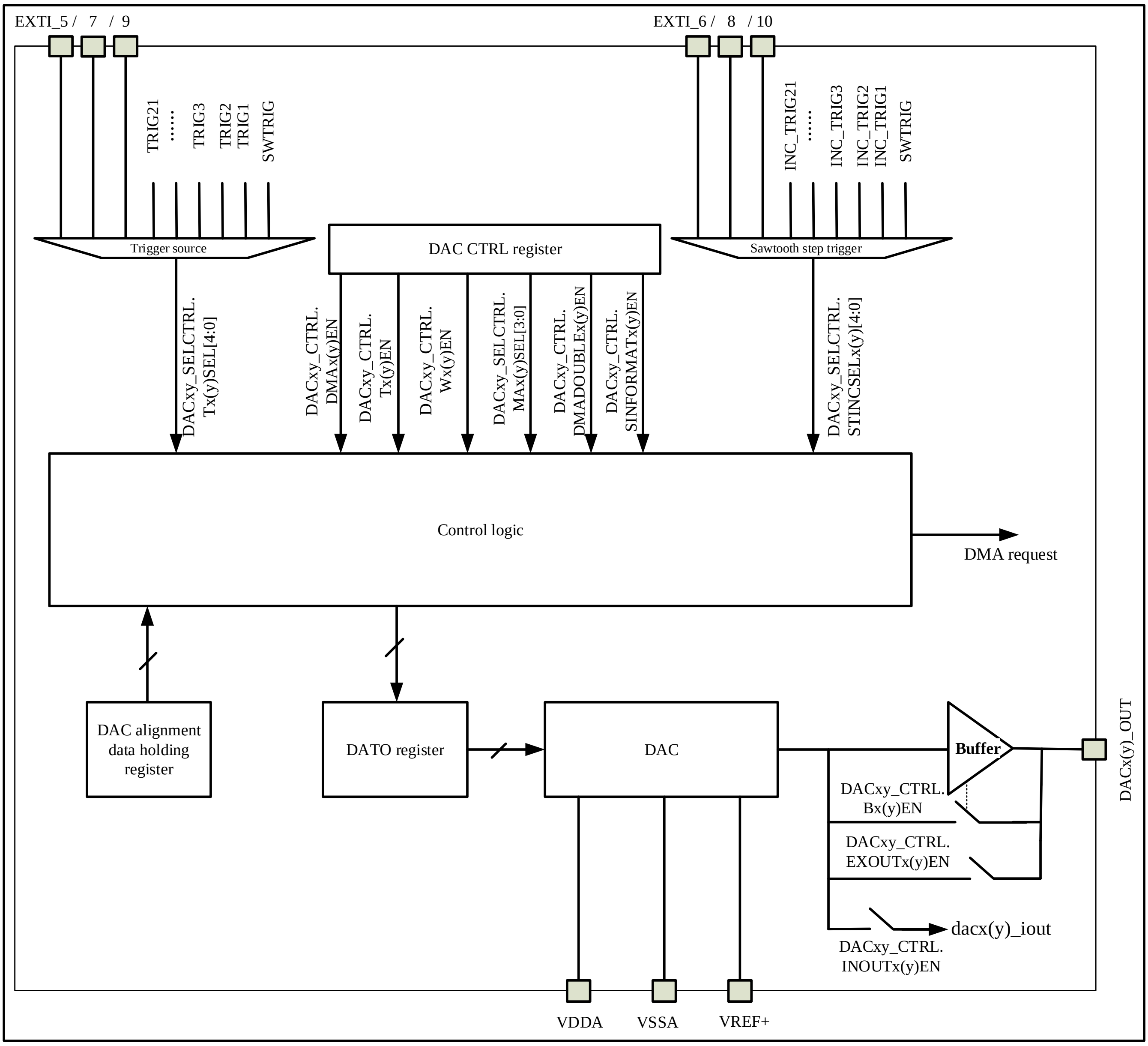


Table 19-2 DAC Pins

Name	Description	Type
V_{REF+}	Positive reference voltage used by the DAC. 1.8V ≤ V_{REF+} ≤ V_{DDA}（3.6V）	input: Positive analog reference voltage

V_{DDA}	Analog power	input: Analog power
V_{SSA}	Analog power ground	input: Analog power ground
DACx(y)_OUT	Analog output from DACx or DACy to external IO	output: Analog output signal
dacx(y)_iout	Analog output from DACx or DACy to internal peripherals	output: Analog output signal

19.4 DAC Function Description And Operation Description

19.4.1 DAC Enable

Powering on the DAC can be done by configuring DAC_CTRL.CHEN = 1. It takes some time for t_{WAKEUP} to open the DAC.

19.4.2 DAC Output Buffer

By configuring DAC_CTRL.Bx(y)EN to disable or enable the output buffer of DAC, if the output buffer is enable, the output impedance is reduced, the driving ability is enhanced, and the external load can be driven without the external operational amplifier.

Table 19-3 DAC1/2/3/4 Output characteristics

Output type	DACxy_CTRL.INOUTx(y)EN	DACxy_CTRL.EXOUTx(y)EN	DACxy_CTRL.Bx(y)EN
Internal output	1	0	0
External output	0	1	0/1
Simultaneous internal and external output	1	1	1

Note: When using simultaneous internal and external output, the output buffer must be enabled (DACxy_CTRL.Bx(y)EN = 1).

19.4.3 DAC Data Format

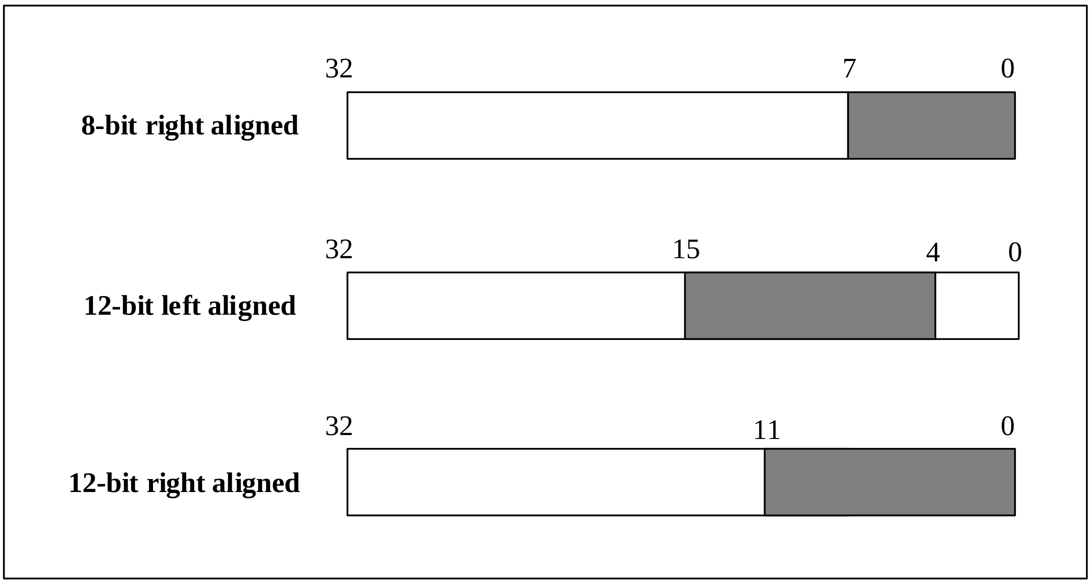
19.4.3.1 Data aligned

When the configuration data is written to the DACx(y)_DR12 register, the data is written to DACx(y)_DR12[11:0], and the 12-bit data is right-aligned. (Actually stored in the register DACx(y)_DHR[11:0] bits, DACx(y)_DHR is the internal data storage register)

When the configuration data is written to the DACx(y)_DL12 register, the data is written to DACx(y)_DL12[15:4], and the 12-bit data is left-aligned. (Actually stored in the register DACx(y)DHR[11:0] bits, DACx(y)_DHR is the internal data storage register)

When the configuration data is written to the DACx(y)_DR8 register, the data is written to DACx(y)_DR8[7:0], and the 8-bit data is right-aligned. (Actually stored in the register DACx(y)DHR[11:4] bits, DACx(y)_DHR is the internal data storage register)

Figure 19-2 Data Register Of Single DAC Channel Mode



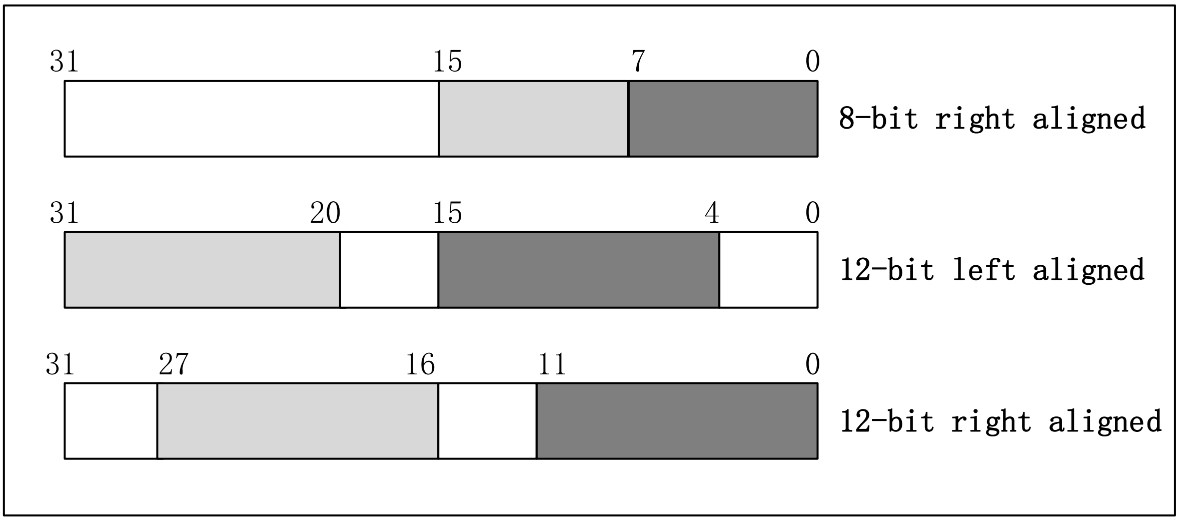
When DAC outputs are synchronized, there are 3 scenarios:

When configuring data writing to the DACxy_DR12D register, DACx data is written to DACxy_DR12D[11:0] (actually stored in the DACx_DHR[11:0] bits, DACx_DHR is the internal data holding register), and DACy data is written to DACxy_DR12D[27:16] (actually stored in the DACy_DHR[11:0] bits, DACy_DHR is the internal data holding register), with 12-bit data right-aligned.

When configuring data writing to the DACxy_DL12D register, DACx data is written to DACxy_DL12D[15:4] (actually stored in the DACx_DHR[11:0] bits, DACx_DHR is the internal data holding register), and DACy data is written to DACxy_DL12D[31:20] (actually stored in the DACy_DHR[11:0] bits, DACy_DHR is the internal data holding register), with 12-bit data left-aligned.

When configuring data writing to the DACxy_DR8D register, DACx data is written to DACxy_DR8D[7:0] (actually stored in the DACx_DHR[11:4] bits, DACx_DHR is the internal data holding register), and DACy data is written to DACxy_DR8D[15:8] (actually stored in the DACy_DHR[11:4] bits, DACy_DHR is the internal data holding register), with 8-bit data left-aligned.

Figure 19-3 Data format when DAC outputs are synchronized



19.4.3.2 Signed/Unsigned Data

The DAC has an unsigned input data range of 0 to 0xFFF.

The DAC can also process signed input data in two's complement format. This is achieved by setting the DACxy_CTRL.SINFORMATx(y) bit. When DACxy_CTRL.SINFORMATx(y) is set to 1, the most significant bit (MSB) of the data written to the holding register is inverted when copied to the data output DACx(y)_DATO register.

Table 19-4 12-bit data format

SINFORMATx(y)-bit	DACx(y)D register value	DACx(y)_DATO register value
0	0x800	0x800
0	0x7FF	0x7FF
0	0xFFF	0xFFF
1	0x000	0x800
1	0x7FF	0xFFF
1	0x800	0x000
1	0xFFF	0x7FF

19.4.4 DAC Trigger

Enabling DAC external trigger can be achieved by configuring DACxy_CTRL.Tx(y)EN = 1, and selecting an external trigger event as the DAC's external trigger source by configuring DACxy_SELCTRL.Tx(y)SEL[4:0].

These events can also be software-triggered or hardware-triggered, as shown in the table below:

Table 19-5 DAC External Trigger

Trigger source	Type	Tx(y)SEL[4:0]
ATIM1 TRGO events	Internal signal from the on-chip timer	00001
ATIM2 TRGO events		00010
ATIM3 TRGO events		00011
Reserved		00100

Trigger source	Type	Tx(y)SEL[4:0]
GTIM1 TRGO events		00101
GTIM2 TRGO events		00110
GTIM3 TRGO events		00111
GTIM4 TRGO events		01000
GTIM5 TRGO events		01001
GTIM6 TRGO events		01010
GTIM7 TRGO events		01011
GTIM8 TRGO events		01100
GTIM9 TRGO events		01101
GTIM10 TRGO events		01110
SHRTIM Reset_TRG1 events		10010
SHRTIM Reset_TRG2 events		10011
SHRTIM Reset_TRG3 events		10100
SHRTIM Reset_TRG4 events		10101
SHRTIM Reset_TRG5 events		10110
SHRTIM Reset_TRG6 events		10111
SHRTIM TRG1 events		11000
EXTI line 5	External pins	01111
EXTI line 7		10000
EXTI line 9		10001
SWTRIG (Software Triggered)	Software control bit	00000

When the DAC's trigger source is a timer output or rising edge of an EXTI line, upon trigger generation, the data held in the alignment data holding register will be transferred to the DACx(y)_DATO register. This data transfer process requires (0.5*PCS) DAC_CLK cycles or 3/5/7 bus_clk cycles (depending on the value of DACxy_HFSEL[1:0]) clock cycles of time.

If the user selects software trigger as the trigger source, by configuring DACxy_SOTTR.TRx(y)EN = 1, the data held in the alignment data holding register will be transferred to the DACx(y)_DATO register. The DACxy_SOTTR.TRx(y)EN bit will automatically clear to 0 by hardware after data transfer.

The step trigger source for sawtooth wave generation can be selected using the DACxy_SELCTRL.STINCSELx(y)[4:0] control bits.

The specific trigger sources are shown in the table below:

Table 19-6 DAC sawtooth wave step trigger source signals

Trigger source	Type	STINCSELx(y)[4:0]
ATIM1 TRGO events	Internal signal from the on-chip timer	00001
ATIM 2 TRGO events		00010
ATIM 3 TRGO events		00011
Resrved		00100
GTIM1 TRGO events		00101

Trigger source	Type	STINCSELx(y)[4:0]
GTIM 2 TRGO events		00110
GTIM 3 TRGO events		00111
GTIM 4 TRGO events		01000
GTIM 5 TRGO events		01001
GTIM 6 TRGO events		01010
GTIM 7 TRGO events		01011
GTIM 8 TRGO events		01100
GTIM 9 TRGO events		01101
GTIM 10 TRGO events		01110
SHRTIM Reset_TRG1 events		10010
SHRTIM Reset_TRG2 events		10011
SHRTIM Reset_TRG3 events		10100
SHRTIM Reset_TRG4 events		10101
SHRTIM Reset_TRG5 events		10110
SHRTIM Reset_TRG6 events		10111
EXTI line 6	External pins	01111
EXTI line 8		10000
EXTI line 10		10001
SWTRIG (Software Triggered)	Software control bit	00000

Notes:

(1) Changing the DACxy_SELCTRL.Tx(y)SEL[4:0] bits is prohibited while the DAC is enabled.

(2) When using software trigger, the transfer of data from the alignment data holding register to the DACx(y)_DATO register takes 1 bus_clk (where bus_clk is either pclk or hclk depending on the clock bus the DAC is connected to) clock cycle.

(3) SHRTIM, ATIM, GTIM, and DAC operate in different clock domains, crossing clock domains can lead to metastability. If the DAC step trigger rate is too close to the DAC operating clock, metastability may occur, resulting in overshoot, and overshoot can cause the loss of step triggers at the overshoot point.

19.4.5 DAC Conversion

Users cannot directly operate the DACx(y)_DATO register; writing to the DHR (including DACx(y)_DR8, DACx(y)_DR12, DACx(y)_DL12, DACxy_DR8D, DACxy_DR12D, DACxy_DL12D) is required for the changes to take effect.

It takes time for the data to transfer from the DHR to the DACx(y)_DATO register (referred to as DATO), as detailed in the following table:

Table 19-7 DAC1-DAC4 Conversion Description

Tx(y)EN	Wx(y)EN	Tx(y)SEL[4:0]	DHR -> DATO(bus_clk)	From DATO to the time T1 for establishing

0	0b’000	00000	2	HFSEL = 0:T1 is (PCS[7:0]/2-1) bus_clk[1];
				HFSEL = 1:T1 is 1 bus_clk;
				HFSEL = 2:T1 is 3 bus_clk;
				HFSEL = 3:T1 is 5 bus_clk;

1	0b’000	00000	4	
	0b’000	non-0b’000	4	
	non-0b’000	-	3	

Table 19-8 DAC5-DAC8 Conversion Description

Tx(y)EN	Wx(y)EN	Tx(y)SEL[4:0]	DHR -> DATO(bus_clk)	From DATO to the time T1 for establishing
0	0b’000	00000	2	T1 fixed 5 bus_clk.
1	0b’000	00000	4	
	0b’000	non-0b’000	4	
	non-0b’000	-	3	

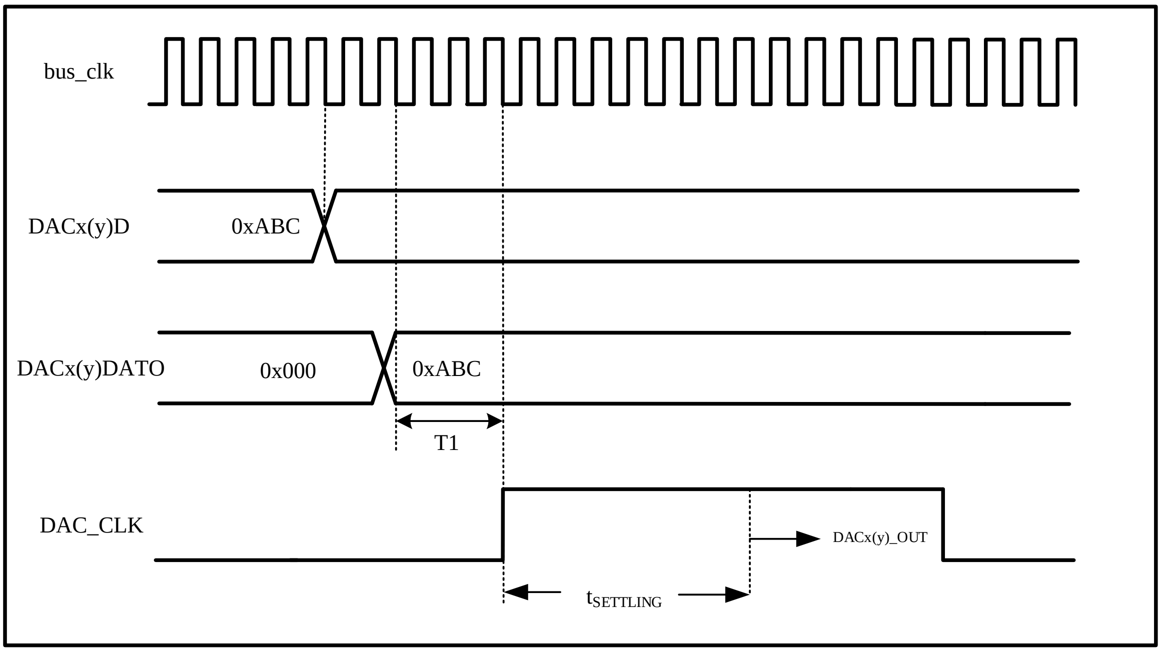
Note: 1. bus_clk is the bus clock to which the DAC is attached, DAC1-DAC4 are attached to the APB bus, and DAC5/6/7/8 are attached to the AHB bus.

As for DAC1-DAC4, When the bus_clk clock speed exceeds 80 MHz, the DACxy_HFSEL[1:0] bits must be set. This will add additional delay to the transfer from the DACx(y)_DHR register to the DACx(y)_DATO register.

If data is updated or a software/hardware trigger event occurs during a prohibited time period, the behavior of the peripheral will be affected. Updating data or triggering events during this period will lead to unpredictable behavior of the peripheral.

The effective output of the DAC needs to go through the following stages. Firstly, the DAC transfers data from the data holding register to the DACx(y)_DATO register. Then, after a delay of time T1, the analog output conversion is initiated from the DACx(y)_DATO register. Finally, the output becomes valid after a settling time $t_{SETTLING}$, which is related to the power supply voltage and the analog output load.

Figure 19-4 Timing diagram for conversion when trigger is disabled(HFSEL = 0b'10)



19.4.6 DAC Output Voltage

The digital input is converted to analog voltage output by a DAC module in a linear relationship ranging from 0 to V_{REF+}. The output voltage of DAC is calculated as follows:

DAC output = V_{REF+} * (DATO / 4095).

19.4.7 DMA Requests

To enable DMA functionality, configure DACxy_CTRL.DMAx(y)EN = 1. When an external trigger occurs (not a software trigger), a DMA request is generated, and subsequently, the data from the alignment data holding register is transferred to the DACx(y)_DATO register.

In dual DAC mode, if both DACxy_CTRL.DMAx(y)EN bits are set, two DMA requests will be generated. If only one DMA request is needed, then set the DMAxEN bit. This allows applications to manage two DACs using a single DMA request.

Since the DMA request occurs after the data is loaded into the data holding register and transferred to the DACx(y)_DATO register, users must first write data to the data holding register before triggering the DAC conversion.

19.4.7.1 DMA Underflow

DAC's DMA requests are not queued, so if a second external trigger arrives before the acknowledgment of the first external trigger (the first request), no new request will be issued. Instead, the DACxy_STS.DMAUDRx(y) flag will be set to report an error condition. The DAC continues to convert old data. Therefore, users must set an appropriate DAC trigger frequency to reduce the workload on DMA and avoid DMA underflow.

When a DMA underflow occurs, users can clear the DACxy_STS.DMAUDRx(y) flag by writing a 1, then disable DMA, and reinitialize both DMA and DAC to correctly restart the transfer.

If the user sets the DACxy_CTRL.DMAUDRx(y)IEN bit, a DMA underflow interrupt will be generated when a DMA underflow event occurs.

19.4.7.2 DMA double data mode

During DMA operation, a single DMA request can only transfer 8-bit or 12-bit effective data due to the 32-bit AHB bandwidth, which greatly wastes DMA bandwidth. Therefore, it is possible to consider transmitting two 12-bit data simultaneously to save DMA transfer efficiency. Setting DACxy_CTRL.DMADOUBLEx(y) to 1 enables DMA double data mode.

When the DACxy_CTRL.DMAx(y)EN bit is set, a DAC DMA request is generated every two external triggers (except for software triggers).

1. When the first trigger event is detected, the value of the DACx(y)DB register will be transferred to the DACx(y)_DATO register. A DMA request is then generated, and new data is written into the DACx(y) D and DACx(y) DB data registers by DMA.
2. When the next trigger event is detected, the value of the DACx(y)D register will be transferred to the DACx(y)_DATO register. The second trigger event will not generate any DMA request.
3. When the next trigger event is detected, the value of the DACx(y)DB register will be transferred to the DACx(y)_DATO register. A DMA request is then generated, and new data is written into the DACx(y) D and DACx(y)DB data registers by DMA.

In DMA double data mode, the DAC also supports DMA underflow function.

In DMA double data mode, a DMA request can only handle one DAC. When using DMA double data mode in dual DAC mode, where both DACs are simultaneously using DMA double data, users need to separately configure two independent DMA channels.

To switch from double data mode to single data mode, or from single data mode to double data mode, the following conditions must be met:

- DAC must be in the off state. (DACxy_CTRL.DACx(y)EN = 0)
- DMA enable must be in the off state (DACxy_CTRL.DMAx(y)EN = 0)

19.4.8 Noise Generation

DAC can generate noise by configuring DACxy_CTRL.Wx(y)EN[2:0] as 3b'001 to enable the noise function. By configuring DACxy_SELCTRL.MAx(y)SEL[3:0] to select which bits of the linear feedback shift register (LFSR) to mask, the value of the LFSR register is added to the value of the aligned data hold register and then written to the DACx(y)_DATO register (overflow bits are discarded). The initial value of the LFSR is 0xAAA, and the value of the LFSR is updated three bus_clk clock cycles after the trigger event occurs.

Figure 19-5 DAC LFSR Algorithm

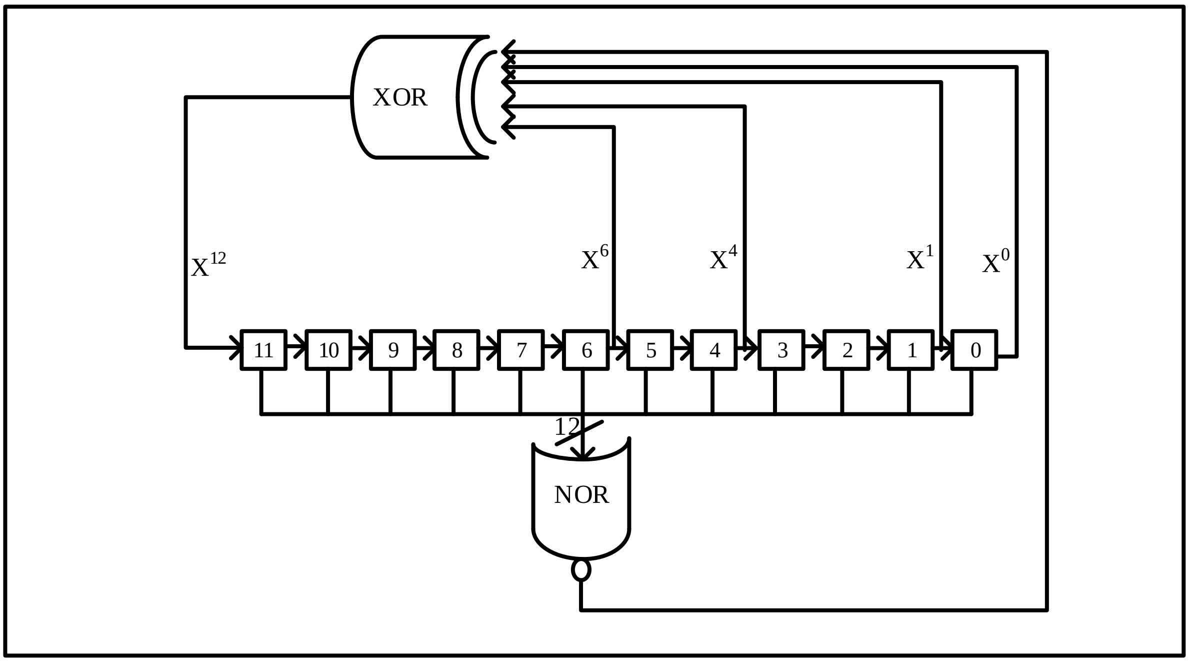
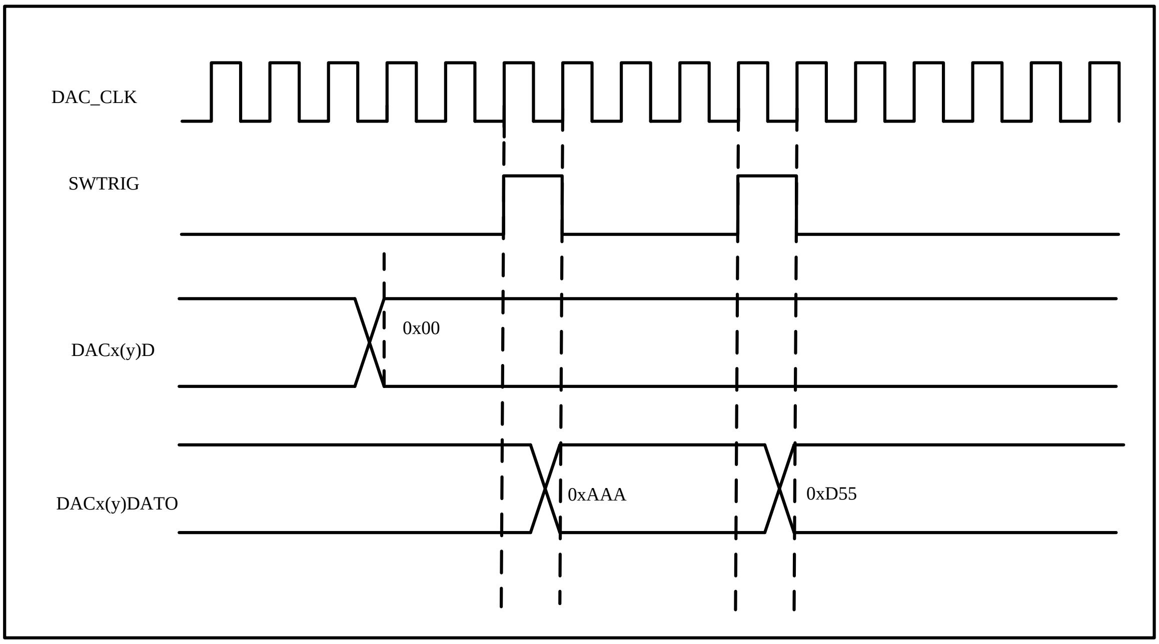


Figure 19-6 DAC Conversion with LFSR Waveform Generation (Enable Software Trigger)



Note: DAC must be configured for trigger enable (DACxy_CTRL.Tx(y)EN = 1) to generate noise.

19.4.9 Triangular Wave Generation

DAC can generate a triangular wave by configuring DACxy_CTRL.Wx(y)EN[2:0] as 3b'010 to enable the triangular wave function. By configuring DACxy_SELCTRL.MAx(y)SEL[3:0] to select the amplitude of the triangular wave, the internal triangular wave counter value is added to the value of the aligned data hold register and then written to the DACx(y)_DATO register (overflow bits are discarded). The triangular wave counter value is updated three bus_clk cycles after the trigger event occurs, and the triangular wave counter will accumulate to the set maximum amplitude, then decrement to 0, repeating this cycle.

Figure 19-7 DAC Triangular Wave Generation

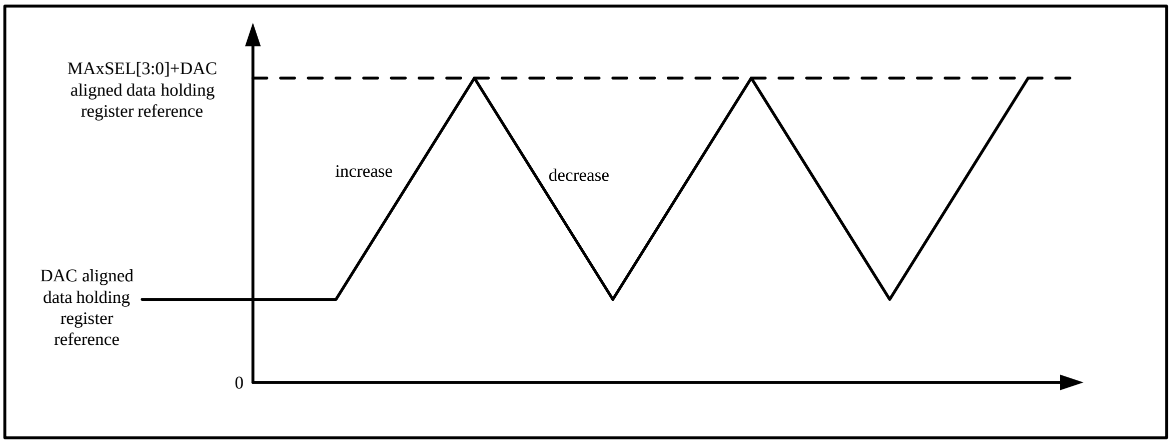
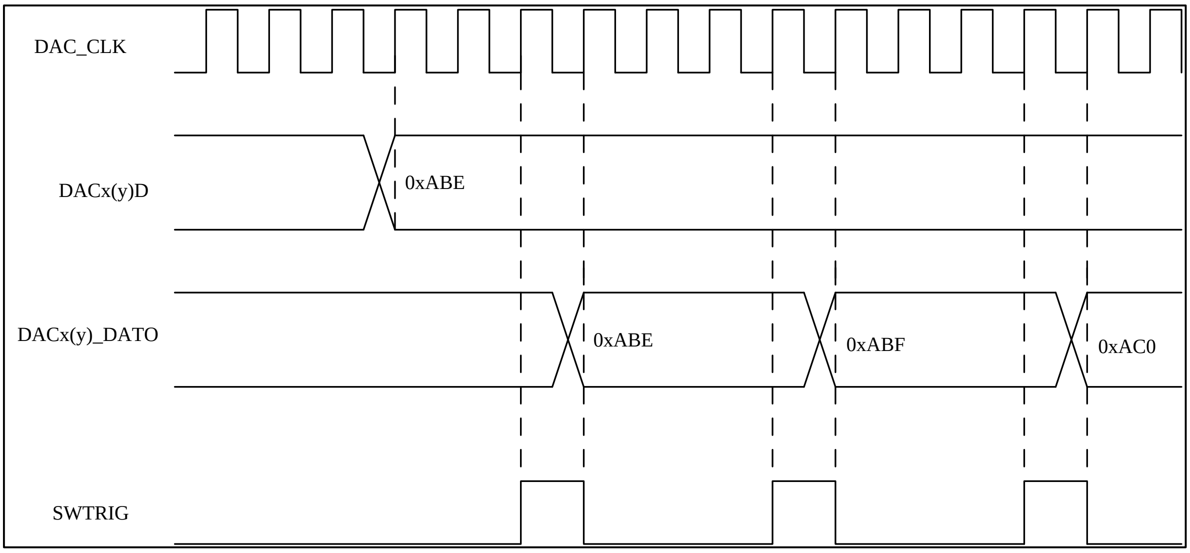

MAxSEL[3:0]+DAC aligned data holding register reference
increase
decrease
DAC aligned data holding register reference
0

Figure 19-8 DAC Conversion with Triangular Wave Generation (Enable Software Trigger, HFSEL[1:0] = 0b'01)



Notes:

(1) DAC must be configured for trigger enable (DACxy_CTRL.Tx(y)EN = 1) to generate a triangular wave;

(2) It is not allowed to set DACxy_SELCTRL.MAx(y)SEL[3:0] after enabling the DAC.

19.4.10 Sawtooth Wave Generation

The steps for generating a sawtooth wave are as follows:

1. Configure DACxy_CTRL.Wx(y)EN[2:0] as 3b'100 or 3b'110 to select either an increasing sawtooth wave or a decreasing sawtooth wave;
2. Set the values of DACxy_SELCTRL.Tx(y)SEL[4:0] and DACxy_SELCTRL.STINCSELx(y)[4:0] to configure the reset trigger source and step trigger source for the sawtooth wave respectively;

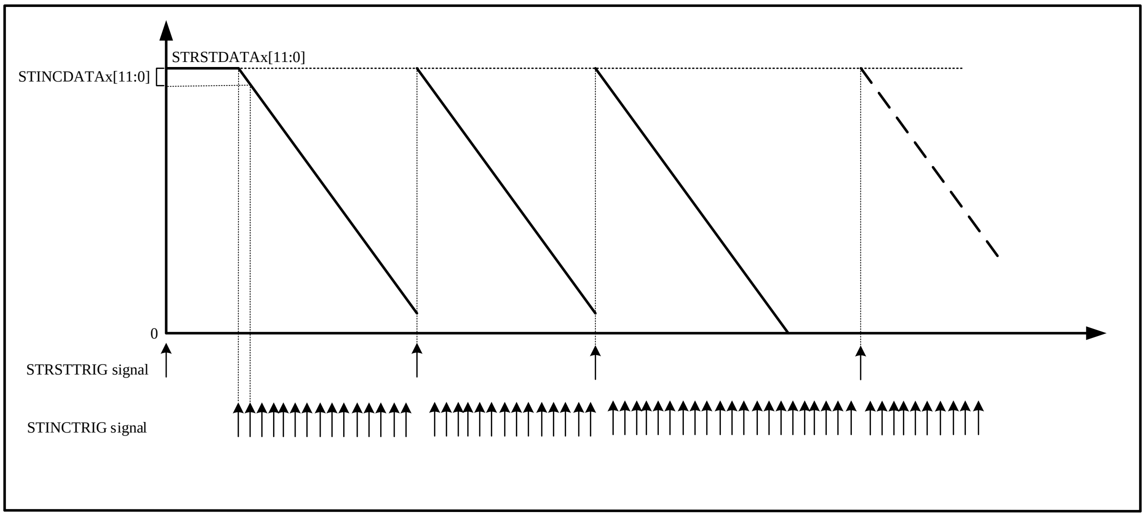
3. Configure the reset value of the sawtooth wave by setting DACxy_STRST.STRSTDATAx(y)[11:0];

4. Configure the step value of the sawtooth wave by setting DAC_STRST.STINCDATAx(y)[11:0].

When configured as an increasing sawtooth wave, the sawtooth wave starts counting from DAC_STRST.STRSTDATAx[11:0]. With each step trigger, the counter increases by DAC_STRST.STINCDATAx(y)[11:0]. When the counter reaches the saturation value of 0xFFF, another step trigger will put the counter in a saturated state. Upon a reset trigger, the counter will reset to DACxy_STRST.STRSTDATAx(y)[11:0].

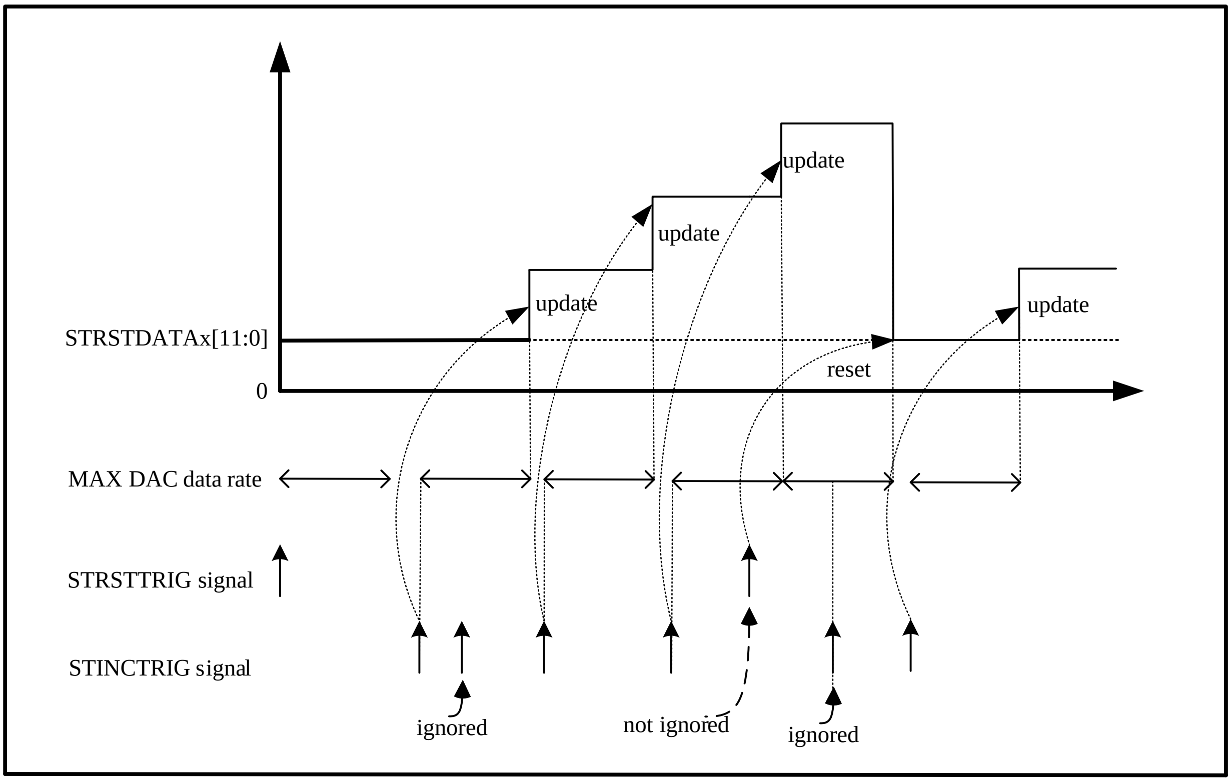
When configured as a decreasing sawtooth wave, the sawtooth wave starts counting from DACxy_STRST.STRSTDATAx(y)[11:0]. With each step trigger, the counter decreases by DAC_STRST.STINCDATAx(y)[11:0]. When the counter reaches the saturation value of 0x000, another step trigger will put the counter in a saturated state. Upon a reset trigger, the counter will reset to DACxy_STRST.STRSTDATAx(y)[11:0].

Figure 19-9 Decreasing Sawtooth Wave Generation



The reset trigger signal STRSTTRIG for the sawtooth wave has higher priority than the step trigger signal STINCTRIG, and the external trigger rate for the sawtooth wave must not exceed the update rate of the DAC. If the step trigger signal DACxy_SELCTRL.STINCSELx(y)[4:0] is higher than the DAC's update rate, the trigger signal will be ignored. If a reset trigger signal STRSTTRIG is suddenly applied after responding to the step trigger signal STINCTRIG, STRSTTRIG will be paused. Then, a reset trigger will be immediately responded to after data increment or decrement.

Figure 19-10 DAC Increasing Sawtooth Wave Reset Signal Trigger Priority over Step Trigger



19.4.11 DAC Buffer Calibration

The formula for an N-bit DAC converter is as follows:

$$V_{OUT} = (\mathrm{Din}/2^N) \times \mathrm{Gain} \times V_{REF} + V_{OFFSET}$$

Where: VOUT is the analog voltage output, Din is the digital input, Gain is the gain error, VREF is the full-scale voltage, VOFFSET is the offset voltage. For an ideal DAC, Gain = 1, VOFFSET = 0.

Due to the output buffer characteristics, voltage offsets may differ between different components and introduce absolute offset errors on the analog output. To compensate for this voltage offset (VOFFSET), calibration is required through trimming techniques. Calibration is only effective when DACx is operating with the buffer enabled (BxEN = 0b'1). It is ineffective if applied in other modes with the buffer disabled. During the calibration process:

- Output disconnected from external pins or internal connections.
- Buffer acting as a comparator, sensing the mid-code value 0x800, and comparing it with the VREF+/2 signal through an internal bridge

Conducting a comparison, and then based on the comparison result (DACxy_SR.CALFLAGx(y) bit), switching its output signal to 0 or 1.

There are two calibration modes: factory calibration and user calibration, users can choose

1. Factory calibration:

   This calibration value is set during manufacturing, and upon DAC reset, it will be loaded into the DACxy_CALC.OTRIMx(y)[4:0] register.

2. User calibration:

   User trimming can be performed under operating conditions different from the nominal factory trimming conditions, especially when VDDA voltage, temperature, and VREF+ values change. User trimming can be done at any point in the application software. The factory calibration environment can be referenced in the datasheet.

The user calibration steps are as follows:

1） Enable DAC calibration (DACxy_CTRL.CALx(y)EN = 1)；

2） Enable the buffer mode for the DAC (DACxy_CTRL.Bx(y)EN = 1);

3）Enable the DAC (DACxy_CTRL.DAC(y)EN = 1)；

4）Apply calibration algorithm:

a) Write 00000b to DACxy_CALC.OTRIMx(y)[4:0];

b) Wait for at least 100us and check if the status register DACxy_STS.CALFLAGx(y) bit is set to 1?

c) If the status is set to 1, end the calibration; otherwise, write 00001b to DACxy_CALC.OTRIMx(y)[4:0] and repeat the above steps.

Software algorithms can use successive approximation or binary search techniques to quickly calculate and set the contents of DACxy_CALC.OTRIMx(y)[4:0] bits. The toggling of the DACxy_STS.CALFLAGx(y) bit indicates that the offset has been correctly compensated, and the corresponding trim code must be retained in the DACxy_CALC.OTRIMx(y)[4:0] bits.

If during device operation VDDA, VREF+, and temperature conditions remain unchanged but it more frequently enters STANDBY or VBAT mode, the software can store the DACxy_CALC.OTRIMx(y)[4:0] bits found during the initial user calibration in flash or backup registers. Then, upon power recovery of the device, these bits can be directly loaded/written to avoid waiting for a new calibration time. When the DACxy_CTRL.CALx(y)EN bit is set, the DACxy_CTRL.DACx(y)EN bit must be set to enable the calibration function.

19.5 Dual DAC Conversion Operation

The two DACs can operate independently or simultaneously. In this mode, there are three registers available: DACxy_DR12D, DACxy_DL12D, and DACxy_DR8D, which can efficiently utilize the bus bandwidth, allowing simultaneous operation on both DAC channels with each register.

When both DACs are enabled for conversion simultaneously, there are a total of 15 modes available. In the case of using only one DAC for conversion, the other DAC can still operate independently. Please refer to the description in the following section for details.

19.5.1 Independent Trigger without Waveform Generator

Configuration process as follows:

- Enable DACx and DACy trigger by configuring DACxy_CTRL.TxEN and DACxy_CTRL.TyEN.
- Select different trigger sources by setting DAC_SELCTRL.TxSEL[4:0] and DAC_SELCTRL.TySEL[4:0] to different values.
- Choose not to use the waveform generator by setting DACxy_CTRL.WxEN[2:0] and DACxy_CTRL.WyEN[2:0] to "000".
- Place the desired data for conversion into the corresponding aligned data hold registers.

When a trigger event occurs for DACx, the value of the aligned data hold register will be transferred to the DACx_DATO register after a delay of 3 bus_clk clock cycles. When a trigger event occurs for DACy, the value of the aligned data hold register will be transferred to the DACy_DATO register after a delay of 3 bus_clk clock cycles.

19.5.2 Independent Trigger for Generating Identical Noise

Configuration process as follows:

- Configure DACxy_CTRL.TxEN and DACxy_CTRL.TyEN to enable DACx and DACy trigger.
- Configure DAC_SELCTRL.TxSEL[4:0] and DAC_SELCTRL.TySEL[4:0] with different values to select different trigger sources.
- Configure DACxy_CTRL.WxEN[2:0] and DACxy_CTRL.WyEN[2:0] as "001" to enable noise generation.
- Configure DAC_SELCTRL.MAxSEL[3:0] and DAC_SELCTRL.MAySEL[3:0] with the same value to obtain

the same LFSR register mask bit.

- Place the desired conversion data into the corresponding aligned data holding registers.

When a trigger event is generated by DACx, the counter value of LFSR register 1 is added to the value of the corresponding data holding register. This sum is then passed to the register DACx_DATO after a delay of 3 bus_clk clock cycles, and the counter value of LFSR register 1 is updated at this time. When a trigger event is generated by DACy, the counter value of LFSR register 2 is added to the value of the corresponding data holding register. This sum is then passed to the register DACy_DATO after a delay of 3 bus_clk clock cycles, and the counter value of LFSR register 2 is updated at this time.

19.5.3 Generating Independent Triggers for Different Noises

Configuration process as follows:

- Configure DACxy_CTRL.TxEN and DACxy_CTRL.TyEN to enable DACx and DACy trigger.
- Configure DAC_SELCTRL.TxSEL[4:0] and DAC_SELCTRL.TySEL[4:0] with different values to select different trigger sources.
- Configure DACxy_CTRL.WxEN[2:0] and DACxy_CTRL.WyEN[2:0] as "001" to enable noise generation.
- Configure DACxy_SELCTRL.MAxSEL[3:0] and DACxy_SELCTRL.MAySEL[3:0] with different values to obtain different LFSR register mask bits.
- Place the desired conversion data into the corresponding aligned data holding registers.

When a trigger event is generated by DACx, the counter value of LFSR register 1 is added to the value of the corresponding data holding register. This sum is then passed to the register DACx_DATO after a delay of 3 bus_clk clock cycles, and the counter value of LFSR register 1 is updated at this time. When a trigger event is generated by DACy, the counter value of LFSR register 2 is added to the value of the corresponding data holding register. This sum is then passed to the register DACy_DATO after a delay of 3 bus_clk clock cycles, and the counter value of LFSR register 2 is updated at this time.

19.5.4 Generating Independent Triggers for the Same Triangle Wave

Configuration process as follows:

- Configure DACxy_CTRL.TxEN and DACxy_CTRL.TyEN to enable DACx and DACy trigger.
- Configure DAC_SELCTRL.TxSEL[4:0] and DAC_SELCTRL.TySEL[4:0] with different values to select different trigger sources.
- Configure DACxy_CTRL.WxEN[2:0] and DACxy_CTRL.WyEN[2:0] as "010" to enable triangle wave generation.
- Configure DACxy_SELCTRL.MAxSEL[3:0] and DACxy_SELCTRL.MAySEL[3:0] with the same value to obtain the same triangle wave amplitude.
- Place the desired conversion data into the corresponding aligned data holding registers.

When a trigger event is generated by DACx, the triangle wave amplitude is added to the value of the corresponding data holding register. This sum is then passed to the register DACx_DATO after a delay of 3 bus_clk clock cycles, and the counter value of DACx's triangle wave is updated at this time. When a trigger event is generated by DACy, the triangle wave amplitude is added to the value of the corresponding data holding register. This sum is then passed to the register DACy_DATO after a delay of 3 bus_clk clock cycles, and the counter value of DACy's triangle wave is updated at this time.

19.5.5 Generating Independent Triggers for Different Triangle Waves

Configuration process as follows:

- Configure DACxy_CTRL.TxEN and DACxy_CTRL.TyEN to enable DACx and DACy trigger.
- Configure DAC_SELCTRL.TxSEL[4:0] and DAC_SELCTRL.TySEL[4:0] with different values to select different trigger sources.
- Configure DACxy_CTRL.WxEN[2:0] and DACxy_CTRL.WyEN[2:0] as "010" to enable triangle wave generation.
- Configure DACxy_SELCTRL.MAxSEL[3:0] and DACxy_SELCTRL.MAySEL[3:0] with different values to obtain different triangle wave amplitudes.
- Place the desired conversion data into the corresponding aligned data holding registers.

When a trigger event is generated by DACx, the triangle wave amplitude is added to the value of the corresponding data holding register. This sum is then passed to the register DACx_DATO after a delay of 3 bus_clk clock cycles, and the counter value of DACx's triangle wave is updated at this time. When a trigger event is generated by DACy, the triangle wave amplitude is added to the value of the corresponding data holding register. This sum is then passed to the register DACy_DATO after a delay of 3 bus_clk clock cycles, and the counter value of DACy's triangle wave is updated at this time.

19.5.6 Generating Independent Triggers for the Same Sawtooth Wave

Configuration process as follows:

- Configure DACxy_CTRL.TxEN and DACxy_CTRL.TyEN to enable DACx and DACy trigger.
- Configure DAC_SELCTRL.TxSEL[4:0] and DAC_SELCTRL.TySEL[4:0] with different values to select different reset trigger sources.
- Configure DAC_SELCTRL.STINCSELx[4:0] and DAC_SELCTRL.STINCSELy[4:0] with different values to select different increment trigger sources.
- Configure DAC_STINC.STINCDATAx[11:0] and DAC_STINC.STINCDATAy[11:0] with the same value as the step value for the sawtooth wave.
- Configure DAC_STRST.STRSTDATAx[11:0] and DAC_STRST.STRSTDATAy[11:0] with the same value as the reset value for the sawtooth wave.
- Configure DACxy_CTRL.WxEN[2:0] and DACxy_CTRL.WyEN[2:0] as "100" or "110" to enable sawtooth wave generation.
- Enable the corresponding external triggers.

When a reset event is triggered by DACx, the sawtooth wave will pass the value of

DAC_STRST.STRSTDATAx[11:0] register to the DACx_DATO register after a delay of 3 bus_clk clock cycles, updating the counter value of DACx's sawtooth wave at that time. When an increment event is triggered, the counter of the sawtooth wave will increase or decrease by DAC_STINC.STINCDATAx[11:0], and the result of the operation will be passed to the DACx_DATO register after a delay of 3 bus_clk clock cycles. When a reset event is triggered by DACy, the sawtooth wave will pass the value of DAC_STRST.STRSTDATAy[11:0] register to the DACy_DATO register after a delay of 3 bus_clk clock cycles, updating the counter value of DACy's sawtooth wave at that time. When an increment event is triggered, the counter of the sawtooth wave will increase or decrease by DAC_STINC.STINCDATAy[11:0], and the result of the operation will be passed to the DACy_DATO register after a delay of 3 bus_clk clock cycles.

19.5.7 Generating Independent Triggers for Different Sawtooth Waves

Configuration process as follows:

- Configure DACxy_CTRL.TxEN and DACxy_CTRL.TyEN to enable DACx and DACy trigger.
- Configure DAC_SELCTRL.TxSEL[4:0] and DAC_SELCTRL.TySEL[4:0] with different values to select different reset trigger sources.
- Configure DAC_SELCTRL.STINCSELx [4:0] and DAC_SELCTRL.STINCSELy[4:0] with different values to select different increment trigger sources.
- Configure DAC_STINC.STINCDATAx [11:0] and DAC_STINC.STINCDATAy[11:0] with different values as the step value for the sawtooth wave.
- Configure DAC_STRST.STRSTDATAx[11:0] and DAC_STRST.STRSTDATAy[11:0] with different values as the reset value for the sawtooth wave.
- Configure DACxy_CTRL.WxEN[2:0] and DACxy_CTRL.WyEN[2:0] as "100" or "110" to enable sawtooth wave generation.
- Enable the corresponding external triggers.

When a reset event is triggered by DACx, the sawtooth wave will pass the value of DAC_STRST.STRSTDATAx[11:0] register to the DACx_DATO register after a delay of 3 bus_clk clock cycles, updating the counter value of DACx's sawtooth wave at that time. When an increment event is triggered, the counter of the sawtooth wave will subtract DAC_STINC.STINCDATAx[11:0], and the result of the operation will be passed to the DACx_DATO register after a delay of 3 bus_clk clock cycles. When a reset event is triggered by DACy, the sawtooth wave will pass the value of DAC_STRST.STRSTDATAy[11:0] register to the DACy_DATO register after a delay of 3 bus_clk clock cycles, updating the counter value of DACy's sawtooth wave at that time. When an increment event is triggered, the counter of the sawtooth wave will subtract DAC_STINC.STINCDATAy[11:0], and the result of the operation will be passed to the DACy_DATO register after a delay of 3 bus_clk clock cycles.

19.5.8 Simultaneous Software Startup

Configuration process as follows:

- Set DAC_SELCTRL.TxSEL[4:0] and DAC_SELCTRL.TySEL[4:0] to "00000" to choose software as the trigger source.

- Enable DACxy_CTRL.TxEN and DACxy_CTRL.TyEN to activate trigger enable for DACx and DACy.
- Place the required conversion data into the corresponding aligned data holding registers.
- Enable DAC_SOTTR.TRxEN and DAC_SOTTR.TRyEN to trigger software conversion.

The value of DACx's aligned data holding register will be passed to the DACx_DATO register after a delay of 1 bus_clk clock cycle.

The value of DACy's aligned data holding register will be passed to the DACy_DATO register after a delay of 1 bus_clk clock cycle.

19.5.9 Synchronous Trigger without Using Waveform Generators

Configuration process as follows:

- Configure DACxy_CTRL.TxEN and DACxy_CTRL.TyEN to enable DACx and DACy trigger.
- Set DAC_SELCTRL.TxSEL[4:0] and DAC_SELCTRL.TySEL[4:0] to the same value to select the same trigger source.
- Set DACxy_CTRL.WxEN[2:0] and DACxy_CTRL.WyEN[2:0] as "000" to choose not to use waveform generators.
- Place the desired conversion data into the corresponding aligned data holding registers.

When a trigger event occurs, the value of DACx's aligned data holding register will be passed to the DACx_DATO register after a delay of 3 bus_clk clock cycles; the value of DACy's aligned data holding register will be passed to the DACy_DATO register after a delay of 3 bus_clk clock cycles.

19.5.10 Generating Synchronous Triggers for the Same Noise

Configuration process as follows:

- Configure DACxy_CTRL.TxEN and DACxy_CTRL.TyEN to enable DACx and DACy trigger.
- Set DAC_SELCTRL.TxSEL[4:0] and DAC_SELCTRL.TySEL[4:0] to the same value to select the same trigger source.
- Set DACxy_CTRL.WxEN[2:0] and DACxy_CTRL.WyEN[2:0] as "001" to enable noise generation.
- Set DACxy_SELCTRL.MAxSEL[3:0] and DACxy_SELCTRL.MAySEL[3:0] to the same value to obtain the same LFSR register mask bits.
- Place the desired conversion data into the corresponding aligned data holding registers.

When a trigger event occurs, the counter value of LFSR register 1 is added to the value of the corresponding data holding register. This sum is then passed to the register DACx_DATO after a delay of 3 bus_clk clock cycles, and the counter value of LFSR register 1 is updated at this time. Similarly, the counter value of LFSR register 2 is added to the value of the corresponding data holding register when a trigger event occurs. This sum is then passed to the register DACy_DATO after a delay of 3 bus_clk clock cycles, and the counter value of LFSR register 2 is updated at this time.

19.5.11 Generating Synchronous Triggers for Different Noises

Configuration process as follows:

- Configure DACxy_CTRL.TxEN and DACxy_CTRL.TyEN to enable DACx and DACy trigger.
- Set DAC_SELCTRL.TxSEL[4:0] and DAC_SELCTRL.TySEL[4:0] to different values to select the same trigger source.
- Set DACxy_CTRL.WxEN[2:0] and DACxy_CTRL.WyEN[2:0] as "001" to enable noise generation.
- Set DACxy_SELCTRL.MAxSEL[3:0] and DACxy_SELCTRL.MAySEL[3:0] to different values to obtain the same LFSR register mask bits.
- Place the desired conversion data into the corresponding aligned data holding registers.

When a trigger event occurs, the counter value of LFSR register 1 is added to the value of the corresponding data holding register. This sum is then passed to the register DACx_DATO after a delay of 3 bus_clk clock cycles, and the counter value of LFSR register 1 is updated at this time. Similarly, the counter value of LFSR register 2 is added to the value of the corresponding data holding register when a trigger event occurs. This sum is then passed to the register DACy_DATO after a delay of 3 bus_clk clock cycles, and the counter value of LFSR register 2 is updated at this time.

19.5.12 Generating Synchronous Triggers for the Same Triangle Wave

Configuration process as follows:

- Configure DACxy_CTRL.TxEN and DACxy_CTRL.TyEN to enable DACx and DACy trigger.
- Set DAC_SELCTRL.TxSEL[4:0] and DAC_SELCTRL.TySEL[4:0] to the same value to select the same trigger source.
- Set DACxy_CTRL.WxEN[2:0] and DACxy_CTRL.WyEN[2:0] as "010" to enable triangle wave generation.
- Set DACxy_SELCTRL.MAxSEL[3:0] and DACxy_SELCTRL.MAySEL[3:0] to the same value to obtain the same triangle wave amplitude.
- Place the desired conversion data into the corresponding aligned data holding registers.

When a trigger event occurs, the triangle wave amplitude of DACx is added to the value of the corresponding data holding register. This sum is then passed to the register DACx_DATO after a delay of 3 bus_clk clock cycles, updating the counter value of DACx's triangle wave at that time. Similarly, the triangle wave amplitude of DACy is added to the value of the corresponding data holding register when a trigger event occurs. This sum is then passed to the register DACy_DATO after a delay of 3 bus_clk clock cycles, updating the counter value of DACy's triangle wave at that time.

19.5.13 Generating Synchronous Triggers for Different Triangle Waves

Configuration process as follows:

- Configure DACxy_CTRL.TxEN and DACxy_CTRL.TyEN to enable DACx and DACy trigger.

- Set DAC_SELCTRL.TxSEL[4:0] and DAC_SELCTRL.TySEL[4:0] to the same value to select the same trigger source.
- Set DACxy_CTRL.WxEN[2:0] and DACxy_CTRL.WyEN[2:0] as "010" to enable triangle wave generation.
- Set DACxy_SELCTRL.MAxSEL[3:0] and DACxy_SELCTRL.MAySEL[3:0] to different values to obtain different triangle wave amplitudes.
- Place the desired conversion data into the corresponding aligned data holding registers.

When a trigger event occurs, the triangle wave amplitude of DACx is added to the value of the corresponding data holding register. This sum is then passed to the register DACx_DATO after a delay of 3 bus_clk clock cycles, updating the counter value of DACx's triangle wave at that time. Similarly, the triangle wave amplitude of DACy is added to the value of the corresponding data holding register when a trigger event occurs. This sum is then passed to the register DACy_DATO after a delay of 3 bus_clk clock cycles, updating the counter value of DACy's triangle wave at that time.

19.5.14 Generating Synchronous Triggers for the Same Sawtooth Wave

Configuration process as follows:

- Configure DACxy_CTRL.TxEN and DACxy_CTRL.TyEN to enable DACx and DACy trigger.
- Set DAC_SELCTRL.TxSEL[4:0] and DAC_SELCTRL.TySEL[4:0] to the same value to select the same reset trigger source.
- Set DAC_SELCTRL.STINCSELx[4:0] and DAC_SELCTRL.STINCSELy[4:0] to the same value to select the same increment trigger source.
- Set DAC_STINC.STINCDATAx[11:0] and DAC_STINC.STINCDATAy[11:0] to the same value as the step value for the sawtooth wave.
- Set DAC_STRST.STRSTDATAx[11:0] and DAC_STRST.STRSTDATAy[11:0] to the same value as the reset value for the sawtooth wave.
- Set DACxy_CTRL.WxEN[2:0] and DACxy_CTRL.WyEN[2:0] as "100" or "110" to enable sawtooth wave generation.
- Enable the corresponding external triggers.

When a reset event is triggered by DACx, the sawtooth wave will pass the value of DAC_STRST.STRSTDATAx[11:0] register to the DACx_DATO register after a delay of 3 bus_clk clock cycles, updating the counter value of DACx's sawtooth wave at that time. When an increment event is triggered, the counter of the sawtooth wave will increase or decrease by DAC_STINC.STINCDATAx[11:0], and the result of the operation will be passed to the DACx_DATO register after a delay of 3 bus_clk clock cycles. When a reset event is triggered by DACy, the sawtooth wave will pass the value of DAC_STRST.STRSTDATAy[11:0] register to the DACy_DATO register after a delay of 3 bus_clk clock cycles, updating the counter value of DACy's sawtooth wave at that time. When an increment event is triggered, the counter of the sawtooth wave will increase or decrease by DAC_STINC.STINCDATAy[11:0], and the result of the operation will be passed to the DACy_DATO register after a delay of 3 bus_clk clock cycles.

19.5.15 Generating Synchronous Triggers for Different Sawtooth Waves

Configuration process as follows:

- Configure DACxy_CTRL.TxEN and DACxy_CTRL.TyEN to enable DACx and DACy trigger.
- Set DAC_SELCTRL.TxSEL[4:0] and DAC_SELCTRL.TySEL[4:0] to the same value to select the same reset trigger source.
- Set DAC_SELCTRL.STINCSELx[4:0] and DAC_SELCTRL.STINCSELy[4:0] to the same value to select the same increment trigger source.
- Set DAC_STINC.STINCDATAx[11:0] and DAC_STINC.STINCDATAy[11:0] to different values as the step value for the sawtooth wave.
- Set DAC_STRST.STRSTDATAx[11:0] and DAC_STRST.STRSTDATAy[11:0] to different values as the reset value for the sawtooth wave.
- Set DACxy_CTRL.WxEN[2:0] and DACxy_CTRL.WyEN[2:0] as "100" or "110" to enable sawtooth wave generation.
- Enable the corresponding external triggers.

When a reset event is triggered by DACx, the sawtooth wave will pass the value of DAC_STRST.STRSTDATAx[11:0] register to the DACx_DATO register after a delay of 3 bus_clk clock cycles, updating the counter value of DACx's sawtooth wave at that time. When an increment event is triggered, the counter of the sawtooth wave will increase or decrease by DAC_STINC.STINCDATAx[11:0], and the result of the operation will be passed to the DACx_DATO register after a delay of 3 bus_clk clock cycles. When a reset event is triggered by DACy, the sawtooth wave will pass the value of DAC_STRST.STRSTDATAy[11:0] register to the DACy_DATO register after a delay of 3 bus_clk clock cycles, updating the counter value of DACy's sawtooth wave at that time. When an increment event is triggered, the counter of the sawtooth wave will increase or decrease by DAC_STINC.STINCDATAy[11:0], and the result of the operation will be passed to the DACy_DATO register after a delay of 3 bus_clk clock cycles.

19.6 DAC Interrupt

If when the external trigger arrives for DAC, DMA has not yet responded to the previous external trigger request, then DAC will not issue a new request. Instead, it will set the DAC_STS.DMAUDRx(y) flag to report an error condition. This flag can be cleared by writing a 1 through software.

If the DACxy_CTRL.DMAUDRx(y)IEN bit is enabled, an interrupt will be generated at this time. This interrupt can wake the chip from SLEEP mode.

19.7 DAC Register

19.7.1 DAC Registers Overview

Table 19-9 DAC Registers Overview

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	DACxy_CTRL	Reserved	TROVCyIEN	EXOUTyEN	INOUTyEN	CALyEN	WyEN[2:0]			SINFORMATy	DMADOUBLEy	HDByEN	DMAUDRyIEN	TyEN	ByEN	DMAyEN	DACyEN	Reserved	TROVCxIEN	EXOUTxEN	INOUTxEN	CALxEN	WxEN[2:0]			SINFORMATx	DMADOUBLEx	HDBxEN	DMAUDRxIEN	TxEN	BxEN	DMAxEN	DACxEN
	Reset value		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x04	DAC_SOTTR	Reserved														TRByEN	TRBxEN	Reserved														TRyEN	TRxEN
	Reset value															0	0															0	0
0x08	DACx_DATO	Reserved																				DACxDO[11:0]											
	Reset value																					0	0	0	0	0	0	0	0	0	0	0	0
0x0C	DACy_DATO	Reserved																				DACyDO[11:0]											
	Reset value																					0	0	0	0	0	0	0	0	0	0	0	0
0x10	DACx_DR8	Reserved																DACxDB[7:0]								DACxD[7:0]							
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x14	DACx_DL12	DACxDB[11:0]												Reserved				DACxD[11:0]												Reserved			
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0					0	0	0	0	0	0	0	0	0	0	0	0				
0x18	DACx_DR12	Reserved				DACxDB[11:0]												Reserved				DACxD[11:0]											
	Reset value					0	0	0	0	0	0	0	0	0	0	0	0					0	0	0	0	0	0	0	0	0	0	0	0
0x1C	DACy_DR8	Reserved																DACyDB[7:0]								DACyD[7:0]							
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x20	DACy_DL12	DACyDB[11:0]												Reserved				DACyD[11:0]												Reserved			
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0					0	0	0	0	0	0	0	0	0	0	0	0				
0x24	DACy_DR12	Reserved				DACyDB[11:0]												Reserved				DACyD[11:0]											
	Reset value					0	0	0	0	0	0	0	0	0	0	0	0					0	0	0	0	0	0	0	0	0	0	0	0
0x28	DACxy_DR8D	Reserved																DACyD[7:0]								DACxD[7:0]							
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x2C	DACxy_DL12D	DACyD[11:0]												Reserved				DACxD[11:0]												Reserved			
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0					0	0	0	0	0	0	0	0	0	0	0	0				
0x30	DACxy_DR12D	Reserved				DACyD[11:0]												Reserved				DACxD[11:0]											
	Reset value					0	0	0	0	0	0	0	0	0	0	0	0					0	0	0	0	0	0	0	0	0	0	0	0
0x38	DACxy_SELCTRL	TySEL[4:0]					Reserved	MAySEL[3:0]				STINCSELy[4:0]					Reserved	TxSEL[4:0]					Reserved	MAxSEL[3:0]				STINCSELx[4:0]					Reserved
	Reset value	0	0	0	0	0		0	0	0	0	0	0	0	0	0		0	0	0	0	0		0	0	0	0	0	0	0	0	0	
0x3C	DACxy_STS	Reserved											VFLAGyCOMP	TROVCFLAGy	DORSTATy	CALFLAGy	DMAUDRy	Reserved											VFLAGxCOMP	TROVCFLAGx	DORSTATx	CALFLAGx	DMAUDRx
	Reset value												0	0	0	0	0												0	0	0	0	0
0x40	DACxy_GCTRL	Reserved								PCS[7:0]								Reserved													HFSEL[1:0]		Reserved
	Reset value									0	1	1	1	0	1	1	1														0	0	
0x44	DACxy_STINC	Reserved				STINCDATAy[11:0]												Reserved				STINCDATAx[11:0]											
	Reset value					0	0	0	0	0	0	0	0	0	0	0	0					0	0	0	0	0	0	0	0	0	0	0	0
0x48	DACxy_STRST	Reserved				STRSTDATAy[11:0]												Reserved				STRSTDATAx[11:0]											
	Reset value					0	0	0	0	0	0	0	0	0	0	0	0					0	0	0	0	0	0	0	0	0	0	0	0
0x5C	DACxy_CALC	Reserved											OTRIMy[4:0]					Reserved											OTRIMx[4:0]				
	Reset value												0	0	0	0	0												0	0	0	0	0

19.7.2 DACxy Control Register（DACxy_CTRL）(xy = 12、34、56、78)

Offset address: 0x00

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved	TROVCyIEN	EXOUTDACy	INOUTDACy	CALyEN	WyEN			SINFROMATy	DMADOUBLEy	HDByEN	DMAUDRyIEN	TyEN	ByEN	DMAyEN	DACyEN
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	TROVCxIEN	EXOUTDACx	INOUTDACx	CALxEN	WxEN			SINFROMATx	DMADOUBLEx	HDBxEN	DMAUDRxIEN	TxEN	BxEN	DMAxEN	DACxEN
	rw	rw	rw	rw		rw		rw	rw	rw	rw	rw	rw	rw	rw

Field offset	Register name	Description
31	Reserved	Reserved, must be kept at reset value.
30	TROVCyIEN	DACy trigger overclock interrrupt enable This bit is set and cleared by software. 0: DACy trigger interrrupt overclock interrupt disabled 1: DACy trigger interrrupt overclock interrupt enabled
29	EXOUTyEN	DACy external output channel enable. This bit set and cleared by software to enable/disable DACy external output channel. 0: DACy external output disabled 1: DACy external output enabled *Note: This bit is only configurable for DAC1, DAC2, DAC3, and DAC4. It is reserved as 0 for DAC5, DAC6, DAC7, and DAC8.*
28	INOUTyEN	DACy internal output channel enable. This bit set and cleared by software to enable/disable DACy internal output channel. 0: DACy internal output disabled 1: DACy internal output enabled *Note: This bit is only configurable for DAC1, DAC2, DAC3, and DAC4. It is reserved as 0 for DAC5, DAC6, DAC7, and DAC8.*
27	CALyEN	DACy calibration enable This bit is set and cleared by software to enable/disable DACy calibration,it can be written only if DACyEN bit is set to 0 into DAC_CTRL(the calibration mode can be entered/exit only when the DAC channel is disabled)Otherwise, the write operation is ignored. 0:DACy in nomal operating mode 1:DACy in calibration mode *Note: This bit is only configurable for DAC1, DAC2, DAC3, and DAC4. It is reserved as 0 for DAC5, DAC6, DAC7, and DAC8.*
26:24	WyEN[2:0]	DACy noise/triangle wave/sawtooth wave generation enable These bits are set/reset by software. 000: wave generation disabled 001: Noise wave generation enabled 010: Triangle wave generation enabled 100: increase sawtooth wave generation enable 110: decrease sawtooth wave generation enable *Note: only used if bit T1EN = 1 (DAC 2 trigger enabled)*
23	SINFORMATy	DACy signed format enable This bit is set and cleared by software. 0: DACy input data is in unsigned format 1: DACy input data is in signed format(2's complement).The MSB bit represents the sign
22	DMADOUBLEy	DACy DMA double data mode This bit is set and cleared by software. 0: DACy DMA normal mode selected

Field offset	Register name	Description
		1: DACy DMA double data mode selected
21	HDByEN	DACy drive buffer enable This bit is set and cleared by software. 0: DACy output normal drive 1: DACy output high drive *Note: This bit is only configurable for DAC1, DAC2, DAC3, and DAC4. It is reserved as 0 for DAC5, DAC6, DAC7, and DAC8.*
20	DMAUDRyIEN	DACy DMA Underrun Interrupt enable This bit is set and cleared by software. 0: DACy DMA Underrun Interrupt disabled 1: DACy DMA Underrun Interrupt enabled
19	TyEN	DACy trigger enable This bit set and cleared by software to enable/disable DACy trigger 0：DACy trigger disable 1：DACy trigger enable
18	ByEN	DACy output buffer enable This bit set and cleared by software to enable/disable DACy output buffer. 0: DACy output buffer disabled 1: DACy output buffer enabled *Note: This bit is only configurable for DAC1, DAC2, DAC3, and DAC4. It is reserved as 0 for DAC5, DAC6, DAC7, and DAC8.*
17	DMAyEN	DACy DMA enable This bit is set and cleared by software. 0: DACy DMA mode disabled 1: DACy DMA mode enabled
16	DACyEN	DACy enable This bit set and cleared by software to enable/disable DACy. 0: DACy disabled 1: DACy enabled
15	Reserved	Reserved, must be kept at reset value.
14	TROVCxIEN	DACx trigger overclock interrrupt enable This bit is set and cleared by software. 0: DACx trigger interrrupt overclock interrupt disabled 1: DACx trigger interrrupt overclock interrupt enabled
13	EXOUTxEN	DACx external output channel enable. This bit set and cleared by software to enable/disable DACx external output channel. 0: DACx external output disabled 1: DACx external output enabled *Note: This bit is only configurable for DAC1, DAC2, DAC3, and DAC4. It is reserved as 0 for DAC5, DAC6, DAC7, and DAC8.*
12	INOUTxEN	DACx internal output channel enable. This bit set and cleared by software to enable/disable DACx internal output channel.

Field offset	Register name	Description
		0: DACx internal output disabled 1: DACx internal output enabled *Note: This bit is only configurable for DAC1, DAC2, DAC3, and DAC4. It is reserved as 0 for DAC5, DAC6, DAC7, and DAC8.*
11	CALxEN	DACx calibration enable This bit is set and cleared by software to enable/disable DAC 1 calibration. 0:DAC1 in nomal operating mode 1:DAC1 in calibration mode *Note: This bit is only configurable for DAC1, DAC2, DAC3, and DAC4. It is reserved as 0 for DAC5, DAC6, DAC7, and DAC8.*
10:8	WEN[2:0]	DACx noise/triangle wave/sawtooth wave generation enable These bits are set/reset by software. 000: wave generation disabled 001: Noise wave generation enabled 010: Triangle wave generation enabled 100: increase sawtooth wave generation enable 110: decrease sawtooth wave generation enable *Note: only used if bit T1EN = 1 (DACx trigger enabled)*
7	SINFORMATx	DACx signed format enable This bit is set and cleared by software. 0: DACx input data is in unsigned format 1: DACx input data is in signed format(2's complement).The MSB bit represents the sign
6	DMADOUBLEx	DACx DMA double data mode This bit is set and cleared by software. 0: DACx DMA normal mode selected 1: DACx DMA double data mode selected
5	HDBxEN	DACx drive buffer enable This bit is set and cleared by software. 0: normal drive 1: high drive *Note: This bit is only configurable for DAC1, DAC2, DAC3, and DAC4. It is reserved as 0 for DAC5, DAC6, DAC7, and DAC8.*
4	DMAUDRxIEN	DACx DMA Underrun Interrupt enable This bit is set and cleared by software. 0: DACx DMA Underrun Interrupt disabled 1: DACx DMA Underrun Interrupt enabled
3	TxEN	DACx trigger enable This bit set and cleared by software to enable/disable DACx trigger 0: DACx trigger disabled 1: DACx trigger enabled

Field offset	Register name	Description
2	BxEN	DACx output buffer enable This bit set and cleared by software to enable/disable DACx output buffer. 0: DACx output buffer disabled 1: DACx output buffer enabled *Note: This bit is only configurable for DAC1, DAC2, DAC3, and DAC4. It is reserved as 0 for DAC5, DAC6, DAC7, and DAC8.*
1	DMAxEN	DACx DMA enable This bit is set and cleared by software. 0: DACx DMA mode disabled 1: DACx DMA mode enabled
0	DACxEN	DACx enable This bit set and cleared by software to enable/disable DACx. 0: DACx disabled 1: DACx enabled

19.7.3 DACxy Software Trigger Register（DACxy_SOTTR）

Offset address:： 0x04

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved														TRByEN	TRBxEN
														wo	wo

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved														TRyEN	TRxEN
														wo	wo

Field offset	Register name	Description
31:18	Reserved	Reserved, must be kept at reset value.
17	TRByEN	Enable sawtooth wave step trigger for DACy. This bit is set to 1 by software to enable/disable software triggering. This signal is only used for the sawtooth wave step trigger signal. 0: Disable software triggering for DACy sawtooth wave step; 1: Enable software triggering for DACy sawtooth wave step.
16	TRBxEN	Enable sawtooth wave step trigger for DACx. This bit is set to 1 by software to enable/disable software triggering. This signal is only used for the sawtooth wave step trigger signal. 0: Disable software triggering for DACx sawtooth wave step; 1: Enable software triggering for DACx sawtooth wave step.
15:2	Reserved	Reserved, must be kept at reset value.

Field offset	Register name	Description
1	TRyEN	DACy software trigger enable. This bit is set to 1 by the software to enable/disable software triggering. This signal can be used as the trigger signal for triangle wave/noise wave, or as the reset trigger signal for sawtooth wave. 0: Disable DACy software trigger; 1: Enable DACy software trigger. *Note: After the aligned data is transferred to the DACy_DATO register, this bit will be cleared to 0 by the hardware one bus_clk clock later.*
0	TRxEN	DACx software trigger enable. This bit is set to 1 by the software to enable/disable software triggering. This signal can be used as the trigger signal for triangle wave/noise wave, or as the reset trigger signal for sawtooth wave. 0: Disable DACx software trigger; 1: Enable DACx software trigger. *Note: After the aligned data is transferred to the DACx_DATO register, this bit will be cleared to 0 by the hardware one bus_clk clock later.*

19.7.4 DACx Data Output Register（DACx_DATO）

Offset address: 0x08

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				DACxDO[11:0]											
				r											

Field offset	Register name	Description
31:12	Reserved	Reserved, must be kept at reset value.
11:0	DACxDO[11:0]	DACx data output These bits are read only, they contain data output for DACx

19.7.5 DACy Data Output Register（DACy_DATO）

Offset address: 0x0C

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				DACyDO[11:0]											
				r											

Field offset	Register name	Description
31:12	Reserved	Reserved, must be kept at reset value.
11:0	DACyDO[11:0]	DACy data output These bits are read only, they contain data output for DACy

19.7.6 DACx's 8-bit right-aligned data holding register（DACx_DR8）

Offset address: 0x10

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DACxDB[7:0]								DACxD[7:0]							
rw								rw							

Field offset	Register name	Description
31:16	Reserved	Reserved, must be kept at reset value.
15:8	DACxDB[7:0]	DACx 8-bit right-aligned data B These bits are written by software.They specify 8-bit data for DACx when the DAC operates in double data mode
7:0	DACxD[7:0]	DACx 8-bit right-aligned data These bits are written by software.They specify 8-bit data for DACx.

19.7.7 DACx's 12-bit left-aligned data holding register（DACx_DL12）

Offset address: 0x14

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DACxDB[11:0]												Reserved			

rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DACxD[11:0]												Reserved			

rw

Field offset	Register name	Description
31:20	DACxDB[11:0]	DACx 12-bit left-aligned data B These bits are written by software.They specify 12-bit data for DACx when the DAC operates in double data mode
19:16	Reserved	Reserved, must be kept at reset value.
15:4	DACxD[11:0]	DACx 12-bit left-aligned data These bits are written by software.They specify 12-bit data for DACx.
3:0	Reserved	Reserved, must be kept at reset value.

19.7.8 DACx's 12-bit right-aligned data holding register（DACx_DR12）

Offset address: 0x18

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				DACxDB[11:0]											

rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				DACxD[11:0]											

rw

Field offset	Register name	Description
31:28	Reserved	Reserved, must be kept at reset value.
27: 16	DACxDB[11:0]	DACx 12-bit right-aligned data B These bits are written by software.They specify 12-bit data for DACx when the DAC operates in double data mode
15:12	Reserved	Reserved, must be kept at reset value.
11:0	DACxD[11:0]	DACx 12-bit right-aligned data These bits are written by software.They specify 12-bit data for DACx.

19.7.9 DACy's 8-bit right-aligned data holding register（DACy_DR8）

Offset address: 0x1C

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DACyDB[7:0]								DACyD[7:0]							
rw								rw							

Field offset	Register name	Description
31:16	Reserved	Reserved, must be kept at reset value.
15:8	DACyDB[7:0]	DACy 8-bit right-aligned data B These bits are written by software.They specify 8-bit data for DACy when the DAC operates in double data mode
7:0	DACyD[7:0]	DACy 8-bit right-aligned data These bits are written by software.They specify 8-bit data for DACy.

19.7.10 DACy's 12-bit left-aligned data holding register（DACy_DL12）

Offset address: 0x20

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DACyDB[11:0]												Reserved			
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DACyD[11:0]												Reserved			
rw															

Field offset	Register name	Description
31:20	DACyDB[11:0]	DACy 12-bit left-aligned data B These bits are written by software.They specify 12-bit data for DACy when the DAC operates in double data mode
19:16	Reserved	Reserved, must be kept at reset value.
15:4	DACyD[11:0]	DACy 12-bit left-aligned data These bits are written by software.They specify 12-bit data for DACy.
3:0	Reserved	Reserved, must be kept at reset value.

19.7.11 DACy's 12-bit right-aligned data holding register（DACy_DR12）

Offset address: 0x24

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				DACyDB[11:0]											
				rw											

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				DACyD[11:0]											
				rw											

Field offset	**Register name**	**Description**
31:28	Reserved	Reserved, must be kept at reset value.
27:16	DACyDB[11:0]	DACy 12-bit right-aligned data B These bits are written by software.They specify 12-bit data for DACy when the DAC operates in double data mode
15:12	Reserved	Reserved, must be kept at reset value.
11:0	DACyD[11:0]	DACy 12-bit right-aligned data These bits are written by software.They specify 12-bit data for DACy.

19.7.12 Dual DACxy's 8-bit right-aligned data holding register（DACxy_DR8D）

Offset address: 0x28

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DACyD [7:0]								DACxD[7:0]							
rw								rw							

Field offset	**Register name**	**Description**
31:16	Reserved	Reserved, must be kept at reset value.
15:8	DACyD[7:0]	DACy 8-bit right-aligned data B

Field offset	Register name	Description
		These bits are written by software.They specify 8-bit data for DACy when the DAC operates in double data mode
7:0	DACxD[7:0]	DACx 8-bit right-aligned data These bits are written by software.They specify 8-bit data for DACx.

19.7.13 Dual DACxy's 12-bit left-aligned data holding register（DACxy_DL12D）

Offset address: 0x2C

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DACyD[11:0]												Reserved			
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DACxD[11:0]												Reserved			
rw															

Field offset	Register name	Description
31:20	DACyD[11:0]	DACy 12-bit left-aligned data These bits are written by software.They specify 12-bit data for DACy.
19:16	Reserved	Reserved, must be kept at reset value.
15:4	DACxD[11:0]	DACx 12-bit left-aligned data These bits are written by software.They specify 12-bit data for DACx.
3:0	Reserved	Reserved, must be kept at reset value.

19.7.14 Dual DACxy's 12-bit right-aligned data holding register（DACxy_DR12D）

Offset address: 0x30

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				DACyD[11:0]											
				rw											

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				DACxD[11:0]											
				rw											

Field offset	Register name	Description
31:28	Reserved	Reserved, must be kept at reset value.
27:16	DACyD[11:0]	DACy 12-bit right-aligned data These bits are written by software.They specify 12-bit data for DACy.
15:12	Reserved	Reserved, must be kept at reset value.
11:0	DACxD[11:0]	DACx 12-bit right-aligned data These bits are written by software.They specify 12-bit data for DACx.

19.7.15 DACxy selection control register（DACxy_SELCTRL）

Offset address: 0x38

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TySEL[4:0]					Reserved	MAySEL[3:0]				STINCSELy[4:0]					Reserved
rw						rw				rw					

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TxSEL[4:0]					Reserved	MAxSEL[3:0]				STINCSELx[4:0]					Reserved
rw						rw				rw					

Field offset	Register name	Description
31:27	TySEL[4:0]	DACy trigger selection for LFSR/triangle/sawtooth reset These bits select the external event used to trigger DACy 00000 : software_trgo 00001 : atim1_trgo 00010 : atim2_trgo 00011 : atim3_trgo 00100 : Reserve 00101 : gtim1_trgo 00110 : gtim2_trgo 00111 : gtim3_trgo 01000 : gtim4_trgo 01001 : gtim5_trgo 01010 : gtim6_trgo 01011 : gtim7_trgo 01100 : gtim8_trgo 01101 : gtim9_trgo 01110 : gtim10_trgo 01111 : exti5 10000 : exti7 10001 : exti9

Field offset	Register name	Description
		10010 : shrtim1_dac_reset_trg1 10011 : shrtim1_dac_reset_trg2 10100 : shrtim1_dac_reset_trg3 10101 : shrtim1_dac_reset_trg4 10110 : shrtim1_dac_reset_trg5 10111 : shrtim1_dac_reset_trg6 11000 : shrtim1_dac_trg1 *Note: only used if bit TyEN = 1 (DACy trigger enabled)*
26	Reserved	Reserved, must be kept at reset value.
25:22	MAySEL[3:0]	DACy mask/amplitude selector These bits are written by software to select mask in wave generation mode or amplitude in triangle generation mode. 0000: Unmask bit0 of LFSR/ Triangle Amplitude equal to 1 0001: Unmask bits[1:0] of LFSR/ Triangle Amplitude equal to 3 0010: Unmask bits[2:0] of LFSR/ Triangle Amplitude equal to 7 0011: Unmask bits[3:0] of LFSR/ Triangle Amplitude equal to 15 0100: Unmask bits[4:0] of LFSR/ Triangle Amplitude equal to 31 0101: Unmask bits[5:0] of LFSR/ Triangle Amplitude equal to 63 0110: Unmask bits[6:0] of LFSR/ Triangle Amplitude equal to 127 0111: Unmask bits[7:0] of LFSR/ Triangle Amplitude equal to 255 1000: Unmask bits[8:0] of LFSR/ Triangle Amplitude equal to 511 1001: Unmask bits[9:0] of LFSR/ Triangle Amplitude equal to 1023 1010: Unmask bits[10:0] of LFSR/ Triangle Amplitude equal to 2047 ≥ 1011: Unmask bits[11:0] of LFSR/ Triangle Amplitude equal to 4095
21:17	STINCSELy[4:0]	DACy sawtooth increment trigger selection. 00000 : software_trgo 00001 : atim1_trgo 00010 : atim2_trgo 00011 : atim3_trgo 00100 : reserved 00101 : gtim1_trgo 00110 : gtim2_trgo 00111 : gtim3_trgo 01000 : gtim4_trgo 01001 : gtim5_trgo 01010 : gtim6_trgo 01011 : gtim7_trgo 01100 : gtim8_trgo 01101 : gtim9_trgo 01110 : gtim10_trgo 01111: exti6

Field offset	Register name	Description
		10000: exti8 10001: exti10 10010: shrtim1_step_trg1 10011: shrtim1_step_trg2 10100: shrtim1_step_trg3 10101: shrtim1_step_trg4 10110: shrtim1_step_trg5 10111: shrtim1_step_trg6 These bits are available only on dual-channel DACs.
16	Reserved	Reserved, must be kept at reset value.
15:11	TxSEL[4:0]	DACx trigger selection. These bits are used for selecting the external trigger for DACx. This bit can be used to select the trigger source for noise wave and triangle wave, and can also be used to select the reset trigger source for sawtooth wave. 00000 : software_trgo 00001 : atim1_trgo 00010 : atim2_trgo 00011 : atim3_trgo 00100 : Reserve 00101 : gtim1_trgo 00110 : gtim2_trgo 00111 : gtim3_trgo 01000 : gtim4_trgo 01001 : gtim5_trgo 01010 : gtim6_trgo 01011 : gtim7_trgo 01100 : gtim8_trgo 01101 : gtim9_trgo 01110 : gtim10_trgo 01111 : exti5 10000 : exti7 10001 : exti9 10010 : shrtim1_dac_reset_trg1 10011 : shrtim1_dac_reset_trg2 10100 : shrtim1_dac_reset_trg3 10101 : shrtim1_dac_reset_trg4 10110 : shrtim1_dac_reset_trg5 10111 : shrtim1_dac_reset_trg6 11000 : shrtim1_dac_trg1 *Note: only used if bit TxEN = 1 (DAC 1 trigger enabled)*
10	Reserved	Reserved, must be kept at reset value.
9:6	MAxSEL[3:0]	DACx mask/amplitude selector

Field offset	Register name	Description
		These bits are written by software to select mask in wave generation mode or amplitude in triangle generation mode. 0000: Unmask bit0 of LFSR/ Triangle Amplitude equal to 1 0001: Unmask bits[1:0] of LFSR/ Triangle Amplitude equal to 3 0010: Unmask bits[2:0] of LFSR/ Triangle Amplitude equal to 7 0011: Unmask bits[3:0] of LFSR/ Triangle Amplitude equal to 15 0100: Unmask bits[4:0] of LFSR/ Triangle Amplitude equal to 31 0101: Unmask bits[5:0] of LFSR/ Triangle Amplitude equal to 63 0110: Unmask bits[6:0] of LFSR/ Triangle Amplitude equal to 127 0111: Unmask bits[7:0] of LFSR/ Triangle Amplitude equal to 255 1000: Unmask bits[8:0] of LFSR/ Triangle Amplitude equal to 511 1001: Unmask bits[9:0] of LFSR/ Triangle Amplitude equal to 1023 1010: Unmask bits[10:0] of LFSR/ Triangle Amplitude equal to 2047 ≥ 1011: Unmask bits[11:0] of LFSR/ Triangle Amplitude equal to 4095
5:1	STINCSELx[4:0]	DACx sawtooth increment trigger selection. 00000 : software_trgo 00001 : atim1_trgo 00010 : atim2_trgo 00011 : atim3_trgo 00100 : reserved 00101 : gtim1_trgo 00110 : gtim2_trgo 00111 : gtim3_trgo 01000 : gtim4_trgo 01001 : gtim5_trgo 01010 : gtim6_trgo 01011 : gtim7_trgo 01100 : gtim8_trgo 01101 : gtim9_trgo 01110 : gtim10_trgo 01111: exti6 10000: exti8 10001: exti10 10010: shrtim1_step_trg1 10011: shrtim1_step_trg2 10100: shrtim1_step_trg3 10101: shrtim1_step_trg4 10110: shrtim1_step_trg5 10111: shrtim1_step_trg6 These bits are available only on dual-channel DACs.
0	Reserved	Reserved, must be kept at reset value.

19.7.16 DACxy status register（DACxy_STS）

Offset address: 0x3C

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved											VFLAGx COMP	TROVC FLAGy	DOR STATy	CAL FLAGy	DMA UDRy
											r	rc_w1	r	r	rc_w1

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved											VFLAGx COMP	TROVC FLAGx	DOR STATx	CAL FLAGx	DMA UDRx
											r	rc_w1	r	r	rc_w1

Field offset	Register name	Description
31:21	Reserved	Reserved, must be kept at reset value.
20	VFLAGyCOMP	DACy signal stability indicator for the comparator. 0: The signal from DACy to the comparator is unprocessed and may be unstable; 1: The signal from DACy to the comparator is processed and stable. *Note: This bit is only configurable for DAC5, DAC6, DAC7, and DAC8. It is reserved as 0 for DAC1, DAC2, DAC3, and DAC4.*
19	TROVCFLAGy	DACy trigger source overrun interrupt flag. This bit is set by hardware and cleared by software (cleared by writing 1). 0: No trigger source overrun situation for DACy; 1: Trigger source overrun situation for DACy: The frequency at which the selected trigger source drives DACy conversion is faster than the speed of data updates.
18	DORSTATy	DACy data output register status bit. This bit is set and cleared by hardware and is available when DACy is operating in dual data mode. 0: DACyD is being used for DACy output data. 1: DACyDB is being used for DACy output data.
17	CALFLAGy	DACy calibration offset status. This bit is set and cleared by hardware. 0: Calibration adjustment value is less than the offset calibration value. 1: Calibration adjustment value is greater than or equal to the offset calibration value.
16	DMAUDRy	DACy DMA underrun interrupt flag. This bit is set by hardware and cleared by software (cleared by writing 1). 0: No DMA underrun error occurred for DACy; 1: DMA underrun error occurred for DACy (the frequency at which the selected trigger source drives DACy conversion is faster than the rate provided by DMA).
15:5	Reserved	Reserved, must be kept at reset value.
4	VFLAGxCOMP	VFLAGxCOMP

Field offset	Register name	Description
		DACx signal stability indicator. 0: The signal from DACx to the comparator is unprocessed and may be unstable; 1: The signal from DACx to the comparator is processed and stable. *Note: This bit is only configurable for DAC5, DAC6, DAC7, and DAC8. It is reserved as 0 for DAC1, DAC2, DAC3, and DAC4.*
3	TROVCFLAGx	DACx trigger source overrun flag. This bit is set by hardware and cleared by software (cleared by writing 1). 0: No trigger source overrun situation for DACx; 1: Trigger source overrun situation for DACx: The frequency at which the selected trigger source drives DACx conversion is faster than the speed of data updates.
2	DORSTATx	DACx data output register status bit. This bit is set and cleared by hardware and is available when DACx is operating in dual data mode. 0: DACxD is being used for DACx output data. 1: DACxDB is being used for DACx output data.
1	CALFLAGx	DACx calibration offset status. This bit is set and cleared by hardware. 0: The calibration adjustment value is less than the offset calibration value. 1: The calibration adjustment value is greater than or equal to the offset calibration value.
0	DMAUDRx	DACx DMA underrun flag. This bit is set by hardware and cleared by software (cleared by writing 1). 0: No DMA underrun error occurred for DACx; 1: DMA underrun error occurred for DACx (the frequency at which the selected trigger source drives DACx conversion is faster than the rate provided by DMA).

19.7.17 DACxy general control register（DACxy_GCTRL）

Offset address: 0x40

Reset value: 0x0077 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								PCS[7:0]							
r								rw							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													HFSEL[1:0]		Reserved
r													rw		

Field offset	Register name	Description
31:24	Reserved	Reserved, must be kept at reset value.
23:16	PCS[7:0]	DACx clock divider coefficient. When operating DACx as DAC1~DAC4: the operating frequency range of DACx is 4MHz to APB1_CLKmax. Configure the corresponding divider coefficient to achieve a maximum of 1M for DACxy_PCS[7:0] (x = 1, 2, 3, 4). For example, if the operating frequency of DAC1~4 is 120MHz, the corresponding divider coefficient is 01110111. When operating DACx as DAC5~DAC8: PCS[7:5] reserved as 0; the operating frequency range of DACx is 60MHz to AHB_CLKmax. Configure the corresponding divider coefficient to achieve a maximum of 15M for DACxy_PCS[4:0] (x = 5, 6, 7, 8). For example, if the operating frequency of DAC5~8 is 240MHz, the corresponding divider coefficient is 01111. 0: No division 1：Division by 2 2：Division by 3 …: Division by (PCS[7:0] + 1) *Note: If it is necessary to configure the DAC operating frequency to exceed 1MHz (DAC1, DAC2, DAC3, DAC4) or 15MHz (DAC5, DAC6, DAC7, DAC8), it is important to consider the maximum operating frequency of the DAC.*
15:3	Reserved	Reserved, must be kept at reset value.
2:1	HFSEL[1:0]	DACx and DACy response speed selection 00: DAC_DATO update rate is (PCS[7:0]/2) DAC_CLK cycles; this selection is the default value. 01: DAC_DATO update rate is 3 bus_clk cycles; (Fbus_clk < 80MHz) 10: DAC_DATO update rate is 5 bus_clk cycles; (80MHz < Fbus_clk < 160MHz) 11: DAC_DATO update rate is 7 bus_clk cycles; (Fbus_clk > 160MHz). *Note: This bit is only configurable for DAC1, DAC2, DAC3, and DAC4. It is reserved as 0 for DAC5, DAC6, DAC7, and DAC8.*
0	Reserved	Reserved, must be kept at reset value.

19.7.18 DACxy sawtooth wave step register（DACxy_STINC）

Offset address: 0x44

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				STINCDATAy[11:0]											
				rw											

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				STINCDATAx[11:0]											
				rw											

Field offset	Register name	Description
31:28	Reserved	Reserved, must be kept at reset value.
27:16	STINCDATAy[11:0]	DACy sawtooth wave step value. Range from 0 to 0xfff.
15:12	Reserved	Reserved, must be kept at reset value.
11:0	STINCDATAx[11:0]	DACx sawtooth wave step value. Range from 0 to 0xfff.

19.7.19 DACxy sawtooth wave reset register（DACxy_ STRST）

Offset address: 0x48

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				STRSTDATAy[11:0]											
				rw											

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				STRSTDATAx[11:0]											
				rw											

Field offset	Register name	Description
31:28	Reserved	Reserved, must be kept at reset value.
27:16	STRSTDATAy[11:0]	DACy sawtooth wave reset value. Range from 0 to 0xfff.
15:12	Reserved	Reserved, must be kept at reset value.
11:0	STRSTDATAx[11:0]	DACx sawtooth wave reset value. Range from 0 to 0xfff.

19.7.20 DACxy calibration control register（DACxy_ CALC）

Offset address: 0x5C

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved											OTRIMy[4:0]				
											rw				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved											OTRIMx[4:0]				
											rw				

Field offset	Register name	Description
31:21	Reserved	Reserved, must be kept at reset value.
20:16	OTRIMy[4:0]	DACy offset calibration value. *Note：* *1. Can only be used when ByEN=1.* *2. This bit is only configurable for DAC1, DAC2, DAC3, and DAC4. It is reserved as 0 for DAC5, DAC6, DAC7, and DAC8.*
15:5	Reserved	Reserved, must be kept at reset value.
4:0	OTRIMx[4:0]	DACx offset calibration value. *Note：* *1. Can only be used when BxEN=1.* *2. This bit is only configurable for DAC1, DAC2, DAC3, and DAC4. It is reserved as 0 for DAC5, DAC6, DAC7, and DAC8.*

20 Programmable Gain Amplifiers（PGA）

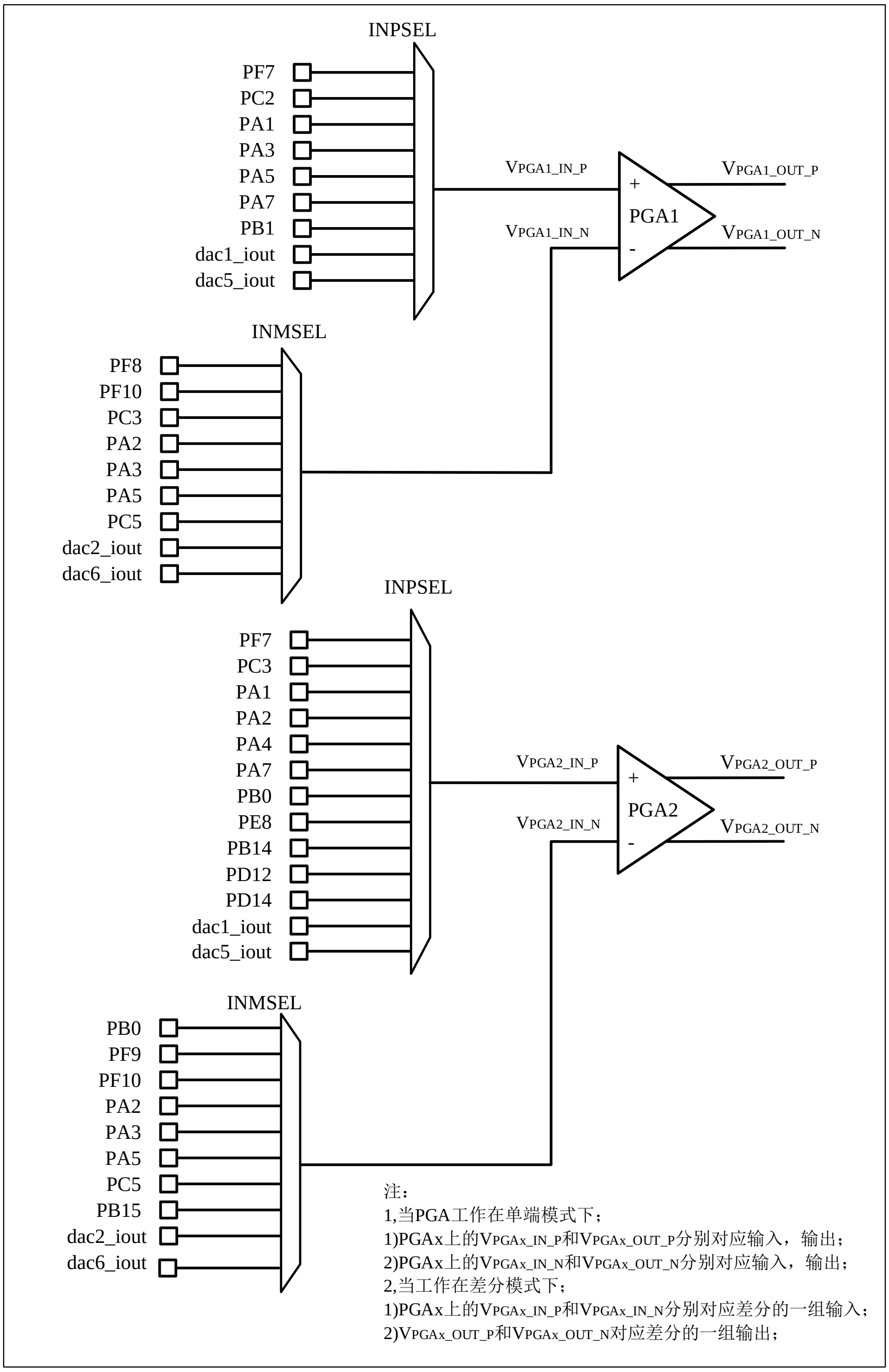
PGA is a programmable gain amplifier used to amplify input voltage. The chip has a total of 4 differential PGAs, each of which can be split into 2 single-ended PGAs and used independently. The output of the PGA can be internally connected to the input channels of ADC or COMP.

20.1 Main Features

- Supports rail-to-rail input
- Supports 12-bit DAC as input for PGA
- Programmable single-ended mode gain settings of 1X, 2X, 4X, 8X, 12X, 16X, 24X, 32X
- Programmable gain settings of 2X, 4X, 8X, 16X, 24X, 32X, 48X, 64X
- Supports automatic switching of input PIN for PGA1 with ATIM1_CC6 and for PGA2 input PIN, as well as automatic switching of input PIN for PGA3 with ATIM2_CC6 and for PGA4 input PIN
- Supports independent write protection

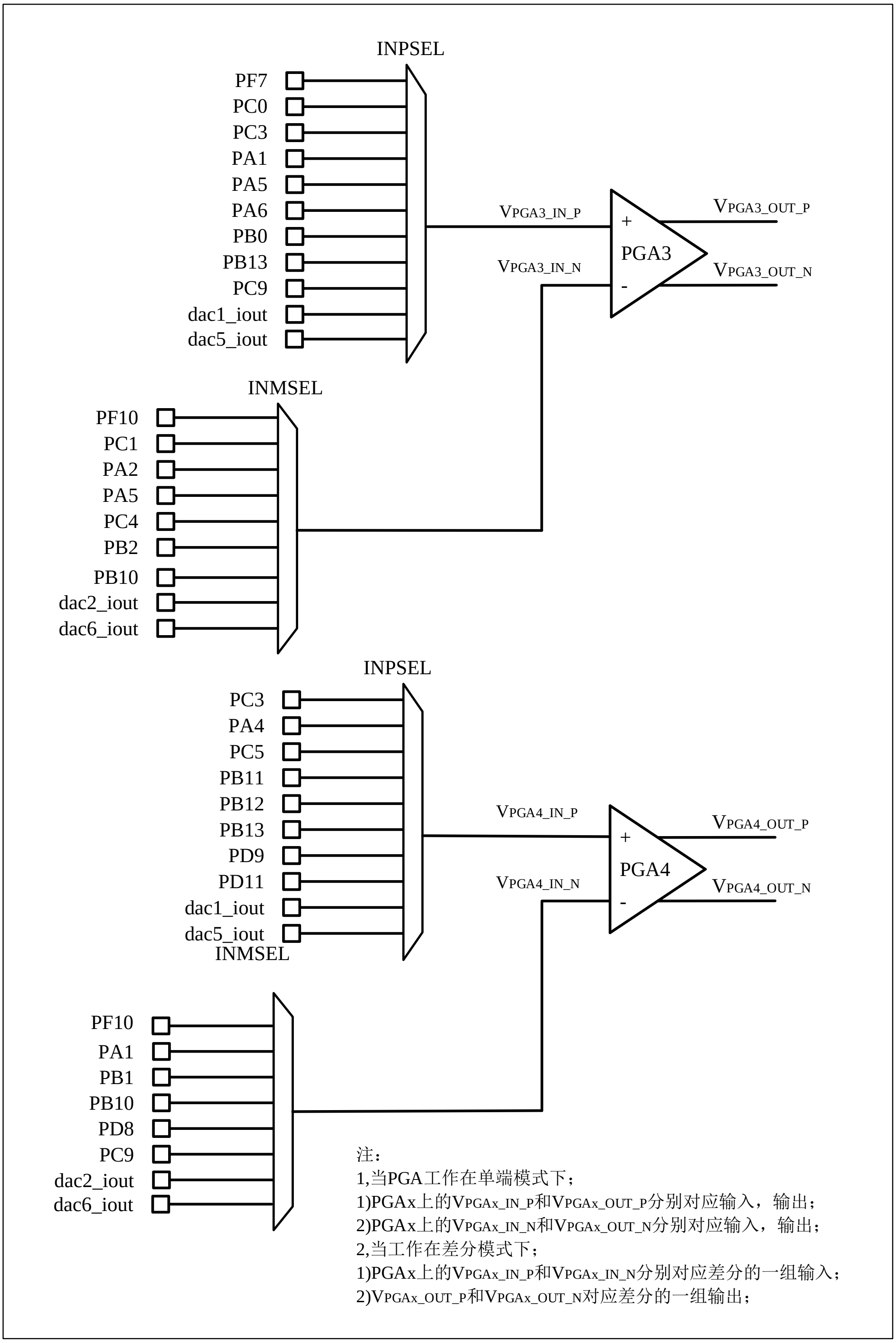
20.2 PGA Function Description

Figure 20-1 Block Diagram of Connection between PGA1 and PGA2



注：
1,当PGA工作在单端模式下；
1)PGAx上的VPGAx_IN_P和VPGAx_OUT_P分别对应输入，输出；
2)PGAx上的VPGAx_IN_N和VPGAx_OUT_N分别对应输入，输出；
2,当工作在差分模式下；
1)PGAx上的VPGAx_IN_P和VPGAx_IN_N分别对应差分的一组输入；
2)VPGAx_OUT_P和VPGAx_OUT_N对应差分的一组输出；

Figure 20-2 Block Diagram of Connection between PGA3 and PGA4



注：
1,当PGA工作在单端模式下；
1)PGAx上的VPGAx_IN_P和VPGAx_OUT_P分别对应输入，输出；
2)PGAx上的VPGAx_IN_N和VPGAx_OUT_N分别对应输入，输出；
2,当工作在差分模式下；
1)PGAx上的VPGAx_IN_P和VPGAx_IN_N分别对应差分的一组输入；
2)VPGAx_OUT_P和VPGAx_OUT_N对应差分的一组输出；

Note: Do not switch input channel pins for B-version chips

20.2.1 PGA Gain multiplier

Single-ended mode supports gain multiplier configurations of 1/2/4/8/12/16/24/32;

Differential mode supports gain multiplier configurations of 2/4/8/16/24/32/48/64.

20.2.2 PGA Single-ended mode operation process

Taking working in single-ended mode with PGAx channel 1 as an example:

1. Enable bias current source PGA_BIASEN.IBIASEN = 1;
2. Configure single-ended operating mode PGAx_CTRL.DIFFEN = 0;
3. Configure the positive input terminal by selecting the appropriate input channel through PGAx_CTRL.VPSEL;
4. Select the gain coefficient A by configuring PGAx_CTRL.CH1GAIN[2:0];
5. Enable PGA channel 1 by setting PGAx_CTRL.CH1EN = 1;
6. When the VINP voltage is Vinp, the PGA output voltage Vop is given by the following formula:

$$\mathrm{Vop} = A \times \mathrm{Vinp} \qquad (1)$$

Note: Vop should be ensured to be less than (VDDA – 0.3)V

20.2.3 PGA Differential mode operation process

Taking working in differential mode with PGAx as an example:

1. Enable bias current source PGA_BIASEN.IBIASEN = 1;
2. Configure differential operating mode PGAx_CTRL.DIFFEN = 1;
3. Configure positive and negative input terminals by selecting appropriate input channels through PGAx_CTRL.VPSEL, PGAx_CTRL.VMSEL;
4. 4. Select gain coefficient A by configuring PGAx_CTRL.CH1GAIN[2:0] (writing to PGAx_CTRL.CH2GAIN is invalid);
5. 5. Enable PGA channel 1 (PGAx_CTRL.CH1EN = 1) and channel 2 (PGAx_CTRL.CH2EN = 1);
6. When the VINP voltage is Vinp, VINN voltage is Vinn, Vop voltage is the output voltage at the P terminal, Von voltage is the output voltage at the N terminal, then the PGA output voltage satisfies the following three formulas:

$$\mathrm{Vop} = \mathrm{Vinn} + \left(\frac{1}{2} \times A \times (\mathrm{Vinp} - \mathrm{Vinn})\right) \qquad (2)$$

$$\mathrm{Von} = \mathrm{Vinn} - \left(\frac{1}{2} \times A \times (\mathrm{Vinp} - \mathrm{Vinn})\right) \qquad (3)$$

$$(\mathrm{Vop} - \mathrm{Von}) = A \times (\mathrm{Vinp} - \mathrm{Vinn}) \qquad (4)$$

Notes:

1. *In differential mode, the N terminal of the PGA needs to provide VDDA/2 bias voltage;*

2. *Vop, Von should be ensured to be between 0.3 and VDDA-0.3, otherwise distortion may occur.*

20.2.4 PGA User Calibration Process

Due to the characteristics of the amplifier, there are differences in offset voltages in different parts of the PGA. In order to compensate for errors introduced by various components, the PGA supports user calibration to calibrate the PGA. Here are the specific calibration steps:

1. Enable bias current source PGA_BIASEN.IBIASEN = 1;

2. Enable calibration PGAx_CTRL.TRIMEN = 1;

3. Configure the positive input terminal by selecting the appropriate input channel through PGAx_CTRL.VPSEL;

4. Enable PGA channel 1 by setting PGAx_CTRL.CH1EN = 1;

5. Input 0.3V to the VINP pin, sequentially configure PGAx_TRIM.TRIMCH1P[5:0] from 0 to 0x3F. After each modification, wait for about 100us, read PGAx_TRIM.TRIMCH1FLAG, and record the value of PGAx_TRIM.TRIMCH1P[5:0] at the moment when this flag flips (from 0->1 or from 1->0) (assume this value is Y1);

6. Input (VDD-0.3)V to the VINP pin, sequentially configure PGAx_TRIM.TRIMCH1N[5:0] from 0 to 0x3F. After each modification, wait for about 100us, read PGAx_TRIM.TRIMCH1FLAG, and record the value of PGAx_TRIM.TRIMCH1N[5:0] at the moment when this flag flips (from 0->1 or from 1->0) (assume this value is Y2);

7. Disable PGA positive channel 1 enable, i.e., PGAx_CTRL.CH1EN = 0;

8. Configure the negative input terminal by selecting the appropriate input channel through PGAx_CTRL.VNSEL and enable the negative channel 2, PGAx_CTRL.CH2EN = 1;

9. Input 0.3V to the VINN pin, sequentially configure PGAx_TRIM.TRIMCH2P[5:0] from 0 to 0x3F. After each modification, wait for about 100us, read PGAx_TRIM.TRIMCH2FLAG, and record the value of PGAx_TRIM.TRIMCH2P[5:0] at the moment when this flag flips (from 0->1 or from 1->0) (assume this value is Y3);

10. Input (VDD-0.3)V to the VINN pin, sequentially configure PGAx_TRIM.TRIMCH2N[5:0] from 0 to 0x3F. After each modification, wait for about 100us, read PGAx_TRIM.TRIMCH2FLAG, and record the value of PGAx_TRIM.TRIMCH2N[5:0] at the moment when this flag flips (from 0->1 or from 1->0) (assume this value is Y4);

11. Write the recorded values of Y1, Y2, Y3, Y4 to non-volatile memory (e.g., FLASH). After each power-up, before enabling the PGA, load these values sequentially into PGAx_TRIM.TRIMCH1P[5:0], PGAx_TRIM.TRIMCH1N[5:0], PGAx_TRIM.TRIMCH2P[5:0], PGAx_TRIM.TRIMCH2N[5:0] to avoid repeated calibration.

20.2.5 PGA TIMER Control Switching Mode

In some applications, the master-slave input switching of PGA1 and PGA2 can be controlled through ATIM1_CC6. ATIM2_CC6 controls the input switching of PGA3 and PGA4.

- When PGAx_CTRL.TCMEN is set to 0, the automatic switching input function is not used by default. That is, PGAx will always select the ports configured in PGAx_CTRL.VPSEL[3:0] and PGAx_CTRL.VMSEL[3:0] as inputs.
- When PGAx_CTRL.TCMEN is set to 1, the automatic switching input function is enabled by default. For PGA1 and PGA2, when ATIM1_CC6 is high, PGAx will select the ports configured in PGAx_CTRL.VPSSEL[3:0]/PGAx_CTRL.VMSSEL[3:0] as inputs; otherwise, it will use PGAx_CTRL.VPSEL[3:0]/PGAx_CTRL.VMSEL[3:0]. For PGA3 and PGA4, when ATIM2_CC6 is high, PGAx will select the ports configured in PGAx_CTRL.VPSSEL[3:0]/PGAx_CTRL.VMSSEL[3:0] as inputs; otherwise, it will use PGAx_CTRL.VPSEL[3:0]/PGAx_CTRL.VMSEL[3:0].

When enabling the automatic switching input function, configure the automatic switching process as follows.

- Enable automatic switching function PGAx_CTRL.TCM_EN
- Configure the MUX settings for the two conversions (VPSEL, VMSEL, VPSSEL, VMSSEL)
- Start ATIM and PGA

20.2.6 PGA Independent Write Protection

By configuring the PGA_LOCK register, the write protection of the PGA can be independently set. Once write protection is enabled, software will not be able to write to the corresponding PGAx_CTRL and PGAx_TRIM registers, and the write protection can only be disabled after a chip reset.

20.3 PGA Register

20.3.1 PGA Register Overview

Table 20-1 PGA Overview of PGA Registers

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	PGA1_CTRL	Reserved			TRIMEN	VMSSEL[3:0]				VPSSEL[3:0]				VMSEL[3:0]				VPSEL[3:0]				Reserved	CH2GAIN[2:0]			Reserved	CH1GAIN[2:0]			TCMEN	DIFFEN	CH2EN	CH1EN
	Reset value				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		1	0	0		1	0	0	0	0	0	0
0x04	PGA1_TRIM	Reserved						TRIM2FLAG	TRIM1FLAG	TRIMCH2N[4:0]						TRIMCH2P[4:0]						TRIMCH1N[4:0]						TRIMCH1P[4:0]					
	Reset value							0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x10	PGA2_CTRL	Reserved			TRIM_EN	VMSSEL[3:0]				VPSSEL[3:0]				VMSEL[3:0]				VPSEL[3:0]				Reserved	CH2GAIN[2:0]			Reserved	CH1GAIN[2:0]			TCMEN	DIFFEN	CH2EN	CH1EN
	Reset value				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		1	0	0		1	0	0	0	0	0	0
0x14	PGA2_TRIM	Reserved						TRIM2FLAG	TRIM1FLAG	TRIMCH2N[4:0]						TRIMCH2P[4:0]						TRIMCH1N[4:0]						TRIMCH1P[4:0]					
	Reset value							0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x20	PGA3_CTRL	Reserved			TRIM_EN	VMSSEL[3:0]				VPSSEL[3:0]				VMSEL[3:0]				VPSEL[3:0]				Reserved	CH2GAIN[2:0]			Reserved	CH2GAIN[2:0]			TCM_EN	DIFFEN	CH2EN	CH1EN
	Reset value				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		1	0	0		1	0	0	0	0	0	0
0x24	PGA3_TRIM	Reserved						TRIM2FLAG	TRIM1FLAG	TRIMCH2N[4:0]						TRIMCH2P[4:0]						TRIMCH1N[4:0]						TRIMCH1P[4:0]					
	Reset value							0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x30	PGA4_CTRL	Reserved			TRIM_EN	VMSSEL[3:0]				VPSSEL[3:0]				VMSEL[3:0]				VPSEL[3:0]				Reserved	CH2GAIN[2:0]			Reserved	CH1GAIN[2:0]			TCMEN	DIFFEN	CH2EN	CH1EN
	Reset value				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		1	0	0		1	0	0	0	0	0	0
0x34	PGA4_TRIM	Reserved						TRIM2FLAG	TRIM1FLAG	TRIMCH2N[4:0]						TRIMCH2P[4:0]						TRIMCH1N[4:0]						TRIMCH1P[4:0]					
	Reset value							0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x40	PGA_LOCK	Reserved																												PGA4LK	PGA3LK	PGA2LK	PGA1LK

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reset value																													0	0	0	0
0x50	PGA_BIASEN	Reserved																															IBIASEN
	Reset value																																0

20.3.2 PGA1 Control Register (PGA1_CTRL)

Offset address: 0x00

Reset value: 0x0000 0440

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved			TRIMEN	VMSSEL[3:0]				VPSSEL[3:0]				VMSEL[3:0]			
			rw	rw				rw				rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VPSEL[3:0]				Reserved	CH2GAIN[2:0]			Reserved	CH1GAIN[2:0]			TCMEN	DIFFEN	CH2EN	CH1EN
rw					rw				rw			rw	rw	rw	rw

Field offset	Register name	Description
31:29	Reserved	Reserved, the reset value must be maintained
28	TRIMEN	PGA1 MOS Trimming Enable. 0: Disable the trimming function of PGA1 MOS; 1: Enable the trimming function of PGA1 MOS.
27:24	VMSSEL[3:0]	PGA1 Inverted secondary input selection: 0000: PF8； 0001: PF10； 0010: PC3； 0011: PA2； 0100: PA3； 0101: PA5； 0110: PC5； 0111: dac2_iout； 1000: dac6_iout；
23:20	VPSSEL[3:0]	PGA1 Non inverted secondary input selection: 0000: PF7； 0001: PC2； 0010: PA1； 0011: PA3； 0100: PA5；

Field offset	Register name	Description
		0101: PA7； 0110: PB1； 0111: dac1_iout； 1000: dac5_iout；
19:16	VMSEL[3:0]	PGA1 Inverted input selection: 0000: PF8； 0001: PF10； 0010: PC3； 0011: PA2； 0100: PA3； 0101: PA5； 0110: PC5； 0111: dac2_iout； 1000: dac6_iout；
15:12	VPSEL[3:0]	PGA1 Non inverted input selection: 0000: PF7； 0001: PC2； 0010: PA1； 0011: PA3； 0100: PA5； 0101: PA7； 0110: PB1； 0111: dac1_iout； 1000: dac5_iout；
11	Reserved	Reserved, the reset value must be maintained
10:8	CH2GAIN [2:0]	PGA1 channel 2 programmable amplifier gain value: 000: Gain of 1 in internal PGA1 single-ended mode and gain of 2 in differential mode; 001: Gain of 2 in internal PGA1 single-ended mode and gain of 4 in differential mode; 010: Gain of 4 in internal PGA1 single-ended mode and gain of 8 in differential mode; 011: Gain of 8 in internal PGA1 single-ended mode and gain of 16 in differential mode; 100: Gain of 12 in internal PGA1 single-ended mode and gain of 24 in differential mode; 101: Gain of 16 in internal PGA1 single-ended mode and gain of 32 in differential mode; 110: Gain of 24 in internal PGA1 single-ended mode and gain of 48 in differential mode; 111: Gain of 32 in internal PGA1 single-ended mode and gain of 64 in differential mode.

Field offset	Register name	Description
		Note: This parameter is invalid in differential mode.
7	Reserved	Reserved, the reset value must be maintained
6:4	CH1GAIN [2:0]	PGA1 channel 1 programmable amplifier gain value: 000: Gain of 1 in internal PGA1 single-ended mode and gain of 2 in differential mode; 001: Gain of 2 in internal PGA1 single-ended mode and gain of 4 in differential mode; 010: Gain of 4 in internal PGA1 single-ended mode and gain of 8 in differential mode; 011: Gain of 8 in internal PGA1 single-ended mode and gain of 16 in differential mode; 100: Gain of 12 in internal PGA1 single-ended mode and gain of 24 in differential mode; 101: Gain of 16 in internal PGA1 single-ended mode and gain of 32 in differential mode; 110: Gain of 24 in internal PGA1 single-ended mode and gain of 48 in differential mode; 111: Gain of 32 in internal PGA1 single-ended mode and gain of 64 in differential mode.
3	TCMEN	Timer controlled Mux mode enable. This bit is set or cleared by software to control the automatic switching of primary and secondary inputs (VPSEL, VMSEL, VPSSEL, VMSSEL). ATIM1_CC6 automatic switching to PGA1 input. 0：Disable automatic switching. 1：Ensable automatic switching.
2	DIFFEN	PGA1 Differential mode enable. 0：Disable； 1：Enable.
1	CH2EN	PGA1 Programmable gain amplifier channel 2 enable. 0：Disable； 1：Enable.
0	CH1EN	PGA1 Programmable gain amplifier channel 1 enable. 0：Disable； 1：Enable.

20.3.3 PGA1 Trim Register (PGA1_TRIM)

Offset address: 0x04

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved						TRIMCH2 FLAG	TRIMCH1 FLAG	TRIMCH2N[5:0]						TRIMCH2P[5:4]	

						r	r			rw				rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRIMCH2P[3:0]				TRIMCH1N[5:0]						TRIMCH1P[5:0]					
rw				rw						rw					

Field offset	Register name	Description
31:26	Reserved	Reserved, the reset value must be maintained
25	TRIMCH2FLAG	PGA1 Channel 2 Trim Finish Flag
24	TRIMCH1FLAG	PGA1 Channel 1 Trim Finish Flag
23:18	TRIMCH2N[5:0]	NMOS trimming value for PGA1 Channel 2
17:12	TRIMCH2P[5:0]	PMOS trimming value for PGA1 Channel 2
11:6	TRIMCH1N[5:0]	NMOS trimming value for PGA1 Channel 1
5:0	TRIMCH1P[5:0]	PMOS trimming value for PGA1 Channel 1

20.3.4 PGA2 Control Register (PGA2_CTRL)

Offset address: 0x10

Reset value: 0x0000 0440

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved			TRIMEN	VMSSEL[3:0]				VPSSEL[3:0]				VMSEL[3:0]			
			rw	rw				rw				rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VPSEL[3:0]				Reserved	CH2GAIN[2:0]			Reserved	CH1GAIN[2:0]			TCMEN	DIFFEN	CH1EN	CH1EN
rw					rw				rw			rw	rw	rw	rw

Field offset	**Register name**	**Description**
31:29	Reserved	Reserved, the reset value must be maintained
28	TRIMEN	PGA2 MOS Trimming Enable. 0: Disable the trimming function of PGA2 MOS; 1: Enable the trimming function of PGA2 MOS.
27:24	VMSSEL[3:0]	PGA2 Inverted secondary input selection. 0000: PB0； 0001: PF9； 0010: PF10； 0011: PA2； 0100: PA3； 0101: PA5； 0110: PC5；

Field offset	Register name	Description
		0111: PB15 1000:dac2_iout； 1001:dac6_iout；
23:20	VPSSEL[3:0]	PGA2 Non inverted secondary input selection. 0000: PF7； 0001: PC3； 0010: PA1； 0011: PA2； 0100: PA4； 0101: PA7； 0110: PB0； 0111: PE8； 1000: PB14 1001: PD12 1010: PD14 1011: dac1_iout； 1100: dac5_iout；
19:16	VMSEL[3:0]	PGA2 Inverted input selection. 0000: PB0； 0001: PF9； 0010: PF10； 0011: PA2； 0100: PA3； 0101: PA5； 0110: PC5； 0111: PB15 1000:dac2_iout； 1001:dac6_iout；；
15:12	VPSEL[3:0]	PGA2 Non inverted input selection. 0000: PF7； 0001: PC3； 0010: PA1； 0011: PA2； 0100: PA4； 0101: PA7； 0110: PB0； 0111: PE8； 1000: PB14 1001: PD12 1010: PD14 1011: dac1_iout； 1100: dac5_iout；

Field offset	Register name	Description
11	Reserved	Reserved, the reset value must be maintained
10:8	CH2GAIN[2:0]	PGA2 channel 2 programmable amplifier gain value: 000: Gain of 1 in internal PGA2 single-ended mode and gain of 2 in differential mode; 001: Gain of 2 in internal PGA2 single-ended mode and gain of 4 in differential mode; 010: Gain of 4 in internal PGA2 single-ended mode and gain of 8 in differential mode; 011: Gain of 8 in internal PGA2 single-ended mode and gain of 16 in differential mode; 100: Gain of 12 in internal PGA2 single-ended mode and gain of 24 in differential mode; 101: Gain of 16 in internal PGA2 single-ended mode and gain of 32 in differential mode; 110: Gain of 24 in internal PGA2 single-ended mode and gain of 48 in differential mode; 111: Gain of 32 in internal PGA2 single-ended mode and gain of 64 in differential mode. *Note: This parameter is invalid in differential mode.*
7	Reserved	Reserved, the reset value must be maintained
6:4	CH1GAIN[2:0]	PGA2 channel 1 programmable amplifier gain value: 000: Gain of 1 in internal PGA2 single-ended mode and gain of 2 in differential mode; 001: Gain of 2 in internal PGA2 single-ended mode and gain of 4 in differential mode; 010: Gain of 4 in internal PGA2 single-ended mode and gain of 8 in differential mode; 011: Gain of 8 in internal PGA2 single-ended mode and gain of 16 in differential mode; 100: Gain of 12 in internal PGA2 single-ended mode and gain of 24 in differential mode; 101: Gain of 16 in internal PGA2 single-ended mode and gain of 32 in differential mode; 110: Gain of 24 in internal PGA2 single-ended mode and gain of 48 in differential mode; 111: Gain of 32 in internal PGA2 single-ended mode and gain of 64 in differential mode.
3	TCMEN	Timer controlled Mux mode enable. This bit is set or cleared by software to control the automatic switching of primary and secondary inputs (VPSEL, VMSEL, VPSSEL, VMSSEL). ATIM1_CC6 automatic switching to PGA2 input. 0：Disable automatic switching.

Field offset	Register name	Description
		1：Ensable automatic switching.
2	DIFFEN	PGA2 Differential mode enable. 0：Disable； 1：Enable.
1	CH2EN	PGA2 Programmable gain amplifier channel 2 enable. 0：Disable； 1：Enable.
0	CH1EN	PGA2 Programmable gain amplifier channel 1 enable. 0：Disable； 1：Enable.

20.3.5 PGA2 Trim Register (PGA2_TRIM)

Offset address: 0x14

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved						TRIMCH2 FLAG	TRIMCH1 FLAG	TRIMCH2N[5:0]						TRIMCH2P[5:4]	
						r	r	rw						rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRIMCH2P[3:0]				TRIMCH1N[5:0]						TRIMCH1P[5:0]					
rw				rw						rw					

Field offset	Register name	Description
31:26	Reserved	Reserved, the reset value must be maintained
25	TRIMCH2FLAG	PGA2 Channel 2 Trim Finish Flag
24	TRIMCH1FLAG	PGA2 Channel 1 Trim Finish Flag
23:18	TRIMCH2N[5:0]	NMOS trimming value for PGA2 Channel 2
17:12	TRIMCH2P[5:0]	PMOS trimming value for PGA2 Channel 2
11:6	TRIMCH1N[5:0]	NMOS trimming value for PGA2 Channel 1
5:0	TRIMCH1P[5:0]	PMOS trimming value for PGA2 Channel 1

20.3.6 PGA3 Control Register (PGA3_CTRL)

Offset address: 0x20

Reset value: 0x0000 0440

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved			TRIMEN	VMSSEL[3:0]				VPSSEL[3:0]				VMSEL[3:0]			

r			rw	rw				rw				rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VPSEL[3:0]				Reserved	CH2GAIN[2:0]			Reserved	CH1GAIN[2:0]			TCMEN	DIFFEN	CH2EN	CH1EN
rw				r	rw			r	rw			rw	rw	rw	rw

Field offset	Register name	Description
31:29	Reserved	Reserved, the reset value must be maintained
28	TRIMEN	PGA3 MOS Trimming Enable. 0: Disable the trimming function of PGA3 MOS; 1: Enable the trimming function of PGA3 MOS.
27:24	VMSSEL[3:0]	PGA3 Inverted secondary input selection. 0000: PF10； 0001: PC1； 0010: PA2； 0011: PA5； 0100: PC4； 0101: PB2； 0110: PB10； 0111: dac2_iout； 1000: dac6_iout；
23:20	VPSSEL[3:0]	PGA3 Non inverted secondary input selection. 0000: PF7； 0001: PC0； 0010: PC3； 0011: PA1； 0100: PA5； 0101: PA6； 0110: PB0； 0111: PB13； 1000: PC9； 1001: dac1_iout； 1010: dac5_iout；
19:16	VMSEL[3:0]	PGA3 Inverted input selection. 0000: PF10； 0001: PC1； 0010: PA2； 0011: PA5； 0100: PC4； 0101: PB2； 0110: PB10； 0111: dac2_iout；

Field offset	Register name	Description
		1000: dac6_iout；
15:12	VPSEL[3:0]	PGA3 Non inverted input selection. 0000: PF7； 0001: PC0； 0010: PC3； 0011: PA1； 0100: PA5； 0101: PA6； 0110: PB0； 0111: PB13； 1000: PC9； 1001: dac1_iout； 1010: dac5_iout；
11	Reserved	Reserved, the reset value must be maintained
10:8	CH2GAIN[2:0]	PGA3 channel 2 programmable amplifier gain value: 000: Gain of 1 in internal PGA3 single-ended mode and gain of 2 in differential mode; 001: Gain of 2 in internal PGA3 single-ended mode and gain of 4 in differential mode; 010: Gain of 4 in internal PGA3 single-ended mode and gain of 8 in differential mode; 011: Gain of 8 in internal PGA3 single-ended mode and gain of 16 in differential mode; 100: Gain of 12 in internal PGA3 single-ended mode and gain of 24 in differential mode; 101: Gain of 16 in internal PGA3 single-ended mode and gain of 32 in differential mode; 110: Gain of 24 in internal PGA3 single-ended mode and gain of 48 in differential mode; 111: Gain of 32 in internal PGA3 single-ended mode and gain of 64 in differential mode. *Note: This parameter is invalid in differential mode.*
7	Reserved	Reserved, the reset value must be maintained
6:4	CH1GAIN[2:0]	PGA3 channel 1 programmable amplifier gain value: 000: Gain of 1 in internal PGA3 single-ended mode and gain of 2 in differential mode; 001: Gain of 2 in internal PGA3 single-ended mode and gain of 4 in differential mode; 010: Gain of 4 in internal PGA3 single-ended mode and gain of 8 in differential mode; 011: Gain of 8 in internal PGA3 single-ended mode and gain of 16 in differential mode;

Field offset	Register name	Description
		100: Gain of 12 in internal PGA3 single-ended mode and gain of 24 in differential mode; 101: Gain of 16 in internal PGA3 single-ended mode and gain of 32 in differential mode; 110: Gain of 24 in internal PGA3 single-ended mode and gain of 48 in differential mode; 111: Gain of 32 in internal PGA3 single-ended mode and gain of 64 in differential mode.
3	TCMEN	Timer controlled Mux mode enable. This bit is set or cleared by software to control the automatic switching of primary and secondary inputs (VPSEL, VMSEL, VPSSEL, VMSSEL). ATIM1_CC6 automatic switching to PGA3 input. 0：Disable automatic switching. 1：Ensable automatic switching.
2	DIFFEN	PGA3 Differential mode enable. 0：Disable； 1：Enable.
1	CH2EN	PGA3 Programmable gain amplifier channel 2 enable. 0：Disable； 1：Enable.
0	CH1EN	PGA3 Programmable gain amplifier channel 1 enable. 0：Disable； 1：Enable.

20.3.7 PGA3 Trim Register (PGA3_TRIM)

Offset address: 0x24

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved						TRIMCH2 FLAG	TRIM1 FLAG	TRIMCH2N[5:0]						TRIMCH2P[5:4]	
						r	r	rw						rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRIMCH2P[3:0]				TRIMCH1N[5:0]						TRIMCH1P[5:0]					
rw				rw						rw					

Field offset	Register name	Description
31:26	Reserved	Reserved, the reset value must be maintained
25	TRIMCH2FLAG	PGA3 Channel 2 Trim Finish Flag
24	TRIMCH1FLAG	PGA3 Channel 1 Trim Finish Flag

Field offset	Register name	Description
23:18	TRIMCH2N[5:0]	NMOS trimming value for PGA3 Channel 2
17:12	TRIMCH2P[5:0]	PMOS trimming value for PGA3 Channel 2
11:6	TRIMCH1N[5:0]	NMOS trimming value for PGA3 Channel 1
5:0	TRIMCH1P[5:0]	PMOS trimming value for PGA3 Channel 1

20.3.8 PGA4 Control Register (PGA4_CTRL)

Offset address: 0x30

Reset value: 0x0000 0440

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved			TRIMEN	VMSSEL[3:0]				VPSSEL[3:0]				VMSEL[3:0]			
			rw	rw				rw				rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VPSEL[3:0]				Reserved	CH2GAIN[2:0]			Reserved	CH1GAIN[2:0]			TCMEN	DIFFEN	CH2EN	CH1EN
rw					rw				rw			rw	rw	rw	rw

Field offset	Register name	Description
31:29	Reserved	Reserved, the reset value must be maintained
28	TRIMEN	PGA4 MOS Trimming Enable. 0: Disable the trimming function of PGA4 MOS; 1: Enable the trimming function of PGA4 MOS.
27:24	VMSSEL[3:0]	PGA4 Inverted secondary input selection. 0000: PF10； 0001: PA1； 0010: PB1； 0011: PB10； 0100: PD8； 0101: PC9； 0110: dac2_iout； 0111: dac6_iout；
23:20	VPSSEL[3:0]	PGA4 Non inverted secondary input selection. 0000: PC3； 0001: PA4； 0010: PC5； 0011: PB11； 0100: PB12； 0101: PB13； 0110: PD9；

Field offset	Register name	Description

		0111: PD11;
		1000: dac1_iout;
		1001: dac5_iout;

19:16	VMSEL[3:0]	PGA4 Inverted input selection.
		0000: PF10;
		0001: PA1;
		0010: PB1;
		0011: PB10;
		0100: PD8;
		0101: PC9;
		0110: dac2_iout;
		0111: dac6_iout;

15:12	VPSEL[3:0]	PGA4 Non inverted input selection.
		0000: PC3;
		0001: PA4;
		0010: PC5;
		0011: PB11;
		0100: PB12;
		0101: PB13;
		0110: PD9;
		0111: PD11;
		1000: dac1_iout;
		1001: dac5_iout;

11	Reserved	Reserved, the reset value must be maintained
10:8	CH2GAIN[2:0]	PGA4 channel 2 programmable amplifier gain value: 000: Gain of 1 in internal PGA4 single-ended mode and gain of 2 in differential mode; 001: Gain of 2 in internal PGA4 single-ended mode and gain of 4 in differential mode; 010: Gain of 4 in internal PGA4 single-ended mode and gain of 8 in differential mode; 011: Gain of 8 in internal PGA4 single-ended mode and gain of 16 in differential mode; 100: Gain of 12 in internal PGA4 single-ended mode and gain of 24 in differential mode; 101: Gain of 16 in internal PGA4 single-ended mode and gain of 32 in differential mode; 110: Gain of 24 in internal PGA4 single-ended mode and gain of 48 in differential mode; 111: Gain of 32 in internal PGA4 single-ended mode and gain of 64 in differential mode. *Note: This parameter is invalid in differential mode.*

Field offset	Register name	Description
7	Reserved	Reserved, the reset value must be maintained
6:4	CH1GAIN[2:0]	PGA4 channel 1 programmable amplifier gain value: 000: Gain of 1 in internal PGA4 single-ended mode and gain of 2 in differential mode; 001: Gain of 2 in internal PGA4 single-ended mode and gain of 4 in differential mode; 010: Gain of 4 in internal PGA4 single-ended mode and gain of 8 in differential mode; 011: Gain of 8 in internal PGA4 single-ended mode and gain of 16 in differential mode; 100: Gain of 12 in internal PGA4 single-ended mode and gain of 24 in differential mode; 101: Gain of 16 in internal PGA4 single-ended mode and gain of 32 in differential mode; 110: Gain of 24 in internal PGA4 single-ended mode and gain of 48 in differential mode; 111: Gain of 32 in internal PGA4 single-ended mode and gain of 64 in differential mode.
3	TCMEN	Timer controlled Mux mode enable. This bit is set or cleared by software to control the automatic switching of primary and secondary inputs (VPSEL, VMSEL, VPSSEL, VMSSEL). ATIM1_CC6 automatic switching to PGA4 input. 0：Disable automatic switching. 1：Ensable automatic switching.
2	DIFFEN	PGA4 Differential mode enable. 0：Disable； 1：Enable.
1	CH2EN	PGA4 Programmable gain amplifier channel 2 enable. 0：Disable； 1：Enable.
0	CH1EN	PGA4 Programmable gain amplifier channel 1 enable. 0：Disable； 1：Enable.

20.3.9 PGA4 Trim Register (PGA4_TRIM)

Offset address: 0x34

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved						TRIMCH2 FLAG	TRIMCH1 FLAG	TRIM2N[5:0]						TRIMCH2P[5:4]	
						r	r	rw						rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRIMCH2P[3:0]				TRIMCH1N[5:0]						TRIMCH1P[5:0]					
rw				rw						rw					

Field offset	Register name	Description
31:26	Reserved	Reserved, the reset value must be maintained
25	TRIMCH2FLAG	PGA4 Channel 2 Trim Finish Flag
24	TRIMCH1FLAG	PGA4 Channel 1 Trim Finish Flag
23:18	TRIMCH2N[5:0]	NMOS trimming value for PGA4 Channel 2
17:12	TRIMCH2P[5:0]	PMOS trimming value for PGA4 Channel 2
11:6	TRIMCH1N[5:0]	NMOS trimming value for PGA4 Channel 1
5:0	TRIMCH1P[5:0]	PMOS trimming value for PGA4 Channel 1

20.3.10 PGA Lock Register (PGA_LOCK)

Offset address: 0x40

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												PGA4LK	PGA3LK	PGA2LK	PGA1LK
												rw	rw	rw	rw

Field offset	Register name	Description
31:4	Reserved	Reserved, the reset value must be maintained
3	PGA4LK	PGA4 Lock Bit (PGA4 lock bit) After reset, this bit can only be written once and can only be cleared by a system reset. 0: PGA4_CTRL and PGA4_TRIM registers are readable and writable; 1: PGA4_CTRL and PGA4_TRIM registers are read-only
2	PGA3LK	PGA3 Lock Bit (PGA3 lock bit) After reset, this bit can only be written once and can only be cleared by a system reset. 0: PGA3_CTRL and PGA3_TRIM registers are readable and writable; 1: PGA3_CTRL and PGA3_TRIM registers are read-only
1	PGA2LK	PGA2 Lock Bit (PGA2 lock bit)

Field offset	Register name	Description
		After reset, this bit can only be written once and can only be cleared by a system reset. 0: PGA2_CTRL and PGA2_TRIM registers are readable and writable; 1: PGA2_CTRL and PGA2_TRIM registers are read-only
0	PGA1LK	PGA1 Lock Bit (PGA1 lock bit) After reset, this bit can only be written once and can only be cleared by a system reset. 0: PGA1_CTRL and PGA1_TRIM registers are readable and writable; 1: PGA1_CTRL and PGA1_TRIM registers are read-only

20.3.11 PGA Bias Current Source Enable (PGA_BIASEN)

Offset address: 0x50

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															IBIASEN
															rw

Field offset	Register name	Description
31:1	Reserved	Reserved, the reset value must be maintained
0	IBIASEN	PGA bias current source enable bit. 0: Disable. 1: Enable.

21 Filter Math Accelerator（FMAC）

21.1 FMAC Introduction

The mathematical acceleration unit of the filter performs arithmetic operations on vectors, including a multiplier, an accumulator, and address generation logic that allows indexing of vector elements in local memory. This unit supports input and output circular buffers to facilitate the implementation of digital filters such as Finite Impulse Response (FIR) and Infinite Impulse Response (IIR) filters.

This unit allows the processor to be freed from frequent or lengthy filtering operations, thereby enabling the processor to perform other tasks. In many cases, it can accelerate such calculations compared to software implementations, speeding up the processing of critical time tasks.

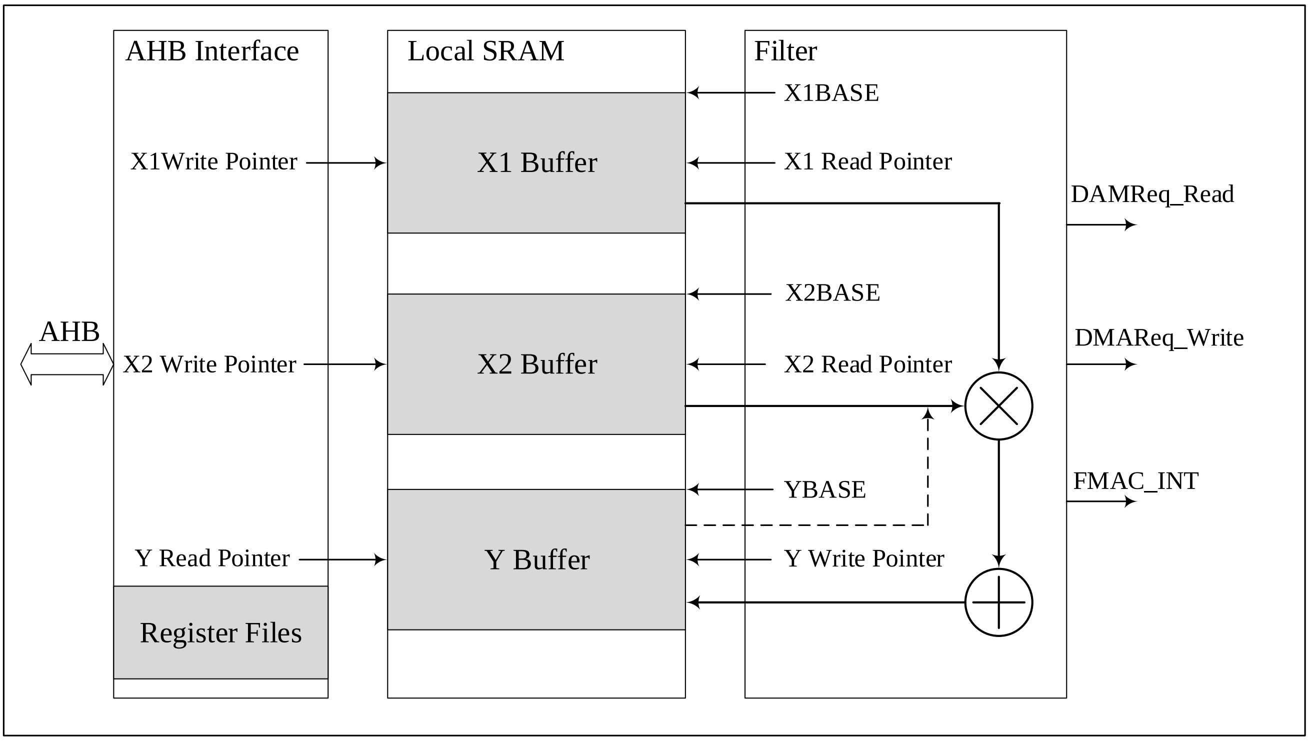
21.2 FMAC Main Features

- 16 x 16-bit multiplier
- 24 + 2-bit accumulator with addition and subtraction capabilities
- 16-bit fixed-point input and output data
- 256 x 16-bit data buffer
- Up to three data buffers can be defined in memory (two inputs, one output), defined by programmable base address pointers and related size registers
- Input and output buffers can be recycled
- Filter functions: FIR, IIR (direct form 1)
- Vector functions: dot product, convolution, correlation
- Supports DMA for reading and writing data

21.3 FMAC Function Description

21.3.1 General Description

Figure 21-1 FMAC Block Diagram



This unit consists of a fixed-point multiplier, accumulator, and buffer.

The two vector input data of the multiplier are loaded from the buffer by the core or DMA, and then multiplied by the multiplier using dot product computation.

The accumulator accumulates the output result of the multiplier, and after the calculation is completed, the output result in the accumulator is stored in the buffer for the core or DMA to read.

The module buffer can be divided into input data buffer (X1), coefficient buffer (X2), and output data buffer (Y). The coefficient buffer stores the feedforward and feedback filter coefficients of FIR and IIR filters, the input data buffer stores the input vector data, and the output data buffer stores the results calculated by FIR and IIR filters, as well as the pre-stored filter output data of the IIR filter.

The output result of the Finite Impulse Response (FIR) filter is the dot product of the input vector data and the feedforward filter coefficients. The input vector data is continuously updated, meaning the earliest sampled data is discarded, and a new sample data is added to form the new vector data.

The output result of the Infinite Impulse Response (IIR) filter is the product between the feedback filter coefficients and the previous filter output data, added to the convolution of FIR.

21.3.2 Buffer

This unit contains a 256 x 16-bit read/write storage area: input values are stored in buffer X1 and buffer X2, while output values are stored in buffer Y. The addresses and sizes of the buffers are specifically described as follows:

- X1BASE: Base address of buffer X1;
- X2BASE: Base address of buffer X2;
- YBASE: Base address of buffer Y;

The length of the buffers is as follows:

- X1BUFSIZE: Number of 16-bit words allocated for buffer X1;
- X2BUFSIZE: Number of 16-bit words allocated for buffer X2;
- YBUFSIZE: Number of 16-bit words allocated for buffer Y;

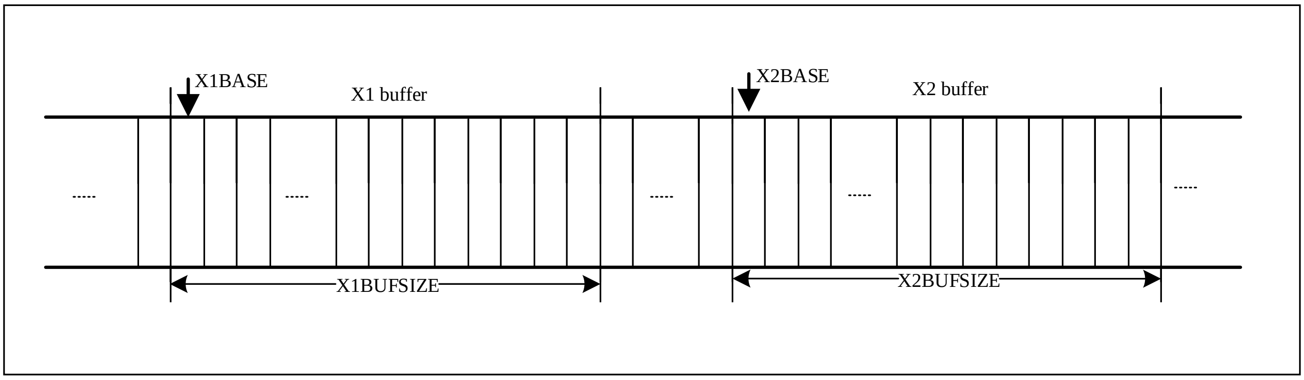
The above parameters need to be configured into the corresponding FMAC_X1BUFCFG, FMAC_X2BUFCFG, FMAC_YBUFCFG registers before starting the FMAC module. The base addresses of the three buffers can be freely configured within the range of 0x00-0xFF, and the total length of the three buffers must be less than 256. There are no restrictions on the position and size of the buffers, which may lead to overlapping of the three buffer locations. To avoid abnormal operation, it is not recommended to overlap the buffers when configuring the base addresses and lengths of the buffers.

Prior to initiating the conversion in the FMAC module, corresponding data can be preloaded into the buffers. Buffer X1 preloads the input vector data, buffer X2 preloads the filter coefficients, and buffer Y preloads the previous filter output data (only required for IIR filters). Data loading into the target buffer indicated by the write pointer is accomplished by writing data to the FMAC_WDAT register. After completing a write operation, the write pointer is incremented. If the pointer reaches the end of the allocated buffer space, the write pointer returns to the base address.

If buffer recycling is required, an optional reserved area d can be added to the buffer. Additionally, to adjust the operation of the CPU or DMA, it is necessary to set the watermark threshold size. To meet the performance requirements of the application, the values of d and the watermark need to be configured reasonably. Typically, for higher data throughput, the input buffer should not be empty, so d should be slightly larger than the watermark to accommodate any interrupts or DMA delays. On the other hand, if the speed of incoming data is slower than the processing speed, the input buffer can be empty and wait for the next data to be written. In this case, d and the watermark can be set equal to ensure that the input buffer does not overflow.

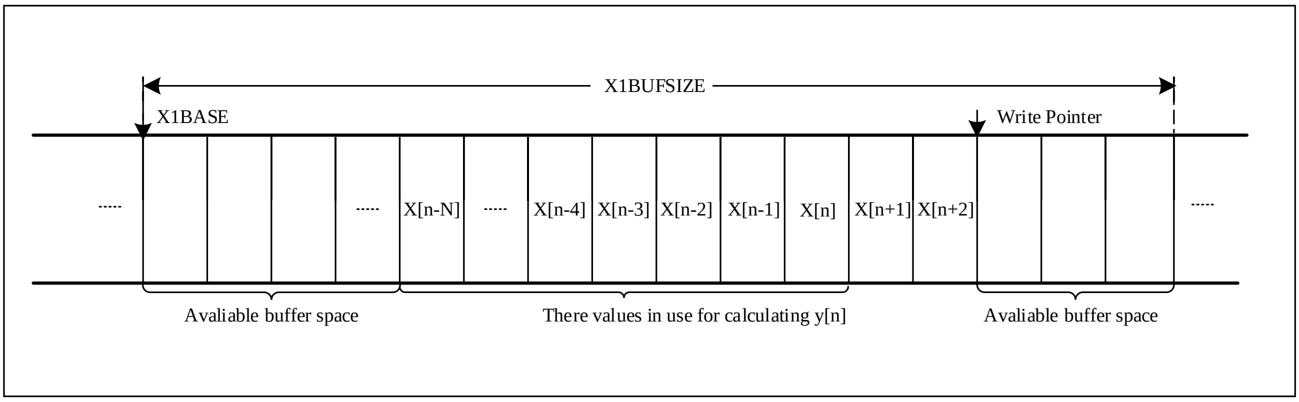
21.3.3 Input Buffer

Figure 21-2 Input Buffer Schematic



As shown in Figure Figure 21-2, the X1 buffer and X2 buffer are used to store the data input to the multiplier in the FMAC module. For each multiplication operation, the multiplier extracts one data from the X1 buffer and one from the X2 buffer, and multiplies them. After each multiplication operation, the control unit generates a read pointer address offset based on the current function, which is relative to the buffer base address. The read pointer address offset is controlled by hardware and automatically performed.

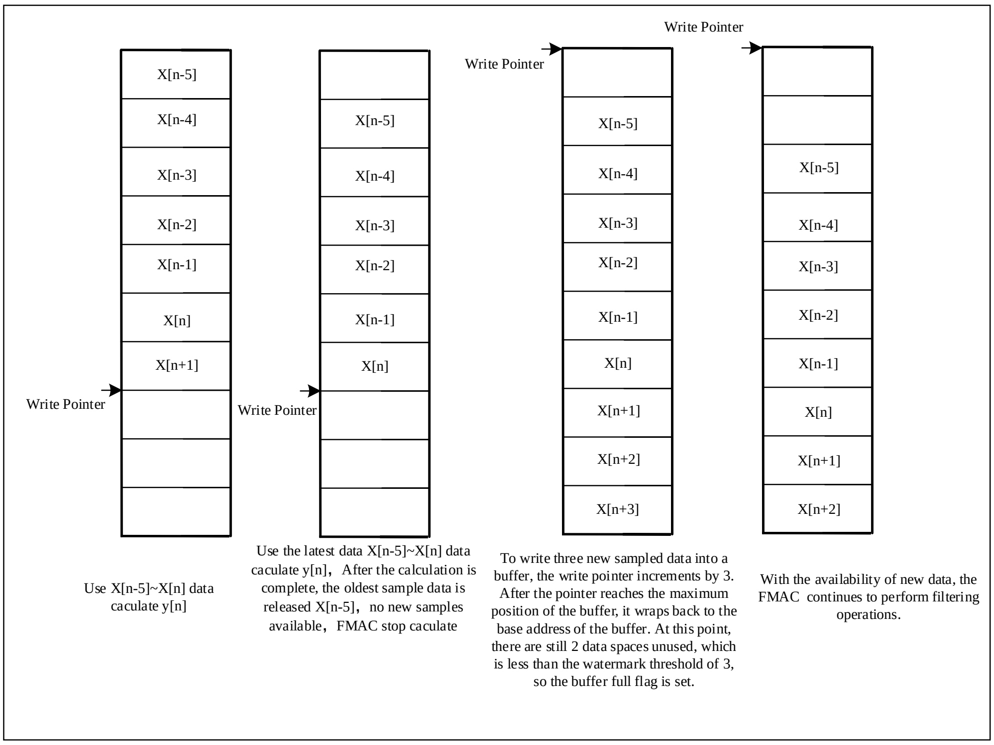
Figure 21-3 Schematic of Circular Input Buffer



The X1 buffer can operate in circular mode, as shown in Figure Figure 21-3. In the presence of available space, new data is continuously transferred to the input buffer. For digital filters, preloading the buffer is optional. When the filtering operation begins, if the input sample values have not been written to the X1 buffer, the buffer is marked as 'empty'. The FMAC module generates requests to load new sample data to the DMA or CPU, which continue until there is sufficient sample data for filtering calculations (FIR/IIR).

The X2 buffer only operates in non-circular mode and is used to store filter coefficients, which typically need to be preloaded.

Figure 21-4 Schematic of Circular Input Buffer Operation



Use X[n-5]~X[n] data caculate y[n]

Use the latest data X[n-5]~X[n] data caculate y[n]，After the calculation is complete, the oldest sample data is released X[n-5]，no new samples available，FMAC stop caculate

To write three new sampled data into a buffer, the write pointer increments by 3. After the pointer reaches the maximum position of the buffer, it wraps back to the base address of the buffer. At this point, there are still 2 data spaces unused, which is less than the watermark threshold of 3, so the buffer full flag is set.

With the availability of new data, the FMAC continues to perform filtering operations.

When the buffer operates in circular mode, its space (X1BUFSIZE) should be larger than the number of elements used by the current calculation function to ensure that there are new values available in the buffer for computation. The schematic of the circular input buffer operation during filtering is shown in Figure 21-4. When computing the output y[n], the FMAC calls N+1 input samples from x[n-N] to x[n]. After computing y[n], it calls the input sample sequence from x[n-N+1] to x[n+1] to begin computing y[n+1]. The earliest input sample (x[n-N]) is discarded, and a new sample value (x[n+1]) is added.

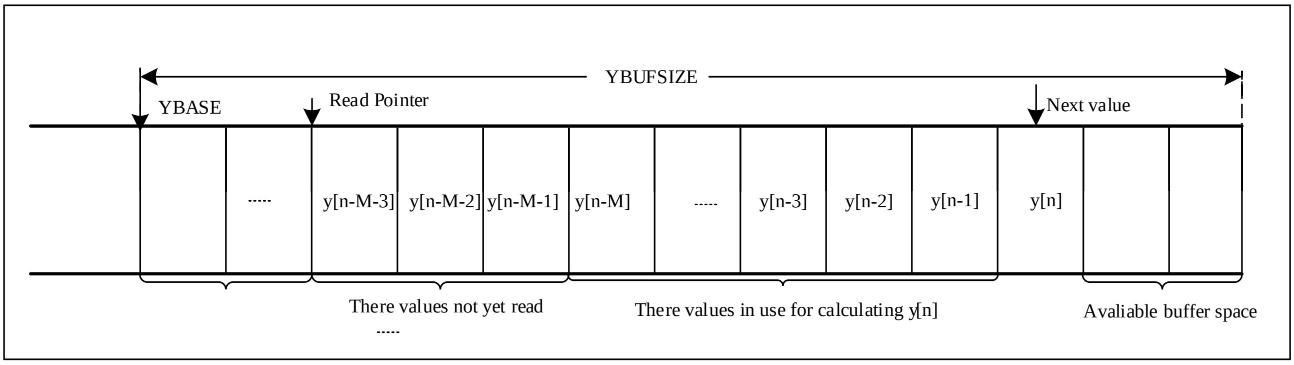
When computing y[n+1], the CPU or DMA must ensure that the new sample data x[n+1] is available in the buffer space. If x[n+1] is not available, the hardware in the FMAC module automatically marks the buffer as 'empty', blocking the calculation of the current y[n+1] until the buffer is written with the new sample value. The X1 buffer will not generate an underflow signal. If the input data is sampled by a timer or other external device where the sampling data input rate is typically slower than the filtering processing speed, the buffer usually operates in an 'empty' state (no data involved in the computation).

FMAC_X1BUFCFG. The X1BUFWM[1:0] bits configure the watermark threshold value. If the available free space in the buffer (free space refers to the space that does not contain data or has been used) is less than the watermark threshold, the buffer is considered 'full'. When an FMAC interrupt or DMA write enable is active, if the FMAC_STS.X1BUFFF bit is not set, an interrupt or DMA write data request is generated. Without considering the risk of overflow, when an interrupt or DMA write data request occurs, several data can be transferred to the buffer. However, in case of overflow, the FMAC_STS.OVF flag is set, the written data is ignored, and the write pointer does not increment. Figure Figure 21-4 illustrates the changes in the X1 buffer during the operation of an FIR filter with a length of 8, with the watermark set to 4.

21.3.4 Output Buffer

The Y buffer is the output buffer used to store the accumulated results. The processor or DMA controller can read the output values, after which the buffer space will be released. Each time data is read from the register, data is obtained from the address indicated by the read pointer. The read pointer is then incremented, and when the pointer reaches the end of the allocated Y buffer space, it returns to the base address.

Figure 21-5 Loopback Buffer Diagram



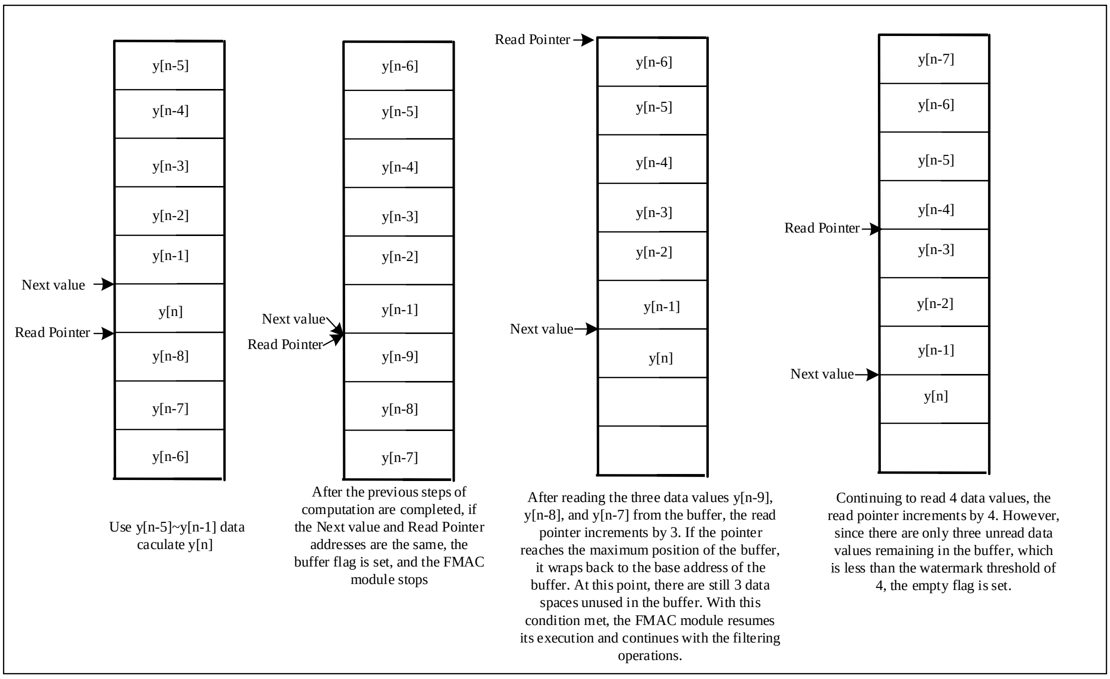
Translate the content in parentheses to English: (The Y buffer can also function as a circular buffer. If the address of the next output value is the same as the address indicated by the read pointer (unread data), the buffer is marked as full, filtering is stopped until the output value is read.

In an IIR filter, M previous output sample values (y[n-M] to y[n-1]) are used to calculate the next output sample y[n]. When a new output sample value is generated, the earliest output sample value y[n-M] is discarded.

If the number of unread data in the buffer is less than the watermark threshold programmed in FMAC_YBUFCFG.YBUFWM[1:0] bit field, the buffer will be marked as empty. If the empty flag is not set to 1 (indicating that the output buffer is not empty, i.e., the number of unread data is greater than the watermark threshold), an interrupt or DMA request (if enabled) will be generated to read data from the buffer. The presence of the watermark allows for the transfer of multiple data under a single interrupt without the risk of data shortage. In case of data underflow, the FMAC_STS.UNF flag will be set to 1. In this scenario, the read pointer will not increment, and the read operation will return the content at the address pointed to by the read pointer.

Figure Figure 21-5 llustrates the dynamic process of the Y buffer during filter operation. The filter is a 7-tap IIR filter with a watermark set to 4.

Figure 21-6 Circular Output Buffer Operation Diagram



Use y[n-5]~y[n-1] data caculate y[n]

After the previous steps of computation are completed, if the Next value and Read Pointer addresses are the same, the buffer flag is set, and the FMAC module stops

After reading the three data values y[n-9], y[n-8], and y[n-7] from the buffer, the read pointer increments by 3. If the pointer reaches the maximum position of the buffer, it wraps back to the base address of the buffer. At this point, there are still 3 data spaces unused in the buffer. With this condition met, the FMAC module resumes its execution and continues with the filtering operations.

Continuing to read 4 data values, the read pointer increments by 4. However, since there are only three unread data values remaining in the buffer, which is less than the watermark threshold of 4, the empty flag is set.

21.3.5 Initialization Function

By writing the corresponding value in the FUNC field of the FMAC_PARAMCFG register and setting the START bit, the calculation of the corresponding function can be initiated. The P and Q bit fields must also contain the respective parameter values for each function, without using the R field. Once the function is completed, the START bit will be automatically reset.

During initialization, it is advisable to disable DMA requests and interrupts. Data transfer to the FMAC memory can be accomplished through software or DMA controller transfers.

Loading the X1 buffer

This function starts from the address in X1BASE and preloads N values for the X1 buffer. Continuous writes to the FMAC_WDAT register load the written data into the X1 buffer and increment the write pointer. Upon completion of the function, the write pointer points to the address X1BASE + N. Parameter P represents the number of data values loaded into the X1 buffer, while Q and R are not used. The X1 buffer loading operation is completed after N writes to the FMAC_WDAT register.

Loading the X2 buffer

This function starts from the address in X2BASE and preloads N + 1 + M values for the X2 buffer. Starting from the address in X2BASE, continuous writes to the FMAC_WDAT register load the written data into the X2 buffer and increment the write pointer.

In an IIR filter, N+1 feedforward coefficients and M feedback coefficients are concatenated and loaded into the X2 buffer. The total number of coefficients is equal to N + 1 + M. For FIR filters, there are no feedback coefficients, so

M = 0.

Parameter P contains N+1 feedforward filter coefficients, parameter Q contains M feedback filter coefficients, P starts at the address X2BASE, Q starts at the address X2BASE+N+1, and R is not used. The X2 buffer loading operation is completed after N + 1 + M writes to the FMAC_WDAT register.

Loading the Y buffer

This function starts from the address in YBASE, loads M written data from the FMAC_WDAT register into the Y buffer, increments the write pointer, and upon completion of the function, the read pointer points to the address YBASE + M.

This function can be used to preload the feedback storage parameters of an IIR filter, with P containing M feedback storage parameters, while Q and R are not used. The Y buffer loading operation is completed after M writes to the FMAC_WDAT register.

21.3.6 Filter Function

By writing the corresponding value in the FUNC field of the FMAC_PARAMCFG register and setting the START bit, the corresponding function can be triggered, while the P, Q, and R bit fields must also contain the respective parameter values. Once configured, the filtering function will continue to operate until the START bit is reset by software.

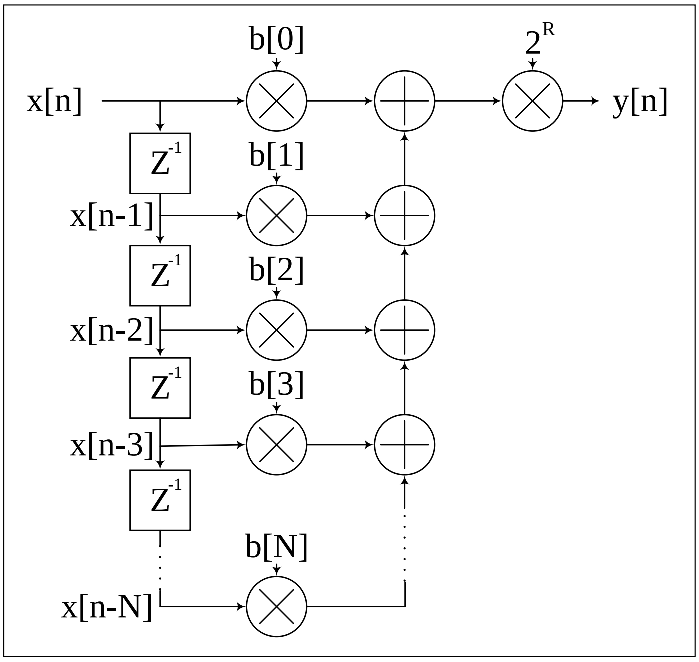
FIR filter: Y = B*X

$$\mathrm{y[n]} = 2^{R} \cdot \sum_{j=0}^{N} \mathrm{b[j]} \cdot \mathrm{x[n-j]}$$

This function is used to implement a Finite Impulse Response (FIR) filter, where vector B represents the coefficients of the FIR filter with a length of N+1. X is an infinite-length input sample vector data. The elements of Y are computed as the dot product yn = B·Xn, where Xn = [xn-N,...,xn].

The FIR filter structure is shown in Figure Figure 21-7.

Figure 21-7 FIR Filter Structure Diagram



FIR filter input:

The X1 buffer contains the elements of vector X, which is a circular buffer of length N + 1 + d.

The X2 buffer contains the elements of vector B, which is a non-circular buffer of length N + 1.

FIR filter output:

The Y buffer contains the output values yn, which is a circular buffer of length d.

FIR filter parameters:

Parameter P is the length of coefficient vector B, N+1, ranging from [2:127].

Parameter R is the gain applied to the accumulator output. The value of the accumulator multiplied by 2^R is output to the Y buffer, where the range of R is [0:7]. Parameter Q is not used.

The function ends when the FMAC_PARAMCFG.START bit is reset by software.

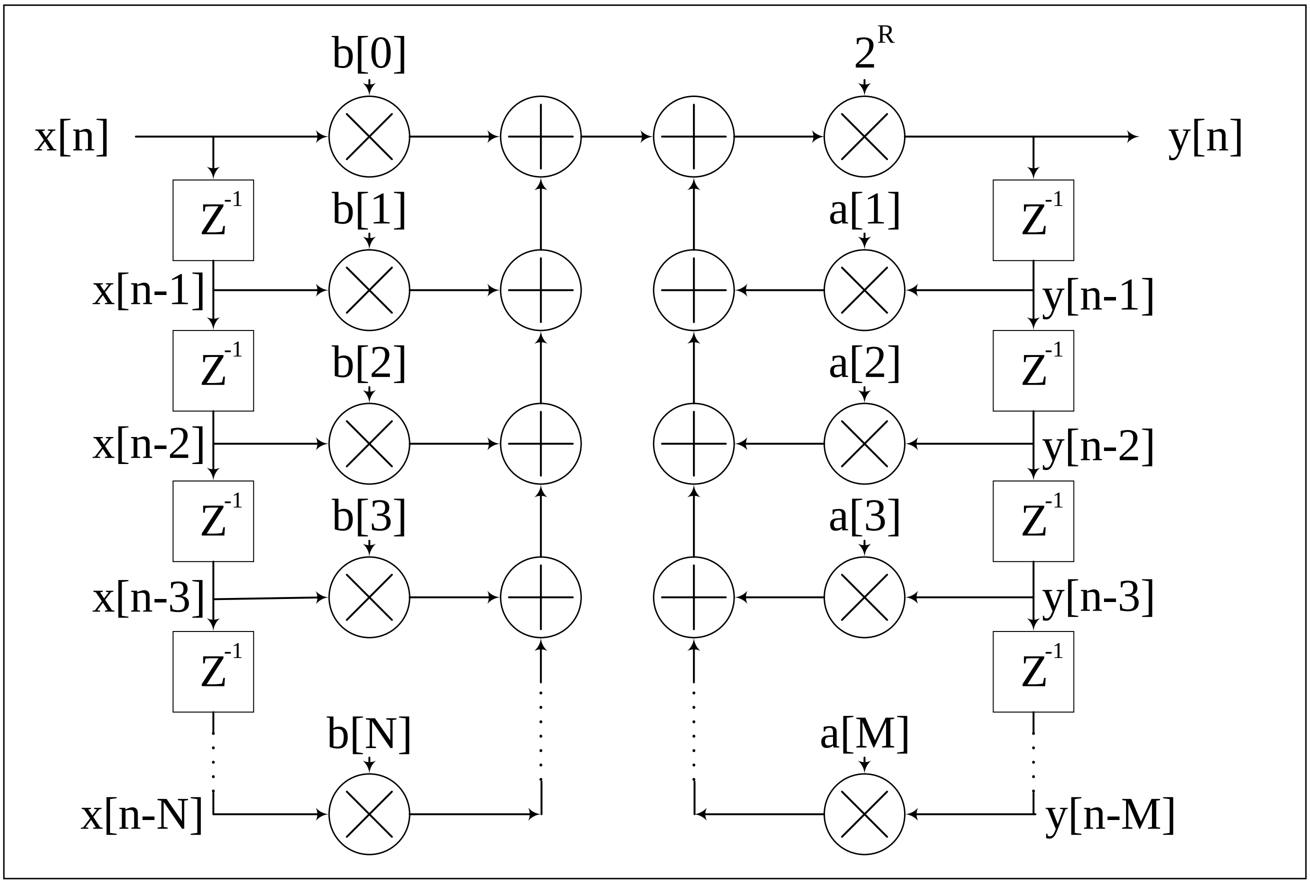
IIR filter: Y = B*X+A*Y

$$y[n] = 2^R \cdot (\sum_{j=0}^{N} b[j] \cdot x[n-j] + \sum_{j=1}^{M} a[j]y[n-j])$$

This function is used to implement an Infinite Impulse Response (IIR) filter, where the output vector Y is the convolution of the coefficient vector B (length N+1) with the infinite-length vector X, added to the convolution of the vector Y (previous output sequence) and the vector A (feedback filter coefficients). The elements of the output vector Y are calculated as yn = B·Xn + A·Yn-1, where Xn = [xn-N, ..., xn] and Yn-1 = [yn-M, ..., yn-1].

The IIR filter structure is shown in FigureFigure 21-8.

Figure 21-8 IIR Filter Structure Diagram



IIR filter input: The X1 buffer contains the elements of vector X, which is a circular buffer of length N + 1 + d. The X2 buffer contains the elements connected from coefficient vectors B and A (b0, b1, b2, ..., bN, aM, ..., a2, a1). This is a non-circular buffer of length M + N + 1.

IIR filter output: The Y buffer contains the output values yn. It is a circular buffer of length M + d.

IIR filter parameters: Parameter P is the length of coefficient vector B, N + 1, ranging from [2:64]. Parameter Q is the length of coefficient vector A, M, ranging from [1:63]. Parameter R is the gain applied to the accumulator output, where the value of the accumulator multiplied by 2^R is output to the Y buffer, and the range of R is [0:7].

When the FMAC_PARAMCFG.START bit is reset by software, the function ends.

21.3.7 Fixed-Point Data Format

FMAC operates in signed fixed-point format, with input and output values in q1.15 format. In the q1.15 format, numbers consist of a sign bit and 15 fractional bits (binary decimal places), with a numerical range from -1 (0x8000) to 1 - 2^-15 (0x7FFF). The accumulator has 26 bits, with 22 bits for fractional parts and 4 bits for integer/sign bits (q4.22), supporting an accumulator result range from -8 (0x2000000) to +7.99999976 (0x1FFFFFF).

At the output of the accumulator, programmable gain from 0dB to 42dB can be used, with a step size of 6dB. Note that if the value exceeds the numerical range, greater than +7.99999976 or less than -8, the accumulator content wraps around. The wraparound triggered is harmless as subsequent accumulation operations will cancel out the wraparound. If a wraparound occurs, the FMAC_STS.SATF bit is set, and if the FMAC_CTRL.SATINTEN bit is set, an interrupt is generated. This aids in debugging the filter.

By setting the FMAC_CTRL.LIMITEN bit, after applying programmable gain, the data at the output of the accumulator can be optionally saturated. If LIMITEN is set to 1, any values exceeding the q1.15 output numerical range will be set to 1 - 2^-15 or -1. If LIMITEN is set to 0, any unused accumulator bits after applying gain will be simply truncated.

21.3.8 FIR Filter

FMAC can be configured as an FIR filter of length N, where N is the number of taps or filter coefficients. The minimum buffer required for an FIR filter of length N is 2N + 1, consisting of N sample data, 1 output sample, and N filter coefficients. Since the buffer size is 256, the maximum value for N is 127. For maximum throughput, a small additional space (d1 and d2) needs to be allocated for the input and output sample buffers to ensure that the filter does not stall while waiting for new input samples or reading output samples. In this scenario, the buffer requirement is 2N + d1 + d2

The FIR filter buffer can be configured as follows: X1BUFSIZE equals N + d1, X2BUFSIZE equals N, YBUFSIZE equals d2. If the output buffer does not require additional space, YBUFSIZE can be set to 1.

The base addresses of the buffers can be arbitrarily assigned, but the X2 buffer must not overlap with other buffers, as this would overwrite the coefficients. A configuration example is as follows: X2BASE equals N, X1BASE equals 0, YBASE equals 2N+d1. If storage space is limited, the X1 buffer and Y buffer can overlap, but ensure that X1BASE equals 0, X2BASE equals N, YBASE equals N. In this case, the latest output sample replaces the oldest input sample, maintaining the buffers in sync.

Note: The X1 buffer configuration register's X1BUFWM field must be configured as less than or equal to log2(d1). Otherwise, before all N input parameters are fully written (e.g., writing N-2), the buffer will be marked as full, indicating that the buffer space is full and no more input samples will be requested. Similarly, the Y buffer configuration register's YBUFWM field must be less than or equal to log2(d2), otherwise the output buffer will be marked as empty.

The X2 buffer must preload the FIR filter coefficients, while the X1 buffer can optionally preload any number of sample data, up to a maximum of N. Based on the FIR filter structure, preloading the Y buffer does not make sense.

There are three ways to read and write data in the FMAC memory: polling, interrupt, and DMA.

Polling: No DMA or interrupt request is generated. Before writing to WDAT, software needs to ensure that the

X1BUFFF flag is 0 (indicating the buffer is not full), and before reading from RDAT, software needs to ensure that the YBUFEF flag is 0 (indicating the buffer is not empty).

Interrupt: Write interrupt request is generated when X1BUFFF flag is low; read interrupt request is generated when YBUFEF flag is low.

DMA: DMA write request is generated when X1BUFFF flag is low; DMA read request is generated when YBUFEF flag is low.

Reading and writing can use different methods. However, it is not recommended to use interrupts and DMA requests simultaneously in the same operation. If interrupts and DMA requests are enabled simultaneously, only DMA can perform the transfer.

To start the filter, write the following bit field values to the FMAC_PARAMCFG register: FUNC = 8 (FIR filter); P = N (number of filter coefficients); Q = "any value"; R = gain; START = 1.

If the preloaded values in the X1 buffer are less than N + d1 - 2X1BUFWM, the X1BUFFF flag remains 0. If the FMAC_CTRL.WINTEN bit is set, an immediate write interrupt request is generated, requiring the processor to write 2X1BUFWM additional sample values into the buffer. The interrupt request remains pending until the FMAC_STS.X1BUFFF flag is set.

In the interrupt service function, the X1BUFFF flag should be checked after writing every 2X1BUFWM data until the X1BUFFF flag is set.

Similarly, if the FMAC_CTRL.DMAWEN bit is set to 1, DMA write channel requests will continue to be generated until the X1BUFFF flag is set.

The filter will only calculate the first output sample when at least N samples (including any preloaded samples) are written to the X1 buffer.

When 2^{X1BUFWM} output samples have been written to the Y buffer, the FMAC_STS.YBUFEF flag becomes 0.

If the FMAC_CTRL.RINTEN bit is set, an immediate read interrupt request is generated at this point, requiring the processor to read 2^{YBUFWM} samples from the buffer. The interrupt request remains pending until the FMAC_STS.YBUFEF flag is set.

In the interrupt service function, the FMAC_STS.YBUFEF flag should be checked after reading every 2^{YBUFWM} data until the FMAC_STS.YBUFEF flag is set.

Similarly, if the FMAC_CTRL.DMAREN bit is set, DMA read channel requests will continue to be generated until the FMAC_STS.YBUFEF flag is set.

The filter continues to operate in this manner until the software resets the FMAC_PARAMCFG.START bit, stopping the filter operation.

21.3.9 IIR Filter

FMAC can be configured as an IIR filter with a length of N+1 and M, where N+1 is the number of feedforward taps or coefficients, and M is the number of feedback filter coefficients, with values ranging from 1 to N.

The minimum buffer required for an IIR filter is 2N + 2M, consisting of N feedforward filter coefficients, M feedback filter coefficients, N input sample data, and M output sample values. Since the buffer size is 256, if M = N-1, the maximum length of the IIR filter is N = 64.

For maximum throughput, a small additional space (d1 and d2) needs to be allocated for the input and output sample buffers, in which case the buffer requirement is 2N + d1 + d2.

The IIR filter buffer can be configured as follows: X1BUFSIZE equals N + d1, X2BUFSIZE equals N+M (number of coefficients), YBUFSIZE equals M + d2.

The base addresses of the buffers can be arbitrarily assigned, but the buffers must not overlap. A configuration example is as follows: X2BASE equals 0, X1BASE equals N+M, YBASE equals 2N+M+d1.

Note: The X1 buffer configuration register's X1BUFWM field must be configured as less than or equal to log2(d1). Otherwise, before all N input parameters are fully written (e.g., writing N-2), the buffer will be marked as full, indicating that the buffer space is full and no more input samples will be requested. Similarly, the Y buffer configuration register's YBUFEF field must be less than or equal to log2(d2), otherwise the output buffer will be marked as empty.

The X2 buffer must preload the IIR filter coefficients (N feedforward, followed by M feedback), while the X1 buffer can optionally preload any number of sample data, up to a maximum of N. The Y buffer can optionally preload any number of values, with a maximum not exceeding M.

To start the IIR filter, write the following bit field values to the FMAC_PARAMCFG register: FUNC = 9 (IIR filter); P = N (number of feedforward filter coefficients); Q = M (number of feedback filter coefficients); R = gain; START = 1.

If the preloaded values in the X1 buffer are less than N + d1 - $2^{X1BUFWM}$, the X1BUFFF flag remains 0. If the FMAC_CTRL.WINTEN bit is set, an immediate write interrupt request is generated, requiring the processor to write $2^{X1BUFWM}$ additional sample values into the buffer. The interrupt request remains pending until the FMAC_STS.X1BUFFF flag is set.

In the interrupt service function, the X1BUFFF flag should be checked after writing every $2^{X1BUFWM}$ data until the X1BUFFF flag is set.

Similarly, if the FMAC_CTRL.DMAWEN bit is set, DMA write channel requests will continue to be generated until the X1BUFFF flag is set.

The filter will only calculate the first output sample when at least N samples (including any preloaded samples) are written into the X1 buffer. The calculation of the first output value uses the first N samples in the X1 buffer and the first M samples in the Y buffer (whether preloaded or not) to compute the first output sample. The first output sample is written to YBASE + M in the Y buffer.

When 2^{YBUFWM} output samples have been written to the Y buffer, the FMAC_STS.YBUFEF flag becomes 0.

At this point, if the FMAC_CTRL.RINTEN bit is set, an immediate read interrupt request is generated, requiring the processor to read 2^{YBUFWM} samples from the buffer. The interrupt request remains pending until the FMAC_STS.YBUFEF flag is set.

In the interrupt service function, the FMAC_STS.YBUFEF flag should be checked after reading every 2^{YBUFWM} data until the YBUFEF flag is set.

Similarly, if the FMAC_CTRL.DMAREN bit is set, DMA read channel requests will continue to be generated until the YBUFEF flag is set. The filter continues to operate in this manner until the software resets the FMAC_PARAMCFG.START bit, stopping the filter operation.

21.4 FMAC Register

21.4.1 FMAC Register Overview

Table 21-1 Overview of FMAC Registers

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
000h	FMAC_X1BUFCFG	Reserved						X1BUFWM[1:0]		Reserved								X1BUFSIZE[7:0]								X1BASE[7:0]							
	Reset Value							0	0									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
004h	FMAC_X2BUFCFG	Reserved																X2BUFSIZE[7:0]								X2BASE[7:0]							
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
008h	FMAC_YBUFCFG	Reserved						YBUFWM[1:0]		Reserved								YBUFSIZE[7:0]								YBASE[7:0]							
	Reset Value							0	0									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
00Ch	FMAC _PARAMCFG	START	FUNC[6:0]							R[7:0]								Q7:0]								P[7:0]							
	Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
010h	FMAC _CTRL	Reserved															RESET	LIMITEN	Reserved					DMAWEN	DMAREN	Reserved			SATINTEN	UNINTEN	OVINTEN	WINTEN	RINTEN
	Reset Value																0	0						0	0				0	0	0	0	0
014h	FMAC _STS	Reserved																					SATF	UNF	OVF	Reserved						X1BUFFF	YBUFEF
	Reset Value																						0	0	0							0	0
018h	FMAC_WDAT	Reserved																WDAT[15:0]															
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
01Ch	FMAC_RDAT	Reserved																RDAT[15:0]															
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

21.4.2 FMAC X1 Buffer Configuration Register (FMAC_X1BUFCFG)

Offset address: 0x00

Reset value: 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved						X1BUFWM [1:0]		Reserved							

rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
X1BUFSIZE[7:0]								X1BASE[7:0]							
rw								rw							

Field offset	Register name	Description
31:26	Reserved	Reserved, the reset value must be maintained
25:24	X1BUFWM [1:0]	Buffer watermark size Defined as the threshold set in circular mode to trigger the X1 buffer full flag. The corresponding flag will be set if the number of free spaces in the buffer is less than $2^{X1BUFWM}$. 00: Threshold is 1, 01: Threshold is 2, 10: Threshold is 4, 11: Threshold is 8 Setting a threshold greater than 1 allows multiple data transfers to the buffer in a single interrupt. If DMA write request is enabled (FMAC_CTRL.DMAWEN = 1), the threshold should be set to 1.
23:16	Reserved	Reserved, the reset value must be maintained
15:8	X1BUFSIZE[7:0]	X1 buffer size. The X1 buffer size in 16-bit words, the minimum buffer size is the number of feedforward filter coefficients in the filter (+ watermark threshold - 1).
7:0	X1BASE[7:0]	X1 buffer base address.

21.4.3 FMAC X2 Buffer Configuration Register (FMAC_X2BUFCFG)

Offset address: 0x04

Reset value: 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

X2BUFSIZE[7:0]	X2BASE[7:0]
rw	rw

Field offset	Register name	Description
31:16	Reserved	Reserved, the reset value must be maintained
15:8	X2BUFSIZE[7:0]	X2 buffer size in 16-bit words. This bit field cannot be modified while the function is executing (START = 1).
7:0	X2BASE[7:0]	X2 buffer base address. When START=1, the X2 buffer base address can be modified, for example when changing coefficient values. Since changing coefficients during computation can affect the results, the filter should be in a stopped state when performing this operation.

21.4.4 FMAC Y Buffer Configuration Register (FMAC_YBUFCFG)

Offset address: 0x08

Reset value: 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved						YBUFWM [1:0]		Reserved							
						rw									

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
YBUFSIZE[7:0]								YBASE[7:0]							
rw								rw							

Field offset	Register name	Description
31:26	Reserved	Reserved, the reset value must be maintained
25:24	YBUFWM [1:0]	Buffer watermark size Defined as the threshold set in circular mode to trigger the Y buffer full flag. The corresponding flag will be set if the number of free spaces in the buffer is less than 2^{Y1BUFWM}.

Field offset	Register name	Description
		00: Threshold is 1, 01: Threshold is 2, 10: Threshold is 4, 11: Threshold is 8 Setting a threshold greater than 1 allows multiple data transfers to the buffer in a single interrupt. If DMA write request is enabled (FMAC_CTRL.DMAWEN = 1), the threshold should be set to 1.
23:16	Reserved	Reserved, the reset value must be maintained
15:8	YBUFSIZE[7:0]	Y buffer size.
7:0	YBASE[7:0]	Y buffer base address.

21.4.5 FMAC Parameter Configuration Register (FMAC _PARAMCFG)

Offset address: 0xC

Reset value: 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
START	FUNC[6:0]							R[7:0]							
rw	rw							rw							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Q[7:0]								P[7:0]							
rw								rw							

Field offset	Register name	Description
31	START	Enable execution. 0: Stop execution 1: Start execution Setting this bit will trigger the execution of the function selected in the FUNC bit field. Resetting this bit by software can stop any ongoing function execution. For initialization functions, this bit is reset by hardware.

Field offset	Register name	Description
30:24	FUNC[6:0]	Function. 0: Reserved 1: Load X1 buffer 2: Load X2 buffer 3: Load Y buffer 4 to 7: Reserved 8: FIR filter 9: IIR filter (direct form 1) 10 to 127: Reserved This bit field cannot be modified while the function is executing (START = 1).
23:16	R[7:0]	Input parameter R. The value of this parameter depends on the function. This bit field cannot be modified while the function is executing (START = 1).
15:8	Q[7:0]	Input parameter Q. The value of this parameter depends on the function. This bit field cannot be modified while the function is executing (START = 1).
7:0	P[7:0]	Input parameter P. The value of this parameter depends on the function. This bit field cannot be modified while the function is executing (START = 1).

21.4.6 FMAC Control Register (FMAC_CTRL)

Offset address: 0x10

Reset value: 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															RESET
															rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LIMITEN	Reserved					DMAWEN	DMAREN	Reserved			SATINTEN	UNINTEN	OVINTEN	WINTEN	RINTEN

rw rw rw rw rw rw rw rw

Field offset	Register name	Description
31:17	Reserved	Reserved, the reset value must be maintained
16	RESET	Reset FMAC. This will reset the write and read pointers, internal control logic, FMAC_STS register, and FMAC_PARAMCFG register (if activated), including the START bit. Other register settings are not affected. This bit is reset by hardware. 0: Disable reset 1: Enable reset
15	LIMITEN	Clipping enable. 0: Clipping disabled, values exceeding the accumulator range are truncated. 1: Clipping enabled, values exceeding the accumulator range are clipped to the maximum positive value or minimum negative value (+1 or -1)
14:10	Reserved	Reserved, the reset value must be maintained
9	DMAWEN	Enable DMA write channel request. 0: Disabled. No DMA requests generated. 1: Enabled. DMA write requests are generated when X1 buffer is not full. This bit can only be modified when START=0 in the FMAC_PARAM register.
8	DMAREN	Enable DMA read channel request. 0: Disabled. No DMA requests generated. 1: Enabled. DMA read requests are generated when Y buffer is not full. This bit can only be modified when START=0 in the FMAC_PARAM register.
7:5	Reserved	Reserved, the reset value must be maintained
4	SATINTEN	Saturation error interrupt enable. 0: Disabled. No interrupt generated when saturation is detected. 1: Enabled. Interrupt request generated if the SAT flag is set. This bit can be set and cleared by software. Reading returns the current status of this bit.
3	UNINTEN	Underflow error interrupt enable.

Field offset	Register name	Description
		0: Disabled. No interrupt generated when underflow is detected. 1: Enabled. Interrupt request generated if the UNFL flag is set. This bit can be set and cleared by software. Reading returns the current status of this bit.
2	OVINTEN	Overflow error interrupt enable. 0: Disabled. No interrupt generated when overflow is detected. 1: Enabled. Interrupt request generated if the OVFL flag is set. This bit can be set and cleared by software. Reading returns the current status of this bit.
1	WINTEN	Write interrupt enable. 0: Disabled. No write interrupt requests generated. 1: Enabled. Interrupt request generated when the X1 buffer FULL flag is not set. This bit can be set and cleared by software. Reading returns the current status of this bit.
0	RINTEN	Read interrupt enable. 0: Disabled. No read interrupt requests generated. 1: Enabled. Interrupt request generated when the Y buffer EMPTY flag is not set. This bit can be set and cleared by software. Reading returns the current status of this bit.

21.4.7 FMAC Status Register (FMAC _STS)

Offset address: 0x14

Reset value: 0x0000

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="16">Reserved</td></tr>
</table>

<table>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="5">Reserved</td><td>SATF</td><td>UNF</td><td>OVF</td><td colspan="6">Reserved</td><td>X1BUFFF</td><td>YBUFEF</td></tr>
<tr><td colspan="5"></td><td>r</td><td>r</td><td>r</td><td colspan="6"></td><td>r</td><td>r</td></tr>
</table>

Field offset	Register name	Description
31:11	Reserved	Reserved, the reset value must be maintained

Field offset	Register name	Description
10	SATF	Saturation error flag. Saturation occurs when the accumulation result exceeds the range of the accumulator. 0: No saturation detected 1: Saturation detected. If the SATINTEN bit is set, an interrupt will be generated. This flag will be cleared after device reset.
9	UNF	Underflow error flag. When reading data from FMAC_RDAT, underflow error occurs if there is no valid data in the Y buffer. 0: No underflow detected 1: Underflow detected. If the UNINTEN bit is set, an interrupt will be generated. This flag will be cleared after device reset.
8	OVF	Overflow error flag. Overflow occurs when writing to FMAC_WDAT and there is no available space in the X1 buffer. 0: No overflow detected 1: Overflow detected. If the OVINTEN bit is set, an interrupt will be generated. This flag will be cleared after device reset.
7:2	Reserved	Reserved, the reset value must be maintained
1	X1BUFFF	X1 buffer full flag. If the available space count is less than the X1BUFWM threshold, the buffer will be marked as full. The available space count is the difference between the write pointer and the latest sample being used. 0: X1 buffer not full. If the WINTEN bit is set, interrupt requests will be generated until the flag is set. If DMAWEN is set, DMA write channel requests will be generated until this flag is set. 1: X1 buffer full. This flag is set and cleared by hardware or reset. *Note: After the last available space in the X1 buffer is filled, there will be a delay of 3 clock cycles before the X1BUFFF flag is cleared. To avoid any risk of overflow, it is recommended to insert a software delay after writing to the X1 buffer before reading FMAC_STS. Additionally, X1BUFWM=2 can be used to configure the threshold.*
0	YBUFEF	Y buffer empty flag.

Field offset	Register name	Description
		If the number of unread data is less than the YBUFWM threshold, the buffer will be marked as empty. The number of unread data is the difference between the read pointer and the current output target address. 0: Y buffer not empty. If the RINTEN bit is set, interrupt requests will be generated until the flag is set. If the DMAREN bit is set, DMA read channel requests will be generated until this flag is set." 1: Y buffer is empty. This flag is set and cleared by hardware or reset. *Note: After reading the last sample from the Y buffer, there will be a delay of 3 clock cycles before the YBUFEF flag is set. To avoid any risk of underflow, it is recommended to insert a software delay after reading data from the Y buffer before reading FMAC_STS. Additionally, YBUFWM=2 can be used to configure the threshold.*

21.4.8 FMAC Write Data Register (FMAC_WDAT)

Offset address: 0x18

Reset value: 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WDAT[15:0]															
rw															

Field offset	Register name	Description
31:16	Reserved	Reserved, the reset value must be maintained
15:0	WDAT[15:0]	Write data. When accessing this register for writing, the written data will be transferred to the address offset indicated by the write pointer. After each write operation, the pointer address will automatically increment.

21.4.9 FMAC Read Data Register (FMAC_RDAT)

Offset address: 0x1C

Reset value: 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RDAT[15:0]															
rw															

Field offset	Register name	Description
31:16	Reserved	Reserved, the reset value must be maintained
15:0	RDAT[15:0]	Read data. When accessing this register for reading, the data read is the content in the Y output buffer. After each read operation, the pointer address will automatically increment.

22 CORDIC Processors (CORDIC)

22.1 Introduction

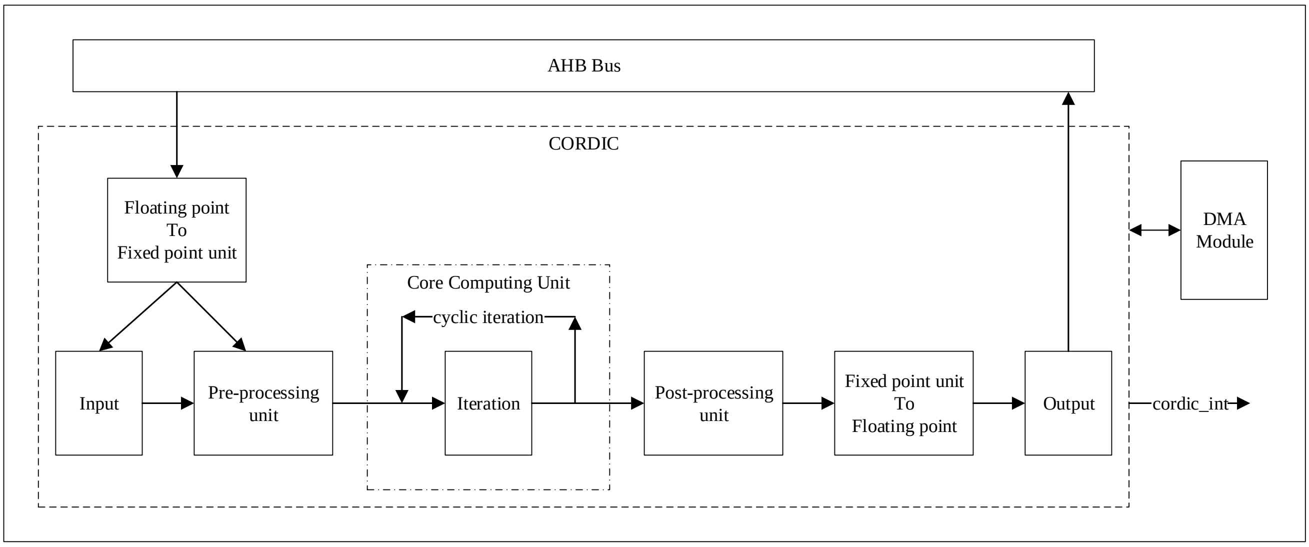
CORDIC hardware computing units can accelerate mathematical functions (primarily trigonometric functions) in hardware. They are commonly used for accelerating mathematical function calculations in applications such as motor control, measurement, signal processing, and many other applications.

22.2 Main Features

- Supports rotation and vector calculation modes.
- Supports circular and hyperbolic coordinate systems.
- Once the calculation starts, any operation to read the result register will insert the bus into a waiting state until the calculation is completed, allowing the calculation result to be read out when completed without the need for polling or interrupts.
- Calculate 10 functions: sin, cosine, sinh, cosh, atan, atan2, atanh, modulus, square root, natural logarithm.
- Supports fixed-point and floating-point input/output modes.
- Supports interrupt, polling, and DMA request read/write modes.
- Programmable iteration precision.

22.3 Block Diagram

Figure 22-1 CORDIC Block Diagram



The CORDIC hardware computing unit structural diagram above includes its main core computing unit, data preprocessing unit, data post-processing unit, floating-point to fixed-point unit, and fixed-point to floating-point unit.

Floating-point to fixed-point unit: The floating-point to fixed-point unit converts floating-point or half-precision data into corresponding q1.31 or q1.15 formats and stores them in the CORDIC_WDAT register. It is configured by the

FLOATIN bit of the CORDIC_CTRLSTS register. The default is fixed-point input, and if the FLOATIN bit is configured as 1, it is set to floating-point input.

Data preprocessing unit: It processes the data in the CORDIC_WDAT register before sending it to the core computing unit of CORDIC.

Core computing unit: The processed input data is iteratively calculated to obtain the computed value, with the computational precision determined by the number of iterations and programmable. It supports circular and hyperbolic systems, with each system supporting rotation and vector modes.

Data post-processing unit: It processes the computed values from the core computing unit of CORDIC to obtain data in q1.31 or q1.15 format.

Fixed-point to floating-point unit: It converts q1.31 or q1.15 format data to the corresponding single-precision floating-point data or half-precision floating-point data.

22.4 Function Description

22.4.1 CORDIC Function

The functions supported by Cordic are shown in the table below.

Table 22-1 CORDIC Function Overview

Function	Primary Parameter	Secondary Parameter	Primary Result	Secondary Result
Consine	Angle θ	Modulus m	m * cos θ	m * sin θ
Sine	Angle θ	Modulus m	m * sin θ	m * cos θ
Phase	x	y	atan2(y,x)	$\sqrt{x^2 + y^2}$
Modulus	x	y	$\sqrt{x^2 + y^2}$	atan2(y,x)
Arctangent	x	none	$\tan^{-1}x$	none
Hyperbolic cosine	x	none	cosh x	sinh x
Hyperbolic sine	x	none	sinh x	cosh x
Hyperbolic actangent	x	none	$\tanh^{-1}x$	none
Natural logarithm	x	none	ln x	none
Square root	x	none	$\sqrt{x}$	none

Through the above 10 functions, more functions can be indirectly obtained.

For example, ex = sinh(x) + cosh(x).

Cosine Function

Table 22-2 Cosine Parameters

Parameter	Range	Description
First input parameter (θ)	[-1, 1)	It is the angle value θ, measured in radians (rad), with the rangeθ∈[-π, π). When using it, first divide the target angle θ by π to convert it to the range [-1, 1),

		and then write it in signed fixed-point (q1.31 and q1.15 formats) or floating-point data (single-precision and half-precision) to the CORDIC_WDATA register.
Second input parameter (m)	[0, 1]	It represents the magnitude of the function. When m ≥ 1, the software scales m down to the [0, 1] range, and then writes it in signed fixed-point (q1.31 and q1.15 formats) or floating-point data (single-precision and half-precision) to the CORDIC_WDATA register.
First output result m * cos(θ)	[-1, 1)	If m has been scaled down by software before, the output data needs to be correspondingly scaled up to obtain the actual result.
Second output result m * sin(θ)	[-1, 1)	If m has been scaled down by software before, the output data needs to be correspondingly scaled up to obtain the actual result.
Scaling factor SCALE[10:8]	Unavailable	maintain reset value 3'b0.

Note:

1. When the magnitude m > 1, the software can choose the scaling ratio.

Sine Function

Table 22-3 Sine Parameters

Parameter	Range	Description
First input parameter (θ)	[-1, 1)	It is the angle value θ, measured in radians (rad), with the rangeθ∈[-π, π). When using it, first divide the target angle θ by π to convert it to the range [-1, 1), and then write it in signed fixed-point (q1.31 and q1.15 formats) or floating-point data (single-precision and half-precision) to the CORDIC_WDATA register.
Second input parameter (m)	[0, 1]	It represents the magnitude of the function. When m ≥ 1, the software scales m down to the [0, 1] range, and then writes it in signed fixed-point (q1.31 and q1.15 formats) or floating-point data (single-precision and half-precision) to the CORDIC_WDATA register.
First output result m * cos(θ)	[-1, 1)	If m has been scaled down by software before, the output data needs to be correspondingly scaled up to obtain the actual result.
Second output result m * sin(θ)	[-1, 1)	If m has been scaled down by software before, the output data needs to be correspondingly scaled up to obtain the actual result.
Scaling factor SCALE[10:8]	Unavailable	maintain reset value 3'b0.

Note:

1. When the magnitude m > 1, the software can choose the scaling ratio.

Phase(atan2(y, x)) Function

Table 22-4 Phase Parameters

Parameter	Range	Description
First input parameter (θ)	[-1, 1)	x represents the abscissa value in the Cartesian coordinate system. If x ≥ 1 or x <-1,

Parameter	Range	Description
		software scaling to the q1.31 range is required.
Second input parameter (m)	[-1, 1)	y represents the ordinate value in the Cartesian coordinate system. If y ≥ 1 or y <-1, software scaling to the q1.31 range is required.
First output result m * cos(θ)	[-1, 1)	The angle corresponding to the coordinate position (x, y), [-1, 1) corresponds to [-π, π). Multiplying this output data by π gives the actual angle value. Note that, due to the nature of phase angles, values close to π may sometimes be calculated as -π.
Second output result m * sin(θ)	[0, 1)	m=$\sqrt{x^2+y^2}$. If x and y have been scaled before, the magnitude needs to be proportionally enlarged.
Scaling factor SCALE[10:8]	Unavailable	maintain reset value 3'b0.

Note:

1. *If either x or y exceeds the range [-1, 1), both x and y need to be proportionally scaled together, not just one. This ensures that the angle corresponding to the coordinates remains unchanged before and after scaling.*

2. *when $\sqrt{x^2+y^2} \geq 1$, the magnitude m can only be saturated to the maximum value of the fixed-point format (1-2^{-15} or 1-2^{-31}). Before proportionally scaling x and y, consider the magnitude of the scaling factor to avoid saturation of the magnitude as much as possible.*

Modulus($\sqrt{x^2+y^2}$) Function

Table 22-5 Modulus Parameters

Parameter	Range	Description
First input parameter (θ)	[-1, 1)	x represents the abscissa value in the Cartesian coordinate system. If the desired x value is not within the range, it needs to be scaled to the range [-1, 1).
Second input parameter (m)	[-1, 1)	y represents the abscissa value in the Cartesian coordinate system. If the desired y value is not within the range, it needs to be scaled to the range [-1, 1).
First output result m * cos(θ)	[0, 1)	m=$\sqrt{x^2+y^2}$. If x and y have been scaled before, the magnitude needs to be proportionally enlarged.
Second output result m * sin(θ)	[-1, 1)	The angle corresponding to the coordinate position (x, y) corresponds to [-1, 1) corresponding to [-π, π). Multiplying this output data by π gives the actual angle value. Note that, due to the nature of phase angles, values close to π may sometimes be calculated as -π.
Scaling factor SCALE[10:8]	Unavailable	maintain reset value 3'b0.

Note:

1. *If either x or y exceeds the range [-1, 1), both x and y need to be proportionally scaled together, not just one. This ensures that the angle corresponding to the coordinates remains unchanged before and after scaling.*

2. *when $\sqrt{x^2+y^2} \geq 1$, the magnitude m can only be saturated to the maximum value of the fixed-point format (1-2^{-15} or 1-2^{-31}). Before proportionally scaling x and y, consider the magnitude of the scaling factor to avoid*

saturation of the magnitude as much as possible.

Arctangent(tan^{-1}(x)) Function

This mode is used to calculate the tan^{-1}(x) function. It has one input and one output.

Table 22-6 Arctangent Parameters

Parameter	Range	Description
Input parameter (x)	[-1, 1)	If |x| > 1, the software must scale it down by a factor of 2^(-n) to ensure x is within the range [-1, 1). The scaling factor n is written in the SCALE[10:8] bit field.
Output parameter (θ)	[-1, 1)	[-1, 1) corresponds to [-π, π), so the output result needs to be multiplied by π and 2^n to obtain the actual angle value.
Scaling factor SCALE[10:8]	[0, 7]	Configuration of the SCALE[10:8] bit field.

Hyperbolic cosine (cosh(x)) Function

This mode is used to calculate the cosh(x) function. It has one input and two outputs.

Table 22-7 Hyperbolic cosine Parameters

Parameter	Range	Description
Input parameter (x)	[-0.559, 0.559]	Only x values within the range of [-1.118, +1.118] are supported. The software scales x down by a factor of 2^(-n). For n = 1, it can only be 1, and is written into the SCALE[10:8] bit threshold.
The first output result cosh (x) /2	[-0.5, 0.846]	Hyperbolic cosine value cosh x. The first output result must be multiplied by 2 to obtain the correct result.
The second output result sinh (x) /2	[-0.683, 0.683]	Hyperbolic sine value sinh x. The second output result must also be multiplied by 2 to obtain the correct result.
Scaling factor SCALE[10:8]	1	Can only be configured as 3'b001.

Hyperbolic sine (sinh(x)) Function

This mode is used to calculate the sinh(x) function. It has one input and two outputs.

Table 22-8 Hyperbolic sine Parameters

Parameter	Range	Description
Input parameter (x)	[-0.559, 0.559]	Only x values within the range of [-1.118, +1.118] are supported. The software scales x down by a factor of 2^(-n). For n = 1, it can only be 1, and is written into the SCALE[10:8] bit threshold.
The first output result sinh (x) /2	[-0.683, 0.683]	Hyperbolic cosine value cosh x. The first output result must be multiplied by 2 to obtain the correct result.
The second output result cosh (x) /2	[-0.5, 0.846]	Hyperbolic sine value sinh x. The second output result must also be multiplied by 2 to obtain the correct result.
Scaling factor	1	Can only be configured as 3'b001.

Parameter	Range	Description
SCALE[10:8]		

Hyperbolic arctangent (tanh^{-1}(x)) Function

This mode is used to calculate the tanh^{-1}(x) function. It has one input and one outputs.

Table 22-9 Hyperbolic arctangent Parameters

Parameter	Range	Description
Input parameter (x)	[-0.403, 0.403]	Only x values within the range of [-0.806, 0.806] are supported. The software scales x down by a factor of 2^(-n). For n = 1, it can only be 1, and is written into the SCALE[10:8] bit threshold.
The first output result sinh tanh^{-1}(x)/2	[-0.559, 0.559]	Multiplying the output data by 2 yields the inverse hyperbolic tangent tanh^{-1}(x).
Scaling factor SCALE[10:8]	1	Can only be configured as 3’b001.

Natural logarithm (ln(x)) Function

This mode is used to calculate the ln(x) function. It has one input and one outputs.

Table 22-10 Natural logarithm Parameters

Parameter	Range	Description
Input parameter ($\frac{x}{2^n}$)	[0.054, 0.875]	For x∈[0.107, 9.35], if not within the range, the software needs to scale x down by a factor of 2^(-n) to [0.054, 0.875]. Then write n into the SCALE[10:8] bit threshold.
The first output result $\frac{\ln(x)}{2^{n+1}}$	[-0.279, 0.137]	Multiplying the output data by 2(n+1) yields the natural logarithm ln(x).
Scaling factor SCALE[10:8]	[1, 4]	Configure SCALE[10:8] as n.

To ensure calculation accuracy, it is recommended to use the scaling factors in the table below for different inputs:

n	x range	Input parameter range
1	$0.107 \leq x < 1$	$0.0535 \leq \frac{x}{2^n} < 0.5$
2	$1 \leq x < 3$	$0.25 \leq \frac{x}{2^n} < 0.75$
3	$3 \leq x < 7$	$0.375 \leq \frac{x}{2^n} < 0.875$
4	$7 \leq x \leq 9.35$	$0.4375 \leq \frac{x}{2^n} < 0.584$

Square root ($\sqrt{x}$) Function

This function calculates the square root of the input parameter x, with one input and one output.

Table 22-11 Square root Parameters

Parameter	Range	Description
Input parameter ($\frac{x}{2^n}$)	[0.027, 0.875]	For x∈[0.027, 2.34], if not within the range, the software needs to scale x down by a factor of 2^(-n) to [0.054, 0.875]. Then write n into the SCALE[10:8] bit threshold.
The first output result $\frac{\ln(x)}{2^{n+1}}$	[0.04, 1]	Multiplying the output data by 2n yields √x.
Scaling factor SCALE[10:8]	[0, 2]	Configure SCALE[10:8] as n.

To ensure calculation accuracy, it is recommended to use the scaling factors in the table below for different inputs.

n	x range	Input parameter range
0	$0.027 \leq x < 0.75$	$0.027 \leq (\frac{x}{2}) < 0.75$
1	$0.75 \leq x < 1.75$	$0.375 \leq (\frac{x}{2}) < 0.875$
2	$1.75 \leq x \leq 2.341$	$0.4375 \leq (\frac{x}{2}) \leq 0.585$

22.4.2 Data Format

The input and output data of the CORDIC module support signed integer fixed-point formats (q1.31 and q1.15) and floating-point data (single-precision and half-precision). Floating-point data needs to be converted according to the IEEE 754 standard before being written to the data register.

The FLOATIN bit of the CORDIC_CTRLSTS register configures the data input format, with the default being fixed-point format input. Setting the FLOATIN bit to 1 enables floating-point input.

The FLOATOUT bit of the CORDIC_CTRLSTS register configures the data output format, with the default being fixed-point format output. Setting the FLOATOUT bit to 1 enables floating-point output.

The INSIZE bit of the CORDIC_CTRLSTS register is used to configure the width of the input data, with the default being 32 bits wide.

If configured as 32-bit, the input data should be in q1.31 fixed-point format or single-precision floating point, meaning that the input data written to the CORDIC_WDAT register should be in q1.31 fixed-point format (FLOATIN = 0) or single-precision floating point (FLOATIN = 1) format.

If configured as 16-bit, the input data should be in q1.15 fixed-point format or half-precision floating-point format, that is, the input data written to the CORDIC_WDAT register is in q1.15 fixed-point format (FLOATIN = 0) or half-precision floating-point format (FLOATIN = 1).

The NUMWRITE bit of the CORDIC_CTRLSTS register is used to configure the number of input data, with the default setting being to input a single data parameter (NUMWRITE = 0).

If configured with NUMWRITE = 0, and if the data input width is set to 32-bit (INSIZE = 0), one can input a q1.31 fixed-point format or single-precision floating-point data; if the data input width is set to 16-bit (INSIZE = 1), two q1.15 fixed-point format or half-precision floating-point data can be input, with the first parameter placed in the lower 16 bits and the second parameter placed in the higher 16 bits.

If configured with NUMWRITE = 1, only two q1.31 fixed-point format or single-precision floating-point data can be input. The output data format remains the same as above.

22.4.3 Scaling Factor

The preceding functions specify a scaling factor SCALE. This allows the extended function input range to cover all values supported by CORDIC without causing input, output, or internal register saturation. If a scaling factor is required, it must be calculated in software and programmed into the SCALE field of the CORDIC_CTRLSTS register. When scaling down the value in the CORDIC_WDAT register and programming the scaling factor SCALE, the input parameters must also be correspondingly scaled down. Finally, the scaling down must be undone for the result read from the CORDIC_RDAT register, i.e., multiplied by 2^n to obtain the correct value.

Note: The scaling factor may result in loss of precision due to truncation of the scaling value.

22.4.4 Precision

Each clock cycle, CORDIC can perform four iterations. TableTable 22-12 lists the maximum error after every four iterations for each function and the number of clock cycles required to achieve that precision. Based on this table, the required number of cycles can be determined and programmed into the precision field of the CORDIC_CTRLSTS register. Once the programmed number of iterations is completed, the CORDIC processor stops immediately, and the results can be read instantly.

Table 22-12 CORDIC Precision

Iteration Cycle.	**Maximum Residual Error (2^{-n})**			
	Normal	**16-bit fixed-point output**	**16-bit floating-point output**	**16-bit floating-point input**
COS，SIN				
1	3	3	2	2
2	7	6	6	6
3	10	10	9	10
4	14	13	10	12
5	18	14	10	12
6	18	14	10	12
PHASE，MODULUS				
1	3	3	3	3
2	7	7	7	7

Iteration Cycle.	Maximum Residual Error (2^{-n})			
	Normal	16-bit fixed-point output	16-bit floating-point output	16-bit floating-point input
3	12	12	9	12
4	13	13	9	13
5	13	13	9	13
6	13	13	9	13
ARCTAN				
1	3	3	3	3
2	7	7	7	7
3	12	12	10	12
4	15	14	10	16
5	18	14	10	16
6	19	14	10	16
HB_COS，HB_SIN，HB_ARCT，LN				
1	3	3	3	3
2	7	7	7	7
3	11	11	9	11
4	14	13	10	14
5	17	14	10	14
6	18	14	10	14
SQRT				
1	7	7	7	7
2	14	13	10	14
3	18	13	10	14
4	18	13	10	14
5	18	13	10	14
6	18	13	10	14

Note:

1. *All other cases (32-bit fixed-point input, 32-bit fixed-point output, 32-bit floating-point input, 32-bit floating-point output, 16-bit fixed-point input) are categorized under normal mode. In TableTable* 22-12*, if there are multiple applicable modes, the maximum error is considered.*
2. *In PHASE and MODULUS modes, the precision of the resulting angle is limited by the precision of the input y and x ratios. x and y should be scaled synchronously. To ensure a precision of* 10^{-4}*, one of the input data x or y should be greater than 0.005 (0.125 for 16-bit input). The precision data in TableTable* 22-12 *is calculated under the condition that one of x or y is greater than 0.005 (0.125). For higher precision, optimal precision can be achieved when one of x or y is close to 1. When one of x or y is greater than 0.1, the number of iteration cycles*

is 5 or 6 (in 32-bit mode), and precision can reach 10^{-5}. Table Table 22-13 *lists the precision of PHASE and MODULUS when the x and y ratios are larger.*

Table 22-13 Supplementary Table for PHASE and MODULUS Precision

Iteration Cycle.	Maximum Residual Error (2^{-n})			
	32-bit input		16-bit input	
	max(x,y) > 0.005 (able to achieve a precision of 10^{-4})	max(x,y) > 0.05 （optimal precision）	max(x,y) > 0.125 (able to achieve a precision of 10^{-4})	max(x,y) > 0.5 （optimal precision）
1	3	3	3	3
2	7	7	7	7
3	12	12	12	12
4	13	15	13	14
5	13	18	13	14
6	13	18	13	14

22.4.5 Operation Mode

CORDIC operation modes: Zero-Overhead Mode, Polling Mode, Interrupt Mode, and DMA Mode.

22.4.5.1 Zero-Overhead Mode

The fastest way to use the coprocessor is to pre-program the CORDIC_CTRLSTS register, which includes the desired function (FUNC), the required number of clock cycles (PRECISION), the number of bits for input and output values (INSIZE, OUTSIZE), the number of input parameters (NUMWRITE) and results (NUMREAD), and the scaling factor (SCALE) if applicable.

Subsequently, triggering the computation is done by writing the input parameters into the CORDIC_WDAT register. Once the correct number of input parameters is written (and any ongoing computation has completed), a new computation is initiated using these input parameters and the current CORDIC_CTRLSTS settings. If there are no changes, there is no need to reprogram the CORDIC_CTRLSTS register.

If dual 32-bit input parameters are required (INSIZE = 0, NUMWRITE = 1), the first input parameter must be written first, followed by the second input parameter. If the second parameter remains constant across a series of computations, it can be avoided to write it a second time by reprogramming the parameter count to one (INSIZE = 0) after the first calculation has started.

If using two 16-bit parameters (INSIZE = 1), they must be packed into a 32-bit word, with the primary parameter in the least significant half-word and the secondary parameter in the most significant half-word. The packed 32-bit word is then written into the CORDIC_WDAT register. In this scenario (NUMWRITE = 0), only one write operation is required.

For functions that only require one input parameter, it is recommended to set NUMWRITE = 0. If NUMWRITE = 1, a second write operation to CORDIC_WDAT must be performed to trigger the computation. In this case, the data for the second input is not used.

Once the computation begins, any attempt to read the CORDIC_RDAT register will result in a bus wait state until the computation is completed, and then the result will be returned. Therefore, software can write the input and immediately read the result without needing to poll to see if it is valid. Alternatively, the processor can wait for the appropriate number of clock cycles before reading the result. If needed, this time can be used to program the CORDIC_CTRLSTS register for the next computation and prepare the next input data. The CORDIC_CTRLSTS register can be reprogrammed during computation without affecting the results of the ongoing computation. Similarly, once the previous parameters start the computation, the CORDIC_WDAT register can be updated with the next parameters. The next parameters and settings remain pending until the previous computation is completed.

After the computation is completed, the result can be read from the CORDIC_RDAT register. If two 32-bit results are needed (OUTSIZE = 1, NUMWRITE = 0), the first result is read first, followed by the second result. If only one 32-bit result is needed (NUMWRITE = 0, OUTSIZE = 0), the first result is output during the first read.

If a 16-bit result is needed (OUTSIZE = 1), a single read of CORDIC_RDAT will pack both results into a single 32-bit word. The first result will be in the lower half-word, and the second result will be in the upper half-word. In this case, it is recommended to program NUMWRITE = 0. If NUMWRITE = 1, a second read of CORDIC_RDAT must be performed to release CORDIC for the next operation. The data from the second read must be discarded.

The next computation starts when the expected number of results has been read, provided that the expected number of parameters has been written. This means that at any given time, there may be one ongoing computation or waiting for result, and one pending operation. Any further access to CORDIC_WDAT while the operation is pending will cancel the pending operation and overwrite the data.

22.4.5.2 Polling Mode

When new results are available in the CORDIC_RDAT register, the RRF flag is set in the CORDIC_CTRLSTS register. This flag can be queried by reading the register, and reading the CORDIC_RDAT register once or twice (depending on the NUMWRITE field of the CORDIC_CTRLSTS register) will reset the RRF flag.

Polling the RRF flag takes slightly longer than directly reading the CORDIC_RDAT register because the results are not read immediately when available. However, the program and bus interface do not halt when reading the CORDIC_CTRLSTS register, so this mode might be a good choice if stopping the program is unacceptable (e.g., low-latency interrupt handling is required).

22.4.5.3 Interrupt Mode

By setting the interrupt enable (INTEN) bit in the CORDIC_CTRLSTS register, an interrupt is generated whenever the RRF flag is set. The interrupt is cleared when the RRF flag is reset. This mode allows reading the computation results under an interrupt service routine, giving it a priority over other tasks. However, due to the delay in interrupt handling, it is slower than directly reading the results or polling the flag.

22.4.5.4 DMA Mode

If the DMA write enable (DMAWEN) bit is set in the CORDIC_CTRLSTS register and there are no pending operations, a DMA write channel request is issued. The DMA controller can transfer the first input parameter from memory to the CORDIC_WDAT register. If NUMWRITE = 1 in the CORDIC_CTRLSTS register, a second DMA write channel request is issued to transfer the second input parameter to the CORDIC_WDAT register. Once all input parameters have been written and any ongoing computation has been completed (verified by reading the results), a new computation will be initiated, triggering another DMA write channel request.

If the DMA read enable (DMAREN) bit is set in the CORDIC_CTRLSTS register, the RRF flag being valid will generate a DMA read channel request. Subsequently, the DMA controller can transfer the first result from the CORDIC_RDAT register to memory. If NUMWRITE = 1 in the CORDIC_CTRLSTS register, a second DMA request is generated to read the second result. Once all results have been read, the RRF flag becomes invalid.

The DMA read and write channels can be enabled separately. If both channels are enabled, CORDIC can autonomously perform repetitive computations on the data buffer without processor intervention, allowing the program to execute other tasks. The DMA controller operates in memory-to-peripheral mode for the write channel and in peripheral-to-memory mode for the read channel. It is important to note that the timing is initiated by setting the DMAWEN flag in the program. Subsequently, the generation speed of DMA read and write requests can match the speed at which CORDIC processes data.

In some cases, input data may be stored in memory, and the output is periodically transferred to another peripheral device, such as a digital-to-analog converter. In this scenario, a DMA request is generated by the target peripheral every time new data is needed. The DMA controller can directly fetch the next sample from the CORDIC_RDAT register (in this case, the DMA controller operates in memory-to-peripheral mode, even if the source is a peripheral register). The behavior of reading results allows CORDIC to start a new computation, thereby generating a DMA write channel request for the DMA controller to transfer the next input value to the CORDIC_WDAT register. At this point, the DMA write channel is enabled (DMAWEN = 1), but the read channel is not enabled.

Similarly, data from another peripheral device (such as an ADC) can be directly transferred to the CORDIC_WDAT register in a similar manner (in peripheral-to-memory mode). The DMA write channel cannot be enabled in this case as well. If DMAREN = 1, CORDIC processes the input data and generates a DMA read request upon completion. The DMA controller then transfers the results from the CORDIC_RDAT register to memory (peripheral-to-memory mode).

Note:

1. *DMA requests are not generated to program the CORDIC_CTRLSTS register. Therefore, DMA mode is only useful when repeatedly executing the same function with the same settings. It is also important to note that the scaling factor cannot be changed during a series of DMA transfers.*
2. *For each DMA request, a response signal must be issued before the DMA can access the CORDIC_WDAT or CORDIC_RDAT registers. If unrelated accesses to the relevant registers occur before the DMA issues the response signal, it may block the DMA channel. Therefore, when the DMA read channel is enabled, CPU accesses to the CORDIC_RDAT register must be avoided. Similarly, when the DMA write channel is enabled, the program must avoid accessing the CORDIC_WDAT register.*

22.5 CORDIC Register

22.5.1 CORDIC Register Overview

Table 22-14 Cordic Register Overview

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0000h	CORDIC_CTRLSTS	RRF	INOVF	Reserved		INOVINTEN	CODINLIMIT	PHASELIMIT	FLOATOUT	FLOATIN	INSIZE	OUTSIZE	NUMWRITE	NUMREAD	DMAWEN	DMAREN	INTEN	Reserved					SCALE[2:0]			PRECISION[3:0]				FUNC[3:0]			
	Reset Value	0	0			0	0	0	0	0	0	0	0	0	0	0	0						0	0	0	0	1	0	1	0	0	0	0
0004h	CORDIC_WDAT	WDAT[31:0]																															
	Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0008h	CORDIC_RDAT	RDAT[31:0]																															
	Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

22.5.2 CORDIC Control Status Register (CORDIC_CTRLSTS).

Offset address: 0x00

Reset value: 0x0050

Each time data is written to the control status register, any stored but unprocessed data is discarded to prevent previous data from being computed using the current control configuration.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RRF	INOVF	Reserved		INOVINTEN	CODINLIMIT	PHASELIMIT	FLOATOUT	FLOATIN	INSIZE	OUTSIZE	NUMWRITE	NUMREAD	DMAWEN	DMAREN	INTEN
r	rw			rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved					SCALE			PRECISION				FUNC			
					rw			rw				rw			

Field offset	Register name	Description
31	RRF	Result Ready Flag. 0: No new data in the CORDIC_RDAT register. 1: New data in the CORDIC_RDAT register. This bit is set by hardware when the CORDIC operation is completed. It is reset by hardware after reading the CORDIC_RDAT register (NUMREAD + 1) times. When this bit is set, if the INTEN bit is also set, the CORDIC generates a read interrupt. If the DMAREN bit is also set, a DMA read channel request is generated. When this bit is set, no new computation is initiated
30	INOVF	Input Overflow Flag. 0: No input overflow event has occurred or input overflow interrupt control is not enabled. 1: An input overflow event has occurred. *Note: Writing a 1 to this bit clears it to 0, writing a 0 has no effect.*
29:28	Reserved	Reserved, must be kept at the reset value.
27	INOVINTEN	Input Overflow Interrupt Control Bit. 0: Disabled 1: Enabled. Setting INOVF to 1 when an input overflow occurs.
26	CODINLIMIT	Coordinate Output Limit Control. 0: No restriction on coordinate output result 1: Limit coordinate output result to prevent overflow In some cases, outputs close to 1 may overflow to -1 or vice versa. If CODINLIMIT = 1, when an overflow occurs, the positive output is limited to 1, and the negative output is limited to -1.
25	PHASELIMIT	Phase Output Limit Control. 0: No restriction on phase output 1: Limit phase output If PHASELIMIT = 1, when an overflow occurs, the positive phase is limited to π, and the

Field offset	Register name	Description
		negative phase is limited to -π.
24	FLOATOUT	Floating-point Fixed-point Output Control. 0: Fixed-point output 1: Floating-point output
23	FLOATIN	Floating-point Fixed-point Input Control. 0: Fixed-point input 1: Floating-point input
22	INSIZE	Input Data Width. 0: 32-bit 1: 16-bit If configured as 32-bit, the data written to the CORDIC_WDAT register is in q1.31 fixed-point format or single-precision floating-point. If configured as 16-bit, the data written to the CORDIC_WDAT register is in q1.15 fixed-point format or half-precision floating-point.
21	OUTSIZE	Output Data Width. 0: 32-bit 1: 16-bit If configured as 32-bit, the data output from the CORDIC_RDAT register is in q1.31 fixed-point format or single-precision floating-point. If configured as 16-bit, the data output from the CORDIC_RDAT register is in q1.15 fixed-point format or half-precision floating-point.
20	NUMWRITE	CORDIC_WDAT Register Parameter Count. 0: Only one 32-bit value needs to be written for the next computation (or two 16-bit values if INSIZE = 1). 1: Two 32-bit values must be written to the CORDIC_WDAT register to trigger the next computation. Reading will return the current state of the bit.
19	NUMREAD	Number of results that can be read from the CORDIC_RDAT register. 0: When the next computation is completed, only one 32-bit value (or two 16-bit values if OUTSIZE = 1) is transferred to the CORDIC_RDAT register. 1: Two 32-bit values are transferred to the CORDIC_RDAT register. When the next computation is completed, two 32-bit values are transferred to the CORDIC_RDAT register. It requires two reads of CORDIC_RDAT to reset the RRF flag. Some functions (such as PHASE) have two result outputs, and if the second result is not needed, it is allowed to set NUMREAD = 0.
18	DMAWEN	DMA Write Request Enable. 0: Disabled. No DMA write request is generated. 1: Enabled. DMA write requests are generated whenever the Cordic is not in the middle of a computation. This bit is set and cleared by software. The current state of this bit is returned when read.

Field offset	Register name	Description
17	DMAREN	DMA Read Request Enable. 0: Disabled. No DMA read request is generated. 1: Enabled. DMA read requests are generated whenever the RRF is set. This bit is set and cleared by software. The current state of this bit is returned when read.
16	INTEN	Interrupt Enable. 0: Disabled. No interrupt request is generated. 1: Enabled. An interrupt request is generated whenever the RRF flag is set. This bit is set and cleared by software. The current state of this bit is returned when read.
15:11	Reserved	Reserved, must be kept at the reset value.
10:8	SCALE[2:0]	Scaling Factor: $2^{SCALE[2:0]}$ 000: 2^0 001: 2^1 010: 2^2 … 110: 2^6 111: 2^7 When the actual input parameters exceed the specified input data range [-1,1), the actual input parameters need to be divided by $2^{SCALE[2:0]}$ and the output data needs to be multiplied by $2^{SCALE[2:0]}$ to obtain the actual output result. For example: CORDIC_WDAT = actual input parameters / $2^{SCALE[2:0]}$ Actual output result = CORDIC_RDAT * $2^{SCALE[2:0]}$
7:4	PRECISION	Iteration Count. 0: Reserved 1-15: Iteration count/4 Determines the number of iterations required for a specific precision. For most functions, the suggested range for this field is 3-6 iterations. Note: Higher iteration count results in higher precision.
3:0	FUNC	Functions: 4'b0: COS 4'b1: SIN 4'b2: PHASE 4'b3: MODULUS 4'b4: ARCTANT 4'b5: HB_COS: Hyperbolic Cosine 4'b6: HB_SIN: Hyperbolic Sine 4'b7: HB_ARCT: Hyperbolic Arctangent 4'b8: NATL: Natural Logarithm 4'b9: SQRT: Square Root 4'b10~4'b15: Reserved

22.5.3 CORDIC Write Data Register (CORDIC_WDAT)

Offset address: 0x04

Reset value: 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
WDAT[31:16]															
w															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WDAT[15:0]															
w															

Field offset	Register name	Description
31:0	WDAT[31:0]	Write Data. When a write access is made to this register, the write data is transferred to the address offset indicated by the write pointer. After each write, the pointer address automatically increments. This register is programmed based on the input parameters of the function selected in the FUNC field of the CORDIC_CTRLSTS register. If selecting the 32-bit format (INSIZE = 0) and requiring two input parameters (NUMWRITE = 1), then this register needs to be written to consecutively twice. The main parameter is written in the first instance, followed by the secondary parameter in the second instance. If selecting the 32-bit format and only needing one input parameter (NUMWRITE = 0), then writing to this register only once is required, containing the main parameter. If the selected format is 16-bit (INSIZE = 1), then writing to this register only once is needed, containing both parameters. The main parameter is located in WDAT[15:0], and the secondary parameter is located in WDAT[31:16]. In this case, NUMWRITE must be set to 0. For the parameters required for each function and their allowable ranges, please refer to section 22.4.1. After writing the required number of parameters, if the previous computation has been completed, CORDIC will evaluate the function specified in the FUNC field using the provided input parameters. If a computation is in progress, the values of the main parameter and secondary parameter remain pending until the computation is completed and the results are read. During this time, writes to the register will cancel the pending operation and overwrite the parameter data.

22.5.4 CORDIC Read Data Register (CORDIC_RDAT)

Offset address: 0x08

Reset value: 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RDAT[31:16]															

r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RDAT[15:0]															
r															

Field offset	Register name	Description
31:0	RDAT[31:0]	Output Data. If the 32-bit format is selected (OUTSIZE = 0) and two output values are expected (NUMREAD = 1), this register must be read twice when setting the RRF flag. The first read retrieves the main result (the first output result). The second read retrieves the secondary result (the second output result) and resets the RRF. If the 16-bit format is selected (OUTSIZE = 1), the lower half of this register, RDAT[15:0], contains the main result (the first output result), and the upper half, RDAT[31:16], contains the secondary result (the second output result). In this case, NUMREAD must be set to 0, and only one read operation is required. Reading data from this register will reset the RRF flag in the CORDIC_CTRLSTS register.

23 Universal Synchronous Asynchronous Receiver Transmitter (USART)

23.1 Introduction

USART is a full-duplex universal synchronous/asynchronous serial transceiver module. This interface is a highly flexible serial communication device that can perform full-duplex data exchange with external devices.

The USART has programmable transmit and receive baud rates and can communicate continuously using DMA. It also supports multiprocessor communication, LIN mode, synchronous mode, single-wire half-duplex communication, smartcard asynchronous protocol, IrDA SIR ENDEC function, and hardware flow control function.

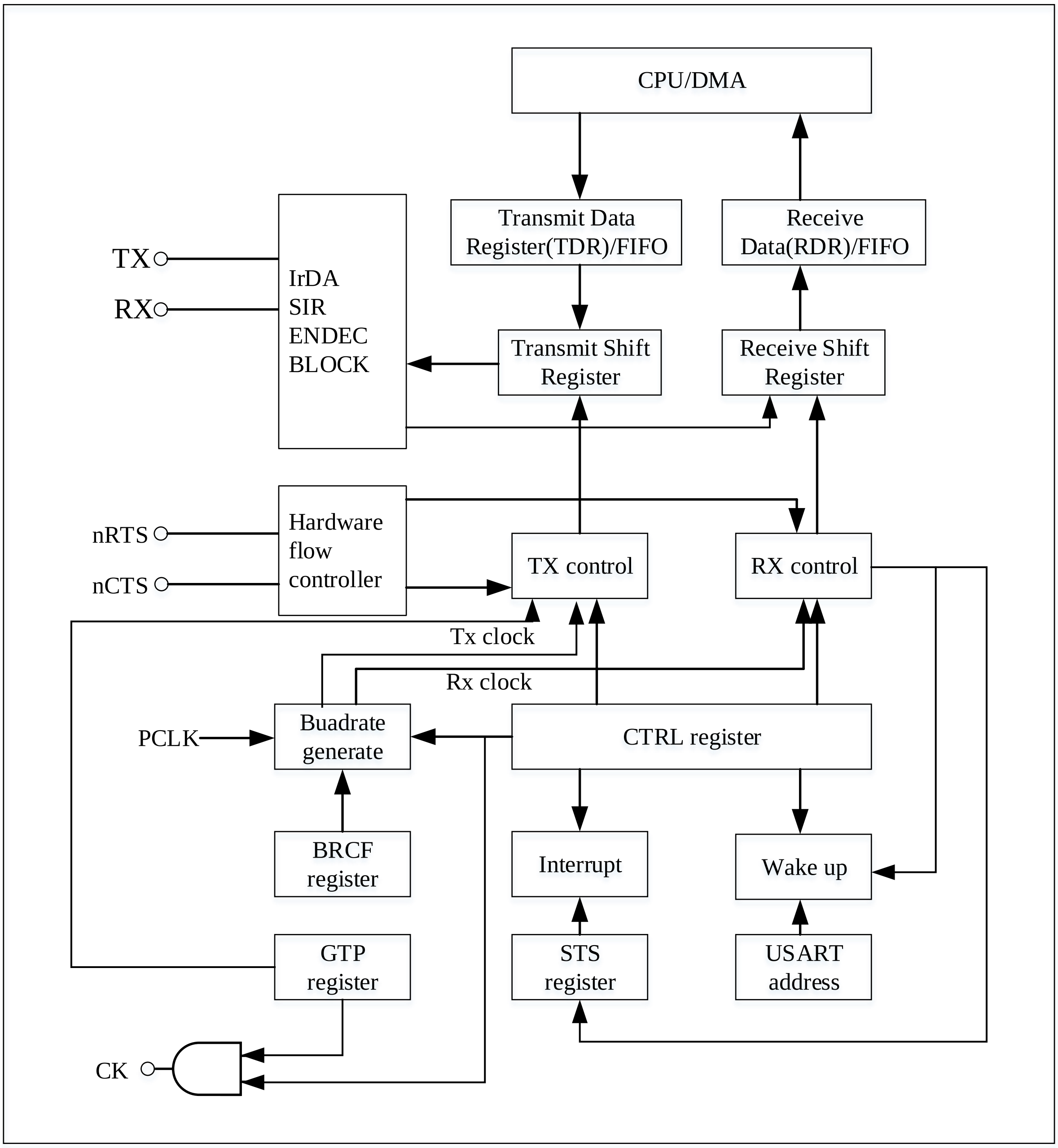
23.2 Main Features

- Supports full-duplex, asynchronous communication
- Supports single-wire half-duplex communication
- Configurable baud rate, with a maximum baud rate of 15Mbit/s
- Supports 8x or 16x oversampling
- Supports 8-bit or 9-bit data frames
- Features two internal FIFOs for transmitting and receiving data
- Supports 1-bit or 2-bit stop bits
- Hardware generation of parity bits and parity check
- Hardware flow control: RTS, CTS
- Supports RS-485
- Supports DMA for both transmission and reception
- Supports multi-processor communication: enters silent mode if address does not match, can be woken up by idle bus detection or address identification
- Supports synchronous mode, allowing users to control bidirectional synchronous serial communication in master mode
- Supports smart card asynchronous protocol, compliant with ISO7816-3 standard
- Supports Serial Infrared (IrDA SIR) protocol encoding and decoding, providing normal and low-power operation modes
- Supports LIN mode
- Supports multi-clock error detection: data overflow error, frame error, noise error, parity error
- Supports multiple interrupt requests: transmit data register empty, CTS flag, transmission complete, data

received, data overflow, bus idle, parity error, LIN mode break frame detection, and noise flag/overflow error/frame error in multi-buffer communication (DMA)

23.3 Functional Block Diagram

Figure 23-1 USART Block Diagram



23.4 Function Description

The bidirectional communications of any USART need to use the RX and TX pins of the external connection. Among them, TX is the output pin for serial data transmission. When the transmitter is active and not sending data, the TX pin is pulled high. When the transmitter is inactive, the TX pin reverts to the I/O port configuration. RX is an input pin for serial data reception, data is received by the oversampling technique.

The data packets of serial communication are transmitted from the sending device to the RX interface of the receiving device through its own TX interface, and the bus is in an idle state before transmitting or receiving. Frame format is: 1 start bit + 8 or 9 data bits (least significant bit first) + 1 parity bit (optional) + 0.5,1,1.5 or 2 stop bit.

Use the fractional baud rate generator to configure transmit and receive baud rates.

According to the block diagram, when using the hardware flow control mode, the nRTS output and nCTS input pins are required. When the USART receiver is ready to receive new data, nRTS becomes low level. If nCTS is valid (pulled to a low level), the next data is sent, otherwise the next frame of data is not sent.

When using synchronous mode, the CK pin is required. The CK pin is used for clock output for synchronous transfers. Clock phase and polarity are software programmable. During the start and stop bits, the CK pin does not output clock pulses.The CK pin is also used when using smartcard mode.

23.4.1 USART Frame Format

Start bit: 1 bit, active low.

Data bits: Configurable via USART_CTRL1.WL as 8 or 9 bits, with the LSB first.

Stop bit: Active high.

Idle frame: A complete data frame consisting entirely of '1's, including the start bit. followed by the start bit of a data frame containing the data .

Break frame: A break frame is a complete data frame consisting of '0's, including the stop bit. at the end of the break frame, the transmitter inserts 1 or 2 more stop bits ('1') to acknowledge the start bit.

Figure 23-2 Programming for Word Length = 8

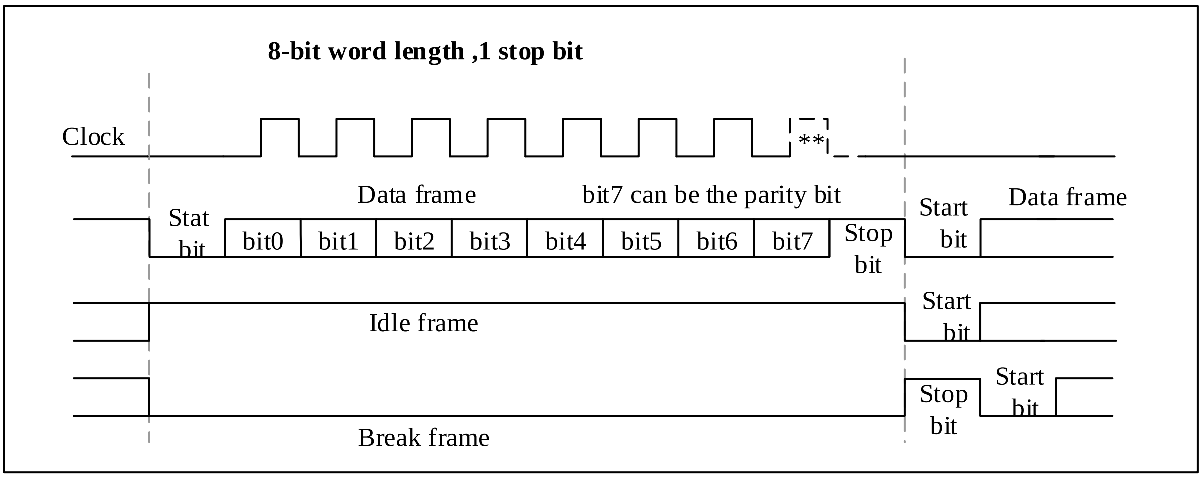
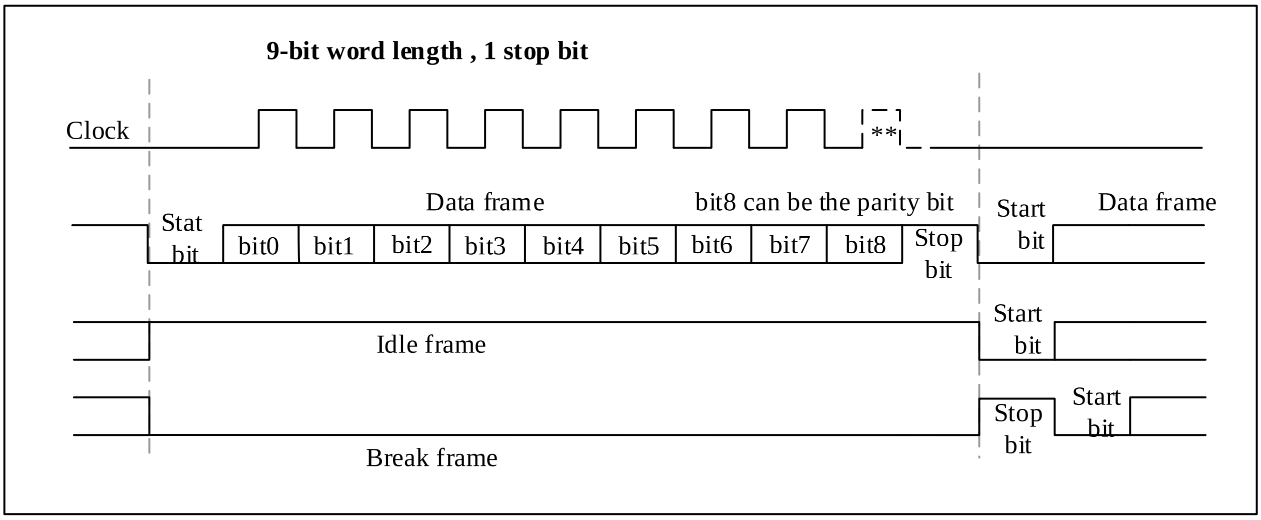


Figure 23-3 Programming for Word Length = 9



23.4.2 USART FIFO and Threshold

USART can operate in FIFO mode.

USART has a transmit FIFO (TXFIFO) and a receive FIFO (RXFIFO). FIFO mode can be enabled by setting USART_FIFO.EN to 1.

The maximum data word length is 9 bits, so TXFIFO is 9 bits wide. However, the default width of RXFIFO is 12 bits. This is because the receiver not only stores data in the FIFO but also stores error flags associated with each character (parity error, noise error, and frame error flags).

Note: Received data is stored together with the corresponding flags in RXFIFO, but only the data is read when reading USART_DAT. Status flags are located in the USART_STS register.

The TXFIFO and RXFIFO thresholds that trigger TX and RX interrupts can be configured. These thresholds are programmed through the RXFTCFG and TXFTCFG bit fields in the USART_FIFO control register.

In this case:

When the amount of data received in the RXFIFO reaches the threshold programmed in the RXFTCFG bit field, an Rx interrupt is generated. At this point, the RXFT flag in the USART_STS register is set to 1. This indicates that RXFTCFG data has been received: there is 1 data in USART_RDR and (RXFTCFG-1) data in RXFIFO. For example, if RXFTCFG is programmed as "101", the RXFT flag will be set to 1 when the corresponding amount of data is received (1 data in USART_RDR and (FIFO size-1) data in RXFIFO). Since the RX FIFO is not full, the next received data will not set the overflow flag to 1.

When the number of empty slots in the TXFIFO reaches the threshold programmed in the TXFTCFG bit field, a Tx interrupt is generated.

There are internal registers available to view the number of data entries in the TXFIFO/RXFIFO.

Recommended configuration process for USART RX/TX FIFO:

1. Enable USART_FIFO.CLR to clear TX/RX FIFO data when in use, or operate on the USART_FIFO.CLR bit when wanting to clear FIFO data.

2. Configure interrupt enable: TX interrupt/RX interrupt.
3. Configure FIFO waterline thresholds: USART_FIFO.RXFTCFG/USART_FIFO.RXFTCFG (do not configure as 2'b00).
4. Enable the FIFO module by setting USART_FIFO.EN.

23.4.3 Transmitter

After the transmitter is enabled, the data in the transmit shift register is sent out through the TX pin.

23.4.3.1 Idle frame

Setting USART_CTRL1.TXEN will cause the USART to transmit an idle frame before the first data frame.

23.4.3.2 Character transmission

Idle frames are followed by characters sent. Each character is preceded by a low start bit. The transmitter sends 8-bit or 9-bit data according to the configuration of the data bit length, with the least significant bit first. If USART_CTRL1.TXEN is reset during a data transfer, it will cause the baud rate counter to stop counting and the data being transferred will be corrupted.

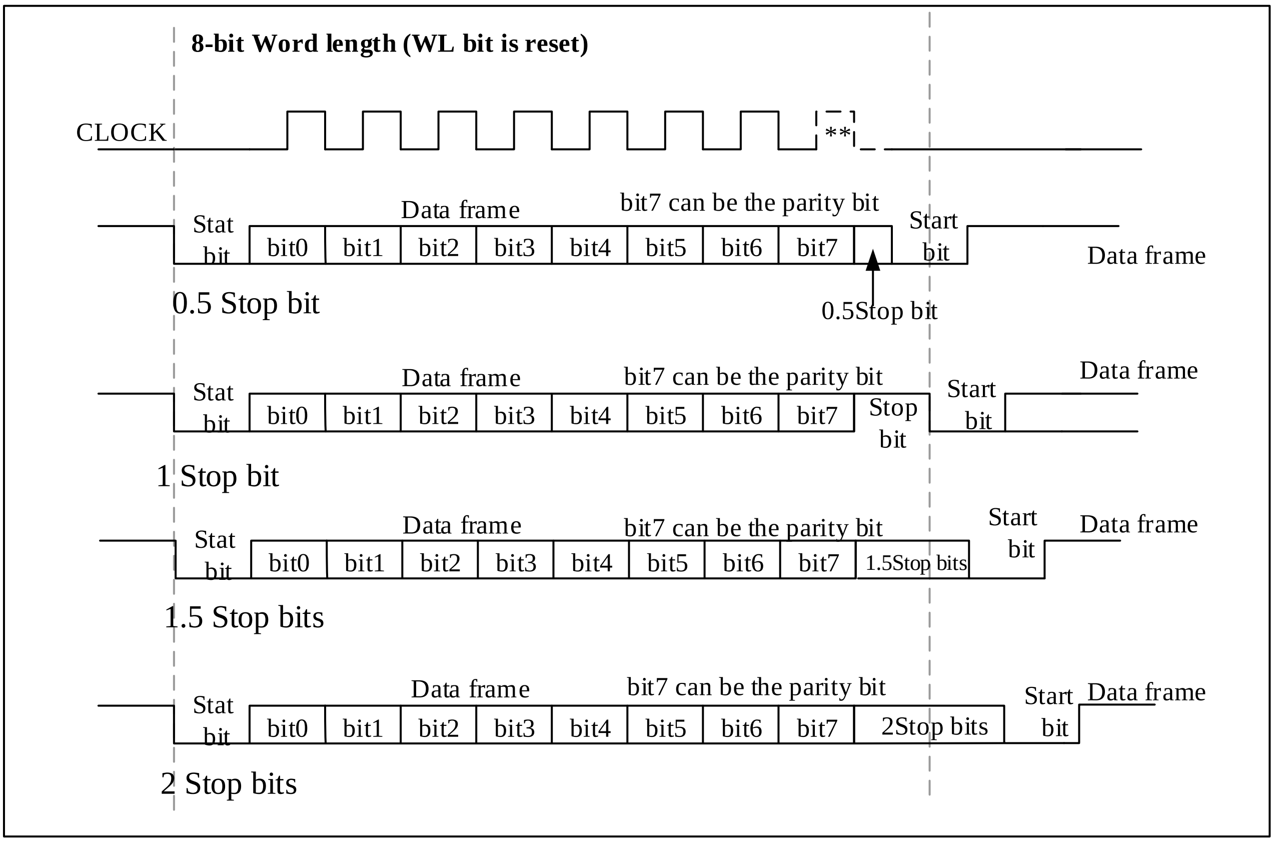
23.4.3.3 Stop bit

The characters are followed by stop bits, the number of which can be configured by setting USART_CTRL2.STPB[1:0].

Table 23-1 Stop Bit Configuration

USART_CTRL2.STPB[1:0]	Stop Bit Length (Bits)	Functional Description
00	1	default
01	0.5	Receiving in Smartcard mode
10	2	General USART mode, single-wire mode and modem mode.
11	1.5	Transmitting and receiving in Smartcard mode

Figure 23-4 Configuration Stop Bit



23.4.3.4 Break frame

Use USART_CTRL1.SDBRK to send the break character. When there is 8-bit data, the break frame consists of 10 bits of low level, followed by a stop bit; when there is 9-bit data, the break frame consists of 11 bits of low level, followed by a stop bit.

After the break frame is sent, USART_CTRL1.SDBRK is cleared by hardware, and the stop bit of the break frame is automatically sent. Therefore, to send a second break frame, USART_CTRL1.SDBRK should be set after the stop bit of the previous break frame has been sent.

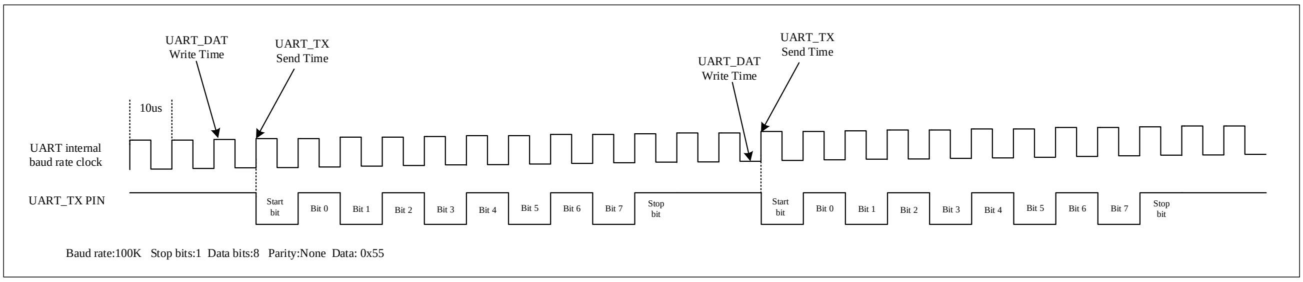
If software resets the USART_CTRL1.SDBRK bit before starting to send the break frame, the break frame will not be sent.

23.4.3.5 Transmission process

1. Enable USART_CTRL1.UEN to activate USART;
2. Configure the transmitter's baud rate, data bit length, parity bit (optional), the number of stop bits or DMA configuration;
3. Activate the transmitter (USART_CTRL1.TXEN);
4. Send each data to be sent to the USART_DAT register through the CPU or DMA, and the write operation to the USART_DAT register will clear USART_STS.TXDE;
5. After writing the last data word in the USART_DAT register, wait for USART_STS.TXC =1, which indicates

the end of the transmission of the last data frame.

Note: There will be a delay of 0~1 baud rate cycle from writing data to UART_DAT to data to UART_TX pin. For example, in the following figure with 100K baud rate, writing data to UART_DAT at any moment of one baud rate cycle will be transmitted to the UART_TX pin at the beginning of the next baud rate cycle.



23.4.3.6 Single Byte Communication

A write to the USART_DAT register clears the USART_STS.TXDE bit.

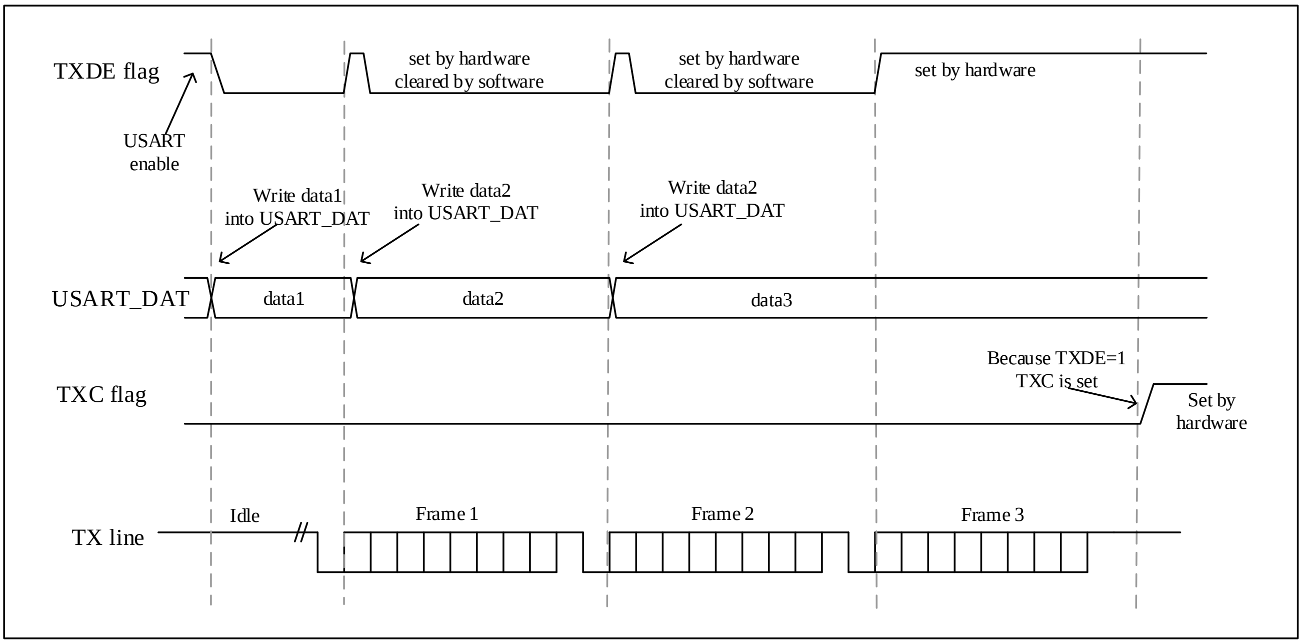
The USART_STS.TXDE bit is set by hardware when the data in the TDR register is transferred to the transmit shift register (indicating that data is being transmitted). An interrupt will be generated if USART_CTRL1.TXDEIEN is set. At this point, the next data can be sent to the USART_DAT register.

Write operation to USART_DAT register:

- When the transmit shift register is not sending data and is in an idle state, the data is directly put into the shift register for transmission, and the USART_STS.TXDE bit is set by hardware;
- When the transmit shift register is sending data, the data is stored in the TDR register, and after the current transmission is completed, the data is put into the shift register.

When a frame containing data is sent and USART_STS.TXDE=1, the USART_STS.TXC bit is set to '1' by hardware. An interrupt is generated if USART_CTRL1.TXCIEN is '1'. USART_STS.TXC bit is cleared by a software sequence (read USART_STS register first, then write USART_DAT register).

Figure 23-5 TXC/TXDE Changes During Transmission



23.4.4 Receiver

23.4.4.1 Start bit detection

When the received sampling sequence is: 1 1 1 0 X 0 X 0 X 0 0 0 0, it is considered that a start bit is detected.

The samples at the 3rd, 5th, and 7th bits, and the samples at the 8th, 9th, and 10th bits are all '0' (that is, 6 '0'), then confirm the receipt of the start bit, the USART_STS.RXDNE flag bit is set, and if USART_CTRL1.RXDNEIEN=1, an interruption occurs and will not Set the NEF noise flag.

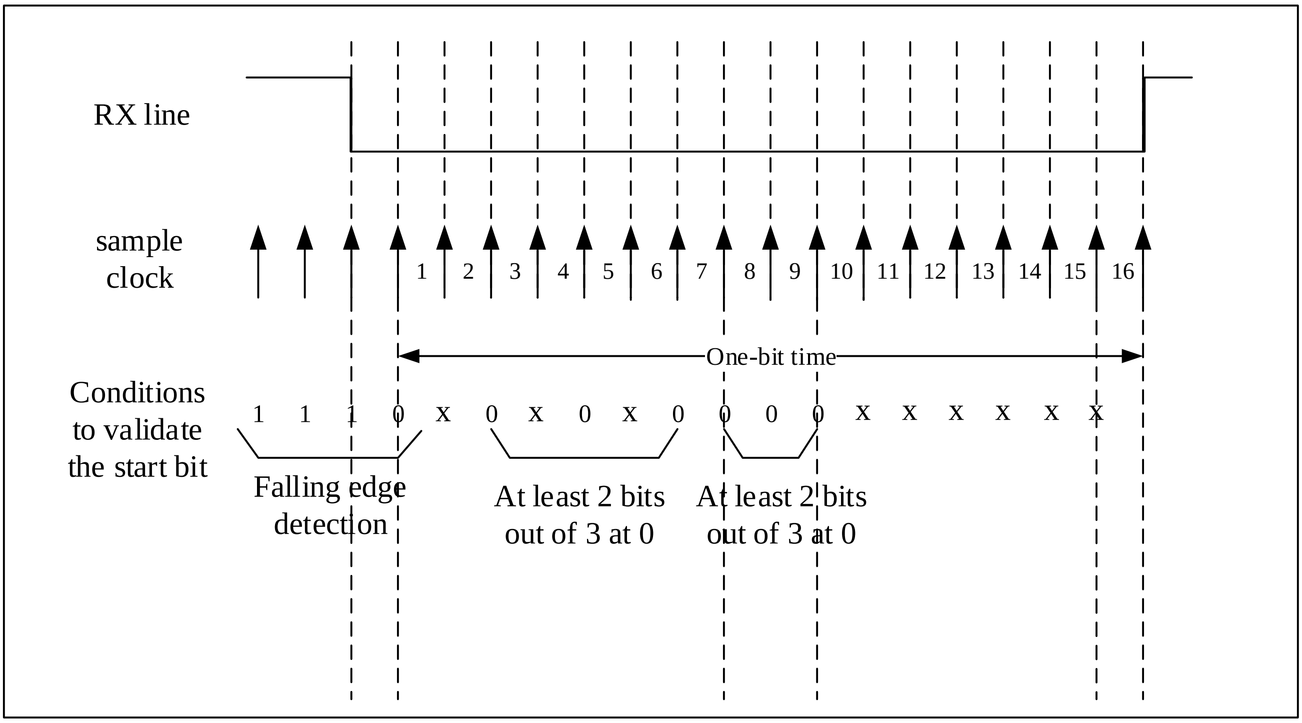
If there are six ‘0’ samples at the 3rd, 5th, 7th bits, and at the 8th, 9th, 10th bits, a start bit is confirmed to have been received, and USART_STS.RXDNE is set to 1, but the NE noise flag will not be set. If USART_CTRL1.RXDNEIEN has been set to 1, an interrupt will be generated.

If there are two '0' samples at the 3rd, 5th, 7th bits, and at the same time, there are two '0' samples at the 8th, 9th, 10th bits, a start bit is also confirmed to have been received, but the NE noise flag will be set.

If there are three '0' samples at the 3rd, 5th, 7th bits, and at the same time, there are two '0' samples at the 8th, 9th, 10th bits, a start bit is also confirmed to have been received, and the NE noise flag will be set.

If there are two '0' samples at the 3rd, 5th, 7th bits, and at the same time, there are three '0' samples at the 8th, 9th, 10th bits, a start bit is also confirmed to have been received, and the NE noise flag will be set.

Figure 23-6 Start Bit Detection


RX line
sample clock
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
One-bit time
Conditions to validate the start bit
1 1 1 0 X 0 X 0 X 0 0 0 0 X X X X X X
Falling edge detection
At least 2 bits out of 3 at 0
At least 2 bits out of 3 at 0

23.4.4.2 Stop bit description

During data reception, the number of data stop bits can be configured by the USART_CTRL2.STPB[1:0]. In normal mode, 1 or 2 stop bits can be selected. In Smartcard mode, 0.5 or 1.5 stop bits can be selected.

1. 0.5 stop bits (receive in smartcard mode): 0.5 stop bits are not sampled. Therefore, if 0.5 stop bits is selected, framing errors and broken frames cannot be detected.
2. 1 stop bit: sampling for 1 stop bit is done on the 8th, 9th and 10th samples.
3. 1.5 stop bits (Smartcard mode): when transmitting in Smartcard mode, the device must check whether the data is sent correctly. So the receiver function block must be activated (USART_CTRL1.RXEN=1) and sample the signal on the data line during the transmission of the stop bit. If a parity error occurs, the smartcard will pull down the data line when the transmitter samples the NACK signal, that is, within the time corresponding to the stop bit on the bus, indicating that a framing error has occurred. The USART_STS.FEF is set together with the USART_STS.RXDNE at the end of the 1.5th stop bit. The 1.5 stop bits are sampled at points 16, 17 and 18. The 1.5 stop bits can be divided into two parts: one is 0.5 clock cycles, during which nothing happens. This is followed by the stop bit of 1 clock cycle, which is sampled at the midpoint of this period of time. For details, see 23.4.17 Smartcard mode.
4. 2 stop bits: the sampling of the 2 stop bits is completed at the 8th, 9th and 10th sampling points of the first stop position. If a frame error is detected during the first stop bit, the frame error flag is set. The second stop bit does not detect framing error. The USART_STS.RXNE flag will be set at the end of the first stop bit.

23.4.4.3 Receiving process

1. Enable USART_CTRL1.UEN to activate USART;
2. Configure the receiver's baud rate, data bit length, parity bit (optional), stop bit number or DMA configuration;

3. Activate the receiver (USART_CTRL1.RXEN) and start looking for the start bit;
4. The receiver receives 8-bit or 9-bit data according to the configuration of the data bit length, and the least significant bit of the data is first shifted from the RX pin into the receive shift register;
5. When the data of the received shift register is moved to the RDR register, USART_STS.RXDNE is set, and the data can be read out. If USART_CTRL1.RXNEIEN is 1, an interrupt will be generated;
6. When an overflow error, noise error, or frame error is detected in the received frame, the corresponding error flag status bit will be set. If USART_CTRL1.RXEN is reset during data transmission, the data being received will be lost;
7. USART_STS.RXDNE is set after receiving data, and a read operation of USART_DAT can clear this bit:
   - During multi-buffer communication, the data register is cleared by the DMA read operation;
   - During single-buffer communication, it is cleared by software reading the USART_DAT register.

23.4.4.4 Idle frame detection

The receiver of the USART can detect idle frames. An interrupt is generated if USART_CTRL1.IDLEIEN is '1'. USART_STS.IDLEF bit is cleared by a software sequence (read USART_STS register first, then read USART_DAT register).

23.4.4.5 Break frame detection

The frame error flag(USART_STS.FEF) is set by hardware when the receiver detects a break frame. It can be cleared by a software sequence (read USART_STS register first, then read USART_DAT register).

23.4.4.6 Frame error

A framing error occurs when a stop bit is not received and recognized at the expected time. At this time, the frame error flag USART_STS.FEF will be set by hardware, and the invalid data will be transferred from the shift register to the USART_DAT register. During single-byte communication, no interrupt framing error will be generated because it occurs with USART_STS.RXDNE and the hardware will generate an interrupt when the USART_STS.RXDNE flag is set. In multi-buffer communication mode, an interrupt will be generated if the USART_CTRL3.ERRIEN bit is set.

23.4.4.7 Overflow error

In non-FIFO mode:

If USART_STS.RXDNE is set to 1 and there is data in the receive shift register that needs to be transferred to the data register, an overflow error occurs, and the USART_STS.OREF flag is set to 1.

In FIFO mode:

If the RX FIFO is full and there is data in the receive shift register that needs to be transferred to the data register, an overflow error occurs, and the USART_STS.OREF flag is set to 1. The data in the data register will not be lost in this case, but the data in the shift register will be overwritten.

The USART_STS.OREF flag can be cleared through the following software operation: first read the USART_STS register, then read the USART_DAT register.

When an overflow error occurs and USART_CTRL1.RXDNEIEN is set to 1, a receive interrupt will be generated.

In multi-buffer communication mode (DMA), if USART_CTRL3.ERRIEN is set to 1, an error interrupt will be generated.

23.4.4.8 Noise error

USART_STS.NEF is set by hardware when noise is detected on a received frame. It is cleared by software sequence (read USART_STS register first, then write USART_DAT register). During single-byte communication, no noise interrupt generated because it occurs with USART_STS.RXDNE and the hardware will generate an interrupt when the USART_STS.RXDNE flag is set. In multi-buffer communication mode(DMA), an interrupt is generated when the USART_STS.NEF flag is set if the USART_CTRL3.ERRIEN bit is set.

Table 23-2 Data Sampling for Noise Detection

Sample value	NE status	Received bits	Data validity
000	0	0	Effective
001	1	0	be invalid
010	1	0	be invalid
011	1	1	be invalid
100	1	0	be invalid
101	1	1	be invalid
110	1	1	be invalid
111	0	1	Effective

23.4.5 Fractional Baud Rate Calculation

The baud rate is configured through the USART_BRCF register, which consists of integer and fractional parts of the divisor and is applicable to both the transmitter and receiver. After writing to the USART_BRCF register, the baud rate counter will be replaced by the new value of the baud rate register. Therefore, do not change the value of the baud rate register during communication.

Oversampling set to 8:

$$\text{TX / RX baud rate} = f_{\text{PCLK}} /(8 * \text{USARTDIV})$$

Oversampling set to 16:

$$\text{TX / RX baud rate} = f_{\text{PCLK}} /(16 * \text{USARTDIV})$$

where f_{PCLK} is the clock provided to the peripheral:

- PCLK1 is used for USART2, USART3, USART5, USART8, up to 180MHz
- PCLK2 is used for USART1, USART4, UART6, UART7, up to 180MHz.

USARTDIV is an unsigned fixed-point number.

23.4.5.1 USARTDIV and USART_BRCF register configuration

Oversampling is set to 16:

Example 1:

If USARTDIV = 27.75, then:

DIV_Decimal = 16*0.75 = 12 = 0x0C

DIV_Integer = 27=0x1B

So USART_BRCF = 0x1BC

Example 2:

If USARTDIV = 20.98, then:

DIV_Decimal = 16*0.98 = 15.68

Nearest integer: DIV_Decimal = 16 = 0x10, out of configurable range, so a carry to integer is required

So DIV_Integer = 20+1 = 21 = 0x15

DIV_Decimal = 0x0

So USART_BRCF = 0x150

Example 3:

If USART_BRCF = 0x19B：

DIV_Integer = 0x19 = 25

DIV_Decimal = 0x0B = 11

So USARTDIV = 25+11/16 = 25.6875

Oversampling is set to 8:

Example 1:

If USARTDIV = 27.75, then:

DIV_Decimal = 8*0.75 = 6 = 0x06

DIV_Integer = 27=0x1B

So USART_BRCF = 0x1B6

Example 2:

If USARTDIV = 20.98, then:

DIV_Decimal = 8*0.98 = 7.84

Nearest integer: DIV_Decimal = 8 = 0x08, out of configurable range, so a carry to integer is required

So DIV_Integer = 20+1 = 21 = 0x15

DIV_Decimal = 0x0

So USART_BRCF = 0x150

Example 3:

If USART_BRCF = 0x196：

DIV_Integer = 0x19 = 25

DIV_Decimal = 0x06 = 6

USARTDIV = 25+6/8 = 25.75

Table 23-3 Error Calculation When Setting Baud Rate

16x oversampling（USART_CTRL1.OSPM = 0）									
Baud rate		**f_{PCLK}=120MHz**				**f_{PCLK}=180MHz**			
serial number	**Kbps**	**reality**	**Set value in register**	**Error(%)**	**reality**	**Set value in register**	**Error(%)**	**serial number**	**Kbps**
1	2.4	2.4	16	3125	0%	2.4	16	impossible	impossible
2	9.6	9.6	16	781.25	0%	9.6	16	1171.875	0%
3	19.2	19.2	16	390.625	0%	19.2	16	585.9375	0%
4	57.6	57.609	16	130.1875	0.02%	57.6	16	195.5	0%
5	115.2	115.163	16	65.125	0.03%	115.2	16	97.75	0%
6	230.4	230.326	16	32.5625	0.03%	230.474	16	48.8125	0.03%
7	460.8	461.538	16	16.25	0.16%	461.538	16	24.375	0.16%
8	921.6	923.076	16	8.125	0.16%	923.076	16	12.1875	0.16%
9	7500	7500	16	1	0%	7500	16	1.5	0%
10	15000	impossible	16	impossible	impossible	15000	16	impossible	impossible

Table 23-4 Error Calculation When Setting Baud Rate

8x oversampling（USART_CTRL1.OSPM = 1）		
Baud rate	**f_{PCLK}=120MHz**	**f_{PCLK}=180MHz**

serial number	Kbps	reality	Set value in register	Error(%)	reality	Set value in register	Error(%)	serial number	Kbps
1	2.4	2.4	8	impossible	impossible	2.4	8	impossible	impossible
2	9.6	9.6	8	1562.5	0%	9.6	8	2343.75	0%
3	19.2	19.2	8	781.25	0%	19.2	8	1171.875	0%
4	57.6	57.56	8	390.625	0.02%	57.6	8	390.625	0%
5	115.2	115.163	8	130.25	0.03%	115.2	8	195.5	0%
6	230.4	230.769	8	65.125	0.03%	230.474	8	97.75	0.03%
7	460.8	461.538	8	32.5	0.16%	460.358	8	48.875	0.10%
8	921.6	923.076	8	16.25	0.16%	923.076	8	24.375	0.16%
9	7500	7500	8	2	0%	7500	8	3	0%
10	15000	15000	8	1	0%	15000	8	1.5	0%

Note: The lower the CPU clock frequency, the lower the error for a specific baud rate..

23.4.6 USART Receiver Tolerance to Clock Variations

In the application, there may be transmission errors (including changes in the transmitter clock), receiver baud rate errors and oscillator variations, and changes in transmission lines (usually caused by inconsistent data rising and falling edge timing). These factors will all affect the overall clock system variations. Only when the sum of the above four variations is less than the tolerance of the USART receiver, can the USART asynchronous receiver work properly.

During normal data reception, the tolerance of the USART receiver is the maximum allowable variation, depending on the choice of data bit length and whether fractional baud rate divider is used.

Table 23-5 When DIV_Decimal = 0, the tolerance of the USART receiver

WL bit	PCEN bit	Considering NEF as an error	Not considering NEF as an error
0	0	3.75%	4.375%
1	0	3.41%	3.97%
0	1	2.5%	3.75%
1	1	2.27%	3.41%

Table 23-6 When DIV_Decimal != 0, the tolerance of the USART receiver

WL bit	PCEN bit	Considering NEF as an error	Not considering NEF as an error
0	0	3.33%	4.0%
1	0	3.03%	3.63%
0	1	1.25%	2.5%
1	1	1.13%	2.27%

23.4.7 Parity Control

Parity can be enabled by configuring the USART_CTRL1.PCEN bit.

When the parity bit is enabled for transmission, a parity bit is generated, parity check is performed on reception.

Table 23-7 Frame Format

WL bit	PCEN bit	USART frame
0	0	| Start bit | 8-bit data | Stop bit |
0	1	| Start bit | 7 bits of data | Parity bit | Stop bit |
1	0	| Start bit | 9-bit data | Stop bit |
1	1	| start bit | 8-bit data | parity bit | stop bit |

Even parity

Configure USART_CTRL1.PSEL to 0, and even parity can be selected.

Make the number of '1' in the transmitted data (including parity bit) be an even number. For example: if Data=11000101, there are 4 '1's, then the parity bit will be '0' (4 '1' in total). The receiver confirms the number of '1's in the data If it is an even number, the check is passed, indicating that no errors occurred during the transmission process. If it is not even, it means that an error has occurred, the USART_STS.PEF flag is set to '1', and if USART_CTRL1.PEIEN is enabled, an interrupt is generated.

Odd parity

Configure USART_CTRL1.PSEL to 1, you can choose odd parity.

Make the number of '1' in the transmitted data (including parity bit) be an odd number. For example: if Data=11000101, there are 4 '1's, then the parity bit will be '1' (5 '1' in total). The receiver confirms the number of '1's in the data If it is an odd number, the check is passed, indicating that no errors occurred during the transmission process. If it is not an odd number, it means that an error has occurred, the USART_STS.PEF flag is set to '1', and if USART_CTRL1.PEIEN is enabled, an interrupt is generated.

23.4.8 DMA Communication

The USART supports the DMA mode using multi-buffer configuration, which can realize high-speed data communication.

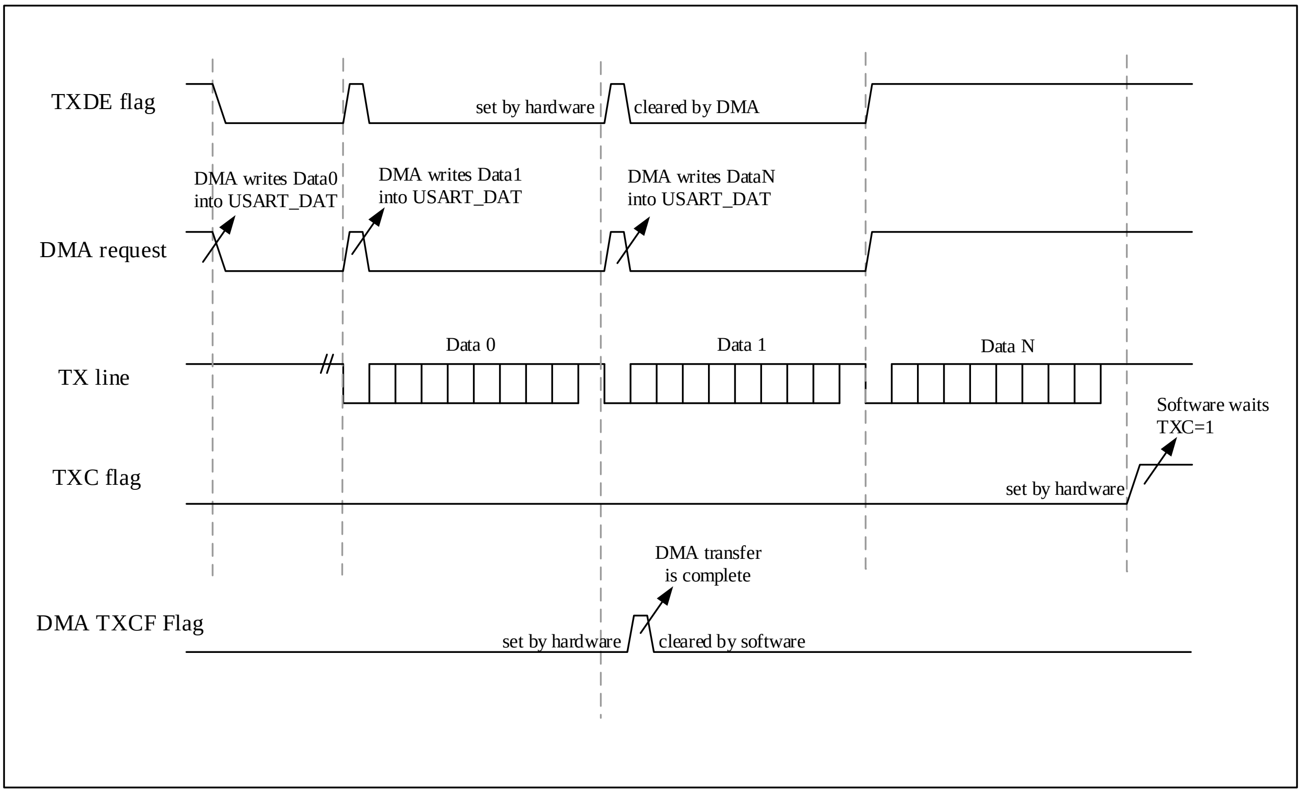
23.4.8.1 DMA transmission

Set USART_CTRL3.DMATXEN to enable DMA mode when transmitting. When the USART's transmit shift register is empty (USART_STS.TXDE=1), the DMA will transfer the data from the SRAM to the USART_DAT register of the USART.

When using DMA transmission, the process of configuring the DMA channel is as follows:

1. Set the address of the data memory. When a data transfer request occurs, the transferred data will be read from this address.
2. Set the address of the USART_DAT register. When a data transfer request occurs, this address will be the destination address of the data transfer.
3. Set the amount of data to transfer.
4. Set the priority of the channel, set whether to use the cyclic mode, the incremental mode of peripherals and memory, the data width of peripherals and memory, the interrupt generated by half of the transfer or the interrupt when the transfer is completed.
5. Start the current DMA channel.
6. After the data transfer is completed, the transfer complete flag (DMA_INTSTS.TXCFx) is set to 1.

Figure 23-7 DMA Transmission



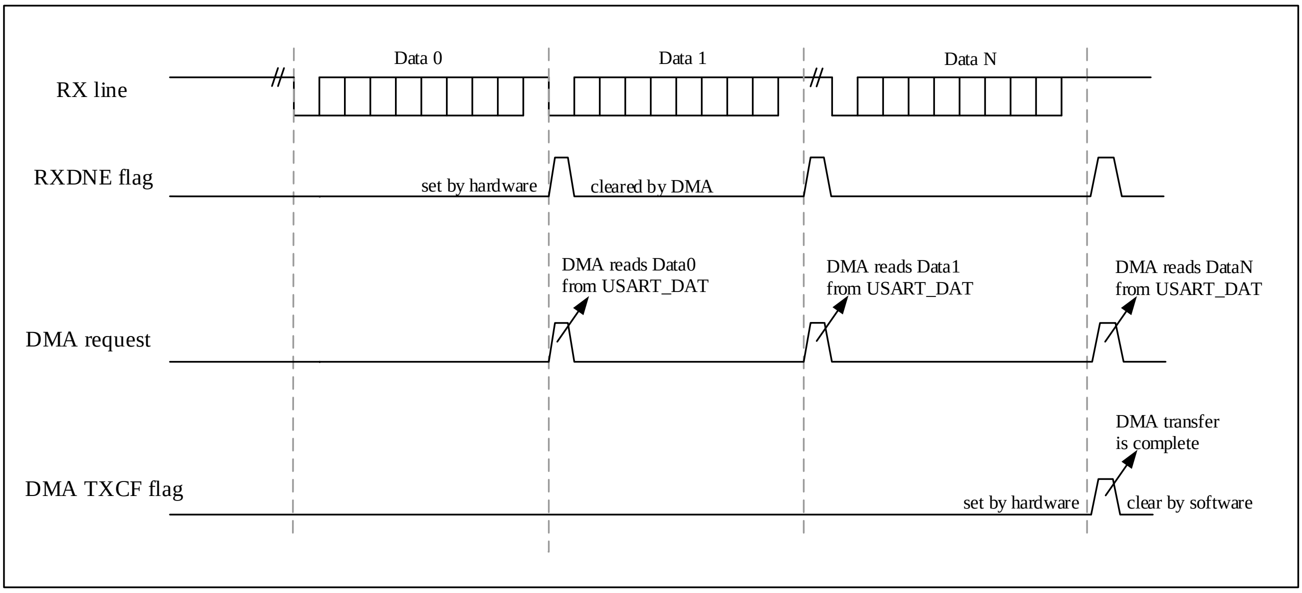
23.4.8.2 DMA reception

Set USART_CTRL3.DMARXEN to enable DMA mode when receiving. When a byte is received (USART_STS.RXDNE=1), the DMA will transfer the data from the USART_DAT register of the USART to the SRAM.

When using DMA reception, the process of configuring the DMA channel is as follows:

1. Set the address of the USART_DAT register. When a data transfer request occurs, this address will be the source address of the data transfer.
2. Set the address of the data memory. When a data transfer request occurs, the transferred data will be written to this address.
3. Set the amount of data to transfer.
4. Set the priority of the channel, set whether to use the cyclic mode, the incremental mode of peripherals and memory, the data width of peripherals and memory, the interrupt generated by half of the transfer or the interrupt when the transfer is completed.
5. Start the current DMA channel.

Figure 23-8 DMA Reception

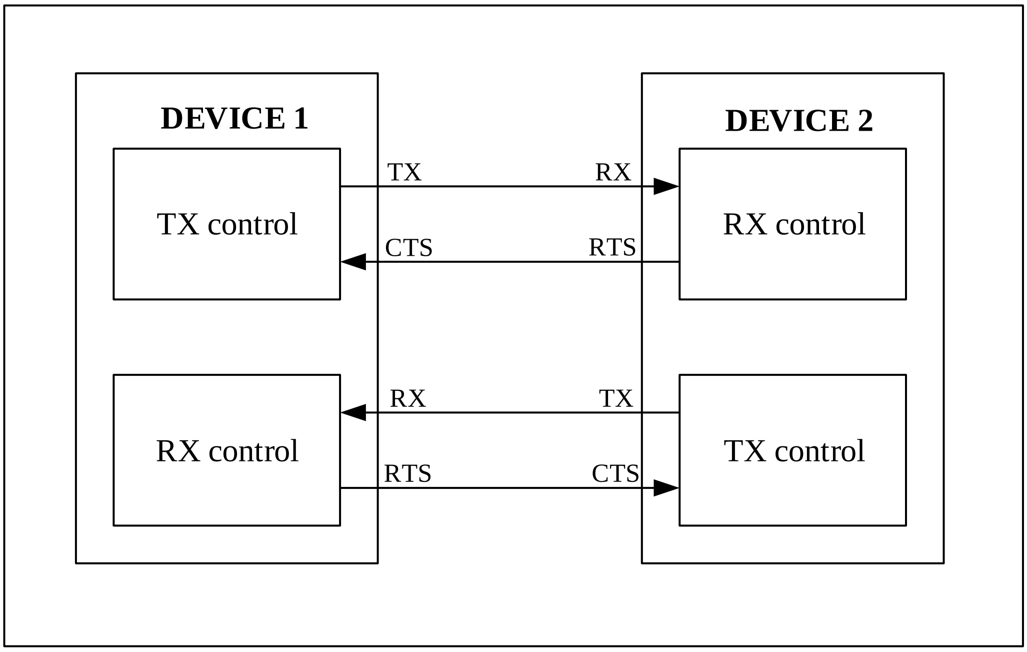


In multi-buffer communication mode, the error flag will be set when there is a frame error, overflow or noise error. An interrupt will be generated if the error interrupt is enabled (USART_CTRL3.ERRIEN=1).

23.4.9 Hardware Flow Control

USART supports hardware flow control. The purpose is to coordinate the transmitting and receiving parties so that the data will not be lost. The connection method is shown in the following figure.

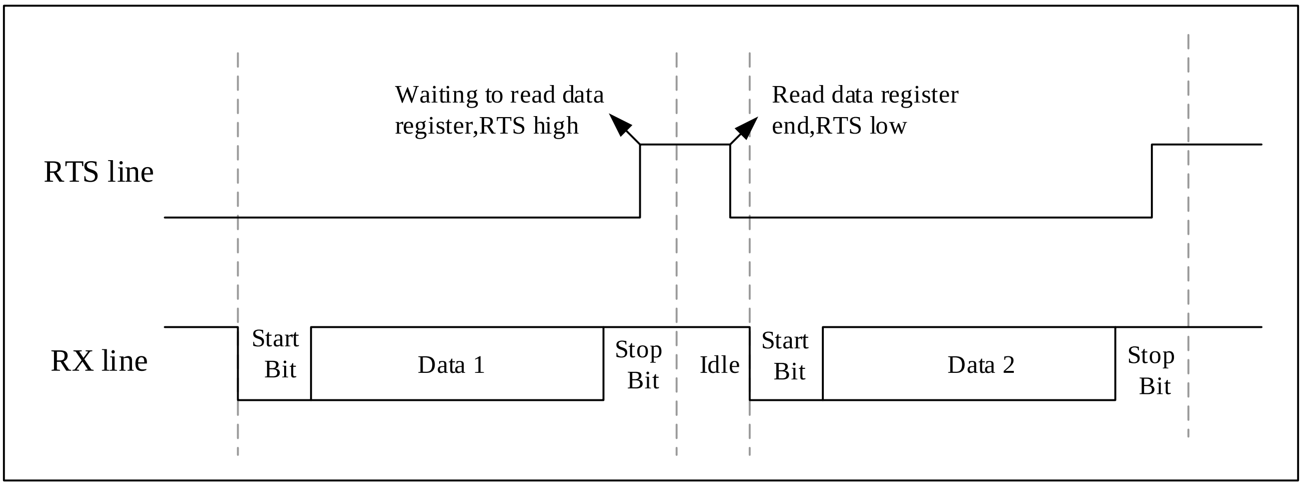
Figure 23-9 Hardware Flow Control Between Two USART



23.4.9.1 RTS flow control

Set USART_CTRL3.RTSEN to enable RTS. RTS is the output signal used to indicate that the receiver is ready. When data arrives in RDR, nRTS is asserted, notifying the sender to stop data transmission at the end of the current frame. When receiver is ready to receive new data, nRTS is deasserted.

Figure 23-10 RTS Flow Control

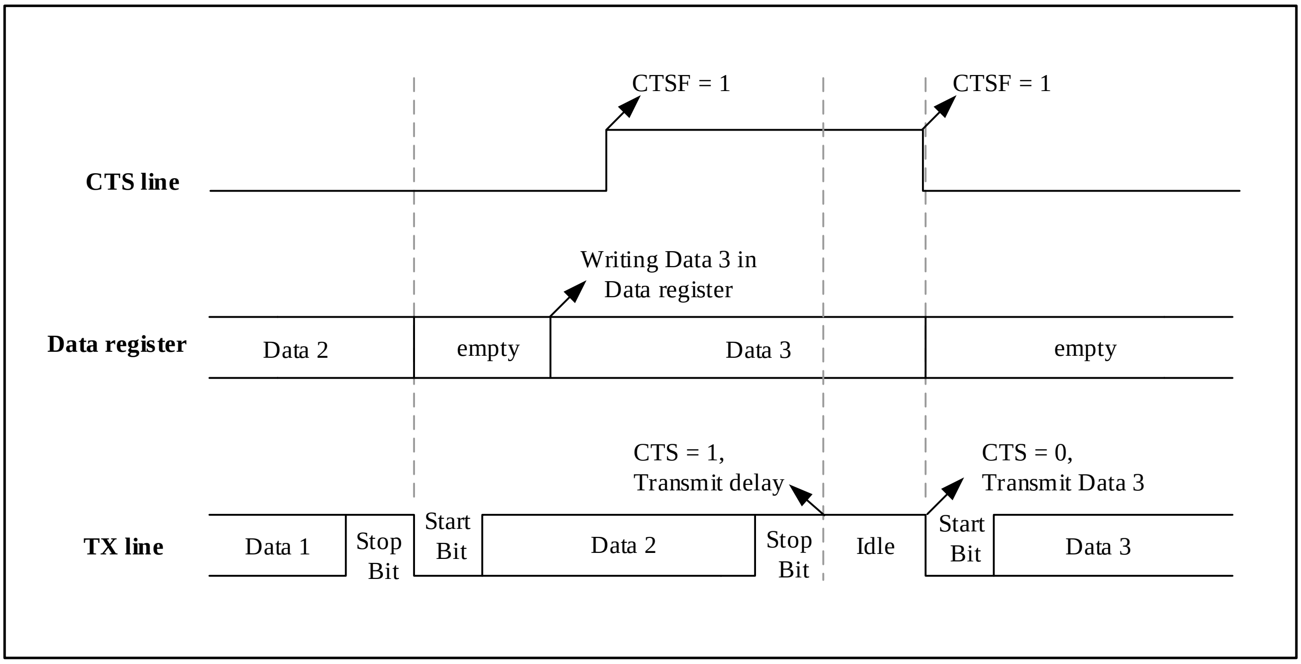


23.4.9.2 CTS flow control

Set USART_CTRL3.CTSEN to enable CTS. CTS is an input signal, used to judge whether data can be sent to the other device. The low level is valid, and the low level indicates that the device can send data to the other device. If the nCTS signal becomes invalid during data transmission, the transmission will stop after sending the data. If you write data to the data register when nCTS is invalid, the data will not be sent until nCTS is valid.

If the USART_CTRL3.CTSEN bit is set, the USART_STS.CTSF bit will be set high by hardware when the nCTS input changes state. An interrupt will be generated if USART_CTRL3.CTSIEN is enabled.

Figure 23-11 CTS Flow Control



23.4.10 Multiprocessor Communication

USART allows multiprocessor communication. When multiple processors communicate through USART, it is necessary to determine who is the master device, and the remaining processors are all slave devices. The TX output of the master device is directly connected to the RX port of all slave device. The TX outputs of the slaves are logically AND together and connected to the RX inputs of the master.

When multi-processor communication is performed, the slave devices are all in mute mode, and the host uses a specific method to wake up a slave device to be communicated when needed, so that the slave device is in an active state and transmits data with the master device.

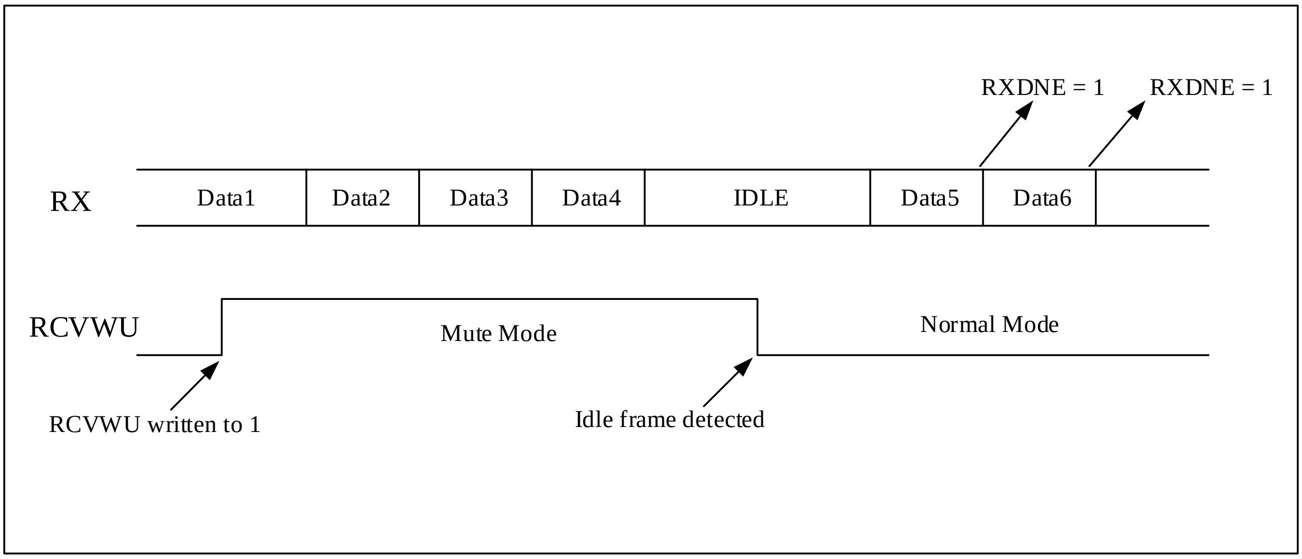
The USART can wake up from mute mode by idle line detection or address mark detection.

23.4.10.1 Idle bus detection

The idle line detection configuration process is as follows:

1. Configure the USART_CTRL1.WUM bit to 0, and the USART performs idle line detection;
2. When USART_CTRL1.RCVWU is set (which can be automatically controlled by hardware or written by software under certain conditions), USART enters mute mode. In mute mode, none of the receive status bits are set, and all receive interrupts are disabled;
3. As shown in the Figure 23-12 below, when an idle frame is detected, USART is woken up, and then USART_CTRL1.RCVWU is cleared by hardware. At this time, USART_STS.IDLEF is not set.

Figure 23-12 Mute Mode Using Idle Bus Detection



23.4.10.2 Address mark detection

By configuring the USART_CTRL1.WUM bit to 1, the USART performs address mark detection. The address of the receiver is programmable through the USART_CTRL2.ADDR[3:0] bits. If the MSB is 1, the byte is considered as an address, otherwise it is considered as data.

In this mode, the USART can enter mute mode by:

- When the receiver does not contain data, USART_CTRL1.RCVWU can be written to 1 by software, and USART enters mute mode;

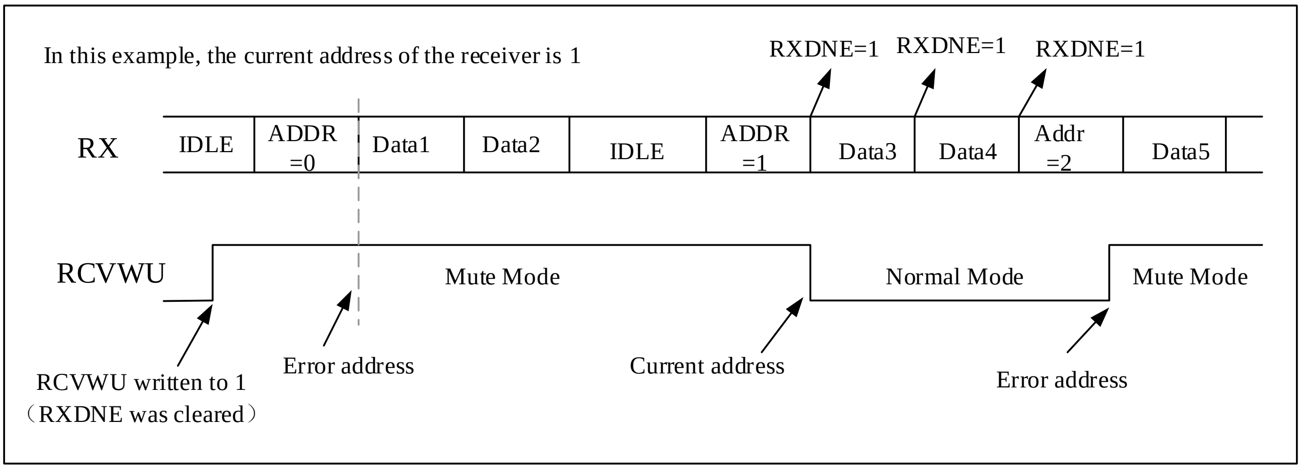
Note: When the receive buffer contains no data (RXNE=0 in USART_SR), the USART_CTRL1.RCVWU bit can be written to 0 or 1. Otherwise, the write operation is ignored.

- When the received address does not match the address of the USART_CTRL2.ADDR[3:0] bits, USART_CTRL1.RCVWU is written to 1 by hardware.

In mute mode, none of the receive status bits are set and all receive interrupts are disabled.

When the received address matches the address of the USART_CTRL2.ADDR[3:0] bits, the USART is woken up and USART_CTRL1.RCVWU is cleared. The USART_STS.RXDNE bit will be set when this matching address is received. Data can then be transmitted normally.

Figure 23-13 Mute Mode Detected Using Address Mark



23.4.11 Synchronous Mode

USART supports synchronous serial communication. The USART only supports the master mode, and cannot use the input clock from other devices to receive and transmit data. Synchronous mode can be enabled by configuring the USART_CTRL2.CLKEN bit.

Note: When using synchronous mode, USART_CTRL2.LINMEN, USART_CTRL3.SCMEN, USART_CTRL3.HDMEN, USART_CTRL3. IRDAMEN, these bits need to be kept clear.

23.4.11.1 Synchronized clock

The CK pin is the output of the USART transmitter clock. During the bus idle period, before the actual data arrives and when the break symbol is sent, the clock not output.

Clock phase and polarity are software programmable and need to be configured when both the transmitter and receiver are disabled. When the clock polarity is 0 (USART_CTRL2.CLKPOL=0), the default level of CLK is low; when the clock polarity is 1 (USART_CTRL2.CLKPOL=1), the default level of CLK is high. When the phase polarity is 0 (USART_CTRL2.CLKPHA=0), the data is sampled on the first edge of the clock; when the phase polarity is 1 (USART_CTRL2.CLKPHA=1), the data is sampled on the second edge.

During the start and stop bits, the CK pin does not output clock pulses.

A sync data cannot be received when no data is sent. Because the clock is only available when the transmitter is activated and data is written to the USART_DAT register.

The USART_CTRL2.LBCLK bit controls whether to output the clock pulse corresponding to the last data byte (MSB) sent on the CK pin. This bit needs to be configured when both the transmitter and receiver are disabled. If USART_CTRL2.LBCLK is 1, the clock pulse of the last bit of data will be output from CK. If USART_CTRL2.LBCLK is 0, the clock pulse of the last bit of data is not output from CK.

23.4.11.2 Synchronous transmitting

The transmitter in synchronous mode works the same as in asynchronous mode. Data on the TX pin is sent out synchronously with CK.

23.4.11.3 Synchronous receiving

The receiver in synchronous mode works differently than in asynchronous mode. Data is sampled on CK without any oversampling. But setup time and hold time (depending on baud rate, 1/16 bit time) must be considered.

Figure 23-14 USART Synchronous Transmission Example

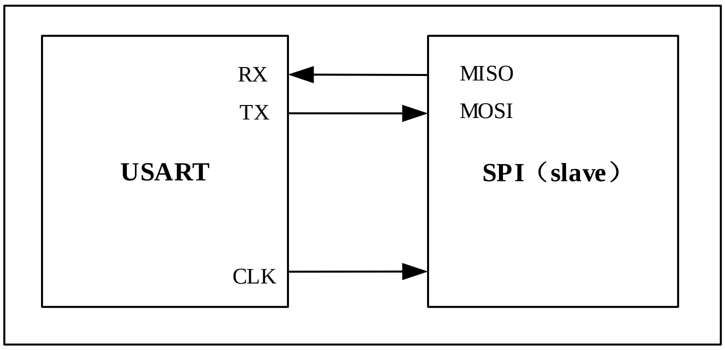


Figure 23-15 USART Data Clock Timing Example (WL=0)

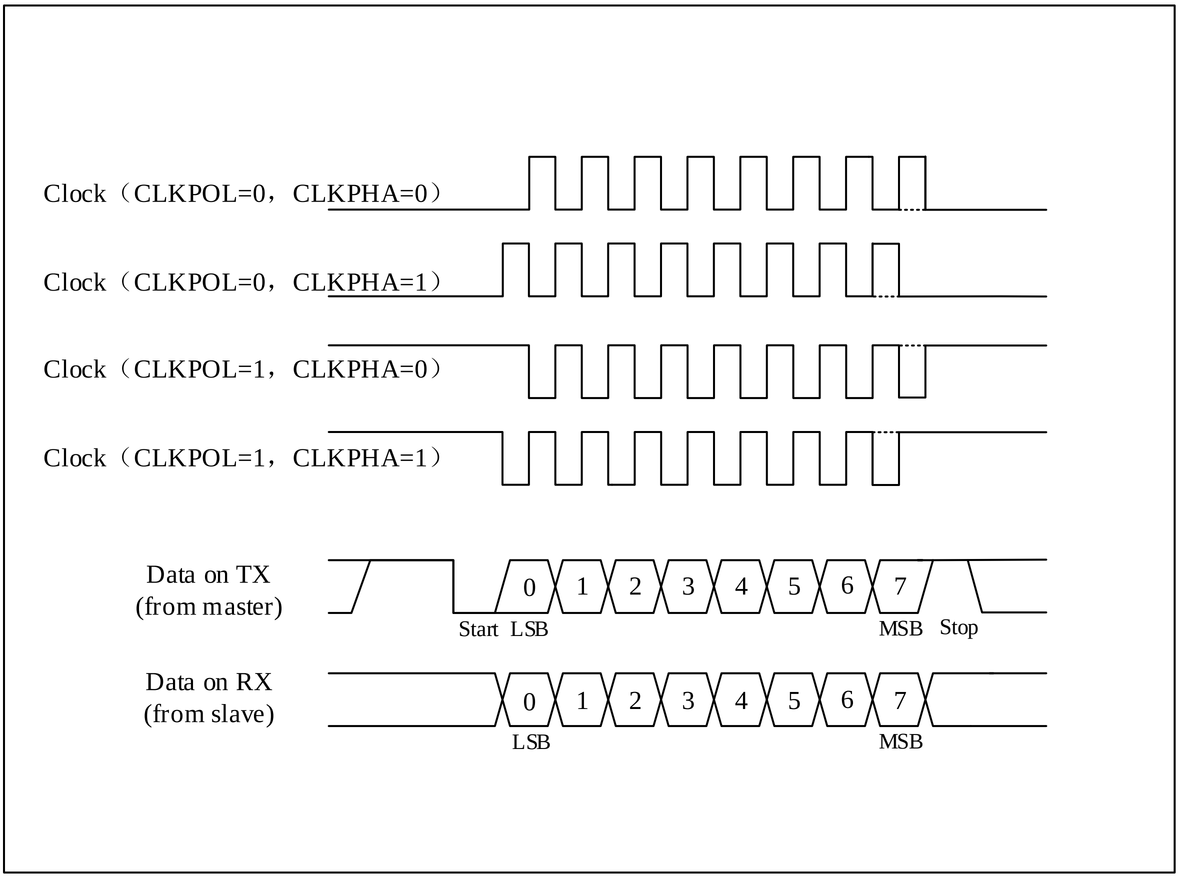


Figure 23-16 USART Data Clock Timing Example (WL=1)

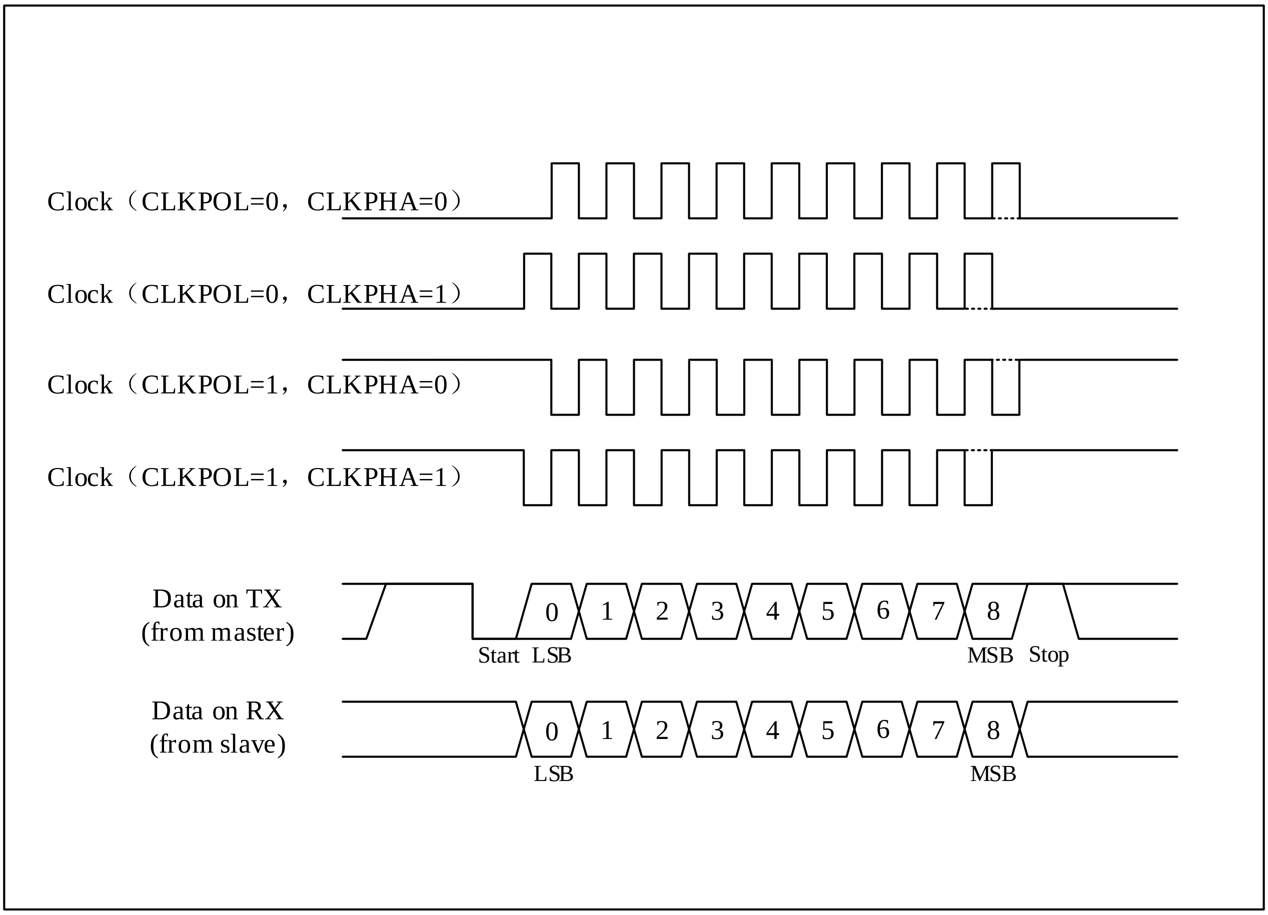
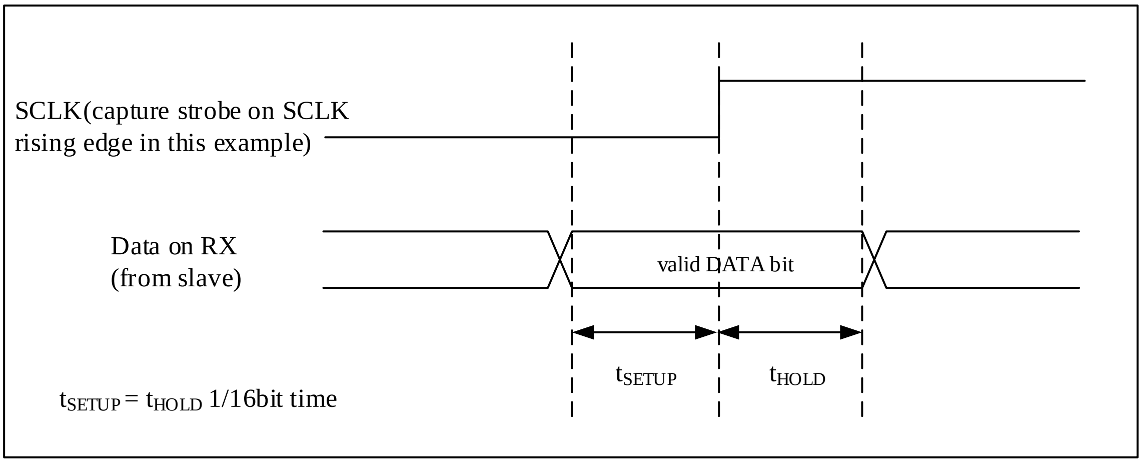


Figure 23-17 RX Data Sampling / Holding Time


SCLK(capture strobe on SCLK
rising edge in this example)
Data on RX
(from slave)
valid DATA bit
tSETUP
tHOLD
tSETUP = tHOLD 1/16bit time

Note: the function of CK is different in Smartcard mode, please refer to the Smartcard mode section for details.

23.4.12 Single-wire Half-duplex Mode

USART supports single-wire half-duplex communication, allowing data to be transmitted in both directions, but only allows data to be transmitted in one direction at the same time. Communication conflicts are managed by software.

Through the USART_CTRL3.HDMEN bit, you can choose whether to enable half-duplex mode. When using single-wire half-duplex, USART_CTRL2. CLKEN, USART_CTRL2. LINMEN, USART_CTRL3. SCMEN, USART_CTRL3. IRDAMEN, these bits should be kept clear.

After the half-duplex mode is turned on, the TX pin and the RX pin are internally connected, and the RX pin is no longer used. When there is no data to be transmitted, TX is always released. Therefore, when not driven by the USART, the TX pin must be configured as a floating input or an open-drain output high.

23.4.13 Receiver Timeout

Receiver timeout function can be enabled by setting USART_CTRL2.RTOEN to 1.

The timeout interval is programmed through the USART_RTO.TIME field.

The receiver timeout counter starts counting following these rules: it begins counting from the end of the stop bit reception.

After the timeout interval elapses, the USART_STS.RTOF flag is set to 1. If USART_CTRL1.RTOIE is set to 1, a timeout interrupt will be generated.

Note: After a receiver timeout, it is necessary to clear the timeout flag USART_STS.RTOF, otherwise, the timeout counting will not restart; after clearing the flag, the receiver timeout counting starts only after receiving the next data.

23.4.14 Data error discard function

USART has parity error flag (PEF), framing error flag (FEF), and noise error flag (NEF) for data checking on USART. Configuration of USART_CTRL2.PEFLOSE/FEFLOSE/NEFLOSE can be done on USART to check for data errors in the data written to the receive FIFO. If one of these errors occurs in the receive FIFO, the data will not be written into the FIFO. It is as if the data is directly discarded, and the USART_STS.PELOSEF/FELOSEF/NELOSEF three status bits indicate the occurrence of the respective errors in the previous data. This function is used under the condition of receiving data in FIFO mode.

Indicate error flag:

Set: Indicates the type of error that occurred when a data error occurs;

Clear: Write a 1 to USART_STS.PELOSEF/FELOSEF/NELOSEF to clear.

23.4.15 Serial IrDA Infrared Encoding/Decoding Mode

USART supports the IrDA (Infrared Data Association).

Through the USART_CTRL3. IRDAMEN bit, you can choose whether to enable the infrared mode. When using the infrared function, USART_CTRL2.CLKEN, USART_CTRL2.STPB[1:0], USART_CTRL2.LINMEN, USART_CTRL3.HDMEN, USART_CTRL3.SCMEN, these bits should be kept clear.

Through the USART_CTRL3. IRDALP bit, it can be used to select normal mode or low power infrared mode.

23.4.15.1 IrDA normal mode

When USART_CTRL3.IRDALP=0, select normal infrared mode.

IrDA is a half-duplex communication protocol, so there should be a minimum delay of 10ms between sending and receiving.that uses a inverted return-to-zero modulation scheme (RZI), which uses an infrared light pulse to represent a logic '0', and the pulse width is specified as 3/16 of a bit period in normal mode, as shown in the Figure 23-19. USART only supports up to 115200bps.

The USART sends data to the SIR encoder, and the bit stream output by the USART will be modulated. A modulated stream of pulses is sent from the infrared transmitter and then received by the infrared receiver. The SIR receiver decoder demodulates it and outputs the data to the USART.

The transmit encoder output has opposite polarity to the decoder input. When idle, SIR transmit is low, while SIR receive is high. The high pulse sent by SIR is '0' and the low level is '1', while SIR reception is the opposite.

If the USART is sending data to the IrDA transmit encoder, then the IrDA receive decoder will ignore any data on the IrDA receive line. If the USART is receiving data sent from the SIR receiver decoder, the data sent by the USART to the IrDA transmitter encoder will not be encoded.

Pulse width is programmable. The IrDA specification requires pulses to be wider than 1.41us. For pulse widths less than 2 cycles, the receiver will filter them out. PSCV is the prescaler value programmed in the USART_GTP register.

23.4.15.2 IrDA low power mode

When USART_CTRL3.IRDALP=1, select low power infrared mode.

For the transmitter, when in low power mode, the pulse width is 3 times the low power baud rate, which is a minimum of 1.42MHz. Typically this value is 1.8432MHz (1.42 MHz < PSC < 2.12 MHz).

For the receiver, the requirement for a valid signal is that the duration of the low level signal must be greater than 2 cycles of the IrDA low power baud rate clock.

Figure 23-18 IrDA SIR ENDEC–Block Diagram

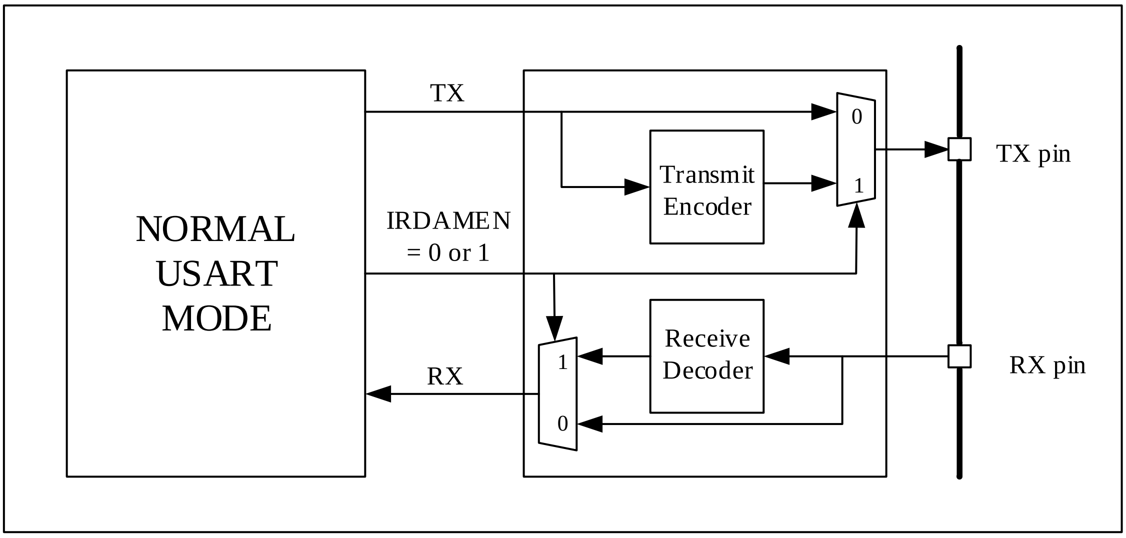
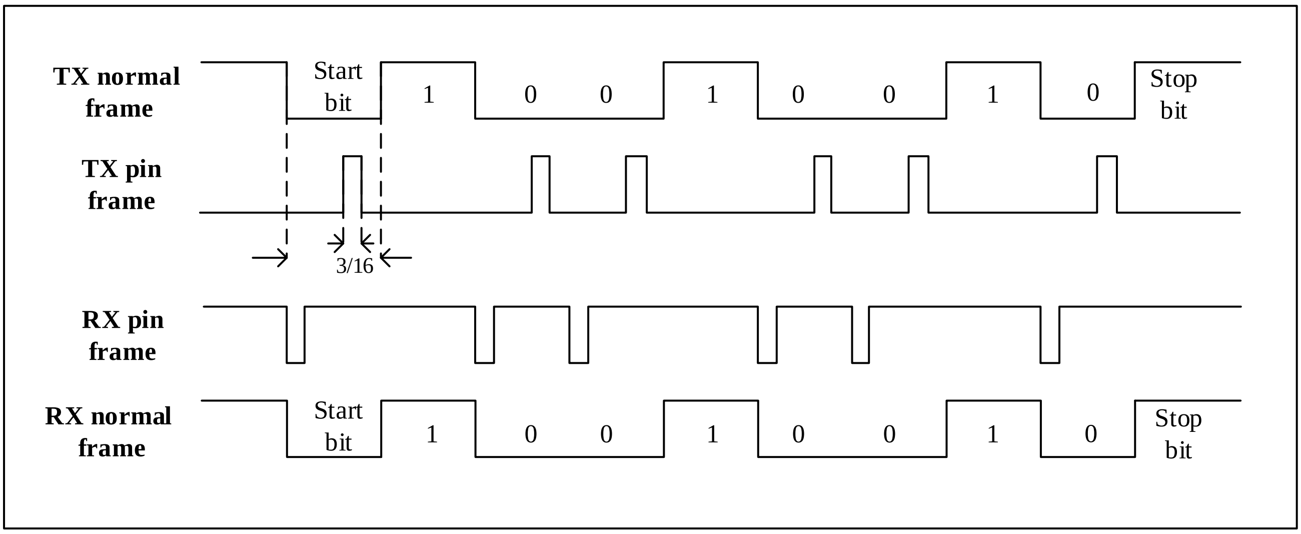


Figure 23-19 IrDA Data Modulation (3/16)-Normal Mode



23.4.16 LIN Mode

USART supports the ability of a LIN(Local interconnection Network) master to send a synchronization break and the ability of a LIN slave to detect a break. LIN mode can be enabled by configuring the USART_CTRL2.LINMEN bit.

Note: When using LIN mode, USART_CTRL2.STPB[1:0], USART_CTRL2.CLKEN, USART_CTRL3.SCMEN, USART_CTRL3.HDMEN, USART_CTRL3. IRDAMEN, these bits should be kept clear.

23.4.16.1 LIN transmission

When LIN is sent, the length of the data bits sent can only be 8 bits. By setting USART_CTRL1.SDBRK, a 13-bit '0' will be sent as the break symbol, and insert a stop bit.

23.4.16.2 LIN reception

Whether the bus is idle or during the transmission of a data frame, as long as the break frame appears, it can be detected. The break symbol detection is independent of the USART receiver.

By configuring the USART_CTRL2.LINBDL bit, 10-bit or 11-bit break character detection can be selected.

When the receiver detects the start bit, the circuit samples each subsequent bit at the 8th, 9th, and 10th oversampling clock points of each bit. When 10 or 11 consecutive bits are detected as '0' and followed by a delimiter, it means that a LIN break is detected, and USART_STS.LINBDF is set. Before confirming the break symbol, check the delimiter as it means the RX line has gone back to high level. An interrupt is generated if the LIN breaker detection interrupt (USART_CTRL2.LINBDIEN) is enabled.

If a '1' is sampled before the 10th or 11th sample point, the current detection is canceled and the start bit is searched again.

Figure 23-20 LIN Break Detection In LIN Mode (11-bit break length-the LINBDL bit is set)

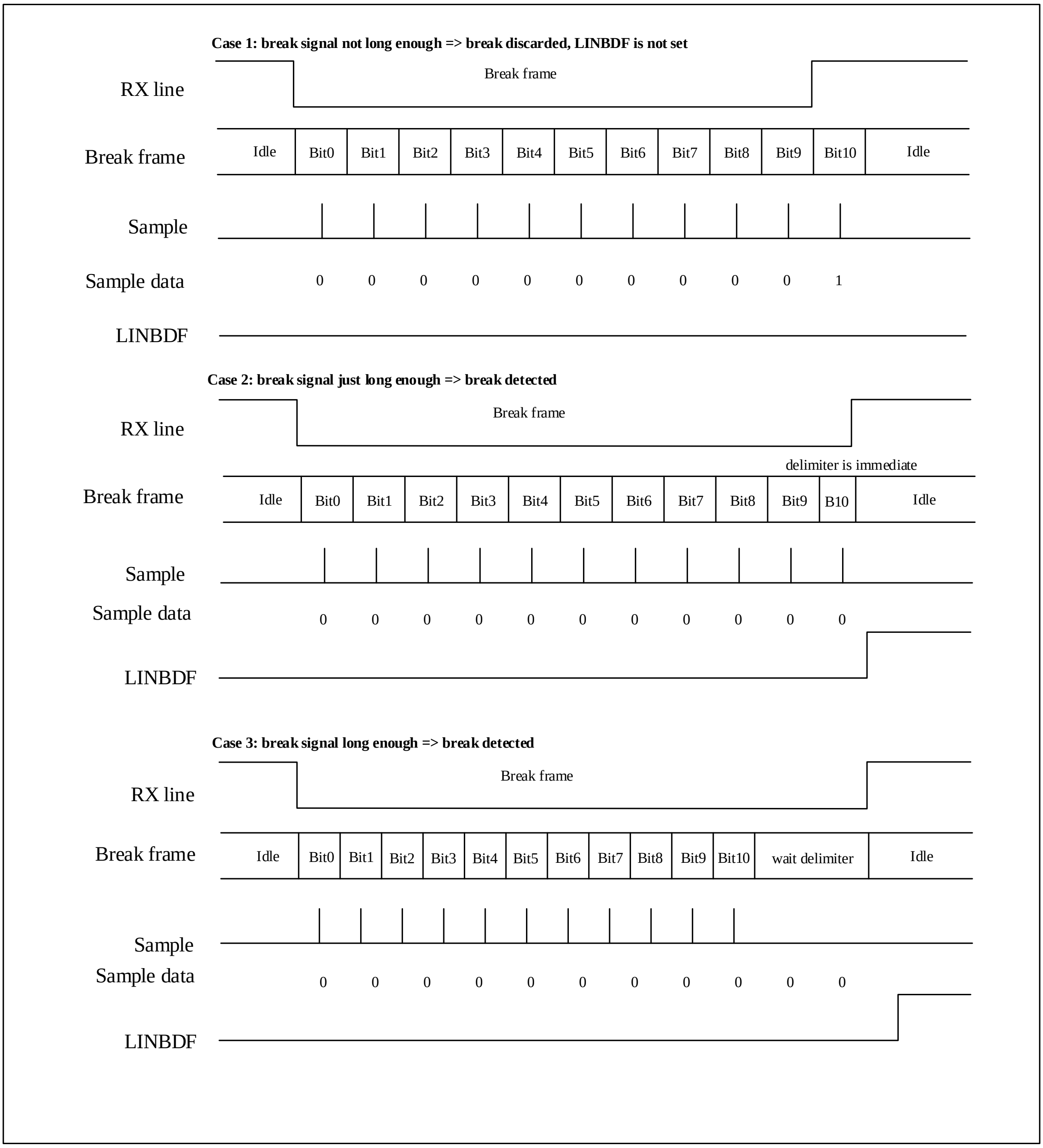
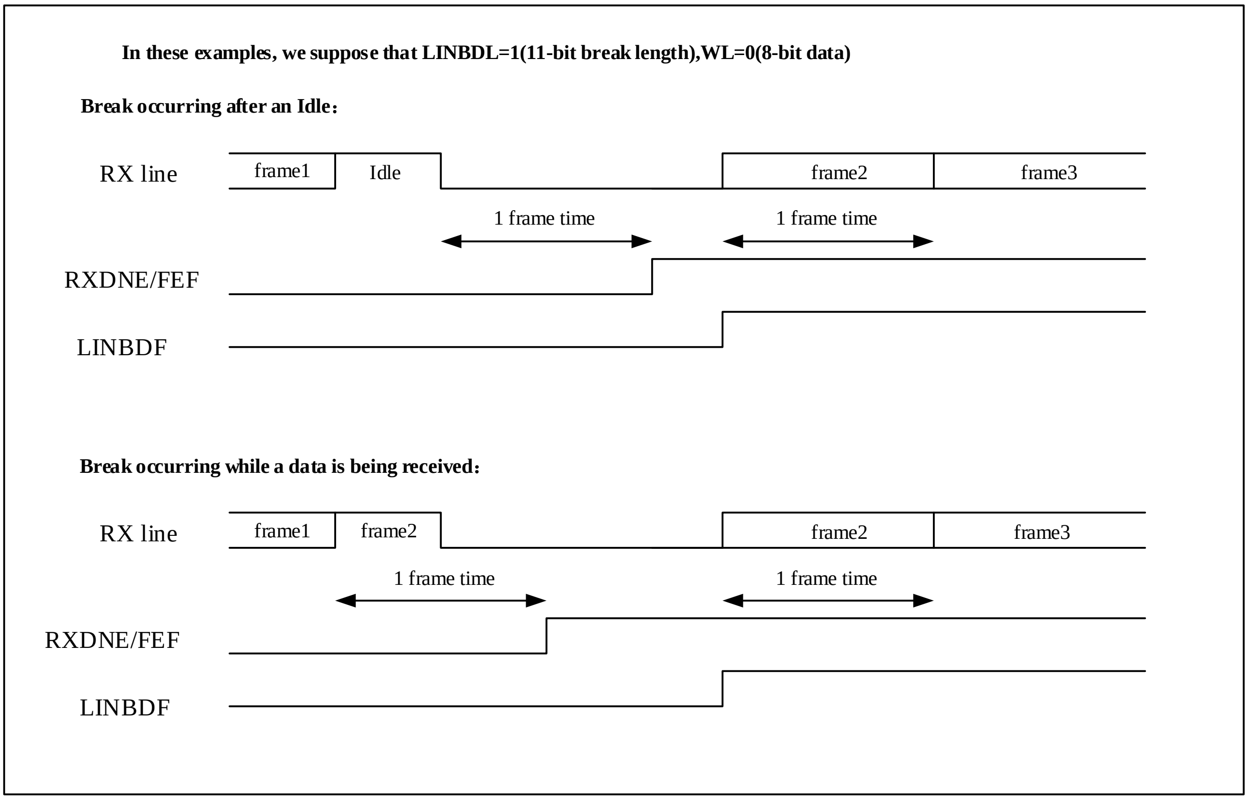


Figure 23-21 Break Detection And Framing Error Detection In LIN Mode


In these examples, we suppose that LINBDL=1(11-bit break length),WL=0(8-bit data)
Break occurring after an Idle:
RX line
frame1
Idle
frame2
frame3
1 frame time
1 frame time
RXDNE/FEF
LINBDF
Break occurring while a data is being received:
RX line
frame1
frame2
frame2
frame3
1 frame time
1 frame time
RXDNE/FEF
LINBDF

23.4.17 Smartcard Mode (ISO7816)

USART supports smartcard protocol. The smartcard interface supports the asynchronous smartcard protocol defined in the ISO7816-3 standard.

Through the USART_CTRL3.SCMEN bit, you can choose whether to enable smartcard mode. When using smartcard mode, USART_CTRL2.LINMEN, USART_CTRL3.HDMEN, USART_CTRL3.IRDAMEN, these bits should be kept clear.

In smartcard mode, the USART can provide a clock through the CK pin. The system clock is divided by the prescaler register to provide the clock to the smartcard. The CK frequency can be from $f_{CK/}2$ to $f_{CK}/62$, where f_{CK} is the peripheral input clock.

In smartcard mode, 0.5 and 1.5 stop bits can be used when receiving data, and only 1.5 stop bits can be used when transmitting data. So 1.5 stop bits are recommended as this avoids configuration transitions.

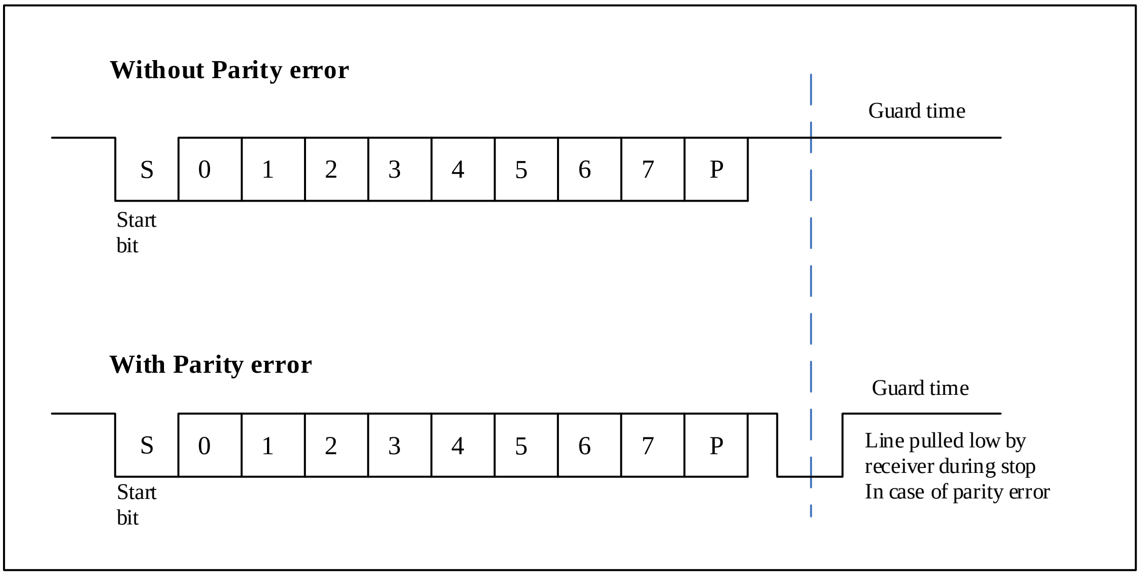
In smartcard mode, the data bits should be configured as 8 bits, and the parity bit should be configured.

When a parity error is detected by receiver, the transmit data line is pulled low for one baud clock cycle at the end of the stop bit as NACK signal(If USART_CTRL3.SCNACK is set). This NACK signal will generate a framing error on the transmitter side (transmitter side is configured with 1.5 stop bits).

When the transmitter receives a NACK signal (framing error) from the receiver, it does not detect the NACK as a start bit (according to the ISO protocol, the duration of the received NACK can be 1 or 2 baud clock cycles).

The example given in the following figure illustrates the signal on the data line with and without parity errors.

Figure 23-22 ISO7816-3 Asynchronous Protocol



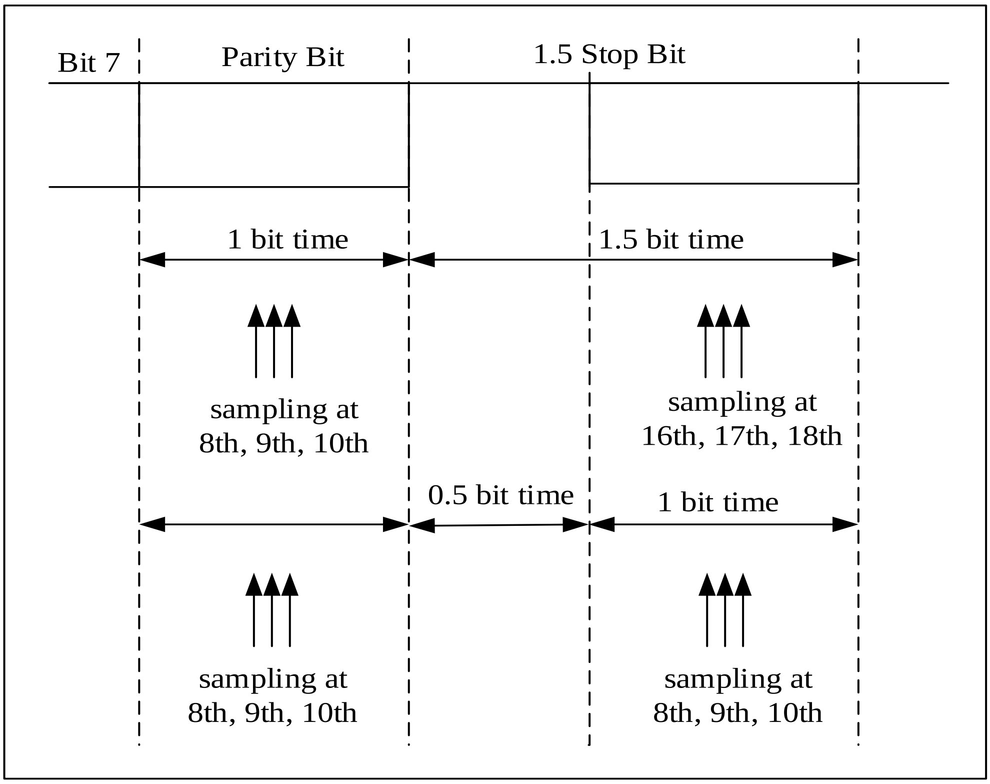
The break frame has no meaning in smartcard mode. A 00h data with a framing error will be treated as data instead of a break symbol.

Under normal operation, data will be shifted out of the transmit shift register on the next baud clock. The smartcard mode is delayed by a minimum of 1/2 baud clock than normal operation.

In normal operation, USART_STS.TXC is set when a frame containing data is sent and USART_STS.TXDE=1.In smartcard mode, the transmission completion flag (USART_STS.TXC) is set high when the guard time counter reaches the value (USART_GTP.GTV[7:0]). The clearing of the USART_STS.TXC flag is not affected by the smartcard mode.

The following figure details how USART samples NACK signals.

Figure 23-23 Use 1.5 Stop Bits To Detect Parity Errors


Bit 7
Parity Bit
1.5 Stop Bit
1 bit time
1.5 bit time
sampling at 8th, 9th, 10th
sampling at 16th, 17th, 18th
0.5 bit time
1 bit time
sampling at 8th, 9th, 10th
sampling at 8th, 9th, 10th

23.5 Interrupt Request

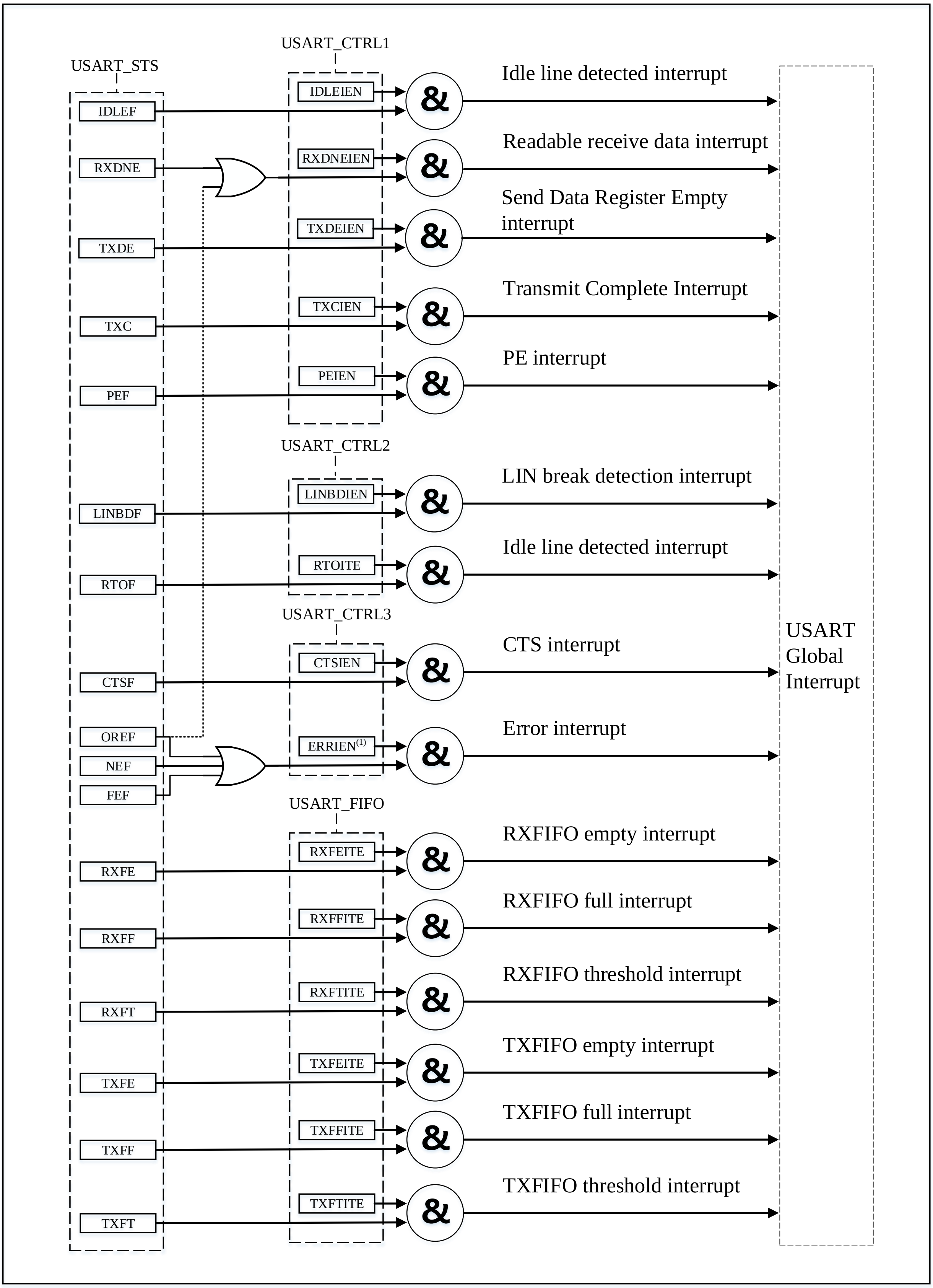
The various interrupt events of USART are logical OR relations, if the corresponding enable control bit is set, these events can generate their own interrupts, but only one interrupt request can be generated at the same time.

Table 23-8 USART USART nterrupt Request

Interrupt function	Interrupt event	Event flag	Enable bit
USART global interrupt	Transmission data register is empty.	TXDE	TXDEIEN
	CTS flag	CTSF	CTSIEN
	Transmission complete	TXC	TXCIEN
	Receive data ready to be read	RXDNE	RXDNEIEN
	Data overrun error detected.	**OREF**	
	Idle line detected	IDLEF	IDLEIEN
	Parity error	PEF	PEIEN
	Disconnect flag	LINBDF	LINBDIEN

	Noise Flag / Overflow Error in Multi-Buffer Communication (DMA) / Frame Error(1)	NEF/OREF/FEF	ERRIEN(1)
	Receiver Timeout	RTOF	RTOITE
	RXFIFO Receive FIFO Interrupt	RXFE	RXFEITE
		RXFF	RXFFITE
		RXFT	RXFTITE
	TXFIFO Transmit FIFO Interrupt	TXFE	TXFEITE
		TXFF	TXFFITE
		TXFT	TXFTITE

FIgure 23-24 USART Interrupt Request



(1) These flags will only generate an error interrupt when receiving data using DMA

(USART_CTRL3.DMARXEN=1).

23.6 Mode Configuration

Table 23-9 USART Mode Setting (1)

Communication mode	USART1	USART2	USART3	USART4	UART5	UART6	UART7	UART8
Asynchronous mode	Y	Y	Y	Y	Y	Y	Y	Y
Multiprocessor	Y	Y	Y	Y	Y	Y	Y	Y
LIN	Y	Y	Y	Y	Y	Y	Y	Y
Synchronous mode	Y	Y	Y	Y	N	N	N	N
Single-wire half duplex mode	Y	Y	Y	Y	Y	Y	Y	Y
Smartcard mode	Y	Y	Y	Y	N	N	N	N
IrDA infrared mode	Y	Y	Y	Y	Y	Y	Y	Y
DMA communication mode	Y	Y	Y	Y	Y	Y	Y	Y
Hardware flow control mode	Y	Y	Y	Y	Y	Y	Y	Y

(1) Y = support this mode, N = do not support this mode.

23.7 UART Registers

23.7.1 UART Register Overview

Table 23-10 UART Register Overview

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	USART_CTRL1	Reserved			IFCEN	SWAP	OVER8	DEAT					DEDT					DEP	DEM	SDBRK	PEIEN	TXCIEN	TXDEIEN	RXDNEIEN	IDLEIEN	WUM	RCVWU	WL	PCEN	PSEL	TXEN	RXEN	UEN
	Reset value				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x04	USART_CTRL2	Reserved											FEFLOSE	NEFLOSE	PEFLOSE	RTOITE	RTOCF	RTOEN	LINBDL	LINBDIEN	LINMEN	LBCLK	CLKPHA	CLKPOL	CLKEN	Reserved	STPB		Reserved	ADDR			
	Reset value												0	0	0	0	0	0	0	0	0	0	0	0	0		0	0		0	0	0	0
0x08	USART_CTRL3	Reserved																					SCNACK	SCMEN	IRDALP	IRDAMEN	ERRIEN	DMARXEN	DMATXEN	HDMEN	RTSEN	CTSIEN	CTSEN
	Reset value																						0	0	0	0	0	0	0	0	0	0	0

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x0C	USART_STS	Reserved												FELOSEF	NELOSEF	PELOSEF	RTOF	FEF	NEF	OREF	PEF	LINBDF	CTSF	RXDNE	TXC	TXDE	IDLEF	TXFT	RXFT	RXFE	TXFE	RXFF	TXFF
	Reset value													0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0
0x10	USART_DAT	Reserved																							DATV								
	Reset value																								0	0	0	0	0	0	0	0	0
0x14	USART_BRCF	Reserved																DIV_Integer												DIV_Decimal			
	Reset value																	h	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x18	USART_GTP	Reserved																GTV								PSCV							
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x1C	USART_FIFO	Reserved										TXCNT				RXCNT				TXFTITE	RXFTITE	RXFEITE	TXFEITE	RXFFITE	TXFFITE	RXFTCFG			TXFTCFG			CLR	EN
	Reset value											0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x20	USART_IFW	Reserved																WIDTH															
	Reset value																	h	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x24	USART_RTO	Reserved				TIME																											
	Reset value					h	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.7.2 USART Control Register 1 Register (USART_CTRL1)

Address offset : 0x00

Reset value : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved			IFCEN	SWAP	OSPM	DEAT					DEDT				
			rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DEP	DEM	SDBRK	PEIEN	TXC IEN	TXDE IEN	RXDNE IEN	IDLE IEN	WUM	RCVWU	WL	PCEN	PSEL	TXEN	RXEN	UEN
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit Field	Name	Description
31:29	Reserved	Reserved, the reset value must be maintained
28	IFCEN	Idle frame controllable enable. 0: Disable idle frame controllable adjustment, idle frame length is consistent with the number of data bits; 1: Enable idle frame controllable adjustment, idle frame length is controlled by USART_IFW. *Note: This bit can only be written when USART is disabled (UEN='0').*
27	SWAP	Swap TX/RX pins.

Bit Field	Name	Description
		This bit is set to 1 and cleared by software. 0: Use TX/RX pins as defined in standard pinout. 1: Swap the functions of TX and RX pins. Allows operation in cross-connection with another USART. *Note: This bit can only be written when USART is disabled (UEN='0').*
26	OSPM	Oversampling mode 0: x16 Oversampling 1: x8 Oversampling *Notes: 1. This bit can only be written when USART is disabled (UEN='0').* *2.In LIN, IrDA, and smart card modes, this bit must remain cleared.*
25:21	DEAT	Driver Enable assertion time. These 5 bits are used to define the time between activating the DE (Driver Enable) signal and the start of the start bit. This time is represented in sampling time units (1/8 or 1/16 of a bit time, depending on the oversampling rate). This bit field can only be written when USART is disabled (UEN='0'). *Note: After enabling USART_CTRL1.DEM, this bit cannot be configured as 0 (the minimum configuration is 1).*
20:16	DEDT	Driver Enable deassertion time. These 5 bits are used to define the time between the end of the last stop bit in the transmitted message and the deactivation of the DE (Driver Enable) signal. This time is represented in sampling time units (1/8 or 1/16 of a bit time, depending on the oversampling rate). If a write operation is performed on the USART_TDR register within the DEDT time, the new data will only be sent after the DEDT and DEAT times have elapsed. *Note: This bit can only be written when USART is disabled (UEN='0').*
15	DEP	Driver enable polarity selection 0: DE signal active high. 1: DE signal active low. *Note: This bit can only be written when USART is disabled (UEN='0').*
14	DEM	Driver enable mode. This bit is used to activate external transceiver control via the DE signal (DE signal output on the RTS pin). 0: Disable DE function. 1: Enable DE function. *Note: This bit can only be written when USART is disabled (UEN='0').*
13	SDBRK	Send Break Character. The software transmits a break character by setting this bit to 1. This bit is cleared by hardware during stop bit of the break frame transmission. 0: No break character was sent. 1: Send a break character.
12	PEIEN	PE interrupt enable If this bit is set to 1, an interrupt is generated when USART_STS.PEF bit is set.

Bit Field	Name	Description
		0: Parity error interrupt is disabled. 1: Parity error interrupt is enabled.
11	TXCIEN	Transmit complete interrupt enable. If this bit is set to 1, an interrupt is generated when USART_STS.TXC is set. 0: Transmission completion interrupt is disabled. 1: Transmission completion interrupt is enabled.
10	TXDEIEN	TXDE interrupt enable If this bit is set to 1, an interrupt is generated when USART_STS.TXDE bit is set. 0: Send buffer empty interrupt is disabled. 1: Send buffer empty interrupt is enabled.
9	RXDNEIEN	RXDNE interrupt enable If this bit is set to 1, an interrupt is generated when USART_STS.RXDNE or USART_STS.OREF is set. 0: Data buffer non-empty interrupt o and overrun error interrupt are disabled. 1: Data buffer non-empty interrupt o and overrun error interrupt are enabled.
8	IDLEIEN	IDLE interrupt enable. If this bit is set to 1, an interrupt is generated when USART_STS.IDLEF is set. 0:IDLE line detection interrupt is disabled. 1: IDLE line detection interrupt is enabled.
7	WUM	Wake up mode from mute mode. 0: Idle frame wake up. 1: Address identifier wake up.
6	RCVWU	The receiver wakes up Software can set this bit to 1 to make USART enter mute mode, and clear this bit to 0 to wake up USART. In idle frame wake-up mode (USART_CTRL1.WUM=0), this bit is cleared by hardware when an idle frame is detected. In address wake-up mode (USART_CTRL1.WUM=1), when an address matching frame is received, this bit is cleared by hardware. Or when an address mismatch frame is received, it is set to 1 by hardware. 0: The receiver is in normal operation mode. 1: The receiver is in mute mode.
5	WL	Word length. 0:8 data bits. 1:9 data bits. *Note: If data is in transit, this bit cannot be configured.*
4	PCEN	Parity control enable 0: Parity control is disabled. 1: Parity control is enabled.
3	PSEL	Parity selection. 0: even check. 1: odd check.
2	TXEN	Transmitter enable.

Bit Field	Name	Description
		0: The transmitter is disabled. 1: the transmitter is enabled.
1	RXEN	Receiver enable 0: The receiver is disabled. 1: the receiver is enabled.
0	UEN	USART enable When this bit is cleared, the divider and output of USART stop working after the current byte transmission is completed to reduce power consumption. Software can set or clear this bit. 0:USART is disabled. 1:USART is enabled.

23.7.3 USART Control Register 2 Register (USART_CTRL2)

Address offset : 0x04

Reset value : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved											FEFLOSE	NEFLOSE	PEFLOSE	RTOITE	RTOCF
											rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTOEN	LINBDL	LINBDIEN	LINMEN	LBCLK	CLKPHA	CLKPOL	CLKEN	Reserved	STPB		Reserved	ADDR			
rw	rw	rw	rw	rw	rw	rw	rw		rw	rw		rw	rw	rw	rw

Bit Field	Name	Description
31:21	Reserved	Reserved, the reset value must be maintained
20	FEFLOSE	FEF Data Discard Enable Bit. Enabling this bit prevents data from being written into the FIFO when a framing error occurs during data reception in FIFO mode, and no FEF flag will be generated. 1: Enable 0: Disable
19	NEFLOSE	NEF Data Discard Enable Bit. Enabling this bit prevents data from being written into the FIFO when a noise error occurs during data reception in FIFO mode, and no NEF flag will be generated. 1: Enable 0: Disable
18	PEFLOSE	PEF Data Discard Enable Bit. Enabling this bit prevents data from being written into the FIFO when a framing error occurs during data reception in FIFO mode, and no PEF flag will be generated. 1: Enable

Bit Field	Name	Description
		0: Disable
17	RTOITE	Receiver timeout interrupt enable. This bit is set to 1 and cleared by software. 0: Disable interrupt 1: Generate USART interrupt when RTOF bit in USART_STS register is set to 1.
16	RTOCF	Receiver timeout clear flag. When writing '1' to this bit, the RTOF flag in the USART_STS register will be cleared.
15	RTOEN	Receiver timeout enable. This bit is set to 1 and cleared by software. 0: Disable receiver timeout function. 1: Enable receiver timeout function. Upon enabling this function, if the RX line remains idle (no reception) for the programmed duration in RTOR (Receiver Timeout Register), the RTOF flag in the USART_ISR register is set to 1.
14	LINBDL	LIN break detection length. This bit is used to set the length of the break frame. 0:10 bit break detection 1:11 bit break detection *Note: LINBDL can be used to control the detection length of Break Characters in LIN mode and other modes, and the detection length is the same as that in LIN mode.*
13	LINBDIEN	LIN break detection interrupt enable. If this bit is set to 1, an interrupt will be generated when USART_STS.LINBDF bit is set. 0: Disconnect signal detection interrupt is disabled. 1: Turn-off signal detection interrupt enabled
12	LINMEN	LIN mode enable 0:LIN mode is disabled 1:LIN mode enabled
11	LBCLK	The Last bit clock pulse. This bit is used to set whether the clock pulse corresponding to the last transmitted data byte (MSB) is output on CK pin in synchronous mode. 0: The clock pulse of the last bit of data is not output from CK. 1: The clock pulse of the last bit of data will be output from CK. *Note: This bit cannot be used for UART5/6/7/8.*
10	CLKPHA	Clock phase. This bit is used to set the phase of CK pin in synchronous mode. 0: Sample the first data at the first clock edge. 1: Sample the first data at the second clock edge. *Note: This bit cannot be used for UART5/6/7/8.*
9	CLKPOL	Clock polarity.

Bit Field	Name	Description
		This bit is used to set the polarity of CK pin in synchronous mode. 0: CK pin remains low when it is not transmitted to the outside. 1: CK pin remains high when it is not sent to the outside. *Note: This bit cannot be used for UART5/6/7/8.*
8	CLKEN	Clock enable 0:CK pin is disabled 1:CK pin enabled *Note: This bit cannot be used for UART5/6/7/8.*
7	Reserved	Reserved, the reset value must be maintained
6:5	STPB[1:0]	STOP bits. 00:1 stop bit. 01:0.5 stop bit. 10:2 stop bit. 11:1.5 stop bit. *Note: For UART5/6/7/8, only one stop bit and two stop bits are valid.*
4	Reserved	Reserved, the reset value must be maintained
3:0	ADDR[3:0]	USART address. Used in the mute mode of multiprocessor communication, using address identification to wake up a USART device. In address wake-up mode (USART_CTRL1.WUM=1), if the lower four bits of the received data frame are not equal to the ADDR[3:0] value, USART will enter the mute mode; If the lower four bits of the received data frame are equal to the ADDR[3:0] value, USART will be awakened.

23.7.4 USART Control Register 3 Register (USART_CTRL3)

Address offset : 0x08

Reset value : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved					SCNACK	SCMEN	IRDALP	IRDAMEN	ERRIEN	DMARXEN	DMATXEN	HDMEN	RTSEN	CTSIEN	CTSEN
					rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit Field	Name	Description
31:11	Reserved	Reserved, the reset value must be maintained
10	SCNACK	Smartcard NACK enable. This bit is used for Smartcard mode to enable transmitting NACK when parity error

Bit Field	Name	Description
		occurs. 0: Do not send NACK when there is a parity error. 1: send NACK when there is a parity error. *Note: This bit cannot be used for UART5/6/7/8*
9	SCMEN	Smartcard mode enable. This bit is used to enable Smartcard mode. 0: Smartcard mode is disabled. 1: Smartcard mode is enabled. *Note: This bit cannot be used for UART5/6/7/8*
8	IRDALP	IrDA low-power mode. This bit is used to select the low power consumption mode for IrDA mode. 0: Normal mode. 1: Low power mode.
7	IRDAMEN	IrDA mode enable. 0:IrDA is disabled. 1:IrDA is enabled.
6	ERRIEN	Error interrupt enable. When DMA receive mode (USART_CTRL3.DMARXEN=1) is enabled, an interrupt will be generated when USART_STS.FEF, USART_STS. OREF or USART_STS. NEF bit is set. 0: Error interrupt is disabled. 1: Error interrupt enabled.
5	DMARXEN	DMA receiver enable. 0:DMA receive mode is disabled. 1:DMA receive mode is enabled.
4	DMATXEN	DMA transmitter enable. 0:DMA transmission mode is disabled. 1:DMA transmission mode is enabled.
3	HDMEN	Half-duplex mode enable. This bit is used to enable half-duplex mode. 0: Half-duplex mode is disabled. 1: Half-duplex mode is enabled.
2	RTSEN	RTS enable. This bit is used to enable RTS hardware flow control function. 0:RTS hardware flow control is disabled. 1:RTS hardware flow control is enabled. *Note: This bit cannot be used for UART5/6/7/8*
1	CTSIEN	CTS interrupt enable. If this bit is set to 1, an interrupt will be generated when USART_STS.CTSF bit is set. 0:CTS interrupt is disabled. 1:CTS interrupt is enabled. *Note: This bit cannot be used for UART5/6/7/8*

Bit Field	Name	Description
0	CTSEN	CTS enable. This bit is used to enable the CTS hardware flow control function. 0:CTS hardware flow control is disabled. 1:CTS hardware flow control is enabled. *Note: This bit cannot be used for UART5/6/7/8*

23.7.5 USART Status Register (USART_STS)

Address offset : 0x0C

Reset value : 0x0000 0180

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved												FELOSEF	NELOSEF	PELOSEF	RTOF
												rc_w1	rc_w1	rc_w1	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FEF	NEF	OREF	PEF	LINBDF	CTSF	RXDNE	TXC	TXDE	IDLEF	TXFT	RXFT	RXFE	TXFE	RXFF	TXFF
r	r	r	r	rc_w0	rc_w0	rc_w0	rc_w0	r	r	r	r	r	r	r	r

Bit Field	Name	Description
31:20	Reserved	Reserved, the reset value must be maintained
19	FELOSEF	Received Data FE Error Discard Flag. In the FIFO mode, when an FE error is received, indicating that erroneous data is discarded and not written into the FIFO. Writing a 1 to this bit clears it. In non-FIFO mode, this bit is 0. 1: Discard data with FE error; 0: No data FE error
18	NELOSEF	Received Data NE Error Discard Flag. In the FIFO mode, when an NE error is received, indicating that erroneous data is discarded and not written into the FIFO. Writing a 1 to this bit clears it. In non-FIFO mode, this bit is 0. 1: Discard data with NE error; 0: No data NE error
17	PELOSEF	Received Data PE Error Discard Flag Bit. In the FIFO mode, when a PE error is received, indicating that erroneous data is discarded and not written into the FIFO. Writing a 1 to this bit clears it. In non-FIFO mode, this bit is 0. 1: Discard data with PE error; 0: No data PE error
16	RTOF	Recevier Timeout. After the programmed timeout value in the RTO register has elapsed and there is no

Bit Field	Name	Description
		communication, this bit is set to 1 by hardware. This bit is cleared by software by writing a '1' to the USART_CTRL2.RTOCF bit. If RTOITE in the USART_CTRL2 register is set to '1', an interrupt is generated. 0: Timeout value not reached, 1: Timeout value reached, no data received
15	FEF	Framing error. When the data is not synchronized or a large amount of noise is detected, and the stop bit is not received and recognized at the expected time, it will be judged that a framing error has been detected, and this bit will be set to 1. First read USART_STS, then read USART_DAT can cleared this bit. 0: No framing errors were detected. 1: A framing error or a Break Character is detected. *Note: this bit will not generate an interrupt because it appears with USART_STS.RXDNE, and the hardware will generate an interrupt when setting the USART_STS.RXDNE flag. If the currently transmitted data has both framing errors and overload errors, the hardware will continue to transmit the data and only set the USART_STS.OREF flag bit.* *In the multi-buffer communication mode, if the USART_CTRL3.ERRIEN bit is set, an interrupt will be generated when the FEF flag is set.*
14	NEF	Noise error flag. When noise is detected in the received frame, this bit is set by hardware. It is cleared by the software sequence (read first USART_STS, read USART_DAT again). 0: No noise error detected. 1: Noise error detected. *Note: this bit will not generate an interrupt because it appears with USART_STS.RXDNE, and the hardware will generate an interrupt when setting the USART_STS.RXDNE flag. In the multi-buffer communication mode, if the USART_CTRL3.ERRIEN bit is set, an interrupt will be generated when the NEF flag is set.*
13	OREF	Overrun error The placement of overflow errors is specified in section 23.4.4.7 Overflow Errors. The software can clear this bit by reading USART_STS first and then reading USART_DAT. 0: No overrun error was detected. 1: Overflow error detected. *Note: After OREF is set, UARD_DAT will not update data again; If RXDNE is 0 at this time, it will not reset to 1 because the data is no longer updated.*
12	PEF	Parity error. This bit is set when the parity bit of the received data frame is different from the expected check value. The software can clear this bit by reading USART_STS first and then reading USART_DAT.

Bit Field	Name	Description
		0: No parity error was detected. 1: Parity error detected.
11	LINBDF	LIN break detection flag. If USART_CTRL2.LINMEN bit is set, this bit is set by hardware when LIN disconnection is detected. If USART_CTRL2.LINBDIEN bit is set, an interrupt will be generated. This bit is cleared by software. 0: LIN break character not detected. 1: LIN break character detected.
10	CTSF	CTS flag If USART_CTRL3.CTSEN bit is set, this bit is set by hardware when the nCTS input changes. If USART_CTRL3.CTSIEN bit is set, an interrupt will be generated. This bit is cleared by software. 0:nCTS status line has not changed. 1:nCTS status line changes.
9	RXDNE	The Read data register not empty. This bit is set when the read data buffer receives data from the shift register. When USART_CTRL1.RXDNEIEN bit is set, an interrupt will be generated. Software can clear this bit by writing 0 to it or reading the USART_DAT register. 0: The read data buffer is empty. 1: The read data buffer is not empty.
8	TXC	Transmission complete. This bit is set to 1 after power-on reset. If USART_STS.TXDE is set, this bit is set when the current data transmission is completed. Setting USART_CTRL1.TXCIEN bit will generate an interrupt. This bit is cleared by software. 0: Transmitting did not complete. 1: Send completed.
7	TXDE	The Transmit data register empty. Set to 1 after power-on reset or data to be sent has been sent to the shift register. Setting USART_CTRL1.TXDEIEN will generate an interrupt. This bit is cleared to 0 when the software writes the data to be sent into USART_DAT. 0: Send data buffer is not empty. 1: The transmitting data buffer is empty.
6	IDLEF	IDLE line detected flag. Within one frame time, the idle state is detected at the RX pin, and this bit is set to 1. When USART_CTRL1.IDLEIEN bit is set, an interrupt will be generated. The software can clear this bit by reading USART_STS first and then reading USART_DAT. 0: No idle frame detected. 1: idle frame detected. *Note: IDLEF bit will not be set high again until USART_STS.RXDNE bit is set (that*

Bit Field	Name	Description
		is, an idle line is detected again).
5	TXFT	TX FIFO threshold. 0: Number of data in the TX FIFO has not reached the threshold. 1: Number of data in the TX FIFO has reached the threshold.
4	RXFT	RX FIFO threshold 0: Number of data in the RX FIFO has not reached the threshold. 1: Number of data in the RX FIFO has reached the threshold.
3	RXFE	Receive FIFO empty 0: Receive FIFO data is not empty. 1: Receive FIFO data is empty.
2	TXFE	Send FIFO empty 0: Send FIFO data is not empty. 1: Send FIFO data is empty.
1	RXFF	Receive FIFO full 0: Receive FIFO data is not full. 1: Receive FIFO data is full.
0	TXFF	Send FIFO full 0: Send FIFO data is not full. 1: Send FIFO data is full.

23.7.6 USART Data Register (USART_DAT)

Address offset : 0x10

Reset value : undefined (uncertain value)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved							DATV[8:0]								
							rw								

Bit Field	Name	Description
31:9	Reserved	Reserved, the reset value must be maintained
8:0	DATV[8:0]	Data value Contains the data sent or received; Software can change the transmitted data by writing these bits, or read the values of these bits to obtain the received data. If parity is enabled, when the transmitted data is written into the register, the highest bit of the data (the 7th or 8th bit depends on USART_CTRL1.WL bit) will be replaced by the parity bit.

23.7.7 USART Baud Rate Register (USART_BRCF)

Address offset : 0x14

Reset value : 0x0000 0000

Note: When USART_CTRL1.UEN=1, this register cannot be written;The baud counter stops counting if USART_CTRL1.TXEN or USART_CTRL1.RXEN are disabled respectively.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DIV_Integer[11:0]												DIV_Decimal[3:0]			
rw												rw			

Bit Field	Name	Description
31:16	Reserved	Reserved, the reset value must be maintained
15:4	DIV_Integer [11:0]	Integer part of baud rate divider.
3:0	DIV_Decimal[3:0]	Fractional part of baud rate divider. *Note: Under 8x oversampling, only the lower three bits of DIV_Decimal[3:0] are valid.*

23.7.8 USART Guard Time And Prescaler Register (USART_GTP)

Address offset : 0x18

Reset value : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GTV[7:0]								PSCV[7:0]							
rw								rw							

Bit Field	Name	Description
31:16	Reserved	Reserved, the reset value must be maintained
15:8	GTV[7:0]	Guard time value in Smartcard mode. This Bit Field specifies the guard time in baud clock. In Smartcard mode, this function is required. The setting time of USART_STS.TXC flag is delayed by GTV[7:0] baud

Bit Field	Name	Description
		clock cycles. *Note: This bit is invalid for UART5/6/7/8.*
7:0	PSCV[7:0]	Prescaler value. In IrDA low power consumption mode: these bits are used to set the frequency division coefficient for dividing the peripheral clock (PCLK1/PCLK2) to generate low power consumption frequency. 00000000: reserved-do not write this value. 00000001: divide the source clock by 1. … 11111111: divide the source clock by 255. In IrDA normal mode: PSCV can only be set to 00000001. In Smartcard mode: PSCV[4:0] is used to set the frequency division of Smartcard clock generated by peripheral clock (PCLK1/ PCLK2). Coefficient. The actual frequency division coefficient of is twice the set value of PSCV[4:0]. 0000: reserved-do not write this value. 0001: Divide the source clock by 2. 0010: Divide the source clock by 4. … 1111: Divide the source clock by 62. In Smartcard mode, PSCV[7:5] is reserved.

23.7.9 USART FIFO Register (USART_FIFO)

Address offset : 0x1C

Reset value : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved										TXCNT				RXCNT	
										r				r	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RXCNT		TXFTITE	RXFTITE	RXFEITE	TXFEITE	RXFFITE	TXFFITE	RXFTCFG			TXFTCFG			CLR	EN
r		rw	rw	rw	rw	rw	rw	rw			rw			rw	rw

Bit Field	Name	Description
31:22	Reserved	Reserved, the reset value must be maintained
21:18	TXCNT	number of TXFIFO valid data
17:14	RXCNT	number of RXFIFO valid data

Bit Field	Name	Description
13	TXFTITE	TXFIFO threshold interrupt enable If this bit is set to 1, an interrupt is generated when USART_STS.TXFT is set. 0: Disable TXFT interrupt. 1: Enable TXFT interrupt.
12	RXFTITE	RXFIFO threshold interrupt enable If this bit is set to 1, an interrupt is generated when USART_STS.RXFT is set. 0: Disable RXFT interrupt. 1: Enable RXFT interrupt.
11	RXFEITE	RXFIFO empty interrupt enable. If this bit is set to 1, an interrupt is generated when USART_STS.RXFE is set. 0: Disable RXFE interrupt. 1: Enable RXFE interrupt.
10	TXFEITE	TXFIFO empty interrupt enable. If this bit is set to 1, an interrupt is generated when USART_STS.TXFE is set. 0: Disable TXFE interrupt. 1: Enable TXFE interrupt.
9	RXFFITE	RXFIFO full interrupt enable. If this bit is set to 1, an interrupt is generated when USART_STS.TXFF is set. 0: Disable RXFF interrupt. 1: Enable RXFF interrupt.
8	TXFFITE	TXFIFO full interrupt enable. If this bit is set to 1, an interrupt is generated when USART_STS.TXFF is set. 0: Disable TXFF interrupt. 1: Enable TXFF interrupt.
7:5	RXFTCFG	RXFIFO threshold configuration. 000: Receive FIFO reaches 1/8 of its depth 001: Receive FIFO reaches 2/8 of its depth 010: Receive FIFO reaches 4/8 of its depth 011: Receive FIFO reaches 6/8 of its depth 100: Receive FIFO reaches 7/8 of its depth 101: Receive FIFO is full All other combinations: Reserved
4:2	TXFTCFG	TXFIFO threshold configuration. 000: TXFIFO reaches 1/8 of its depth 001: TXFIFO reaches 1/4 of its depth 010: TXFIFO reaches 1/2 of its depth 011: TXFIFO reaches 3/4 of its depth 100: TXFIFO reaches 7/8 of its depth 101: TXFIFO becomes empty All other combinations: Reserved
1	CLR	Clear data and pointers in the FIFO. This bit is a pulse signal, writing 1 clears it, and CLR is also automatically cleared.

Bit Field	Name	Description
		0: No clear. 1: Clear.
0	EN	FIFO model enable. 0: FIFO mode disabled. 1: FIFO mode enabled.

23.7.10 USART Idle Frame Width Register (USART_IFW)

Address offset : 0x20

Reset value : 0x0000 0000

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="16">Reserved</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="16">WIDTH</td></tr>
<tr><td colspan="16">rw</td></tr>
</table>

Bit Field	Name	Description
31:16	Reserved	Reserved, the reset value must be maintained
15:0	WIDTH	Idle Frame Width *Note: In units of baud rate time*

23.7.11 USART Receive Timeout Width Register (USART_RTO)

Address offset : 0x24

Reset value : 0x0000 0000

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="4">Reserved</td><td colspan="12">TIME</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="16">TIME</td></tr>
<tr><td colspan="16">rw</td></tr>
</table>

Bit Field	Name	Description
31:28	Reserved	Reserved, the reset value must be maintained
27:0	TIME	Receiver Timeout Value

Bit Field	Name	Description
		Note: In units of baud rate time

24 Internal Integrated Circuit Bus (I2C)

24.1 Introduction

The I^{2}C (Inter-Integrated Circuit) bus is a widely used bus structure that consists of only two bidirectional lines, namely the data line SDA and the clock line SCL. Through these two lines, all devices compatible with the I^{2}C bus can communicate directly with each other via the I^{2}C bus.

I^{2}C interface connects the microcontroller and the serial I^{2}C bus, which can be used for communication between the MCU and external I^{2}C devices. The I^{2}C interface module implements standard speed and fast mode of the I^{2}C protocol, with CRC calculation and verification functions, supporting SMBus (System Management Bus) and PMBus (Power Management Bus). Additionally, it provides multi-master functionality, controlling all I^{2}C bus-specific timing, protocol, and arbitration. The I^{2}C interface module also supports DMA mode, effectively reducing the burden on the CPU.

24.2 Main Features

- Can implement both master and slave functions on the same interface
- Acts as a converter from parallel bus to I^{2}C bus protocol
- Supports 7-bit and 10-bit address modes and broadcast addressing
- As an I^{2}C master device, can generate clock, start, and stop signals
- As an I^{2}C slave device, has programmable I^{2}C address detection and stop bit detection functions
- Supports standard speed (up to 100kHz), fast speed (up to 400kHz), and fast mode+ (up to 1MHz) modes
- Supports interrupt vectors, byte transfer success interrupts, and error event interrupts
- Optional clock stretching function
- Supports DMA mode
- Optional PEC (Packet Error Checking) generation and verification
- Compatible with SMBus 2.0 and PMBus
- Supports FIFO mode

Note: Not all products include all the features mentioned above. Please refer to the relevant datasheet to confirm the I^{2}C functionalities supported by the product.

24.3 Function Description

The I^{2}C interface is connected to I^{2}C bus through data pin (SDA) and clock pin (SCL) to communicate with external devices. It can be connected to standard (up to 100kHz) or fast (up to 400kHz) or fast$^+$ (up to 1MHz) I^{2}C bus. I^{2}C module converts data from serial to parallel when receiving, and converts data from parallel to serial when transmitting. It supports interrupt mode, users can enable or disable interrupt according to their needs.

24.3.1 SDA and SCL Control

I^{2}C module has two interface lines: serial data line (SDA) and serial clock line (SCL). Devices connected to the bus and exchange information through these two wires. Both SDA and SCL are bidirectional lines, connected to positive power supply with a pull-up resistor. When the bus is idle, both lines are high level. The output of device which is connected to the bus must have open drain or open collector to provide wired-AND functionality. The data on I^{2}C bus can reach 100 kbit/s in standard mode and 1000 kbit/s in fast mode. Since devices of different processors may be connected to the I2C bus, the levels of logic '0' and logic '1' are not fixed and depend on the actual level of V_{DD}.

If the clock stretchingis allowed, the SCL line is pulled down which can be avoided the overload error during receiving and the under load error during transmission.

For example, when in the transmission mode, if the transmit data register is empty and the byte transmit end bit is set (I2C_STS1.TXDATE = 1, I2C_STS1.BSF = 1), the I^{2}C interface keeps the clock line low before transmission to wait for the software to read STS1 and write the data into the data register (both buffer and shift register are empty); when In the receive mode, if the data register is not empty and the byte sending end bit is set (I2C_STS1.RXDATNE = 1, I2C_STS1.BSF = 1), the I^{2}C interface keeps the clock line low after receiving the data byte, waiting for the software to read STS1, and then read the data register(buffer and shift register are full).

If clock stretchingis disable in slave mode, if the receive data register is not empty (I2C_STS1.RXDATNE = 1) in the receive mode, and the data has not been read before receiving the next byte, an overrun error will issue and the last word byte will be discarded. In transmit mode, if the transmit data register is empty (I2C_STS1.TXDATE = 1), no new data is written into the data register before the next byte must be sent, an underrun error will issue. The same byte will be send repeatedly. In this case, duplicate write conflicts are not controlled.

24.3.2 21.3.2 Software Communication Process

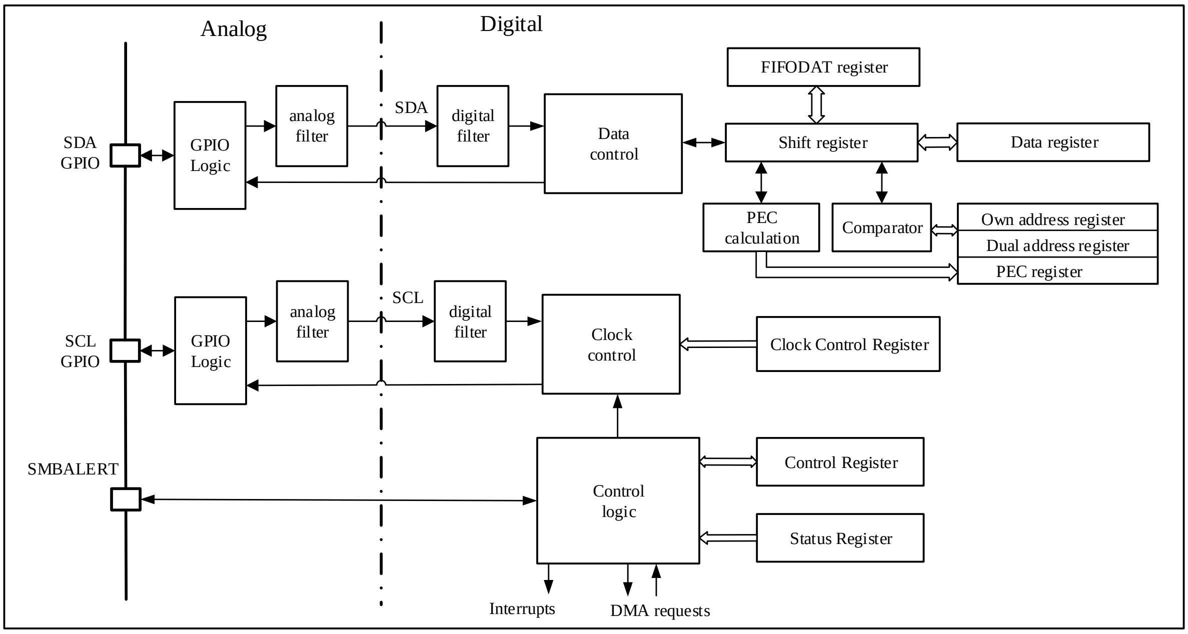
The data transmission of I^{2}C device is divided into master and slave. Master is the device responsible for initializing the transmission of data on the bus and generating clock signal. At this time, any addressed device is a slave. Whether the I^{2}C device is a master or a slave, it can send or receive data. Therefore, the I2C interface supports four operation modes:

- Slave transmitter mode
- Slave receiver mode
- Master transmitter mode
- Master receiver mode

After system reset, I^{2}C works in slave mode by default. The I^{2}C interface is configured by software to send a start bit on the bus, and then the interface automatically switches from the slave mode to the master mode. When arbitration is lost or a stop signal is generated, the interface will switched to the slave mode from the master mode.

The block diagram of I^{2}C interface is shown in the figure below.

Figure 24-1 I2C Functional Block Diagram

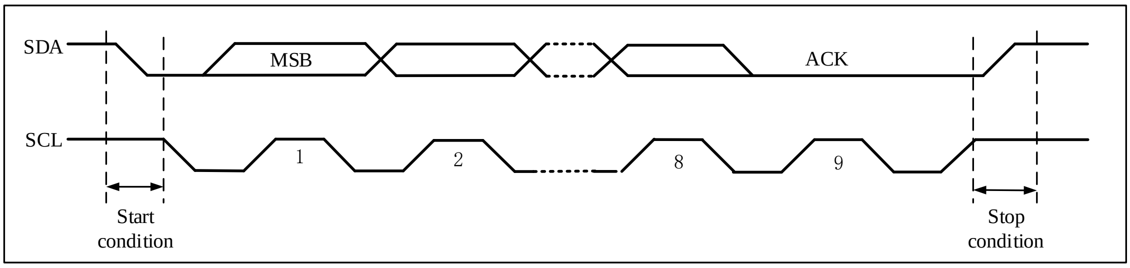


Note: in SMBus mode, SMBALERT is an optional signal. If SMBus is disabled, the signal cannot be used.

24.3.2.1 Start and stop conditions

All data transfers always start with the start bit and end with the stop bit. The start and stop conditions are generated by software in the master mode. Start bit is a level conversion from high to low on SDA line when SCL is high. Stop bit is a level transition from low to high on SDA line when SCL is high. as shown in the figure below.

Figure 24-2 I2C Bus Protocol



24.3.2.2 Clock synchronization and arbitration

The I^{2}C module supports multi-master arbitration, which means two masters can initiate an I^{2}C start operation concurrently when the bus is inactive. So some mechanisms are needed to grant a master the access to the bus. This process is generally named Clock Synchronization and Arbitration.

I^{2}C module has two key features:

- SDA and SCL are open-drain circuit structures, and the signal 'wire-AND' logic is realized through an external pull-up resistor.
- SDA and SCL pins will also detect the level on the pin while outputting the signal to check whether the output is consistent with the previous output. This provides the hardware basis for "Clock Synchronization" and "Bus Arbitration".

The I^{2}C device on the bus is to output logic 0 by grounding the line. Based on the characteristics of the I^{2}C bus, if one device sends logic 0 and the other sends logic 1, then the line sees only logic 0, so there is no possibility of level conflicts on the line.

The physical connection of the bus allows the master to read data while writing data to the bus. In this way, when two masters are competing for the bus, the one that sends logic 0 does not know the occurrence of the competition. Only the one that sends logic 1 will find the conflict (when writing a logic 1, but read 0) and exit the competition.

Clock synchronization

The high-to-low switching of the SCL line causes the devices to begin counting their low-level periods, and once the device's clock goes low, it keeps the SCL line in this state until the high-level of the clock is reached. However, if another clock is still in the low period, the low-to-high switch of this clock will not change the state of the SCL line. Therefore, the SCL line is kept low by the device with the longest low-level period. A device with a short low-level period will enter a high-level wait state.

When all related devices have counted their low-level periods, the clock line is released and goes high-level, after which there is no difference in the state of the device clock and SCL lines, and all devices will begin counting their high-level periods, the device that completes the high-level period first will pull the SCL line low again.

In this way, the low-level period of the generated synchronous SCL clock is determined by the device with the longest low-level clock period, and the high-level period is determined by the device with the shortest high-level clock period.

Arbitration:

Arbitration, like synchronization, is to resolve bus control conflicts in the case of multiple masters. The arbitration process has nothing to do with the slave. When the two masters both produce a valid start bit when the bus is idle, in this case, it is necessary to decide which master will complete the data transmission. This is the process of arbitration.

Each master controller does not have the priority level of controlling the bus, which is all determined by arbitration. The bus control is determined and carried out bit by bit. They follow the principle of "low level first", that is, whoever sends the low level first will control the bus. During the arbitration of each bit, when SCL is high, each host checks whether its own SDA level is the same as that sent by itself. In theory, if the content transmitted by two hosts is exactly the same, then they can successfully transmit without errors. If a host sends a high level but detects that the SDA line is low, it considers that it has lost arbitration and shuts down its SDA output driver, while the other host continues to complete its own transmission.

24.3.2.3 I2C data communication process

Each I^{2}C device is identified by a unique address. According to the device function, they can be either a transmitter or a receiver.

The I^{2}C host is responsible for generating the start bit and the end bit in order to start and end a transmission. And is responsible for generating the SCL clock.

The I^{2}C module supports 7-bit and 10-bit addresses, and the user can configure the address of the I^{2}C slave through software. After the I^{2}C slave detects the start bit on the I^{2}C bus, it starts to receive the address from the bus, and compares the received address with its own address. Once the two addresses are matched, the I^{2}C slave will send an acknowledgement (ACK) and respond to subsequent commands on the bus: transmit or receive the requested data. In addition, if the software opens a broadcast call, the I^{2}C slave always sends a confirmation response to a broadcast address (0x00).

Data and address are transmitted in 8-bit width, with the most significant bit first. The 1 or 2 bytes following the start condition is the address (1 byte in 7-bit mode, 2 bytes in 10-bit mode). The address is only sent in master mode. During the 9th clock period after 8 clocks of a byte transmission, the receiver must send back an acknowledge bit (ACK) to the transmitter, as shown in the Figure 24-2 I2C I^{2}C bus protocol.

Software can enable or disable acknowledgement (ACK), and can set the I^{2}C interface address (7-bit, 10-bit address or general call address).

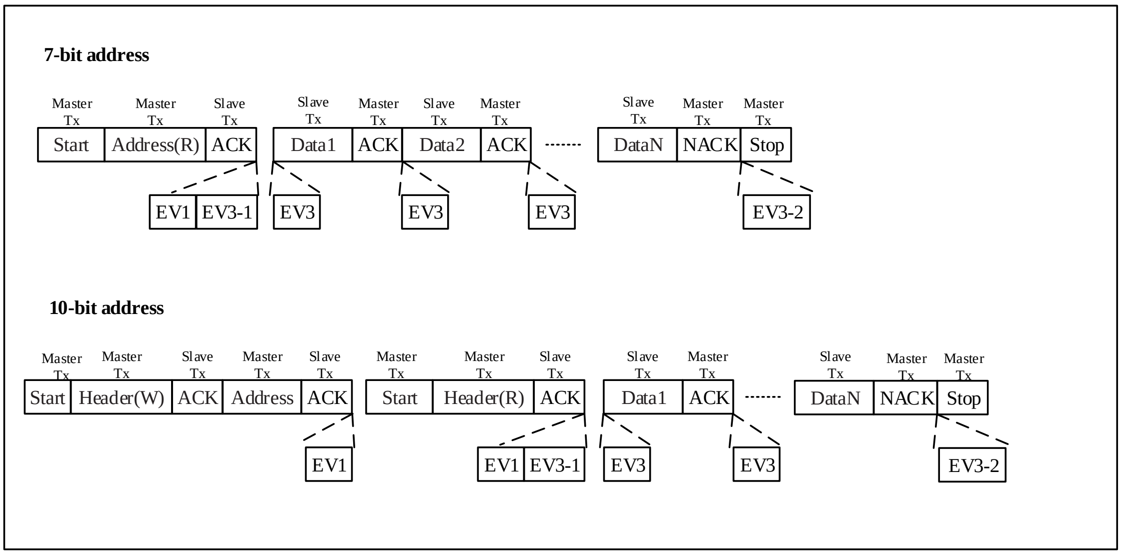
24.3.2.4 I2C slave transmission mode

FIFO Disable

In slave mode, the transmission reception flag bit (I2C_STS2.TRF) indicates whether it is currently in receiver mode or transmission mode. When sending data to I^{2}C bus in transmission mode, the software should follow the following steps:

1. First, enable I^{2}C peripheral clock and configure the clock related register in I2C_CTRL1, ensuring the correct I^{2}C timing. After these two steps are completed, I^{2}C runs in slave mode, waiting for receiving start bit and address.
2. I^{2}C slave receives a start bit first, and then receives a matching 7-bit or 10-bit address. I^{2}C hardware will set the I2C_STS1.ADDRF(received address and matched its own address). The software should monitor this bit regularly or have an interrupt to monitor this bit. After this bit is set, the software reads I2C_STS1 register and then read I2C_STS2 register to clear the I2C_STS1.ADDRF bit. If the address is in 10 bit format, the I^{2}C master should then generate a START and send an address header to the I^{2}C bus. After detecting START and the following address header, the slave will continue to set I2C_STS1.ADDRF bit. The software continues to read I2C_STS1 register and read I2C_STS2 register to clear the I2C_STS1.ADDRF bit a second time.
3. I^{2}C enters the data sending state, and now shift register and data register I2C_DAT are all empty, so the hardware will set the I2C_STS1.TXDATE(send data empty). At this time, the software can write the first byte data to I2C_DAT register, however, because the byte of the I2C_DAT register is immediately moved into the internal shift register, the I2C_STS1.TXDATE bit is not cleared to zero. When the shift register is not empty, I2C starts to send data to I^{2}C bus.
4. During the transmission of the first byte, the software writes the second byte to I2C_DAT, neither the I2C_DAT register nor the shift register is empty. The I2C_STS1.TXDATE bit is cleared to 0.
5. After the first byte is sent, I2C_STS1.TXDATE is set again, and the software writes the third byte to I2C_DAT, the same time, the I2C_STS1.TXDATE bit is cleared.After that, as long as there is still data to be sent and I2C_STS1.TXDATE is set to 1, the software can write a byte to I2C_DAT register.
6. During the sending of the second last byte, the software writes the last data to the I2C_DAT register to clear the I2C_STS1.TXDATE flag bit, and then the I2C_STS1.TXDATE status is no longer concerned. I2C_STS1.TXDATE bit is set after the second last byte is sent until the stop end bit is detected.
7. According to the I^{2}C protocol, the I^{2}C master will not send a ACK to the last byte received. Therefore, after the last byte is sent, the I2C_STS1.ACKFAIL bit (acknowledge fail) of the I^{2}C slave will be set to notify the software of the end of sending. The software writes 0 to the I2C_STS1.ACKFAIL bit to clear this bit.

Figure 24-3 Slave Transmitter Transfer Sequence Diagram



FIFO Enable

In slave mode, the transmission reception flag bit (I2C_STS2.TRF) indicates whether it is currently in receiver mode or transmission mode. When sending data to I^{2}C bus in transmission mode, the software should follow the following steps:

1. First, enable the I^{2}C peripheral clock, configure the clock-related registers in I2C_CTRL1 to ensure correct I^{2}C timing. Once these two steps are completed, the I^{2}C operates in slave mode, waiting for data to be written into the FIFO and to receive the start bit and address.

Note: When FIFOEN is enabled, BUFINTEN must be disabled; when using DMA, DMAFIFOEN should be enabled and DMAEN should be disabled.

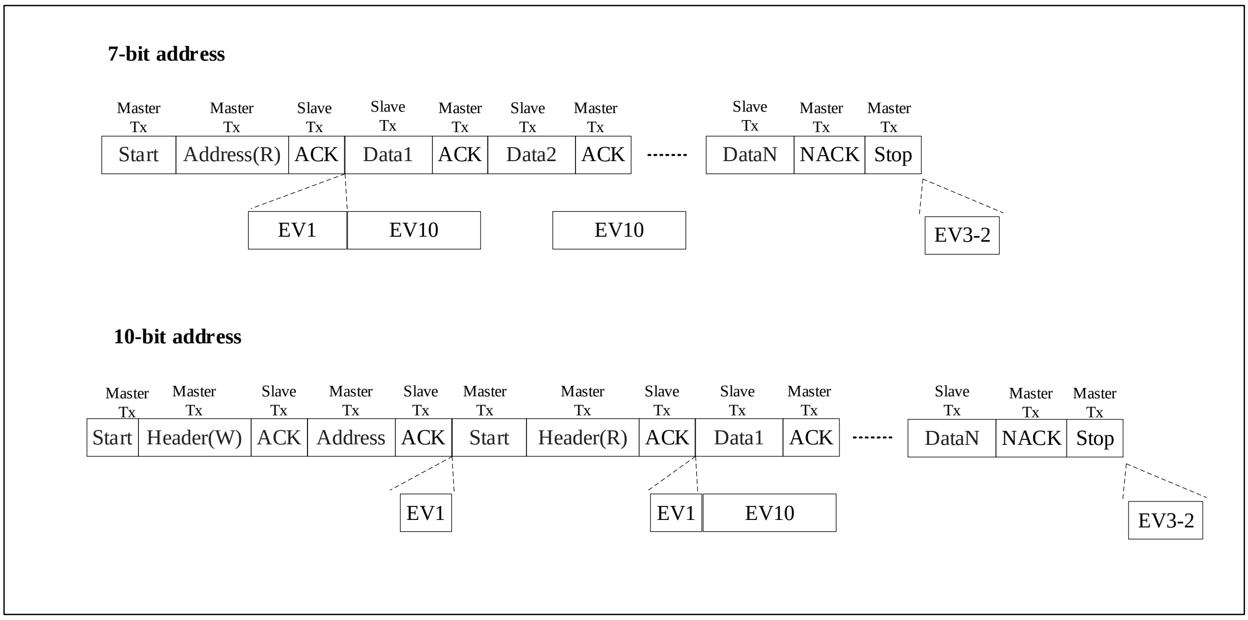
2. The I^{2}C slave first receives a start bit, followed by a matching 7-bit or 10-bit address. The I^{2}C hardware sets the ADDRF bit in the status register (I2C_STS1) to 1 (indicating that it has received the address and it matches its own address). The software should periodically check this bit or monitor it with an interrupt. Upon finding it set, the software reads the I2C_STS1 register and then the I2C_STS2 register to clear the I2C_STS1.ADDRF bit. If the address is in 10-bit format, the I^{2}C master should then generate another START (Sr) and send an address header to the I^{2}C bus. The slave, upon detecting the START (Sr) and the subsequent address header, will continue to set the I2C_STS1.ADDRF bit to 1. The software continues to clear the ADDRF bit by reading the I2C_STS1 register and then the I2C_STS2 register for the second time.

3. I^{2}C enters the data transmission state. To ensure proper data transmission, the software needs to ensure that there is at least one byte of data in the FIFO. The data transmission is handled by the hardware, which reads data from the FIFO. If PECEN is enabled, the hardware controls the transmission of the last byte for checksum data.

4. According to the I^{2}C protocol, the I^{2}C master does not acknowledge the last byte it receives. Therefore, after sending the last byte, the I^{2}C slave's ACKFAIL bit (acknowledge failure) is set to notify the software of the end of transmission. Writing 0 to the I2C_STS1.ACKFAIL bit can clear this bit.

5. Upon completion of communication, the slave clears the FIFO by configuring I2C_CTRL1.FIFOCLR.

Figure 24-4 Enable FIFO to transmit sequences from the transmitter



Instructions：

1. EV1: I2C_STS1.ADDRF = 1, reading STS1 and then STS2 will clear this event.
2. EV3-1: I2C_STS1.TXDATE=1, shift register empty, data register empty, write DAT.
3. EV3: I2C_STS1.TXDATE=1, shift register not empty, data register empty, writing DAT will clear this event.
4. EV3-2: I2C_STS1.ACKFAIL=1, writing '0' to the ACKFAIL bit in STS1 will clear this event.
5. EV10: I2C_STS1.FIFOHE=1, I2C_STS1.FIFOE=1, write FIFODAT, can be written multiple times, I2C_STS1.FIFOHE=0 or I2C_STS1.FIFOE=0.

Notes:

(1) EV1 and EV3_1 event prolongs the low SCL time until the end of the corresponding software sequence.

(2) The software sequence of EV3 must be completed before the end of the current byte transfer.

24.3.2.5 I2C slave receiving mode

FIFO Disable

When receiving data in slave mode, the software should follow these steps:

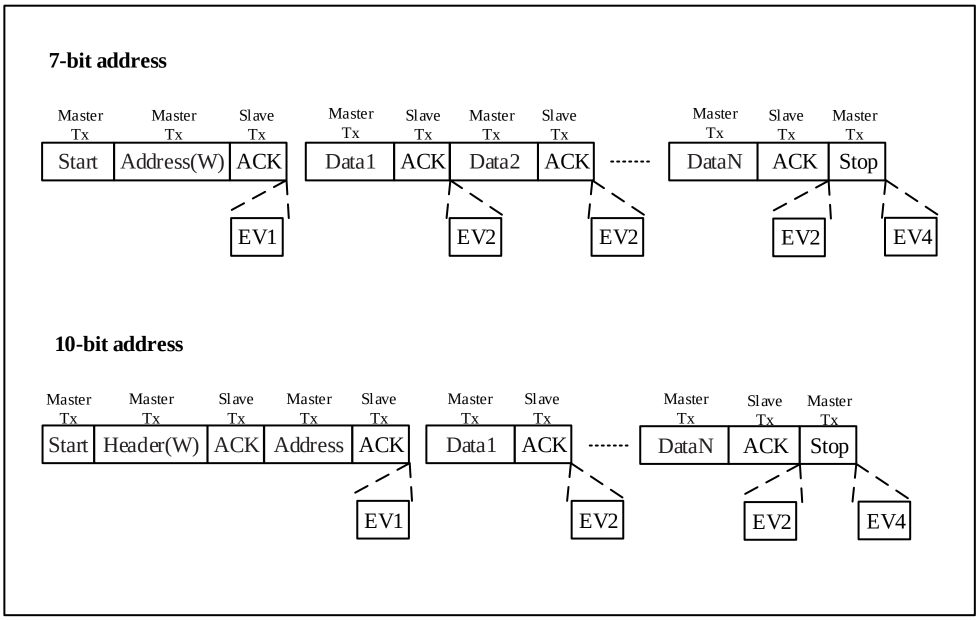
1. First, enable the I^{2}C peripheral clock and configure the clock-related registers in I2C_CTRL1 to ensure correct I^{2}C timing. Once these steps are completed, the I^{2}C operates in slave mode, waiting to receive the start bit and address.
2. After receiving the START condition and a matching 7-bit or 10-bit address, the I^{2}C hardware sets the I2C_STS1.ADDRF bit (indicating the address received matches its own address) to 1. This bit should be detected by software through polling or interrupts. Once detected, the software clears the I2C_STS1.ADDRF bit by first reading the I2C_STS1 register and then the I2C_STS2 register. Once the I2C_STS1.ADDRF bit is cleared, the I2C slave begins receiving data from the I^{2}C bus.

3. Upon receiving the first byte, the I2C_STS1.RXDATNE bit (receive data not empty) is set to 1 by the hardware. If the I2C_CTRL2.EVTINTEN and I2C_CTRL2.BUFINTEN bits are set, an interrupt is generated. The software should detect this bit through polling or interrupts. Once detected, the software can read the first byte from the I2C_DAT register, and the I2C_STS1.RXDATNE bit is cleared. Note that if the I2C_CTRL1.ACKEN bit is set, the slave should generate an acknowledgment pulse after receiving a byte.

4. At any time the I2C_STS1.RXDATNE bit is set to 1, the software can read a byte from the I2C_DAT register. When the last byte is received, the I2C_STS1.RXDATNE is set to 1, and the software reads the last byte.

5. When the slave detects a stop condition (STOP) on the I^{2}C bus, the I2C_STS1.STOPF bit is set to 1. If the I2C_CTRL2.EVTINTEN bit is set, an interrupt is generated. The software clears the I2C_STS1.STOPF bit by reading the I2C_STS1 register and then writing to the I2C_CTRL1 register (see EV4 in the diagram below).

Figure 24-5 Slave receiver transmitting sequence



FIFO Enable

When receiving data in slave mode, the software should follow these steps:

1. Firstly, enable the I^{2}C peripheral clock, configure the BYTENUM register to specify the number of bytes the slave should receive, and configure the clock-related registers in I2C_CTRL1 to ensure correct I^{2}C timing. Once these steps are completed, the I^{2}C operates in slave mode, waiting to receive the start bit and address.

Note: When FIFOEN is enabled, BUFINTEN must be disabled; when using DMA, DMAFIFOEN should be enabled and DMAEN should be disabled.

2. After receiving the START condition and a matching 7-bit or 10-bit address, the I^{2}C hardware sets the I^{2}C status register I2C_STS1.ADDRF bit (indicating the address received matches its own address) to 1. This bit should be detected by software through polling or interrupts. Once detected, the software clears the I2C_STS1.ADDRF bit by first reading the I2C_STS1 register and then the I2C_STS2 register. Once the I2C_STS1.ADDRF bit is cleared, the I^{2}C slave begins receiving data from the I^{2}C bus.

3. When the first byte is received, the hardware writes the data into the FIFO. If the I2C_CTRL2.FIFOHFINTEN and I2C_CTRL2.FIFOFINTEN bits are set, an interrupt is generated. The software should detect this bit through

polling or interrupts. Once FIFOF or FIFOHF is detected, the software can read the received data from the I2C_FIFODAT register. Note that if the I2C_CTRL1.ACKEN bit is set, the slave should generate an acknowledgment pulse after receiving a byte.

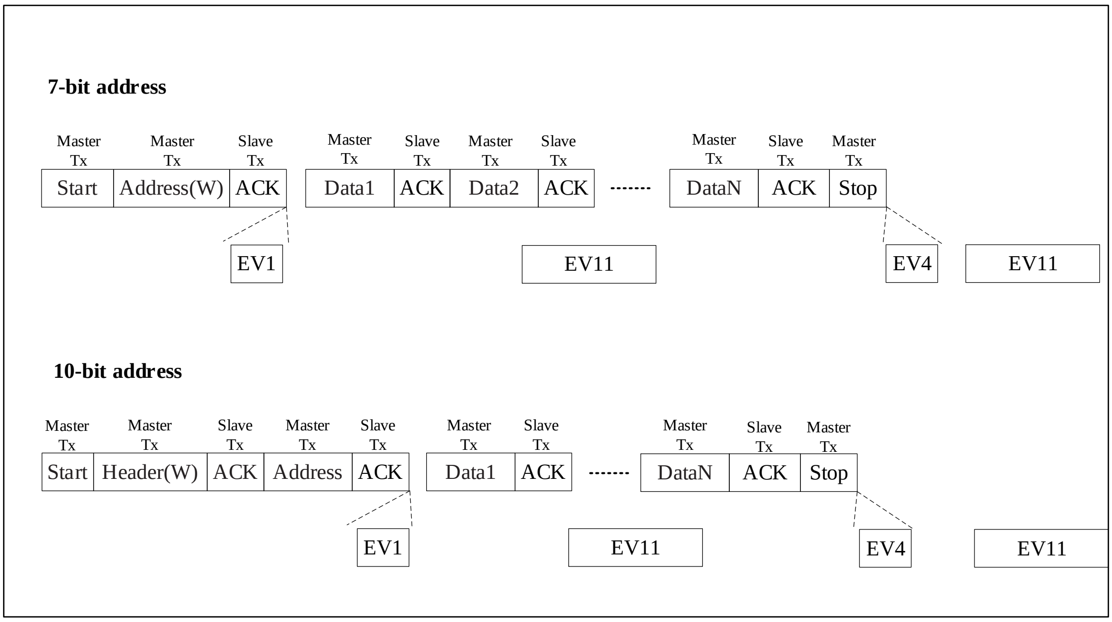
4. At any time, when the I2C_STS1.FIFOHF or FIFOF bit is set to 1, the software can read a byte from the I2C_FIFODAT register.

Note: If PECEN is set, the last received byte is a checksum data. The correctness of the checksum data can be determined by PECERR. The received data should be processed from the FIFO before disabling the peripheral.

5. When the slave detects the stop condition (STOP) on the I^{2}C bus, the I2C_STS1.STOPF bit is set to 1. If the I2C_CTRL2.EVTINTEN bit is set, an interrupt is generated. The software clears the I2C_STS1.STOPF bit by first reading the I2C_STS1 register and then writing to the I2C_CTRL1 register (see EV4 in the diagram below).

6. If I2C_STS1.FIFOHF=1, read FIFODAT multiple times until FIFOHF=0, indicating completion of communication reception by the slave device, then clear the FIFO by configuring I2C_CTRL1.FIFOCLR.

Figure 24-6 FIFO Enable opening the slave receiver to transmit sequences.



Instructions：

1. EV1: I2C_STS1.ADDRF = 1, reading STS1 and then STS2 to clear the event.
2. EV2: I2C_STS1.RXDATNE =1, reading DAT will clear this event.
3. EV4: I2C_STS1.STOPF=1, reading STS1 and then writing the CTRL1 register will clear this event.
4. EV11:I2C_STS1.FIFOHF=1, reading FIFODAT multiple times until I2C_STS1.FIFOHF=0.

Notes:

(1) EV1 event prolongs the time when SCL is low until the end of the corresponding software sequence.

(2) The software sequence of EV2 must be completed before the end of the current byte transmission.

24.3.2.6 I2C master transmission mode

FIFO Disable

In the master mode, the I^{2}C interface starts data transmission and generates a clock signal. Serial data transmission always starts with a start condition and ends with a stop condition. When the START condition is generated on the bus through the start bit, the device enters the master mode.

When sending data to I^{2}C bus in master mode, the software should operate as follows:

1. First, enable the I^{2}C peripheral clock, and configure the clock-related registers in I2C_CTRL1 to ensure the correct I^{2}C timing. When these two steps are completed, I^{2}C runs in the slave mode by default, waiting for receiving the start bit and address.

2. When BUSY=0, I2C_CTRL1.STARTGEN bit set to 1, and the I^{2}C interface will generate a start condition and switch to the master mode (I2C_STS2.MSMODE bit set to “1”).

3. Once the start condition is issued, I^{2}C hardware will set I2C_STS1.STARTBF bit(START bit flag) and then enters the master mode. If the I2C_CTRL2.EVTINTEN bit is set, an interrupt will be generated. Then the software reads the I2C_STS1 register and then writes a 7-bit address bit or a 10-bit address bit with an address header to the I2C_DAT register to clear the I2C_STS1.STARTBF bit. After the I2C_STS1.STARTBF bit is cleared to 0, I^{2}C starts sending addresses or address headers to I^{2}C bus.

   In 10-bit address mode, sending a header sequence will generate the following events:

   – I2C_STS1.ADDR10F bit is set by hardware, and if I2C_CTRL2.EVTINTEN bit is set, an interrupt is generated. Then the master reads the STS1 register, and then writes the second address byte into the DAT register.

   – I2C_STS1.ADDRF bit is set by hardware, and if I2C_CTRL2.EVTINTEN bit is set, an interrupt is generated. Then the master reads the STS1 register, followed by the STS2 register.

   Note: In the transmitter mode, the master device first transmits the header byte (11110xx0) and then transmits the lower 8 bits of the slave address. (where xx represents the highest 2 bits of the 10-bit address).

   In the 7-bit address mode, only one address byte needs to be sent out. Once the address byte is sent out:

   – I2C_STS1.ADDRF bit is set by hardware, and if I2C_CTRL2.EVTINTEN bit is set, an interrupt is generated. Then the master device waits for reading the STS1 register once, followed by reading the STS2 register.

   Note: in the transmitter mode, when the master transmits the slave address, set the lowest bit to "0".

   Note: When the master transmission and in 7-bit address mode, the slave address cannot be configured as 0xF0, 0xF2, 0xF4, or 0xF6.

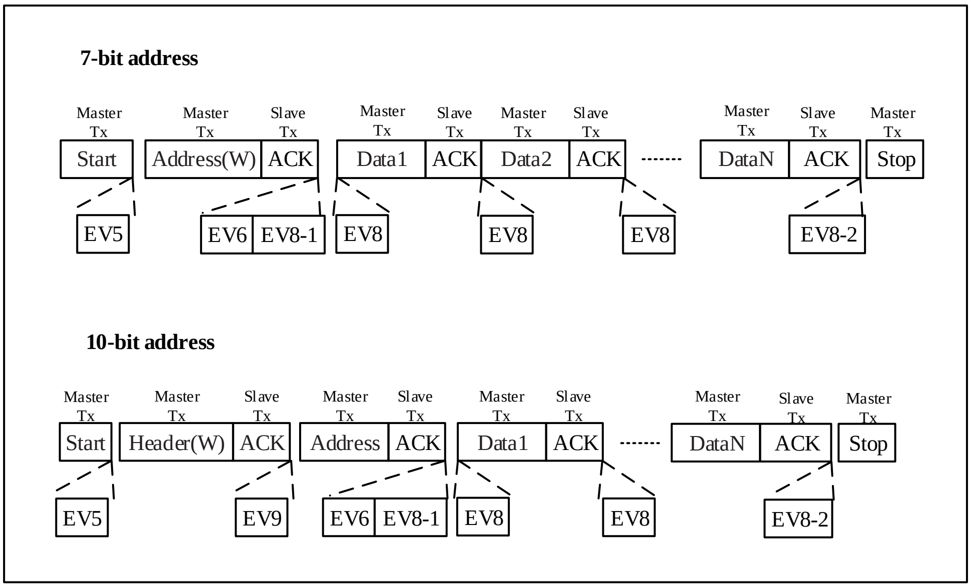
4. After the 7-bit or 10-bit address bit is sent, the I2C hardware sets the I2C_STS1.ADDRF bit (address has been sent) to 1, if the I2C_CTRL2.EVTINTEN bit is set, an interrupt is generated, and the software is cleared by reading the I2C_STS1 register and then the I2C_STS2 register I2C_STS1.ADDRF.

5. I^{2}C enters the data transmission state. Because the shift register and the data register (I2C_DAT) are empty, the hardware sets the I2C_STS1.TXDATE bit (transmission data empty) to 1, and then the software writes the first byte of data to the I2C_DAT register, but because the byte written into the I2C_DAT register is immediately moved into the internal shift register, the I2C_STS1.TXDATE bit will not be cleared at this time. Once the shift

register is not empty, I^{2}C starts transmitting data to the bus.

6. During the transmission of the first byte, the software writes the second byte to I2C_DAT, and I2C_STS1.TXDATE is cleared at this time. At any time, as long as there is data waiting to be sent and the I2C_STS1.TXDATE bit is set to 1, the software can write a byte to the I2C_DAT register.
7. In the process of sending the penultimate byte, the software writes the last byte of data to I2C_DAT to clear the I2C_STS1.TXDATE flag bit. After that, there is no need to care about the status of the I2C_STS1.TXDATE bit. The I2C_STS1.TXDATE bit will be set after the penultimate byte is sent, and will be cleared when the stop bit (STOP) is sent.
8. After the last byte is sent, because the shift register and the I2C_DAT register are empty at this time, the I^{2}C host sets the I2C_STS1.BSF bit (byte transmission end), and the I2C interface will keep SCL low before clearing the I2C_STS1.BSF bit. After reading I2C_STS1, writing to the I2C_DAT register will clear the I2C_STS1.BSF bit. The software sets the I2C_CTRL1.STOPGEN bit at this time to generate a stop condition, and then the I2C interface will automatically return to the slave mode (I2C_STS2.MSMODE bit is cleared).

Figure 24-7 Master Transmitter Transfer Sequence Diagram



FIFO Enable

When sending data to the I^{2}C bus in master mode, software should follow the steps below:

1. First, enable the I^{2}C peripheral clock, configure clock-related registers in I2C_CTRL1 to ensure the correct I^{2}C timing. Once these two steps are completed, the I2C defaults to running in slave mode.

Note: When I2C_CTRL1.FIFOEN is enabled, I2C_CTRL2.BUFINTEN must be disabled; when using DMA, enable DMAFIFOEN and disable DMAEN.

2. When BUSY=0, set the STARTGEN bit in the I2C_CTRL1 register to 1. The I^{2}C interface will generate a start condition and switch to master mode (MSMODE bit set).
3. Once the start condition is issued, the I^{2}C hardware sets the STARTBF bit (start bit flag) in I2C_STS1 to 1 and enters master mode. If the EVTINTEN bit is set, an interrupt will be generated. The software then reads the

I2C_STS1 register and writes a 7-bit address or a 10-bit address with an address header to the I2C_DAT register to clear the STARTBF bit. After the STARTBF bit is cleared to 0, the I^{2}C starts sending the address or address header to the I^{2}C bus.

When in 10-bit address mode, sending a header sequence will result in the following events:

– ADDR10F bit is set by hardware, and if EVTINTEN bit is set, an interrupt is generated. The master device then reads the STS1 register and writes the second address byte into the DAT register.

– ADDRF bit is set by hardware, and if EVTINTEN bit is set, an interrupt is generated. The master device then reads the STS1 register, followed by reading the STS2 register.

Note: In transmitter mode, the master device first sends the header byte (11110xx0), then sends the low 8 bits of the slave address. (Here, xx represents the highest 2 bits in the 10-bit address).

In 7-bit address mode, only one address byte needs to be sent out. Once this address byte is sent out:

– ADDRF bit is set by hardware, and if EVTINTEN bit is set, an interrupt is generated. The master device then waits for a read of the STS1 register, followed by a read of the STS2 register.

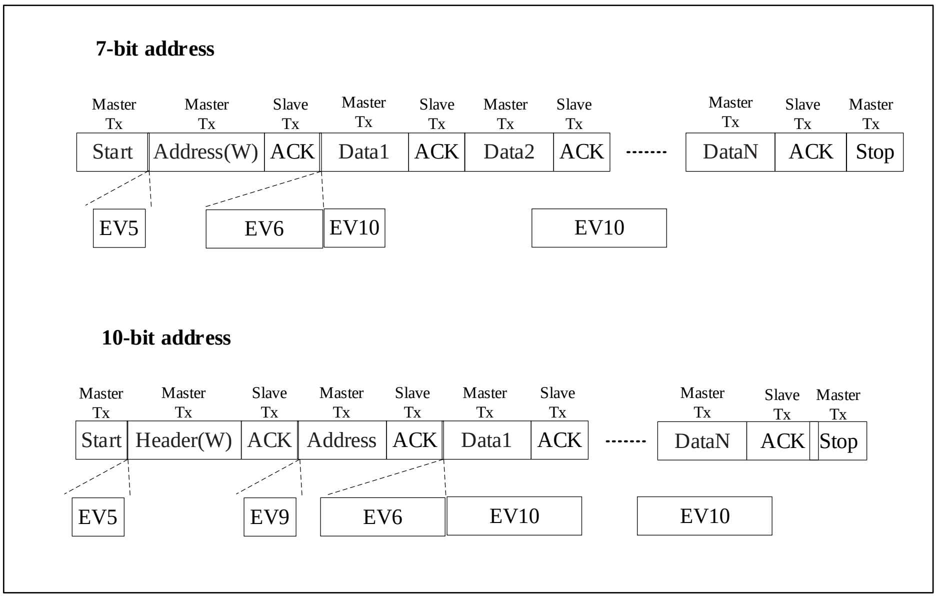
Note: In transmitter mode, the master device sets the least significant bit to '0' when sending the slave address.

4. After sending the 7-bit or 10-bit address bits, the I^{2}C hardware sets the ADDRF bit (address has been sent) to 1. If the EVTINTEN bit is set, an interrupt is generated. The software clears ADDRF by reading the I2C_STS1 register and then reading the I2C_STS2 register.

5. The I^{2}C enters the data transmission state, checking if I2C_STS1.FIFOHE and FIFOE are both 1. The software then writes the data to be transmitted into the FIFODAT register. The data to be sent is moved from the FIFODAT to the shift register for transmission, and this operation is autonomously handled by the hardware until the FIFO data transmission is completed.

6. After the transmission of the last byte is completed, as both the shift register and I2C_FIFODAT register are empty at this point, the I^{2}C master sets the BSF bit (byte send finished). Before clearing the BSF bit, the I^{2}C interface keeps the SCL low; reading I2C_STS1 and then writing to I2C_DAT will clear the BSF bit. At this point, the software sets the STOPGEN bit to generate a stop condition (refer to EV8_2), and then the I^{2}C interface will automatically return to slave mode (MSMODE bit cleared).

Figure 24-8 FIFO Enable opening the master transmitter to transmit sequences



Instructions：

1. EV5: I2C_STS1.STARTBF = 1, reading STS1 and writing the address to the DAT register will clear the event.
2. EV6: I2C_STS1.ADDRF = 1, read STS1 and then STS2 to clear the event.
3. EV8_1: I2C_STS1.TXDATE = 1, shift register is empty, data register is empty, write DAT register.
4. EV8: I2C_STS1.TXDATE = 1, shift register is not empty, data register is empty, write to DAT register will clear the event.
5. EV8_2: I2C_STS1.TXDATE = 1, I2C_STS1.BSF = 1, request to set stop bit. These two events are cleared by the hardware when a stop condition is generated.
6. EV9: I2C_STS1.ADDR10F = 1, read STS1 and then write to DAT register to clear the event.
7. EV10:I2C_STS1.FIFOHE=1， I2C_STS1.FIFOE=1, write FIFODAT, can be written multiple times, until I2C_STS1.FIFOHE=0 or FIFOE=0.

Notes:

(1) EV5, EV6, EV9, EV8_1 and EV8_2 event prolonged the low SCL time until the end of the corresponding software sequence.

(2) The software sequence of EV8 must be completed before the end of the current byte transfer.

(3) When I2C_STS1.TXDATE or I2C_STS1.BSF bit is set, stop condition should be arranged when EV8_2 occurs.

24.3.2.7 I2C master receiving mode

In master mode, software receiving data from I^{2}C bus should follow the following steps:

1. First, enable the I^{2}C peripheral clock and configure the clock-related registers in I2C_CTRL1, in order to ensure that the correct I^{2}C timing is output. After enabling and configuring, I$_2$C runs in slave mode by default, waiting to receive the start bit and address.

2. When BUSY=0, set the I2C_CTRL.STARTGEN bit, and the I^{2}C interface will generate a start condition and switch to the master mode (I2C_STS2.MSMODE bit is set to 1).

3. Once the start condition is issued, the I^{2}C hardware sets I2C_STS1.STARTBF(start bit flag) and enters the host mode. If the I2C_CTRL2.EVTINTEN bit is set to 1, an interrupt will be generated. Then the software reads the I2C_STS1 register and then writes a 7-bits address or a 10-bits address with an address header to the I2C_DAT register, in order to clear the I2C_STS1.STARTBF bit. After the I2C_STS1.STARTBF bit is cleared to 0, I^{2}C begins to send the address or address header to the I^{2}C bus.
   In 10-bits address mode, sending a header sequence will generate the following events:

   – The I2C_STS1.ADDR10F bit is set to 1 by hardware, and if the I2C_CTRL2.EVTINTEN bit is set to 1, an interrupt will be generated. Then the master device reads the STS1 register, and then writes the second byte of address into the DAT register.

   – The I2C_STS1.ADDRF bit is set to 1 by hardware, and if the I2C_CTRL2.EVTINTEN bit is set to 1, an interrupt will be generated. Then the master device reads the STS1 register and the STS2 register in sequence.

   Note: In the receiver mode, the master device sends the header byte (11110xx0) firstly, then sends the lower 8 bits of the slave address, and then resends a start condition followed by the header byte (11110xx1) (where xx represents the highest 2 digits of the 10-bits address).

   In the 7-bits address mode, only one address byte needs to be sent, once the address byte is sent:

   – The I2C_STS1.ADDRF bit is set to 1 by hardware, and if the I2C_CTRL2.EVTINTEN bit is set to 1, an interrupt will be generated. Then the master device waits to read the STS1 register once, and then reads the STS2 register.

   Note: In the receiving mode, the master device sets the lowest bit as '1' when sending the slave address.

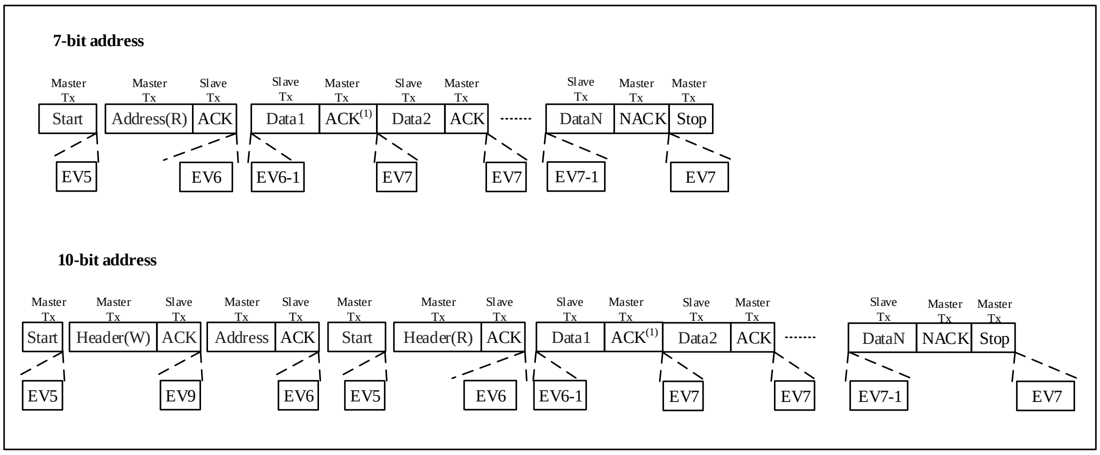
4. After the 7-bits or 10-bits address is sent, the I^{2}C hardware sets the I2C_STS1.ADDRF bit (address has been sent) to 1. If the I2C_CTRL2.EVTINTEN bit is set to 1, an interrupt will be generated. The software clears the I2C_STS1.ADDRF bit by reading the I2C_STS1 register and the I2C_STS2 register in sequence. If in the 10-bit address mode, software should set the I2C_CTRL1.STARTGEN bit again to regenerate a START. After the START is generated, the I2C_STS1.STARTBF bit will be set. The software should clear the I2C_STS1.STARTBF bit by reading I2C_STS1 firstly and then writing the address header to I2C_DAT, and then the address header is sent to the I^{2}C bus, I2C_STS1.ADDRF is set to 1 again. The software should clear the I2C_STS1.ADDRF bit again by reading I2C_STS1 and I2C_STS2 in sequence.
5. After sending the address and clearing the I2C_STS1.ADDRF bit, the I^{2}C interface enters the host receiving mode. In this mode, the I^{2}C interface receives data bytes from the SDA line and sends them to the DAT register through the internal shift register. Once the first byte is received, the hardware will set the I2C_STS1.RXDATNE bit (not empty flag bit of received data) to 1, and if the I2C_CTRL1.ACKEN bit is set to 1, an acknowledge pulse will be sent. At this time, the software can read the first byte from the I2C_DAT register, and then the I2C_STS1.RXDATNE bit is cleared to 0. After that, as long as I2C_STS1.RXDATNE is set to 1, the software can read a byte from the I2C_DAT register.
6. The master device sends a NACK after receiving the last byte from the slave device. After receiving the NACK, the slave device releases the control of SCL and SDA lines; the master device can send a stop/restart condition. In order to generate a NACK pulse after receiving the last byte, the software should clear the

I2C_CTRL1.ACKEN bit immediately after receiving the penultimate byte (N-1). In order to generate a stop/restart condition, the software must set the I2C_CTRL1.STOPGEN bit or I2C_CTRL1.STARTGEN to 1 after reading the penultimate data byte. This process needs to be completed before the last byte is received to ensure that the NACK is sent for the last byte.

7. After the last byte is received, the I2C_STS1.RXDATNE bit is set to 1, and the software can read the last byte. Since I2C_CTRL1.ACKEN has been cleared to 0 in the previous step, I^{2}C no longer sends ACK for the last byte, and generates a STOP bit after the last byte is sent.

Note: The above steps require the number of bytes N>1. If N=1, step 6 should be executed after step 4, and it needs to be completed before the reception of byte is completed.

Figure 24-9 Master receiver transmission sequence diagram



FIFO Enable

When receiving data from the I^{2}C bus in master mode, software should follow the steps below:

1. First, enable the I^{2}C peripheral clock, configure the BYTENUM register to specify the number of bytes the host needs to receive, configure clock-related registers in I2C_CTRL1 to ensure correct I^{2}C timing. After enabling and configuring, the I^{2}C defaults to running in slave mode, waiting to receive the start bit and address.

Note: When FIFOEN is enabled, BUFINTEN must be disabled; when using DMA, enable DMAFIFOEN and disable DMAEN.

2. When BUSY=0, set the STARTGEN bit in the I2C_CTRL1 register to 1. The I^{2}C interface will generate a start condition and switch to master mode (setting the MSMODE bit).

3. Once the start condition is issued, the I^{2}C hardware sets the STARTBF bit (start bit flag) in I2C_STS1 to 1 and enters master mode. If the EVTINTEN bit is set, an interrupt will be generated. The software then reads the I2C_STS1 register and writes a 7-bit address or a 10-bit address with an address header to the I2C_DAT register to clear the STARTBF bit. After the STARTBF bit is cleared to 0, the I^{2}C starts sending the address or address header to the I2C bus.

   When in 10-bit address mode, sending a header sequence will result in the following events:

   – ADDR10F bit is set by hardware, and if EVTINTEN bit is set, an interrupt is generated. The master device then reads the STS1 register, followed by writing the second address byte into the DAT register.

   – ADDRF bit is set by hardware, and if EVTINTEN bit is set, an interrupt is generated. The master device then reads the STS1 register, followed by reading the STS2 register.

Note: In receiver mode, the master device first sends the header byte (11110xx0), followed by sending the low 8 bits of the slave address, then resends a start condition, followed by the header byte (11110xx1) (where xx represents the highest 2 bits in the 10-bit address).

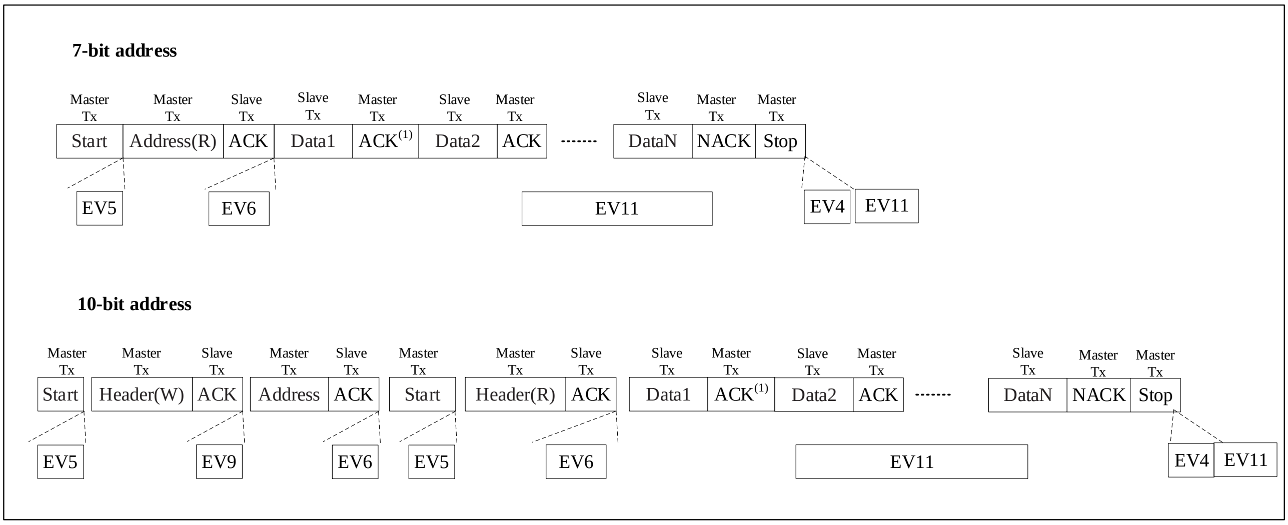
In 7-bit address mode, only one address byte needs to be sent out. Once this address byte is sent out:

– ADDRF bit is set by hardware, and if EVTINTEN bit is set, an interrupt is generated. The master device then waits for a read of the STS1 register, followed by a read of the STS2 register.

Note: In receiver mode, the master device sets the least significant bit to '1' when sending the slave address.

4. After sending the 7-bit or 10-bit address bits, the I^{2}C hardware sets the ADDRF bit (address has been sent) to 1. If the EVTINTEN bit is set, an interrupt is generated. The software clears ADDRF by reading the I2C_STS1 register and then reading the I2C_STS2 register. After sending the address and clearing ADDRF, if the address is in 10-bit format, the software should set STARTGEN to 1 again to generate a new START (Sr). After the START is generated, the STARTBF bit is set to 1. The software should clear the STARTBF bit by first reading I2C_STS1 and then writing the address header to I2C_DAT, then the address header is sent to the I^{2}C bus, and ADDRF is set to 1 again. The software should clear ADDRF by reading I2C_STS1 and then reading I2C_STS2.
5. After sending the address and clearing ADDRF, the I^{2}C interface enters the master receiver mode. In this mode, the I^{2}C interface receives data bytes from the SDA line and sends them to the FIFODAT register through internal shift registers.
6. The master device sends a NACK after receiving the last byte from the slave device. Upon receiving the NACK, the slave device releases control over the SCL and SDA lines; the master device can then send a stop/restart condition.
7. When the stop condition (STOP) on the I^{2}C bus is detected, the I2C_STS1.STOPF bit is set to 1. If the I2C_CTRL2.EVTINTEN bit is set, an interrupt is generated. The software clears the I2C_STS1.STOPF bit by first reading the I2C_STS1 register and then writing to the I2C_CTRL1 register (refer to EV4 in the diagram below).
8. If FIFOHF=1, read FIFODAT, can be read multiple times, until FIFOHF=0, indicating completion of communication reception by the slave device, perform FIFO clearing through configuring I2C_CTRL1.FIFOCLR.

Figure 24-10 Enable FIFO for the master receiver transmission sequence diagram



Instructions:

1. EV4: I2C_STS1.STOPF=1, Reading STS1 and then writing to the CTRL1 register will clear this event.
2. EV5: I2C_STS1.STARTBF=1, Reading STS1 and then writing the address into the DAT register will clear this event.
3. EV6: I2C_STS1.ADDRF=1, Reading STS1 and then reading STS2 will clear this event. In 10-bit master receive mode, after this event, STARTGEN=1 should be set in CTRL1.
4. EV6_1: There is no corresponding event flag, suitable for receiving one byte. Right after EV6 (i.e., after clearing ADDRF), the generation bits for acknowledgment and stop condition need to be cleared.)
5. EV7: I2C_STS1.RXDATNE=1, Reading the DAT register clears this event.
6. EV7_1: I2C_STS1.RXDATNE =1, Reading the DAT register clears this event. Set I2C_CTRL1.ACKEN=0 and I2C_CTRL1.STOPGEN=1.
7. EV9: I2C_STS1.ADDR10F=1, Reading STS1 and then writing to the DAT register will clear this event.
8. EV11:I2C_STS1.FIFOHF=1， Read FIFODAT, can be read multiple times, until FIFOHF=0.

Notes:

(1) If a single byte is received, it is NA.

(2) EV5, EV6, and EV9 events extend the low level of SCL until the corresponding software sequence ends.

(3) The EV7 software sequence shall be completed before the end of the current byte transmission.

(4) The software sequence of EV6_1 or EV7_1 shall be completed before the ACK pulse of the current transmission byte.

24.3.3 21.3.3 Error Conditions

I^{2}C errors mainly include bus error, acknowledge error, arbitration loss, overload\ underload error. These errors may cause communication failure.

24.3.3.1 Acknowledge failure (ACKFAIL)

The interface have a acknowledge bit is detected that does not match the expectation, it will occurs acknowledge fail error, I2C_STS1.ACKFAIL bit is set. An interrupt occurs, when I2C_CTRL2.ERRINTEN bit is set to 1.

When transmitter receives a NACK, The communication must be reset: Device in slave mode, hardware release the bus; Device in master mode, it must generate a stop condition from software.

24.3.3.2 Bus error (BUSERR)

when address or data is transmissing, I^{2}C interface receive external stop or start condition,it will happen a bus error, I2C_STS1.BUSERR bit is set. An interrupt occurs, when I2C_CTRL2.ERRINTEN bit is set to 1.

I^{2}C device as master, the hardware does not release bus, as the same time it done not affect the current status of transfer, The current transfer will determined by software whether suspend.

I^{2}C device as slave, when data is discarded in transmission and the bus releases by hardware, it will have two situation: If an error start condition is detected, the slave device considers a restart condition and waits for an address or a stop condition. If an error stop condition is detected, the slave device operates as a normal stop condition and the hardware releases the bus.

24.3.3.3 Arbitration lost (ARLOST)

The interface have arbitration lost is detected, hardware release the bus, it will occurs arbitration lost error, I2C_STS1.ARLOST bit is set. An interrupt occurs, when I2C_CTRL2.ERRINTEN bit is set to 1.

I^{2}C interface will go to slave mode automatically (I2C_STS2.MSMODE bit is cleared). When the I^{2}C interface lost the arbitration, in the same communication, it can not respond to its slave address, but it can respond when master win the bus retransmits a start signal. Hardware release the bus.

24.3.3.4 Overrun/underrun error (OVERRUN)

In slave mode, Overrun/Underrun error can easily occur if clock stretching is disable.

When I^{2}C interface is receiving data (I2C_STS1.RXDATNE=1, data have received in register), and I2C_DAT register still has previous byte which has not been read, it will occurs an overrun error. In this situation, the last received data is discarded. And software should clear I2C_STS1.RXDATNE bit, transmitter retransmit last byte.

When I2C interface is sending data (I2C_STS1.TXDATE=1, new data has not sent to register), and I2C_DAT register still empty, it will occur an underrun error. In this situation, the previous byte in the I2C_DAT register is sent repeatedly. And user makes sure that in the event of an underrun error, the receiver discard repeatedly byte, and transmitter should update the I2C_DAT register at the specified time according to the I2C bus standard.

In sending the first byte, I2C_DAT register must be written after I2C_STS1.ADDRF bit is cleared and the before the first SCL rising edge. If cannot make sure do that, the first byte should be discard by receiver.

24.3.4 DMA Application

DMA can generate a requests when transfer data register empty or full. DMA can write data to I^{2}C or read data from I^{2}C reduce the CPU overload.

Before the current byte transfers end, DMA requests must be responded. If set the DMA channel transfer data is done, DMA will send EOT(End Of Transmission) to I^{2}C, and ocurr a interrupt when enable interrupt bit.

In the master transfer mode, in EOT interrupt handler DMA request need to be disbale, and set stop condition after waiting for I2C_STS1.BSF event.

In the master receive mode, the data of received is great than or equal to 2, DMA will send a hardware signal EOT_1 in DMA transmission(byte number-1). If set I2C_CTRL2.DMALAST bit, when hardware has sent the EOT_1 next byte it will send a NACK automatically. The user can set a stop condition in the interrupt handler after the DMA transfer is completed if interrupt enable.

24.3.4.1 Transmit process

DMA burst Disable

DMA mode can be enabled by setting the I2C_CTRL2.DMAEN bit. When I2C_STS1.TXDATE bit is set, the data will send to I2C_DAT from storage area by the DMA. DMA assigns a channle for I2C transmission, (x is the channel number) the following step must be opreate:

1. In the DMA_PADDRx register set the I2C_DAT register address. Data will be send to address in every I2C_STS1.TXDATE event.
2. In the DMA_MADDRx register set the memory address. Data will send to I2C_DAT address in every I2C_STS1.TXDATE event.
3. In the DMA_TXNUMx register set the number of need to be transferred.In every I2C_STS1.TXDATE event this number-1 until 0.
4. In the DMA_CHCFGx register set PRIOLVL[1:0] bit to configure the priority of channel.
5. In the DMA_CHCFGx register set DIR bit to configure when ocurrs an interrupt whether send a half data or all completed.
6. In the DMA_CHCFGx register set CHEN bit to enable transfer channel.
7. When DMA transfer data is done, DMA need send a EOT/EOT_1 signal to I2C indicate this transfer is done. If interrupt is enable, DMA ocurrs a interrrupt.

Note: If DMA is used for transmission, do not set I2C_CTRL2.BUFINTEN bit

DMA burst Enable

DMA mode is enabled through the DMAFIFOEN bit in the I2C_CTRL2 register (DMAEN does not need to be enabled). As long as the I2C_STS1.FIFOHE bit is set, data will be loaded from the preset storage area by DMA into the I2C_FIFODAT register. To set up a DMA channel for I2C transmission, the following steps must be performed (x is the channel number):

1. Configure to enable DMA burst, set DMA_BURSTEN.BURST_BYPASS to 0;
2. Configure DMA_CHCFGx.BLEN length (if it is M2P, P2M mode, the value of DMA_CHCFGx.BLEN needs to match the peripheral FIFO watermark), configure channel DMA_CHCFGx.BURST_MODE;
3. Configure I2C FIFO Watermarks, I2C_CTRL1.THRFIFOF and I2C_CTRL1.THRFIFOE.
4. Set the address of the I2C_FIFODAT register in the DMA_PADRx register. Data will be transferred from memory to this address after each I2C_STS1.FIFOHE event.
5. Set the memory address in the DMA_MADDRx register. Data will be transferred from this storage area to

I2C_FIFODAT after each I2C_STS1.FIFOHE event. (When the remaining transmission data is less than the watermark, the single DMA request is indicated by FIFO Not Full status and FIFOF).

6. Set the number of bytes to be transmitted in the DMA_TXNUMx register. After each I2C_STS1.FIFOHE event, this value decrements the watermark value until it reaches 0.
7. Configure the channel priority using the PRIOLVL bit in the DMA_CHCFGx register.
8. Set the DIR bit in the DMA_CHCFGx register, and configure interrupt requests to be issued upon completion of the entire transfer or half of the transfer as required by the application.
9. Activate the channel by setting the CHEN bit on the DMA_CHCFGx register.
10. When the data transfer count set in the DMA controller is completed, the DMA controller sends an End of Transfer (EOT/EOT_1) signal to the I2C interface. If interrupts are enabled, a DMA interrupt will be generated.

Note: If DMA is used for transmission, do not set I2C_CTRL2.BUFINTEN bit

24.3.4.2 Receive process

DMA burst Disable

DMA mode can be enabled by setting I2C_CTRL2.DMAEN bit. When data byte is received, DMA will send I^{2}C data to storage area. To set DMA channel for I^{2}C reception, the following steps must be opreate:

1. In DMA_PADDRx register set the address of the I2C_DAT register. In every I2C_STS1.RXDATEN event, data will send from address to storage area.
2. In DMA_MADDRx register set the memory area address. In every I2C_STS1.RXDATEN event,data will send from I2C_DAT register to storage area.
3. In DMA_TXNUMx register set the number of need to be transferred. In every I2C_STS1.RXDATEN event the number-1 until 0.
4. In DMA_CHCFGx register set PRIOLVL[0:1] to configure the priority of channel.
5. In DMA_CHCFGx register clear DIR to configure when ocurrs a interrupt request whether received half data or all data is received.
6. In the DMA_CHCFGx register set CHEN bit to activate the channle.
7. When DMA tansfer data is done, DMA need to send EOT/EOT_1 signal to I2C indicate this transfer is done, if interrupt is enbale, DMA ocurrs a interrupt.

Note: If DMA is used for receiving, do not set I2C_CTRL2.BUFINTEN bit.

DMA burst Enable

DMA mode is enabled through the DMAFIFOEN bit in the I2C_CTRL2 register (DMAEN does not need to be opened). As long as the I2C_STS1.FIFOHE bit is set, data will be loaded from the I2C_FIFODAT register into the preset storage area by DMA. To set up a DMA channel for I^{2}C reception, the following steps must be performed (where x is the channel number):

1. Configure and enable DMA burst, set DMA_BURSTEN.BURST_BYPASS bit to 0;
2. Configure the DMA_CHCFGx.BLEN length (if it is M2P or P2M mode, the value of DMA_CHCFGx.BLEN

needs to match the peripheral FIFO waterline), configure the DMA_CHCFGx.BURST_MODE for the channel;

3. Configure the I2C FIFO waterline, I2C_CTRL1.THRFIFOF and I2C_CTRL1.THRFIFOE.
4. Set the address of the I2C_FIFODAT register in the DMA_PADDRx register. Data will be transferred from this address to memory after each I2C_STS1.FIFOHE event.
5. Set the memory address in the DMA_MADDRx register. Data will be transferred from I2C_FIFODAT to this memory area after each I2C_STS1.FIFOHE event. (When the remaining received data is below the waterline, a single DMA request is indicated by the FIFO not empty status and FIFOE.)
6. Set the number of bytes to be transferred in the DMA_TXNUMx register. After each I2C_STS1.FIFOHE event, this value decreases from the waterline value until 0.
7. Configure the channel priority level using the PRIOLVL bit in the DMA_CHCFGx register.
8. Set the DIR bit in the DMA_CHCFGx register, and configure interrupt request generation at half or full completion of the entire transfer as required by the application.
9. Activate the channel by setting the CHEN bit on the DMA_CHCFGx register.
10. When the specified number of data transfers set in the DMA controller is completed, the DMA controller sends an end of transfer (EOT/EOT_1) signal to the I2C interface. If interrupts are enabled, a DMA interrupt will be generated.

Note: If DMA is used for receiving, do not set I2C_CTRL2.BUFINTEN bit.

24.3.5 Packet Error Check (PEC)

Setting the I2C_CTRL1.PECEN bit to 1 enable the PEC function. PEC uses CRC-8 algorithm to calculate all information bytes including address and read/write bits. It can improve the reliability of communication. The CRC-8 polynomial uses by the PEC calculator is $C(x) = x^8 + x^2 + x + 1$.

In transmitting mode, software sets I2C_CTRL1.PEC transfer bit in the last I2C_STS1.TXDATE event, and then PEC will be transferred in the last byte. In receiving mode, software sets I2C_CTRL1.PEC transfer bit after the last I2C_STS1.RXDATNE event, and then receives the PEC byte and compares the received PEC byte to the internally calculated PEC value. If it is not equal to the internally calculated PEC value, the receiver needs to send a NACK. If it is host receiver mode, NACK will be sent after PEC regardless of the calculated result. It should pay attention that I2C_CTRL1.PEC bit has to be set before receiving.

If both DMA and PEC calculator are activated, I^2C will automatically send or check the PEC value.

In transmitting mode, when I^2C interface receives EOT signal from DMA controller, it will automatically send PEC following the last byte. In receiving mode, when I^2C interface receives an EOT_1 signal from DMA, it will automatically consider the next byte as PEC and compare it with the internally calculated PEC. It will happen a DMA request after receiving PEC.

In order to allow intermediate PEC transfer, I2C_CTRL2.DMALAST bit is used to determine whether it is the last DMA transfer. And if it does the last DMA request of the master receiver, NACK will be sent automatically after receiving the last byte.

When arbitration is lost, PEC calculation is invalid.

24.3.6 Noise Filtering

The I^{2}C interface standard requires the ability to filter spikes on the SCL/SDA lines with a duration of 50ns, hence analog and digital filters are designed. The analog filter is enabled by default and can also be disabled by setting the I2C_GFLTRCTRL.SCLAFENN/SDAAFENN bits. The analog filter sets the spike filter width to 5ns, 15ns, 25ns, 35ns by configuring the I2C_GFLTRCTRL.SCLAFE/SDAAFW.

The digital filter is enabled by setting I2C_GFLTRCTRL.SCLDFW/SDADFW to a non-zero value. The maximum filter width is (I2C_GFLTRCTRL.SCLDFW[3:0] or I2C_GFLTRCTRL.SDADFW[3:0]) * TPCLK. Enabling the digital filter will increase the hold time of the SDA line by an amount of (SDADFW[3:0]+1) * TPCLK.

24.3.7 SMBus

24.3.7.1 Introduction

The System Management Bus(SMBus or SMB) is a simple single-ended two-wire bus structure. Using SMBus can communicate with other device or other parts of the system, it able to commnicate with multiple devices without other independent control wire. SMBus is a derivate of the I^{2}C bus and provides a control bus for system and power management related tasks. If you want browse more information, please refer to the SMBus specification V2.0 (http://smbus.org/specs/).

SMBus have three types of device standard.

- Master: device send command, generate clocks and stop transmmissions;
- Slave: device receive, respond to commands;
- Host: system have only one host. A device provides a master to system CPU. Host have functions of master and slave, it supports SMBus alert protocol.

SMBus is a subset of the data transmission format of the I^{2}C specification.

Similarities between SMBus and I^{2}C:

- Both bus protocols contain of 2 wires (a clock wire SCL and a data wire SDA), with an optional SMBus alert wire.
- The data format is similar. SMBus data format is similar to 7-bit address format of I2C(See Figure 24-2).
- Both are master-slave communication modes, and the master device provides the clock.
- Both support multi master.

Differences between SMBus and I^{2}C:

Table 24-1 Comparison Between SMBus And I2C

SMBus	I^{2}C
Maximum transmission speed 100kHz	Maximum transmission speed 1MHz
Minimum transmission speed 10kHz	No minimum transmission speed
Low clock timeout 35ms	No clock timeout
Fixed logic level	V_{DD} determined logic level

SMBus	I^2C
Different address types (reserved, dynamic, etc.)	7-bit, 10-bit, and broadcast call slave address types
Different bus protocols (quick command, call handling, etc.)	No bus protocol

24.3.7.2 Purpose of SMBus

SMBus uses the system management bus to meet lightweight communication requirements. In general, SMBus is commonly used on the computer motherboard. It is mainly used to transmit ON/OFF instructions for power unit and provide a control bus for system and power management-related tasks.

24.3.7.3 Device identification

In SMBus, any device acting as a slave device has an address called the slave address.

In order to distribute addres for each devices, it must have a unique device identifier(UDID) to distinguish devices.

24.3.7.4 Bus protocol

SMBus specification includes eight bus protocols. If user wants browse the details on protocols or SMBus address types,it can refer to the SMBus specification v2.0(http://smbus.org/specs/). User's software can device what protocols are implemented.

Every packet through the SMBus complies with the SMBus protocol predefined format. SMBus is a subset of the data transfer format of I^2C specification. As long as an I^2C device can be accessed through one of the SMBus protocols, it is considered to be SMBus compliant.

Note: SMBus does not support Quick command protocol.

24.3.7.5 Address resolution protocol (ARP)

The SMBus resolves address conflicts by dynamically assigning a new unique address to each slave device. This is the address resolution protocol(ARP) .

Any master device can connected bus to access all devices.

SMBus physical layer arbitration enable to distribute addresses.When device powers on, the device's distribute address is not change, the protocol allows address retain when device power off.

When address is distributed, there is no extra SMBus packaging cost (the cost time that access distribute address device and access fixed address device is same).

24.3.7.6 Timeout function

SMBus has a timeout feature: if it has taken too long time during the communication, it automatically resets the device. This is the reason why SMBus has a minimum transmission rate limitation: to prevent the bus from locking up for a long time after the timeout occurs. I^2C bus is essentially a "DC" bus, that is to say, if the slave is executing some subroutines and cannot respond in time while the master is accessing the slave, it can hold the clock. That can remind the host that the slave is busy but does not want to give up the current communication. This session can continue after the current task of the slave is over. I2c doesn't have a maximum limitation for the delay, but it is limited to 35ms in the SMBus system. According to the SMBus protocol, if a session takes too long, it means something is wrong with the bus, and all devices should be reset to eliminate this state. Like this, the slave device is not allowed to pull the clock down for too long. I2C_STS1.TIMOUT bit indicates the status of this feature.

24.3.7.7 SMBus alter mode

SMBus offers a optional interrupt signal SMBALERT(like SCL and SDA, is a wire-AND signal) that devices use to extend their control capabilities at expense of a pin. SMBus broadcast call address often combine with SMBALERT. There are 2 bytes message about SMBus.

A device which only has slave function can set I2C_CTRL1.SMBALERT bit to indicate it want to communicate with host. The host handles the interrupt and accesses all SMBALERT devices through the ARA (Alert Response Address, address value 0001100x). Only those devices that pull SMBALERT low can respond to ARA. This state is identified by the I2C_STS1.SMBALERT. The 7-bit device address provided from the sending device is placed on the 7 most significant bits of the byte, the eighth bit can be either '0' or '1'.

When more than one device's SMBALERT is low, the highest priority(The smaller the address, the higher the priority) can win bus communication through the standard arbitration during address transmission. If confirming the slave address, device's SMBALERT is no longer pulled low. If message transmitted completely, device's SMBALERT still is low, it means host will read ARA again. The host can periodically access the ARA when the SMBALERT signal is not used.

24.3.7.8 SMBus communication process

The communication process on SMBus is similar to that on I^{2}C. To use the SMBus mode, you need to configure SMBus specific registers in the program, respond and process SMBus specific flag, and implement the upper-layer protocols described in the SMBus manual.

1. At first, set I2C_CTRL1.SMBMODE bit, and configure I2C_CTRL1.SMBTYPE bit and I2C_CTRL1.ARPEN bit according to the application requirements. If I2C_CTRL1.ARPEN=1 and I2C_CTRL1.SMBTYPE=0, use the default address of the SMB device. If I2C_CTRL1.ARPEN=1 and I2C_CTRL1.SMBTYPE=1, use the SMB master header field.
2. In order to support ARP (I2C_CTRL1.ARPEN=1), in SMBus host mode (I2C_CTRL1.SMBTYPE=1), software needs to respond to the I2C_STS2.SMBHADDR bit (in SMBus slave mode, respond to I2C_STS2.SMBDADDR bit) and implement the functions according to the ARP protocol.
3. To support the SMBus warning mode, software should respond to the I2C_STS1.SMBALERT bit and implement the corresponding functions.

24.4 Debug Mode

When the microcontroller enters the debug mode (Cortex^{TM}-M4F core is in the stop state), configure the DBG_CTRL.I2CxSMBUS_TIMEOUT bit in the DBG module, select SMBUS timeout to continue normal work or stop. See Section 37.3.2 for details.

24.5 Bus Abnormal Timeout Error

For I^{2}C communication mode, to address bus hang situations caused by operational errors or hardware faults, a timeout error interrupt for abnormal I^{2}C bus levels has been introduced.

When the device is communicating via the I^{2}C bus, if the clock line or data line remains at a low level for a certain period of time, or if the clock line remains at a high level for a certain period of time, a bus fault is considered to have occurred. In this case, hardware intervention is required to interrupt the communication, as a slave device needs to reset communication and release bus control, while a master device needs to issue a stop condition.

The low timeout and high timeout can be enabled by configuring I2C_CTRL2. TOLEN and I2C_CTRL2. TOHEN respectively. The low timeout time and high timeout time can be selected by configuring I2C_CTRL1. LTC[1:0] and I2C_CTRL1. HTC[1:0] respectively. The low timeout time options include 25ms, 100ms, 1s, and 4s, while the high timeout time options include 256us, 512us, 1ms, and 128us.

If high or low level timeout is enabled, a timeout error will be generated after the bus level remains for the respective timeout time. If error interrupt enable and timeout interrupt enable are configured (I2C_CTRL2.SCLTOLIEN: SCL low timeout interrupt enable; I2C_CTRL2.SCLTOHIEN: SCL high timeout interrupt enable; I2C_CTRL2.SDATOLIEN: SDA low timeout interrupt enable), an error interrupt signal will also be generated.

24.6 Monitoring Mode

In monitoring mode, all communication on the I^{2}C bus can be observed. Monitoring mode can be enabled by configuring the two high-level bits I2C_CTRL1.NMENA and I2C_CTRL1.MTHALL. Other configurations can follow the communication reception configuration for slave devices.

In monitoring mode, the device will receive all data on the I^{2}C bus in slave receive mode. The monitoring device does not need to control the bus, so the SDA and SCL of the monitoring device will be forced to pull up. When the device is in monitoring mode, it will forcibly match all addressing on the bus, and all interrupts, including address match interrupts, will appear normally. The processor can view the actual data sent on the bus by reading the data registers of the monitoring device.

24.7 Interrupt Request

All I^{2}C interrupt requests are listed in the following table.

Table 24-2 I2C Interrupt Request

Interrupt Function	Interrupt Event	Event Flag	Set Control Bit
I2C event interrupt	Start bit sent (master)	STARTBF	EVTINTEN
	Address sent (master) or address matched (slave)	ADDRF	
	10-bit header sent (master)	ADDR10F	
	Received stop (slave)	STOPF	
	Data byte transfer completed.	BSF	
	Receive buffer is not empty.	RXDATNE	EVTINTEN and BUFINTEN
	Send buffer is empty.	TXDATE	
	FIFO empty	FIFOE	FIFOEINTEN and FIFOEN
	FIFO half empty	FIFOHE	FIFOHEINTEN and FIFOEN
	FIFO full	FIFOF	FIFOFINTEN and FIFOEN
	FIFO half full	FIFOHF	FIFOHFINTEN and FIFOEN
I2C error interrupt	Bus error	BUSERR	ERRINTEN
	Lost arbitration (master)	ARLOST	

Interrupt Function	Interrupt Event	Event Flag	Set Control Bit
	Acknowledge fail	ACKFAIL	
	Overrun/underrun	OVERRUN	
	PEC error	PECERR	
	Timeout /Tlow error	SMBALERT	
	Timeout error	SCLLTO	SCLTOLIEN
		SCLHTO	SCLTOHIEN
		SDALTO	SDATOLIEN
	FIFO write overflow error	FIFOWRERR	FIFOWREIEN
	FIFO read underflow error	FIFORDERR	FIFORDEIEN
	DMA underrun error	DMAETOERR	DMAETOEIEN

Notes:

(1) STARTBF, ADDRF, ADDR10F, STOPF, BSF, RXDATNE, TXDATE, FIFOE, FIFOHE, FIFOF, FIFOHF are logically ORed to the same interrupt channel.

(2) BUSERR, ARLOST, ACKFAIL, OVERRUN, PECERR, TIMOUT, SMBALERT, FIFOWRERR, FIFORDERR, DMAETOERR are logically ORed to the same interrupt channel.

24.8 I2C Registers

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

24.8.1 I2C Register Overview

Table 24-3 I²C Register Overview

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00h	I2C_CTRL1	Reserved	LTC[1:0]		HTC[1:0]		THRFIFOE[2:0]			Reserved	THRFIFOF[2:0]			Reserved		FIFOCLR	FIFOEN	NMENA	MTHALL	SWRESET	ALERT	PEC	ACKPOS	ACKEN	STOPGEN	STARTGE	NOEXTEN	GCEN	PECEN	ARPEN	SMBTYPE	SMMODE	EN
	Reset value		0	0	0	0	1	0	0		1	0	0			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x04h	I2C_CTRL2	Reserved	TOLEN	TOHEN	SCLTOLIEN	SCLTOHIEN	SDATOLIEN	FIFONFINTE	FIFONEINTE	DMAFIFOE	DMAETOEIE	FIFOWREIE	FIFORDEIE	FIFOHEINTE	FIFOHFINTE	FIFOEINTEN	FIFOFINTEN	DMAEN	ERRINTEN	EVTINTEN	BUFINTEN	Reserved			DMALAST	CLKFREQ[7:0]							
	Reset value		1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				0	0	0	0	0	0	0	0	0
0x08h	I2C_OADDR1	Reserved																ADDR	Reserved					ADDR1[1:0]		ADDR[6:0]							ADDR0
	Reset value																	0						0	0	0	0	0	0	0	0	0	0
0x0ch	I2C_OADDR2	Reserved																								ADDR2[6:0]							DUALEN
	Reset value																									0	0	0	0	0	0	0	0
0x10h	I2C_DAT	Reserved																								DATA[7:0]							
	Reset value																									0	0	0	0	0	0	0	0

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x14h	I2C_STS1	Reserved						SCLLTO	SCLHTO	SDALTO	DMAETOERR	FIFOWRERR	FIFORDERR	FIFOHE	FIFOHF	FIFOE	FIFOF	Reserved	SMBALERT	Reserved	PECERR	OVERRUN	BUSERR	ARLOST	ACKFAIL	Reserved	ADDR10F	TXDATE	RXDATNE	STOPF	BSF	ADDRF	STARTBF
	Reset value							0	0	0	0	0	0	0	0	0	0		0		0	0	0	0	0		0	0	0	0	0	0	0
0x18h	I2C_STS2	Reserved																PECVAL[7:0]								SMBHADDR	SMBDADDR	DUALFLAG	GCALLADDR	Reserved	TRF	MSMODE	BUSY
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0		0	0	0
0x1ch	I2C_CLKCTRL	Reserved																DUTY	FSMODE	Reserved		CLKCTRL[11:0]											
	Reset value																	0	0			0	0	0	0	0	0	0	0	0	0	0	0
0x20h	I2C_TMRISE	Reserved																								TMRISE[7:0]							
	Reset value																									0	0	0	0	0	0	1	0
0x24h	I2C_BYTENUM	Reserved																BTNUMEN	STASTOLB	BYTENUM[13:0]													
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x28h	I2C_GFLTRCTRL	Reserved																SCLAFENN	Reserved	SCLAFW[1:0]		SDAAFENN	Reserved	SDAAFW[1:0]		SCLDFW[3:0]				SDADFW[3:0]			
	Reset value																	0		1	0	0		1	0	0	0	0	0	0	0	0	0
0x2ch	I2C_FIFODAT	Reserved																								FIFODAT[7:0]							
	Reset value																									0	0	0	0	0	0	0	0

24.8.2 I2C Control Register 1 (I2C_CTRL1)

Address offset: 0x00

Reset value: 0x0440 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved	LTC[1:0]		HTC[1:0]		THRFIFOE[2:0]			Reserved	THRFIFOF[2:0]			Reserved		FIFOCLR	FIFOEN
	rw		rw		rw				rw					rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NMENA	MTHALL	SWRESET	ALERT	PEC	ACKPOS	ACKEN	STOPGEN	STARTGEN	NOEXTEND	GCEN	PECEN	ARPEN	SMBTYPE	SMMODE	EN
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit Field	Name	Description
31	Reserved	Reserved, the reset value must be maintained.
30:29	LTC[1:0]	Low Timeout Time Selection

Bit Field	Name	Description

		00:25ms;
		01:100ms;
		10:1s;
		11:4s.
		Note: Only used for I2C mode.

28:27	HTC[1:0]	High Timeout Time Selection
		00:256us;
		01:512us;
		10:1000us;
		11:128us.
		Note: Only used for I2C mode.

26:24	THRFIFOE[2:0]	FIFO Empty Watermark. When the number of FIFO data is less than or equal to THRFIFOE and not equal to 0, the half-empty flag is set
23	Reserved	Reserved, the reset value must be maintained.
22:20	THRFIFOF[2:0]	FIFO Full Watermark. When the number of FIFO data is greater than or equal to THRFIFOF and not equal to 8, the half-full flag is set
19:18	Reserved	Reserved, the reset value must be maintained.
17	FIFOCLR	FIFO Clear Bit. 1: FIFO Initialize, 0: FIFO Do Not Initialize
16	FIFOEN	FIFO Enable Bit. 1: Enable, 0: Disable
15	NMENA	Monitoring Enable Bit: 0: I2C Normal Operation Mode; 1: I2C Monitoring Mode
14	MTHALL	Global Address Match Bit. 0: Compares the received 7-bit address with the data in the address register, sets the address match flag and generates an interrupt if there is a match; 1: Sets the address match flag and generates an interrupt regardless of whether the received address matches the data in the address register
13	SWRESET	Software reset Make sure the I2C bus is idle before resetting this bit. 0:I2C not reset; 1:I2C reset. *Note: This bit can be used when the I2C_STS2.BUSY bit is set to 1 and no stop condition is detected on the bus.*
12	SMBALERT	SMBus alert It can be set or cleared by software. When I2C_CTRL1.EN=0, it will be cleared by hardware. 0: SMBAlert pin go high. The response address header is followed by the NACK signal;

Bit Field	Name	Description
		1: SMBAlert pin go low. The response address header is followed by the ACK signal.
11	PEC	Packet error checking It can be set or cleared by software. It will be cleared by hardware when PEC has been transferred, or by start or stop condition, or when I2C_CTRL1.EN=0. 0: No PEC transfer 1: PEC transfer. *Note: When arbitration is lost, the calculation of PEC is invalid.*
10	ACKPOS	Acknowledge/PEC Position（for data reception） It can be set or cleared by software. Or when I2C_CTRL1.EN=0, it will be cleared by hardware. 0: I2C_CTRL1.ACKEN bit determines whether to send an ACK to the byte currently being received; I2C_CTRL1.PEC bit indicates that the byte in the current shift register is PEC. 1: I2C_CTRL1.ACKEN bit etermines whether to send an ACK to the next received byte; I2C_CTRL1.PEC bit indicates that the next byte received in the shift register is PEC. *Note：* *ACKPOS bit can only be used in 2-byte receiving configuration and must be configured before receiving data.* *For the second byte of NACK, the I2C_CTRL1.ACKEN bit must be cleared after the I2C_STS1.ADDRF bit is cleared.* *To detect the PEC of the second byte, the I2C_CTRL1.PEC bit must be set after the ACKPOS bit is configured and when the ADDR event is extended.*
9	ACKEN	Acknowledge enable It can be set or cleared by software. Or when I2C_CTRL1.EN equals to 0, it will be cleared by hardware. 0：No acknowledge send; 1：Send an acknowledge after receiving a byte
8	STOPGEN	Stop generation It can be set or cleared by software. Or it will be cleared by hardware when a stop condition is detected. Or it will be set by hardware when SMBus timeout error is detected,. In the master mode: 0：No stop condition generates; 1：Generate a stop condition. In the slave mode: 0：No stop condition generates; 1：Release SCL and SDA lines after the current byte. *Note：When the STOPGEN, STARTGEN or PEC bit is set, the software should not take any write operation to I2C_CTRL1 until this bit is cleared by hardware. Otherwise, the STOPGEN, STARTGEN or PEC bits may be set twice.*
7	STARTGEN	Start generation It can be set or cleared by software. Or it will be cleared by hardware when the start condition is transferred or I2C_CTRL1.EN=0. 0: No start condition generates;

Bit Field	Name	Description
		1: Generate a start conditions.
6	NOEXTEND	Clock extending disable (Slave mode) This bit determines whether to pull SCL low when the data is not ready(I2C_STS1.ADDRF or I2C_STS1.BSF flag is set) in slave mode, and is cleared by software reset 0: Enable Clock extending. 1: Disable Clock extending.
5	GCEN	General call enable 0：Disable General call. not respond(NACK) to the address 00h; 1：Enable General call. respond(ACK) the address 00h.
4	PECEN	PEC enable 0：Disable PEC module; 1：Enable PEC module.
3	ARPEN	ARP enable 0: Disable ARP； 1: Enable ARP. If I2C_CTRL1.SMBTYPE=0, the default address of SMBus device is used. If I2C_CTRL1.SMBTYPE=1, the host address of SMBus is used.
2	SMBTYPE	SMBus type 0: Device 1: Host
1	SMMODE	SMBus mode 0：I2C mode； 1：SMBus mode.
0	EN	I2C Peripheral enable 0：Disable I2C module; 1：Enable I2C module *Note: If this bit is cleared when the communication is in progress, the I2C module is disabled and returns to the idle state after the current communication ends,all bits will be cleared.* *In master mode,this bit must never be cleared until the communication has ended.*

24.8.3 I2C Control Register 2 (I2C_CTRL2)

Address offset: 0x04

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved	TOLEN	TOHEN	SCLTOLIEN	SCLTOHIEN	SDATOLIEN	FIFONFINTEN	FIFONEINTEN	DMAFIFOEN	DMAETOEIEN	FIFOWREIEN	FIFORDEIEN	FIFOHEINTEN	FIFOHFINTEN	FIFOEINTEN	FIFOFINTEN
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMAEN	ERRINTEN	EVTINTEN	BUFINTEN	Reserved			DMALAST	CLKFREQ[7:0]							
rw	rw	rw	rw				rw	rw							

Bit Field	Name	Description
31	Reserved	Reserved, the reset value must be maintained.
30	TOLEN	Low Timeout Enable 0: Disable 1: Enable Default: Disable *Note: Only used for I2C mode.*
29	TOHEN	High Timeout Enable 0: Disable 1: Enable Default: Disable *Note: Only used for I2C mode.*
28	SCLTOLIEN	SCL Low Timeout Interrupt Enable 0: Disable 1: Enable Default: Disable *Note: Only used for I2C mode.*
27	SCLTOHIEN	I2C communication mode, configure SCL high timeout interrupt enable 0: Disable 1: Enable Default: Disable *Note: Only used for I2C mode.*
26	SDATOLIEN	SCL Low Extend Time Accumulation Timeout Interrupt Enable 0: Disable 1: Enable Default: Disable *Note: Only used for I2C mode.*
25	FIFONFINTEN	FIFO Not Full Interrupt Enable
24	FIFONEINTEN	FIFO Not Empty Interrupt Enable
23	DMAFIFOEN	DMA Enable when DMA Burst is enabled 0: Disable DMA request when FIFO is enabled 1: Enable DMA request when FIFO is enabled
22	DMAETOEIEN	DMA ETO Error Interrupt Enable
21	FIFOWREIEN	FIFO Overflow Interrupt Enable,
20	FIFORDEIEN	FIFO Underflow Interrupt Enable
19	FIFOHEINTEN	FIFO Half Empty Interrupt Enable
18	FIFOHFINTEN	FIFO Half Full Interrupt Enable
17	FIFOEINTEN	FIFO Empty Interrupt Enable
16	FIFOFINTEN	FIFO Full Interrupt Enable
15	DMAEN	DMA requests enable 0: Disable DMA 1: Enable DMA

Bit Field	Name	Description

14	ERRINTEN	Error interrupt enable
		0：Disable error interrupt；
		1：Enable error interrupt.
		This interrupt is generated when:
		I2C_STS1.BUSERR= 1
		I2C_STS1.ARLOST= 1
		I2C_STS1.ACKFAIL= 1
		I2C_STS1.OVERRUN= 1
		I2C_STS1.PECERR= 1
		I2C_STS1.TIMOUT= 1
		I2C_STS1.SMBALERT= 1

13	EVTINTEN	Event interrupt enable
		0：Disable event interrupt；
		1：Enable event interrupt
		This interrupt is generated when:
		I2C_STS1.STARTBF = 1 (Master)
		I2C_STS1.ADDR F = 1 (Master/Slave)
		I2C_STS1.ADD10F = 1 (Master)
		I2C_STS1.STOPF = 1 (Slave)
		I2C_STS1.BSF = 1 with no I2C_STS1.TXDATE or I2C_STS1.RXDATNE event
		I2C_STS1.TXDATE = 1 if I2C_CTRL2.BUFINTEN = 1
		I2C_STS1.RXDATNE = 1 if I2C_CTRL2.BUFINTEN = 1

12	BUFINTEN	Buffer interrupt enable 0: When I2C_STS1.TXDATE=1 or I2C_STS1.RXDATNE=1, any interrupt is not generated. 1: If I2C_CTRL2.EVTINTEN= 1,When I2C_STS1.TXDATE=1 or I2C_STS1.RXDATNE= 1, interrupt will be generated.
11:9	Reserved	Reserved, the reset value must be maintained.
8	DMALAST	DMA last transfer 0: Next DMA EOT is not the last transfer 1: Next DMA EOT is the last transfer *Note: This bit is used in the master receiving mode, so that a NACK can be generated when data is received for the last time.*
7:0	CLKFREQ[7:0]	I2C Peripheral clock frequency CLKFREQ[5:0] should be the APB clock frequency to generate the correct timming. 00000000：Disable 00000001：Disable 00000010：2MHz 00000011：3MHz … 01001000: 72 MHz ... 11111111: 255MHz

24.8.4 I2C Own Address Register 1 (I2C_OADDR1)

Address offset: 0x08

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDRMODE	Reserved					ADDR1[1:0]		ADDR[6:0]							ADDR0
rw						rw		rw							rw

Bit Field	**Name**	**Description**
31:16	Reserved	**Reserved, the reset value must be maintained.**
15	ADDRMODE	Addressing mode (slave mode) 0: 7-bit slave address 1: 10-bit slave address
14:10	Reserved	Reserved, the reset value must be maintained.
9:8	ADDR1[1:0]	Interface address 9~8 bit of the address. *Note: don't care these bits in 7-bit address mode*
7:1	ADDR[6:0]	Interface address 7~1 bits of the address.
0	ADDR0	Interface address 0 bit of the address. *Note: don't care these bits in 7-bit address mode*

24.8.5 I2C Own Address Register 2 (I2C_OADDR2)

Address offset: 0x0C

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								ADDR2[6:0]							DUALEN
								rw							rw

Bit Field	Name	Description
31:8	Reserved	Reserved, the reset value must be maintained.
7:1	ADDR2[6:0]	Interface address 7~1 bits of address in dual address mode.
0	DUALEN	Dual addressing mode enable 0: Disable dual address mode, only OADDR1 is recognized; 1: Enable dual address mode, both OADDR1 and OADDR2 are recognized. *Note: Valid only for 7-bit address mode*

24.8.6 I2C Data Register (I2C_DAT)

Address offset: 0x10

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								DATA[7:0]							
								rw							

Bit Field	Name	Description
31:8	Reserved	Reserved, the reset value must be maintained.
7:0	DATA[7:0]	8-bit data register Send or receive data buffer. *Note: In the slave mode, the address will not be copied into the data register;* *Note: if* I2C_STS1.*TXDATE =0, data can still be written into the data register;* *Note: If the ARLOST event occurs when processing the ACK pulse, the received byte will not be copied into the data register, so it cannot be read.*

24.8.7 I2C Status Register 1 (I2C_STS1)

Address offset: 0x14

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved						SCLLTO	SCLHTO	SDALTO	DMAETOERR	FIFOWRERR	FIFORDERR	FIFOHE	FIFOHF	FIFOE	FIFOF
						w0c	w0c	w0c	w0c	w0c	w0c	w0c	w0c	w0c	w0c

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	SMBALERT	Reserved	PECERR	OVERRUN	BUSERR	ARLOST	ACKFAIL	Reserved	ADDR10F	TXDATE	RXDATNE	STOPF	BSF	ADDRF	STARTBF

w0c w0c w0c w0c w0c w0c ro ro ro ro ro ro ro

Bit Field	Name	Description
31:26	Reserved	Reserved, the reset value must be maintained.
25	SCLLTO	SCL Low Timeout Error Flag 0: No SCL low timeout error occurred 1: SCL low timeout error occurred Clearable by software write or hardware clear. *Note: Only used for I2C mode.*
24	SCLHTO	SCL Timeout Error Flag: Set to 1 when a timeout error occurs, clearable by software write or hardware clear. I2C Mode 0: No SCL high timeout error occurred 1: SCL high timeout error occurred SMBus Mode 0: No timeout error; 1: Timeout error occurred Error Conditions: - SCL remains low for 25ms (Timeout); - Master's low level accumulated clock extend time exceeds 10ms (Tlow); - Slave device's low level accumulated clock extend time exceeds 25ms (Tlow)
23	SDALTO	SDA Low Extend Time Accumulation Timeout Error Flag 0: No SDA low timeout error occurred 1: SDA low timeout error occurred Clearable by software write or hardware clear. *Note: Only used for I2C mode.*
22	DMAETOERR	DMA ETO Error: In master or slave transmit mode and when DMA FIFO is enabled, if the DMA_ETO signal is not received during the PEC packet transmission phase, it indicates that the DMA transmission is completed but the ETO signal was not provided.
21	FIFOWRERR	Overflow Error: Writing data when FIFO is full.
20	FIFORDERR	Underflow Error: Reading data when FIFO is empty.
19	FIFOHE	FIFO Half Empty 0: Not occurred 1: Occurred
18	FIFOHF	FIFO Half Full 0: Not occurred 1: Occurred
17	FIFOE	FIFO Empty 0: Not occurred

Bit Field	Name	Description
		1: Occurred
16	FIFOF	FIFO Full 0: Not occurred 1: Occurred
15	Reserved	Reserved, the reset value must be maintained.
14	SMBALERT	SMBus alert Writing '0' to this bit by software can clear it, or it is cleared by hardware when I2C_CTRL1.EN=0. 0: No SMBus alert(host mode) or no SMB alert response address header sequence(slave mode); 1: SMBus alert event is generated on the pin(host mode) or receive SMBAlert response address(slave mode)
13	Reserved	Reserved, the reset value must be maintained.
12	PECERR	PEC Error in reception Writing '0' to this bit by software can clear it, or it is cleared by hardware when I2C_CTRL1.EN=0. 0: No PEC error 1: PEC error: receiver will returns NACK Whether the I2C_CTRL1.ACKEN bit is enabled
11	OVERRUN	Overrun/Underrun Writing '0' to this bit by software can clear it, or it is cleared by hardware when I2C_CTRL1.EN=0. 0: No Overrun/Underrun 1: Overrun/Underrun Set by hardware in slave mode when I2C_CTRL1.NOEXTEND=1, and when receiving a new byte in receiving mode, if the data within DAT register has not been read yet, over-run occurs,the new received byte will be lost.When transferring a new byte in transfer mode, but there is not new data that has not been written in DAT register, under-run occurs which leads that the same byte will be send twice.
10	BUSERR	Bus error Writing '0' to this bit by software can clear it, or it is cleared by hardware when I2C_CTRL1.EN=0. 0: No start or stop condition error 1: Start or stop condition error
9	ARLOST	Arbitration lost (master mode) Writing '0' to this bit by software can clear it, or it is cleared by hardware when I2C_CTRL1.EN=0. 0: No arbitration lost; 1: Arbitration lost. When the interface loses control of the bus to another host, the hardware will set this bit to '1', and the I2C interface will automatically switch back to slave mode (I2C_STS2.MSMODE=0).

Bit Field	Name	Description
		Note: In SMBUS mode, the arbitration of data in slave mode only occurs in the data stage or the acknowledge transfer interval (excluding the address acknowledge).
8	ACKFAIL	Acknowledge failure Writing ‘0’ to this bit by software can clear it, or it is cleared by hardware when I2C_CTRL1.EN=0. 0: No acknowledge failed; 1: Acknowledge failed.
7	Reserved	Reserved, the reset value must be maintained.
6	ADDR10F	10-bit header sent (Master mode) After the software reads the STS1 register, the operation of writing to the CTRL1 register will clear this bit, or when I2C_CTRL1.EN=0, the hardware will clear this bit. 0: No ADD10F event; 1: Master has sent the first address byte. In 10-bit address mode, when the master device has sent the first byte, the hardware sets this bit to' 1'. *Note: After receiving a NACK, the I2C_STS1.ADDR10F bit is not set.*
5	TXDATE	Data register empty (transmitters) Writing data to DAT register by software can clear this bit; Or after a start or stop condition occurs, or automatically cleared by hardware when I2C_CTRL1.EN=0. 0: Data register is not empty; 1: Data register is empty. When sending data, this bit is set to' 1' when the data register is empty, and it is not set at the address sending stage. If a NACK is received, or the next byte to be sent is PEC(I2C_CTRL1.PEC=1), this bit will not be set. *Note: After the first data to be sent is written, or data is written when BSF is set, the TXDATE bit cannot be cleared, because the data register is still empty.*
4	RXDATNE	Data register not empty(receivers) This bit is cleared by software reading and writing to the data register, or cleared by hardware when I2C_CTRL1.EN=0. 0: Data register is empty; 1: Data register is not empty. During receiving data, this bit is set to' 1' when the data register is not empty, and it is not set at the address receiving stage. RXDATNE is not set when the ARLOST event occurs. *Note: When BSF is set, the RXDATNE bit cannot be cleared when reading data, because the data register is still full.*
3	STOPF	Stop detection (slave mode) After the software reads the STS1 register, the operation of writing to the CTRL1 register will clear this bit, or when I2C_CTRL1.EN=0, the hardware will clear this bit. 0: No stop condition is detected; 1: Stop condition is detected.

Bit Field	Name	Description
		After a ACK, the hardware sets this bit to' 1' when the slave device detects a stop condition on the bus. *Note: I2C_STS1.STOPF bit is not set after receiving NACK.*
2	BSF	Byte transfer finished After the software reads the STS1 register, reading or writing the data register will clear this bit; Or after sending a start or stop condition in sending mode, or when I2C_CTRL1.EN=0, this bit is cleared by hardware. 0：Byte transfer does not finish. 1：Byte transfer finished. When I2C_CTRL1.NOEXTEND =0, the hardware sets this bit to' 1' in the following cases: In receiving mode, when a new byte (including ACK pulse) is received and the data register has not been read (I2C_STS1.RXDATNE=1).In sending mode, when a new data is to be transmitted and the data register has not been written with the new data (I2C_STS1.TXDATE=1). *Note: After receiving a NACK, the BSF bit will not be set.* *If the next byte to be transferred is PEC (I2C _STS2.TRF is' 1' and I2C_CTRL1.PEC is' 1'), the BSF bit will not be set.*
1	ADDRF	Address sent (master mode) / matched (slave mode) After the STS1 register is read by software, reading the STS2 register will clear this bit, or when I2C_CTRL1.EN=0, it will be cleared by hardware. 0: Address mismatch or no address received(slave mode) or Address sending did not end(master mode); 1: Received addresses matched(slave mode) or Address sending ends(master mode) In master mode: In 7-bit address mode, this bit is set to' 1' after receiving the ACK of the address.In 10-bit address mode, this bit is set to' 1' after receiving the ACK of the second byte of the address. In slave mode: Hardware sets this bit to' 1' (when the corresponding setting is enabled) when the received slave address matches the content in the OADDR register, or a general call or SMBus device default address or SMBus host or SMBus alter is recognized. *Note: After receiving NACK, the I2C_STS1.ADDRF bit will not be set.*
0	STARTBF	Start bit (Master mode) After the STS1 register is read by software, writing to the data register will clear this bit, or when I2C_CTRL1.EN=0, the hardware will clear this bit. 0: Start condition was not sent; 1: Start condition has been sent. This bit is set to' 1' when the start condition is sent.

24.8.8 I2C Status Register 2 (I2C_STS2)

Address offset: 0x18

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PECVAL[7:0]								SMBHADDR	SMBDADDR	DUALFLAG	GCALLADDR	Reserved	TRF	MSMODE	BUSY
ro								ro	ro	ro	ro		ro	ro	ro

Bit Field	Name	Description
31:16	Reserved	Reserved, the reset value must be maintained.
15:8	PECVAL[7:0]	Packet error checking register Stores the internal PEC value When I2C_CTRL1.PECEN =1.
7	SMBHADDR	SMBus host header（Slave mode） Hardware clears this bit when a stop condition or a repeated start condition is generated, or when I2C_CTRL1.EN=0. 0: SMBus host address was not received; 1: when I2C_CTRL1.SMBTYPE=1 and I2C_CTRL1.ARPEN=1, SMBus host address is received.
6	SMBDADDR	SMBus device default address（Slave mode） Hardware clears this bit when a stop condition or a repeated start condition is generated, or when I2C_CTRL1.EN=0. 0: The default address of SMBus device has not been received; 1: when I2C_CTRL1.ARPEN=1, the default address of SMBus device is received.
5	DUALFLAG	Dual flag(Slave mode) Hardware clears this bit when a stop condition or a repeated start condition is generated, or when I2C_CTRL1.EN=0. 0: Received address matches the content in OADDR1; 1: Received address matches the content in OADDR2.
4	GCALLADDR	General call address(Slave mode) Hardware clears this bit when a stop condition or a repeated start condition is generated, or when I2C_CTRL1.EN=0. 0: No general call address was received; 1: when I2C_CTRL1.GCEN=1, general call address was received.
3	Reserved	Reserved, the reset value must be maintained.
2	TRF	Transmitter/receiver After detecting the stop condition (I2C_STS1.STOPF=1), repeated start condition or bus arbitration loss (I2C_STS1.ARLOST=1), or when I2C_CTRL1.EN=0, the hardware clears

Bit Field	Name	Description

		it.
		0: Data receiving mode;
		1: Data transmission mode;
		At the end of the whole address transmission stage, this bit is set according to the R/W bit of the address byte.

1	MSMODE	Master/slave mode Hardware clears this bit when a stop condition is detected on the bus, arbitration is lost (I2C_STS1.ARLOST=1), or when I2C_CTRL1.EN=0. 0: In slave mode; 1: In master mode. When the interface is in the master mode (I2C_STS1.STARTBF=1), the hardware sets this bit;
0	BUSY	Bus busy Hardware clears this bit when a stop condition is detected. 0: No data communication on the bus; 1: Data communication on the bus. When detecting that SDA or SCL is low level, the hardware sets this bit to' 1'; *Note:This bit indicates the bus communication currently in progress, and this information is still updated when the interface is disabled (I2C_CTRL1.EN=0).*

24.8.9 I2C Clock Control Register (I2C_CLKCTRL)

Address offset: 0x1C

Reset value: 0x0000 0000

Note:

1. F_{PCLK1} is required to be an integer multiple of 10 MHz, so that a fast clock of 400KHz can be generated correctly.

2. The CLKCTRL register can only be set when I^2C is turned off (I2C_CTRL1.EN=0)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DUTY	FSMODE	Reserved		CLKCTRL[11:0]											
rw	rw									rw					

Bit Field	Name	Description
31:16	Reserved	Reserved, the reset value must be maintained.
15	DUTY	Duty cycle in fast mode

Bit Field	Name	Description
		0：Tlow/Thigh = 2； 1：Tlow/Thigh = 16/9
14	FSMODE	I2C master mode selection 0: I2C in standard mode(duty cycle defaults to 1/1); 1: I2C in fast mode(duty cycle can be configured).
13:12	Reserved	Reserved, the reset value must be maintained.
11:0	CLKCTRL[11:0]	Clock control register in Fast/Standard mode (Master mode) This division factor is used to set the SCL clock in the master mode. • If duty cycle = Tlow/Thigh = 1/1: CLKCTRL = f_{PCLK1}(Hz)/100000/2 Tlow = CLKCTRL ×T_{PCLK1} Thigh = CLKCTRL ×T_{PCLK1} • If duty cycle = Tlow/Thigh = 2/1: CLKCTRL = f_{PCLK1}(Hz)/100000/3 Tlow = 2 ×CLKCTRL ×TPCLK1 Thigh = CLKCTRL ×T_{PCLK1} • If duty cycle = Tlow/Thigh = 16/9: CLKCTRL = f_{PCLK1}(Hz)/100000/25 Tlow = 16 ×CLKCTRL ×T_{PCLK1} Thigh = 9 ×CLKCTRL ×T_{PCLK1} For example, if f_{PCLK1}(Hz) = 8MHz, duty cycle = 1/1, CLKCTRL = 8000000/100000/2 = 0x28. *Note: 1. The minimum setting value is 0x04 in standard mode and 0x01 in fast mode;* *2. T_{high}= $T_{r(SCL)}$+$T_{w(SCLH)}$. See the definitions of these parameters in the data sheet for details.* *3. T_{low}= $T_{f(SCL)}$+$T_{w(SCLL)}$, see the definitions of these parameters in the data sheet for details;* *4. These delays have no filters;*

24.8.10 I2C Rise Time Register (I2C_TMRISE)

Address offset: 0x20

Reset value: 0x0000 0002

Note: The I2C_TMRISE register function is only valid in master mode. changed when I2C is disabled (I2C_CTRL1.EN=0).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								TMRISE[7:0]							
								rw							

Bit Field	Name	Description
31:8	Reserved	Reserved, the reset value must be maintained.
7:0	TMRISE[7:0]	Maximum Rise Time in Fast/Standard Mode (Master Mode) These bits must be set to the maximum SCL rise time specified in the I2C bus specification, with an increment of 1. For example, the maximum allowed SCL rise time in Standard Mode is 1000ns. If the value in I2C_CTRL2.CLKFREQ[7:0] is 0x78 and TPCLK1=8.333ns, then 0x79 must be written to TMRISE[7:0] (1000ns/8.333 ns + 1). If the result is not an integer, write the integer part to TMRISE[7:0] to ensure the tHIGH parameter.

24.8.11 I2C Filter Control Register (I2C_BYTENUM)

Address offset: 0x24

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BTNUMEN	STASTOLB	BYTENUM [13:0]													
rw	rw	rw													

Bit Field	Name	Description
31:16	Reserved	Reserved, the reset value must be maintained.
15	BTNUMEN	Host Receive Byte Count Function Enable: 1: Enable, 0: Disable Default: Disabled
14	STASTOLB	When BTNUMEN is set to 1: 1: Upon completion of host receive, it will enter the process of sending NACK and RESTART, 0: Upon completion of host receive, it will enter the process of sending NACK and STOP. Default: Send STOP process
13:0	BYTENUM[13:0]	1: Host will receive byte count (byte count greater than or equal to 2) 0: Hold

24.8.12 I2C Digital Filter Control Register (I2C_GFLTRCTRL)

Address offset: 0x28

Reset value: 0x 0000 2200

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCLAFENN	Reserved	SCLAFW[1:0]		SDAAFENN	Reserved	SDAAFW[1:0]		SCLDFW[3:0]				SDADFW[3:0]			
rw		rw		rw		rw		rw				rw			

Bit Field	Name	Description
31:16	Reserved	Reserved, the reset value must be maintained.
15	SCLAFENN	SCL Analog Filter Enable. 0: Enable, 1: Disable
14	Reserved	Reserved, the reset value must be maintained.
13:12	SCLAFW[1:0]	SCL Analog Filter Width Selection. 00: 5ns 01: 15ns 10: 25ns 11: 35ns *Note: The range of analog filter variation is large and depends on different PVT settings*
11	SDAAFENN	SDA Analog Filter. 0: Enable, 1: Disable
10	Reserved	Reserved, the reset value must be maintained.
9:8	SDAAFW[1:0]	SDA Analog Filter Width Selection. 00: 5ns 01: 15ns 10: 25ns 11: 35ns *Note: The range of analog filter variation is large and depends on different PVT settings*
7:4	SCLDFW[3:0]	SCL Digital Filter Width Selection. 0000: Disable digital filter Other values: Number of PCLK cycles
3:0	SDADFW[3:0]	SDA Digital Filter Width Selection: 0000: Disable digital filter Other values: Number of PCLK cycles

24.8.13 I2C FIFO Data Register (I2C_FIFODAT)

Address offset: 0x2C

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								FIFODAT [7:0]							
								rw							

Bit Field	Name	Description
31:8	Reserved	Reserved, the reset value must be maintained.
7:0	FIFODAT[7:0]	FIFIO Data

25 Flexible Data-rate Controller Area Network (FDCAN)

25.1 Introduction

Provide 3 FDCAN modules, compliant with ISO 11898-1:2015 standard, supporting CAN 2.0A/B and CAN FD protocols, compatible with Bosch protocol not based on ISO standard.

The three FDCAN modules share a message RAM area used for receiving message filters, receive FIFO, receive buffer, transmit buffer, and transmit event FIFO. The message RAM is located in the internal SRAM of the MCU, with a configurable starting address and a maximum allocation of 4480 words (32-bit).

25.2 Main Features

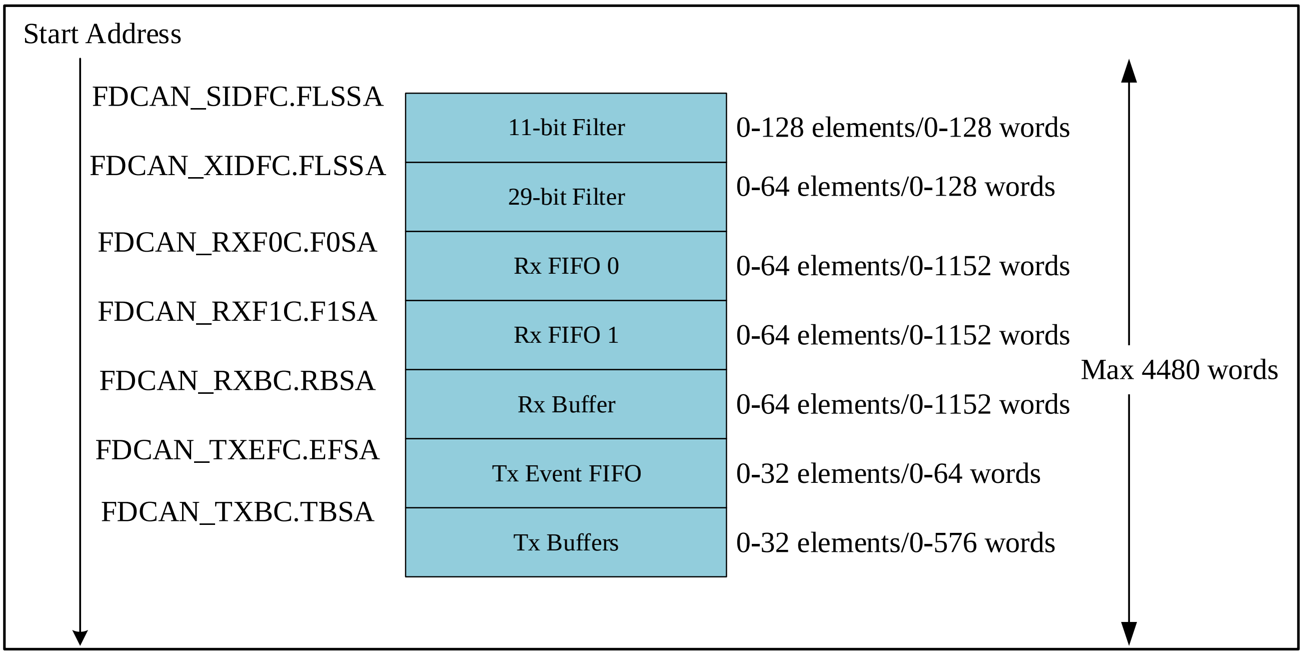
- Compliant with ISO 11898-1:2015 and ISO 11898-4 standards
- Support for CAN FD with up to 64 bytes of data
- Support for CAN error logging
- Support for AUTOSAR standard
- Support for SAE J1939 standard
- Enhanced receive filtering functionality
- Two configurable receive FIFOs
- Separate signaling for receiving high-priority messages
- Up to 64 dedicated receive buffers
- Up to 32 dedicated transmit buffers
- Configurable transmit FIFO or queue
- Configurable transmit event FIFO
- Support for configurable message RAM shared by three FDCAN controllers
- Programmable loopback test mode
- Maskable module interrupts
- Two clock domains: CAN core clock and APB bus clock
- Support for low-power mode

25.3 Message RAM

25.3.1 Message RAM Configuration

FDCAN supports a 32-bit Message RAM area, which includes receive filters, receive FIFOs, dedicated receive buffers, transmit event FIFO, and transmit buffers. The Message RAM is located in the general SRAM, with a configurable starting address (RCC_FDCANBADDR). It only supports access in words (32-bit) and not in bytes, with a maximum allocation of 4480 words, shared by three FDCAN modules. The structure of the Message RAM can be referred to in the following diagram.

Figure 25-1 Example diagram of FDCAN Message RAM allocation



The various segments in the Message RAM are not mandatory, and there are no restrictions on their placement order. Users can configure them as needed. The starting addresses and element counts for each segment can be configured using the registers FDCAN_SIDFC, FDCAN_XIDFC, FDCAN_RXESC.F0DS, FDCAN_RXESC.F1DS, FDCAN_RXESC.RBDS, and FDCAN_TXESC.TBDS. The starting addresses configured in the registers are in word addresses, with the actual values written to the registers being the offsets of the starting addresses of each segment from the starting address of the Message RAM area.

Note: FDCAN does not check for errors in the configuration of the Message RAM. Therefore, extra caution must be taken when configuring the starting addresses and element counts of each segment to avoid data corruption or loss.

25.3.2 Dedicated Receive Buffers and Receive FIFO

Each FDCAN module can configure up to 64 dedicated receive buffers and 2 receive FIFO areas in the Message RAM. The size of each receive FIFO can also be configured separately, with each able to store up to 64 received messages. The data field size of dedicated receive buffers and receive FIFO elements can be configured using the FDCAN_RXESC register, with a maximum of 64 bytes.

The data structure of receive buffers and FIFO elements is shown in the table below.

Table 25-1 Data Structure of Receive Buffers and Receive FIFO

word	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R0	ESI	XTD	RTR	ID[28:0]																												
R1	ANMF	FIDX[6:0]							Reserved		FDF	BRS	DLC[3:0]				RXTS[15:0]															
R2	DB3[7:0]								DB2[7:0]								DB1[7:0]								DB0[7:0]							
R3	DB7[7:0]								DB6[7:0]								DB5[7:0]								DB4[7:0]							
...	...								...								...								...							
Rn	DBm[7:0]								DBm-1[7:0]								DBm-2[7:0]								DBm-3[7:0]							

Bit Field	Name	Description
R0 bit 31	ESI	Error State Indicator. 0: The sending node is in an active error state 1: The sending node is in a passive error state
R0 bit 30	XTD	Extended Identifier, indicating the received frame ID type. 0: 11-bit standard ID 1: 29-bit extended ID
R0 bit 29	RTR	Remote Transmission Request, indicating the received frame type. 0: Received frame is a data frame 1: Received frame is a remote frame *Note: There are no remote frames in CAN FD format. In CAN FD frames (FDF = 1), the explicit bit RRS (Remote Request Substitute) replaces the RTR (Remote Transmission Request) bit.*
R0 bit 28:0	ID[28:0]	Remote Transmission Request (Identifier (Identifier) standard ID or extended ID, depending on the XTD bit. When it is an 11-bit standard ID, store it left-aligned (ID[28:18]).
R1 bit 31	ANMF	Accept Non-matching Frame. can be enabled to accept non-matching frames by FDCAN_GFC.ANFS and FDCAN_GFC.ANFE. 0: Received frame matches filter index FIDX 1: Received frame does not match any receive filter elements
R1 bit 30:24	FIDX[6:0]	Filter Index. 0-127: Index of the receive filter element that matches the received frame (invalid when ANMF = 1). Actual range is from 0 to FDCAN_SIDFC.LSS - 1 (standard frame) or FDCAN_XIDFC.LSE - 1 (extended frame).
R1 bit 21	FDF	FD Format 0: Typical CAN frame format 1: FDCAN frame format (new DLC encoding and CRC)
R1 bit 20	BRS	Bit Rate Switch 0: Do not switch bit rate when receiving frame

Bit Field	Name	Description
		1: Switch bit rate when receiving frame
R1 bit 19:16	DLC[3:0]	Data Length Code. 0-8: CAN + CAN FD: Received frame contains 0-8 data bytes 9-15: CAN: Received frame contains 8 data bytes 9-15: CAN FD: Received frame contains 12/16/20/24/32/48/64 data bytes
R1 bit 15:0	RXTS[15:0]	Rx Timestamp Timestamp counter value captured at the start of receiving the frame. The resolution depends on the configuration of the timestamp counter pre-scaler FDCAN_TSCC.TCP.
R2 bit 31:24	DB3[7:0]	Data Byte 3 (Data Byte 3)
R2 bit 23:16	DB2[7:0]	Data Byte 2 (Data Byte 2)
R2 bit 15:8	DB1[7:0]	Data Byte 1 (Data Byte 1)
R2 bit 7:0	DB0[7:0]	Data Byte 0 (Data Byte 0)
R3 bit 31:24	DB7[7:0]	Data Byte 7 (Data Byte 7)
R3 bit 23:16	DB6[7:0]	Data Byte 6 (Data Byte 6)
R3 bit 15:8	DB5[7:0]	Data Byte 5 (Data Byte 5)
R3 bit 7:0	DB4[7:0]	Data Byte 4 (Data Byte 4)
…	…	…
Rn bit 31:24	DBm[7:0]	Data Byte m (Data Byte m)
Rn bit 23:16	DBm-1[7:0]	Data Byte m-1 (Data Byte m-1)
Rn bit 15:8	DBm-2[7:0]	Data Byte m-2 (Data Byte m-2)
Rn bit 7:0	DBm-3[7:0]	Data Byte m-3 (Data Byte m-3)

Note: Depending on the configuration of FDCAN_RXESC, the data field can occupy 2 to 16 bytes (Rn= 2..17).

25.3.3 Transmission Buffer

The transmission buffer can be configured as dedicated transmission buffer as well as transmission FIFO/transmission queue. If the transmission buffer is shared by dedicated transmission buffer and transmission FIFO/transmission queue, the dedicated transmission buffer is allocated first from the start address of the transmission buffer, and the remaining part is allocated to the transmission FIFO/transmission queue. The transmission processing unit distinguishes between dedicated transmission buffer and transmission FIFO/transmission queue based on the values of FDCAN_TXBC.TFQS and FDCAN_TXBC.NDTB. The size of transmission buffer and transmission FIFO/transmission queue elements can be configured, depending on the data field size, which can be configured up to 64 bytes using register FDCAN_TXESC.

The data structure of dedicated transmission buffer and transmission FIFO/transmission queue elements is shown in the following table.

Table 25-2 Data Structure of Dedicated Transmission Buffer and Transmission FIFO/Queue

word	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
T0	ESI	XTD	RTR	ID[28:0]																												
T1	MM[7:0]								EFC	Reserved	FDF	BRS	DLC[3:0]				Reserved															
T2	DB3[7:0]								DB2[7:0]								DB1[7:0]								DB0[7:0]							
T3	DB7[7:0]								DB6[7:0]								DB5[7:0]								DB4[7:0]							
...	...								...								...								...							
Tn	DBm[7:0]								DBm-1[7:0]								DBm-2[7:0]								DBm-3[7:0]							

Bit Field	Name	Description
T0 bit 31	ESI	Error State Indicator. 0: In CAN FD frames, the ESI bit is solely dependent on the passive error flag 1: In CAN FD frames, the ESI bit will be sent recessive *Note: The actual ESI bit sent is the result of a logical OR operation between the ESI written into the transmit buffer and the passive error flag. As per the requirements of the CAN FD protocol specification, active error nodes may choose to send the ESI bit recessive, but passive error nodes will always send the ESI bit recessive.*
T0 bit 30	XTD	Extended Identifier. 0: 11-bit standard ID 1: 29-bit extended ID
T0 bit 29	RTR	Remote Transmission Request. 0: Send data frame 1: Send remote frame *Note: When RTR = 1, even if CCCR.FDOE enables CAN FD format transmission, FDCAN will still send remote frames according to the ISO 11898-1:2015 standard.*
T0 bit 28:0	ID[28:0]	Identifier. Standard ID or extended ID, depending on the XTD bit. When it is an 11-bit standard ID, store it left-aligned (ID[28:18]).
T1 bit 31:24	MM[7:0]	Message Marker. Written by the CPU when configuring the transmission buffer. Copied to the transmission event FIFO element to identify the transmission message status.
T1 bit 23	EFC	Event FIFO Control. 0: Do not store transmission events 1: Store transmission events
T1 bit 21	FDF	FD Format. 0: Send in typical CAN frame format 1: Send in CAN FD frame format
T1 bit 20	BRS	Bit Rate Switching. 0: Do not switch bit rate when sending CAN FD frames 1: Switch bit rate when sending CAN FD frames

Bit Field	Name	Description
		Note: ESI, FDF, and BRS bits are only valid when CAN FD is enabled (FDCAN_CCCR.FDOE = 1). Furthermore, BRS is only valid when FDCAN_CCCR.BRSE = 1.
T1 bit 19:16	DLC[3:0]	Data Length Code. 0-8: CAN + CAN FD: Received frame contains 0-8 data bytes 9-15: CAN: Received frame contains 8 data bytes 9-15: CAN FD: Received frame contains 12/16/20/24/32/48/64 data bytes
T2 bit 31:24	DB3[7:0]	Data Byte 3 (Data Byte 3)
T2 bit 23:16	DB2[7:0]	Data Byte 2 (Data Byte 2)
T2 bit 15:8	DB1[7:0]	Data Byte 1 (Data Byte 1)
T2 bit 7:0	DB0[7:0]	Data Byte 0 (Data Byte 0)
T3 bit 31:24	DB7[7:0]	Data Byte 7 (Data Byte 7)
T3 bit 23:16	DB6[7:0]	Data Byte 6 (Data Byte 6)
T3 bit 15:8	DB5[7:0]	Data Byte 5 (Data Byte 5)
T3 bit 7:0	DB4[7:0]	Data Byte 4 (Data Byte 4)
…	…	…
Tn bit 31:24	DBm[7:0]	Data Byte m (Data Byte m)
Tn bit 23:16	DBm-1[7:0]	Data Byte m-1 (Data Byte m-1)
Tn bit 15:8	DBm-2[7:0]	Data Byte m-2 (Data Byte m-2)
Tn bit 7:0	DBm-3[7:0]	Data Byte m-3 (Data Byte m-3)

Note: Depending on the configuration of FDCAN_RXESC, the data field can occupy 2 to 16 bytes (Rn= 2..17).

25.3.4 Transmission Events FIFO

Each element stores information related to a sent message. By reading the transmission event FIFO, the host CPU can retrieve relevant information in the order of message transmission. The status information of the transmission event FIFO can be obtained from register FDCAN_TXEFS.

The data structure of transmission event FIFO elements is shown in the following table.

Table 25-3 Transmission Event FIFO Data Structure

word	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E0	ESI	XTD	RTR	ID[28:0]																												
E1	MM[7:0]								ET[1:0]		FDF	BRS	DLC[3:0]				TXTS[15:0]															

Bit Field	Name	Description
E0 bit 31	ESI	Error State Indicator. 0: Sending node is in active error state 1: Sending node is in passive error state
E0 bit 30	XTD	Extended Identifier.

Bit Field	Name	Description
		0: 11-bit standard ID 1: 29-bit extended ID
E0 bit 29	RTR	Remote Transmission Request. 0: Send data frame 1: Send remote frame
E0 bit 28:0	ID[28:0]	Identifier. Standard ID or extended ID, depending on the XTD bit. When it is an 11-bit standard ID, store it left-aligned (ID[28:18]).
E1A/B bit 31:24	MM[7:0]	Message Marker. Copied from the transmission buffer to the transmission event FIFO element to identify the transmission message status.
E1A/B bit 23:22	ET[1:0]	Event Type. 00: Reserved 01: Transmission event 10: Transmit even after cancellation (must be set to this value when sending in DAR mode) 11: Reserve
E1A/B bit 21	FDF	FD Format. 0: Typical CAN frame format 1: FDCAN frame format (new DLC encoding and CRC
E1A/B bit 20	BRS	Bit Rate Switching. 0: Do not switch bit rate when sending CAN FD frames 1: Switch bit rate when sending CAN FD frames
E1A/B bit 19:16	DLC[3:0]	Data Length Code. 0-8: CAN + CAN FD: Received frame contains 0-8 data bytes 9-15: CAN: Received frame contains 8 data bytes 9-15: CAN FD: Received frame contains 12/16/20/24/32/48/64 data bytes
E1A bit 15:0	TXTS[15:0]	Tx Timestamp. Timestamp counter value captured at the start of sending the frame. The resolution depends on the configuration of the timestamp counter pre-scaler FDCAN_TSCC.TCP.

25.3.5 Standard Message ID Filter

Up to 128 filter elements can be configured for 11-bit standard IDs. When accessing standard message ID filter elements, the address is the start address of the standard filter list FDCAN_SIDFC.FLSSA plus the filter element index (0…127).

The data structure of standard message ID filter elements is shown in the following table.

Table 25-4 Data Structure of Standard Message ID Filter

word	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
S0	SFT[1:0]		SFEC[2:0]			SFID1[10:0]											Reserved					SFID2[0:0]										

Bit Field	Name	Description
31:30	SFT[1:0]	Standard Filter Type 00: Range filter from SFID1 to SFID2 01: Dual ID filter: Filter ID is SFID1 or SFID2 10: Typical filter: SFID1 is the filter ID, SFID2 is the mask 11: Disable current filter *Note: When SFT=11b, disable the filter element, but receive filtering remains active (same behavior as SFEC=00b).*
29:27	SFEC[2:0]	Standard Filter Element Configuration All enabled filter elements are used for receiving filtering of standard frames. When the first matching enabled filter element is found, or when the end of the filter list is reached, the receive filtering stops. If SFEC = 100b, 101b, or 110b, when a match is found, the interrupt flag FDCAN_IR.HPM is set to 1 and an interrupt is generated (if enabled). In this case, register FDCAN_HPMS is updated to reflect the priority matching state. 000: Disable current filter 001: Store in receive FIFO 0 if matched by filter 010: Store in receive FIFO 1 if matched by filter 011: Reject reception if matched by filter, not suitable for use with synchronous messages 100: Set priority if matched by filter but do not store, not suitable for use with synchronous messages 101: Set priority if matched by filter and store in FIFO 0 110: Set priority if matched by filter and store in FIFO 1 111: Store in receive buffer or as a debug message, disregarding the configuration of SFT[1:0]
26:16	SFID1[10:0]	Standard Filter ID 1. The first ID of the standard ID filter element. When used for receive buffer, synchronous messages, or debug messages, this field is the message ID to be stored. In this case, the message ID must match exactly, and no mask mechanism is used.
10:0	SFID2[10:0]	Standard Filter ID 2. The definition of this field varies depending on the configuration of SFEC. 1) SFEC = 001b...110b: The second ID of the standard ID filter element 2) SFEC = 111b: Filter configuration used for receive buffer or debug messages, defined as follows: - 10:9: Defines the location where the received message is stored, processed in the receive buffer, or as message A, B, or C in the debug message sequence

Bit Field	Name	Description
		00: Store the message in the receive buffer 01: Debug message A 10: Debug message B 11: Debug message C - 8:6: Reserved - 5:0: Offset of the received message storage location in the receive buffer from the start address FDCAN_RXBC.RBSA.

25.3.6 Extended Message ID Filter

Up to 64 filter elements can be configured for 29-bit extended IDs. When accessing extended message ID filter elements, the address is FDCAN_XIDFC.FLESA plus twice the index of the filter element (0…63).

The data structure of extended message ID filter elements is shown in the following table.

Table 25-5 Data Structure of Extended Message ID Filter

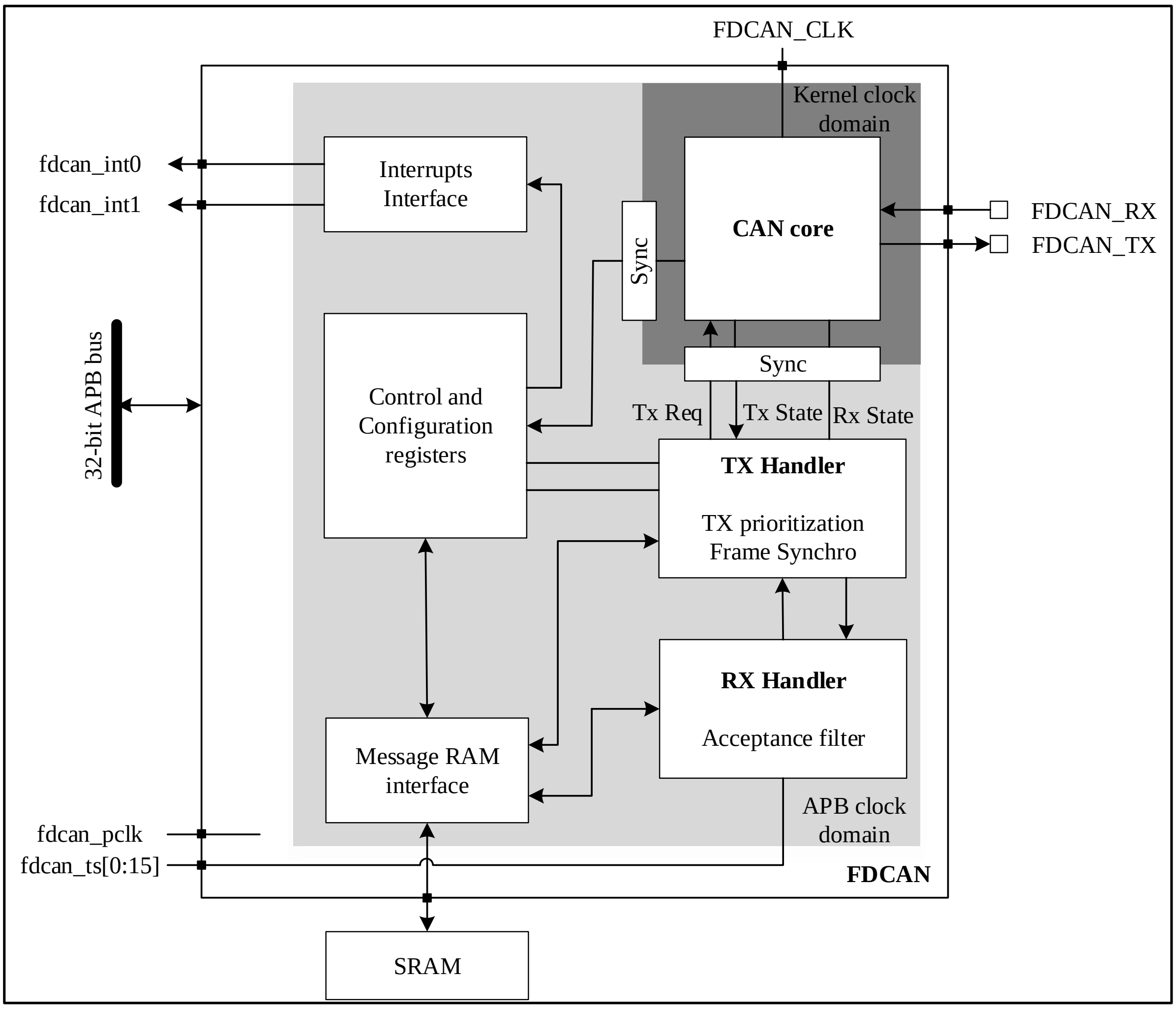
word	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
F0	EFEC[2:0]			EFID1[28:0]																												
F1	EFT[1:0]		Reserved	EFID2[28:0]																												

Bit Field	Name	Description
F0 bit 31:29	EFEC[2:0]	Extended Filter Element Configuration. All enabled filter elements are used for receiving filtering of extended frames. When the first matching enabled filter element is found, or when the end of the filter list is reached, the receive filtering stops. If EFEC = 100b, 101b, or 110b, when a match is found, the interrupt flag FDCAN_IR.HPM is set to 1 and an interrupt is generated (if enabled). In this case, register FDCAN.HPMS is updated to reflect the priority matching state. 000: Disable current filter 001: Store in receive FIFO 0 if matched by filter 010: Store in receive FIFO 1 if matched by filter 011: Reject reception if matched by filter, not suitable for use with synchronous messages 100: Set priority if matched by filter but do not store, not suitable for use with synchronous messages 101: Set priority if matched by filter and store in FIFO 0 110: Set priority if matched by filter and store in FIFO 1 111: Store in receive buffer or as a debug message, disregarding the configuration of

Bit Field	Name	Description
		EFT[1:0]
F0 bit 28:0	EFID1[28:0]	Extended Filter ID 1. The first ID of the extended ID filter element. When used for receive buffer, synchronous messages, or debug messages, this field is the message ID to be stored. In this case, the message ID must match exactly, and only the XIDAM masking mechanism is used.
F1 bit 31:30	EFT[1:0]	Extended Filter Type. 00: Range filter from EFID1 to EFID2 (EFID2 ≥ EFID1) 01: Dual ID filter: Filter ID is EFID1 or EFID2 10: Typical filter: EFID1 is the filter ID, EFID2 is the mask 11: Range filter from EFID1 to EFID2 (EFID2 ≥ EFID1), ignoring XIDAM masking
F1 bit 28:0	EFID2[28:0]	Extended Filter ID 2. Depending on the different configurations of EFEC, the definition of this field also varies. 1) EFEC = 001b...110b: The second ID of the extended ID filter element 2) EFEC = 111b: Filter configuration used for receive buffer or debug messages, defined as follows: - 10:9: Defines the location where the received message is stored, processed in the receive buffer, or as message A, B, or C in the debug message sequence 00: Store the message in the receive buffer 01: Debug message A 10: Debug message B 11: Debug message C - 8:6: Reserved - 5:0: Offset of the received message storage location in the receive buffer from the start address FDCAN_RXBC.RBSA.

25.4 Basic Function Description

Figure 25-2 FDCAN Functional Block Diagram



Interrupt Interface

The FDCAN module supports 2 interrupt lines output, fdcan_int0 and fdcan_int1, which can be enabled or disabled separately. All internal interrupts of the module can be connected to fdcan_int0 or fdcan_int1 and can be independently configured. By default, all interrupts are connected to the interrupt line fdcan_int0.

CAN Core

The CAN core comprises a CAN protocol controller and receive/transmit shift registers, providing all ISO 11898-1:2015 protocol functionalities, supporting both 11-bit and 29-bit identifiers.

Synchronization Unit

The synchronization unit synchronizes signals from the APB clock domain to the CAN core clock domain, or synchronizes signals from the CAN core clock domain to the APB clock domain.

Timestamp

Can receive an external 16-bit vector as a substitute for the internal 16-bit CAN time counter to generate receive and send timestamps.

Transmission Processing

The transmission processing unit controls the data transfer from message RAM to the CAN core. Up to 32 transmission buffers can be configured, which can be used as dedicated transmission buffers, transmission FIFO/queues, or a combination of dedicated transmission buffers and transmission FIFO/queues, with support for transmission cancellation. At any given time, the user can only select either a transmission queue or a transmission FIFO. Additionally, the configuration can include the generation of a transmission event FIFO during transmission, which includes transmission timestamps and associated message IDs.

Reception Processing

The receive processing unit controls the transfer of messages received from the CAN core to external message RAM. The receive processing unit supports two receive FIFOs, each with independently configurable sizes, and up to 64 dedicated receive buffers for storing all messages passing through the receive filters. Additionally, the receive processing unit supports dedicated receive buffers that store messages matching specific identifiers (IDs), different from receive FIFOs. All messages include receive timestamps. Up to 128 receive filters can be defined when using 11-bit IDs, and up to 64 filters can be defined for 29-bit IDs.

Message RAM Interface

The message RAM interface connects the FDCAN module to external 32-bit message RAM for message control and arbitration, shared by three FDCAN modules, supporting a maximum of 4480 words (32-bit).

25.4.1 Operating Modes

25.4.1.1 Software Initialization

When FDCAN_CCCR.INIT is set to 1, the module enters the software initialization state. At this point, both incoming and outgoing messages from the CAN bus will stop, and the state of the CAN bus output pin FDCAN_TX will be recessive (high level). The Error Management Logic (EML) counters remain unchanged, and no configuration register values will be altered. The FDCAN_CCCR.INIT bit can be set to 1 by software or automatically set to 1 when a hardware reset occurs or when entering the Bus_Off state. Clearing the FDCAN_CCCR.INIT bit will end the software initialization, and the module's Bit Stream Processor (BSP) will wait for an 11-bit recessive (Bus_Idle) sequence to synchronize with data transmission on the CAN bus before participating in bus activities and message transmission.

Write operations to the FDCAN configuration registers are only allowed when both the FDCAN_CCCR.INIT bit and FDCAN_CCCR.CCE bit are set to 1.

The FDCAN_CCCR.CCE bit can only be set to 1 or cleared when the FDCAN_CCCR.INIT bit is set to 1, and it is automatically cleared when the FDCAN_CCCR.INIT bit is cleared."

When FDCAN_CCCR.CCE is set to 1, the following registers will be reset:

- FDCAN_HPMS - High Priority Message Status
- FDCAN_RXF0S - Receive FIFO 0 Status
- FDCAN_RXF1S - Receive FIFO 1 Status
- FDCAN_TXFQS - Transmit FIFO/Queue Status
- FDCAN_TXBRP - Transmit Buffer Request Pending
- FDCAN_TXBTO - Transmit Buffer Transmission Occurred
- FDCAN_TXBCF - Transmit Buffer Cancellation Finished
- FDCAN_TXEFS - Transmit Event FIFO Status

When FDCAN_CCCR.CCE is set to 1, the FDCAN_TOCV.TOC bit is preset to the configuration value of FDCAN_TOCC.TOP. At this point, both the transmission processing unit and the reception processing unit state machines remain in idle state.

Write operations to the following registers are only allowed when the FDCAN_CCCR.CCE bit is cleared:

- FDCAN_TXBAR - Transmit Buffer Add Request
- FDCAN_TXBCR - Transmit Buffer Cancellation Request

The FDCAN_CCCR.TEST and FDCAN_CCCR.MON bits can only be set to 1 by software when both the FDCAN_CCCR.INIT and FDCAN_CCCR.CCE bits are set to 1, but can be cleared at any time.

The FDCAN_CCCR.DAR bit can only be set to 1 or cleared when both the FDCAN_CCCR.INIT and FDCAN_CCCR.CCE bits are set to 1.

25.4.1.2 Normal Operating Mode

After initialization is completed and the FDCAN_CCCR.INIT bit is cleared, FDCAN will synchronize with the CAN bus and prepare for communication.

Received messages (including message ID and DLC) filtered by the receive filters will be stored in Receive FIFO 0, Receive FIFO 1, or dedicated receive buffers.

Messages can be sent by initializing or updating dedicated transmission buffers and transmission FIFO/queues. Automatic transmission is not supported after receiving a remote frame.

25.4.1.3 CAN FD Operating Mode

The CAN FD protocol supports two frame modes. The first is the Extended Data Length (EDL) mode, in which the data field in a message frame can exceed eight bytes. The second is the Fast Frame Mode (FFM), where the bit rate of the control field, data field, and CRC field during message frame transmission is higher than that of the start and end fields. Fast Frame Mode can be used in conjunction with Extended Data Length mode.

The reserved bit in a CAN message frame with an 11-bit identifier and the first reserved bit in a CAN message frame with a 29-bit identifier are now used as the FDF bit. When FDF is recessive, it indicates a CAN FD message frame, while it indicates a typical CAN message frame when dominant. In a CAN FD message frame, the additional two bits after the FDF bit, res and BRS, determine whether a bit rate switch should occur. If res is dominant and BRS is recessive, it indicates that the current frame has a bit rate switch. If res is recessive, the functionality is undefined and reserved for future protocol extensions. If both FDF and res are recessive in a received frame, a protocol exception event occurs, setting the FDCAN_PSR.PXE bit to 1. If the protocol exception handling function is enabled (FDCAN_CCCR.PXHD = 0b), the module's operational state will transition from receiver (FDCAN_PSR.ACT = 10b) to synchronization (FDCAN_PSR.ACT = 00b) at the next sample point. If the protocol exception handling function is disabled (FDCAN_CCCR.PXHD = 1b), FDCAN will consider the recessive res bit as a formatting error and send an error frame.

By programming FDCAN_CCCR.FDOE, the CAN FD operating mode can be enabled, allowing the module to send and receive CAN FD message frames while also supporting the sending and receiving of typical CAN message frames. During transmission, the frame type (CAN FD message frame or typical CAN message frame) can be configured by setting the FDF bit in the corresponding transmission buffer. When FDCAN_CCCR.FDOE = 0b, received message frames are treated as typical CAN message frames, and if a CAN FD message frame is received, an error frame is sent. If the CAN FD functionality is disabled, even if the FDF bit in the transmission buffer is set to 1, CAN FD message frames will not be sent. Changes to FDCAN_CCCR.FDOE and FDCAN_CCCR.BRSE bits can only be made when both FDCAN_CCCR.INIT and FDCAN_CCCR.CCE are set to 1.

If FDCAN_CCCR.FDOE = 0b, the FDF and BRS bits are ignored, and only typical CAN message frames are sent. If FDCAN_CCCR.FDOE = 1b and FDCAN_CCCR.BRSE = 0b, the FDF bit is valid, allowing for the transmission of CAN FD message frames. When FDCAN_CCCR.FDOE = 1b and FDCAN_CCCR.BRSE = 1b, bit rate switching is supported when sending CAN FD message frames. For a transmission buffer where both the FDF and BRS bits are set to 1, the message is sent as a CAN FD message frame with bit rate switching enabled.

It is recommended to switch modes only when the following conditions are met:

- The error rate of the CAN FD data field is significantly higher than that of the CAN FD arbitration field. In this case, the bitrate switching function should be disabled.
- After system startup, all nodes only send typical CAN message frames. Only after confirming that the current bus can communicate via CAN FD message frames, all nodes switch to CAN FD mode.
- Wake-up messages in the CAN local network must be sent in the format of typical CAN message frames.
- During the vehicle offline programming process, not all nodes support CAN FD. Non-CAN FD nodes will remain in silent mode until programming is completed. Subsequently, all nodes will switch back to typical CAN communication.

In CAN FD mode, when the DLC value ranges from 0 to 8, the data field length definition is the same as in typical CAN. When the DLC value ranges from 9 to 15 (where the data field in typical CAN is always 8 bytes), the data field length definition is based on the following table.

Table 25-6 CAN FD data field length definition

DLC	9	10	11	12	13	14	15
Number of bytes in the data field	12	16	20	24	32	48	64

In CAN FD fast frame mode, if the BRS bit is recessive, the bit timing will automatically switch after the BRS bit. Before the BRS bit, the CAN FD arbitration field uses the standard bit timing defined by the nominal bit time and the prescaler register FDCAN_NBTP. In the subsequent CAN FD data field, the fast bit timing defined by the data bit time and the prescaler register FDCAN_DBTP is used. Finally, it switches back to standard bit timing at the CRC delimiter or when an error is detected (whichever event occurs first).

The maximum configurable bitrate of the CAN FD data field depends on the FDCAN core clock frequency. For example, if the FDCAN core clock frequency is 20MHz and the shortest configurable bit time is four time quanta (tq), then the bitrate of the data field is 5Mb/s.

In both the CAN FD long frame and fast frame data frame formats, the value of the ESI (Error State Indicator) bit is determined by the transmitter error state at the start of transmission. If the transmitter is in a passive error state, an recessive ESI bit occurs, otherwise, an dominant bit is sent. In CAN FD remote frames, the ESI bit is always dominant, regardless of the transmitter error state, and the data field length is 0.

25.4.1.4 Transceiver Delay Compensation

When sending data in the CAN FD data field, only one node is transmitting data, while all other nodes are in the receiver state. When the FDCAN transmitter sends data via the FDCAN_TX pin, the receiver simultaneously receives the data being sent via the FDCAN_RX pin. There is a delay in receiving the data relative to the transmission (transceiver loop delay). If the delay time is greater than TSEG1 (time segment before the sampling point in a single bit), bit errors occur. Without transceiver delay compensation, the data field bitrate will be limited by the loop delay.

FDCAN adopts a delay compensation mechanism to compensate for the transceiver loop delay, enabling stable transmission of the CAN FD data field at high bit rates without being affected by the loop delay.

To check for bit errors in the data field of the sending node, the delayed transmitted data is compared with the received data at the second sampling point SSP. If a bit error is detected, the transmitter will respond to the bit error at the next regular sampling point. Delay compensation is disabled in the arbitration field.

When the data bit time is less than the transmitter delay, FDCAN_DBTP.TDC can be set to 1 to enable transmitter delay compensation, which is detailed in the new version of the ISO11898-1 standard.

Received data bits are compared with the transmitted data bits at the SSP. SSP is defined as the sum of the delay time measured from the FDCAN transmit pin FDCAN_TX to the receive pin FDCAN_RX and the transmitter delay compensation offset value (configured by FDCAN_TDCR.TDCO). Transmitter delay compensation is used to adjust the position of the SSP in the received bit (e.g., half a data field bit time). The position of the second sampling point is rounded down to the next mtq (minimum time quanta, which is one fdcan_tq_ck clock cycle).

FDCAN_PSR.TDCV is the actual transmitter delay compensation value. When FDCAN_CCCR.INIT is set to 1, the FDCAN_PSR.TDCV value is cleared to zero. If FDCAN_DBTP.TDC is set to 1, the FDCAN_PSR.TDCV value is updated each time a CAN FD message frame is transmitted.

When using transmitter delay compensation in FDCAN, the following boundary conditions must be considered:

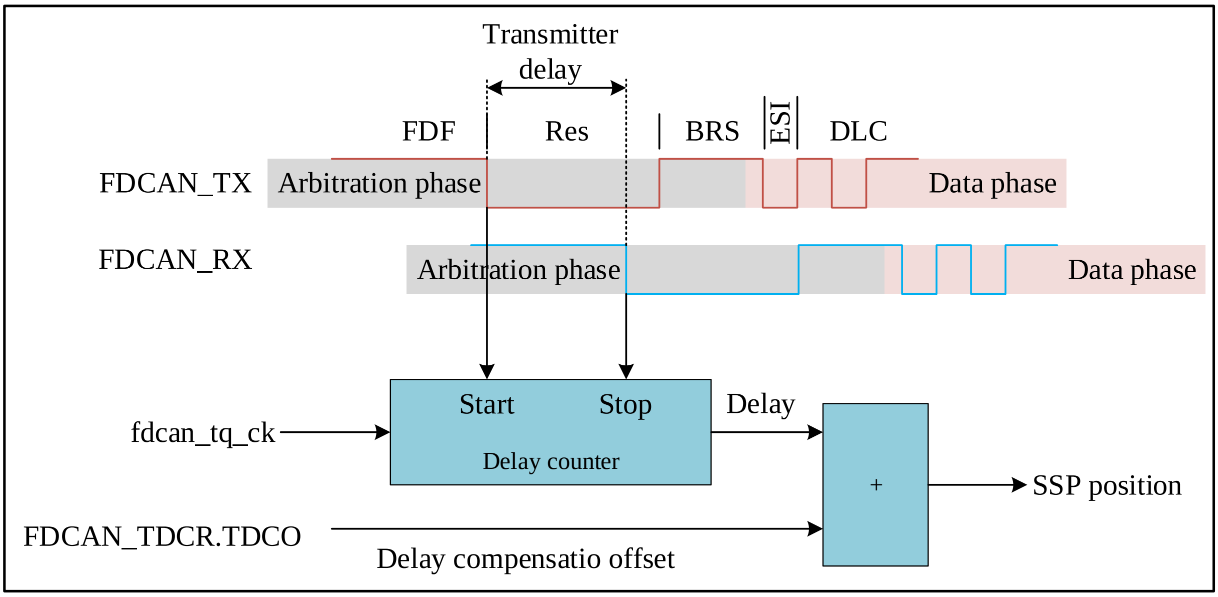
- The sum of the delay time measured between FDCAN_TX and FDCAN_RX and the transmitter delay compensation offset value FDCAN_TDCR.TDCO must be less than 6 bit times of the data field.

- The sum of the delay time measured between FDCAN_TX and FDCAN_RX and the transmitter delay compensation offset value FDCAN_TDCR.TDCO must be less than or equal to 127 mtq. If the sum exceeds 127 mtq, the transmitter delay compensation uses the maximum value (127 mtq).

- The data field ends at the sampling point of the CRC delimiter, at which point the reception of data bits stops at the SSP.

When FDCAN_DBTP.TDC is set to 1, the transmitter delay compensation is enabled. At this time, the measurement starts at the falling edge between the FDF bit and the res bit in each transmitted CAN FD message frame, and stops when this falling edge is observed on the receive pin FDCAN-RX. This measurement is in units of mtq.

Figure 25-3 FDCAN Transceiver Delay Measurement



If there is a dominant glitch signal in the received FDF bit, it may cause an abnormal termination of the delay compensation measurement before the falling edge is received (resulting in an early SSP position). In this case, FDCAN_TDCR.TDCF can be set to 1 to enable the transmitter delay compensation filter window. The delay compensation filter window defines the minimum value of the SSP position, where the transmitter delay measurement will ignore the dominant edges received by FDCAN_RX that cause the SSP position to advance. The measurement stops when the SSP position is not less than the FDCAN_TDCR.TDCF value, and FDCAN_RX is at a low level.

25.4.1.5 Restricted Operating Mode

In Restricted Operating Mode, a node can receive data frames and remote frames, acknowledge valid frames, but will not transmit data frames, remote frames, active error frames, or overload frames. In case of an error or overload, the node will not send dominant bits but will wait for the bus to become idle to resynchronize with the CAN bus. The error counters (FDCAN_ECR.REC, FDCAN_ECR.TEC) are frozen, and the error log (FDCAN_ECR.CEL) remains active. When the FDCAN_CCCR.ASM bit is set to 1 by software, the FDCAN enters Restricted Operating Mode. The FDCAN_CCCR.ASM can only be set to 1 when both FDCAN_CCCR.CCE and FDCAN_CCCR.INIT are set to "1", but it can be cleared at any time by software.

When the transmit processing unit fails to read data from the message RAM in a timely manner, it will automatically enter Restricted Operating Mode. Software must reset the FDCAN_CCCR.ASM bit to exit Restricted Operating Mode.

Restricted Operating Mode can be used for adaptive CAN bitrate applications. In this mode, the application tests different bitrates and exits Restricted Operating Mode upon receiving valid frames.

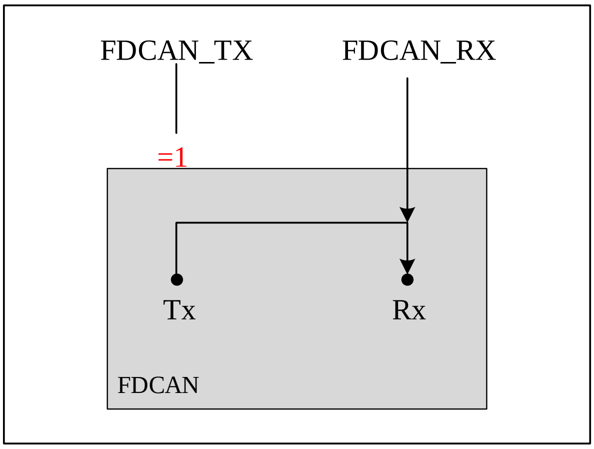
Note: Restricted Operating Mode cannot be combined with loopback mode (internal or external).

25.4.1.6 Bus Monitoring Mode (Silent Mode)

When FDCAN_CCCR.MON is set to 1, the FDCAN enters Bus Monitoring Mode. In Bus Monitoring Mode (for more detailed information, please refer to ISO11898-1, 10.12 Bus Monitoring), the FDCAN is able to receive valid data frames and valid remote frames, but cannot transmit. In this mode, the FDCAN will only send recessive bits on the CAN bus. If the FDCAN needs to send dominant bits (such as ACK bits, overload flags, active error flags), they are only sent internally within the module so that the FDCAN can monitor the dominant bit, while the CAN bus remains in the recessive state. In Bus Monitoring Mode, the FDCAN_TXBRP register will remain in the reset state.

Bus Monitoring Mode can be used to analyze the traffic on the CAN bus without causing interference by sending dominant bits. The connection of FDCAN_TX and FDCAN_RX signals to the FDCAN in Bus Monitoring Mode is shown in the diagram below.

Figure 25-4 FDCAN Bus Monitoring Mode Pin Control



25.4.1.7 Disable Automatic Retransmission (DAR) Mode

According to the CAN specification (see ISO 11898-1:2015, 8.3.4 Recovery Management), FDCAN supports automatic retransmission of message frames in case of arbitration failure or error disturbance during transmission. Automatic retransmission is enabled by default, but can be disabled by setting FDCAN_CCCR.DAR to disable automatic retransmission.

According to the CAN specification, in DAR (Disable Automatic Retransmission) mode, all retransmission operations are automatically canceled once the CAN bus is started. The transmit buffer request pending bit FDCAN_TXBRP.TRPx is automatically reset after a successful transmission, and it is also reset if the transmission has not started when canceled, is aborted due to arbitration loss, or encounters an error during transmission.

- Successful Transmission:

  - The transmit start bit corresponding to the transmit buffer is set to 1 in FDCAN_TXBTO.TOx
  - The cancel complete bit corresponding to the transmit buffer in FDCAN_TXBCF.CFx is not set to 1
- Transmission Canceled but Still Successful:
  - The transmit start bit corresponding to the transmit buffer is set to 1
  - The cancel complete bit corresponding to the transmit buffer in FDCAN_TXBCF.CFx is set to 1
- Arbitration Loss or Frame Transmission Disturbance:
  - The transmit start bit corresponding to the transmit buffer is not set to 1
  - The cancel complete bit corresponding to the transmit buffer in FDCAN_TXBCF.CFx is set to 1

When a message frame is successfully transmitted, if the transmit event storage is enabled, a transmit event FIFO entry of event type ET=10b (even if the transmission is canceled) will be written.

25.4.1.8 Power-down (Sleep) Mode

FDCAN can enter Power-down mode by setting the Clock Stop Request bit FDCAN_CCCR.CSR. Once the clock stop request is effective, reading the FDCAN_CCCR.CSR bit will return a value of 1.

After all transmit requests are completed, FDCAN will wait until it detects bus idle state. Then, FDCAN_CCCR.INIT is set to 1 to prevent further CAN transmissions. FDCAN_CCCR.CSA is then set to 1 to confirm readiness to enter Power-down mode, and the module clock can be turned off. In this mode, if the module clock is not turned off, register access can still continue, but write operations to FDCAN_CCCR.INIT are ineffective.

Note: In case of severe disturbances on the CAN bus, it may not be possible to detect bus idle state, causing FDCAN not to set the FDCAN_CCCR.INIT bit. This situation can be detected by polling FDCAN_PSR.ACT. In this case, software can set FDCAN_CCCR.INIT to 1 to immediately stop CAN communication, regardless of any ongoing transmit/receive operations.

To exit Power-down mode, the application must first enable the module clock, then reset the FDCAN_CCCR.CSR bit. Next, check the FDCAN_CCCR.CSA bit; if it has been reset, confirm that the clock is enabled. The application can then restart CAN communication by resetting the FDCAN_CCCR.INIT bit.

25.4.1.9 Test Mode

To write access the FDCAN_TEST register, FDCAN_CCCR.TEST must be set to 1 first. This allows configuration of the test mode and functions.

By configuring FDCAN_TEST.TX, four output functions can be implemented on the FDCAN transmit pin FDCAN_TX. In addition to the default serial data output function, the bit time can be monitored using CAN sampling point signals, and the output can hold either the dominant or recessive value. The actual status of the FDCAN_RX pin can be read from FDCAN_TEST.RX. Both functions can be used to check the CAN bus physical layer.)

Due to the synchronization mechanism between the CAN core clock domain and the host clock domain, there may be a delay of several APB clock cycles between the new configuration output on the FDCAN_TX output pin and the write operation to FDCAN_TEST.TX. This delay also applies when reading the FDCAN_RX input pin status via FDCAN_TEST.RX.

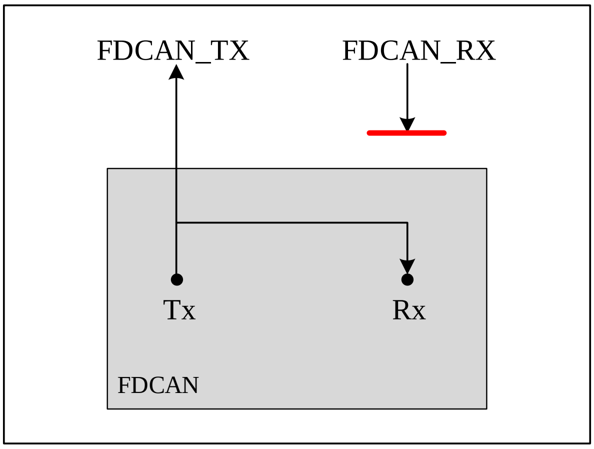
Note: The test mode is only intended for production testing or self-testing. Software control of the FDCAN_TX pin will disrupt all CAN protocol functions. It is not recommended to use the test mode in actual applications.

25.4.1.10 External Loopback Mode

Setting FDCAN_TEST.LBCK to 1 enables the FDCAN to be in External Loopback Mode. In this mode, the FDCAN processes its own transmitted messages as received messages and stores the messages (if they pass through the receive filters) in the receive FIFO. The connection of FDCAN_TX and FDCAN_RX signals to the FDCAN in External Loopback Mode is shown in the diagram below.

External Loopback Mode is used for hardware self-testing. To avoid external simulation interference, the FDCAN ignores ACK errors in loopback mode (ACK is recessive in data/remote frames). In this mode, the FDCAN achieves internal feedback from transmit output to receive input. The actual value on the FDCAN_RX input pin is disregarded. The transmitted messages can be monitored on the FDCAN_TX transmit pin. The pin control in External Loopback Mode is shown in the diagram below.

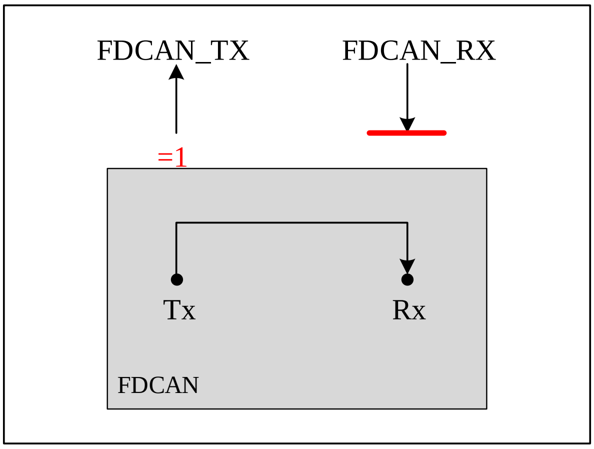
Figure 25-5 FDCAN External Loopback Mode Pin Control



25.4.1.11 Internal Loopback Mode

Setting both FDCAN_TEST.LBCK and FDCAN_CCCR.MON to 1 puts the FDCAN in Internal Loopback Mode. This mode can be used for "hot self-testing," allowing the FDCAN to perform diagnostics without affecting the CAN bus system connected to the FDCAN_TX and FDCAN_RX pins. In this mode, the FDCAN_RX pin is disconnected from the FDCAN, and the FDCAN_TX pin remains recessive. The pin control in Internal Loopback Mode is shown in the diagram below.

Figure 25-6 FDCAN Internal Loopback Mode Pin Control



25.4.1.12 Timestamp Generation

FDCAN provides a 16-bit loopback counter for timestamp generation. The counter clock source is the FDCAN's bit clock, and can be configured with clock pre-scaling (1...16) using FDCAN_TSCC.TCP. The counter value can be read from FDCAN_TSCV.TCV. Writing to the FDCAN_TSCV register resets the counter value to 0. When the counter loops, FDCAN_IR.TSW is set to 1.

When a message frame starts to be received/transmitted, the counter value is captured and stored in the timestamp of the Rx buffer, Rx FIFO, or Tx event FIFO. The 16-bit timestamp can be enabled by configuring the FDCAN_TSCC.TSS bit.

25.4.1.13 Timeout Counter

FDCAN provides a 16-bit timeout counter to indicate timeout conditions for Receive FIFO 0, Receive FIFO 1, and the Tx Event FIFO. The timeout counter is a decrementing counter, with its clock prescaling controlled by FDCAN_TSCC.TCP, similar to the timestamp counter. The timeout counter can be configured using the FDCAN_TOCC register. The counter value can be read from FDCAN_TOCV.TOC. The timeout counter can only be started when FDCAN_CCCR.INIT=0. When FDCAN_CCCR.INIT=1, the timeout counter stops counting, for example when FDCAN enters Bus_Off state.

The operation mode of the timeout counter is selected by FDCAN_TOCC.TOS. When configured in continuous mode, the counter starts counting once FDCAN_CCCR.INIT is reset. Writing to FDCAN_TOCV presets the counter value to FDCAN_TOCC.TOP and continues decrementing the count.

When the timeout counter is controlled by one of the FIFOs, the counter is preset to the FDCAN_TOCC.TOP value when the FIFO is empty. The decrementing count starts when the first element is written to the FIFO. Writing to FDCAN_TOCV does not affect the count value at this time.

When the counter value decrements to zero, FDCAN_IR.TOO is set to 1. If operating in continuous mode, the counter immediately restarts with the initial value set to FDCAN_TOCC.TOP.

Note: The clock signal of the timeout counter is derived from the sampling point signal of the CAN core. Therefore, due to the synchronization/resynchronization mechanism of the CAN core, the timing of the timeout counter decrement may vary. If bitrate switching is used in CAN FD, the clock of the timeout counter in the arbitration field and data field may differ.

25.4.2 Receive Processing

The receive processing unit controls the receive filters, transfers the received messages to dedicated receive buffers or specified receive FIFOs, and updates the write and fetch indexes of the receive FIFOs.

25.4.2.1 Receive Filters

FDCAN can configure two sets of receive filters, one for standard identifiers and the other for extended identifiers. These filters can be assigned to dedicated receive buffers or receive FIFO 0/1. When receiving a message, each filter list is checked starting from element 0 until the first matching element is found. Once a matching element is found, the check stops and subsequent filter elements are ignored.

Main features include:

- Each filter element can be configured as
  - Range filter
  - Single or dual specific ID match filter
  - Classic bit mask filter
- Each filter element can be configured to receive matched messages or reject matched messages
- Each filter element can be individually enabled/disabled
- Filters are checked one by one in list order, and the check stops after the first matching filter element is found.)

relevant configuration registers include:

- Global Filter Configuration FDCAN_GFC
- Standard ID Filter Configuration FDCAN_SIDFC
- Extended ID Filter Configuration FDCAN_XIDFC
- Extended ID and Mask FDCAN_XIDAM

Based on the context of digital chip design, the translation of the content within parentheses into English is: (Based on the configuration of the filter element (SFEC/EFEC), one of the following actions will be triggered after a match:

- Store the received frame in Receive FIFO 0 or FIFO 1
- Store the received frame in a dedicated receive buffer
- Store the received frame in a dedicated receive buffer and generate a pulse on the filter event pin
- Reject the received frame
- Set the high priority message interrupt flag FDCAN_IR.HPM
- Set the high priority message interrupt flag FDCAN_IR.HPM and store the received frame in Receive FIFO 0

or FIFO 1)

After the complete ID reception, the receive filtering begins execution. Upon completion of the receive filtering, if a matching dedicated receive buffer or receive FIFO is found, the message processing unit will write the received message data in 32-bit form to the matching dedicated receive buffer or receive FIFO. If the CAN protocol controller detects an error condition (such as CRC error), the message will be discarded, resulting in the following effects:

- Dedicated Receive Buffer

The new message flag of the matching dedicated receive buffer will not be set to 1, but the newly received data will overwrite a portion of it. Error types refer to FDCAN_PSR.LEC or FDCAN_PSR.DLEC.

- Receive FIFO

The write index of the matching receive FIFO will not be updated, but the newly received data will overwrite a portion of it. Error types refer to FDCAN_PSR.LEC or FDCAN_PSR.DLEC. If the matching receive FIFO operates in overwrite mode, the boundary conditions described in the receive FIFO overwrite mode must be considered.

Note: When a received message is written to the receive FIFO or dedicated receive buffer, the unaltered receive ID will be stored, regardless of the filter used. The outcome of receive filtering largely depends on the order of configured filter elements.

25.4.2.1.1 Range Filtering

When using range filtering, the filter matches all message IDs within the range defined by SF1ID/SF2ID or EF1ID/EF2ID. For extended frames using range filtering, there are two possibilities:

- EFT=00b: Before applying range filtering, the received message ID is first ANDed with the extended ID mask (FDCAN_XIDAM).
- EFT=11b: Range filtering is directly performed, ignoring the extended ID mask (FDCAN_XIDAM).)

25.4.2.1.2 Specific ID Filtering

The filter can be configured to filter one or two specific message IDs. If filtering for a single specific message ID is required, the filter element must be configured as SFID1=SFID2 or EFID1=EFID2.

25.4.2.1.3 Classic Bit Mask Filtering

Classic bit mask filtering is used to mask certain bits of the received message ID to filter specific message groups. In this case, SFID1/EFID1 is used as the filter ID, while SFID2/EFID2 is used as the mask.

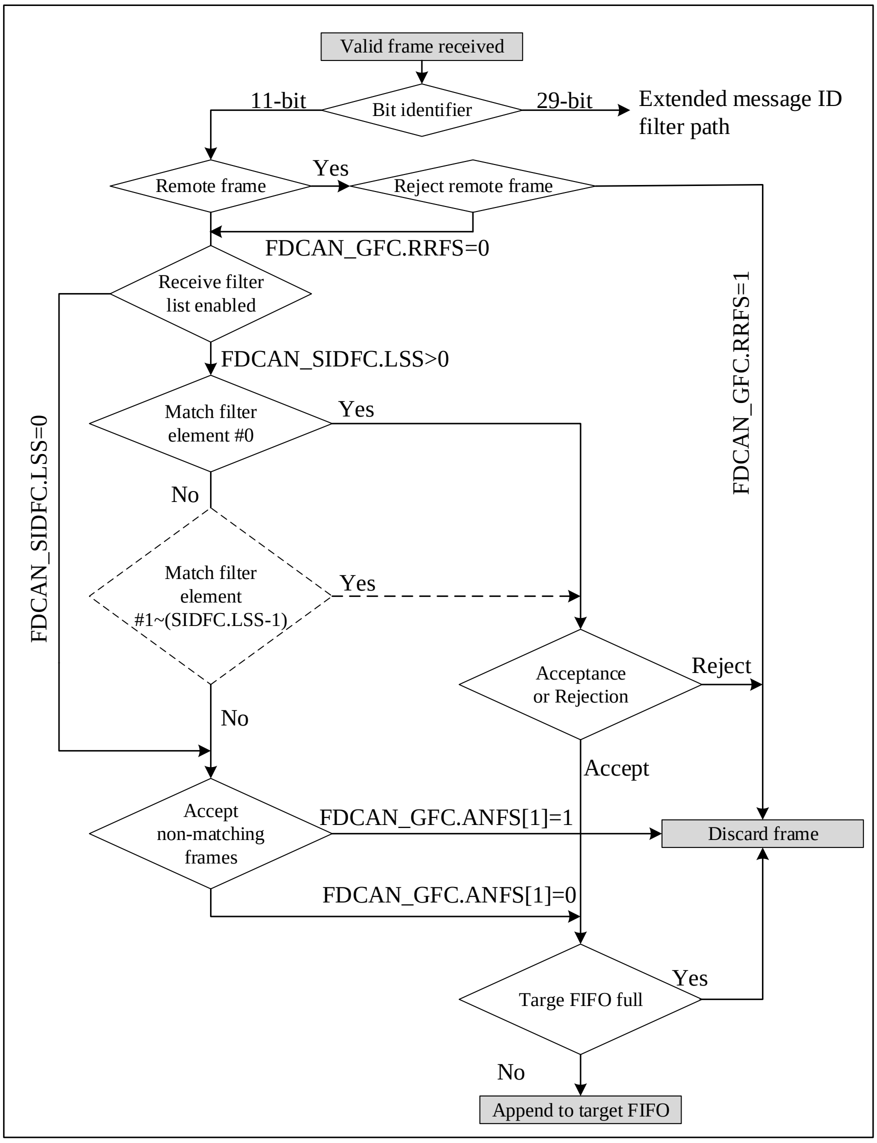
Bit positions in the mask that are 0 will mask the corresponding bit positions of the filter ID, disregarding the message ID's relevant bit positions during filtering, and only checking the receive message ID bit positions corresponding to the bit positions that are 1 in the mask.

If all mask bit positions are 1, a match will only occur when the receive message ID is identical to the filter ID. If all mask bit positions are 0, it matches all message IDs.

25.4.2.1.4 Standard Message ID Filtering

The process of filtering Standard Message IDs (11-bit IDs) is illustrated in the following figure.

Figure 25-7 FDCAN Standard Message ID Filtering Process

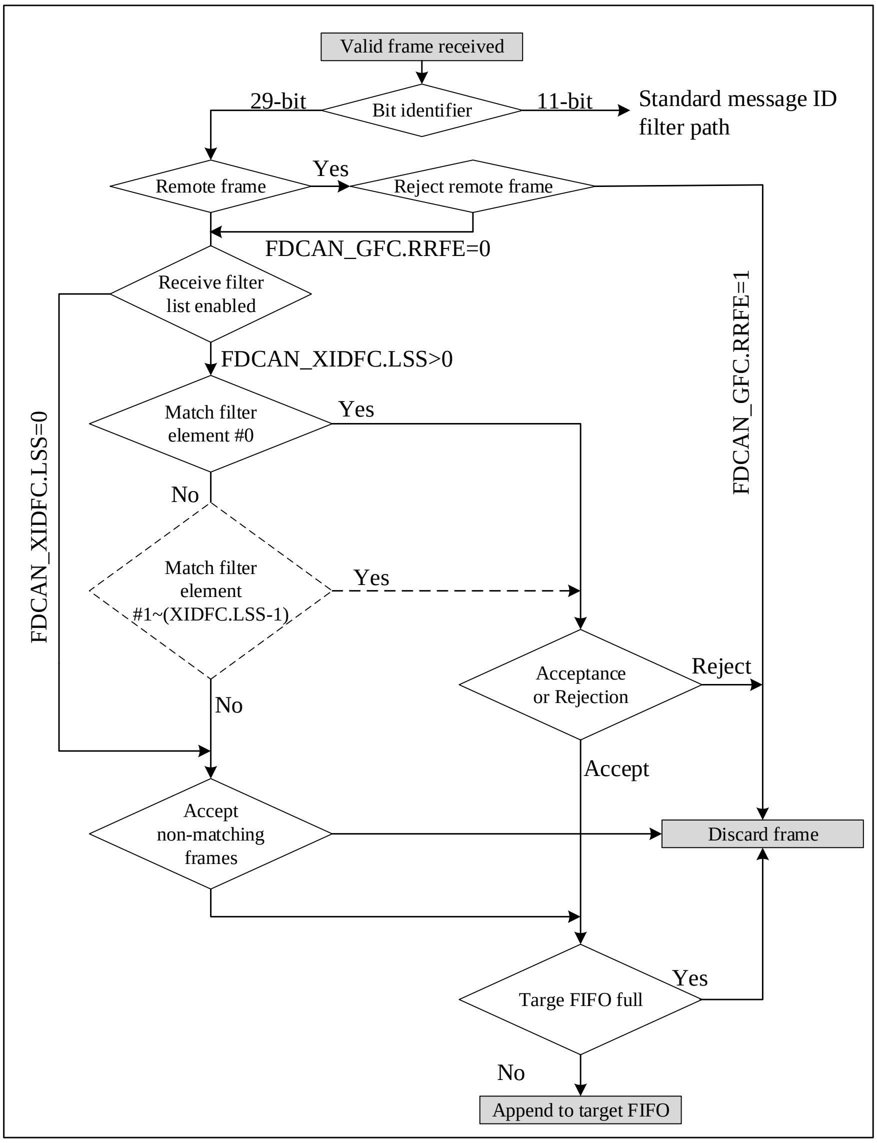


By using the Global Filter Configuration FDCAN_GFC and the Standard ID Filter Configuration FDCAN_SIDFC, it is possible to compare and check the message ID, Remote Transmission Request bit (RTR), and Identifier Extension bit (IDE) of received message frames.

25.4.2.1.5 Extended Message ID Filtering

The process of filtering Extended Message IDs (29-bit IDs) is illustrated in the following diagram.

Figure 25-8 FDCAN Extended Message ID Filtering Process



By using the Global Filter Configuration FDCAN_GFC and the Extended ID Filter Configuration FDCAN_XIDFC, it is possible to compare and check the message ID, Remote Transmission Request bit (RTR), and Identifier Extension bit (IDE) of received message frames. However, before the filter check, the message's extended ID will be ANDed with the mask FDCAN_XIDAM.)

25.4.2.2 Receive FIFO

Receive FIFO 0 and Receive FIFO 1 can be configured individually using the registers FDCAN_RXF0C and FDCAN_RXF1C, with each FIFO capable of holding up to 64 elements.

Messages that have passed through receive filtering will be transmitted to the corresponding receive FIFO based on the matched filter element configuration. In the default mode, when the receive FIFO is full (FDCAN_RXFnS.FnF=1b), new information cannot be written. This continues until at least one message is read out and the index is incremented.

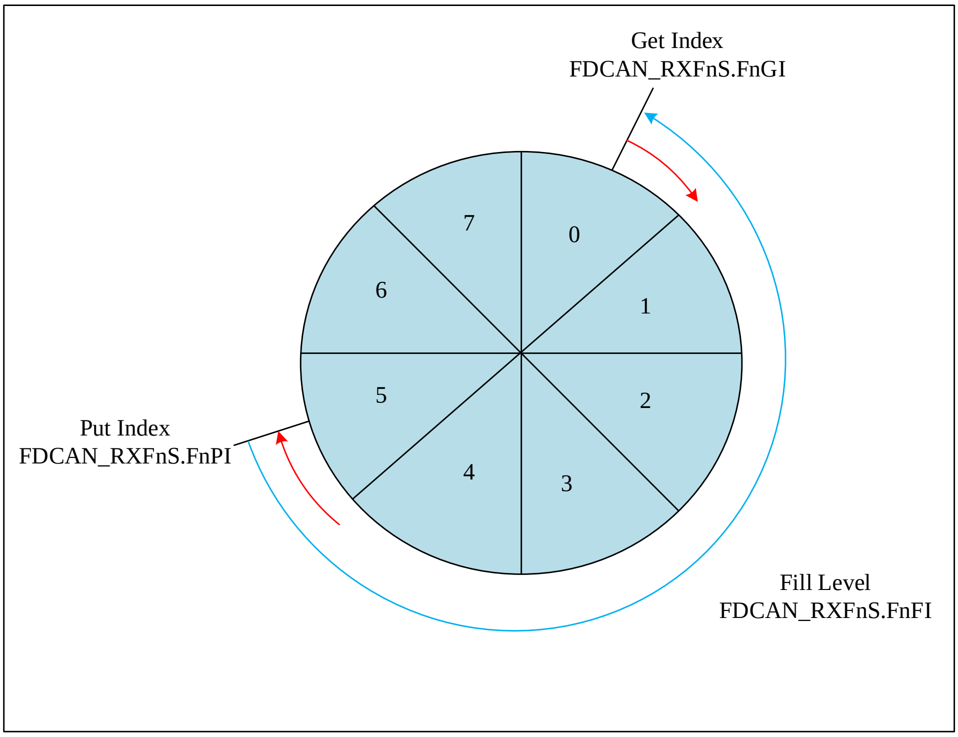
When the receive FIFO is full, any new incoming information is discarded, and the FDCAN_IR.RFnL flag is set to

1.

The receive FIFO watermark flag can be used to prevent receive FIFO overflow. When the fill level of the receive FIFO reaches the receive FIFO watermark configured by FDCAN_RXFnC.FnWM, the FDCAN_IR.RFnW flag is set to 1.

The receive FIFO status is shown in the following diagram.

Figure 25-9 FDCAN Rx FIFO Status Diagram



When reading information from the receive FIFO, the actual address of the FIFO element is calculated by multiplying the get index (FDCAN_RXFnS.FnGI) by the element size and adding the current FIFO start address (FDCAN_RXFnC.FnSA).

The relationship between FIFO element size and register configuration is shown in the following table.

Table 25-7 CAN FD Data Field Length Definition

RXESC.RBDS[2:0] RXESC.FnDS[2:0]	Data Field (Bytes)	Element Size (RAM Words)
000	8	4
001	12	5
010	16	6
011	20	7
100	24	8
101	32	10
110	48	14
111	64	18

25.4.2.2.1 Receive FIFO Blocking Mode

Enabling the receive FIFO Blocking Mode is done by configuring FDCAN_RXFnC.FnOM=0b, which is also the default operating mode of the receive FIFO.

In this mode, when the receive FIFO is full (FDCAN_RXFnS.FnPI = FDCAN_RXFnS.FnGI), new messages cannot be written to the current FIFO until at least one message is read out and the get index is incremented. When the receive FIFO is full, FDCAN_RXFnS.FnF is set to 1. Additionally, the interrupt flag FDCAN_IR.RFnF is also set to 1.

If a message is received while the corresponding receive FIFO is full, the message will be discarded, and the situation of message loss will be indicated by FDCAN_RXFnS.RFnL = '1'. Additionally, the interrupt flag FDCAN_IR.RFnL will also be set to 1.

25.4.2.2.2 Receive FIFO Overwrite Mode

By configuring FDCAN_RXFnC.FnOM=1b, the receive FIFO overwrite mode is enabled.

When the receive FIFO is full, the next new message to be stored in the FIFO will overwrite the oldest message in the FIFO. Both the write index and get index will be incremented by 1.

In overwrite mode, when the FIFO is full, reading receive FIFO elements should start at least from the read get index + 1. This is because when the CPU reads messages from the message RAM via the get index, there may be new messages written to the message RAM via the write index. At this point, inconsistent data may be read from the corresponding receive FIFO elements. Adding an offset to the get index when reading data from the receive FIFO can avoid this issue. The offset depends on the speed at which the CPU accesses the receive FIFO. As shown in the diagram below, when reading the receive FIFO, an offset of 2 is added relative to the get index. At this point, the two messages stored in elements 1 and 2 will be lost.

Figure 25-10 FDCAN Rx FIFO Full

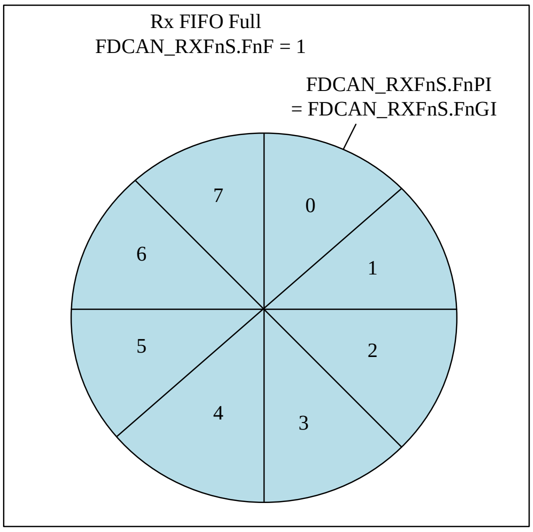
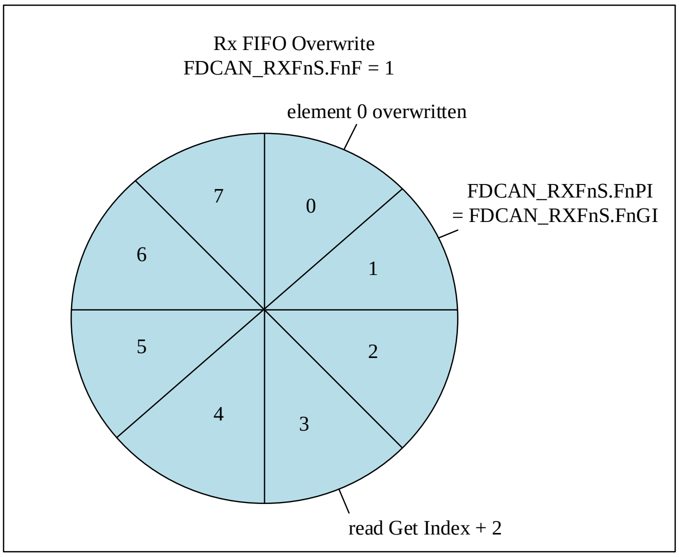


Figure 25-11 FDCAN Rx FIFO Overwrite Example



After reading data from the receive FIFO, the index of the last read element must be written to the receive FIFO acknowledgment index FDCAN_RXFnA.FnA. This will cause the get index to increment to that element index. If the write index has not yet advanced to this receive FIFO element, the FIFO full flag (FDCAN_RXFnS.FnF = '0') will be reset.

25.4.2.3 Dedicated Receive Buffer

FDCAN supports up to 64 dedicated receive buffers. The start address of the dedicated receive buffer area is configured by FDCAN_RXBC.RBSA.

For each dedicated receive buffer, a standard or extended message ID filter must be configured with SFEC/EFEC = 111b and SFID2/EFID2 [10:9] = 00b.

After a message is received through the filter, it will be stored in the dedicated receive buffer indicated by the filter element. The data format is the same as that of the receive FIFO elements. Additionally, the interrupt flag FDCAN_IR.DRX flag (message stored in dedicated receive buffer) is set to 1.

Table 25-8 Dedicated Receive Buffer Filter Configuration Example

Filter Element	SFID1[10:0] EFID1[28:0]	SFID2[10:9] EFID2[10:9]	SFID2[5:0] EFID2[5:0]
0	ID message 1	00	00 0000
1	ID message 2	00	00 0001
2	ID message 3	00	00 0010

After the last byte of the matched received message is written to the message RAM, the corresponding new message flag in registers FDCAN_NDAT1 and FDCAN_NDAT2 is set to 1. Once the new message flag is set to 1, the corresponding dedicated receive buffer will be locked to prevent it from being updated by subsequent received messages. The host must write a "1" to the corresponding flag bit to reset the new message flag.

When the new message flag of a dedicated receive buffer is set to 1, the filter element referencing the message ID of this dedicated receive buffer will be ignored, and the receive filtering will continue to check the filter list. This may result in subsequent messages with the same ID as the dedicated buffer being stored in another dedicated receive buffer, stored in the receive FIFO, or rejected, depending on the filter configuration.

The processing flow of the dedicated receive buffer is as follows:

- Reset the interrupt flag FDCAN_IR.DRX
- Read the new message flag register
- Read the message from the message RAM
- Reset the new data flag of the processed message

25.4.2.4 Debug Message Filtering

Debug messages are stored in dedicated receive buffers. To use them, three consecutive dedicated receive buffers need to be allocated (for example, #61, #62, #63) to store debug messages A, B, and C. Their format is the same as that of dedicated receive buffer or receive FIFO elements.

To filter debug messages, a standard/extended message ID filter element needs to be configured for each debug message, with the standard/extended filter element set to SFEC/EFEC=111b. Messages matching these filter elements will be stored in the dedicated receive buffer specified by SFID2 / EFID2[5:0].

When a debug message is stored in a dedicated receive buffer, neither the new message flag nor the interrupt will be set. The status can be queried through FDCAN_RXF1S.DMS.

Table 25-9 Debug Message Filter Configuration Example

Filter Element	SFID1[10:0] EFID1[28:0]	SFID2[10:9] EFID2[10:9]	SFID2[5:0] EFID2[5:0]
0	ID message 1	01	11 1101
1	ID message 2	10	11 1110
2	ID message 3	11	11 1111

25.4.3 Transmission Processing

The transmission processing unit is responsible for handling the transmission requests of dedicated transmit buffers, transmit FIFOs, and transmit queues. It controls the transfer of messages to the CAN core, write index and get index management, and the transmission event FIFO. Up to 32 transmit buffers can be configured for message transmission, and each transmit buffer element can be individually configured for communication mode (Classical CAN or CAN FD). Depending on the different configurations of FDCAN_RXESC, the data field size of transmit buffer elements ranges from 2 to 16 bytes. The possible configurations for sending message frames are shown in the table below.

Table 25-10 Transmission Message Frame Configuration

FDCAN_CCCR	Transmit Buffer Element	Frame Transmit

BRSE	FDOE	FDF	BRS	
ignore	0	ignore	ignore	Classical CAN
0	1	0	ignore	Classical CAN
0	1	1	ignore	FD CAN without baud rate switching
1	1	0	ignore	Classical CAN
1	1	1	0	FD CAN without baud rate switching
1	1	1	1	FD CAN with baud rate switching

Note: AUTOSAR requires at least three transmit queues and supports transmit cancellation.

When the transmit buffer request pending register FDCAN_TXBRP is updated, the transmission processing unit will initiate a scan of the message RAM to find the highest priority transmission request (the transmit buffer with the smallest message ID). The FDCAN_TXBRP register is updated after the transmission is completed, or when the host writes a transmit request, or when the host writes a cancel transmit request through FDCAN_TXBCR.

25.4.3.1 Transmission Pause

The transmission pause function is suitable for CAN applications where the message ID is permanently set to a specific value and is difficult to change easily. These message IDs have a higher arbitration priority than other messages, but in specific applications, their arbitration priority should be the opposite. This can lead to a situation where one node sends a series of CAN messages, causing delays in the CAN messages of another node due to their lower arbitration priority.

For example, if CAN node 1 enables the transmission pause function and needs to send four messages based on application requirements, after the successful transmission of the first message, it will wait for a bus idle time of two CAN bit periods before allowing the transmission of the next message. If there are other pending messages from other nodes, they will be sent during the idle time without needing to arbitrate with the next message from node 1. Upon receiving messages from other nodes, node 1 can start the next transmission immediately after the CAN bus is released from receiving messages.

The transmission pause function is controlled by FDCAN_CCCR.TXP. When this bit is set to 1, after successfully sending a message, the FDCAN will pause for two CAN bit times before starting the next transmission. This allows other CAN nodes in the network to send messages with lower priority IDs. The transmission pause is disabled by default (FDCAN_CCCR.TXP = 0b).

This feature can make burst transmissions from a single node more relaxed, preventing situations where the application erroneously requests excessive transmissions.

25.4.3.2 Dedicated Transmit Buffer

Dedicated transmit buffers are used for message transmission fully controlled by the host CPU. Each dedicated transmit buffer is configured with a specific message ID. If multiple transmit buffers are configured with the same message ID, the one with the lowest number will transmit first.

If the data section has been updated, a transmission request can be issued using FDCAN_TXBAR.ARn. The requested message will internally arbitrate with the messages in the configurable transmit FIFO/transmit queue and externally arbitrate with the messages on the CAN bus, and then be transmitted based on its message ID.

The size of dedicated transmit buffer elements is in words (32 bits). The starting address of the dedicated transmit buffer in the message RAM can be calculated by multiplying the transmit buffer index FDCAN_TXFQS.TFQPI (0…31) by the element size and then adding it to the transmit buffer start address FDCAN_TXBC.TBSA.

Table 25-11 Dedicated Transmit Buffer, Transmit FIFO/Queue Element Size

TXESC.TBDS[2:0]	Element Size (RAM Words)	Element Size (RAM Words)
000	8	4
001	12	5
010	16	6
011	20	7
100	24	8
101	32	10
110	48	14
111	64	18

25.4.3.3 Transmit FIFO

The transmission FIFO function can be enabled by clearing FDCAN_TXBC.TFQM. Messages stored in the transmit FIFO are sent starting from the message referenced by the get index FDCAN_TXFQS.TFGI. After each transmission, the get index is incremented cyclically until the transmit FIFO is empty. The transmit FIFO allows messages with the same ID to be sent from different transmit buffers in the order they were written into the transmit FIFO. FDCAN calculates the transmit FIFO fill level FDCAN_TXFQS.TFFL by the difference between the get index and the write index, indicating the number of available (idle) transmit FIFO elements.

New transmit messages must be written to the transmit buffer referenced by the write index FDCAN_TXFQS.TFQPI. Adding a request will increment the write index to the next available transmit FIFO element. When the write index reaches the get index, the transmit FIFO full flag (FDCAN_TXFQS.TFQF) is set to 1. At this point, do not continue writing messages to the transmit FIFO until the next message has been sent and the get index has been incremented.

When a message is added to the transmit FIFO, a request for transmission is made by writing a 1 to the FDCAN_TXBAR bit corresponding to the transmit buffer referenced by the transmit FIFO write index.

When multiple (n) messages are added to the transmit FIFO, they will be written into n consecutive transmit buffers starting from the write index. The transmit request can be made through FDCAN_TXBAR. Subsequently, the write index will increment cyclically n times. The number of requested transmit buffers should not exceed the number of available idle transmit buffers indicated by the transmit FIFO fill level.

When a transmission request from the transmit buffer referenced by the get index is canceled, the get index will increment to the next transmit buffer with a pending transmission request, and the transmit FIFO fill level will be recalculated.

If other transmit buffers' transmissions are canceled, the get index and FIFO fill level remain unchanged. The transmit

FIFO element size is in words. The starting address of the next available (idle) transmit FIFO can be calculated by multiplying the transmit FIFO write index FDCAN_TXFQS.TFQPI (0…31) by the element size and then adding it to the transmit buffer start address FDCAN_TXBC.TBSA.

25.4.3.4 Transmission Queue

To enable the transmission queue function, set FDCAN_TXBC.TFQM to 1. Messages stored in the transmission queue will be sent starting from the message with the smallest message ID (highest priority).

If multiple queue buffers are configured with the same message ID, the buffer with the lowest number will transmit first. New messages must be written to the transmit buffer referenced by the write index FDCAN_TXFQS.TFQPI. Adding a transmission request will cyclically increment the write index to the next available transmit buffer. If the transmission queue is full (FDCAN_TXFQS.TFQF=1b), the write index is invalid, and new messages should not be written to the transmission queue until at least one message has been sent or a transmission request has been canceled.

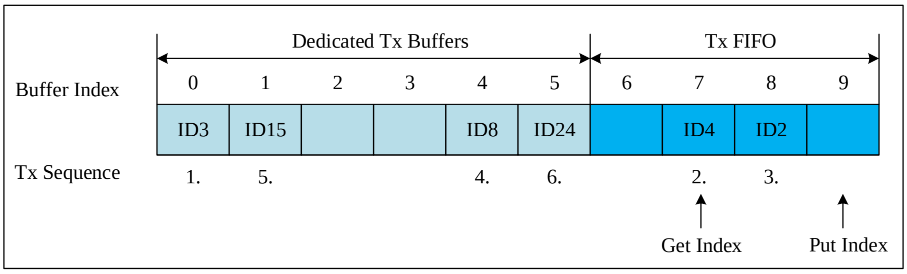
In the application, the register FDCAN_TXBRP can be used instead of the write index, allowing new messages to be written to any transmit buffer that does not have a pending request.

The transmission queue element size is in words. The starting address of the next available (idle) transmission queue can be calculated by multiplying the transmit queue write index FDCAN_TXFQS.TFQPI (0…31) by the element size and then adding it to the transmit buffer start address FDCAN_TXBC.TBSA.

25.4.3.5 Hybrid Use of Dedicated Transmit Buffers and Transmit FIFO

At this point, the transmit buffers in the message RAM are divided into a group of dedicated transmit buffers and a transmit FIFO. The number of dedicated transmit buffers is configured by FDCAN_TXBC.NDTB. The number of transmit buffers allocated to the transmit FIFO is configured by FDCAN_TXBC.TFQS. If TXBC.TFQS is 0, only dedicated transmit buffers are used.

Figure 25-12 FDCAN Tx Buffer and Tx FIFO Hybrid Configuration Example Diagram



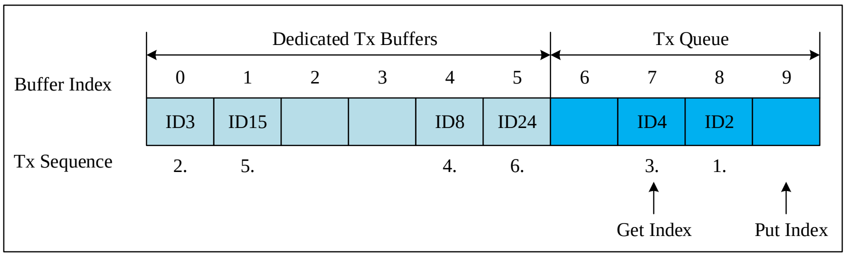
The priority for transmission is as follows:

- Scan the dedicated transmit buffers and the earliest pending transmit FIFO buffer (the transmit buffer referenced by FDCAN_TXFS.TFGI)
- The transmit buffer with the smallest ID has the highest priority and will be sent as the next message to transmit.

25.4.3.6 Hybrid Use of Dedicated Transmit Buffers and Transmit Queue

At this point, the transmit buffers in the message RAM are divided into a group of dedicated transmit buffers and a transmit queue. The number of dedicated transmit buffers is configured by FDCAN_TXBC.NDTB. The number of transmit buffers allocated to the transmit queue is configured by FDCAN_TXBC.TFQS. If TXBC.TFQS is set to zero, only dedicated transmit buffers are used.

Figure 25-13 FDCAN Tx Buffer and Tx Queue Hybrid Configuration Example Diagram



The priority for transmission is as follows:

- Scan all pending transmit buffers
- The transmit buffer with the smallest message ID has the highest priority and will be sent in the next transmission.

25.4.3.7 Transmission Cancellation

FDCAN supports transmission cancellation function. The host must write a 1 to the corresponding bit in the FDCAN_TXBCR register to cancel the transmission request from dedicated transmit buffers or transmit queue buffers. The transmission cancellation function does not apply to the transmit FIFO.

Upon successful cancellation, the corresponding positions in the FDCAN_TXBCF register are set to 1.

When canceling a message that is currently being sent, the corresponding FDCAN_TXBRP bit remains set to 1 until the transmission is completed or fails. If the transmission is successful, both the corresponding FDCAN_TXBTO and FDCAN_TXBCF bits will be set to 1. If the transmission fails, the message will not be retransmitted, and only the corresponding position in FDCAN_TXBCF will be set to 1.

Note: If a message is canceled immediately before starting transmission, a short idle window will occur, during which even if another message is waiting to be sent at that node, transmission will not start. This allows another node to send a message with lower priority than the second message in that node.

25.4.3.8 Transmission Event FIFO

FDCAN provides a Transmission Event FIFO to support transmission event handling. After sending a message on the CAN bus, the message ID and timestamp are stored in a Transmission Event FIFO element. To associate the transmission event with the Transmission Event FIFO element, the message tag from the sent message buffer is copied to the Transmission Event FIFO element. The Transmission Event FIFO can be configured with up to 32 elements.

The purpose of the Transmission Event FIFO is to separate the transmission status from the sent messages, where the transmit buffer only stores the messages to be sent, while the transmission status is stored separately in the Transmission Event FIFO. The advantage of this approach is that when handling dynamically managed transmission queues, the transmit buffer can be immediately used for new messages after a successful transmission. There is no need to retain the transmission status information of the transmit buffer before it is overwritten by a new message.

When the transmit event FIFO is full (FDCAN_IR.TEFF=1b), it is not possible to continue writing elements to the transmit event FIFO until at least one element is read out and the retrieval index of the transmit event FIFO has been incremented. If a transmit event occurs while the transmit event FIFO is full, this event will be discarded, and the interrupt flag FDCAN_IR.TEFL will be set to 1.

To prevent transmit event FIFO overflow, the transmit event FIFO watermark flag can be used. When the fill level of the transmit event FIFO reaches the transmit event FIFO watermark configured by FDCAN_TXEFC.EFWM, the interrupt flag FDCAN_IR.TEFW is set to 1.

When reading data from the transmit event FIFO, the starting address is obtained by multiplying the transmit event FIFO retrieval index FDCAN_TXEFS.EFGI by 2 and then adding it to the transmit event FIFO starting address FDCAN_TXEFC.EFSA.

25.4.4 FIFO acknowledgment

The retrieval indexes of Receive FIFO 0, Receive FIFO 1, and the Transmit Event FIFO are controlled by writing the corresponding FIFO acknowledgment indexes. After writing the FIFO acknowledgment index, the FIFO retrieval index will be set to the FIFO acknowledgment index value plus 1, and the FIFO fill level will be updated. Two scenarios may occur:

- When only a single element is read from the FIFO (i.e., the element pointed to by the retrieval index), this retrieval index value is written to the FIFO acknowledgment index.
- When a series of elements is read from the FIFO, simply write the FIFO acknowledgment index once at the end of the read sequence (the acknowledgment index value is the index of the last element read) to update the FIFO retrieval index.

Due to the CPU's unrestricted access to FDCAN's message RAM, special attention is required when reading FIFO elements in any order (without considering the retrieval index). This operation may be particularly useful when reading high-priority messages from one of the two Receive FIFOs. In this scenario, do not write the FIFO acknowledgment index, as doing so would set the retrieval index to an incorrect position and alter the FIFO fill level, resulting in the loss of some earlier FIFO elements.

Note: The application must ensure that the written FIFO acknowledgment index value is valid. FDCAN will not check whether the written value is correct.

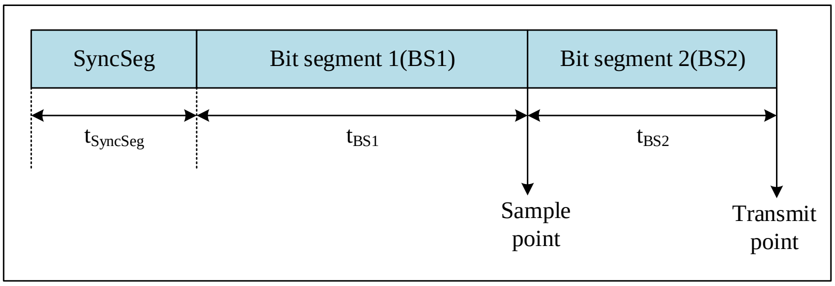
25.4.5 Bit Timing

The bit timing logic monitors the serial bus, performs data sampling, and adjusts the sampling point by synchronizing at the start bit edge and resynchronizing at subsequent bit edges. Its operation can be simplified by dividing the bit time into three time segments, as follows:

- Synchronization Segment (SYNC_SEG): The time segment expected to have bit transitions, with a fixed length of one time quantum (1×tq).
- Bit Segment 1 (BS1): Defines the position of the sampling point, including the PROP_SEG and PHASE_SEG1 as defined in the CAN standard, with a configurable duration between 2 to 256(nominal) or 1 to 32(data) time quantum, automatically extended to compensate for positive phase drift caused by frequency differences between nodes.

- Bit Segment 2 (BS2): Defines the position of the transmission point, namely the PHASE_SEG2 as defined in the CAN standard, with a configurable duration between 2 to 128(nominal) or 1 to 16(data) time quantum, automatically shortened to compensate for negative phase drift.

Figure 25-14 FDCAN Bit Timing



The baud rate is the reciprocal of the bit time (baud rate = 1/bit time), and the bit time is the sum of three time segments. As shown in the figure above, the bit time = $t_{SyncSeg} + t_{BS1} + t_{BS2}$, where:

- Nominal Bit Time
  - tq = (FDCAN_NBTP.NBRP[8:0] + 1) * t_{FDCAN_CLK}
  - $t_{SyncSeg}$ = 1*tq
  - t_{BS1} = tq * (FDCAN_NBTP.NTSEG1[7:0] + 1)
  - t_{BS2} = tq * (FDCAN_NBTP.NTSEG2[6:0] + 1)
- Data Bit Time
  - tq = (FDCAN_DBTP. DBRP [4:0] + 1) * t_{FDCAN_CLK}
  - $t_{SyncSeg}$ = 1*tq
  - t_{BS1} = tq * (FDCAN_DBTP. DTSEG1 [4:0] + 1)
  - t_{BS2} = tq * (FDCAN_DBTP. DTSEG2 [3:0] + 1)

The (re)Synchronization Jump Width (SJW) defines the upper limit for extending or shortening time segments and can be configured between 1 to 128(nominal) or 1 to 16(data) time quantum.

An effective edge is defined as the bus level change from recessive level to dominant level, provided that the node itself does not transmit a dominant bit.

If an effective edge is detected in BS1 instead of in SYNC_SEG, BS1 can extend up to SJW at most, causing a delay in the sampling point. Conversely, if an effective edge is detected in BS2, BS2 can shorten SJW at most, advancing the transmission point.

To prevent programming errors, the bit timing registers can only be configured when the device is in standby mode. The FDCAN_DBTP and FDCAN_NBTP registers (dedicated to data and nominal bit timing, respectively) can only be accessed when both FDCAN_CCCR.CCE and FDCAN_CCCR.INIT are set to 1.

Note: For detailed information on CAN bit timing and resynchronization mechanisms, please refer to the ISO 11898-1 standard.

25.5 FDCAN Registers

25.5.1 FDCAN Register Overview

FDCAN1 Base Address: 0x40005000

FDCAN2 Base Address: 0x40005400

FDCAN3 Base Address: 0x40005800

Table 25-12 FDCAN Register Overview

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
000h	FDCAN_CREL	REL[3:0]				STEP[3:0]				SUBSTEP[3:0]				YEAR[3:0]				MON[7:0]								DAY[7:0]							
	Reset value	0	0	1	1	0	0	1	1	0	0	0	1	0	0	1	1	0	0	0	0	0	0	1	1	0	0	1	0	1	0	0	0
004h	FDCAN_ENDN	ETV[31:0]																															
	Reset value	1	0	0	0	0	1	1	1	0	1	1	0	0	1	0	1	0	1	0	0	0	0	1	1	0	0	1	0	0	0	0	1
00Ch	FDCAN_DBTP	Reserved								TDC	Reserved		DBRP[4:0]					Reserved			DTSEG1[4:0]					DTSEG2[3:0]				DSJW[3:0]			
	Reset value									0			0	0	0	0	0				0	1	0	1	0	0	0	1	1	0	0	1	1
010h	FDCAN_TEST	Reserved																								RX	TX[1:0]		LBCK	Reserved			
	Reset value																									0	0	0	0				
014h	FDCAN_RWD	Reserved																WDV[7:0]								WDC[7:0]							
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
018h	FDCAN_CCCR	Reserved																NISO	TXP	EFBI	PXHD	Reserved		BRSE	FDOE	TESET	DAR	MON	CSR	CSA	ASM	CCE	INIT
	Reset value																	0	0	0	0			0	0	0	0	0	0	0	0	0	1
01Ch	FDCAN_NBTP	NSJW[6:0]							NBRP[8:0]									NTSEG1[7:0]								Reserved	NTSEG2[7:0]						
	Reset value	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0		0	0	0	0	0	1	1
020h	FDCAN_TSCC	Reserved												TCP[3:0]				Reserved														TSS[1:0]	
	Reset value													0	0	0	0															0	0
024h	FDCAN_TSCV	Reserved																TSC[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
028h	FDCAN_TOCC	TOP[15:0]																Reserved													TOS[1:0]		ETOC
	Reset value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1														0	0	0
02Ch	FDCAN_TOCV	Reserved																TOC[15:0]															
	Reset value																	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
040h	FDCAN_ECR	Reserved								CEL[7:0]								RP	REC[6:0]							TEC[7:0]							
	Reset value									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
044h	FDCAN_PSR	Reserved									TDCV[6:0]							Reserved	PXE	RFDF	RBRS	RESI	DLEC[2:0]			BO	EW	EP	ACT[1:0]		LEC[2:0]		
	Reset value										0	0	0	0	0	0	0		0	0	0	0	1	1	1	0	0	0	0	0	1	1	1
048h	FDCAN_TDCR	Reserved																	TDCO[6:0]							Reserved	TDCF[6:0]						
	Reset value																		0	0	0	0	0	0	0		0	0	0	0	0	0	0
050h	FDCAN_IR	Reserved		ARA	PED	PEA	WDI	BO	EW	EP	ELO	Reserved		DRX	TOO	MRAF	TSW	TEFL	TEFF	TEFW	TEFN	TFE	TCF	TC	HPM	RF1L	RE1F	RF1W	RF1N	RF0L	RF0F	RF0W	RF0N
	Reset value			0	0	0	0	0	0	0	0			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
054h	FDCAN_IE	Reserved		ARAE	PEDE	PEAE	WDIE	BOE	EWE	EPE	ELOE	Reserved		DRXE	TOOE	MRAFE	TSWE	TEFLE	TEFFE	TEFWE	TEFNE	TFEE	TCFE	TCE	HPME	RF1LE	RE1FE	RF1WE	RF1NE	RF0LE	RF0FE	RF0WE	RF0NE
	Reset value			0	0	0	0	0	0	0	0			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
058h	FDCAN_ILS	Reserved		ARAL	PEDL	PEAL	WDIL	BOL	EWL	EPL	ELOL	Reserved		DRXL	TOOL	MRAFL	TSWL	TEFLL	TEFFL	TEFWL	TEFNL	TFEL	TCFL	TCL	HPML	RF1LL	RE1FL	RF1WL	RF1NL	RF0LL	RF0FL	RF0WL	RF0NL
	Reset value			0	0	0	0	0	0	0	0			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
05Ch	FDCAN_ILE	Reserved																														EINT1	EINT0
	Reset value																															0	0

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
080h	FDCAN_GFC	Reserved																										ANFS[1:0]		ANFE[1:0]		RRFS	RRFE
	Reset value																											0	0	0	0	0	0
084h	FDCAN_SIDFC	Reserved								LSS[7:0]								FLSSA[13:0]														Reserved	
	Reset value									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
088h	FDCAN_XIDFC	Reserved								LSE[7:0]								FLESA[13:0]														Reserved	
	Reset value									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
090h	FDCAN_XIDM	Reserved			EIDM[28:0]																												
	Reset value				1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
094h	FDCAN_HPMS	Reserved																FLST	FIDX[6:0]							MSI[1:0]		BIDX[5:0]					
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
098h	FDCAN_NDAT1	ND31	ND30	ND29	ND28	ND27	ND26	ND25	ND24	ND23	ND22	ND21	ND20	ND19	ND18	ND17	ND16	ND15	ND14	ND13	ND12	ND11	ND10	ND9	ND8	ND7	ND6	ND5	ND4	ND3	ND2	ND1	ND0
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
09Ch	FDCAN_NDAT2	ND63	ND62	ND61	ND60	ND59	ND58	ND57	ND56	ND55	ND54	ND53	ND52	ND51	ND50	ND49	ND48	ND47	ND46	ND45	ND44	ND43	ND42	ND41	ND40	ND39	ND38	ND37	ND36	ND35	ND34	ND33	ND32
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0A0h	FDCAN_RXF0C	F0OM	F0WM[6:0]							Reserved	F0S[6:0]							F0SA[13:0]														Reserved	
	Reset value	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0A4h	FDCAN_RXF0S	Reserved						RF0L	F0F	Reserved		F0PI[5:0]						Reserved		F0GI[5:0]						Reserved	F0FL[6:0]						
	Reset value							0	0			0	0	0	0	0	0			0	0	0	0	0	0		0	0	0	0	0	0	0
0A8h	FDCAN_RXF0A	Reserved																										F0AI[5:0]					
	Reset value																											0	0	0	0	0	0
0ACh	FDCAN_RXBC	Reserved																RBSA[13:0]														Reserved	
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0B0h	FDCAN_RXF1C	F1OM	F1WM[6:0]							Reserved	F1S[6:0]							F1SA[13:0]														Reserved	
	Reset value	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0B4h	FDCAN_RXF1S	DMS[1:0]		Reserved				RF1L	F1F	Reserved		F1PI[5:0]						Reserved		F1GI[5:0]						Reserved	F1FL[6:0]						
	Reset value	0	0					0	0			0	0	0	0	0	0			0	0	0	0	0	0		0	0	0	0	0	0	0
0B8h	FDCAN_RXF1A	Reserved																										F1AI[5:0]					
	Reset value																											0	0	0	0	0	0
0BCh	FDCAN_RXESC	Reserved																					RBDS[2:0]			Reserved	F1DS[2:0]			Reserved	F0DS[2:0]		
	Reset value																						0	0	0		0	0	0		0	0	0
0C0h	FDCAN_TXBC	Reserved	TFQM	TFQS[5:0]						Reserved		NDTB[5:0]						TBSA[13:0]														Reserved	
	Reset value		0	0	0	0	0	0	0			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0C4h	FDCAN_TXFQS	Reserved										TFQF	TFQPI[4:0]					Reserved			TFGI[4:0]					Reserved		TFFL[5:0]					
	Reset value											0	0	0	0	0	0				0	0	0	0	0			0	0	0	0	0	0
0C8h	FDCN_TXESC	Reserved																													TBDS[2:0]		
	Reset value																														0	0	0
0CCh	FDCAN_TXBRP	TRP31	TRP30	TRP29	TRP28	TRP27	TRP26	TRP25	TRP24	TRP23	TRP22	TRP21	TRP20	TRP19	TRP18	TRP17	TRP16	TRP15	TRP14	TRP13	TRP12	TRP11	TRP10	TRP9	TRP8	TRP7	TRP6	TRP5	TRP4	TRP3	TRP2	TRP1	TRP0
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0D0h	FDCAN_TXBAR	AR31	AR30	AR29	AR28	AR27	AR26	AR25	AR24	AR23	AR22	AR21	AR20	AR19	AR18	AR17	AR16	AR15	AR14	AR13	AR12	AR11	AR10	AR9	AR8	AR7	AR6	AR5	AR4	AR3	AR2	AR1	AR0
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0D4h	FDCAN_TXBCR	CR31	CR30	CR29	CR28	CR27	CR26	CR25	CR24	CR23	CR22	CR21	CR20	CR19	CR18	CR17	CR16	CR15	CR14	CR13	CR12	CR11	CR10	CR9	CR8	CR7	CR6	CR5	CR4	CR3	CR2	CR1	CR0
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0D8h	FDCAN_TXBTO	TO31	TO30	TO29	TO28	TO27	TO26	TO25	TO24	TO23	TO22	TO21	TO20	TO19	TO18	TO17	TO16	TO15	TO14	TO13	TO12	TO11	TO10	TO9	TO8	TO7	TO6	TO5	TO4	TO3	TO2	TO1	TO0
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0DCh	FDCAN_TXBCF	CF31	CF30	CF29	CF28	CF27	CF26	CF25	CF24	CF23	CF22	CF21	CF20	CF19	CF18	CF17	CF16	CF15	CF14	CF13	CF12	CF11	CF10	CF9	CF8	CF7	CF6	CF5	CF4	CF3	CF2	CF1	CF0
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0E0h	FDCAN_TXBTIE	TIE31	TIE30	TIE29	TIE28	TIE27	TIE26	TIE25	TIE24	TIE23	TIE22	TIE21	TIE20	TIE19	TIE18	TIE17	TIE16	TIE15	TIE14	TIE13	TIE12	TIE11	TIE10	TIE9	TIE8	TIE7	TIE6	TIE5	TIE4	TIE3	TIE2	TIE1	TIE0
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0E4h	FDCAN_TXBCIE	CFIE31	CFIE30	CFIE29	CFIE28	CFIE27	CFIE26	CFIE25	CFIE24	CFIE23	CFIE22	CFIE21	CFIE20	CFIE19	CFIE18	CFIE17	CFIE16	CFIE15	CFIE14	CFIE13	CFIE12	CFIE11	CFIE10	CFIE9	CFIE8	CFIE7	CFIE6	CFIE5	CFIE4	CFIE3	CFIE2	CFIE1	CFIE0
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0F0h	FDCAN_TXEFC	Reserved		EFWM[5:0]						Reserved		EFS[5:0]						EFSA[13:0]														Reserved	

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reset value			0	0	0	0	0	0			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0F4h	FDCAN_TXEFS	Reserved						TEFL	EFF	Reserved			EFPI[4:0]					Reserved			EFGI[4:0]					Reserved		EFFL[5:0]					
	Reset value							0	0				0	0	0	0	0				0	0	0	0	0			0	0	0	0	0	0
0F8h	FDCAN_TXEFA	Reserved																											EFAI[4:0]				
	Reset value																												0	0	0	0	0
144h	FDCAN_TTSS	Reserved																							TS_EN	TS_SEL[2:0]			Reserved				
	Reset value																								0	0	0	0					

25.5.2 FDCAN Core Release Register (FDCAN_CREL)

Address offset: 0x00

Reset value: 0x3313 0328

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
REL[3:0]				STEP[3:0]				SUBSTEP[3:0]				YEAR[3:0]			
r				r				r				r			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MON[7:0]								DAY[7:0]							
r								r							

Bit Field	Name	Description
31:28	REL[3:0]	Core release, 1 bit BCD code.
27:24	STEP[3:0]	Step of Core release, 1 bit BCD code.
23:20	SUBSTEP[3:0]	Sub-step of Core release, 1 bit BCD code.
19:16	YEAR[3:0]	Timestamp Year, 1 bit BCD code.
15:8	MON[7:0]	Timestamp Month, 2 bit BCD code.
7:0	DAY[7:0]	Timestamp Day, 2 bit BCD code.

25.5.3 FDCAN Endianness Register (FDCAN_ENDN)

Address offset: 0x04

Reset value: 0x8765 4321

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ETV[31:16]															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETV[15:0]															
r															

Bit Field	Name	Description
31:0	ETV[31:0]	Endianness Test Valu, fixed is 0x8765 4321.

25.5.4 FDCAN Data Bit Timing and Prescaler Register (FDCAN_DBTP)

Address offset: 0x0C

Reset value: 0x0000 0A33

Data can only be written to this register when both FDCAN_CCCR.CCE and FDCAN_CCCR.INIT bits are set to 1. The CAN data time quantum tq is programmable, where tq = (DBRP+1) * t_{FDCAN_CLK} clock cycles, ranging from 1 to 32 FDCAN_CLK clock cycles.

DTSEG1 is the sum of Prop_Seg and Phase_Seg1. DTSEG2 is Phase_Seg2. Therefore, the length of the data bit time is (programmed value) (DTSEG1 + DTSEG2 + 3) * tq, ranging from 3 to 49 tq.

The information processing time (IPT) is zero, meaning that the data for the next bit is available at the first clock edge after the sampling point.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								TDC	Reserved		DBRP[4:0]				
								rw					rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			DTSEG1[4:0]					DTSEG2[3:0]				DSJW[3:0]			
					rw				rw					rw	

Bit Field	Name	Description
31:24	Reserved	Reserved, must be kept at reset value.
23	TDC	Transceiver Delay Compensation. 0: Disable Transceiver Delay Compensation 1: Enable Transceiver Delay Compensation
22:21	Reserved	Reserved, must be kept at reset value.
20:16	DBRP[4:0]	Data Bit Rate Prescaler. The value of the Data Bit Rate Prescaler, used to divide the FDCAN_CLK clock to generate the data time quantum (tq). The data bit time is a multiple of this time quantum (tq). The valid range for the Data Bit Rate Prescaler is 0 to 31. When TDC = '1', the range is restricted to 0 and 1. The hardware interprets this value as the programmed value plus 1.
15:13	Reserved	Reserved, must be kept at reset value.
12:8	DTSEG1[4:0]	Data time segment before sample point. The valid values range from 0 to 31. The hardware interprets this value as the programmed value plus 1.
7::4	DTSEG2[3:0]	Data time segment after sample point. The valid values range from 0 to 31. The hardware interprets this value as the programmed value plus 1.

Bit Field	Name	Description
3:0	DSJW[3:0]	Synchronization Jump Width. The valid values range from 0 to 31. The hardware interprets this value as the programmed value plus 1.

Note:

1. *The bit rate of the CAN FD data phase configured through DBTP must be higher than or equal to the bit rate of the arbitration phase configured through NBTP.*

25.5.5 FDCAN Test Register (FDCAN_TEST)

To enable write access to the test registers, the FDCAN_CCCR.TEST bit must be set to "1". When the FDCAN_CCCR.TEST bit is reset, all test register functions are set to their reset values.

The loopback mode and software control of the receive pin FDCANx_TX belong to the hardware test mode. Programming TX to a value other than "00" may interfere with message transmission on the CAN bus.

Address offset: 0x10

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								RX	TX[1:0]		LBCK	Reserved			
								r	rw		rw				

Bit Field	Name	Description
31:8	Reserved	Reserved, must be kept at reset value.
7	RX	Receive Pin Monitoring the actual value of the receive pin FDCAN_RX 0: CAN bus is dominant (FDCAN_RX = '0') 1: CAN bus is recessive (FDCAN_RX = '1')
6:5	TX[1:0]	Control of Transmit Pin 00: Reset value, FDCAN_TX controlled by the CAN core, updated at the end of the CAN bit time 01: Monitoring the sampling point on the transmit pin FDCAN_TX 10: Dominant ('0') level on pin FDCAN_TX 11: Recessive ('1') level on pin FDCAN_TX
4	LBCK	Loop Back Mode 0: Reset value, disable loopback mode 1: Enable loopback mode
3:0	Reserved	Reserved, must be kept at reset value.

25.5.6 FDCAN RAM Watchdog Register (FDCAN_RWD)

The RAM watchdog monitors the READY signal output from the message RAM. Accessing the message RAM initiates the message RAM watchdog counter, with the initial value configured by FDCAN_RWD.WDC. When the access to the message RAM is successfully completed and the READY signal is issued, the counter is reset to the FDCAN_RWD.WDC value. If the counter decrements to 0 before the message RAM issues the READY signal, the counter stops counting, and the interrupt flag FDCAN_IR.WDI is set. The RAM watchdog counter is driven by the interface clock (fdcan_pclk).

Address offset: 0x14

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WDV[7:0]								WDC[7:0]							
r								rw							

Bit Field	Name	Description
31:16	Reserved	Reserved, must be kept at reset value.
15:8	WDV[7:0]	Watchdog value Value of the message RAM watchdog counter.)
7:0	WDC[7:0]	Watchdog configuration Starting value of the message RAM watchdog counter. If the reset value "00" is used, the counter is disabled. These bits are write-protected and can only be accessed for writing when both FDCAN_CCCR.CCE and FDCAN_CCCR.INIT are set to 1.

25.5.7 FDCAN Control Register (FDCAN_CCCR)

Address offset: 0x18

Reset value: 0x0000 0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NISO	TXP	EFBI	PXHD	Reserved		BRSE	FDOE	TEST	DAR	MON	CSR	CSA	ASM	CCE	INIT
rw	rw	rw	rw			rw	rw	rw	rw	rw	rw	r	rw	rw	rw

Bit Field	Name	Description
31:16	Reserved	Reserved, must be kept at reset value.

Bit Field	Name	Description
15	NISO	Non ISO Operation If this bit is set to 1, FDCAN will use the CAN FD frame format specified in the Bosch CAN FD Specification V1.0. 0: Conforms to the CAN FD frame format specified in ISO 11898-1 1: Conforms to the CAN FD frame format specified in the Bosch CAN FD Specification V1.0
14	TXP	Transmit Pause If this bit is set to 1, after a successful frame transmission, FDCAN will pause for two CAN bit times before starting the next transmission. 0: Disable transmission pause 1: Enable transmission pause
13	EFBI	Edge Filtering during Bus Integration 0: Disable edge filtering 1: Require two consecutive dominant tq to detect a hard synchronization edge
12	PXHD	Protocol Exception Handling Disable 0: Enable protocol exception handling 1: Disable protocol exception handling *Note: When protocol exception handling is disabled, FDCAN will transmit error frames upon detecting protocol exception conditions.*
11:10	Reserved	Reserved, must be kept at reset value.
9	BRSE	Bit Rate Switch Enable 0: Disable bit rate switching during transmission 1: Enable bit rate switching during transmission *Note: BRSE is invalid when CAN FD operation is disabled (FDOE = '0').*
8	FDOE	FD Operation Enable 0: Disable FD operation 1: Enable FD operation
7	TEST	Test Mode Enable 0: Normal operation, register TEST retains reset value 1: Test mode, enables write access to register TEST
6	DAR	Disable Automatic Retransmission 0: Enable automatic retransmission of unsuccessfully transmitted messages 1: Disable automatic retransmission
5	MON	Bus Monitoring Mode The MON bit can only be set to 1 through software when both CCE and INIT are set to "1". This bit can be reset by the host at any time. 0: Disable bus monitoring mode 1: Enable bus monitoring mode
4	CSR	Clock Stop Request 0: Clock stop not requested 1: Clock stop requested. If clock stop is requested, INIT will be set to 1 first, followed by CSA after all pending

Bit Field	Name	Description
		transmission requests are completed and the CAN bus reaches the idle state.
3	CSA	Clock Stop Acknowledge 0: Clock stop not acknowledged 1: FDCAN can be powered down by stopping the FDCAN_CLK clock and fdcan_pclk clock
2	ASM	Restricted Operation Mode ASM can only be set by the host when both CCE and INIT are set to '1'. This bit can be reset by the host at any time. 0: Normal CAN operation 1: Activate restricted operation mode
1	CCE	Configuration Change Enable 0: CPU does not have write access permission to protected configuration registers 1: CPU has write access permission to protected configuration registers (when CCCR.INIT = '1')
0	INIT	Initialization 0: Normal operation 1: Start initialization

Note: Due to the synchronization mechanism between the two clock domains, there is a delay after writing the INIT value before it can be read. Therefore, it is necessary to read INIT to ensure that the previously written INIT value has taken effect before setting it to a new value.

25.5.8 FDCAN Nominal Bit Timing and Prescaler Register (FDCAN_NBTP)

Data can only be written to this register when both FDCAN_CCCR.CCE and FDCAN_CCCR.INIT bits are set to 1. The CAN nominal time quantum tq is programmable, where tq = (NBRP+1) * t_{FDCAN_CLK} clock cycles, ranging from 1 to 512 FDCAN_CLK clock cycles.

NTSEG1 is the sum of Prop_Seg and Phase_Seg1. NTSEG2 is Phase_Seg2. Therefore, the length of the nominal bit time is (programmed value) (NTSEG1 + NDTSEG2 + 3) * tq, ranging from 4 to 385 tq.

Address offset: 0x1C

Reset value: 0x0600 0A03

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NSJW[6:0]							NBRP[8:0]								
rw							rw								

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NTSEG1[7:0]								Reserved	NTSEG2[6:0]						
rw									rw						

Bit Field	Name	Description
31:25	NSJW[6:0]	Nominal (Re)Synchronization Jump Width

Bit Field	Name	Description
		The valid values range from 0 to 127. The hardware interprets this value as the programmed value plus 1. These bits are write-protected and can only be accessed for writing when both FDCAN_CCCR.CCE and FDCAN_CCCR.INIT are set to 1.
24:16	NBRP[8:0]	Nominal Bit Rate Prescaler The value of the nominal bit rate prescaler used to divide the FDCAN_CLK clock to generate the nominal time quantum (tq), and the nominal bit time is a multiple of this time quantum. The valid values for the bit rate prescaler range from 0 to 511. The hardware interprets this value as the programmed value plus 1.
15:8	NTSEG1[7:0]	Nominal time segment before sample point The valid values range from 1 to 255. The hardware interprets this value as the programmed value plus 1.
7	Reserved	Reserved, must be kept at reset value.
6:0	NTSEG2[6:0]	Nominal time segment after sample point The valid values range from 1 to 127. The hardware interprets this value as the programmed value plus 1.

25.5.9 FDCAN Timestamp Counter Configuration Register (FDCAN_TSCC)

Address offset: 0x20

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved												TCP[3:0]			
												rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved														TSS[1:0]	
														rw	

Bit Field	Name	Description
31:20	Reserved	Reserved, must be kept at reset value.
19:16	TCP[3:0]	Timestamp Counter Prescaler Configures the timestamp and timeout counter time units as multiples of CAN bit time [1…16]. The hardware interprets this value as the programmed value plus 1. *Note: In CAN FD mode, an external counter is required for timestamp generation (TSS = "10"). These bits are write-protected and can only be accessed for writing when both FDCAN_CCCR.CCE and FDCAN_CCCR.INIT are set to 1.*
15:2	Reserved	Reserved, must be kept at reset value.
1:0	TSS[1:0]	Timestamp Select 00: Timestamp counter value always 0x0000 01: Timestamp counter value increments with TCP

Bit Field	Name	Description
		10: Use external timestamp counter value 11: Same as "00"

25.5.10 FDCAN Timestamp Counter Value Register (FDCAN_TSCV)

Address offset: 0x24

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSC[15:0]															
rw															

Bit Field	Name	Description
31:16	Reserved	Reserved, must be kept at reset value.
15:0	TSC[15:0]	Timestamp Counter The internal/external timestamp counter value is captured at the start of (receive and transmit) frames. When FDCAN_TSCC.TSS = "01", the timestamp counter increments by multiples of CAN bit time [1…16], depending on the configuration of FDCAN_TSCC.TCP. Counter wraparound sets the interrupt flag FDCAN_IR.TSW to 1. Write access resets the counter value to 0. When FDCAN_TSCC.TSS = "10", TSC reflects the external timestamp counter value, and write access has no effect.

Note:

1. "Wraparound" refers to the timestamp counter value changing from a non-zero value to 0, not due to write access to TSCV.

2. Byte access: Writing to register bytes 3/2/1/0 will reset the timestamp counter.

25.5.11 FDCAN Timeout Counter Configuration Register (FDCAN_TOCC)

Address offset: 0x28

Reset value: 0xFFFF 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TOP[15:0]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													TOS[1:0]		ETOC

rw rw

Bit Field	Name	Description
31:16	TOP[15:0]	Timeout Period Starting value of the timeout counter (decrementing counter).
15:3	Reserved	Reserved, must be kept at reset value.
2:1	TOS[1:0]	TOS: Timeout Select When operating in continuous mode, writing to FDCAN_TOCV will preset the counter to the value specified by FDCAN_TOCC.TOP and continue decrementing the count. When the timeout counter is controlled by one of the FIFOs and the FIFO is empty, the counter will be preset to the FDCAN_TOCC.TOP value. The countdown starts after writing the first FIFO element. 00: Continuous operation 01: Timeout controlled by Transmit Event FIFO 10: Timeout controlled by Receive FIFO 0 11: Timeout controlled by Receive FIFO 1 These bits are write-protected and can only be accessed for writing when both FDCAN_CCCR.CCE and FDCAN_CCCR.INIT are set to 1.
0	ETOC	Enable Timeout Counter. 0: Disable timeout counter 1: Enable timeout counter These bits are write-protected and can only be accessed for writing when both FDCAN_CCCR.CCE and FDCAN_CCCR.INIT are set to 1.

25.5.12 FDCAN Timeout Counter Value Register (FDCAN_TOCV)

Address offset: 0x2C

Reset value: 0x0000 FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TOC[15:0]															

rw

Bit Field	Name	Description
31:16	Reserved	Reserved, must be kept at reset value.
15:0	TOC[15:0]	Timeout Counter The timeout counter decrements by multiples of CAN bit time [1…16], depending on the configuration of FDCAN_TSCC.TCP. When the counter decrements to 0, the interrupt flag FDCAN_IR.TOO is set to 1, and the timeout counter stops counting. The

Bit Field	Name	Description
		start and reset/restart conditions are configured through FDCAN_TOCC.TOS. Note: Byte access, when TOCC.TOS = "00", writing register byte 3/2/1/0 will preset the time-out counter.

25.5.13 FDCAN Error Counter Register (FDCAN_ECR)

Address offset: 0x40

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								CEL[7:0]							
								r							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RP	REC[6:0]							TEC[7:0]							
r	r							r							

Bit Field	Name	Description
31:24	Reserved	Reserved, must be kept at reset value.
23:16	CEL[7:0]	CAN Error Logging Whenever a CAN protocol error causes the 8-bit Transmit Error Counter TEC or the 7-bit Receive Error Counter REC to increment, the Counter Error Limit CEL also increments. The increment of CEL occurs after REC or TEC increment. When reading CEL, the count value resets. The counter stops counting when the count value reaches 0xFF. The interrupt flag FDCAN_IR.ELO is set to 1 upon the next increment of TEC or REC.
15	RP	Receive Error Passive 0: Receive error counter below passive error level of 128 1: Receive error counter has reached passive error level of 128
14:8	REC[6:0]	Receive Error Counter Actual status of the receive error counter, with values ranging from 0 to 127
7:0	TEC[7:0]	Transmit Error Counter Actual status of the transmit error counter, with values ranging from 0 to 255. If FDCAN_CCCR.ASM is set to 1, in case of a CAN protocol error, TEC and REC will not increment, only CEL will increment.

25.5.14 FDCAN Protocol Status Register (FDCAN_PSR)

Address offset: 0x44

Reset value: 0x0000 0707

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved									TDCV[6:0]						
									r						

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	PXE	RFDF	RBRS	RESI	DLEC[2:0]			BO	EW	EP	ACT[1:0]		LEC[2:0]		
	r	r	r	r	r			r	r	r	r		r		

Bit Field	Name	Description
31:23	Reserved	Reserved, must be kept at reset value.
22:16	TDCV[6:0]	Transmitter Delay Compensation Value The position of the second sampling point (SSP) is defined as the sum of the delay time measured between pins FDCAN_TX and FDCAN_RX and FDCAN_TDCR.TDCO. During the data phase, the SSP position is the minimum number of time quanta (mtq) between the start of the transmitted bit and the second sampling point. The valid values range from 0 to 127 mtq.
15	Reserved	Reserved, must be kept at reset value.
14	PXE	Protocol Exception Event 0: No protocol exception event has occurred since the last read access 1: A protocol exception event has occurred
13	RFDF	Received a CAN FD Message This bit setting is independent of receive filtering. 0: No CAN FD message has been received since CPU reset 1: CAN FD format message received with FDF flag set to 1
12	RBRS	BRS flag of last received CAN FD Message This bit setting is independent of receive filtering. 0: No CAN FD message has been received since CPU reset 1: CAN FD format message received with FDF flag set to 1
11	RESI	ESI flag of last received CAN FD Message This bit is set together with RFDF and is independent of receive filtering. 0: ESI flag of the last received CAN FD message was not set to 1 1: ESI flag of the last received CAN FD message was set to 1
10:8	DLEC[2:0]	Data Phase Last Error Code The previous error type that occurred during the data phase of a CAN FD format frame with the BRS flag set to 1. The error type is defined according to LEC. This field is cleared when a CAN FD format frame with the BRS flag set to 1 has been successfully transmitted (received or sent) without errors.
7	BO	Bus_Off Status 0: FDCAN is not in Bus_Off state 1: FDCAN is in Bus_Off state
6	EW	Warning Status 0: Values of both error counters are less than the Error Warning limit of 96

Bit Field	Name	Description
		1: At least one error counter has reached the Error Warning limit of 96)
5	EP	Error Passive 0: FDCAN is in Active Error state. It typically participates in bus communication and will send Active Error flags after detecting errors 1: FDCAN is in Passive Error state
4:3	ACT[1:0]	Activity Monitors the communication status of the CAN module. 00: Synchronized: Node is synchronized during CAN communication 01: Idle: Node is neither a receiver nor a transmitter 10: Receiver: Node operates as a receiver 11: Transmitter: Node operates as a transmitter *Note: ACT is set to "00" in case of a protocol exception event.*
2:0	LEC[2:0]	Last Error Code LEC indicates the type of the last error that occurred on the CAN bus. This field is cleared when a message is transmitted on the bus without errors (received or sent). 000: No Error: No errors occurred since the LEC was reset due to a successful message reception or transmission. 001: Form Error: More than 5 consecutive bits of the same level were detected in a received message, which is not allowed. 010: Stuff Error: The fixed format part of the received frame has an incorrect format. 011: Ack Error: A message sent by FDCAN was not acknowledged by other nodes. 100: Bit1 Error: During message transmission (except for the arbitration field), the device wanted to send a recessive level (bit logical value "1"), but the monitored bus value was dominant. 101: Bit0 Error: During message transmission (or ACK bit, active error flag, overload flag), the node wanted to send a dominant level (data or identifier bit logical value "0"), but the monitored bus value was recessive. During Bus_Off recovery, this status is set to 1 each time a sequence of 11 recessive bits is monitored. This allows the CPU to monitor the progression of the Bus_Off recovery sequence (indicating that the bus is not stuck in the dominant state or in continuous disturbance). 110: CRC Error: The CRC checksum of the received message is incorrect. The CRC value in the message does not match the CRC calculated from the received data. 111: No Change: Any read access to the protocol status register will reinitialize LEC to "7". If the value shown by LEC is "7", it indicates that no CAN bus event has been detected since the last read access to the protocol status register by the CPU.

Note 1: When a CAN FD frame data segment with the BRS flag set to 1 is reached, the next CAN event (error or valid frame) will be displayed in DLEC instead of LEC. Additionally, in a CAN FD CRC sequence, fixed stuffing bit errors will be treated as format errors rather than stuff errors.

Note 2: The Bus_Off recovery sequence (refer to CAN Specification 2.0 or ISO 11898-1:2015) cannot be shortened by setting FDCAN_CCCR.INIT to 1 or by reset. If the device enters Bus_Off state, FDCAN_CCCR.INIT is set to 1, and all bus activities are stopped. Once the CPU clears the FDCAN_CCCR.INIT bit, the device will return to normal operation only after waiting for 129 *consecutive bus idle states (129×11 recessive bits). At the end of the Bus_Off recovery sequence, the error management counter values are reset. During the waiting period after the FDCAN_CCCR.INIT reset, each sequence of 11 recessive bits is monitored, and a Bit0 error code is written to FDCAN_PSR.LEC, allowing the CPU to detect whether the CAN bus remains dominant or is continuously disturbed, thus monitoring the Bus_Off recovery sequence. FDCAN_ECR.REC is used to count these sequences of recessive bits.*

25.5.15 FDCAN Transmitter Delay Compensation Register (FDCAN_TDCR)

Address offset: 0x48

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	TDCO[6:0]							Reserved	TDCF[6:0]						
	rw								rw						

Bit Field	Name	Description
31:15	Reserved	Reserved, must be kept at reset value.
14:8	TDCO[6:0]	Transmitter Delay Compensation SSP Offset Defines the difference between the delay measured from FDCAN_TX to FDCAN_RX and the second sampling point. The valid values range from 0 to 127 mtq.
7	Reserved	Reserved, must be kept at reset value.
6:0	TDCF[6:0]	Transmitter Delay Compensation Filter Window Length Defines the minimum value of the SSP position, disregarding dominant edges on FDCAN_RX that cause the SSP position to advance when measuring transmitter delay. The valid values range from 0 to 127 mtq.

25.5.16 FDCAN Interrupt Register (FDCAN_IR)

If the following conditions are detected (edge-sensitive), the corresponding flag is set to 1. The flag remains set to 1 until cleared by writing a "1" to the corresponding bit before host reset. Writing a "0" has no effect. Hardware reset clears the register. IE controls interrupt generation. ILS controls which interrupt line the interrupt is sent from.

Address offset: 0x50

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		ARA	PED	PEA	WDI	BO	EW	EP	ELO	Reserved		DRX	TOO	MRAF	TSW
		rw	rw	rw	rw	rw	rw	rw	rw			rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TEFL	TEFF	TEFW	TEFN	TFE	TCF	TC	HPM	RF1L	RF1F	RF1W	RF1N	RF0L	RF0F	RF0W	RF0N

rw rw rw rw rw rw rw rw rw rw rw rw rw rw rw rw

Bit Field	Name	Description
31:30	Reserved	Reserved, must be kept at reset value.
29	ARA	Access to Reserved Address 0: No access to reserved address 1: Access to reserved address has occurred *Note: Operating the FDCAN_TTSS register will cause the ARA flag to be set abnormally. Write 1 to clear it*
28	PED	Protocol Error in Data Phase (Data Bit Time is used) 0: No protocol error during data phase 1: Protocol error detected during data phase (PSR.DLEC is not 0 or 7)
27	PEA	Protocol Error in Arbitration Phase(Nominal Bit Time is used) 0: No protocol error during data phase 1: Protocol error detected during data phase (PSR.DLEC is not 0 or 7)
26	WDI	Watchdog Interrupt 0: No message RAM watchdog event occurred 1: Message RAM watchdog event occurred due to missing READY
25	BO	Bus_Off Status 0：Bus_Off state unchanged 1：Bus_Off state changed
24	EW	Warning Status 0：Error_Warning state unchanged 1：Error_Warning state changed
23	EP	Error Passive 0：Error_Passive state unchanged 1：Error_Passive state changed
22	ELO	Error Logging Overflow 0: CAN error log counter not overflowed 1: CAN error log counter overflowed
21:20	Reserved	Reserved, must be kept at reset value.
19	DRX	Message stored to Dedicated Rx Buffer After the received message has been stored in the dedicated receive buffer, this flag is set to 1. 0: The receive buffer has not been updated 1: At least one received message has been stored in the dedicated receive buffer.
18	TOO	Timeout Occurred 0: No timeout 1: Timeout occurred
17	MRAF	When the receive processing meets the following conditions, the flag is set to 1: • Until the arbitration field of the next message is received, the reception filtering or storage of the current received message has not been completed. At this point, the filtering or storage of the current received message is aborted, and the next message is processed.

Bit Field	Name	Description
		- The current message cannot be written to the message RAM. At this point, the current message storage is aborted. In both cases, the FIFO put index will not be updated, and the new data flag of the dedicated receive buffer will not be set to 1. When the next message is stored in this position, the original message is overwritten. When the transmit processing unit cannot read data from the message RAM, this flag is also set to 1. At this point, message transmission is aborted. If the transmit processing unit access fails, the FDCAN will switch to restricted operation mode. To exit restricted operation mode, the host CPU must reset FDCAN_CCCR.ASM. 0: No message RAM access failure occurred 1: Message RAM access failure occurred
16	TSW	Timestamp Wraparound 0: Timestamp not rolled over 1: Timestamp rolled over
15	TEFL	Tx Event FIFO Element Lost 0: No loss of transmit event FIFO element 1: Loss of transmit event FIFO element, also set to 1 when attempting to write to a transmit event FIFO of size zero
14	TEFF	Tx Event FIFO Full 0: Transmit event FIFO not full 1: Transmit event FIFO full
13	TEFW	Tx Event FIFO Watermark Reached 0: Transmit event FIFO fill level below watermark flag 1: Transmit event FIFO fill level reaches watermark flag
12	TEFN	Tx Event FIFO New Entry 0: Transmit event FIFO unchanged 1: Transmit processing unit writes new transmit event FIFO element
11	TFE	Tx FIFO Empty 0: Transmit FIFO not empty 1: Transmit FIFO empty
10	TCF	Transmission Cancellation Finished 0: Transmit cancellation not completed 1: Transmit cancellation completed
9	TC	Transmission Completed 0: Transmit cancellation not completed 1: Transmit cancellation completed
8	HPM	High Priority Message 0: No high-priority message received 1: High-priority message received
7	RF1L	x FIFO 1 Message Lost 0: Receive FIFO 1 message not lost 1: Receive FIFO 1 message lost, also set to 1 when attempting to write to a receive FIFO

Bit Field	Name	Description
		1 of size zero
6	RF1F	Rx FIFO 1 Full 0: Receive FIFO 1 not full 1: Receive FIFO 1 full
5	RF1W	Rx FIFO 1 Watermark Reached 0: Receive FIFO 1 fill level below watermark flag 1: Receive FIFO 1 fill level reaches watermark flag
4	RF1N	Rx FIFO 1 New Message 0: No new message written to receive FIFO 1 1: New message written to receive FIFO 1
3	RF0L	Rx FIFO 0 Message Lost 0: Receive FIFO 0 message not lost 1: Receive FIFO 0 message lost, also set to 1 when attempting to write to a receive FIFO 0 of size zero
2	RF0F	Rx FIFO 0 Full 0: Receive FIFO 0 not full 1: Receive FIFO 0 full
1	RF0W	Rx FIFO 0 Watermark Reached 0: Receive FIFO 0 fill level below watermark flag 1: Receive FIFO 0 fill level reaches watermark flag
0	RF0N	Rx FIFO 0 New Message 0: No new message written to receive FIFO 0 1: New message written to receive FIFO 0

25.5.17 FDCAN Interrupt Enable Register (FDCAN_IE)

The settings in the interrupt enable register determine which status changes in the interrupt register will be indicated on the interrupt line.

Address offset: 0x54

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		ARAE	PEDE	PEAE	WDIE	BOE	EWE	EPE	ELOE	Reserved		DRXE	TOOE	MRAFE	TSWE
		rw	rw	rw	rw	rw	rw	rw	rw			rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TEFLE	TEFFE	TEFWE	TEFNE	TFEE	TCFE	TCE	HPME	RF1LE	RF1FE	RF1WE	RF1NE	RF0LE	RF0FE	RF0WE	RF0NE
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit Field	Name	Description
31:30	Reserved	Reserved, must be kept at reset value.
29	ARAE	Access to Reserved Address Enable

Bit Field	Name	Description
		0: Disable interrupt 1: Enable interrupt
28	PEDE	Protocol Error in Data Phase Enable 0: Disable interrupt 1: Enable interrupt
27	PEAE	Protocol Error in Arbitration Phase Enable 0: Disable interrupt 1: Enable interrupt
26	WDIE	Watchdog Interrupt Enable 0: Disable interrupt 1: Enable interrupt
25	BOE	Bus_Off Status 0: Disable interrupt 1: Enable interrupt
24	EWE	Warning Status Interrupt Enable 0: Disable interrupt 1: Enable interrupt
23	EPE	Error Passive Interrupt Enable 0: Disable interrupt 1: Enable interrupt
22	ELOE	Error Logging Overflow Interrupt Enable 0: Disable interrupt 1: Enable interrupt
21:20	Reserved	Reserved, must be kept at reset value.
19	DRXE	Message stored to Dedicated Rx Buffer Interrupt Enable 0: Disable interrupt 1: Enable interrupt
18	TOOE	Timeout Occurred Interrupt Enable 0: Disable interrupt 1: Enable interrupt
17	MRAFE	Message RAM Access Failure Interrupt Enable 0: Disable interrupt 1: Enable interrupt
16	TSWE	Timestamp Wraparound Interrupt Enable 0: Disable interrupt 1: Enable interrupt
15	TEFLE	Tx Event FIFO Element Lost Interrupt Enable 0: Disable interrupt 1: Enable interrupt
14	TEFFE	Tx Event FIFO Full Interrupt Enable 0: Disable interrupt 1: Enable interrupt

Bit Field	Name	Description
13	TEFWE	Tx Event FIFO Watermark Reached Interrupt Enable 0: Disable interrupt 1: Enable interrupt
12	TEFNE	Tx Event FIFO New Entry Interrupt Enable 0: Disable interrupt 1: Enable interrupt
11	TFEE	Tx FIFO Empty Interrupt Enable 0: Disable interrupt 1: Enable interrupt
10	TCFE	Transmission Cancellation Finished Interrupt Enable 0: Disable interrupt 1: Enable interrupt
9	TCE	Transmission Completed Interrupt Enable 0: Disable interrupt 1: Enable interrupt
8	HPME	High Priority Message Interrupt Enable 0: Disable interrupt 1: Enable interrupt
7	RF1LE	Rx FIFO 1 Message Lost Interrupt Enable 0: Disable interrupt 1: Enable interrupt
6	RF1FE	Rx FIFO 1 Full Interrupt Enable 0: Disable interrupt 1: Enable interrupt
5	RF1WE	Rx FIFO 1 Watermark Reached Interrupt Enable 0: Disable interrupt 1: Enable interrupt
4	RF1NE	Rx FIFO 1 New Message Interrupt Enable 0: Disable interrupt 1: Enable interrupt
3	RF0LE	Rx FIFO 0 Message Lost Interrupt Enable 0: Disable interrupt 1: Enable interrupt
2	RF0FE	Rx FIFO 0 Full Interrupt Enable 0: Disable interrupt 1: Enable interrupt
1	RF0WE	Rx FIFO 0 Watermark Reached Interrupt Enable 0: Disable interrupt 1: Enable interrupt
0	RF0NE	Rx FIFO 0 New Message Interrupt Enable 0: Disable interrupt 1: Enable interrupt

25.5.18 FDCAN Interrupt Line Select Register (FDCAN_ILS)

The interrupt line select register assigns interrupts generated by specific interrupt flags in the interrupt register to one of the two module interrupt lines. To generate an interrupt, the respective interrupt line must be enabled through ILE.EINT0 and ILE.EINT1.

Address offset: 0x58

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		ARAL	PEDL	PEAL	WDIL	BOL	EWL	EPL	ELOL	Reserved		DRXL	TOOL	MRAFL	TSWL
		rw	rw	rw	rw	rw	rw	rw	rw			rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TEFLL	TEFFL	TEFWL	TEFNL	TFEL	TCFL	TCL	HPML	RF1LL	RF1FL	RF1WL	RF1NL	RF0LL	RF0FL	RF0WL	RF0NL
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit Field	Name	Description
31:30	Reserved	Reserved, must be kept at reset value.
29	ARAL	Access to Reserved Address Line 0: Interrupt assigned to interrupt line fdcan_int0 1: Interrupt assigned to interrupt line fdcan_int1
28	PEDL	Protocol Error in Data Phase Line 0: Interrupt assigned to interrupt line fdcan_int0 1: Interrupt assigned to interrupt line fdcan_int1
27	PEAL	Protocol Error in Arbitration Phase Line 0: Interrupt assigned to interrupt line fdcan_int0 1: Interrupt assigned to interrupt line fdcan_int1
26	WDIL	Watchdog Interrupt Line 0: Interrupt assigned to interrupt line fdcan_int0 1: Interrupt assigned to interrupt line fdcan_int1
25	BOL	Bus_Off Status 0: Interrupt assigned to interrupt line fdcan_int0 1: Interrupt assigned to interrupt line fdcan_int1
24	EWL	Warning Status Interrupt Line 0: Interrupt assigned to interrupt line fdcan_int0 1: Interrupt assigned to interrupt line fdcan_int1
23	EPL	Error Passive Interrupt Line 0: Interrupt assigned to interrupt line fdcan_int0 1: Interrupt assigned to interrupt line fdcan_int1
22	ELOL	Error Logging Overflow Interrupt Line 0: Interrupt assigned to interrupt line fdcan_int0 1: Interrupt assigned to interrupt line fdcan_int1

Bit Field	Name	Description
21:20	Reserved	Reserved, must be kept at reset value.
19	DRXL	Message stored to Dedicated Rx Buffer Interrupt Line 0: Interrupt assigned to interrupt line fdcan_int0 1: Interrupt assigned to interrupt line fdcan_int1
18	TOOL	Timeout Occurred Interrupt Line 0: Interrupt assigned to interrupt line fdcan_int0 1: Interrupt assigned to interrupt line fdcan_int1
17	MRAFL	Message RAM Access Failure Interrupt Line 0: Interrupt assigned to interrupt line fdcan_int0 1: Interrupt assigned to interrupt line fdcan_int1
16	TSWL	Timestamp Wraparound Interrupt Line 0: Interrupt assigned to interrupt line fdcan_int0 1: Interrupt assigned to interrupt line fdcan_int1
15	TEFLL	Tx Event FIFO Element Lost Interrupt Line 0: Interrupt assigned to interrupt line fdcan_int0 1: Interrupt assigned to interrupt line fdcan_int1
14	TEFFL	Tx Event FIFO Full Interrupt Line 0: Interrupt assigned to interrupt line fdcan_int0 1: Interrupt assigned to interrupt line fdcan_int1
13	TEFWL	Tx Event FIFO Watermark Reached Interrupt Line 0: Interrupt assigned to interrupt line fdcan_int0 1: Interrupt assigned to interrupt line fdcan_int1
12	TEFNL	Tx Event FIFO New Entry Interrupt Line 0: Interrupt assigned to interrupt line fdcan_int0 1: Interrupt assigned to interrupt line fdcan_int1
11	TFEL	Tx FIFO Empty Interrupt Line 0: Interrupt assigned to interrupt line fdcan_int0 1: Interrupt assigned to interrupt line fdcan_int1
10	TCFL	Transmission Cancellation Finished Interrupt Line 0: Interrupt assigned to interrupt line fdcan_int0 1: Interrupt assigned to interrupt line fdcan_int1
9	TCL	Transmission Completed Interrupt Line 0: Interrupt assigned to interrupt line fdcan_int0 1: Interrupt assigned to interrupt line fdcan_int1
8	HPML	High Priority Message Interrupt Line 0: Interrupt assigned to interrupt line fdcan_int0 1: Interrupt assigned to interrupt line fdcan_int1
7	RF1LL	Rx FIFO 1 Message Lost Interrupt Line 0: Interrupt assigned to interrupt line fdcan_int0 1: Interrupt assigned to interrupt line fdcan_int1
6	RF1FL	Rx FIFO 1 Full Interrupt Line 0: Interrupt assigned to interrupt line fdcan_int0

Bit Field	Name	Description
		1: Interrupt assigned to interrupt line fdcan_int1
5	RF1WL	Rx FIFO 1 Watermark Reached Interrupt Line 0: Interrupt assigned to interrupt line fdcan_int0 1: Interrupt assigned to interrupt line fdcan_int1
4	RF1NL	Rx FIFO 1 New Message Interrupt Line 0: Interrupt assigned to interrupt line fdcan_int0 1: Interrupt assigned to interrupt line fdcan_int1
3	RF0LL	Rx FIFO 0 Message Lost Interrupt Line 0: Interrupt assigned to interrupt line fdcan_int0 1: Interrupt assigned to interrupt line fdcan_int1
2	RF0FL	Rx FIFO 0 Full Interrupt Line 0: Interrupt assigned to interrupt line fdcan_int0 1: Interrupt assigned to interrupt line fdcan_int1
1	RF0WL	Rx FIFO 0 Watermark Reached Interrupt Line 0: Interrupt assigned to interrupt line fdcan_int0 1: Interrupt assigned to interrupt line fdcan_int1
0	RF0NL	Rx FIFO 0 New Message Interrupt Line 0: Interrupt assigned to interrupt line fdcan_int0 1: Interrupt assigned to interrupt line fdcan_int1

25.5.19 FDCAN Interrupt Line Enable Register (FDCAN_ILE)

You can enable/disable the two interrupt lines separately by configuring EINT0 and EINT1.

Address offset: 0x5C

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved														EINT1	EINT0
														rw	rw

Bit Field	Name	Description
31:2	Reserved	Reserved, must be kept at reset value.
1	EINT1	Enable Interrupt Line 1 0: Disable interrupt line fdcan_int1 1: Enable interrupt line fdcan_int1
0	EINT0	Enable Interrupt Line 0 0: Disable interrupt line fdcan_int0 1: Enable interrupt line fdcan_int0

25.5.20 FDCAN Global Filter Configuration Register (FDCAN_GFC)

Used for setting the global filter for message IDs, controls the filter path for standard and extended messages.

Address offset: 0x80

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										ANFS[1:0]		ANFE[1:0]		RRFS	RRFE
										rw		rw		rw	rw

Bit Field	Name	Description
31:6	Reserved	Reserved, must be kept at reset value.
5:4	ANFS[1:0]	Accept Non-matching Frames Standard Defines the handling of 11-bit ID messages received that do not match any element in the filter list. 00: Accept in receive FIFO 0 01: Accept in receive FIFO 1 10: Reject 11: Reject These bits are write-protected and can only be accessed for writing when FDCAN_CCCR.CCE and FDCAN_CCCR.INIT are both 1.
3:2	ANFE[1:0]	Accept Non-matching Frames Extended Defines the handling of 29-bit ID messages received that do not match any element in the filter list. 00: Accept in receive FIFO 0 01: Accept in receive FIFO 1 10: Reject 11: Reject These bits are write-protected and can only be accessed for writing when FDCAN_CCCR.CCE and FDCAN_CCCR.INIT are both 1.
1	RRFS	Reject Remote Frames Standard 0: Accept remote frames with 11-bit standard ID for filtering 1: Reject all remote frames with 11-bit standard ID These bits are write-protected and can only be accessed for writing when FDCAN_CCCR.CCE and FDCAN_CCCR.INIT are both 1.
0	RRFE	Reject Remote Frames Extended 0: Accept remote frames with 29-bit extended ID for filtering 1: Reject all remote frames with 29-bit extended ID These bits are write-protected and can only be accessed for writing when

Bit Field	Name	Description
		FDCAN_CCCR.CCE and FDCAN_CCCR.INIT are both 1.

25.5.21 FDCAN Standard ID Filter Configuration Register (FDCAN_SIDFC)

Used for setting 11-bit standard message ID filtering. The standard ID filter configuration controls the filter path for standard messages, refer to the diagram6.

Address offset: 0x84

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								LSS[7:0]							
								rw							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FLSSA[13:0]														Reserved	
rw															

Bit Field	Name	Description
31:24	Reserved	Reserved, must be kept at reset value.
23:16	LSS[7:0]	List Size Standard 0: No standard message ID filters 1-128: Number of standard message ID filter elements >128: Values greater than 128 will be interpreted as 128. These bits are write-protected and can only be accessed for writing when FDCAN_CCCR.CCE and FDCAN_CCCR.INIT are both 1.
15:2	FLSSA[13:0]	Filter List Standard Start Address The offset of the start address of the standard message ID filter list storage area relative to the start address of the message RAM, in words (32-bit units). These bits are write-protected and can only be accessed for writing when FDCAN_CCCR.CCE and FDCAN_CCCR.INIT are both 1.
1:0	Reserved	Reserved, must be kept at reset value.

25.5.22 FDCAN Extended ID Filter Configuration Register (FDCAN_XIDFC)

Used for setting 29-bit extended message ID filtering. The extended ID filter configuration controls the filter path for extended messages, refer to Figure 7

Address offset: 0x88

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved									LSE[6:0]						

rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FLESA[13:0]														Reserved	
rw															

Bit Field	Name	Description
31:23	Reserved	Reserved, must be kept at reset value.
22:16	LSE[6:0]	List Size Extended 0: No extended message ID filters 1-64: Number of extended message ID filter elements >64: Values greater than 64 will be interpreted as 64. These bits are write-protected and can only be accessed for writing when FDCAN_CCCR.CCE and FDCAN_CCCR.INIT are both 1.
15:2	FLESA[13:0]	Filter List Extended Start Address The offset of the start address of the extended message ID filter storage area relative to the start address of the message RAM, in words (32-bit units). These bits are write-protected and can only be accessed for writing when FDCAN_CCCR.CCE and FDCAN_CCCR.INIT are both 1.
1:0	Reserved	Reserved, must be kept at reset value.

25.5.23 FDCAN Extended ID and Mask Register (FDCAN_XIDAM)

Address offset: 0x90

Reset value: 0x1FFF FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved			EIDM[28:16]												
			rw												

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EIDM[15:0]															
rw															

Bit Field	Name	Description
31:29	Reserved	Reserved, must be kept at reset value.
28:0	EIDM[28:0]	Extended ID Mask Before receiving filtering for extended frames, the extended ID mask is first ANDed with the message ID of the received frame. Used to mask the 29-bit ID in SAE J1939. All bits are set to 1 after reset, making the mask invalid. These bits are write-protected and can only be accessed for writing when FDCAN_CCCR.CCE and FDCAN_CCCR.INIT are both 1.

25.5.24 FDCAN High Priority Message Status Register (FDCAN_HPMS)

When the message ID filter element is configured to generate a priority event, this register is updated with each match. This register can be used to monitor the status of incoming high-priority messages for quick access.

Address offset: 0x94

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FLST	FIDX[6:0]							MSI[1:0]		BIDX[5:0]					
r	r							r		r					

Bit Field	Name	Description
31:16	Reserved	Reserved, must be kept at reset value.
15	FLST	Filter List Indicates the filter list where the matching filter element is located. 0: Standard filter list 1: Extended filter list
14:8	FIDX[6:0]	Filter Index Index of the filter element that matches the current message. Range is from 0 to FDCAN_SIDFC.LSS-1 or FDCAN_XIDFC.LSE-1.
7:6	MSI[1:0]	Message Storage Indicator 00: No FIFO selected 01: FIFO message lost 10: Message stored in FIFO 0 11: Message stored in FIFO 1
5:0	BIDX[5:0]	Buffer Index Index of the receive FIFO element where the message is stored. Only valid when MSI[1] = "1".

25.5.25 FDCAN New Data 1 Register (FDCAN_NDAT1)

Address offset: 0x98

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ND31	ND30	ND29	ND28	ND27	ND26	ND25	ND24	ND23	ND22	ND21	ND20	ND19	ND18	ND17	ND16
rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

ND15	ND14	ND13	ND12	ND11	ND10	ND9	ND8	ND7	ND6	ND5	ND4	ND3	ND2	ND1	ND0
rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1

Bit Field	Name	Description
31:0	ND[31:0]	New Data[31:0] This register is the New Data flag for receive buffers 0 to 31. When a receive buffer is updated with a received frame, the corresponding flag is set to 1. The flag remains set to 1 until cleared by the host. Writing a 1 to the corresponding bit clears the flag. Writing "0" has no effect. The register is cleared on hardware reset. 0: Receive buffer not updated 1: New message received in receive buffer

25.5.26 FDCAN New Data 2 Register (FDCAN_NDAT2)

Address offset: 0x9C

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ND63	ND62	ND61	ND60	ND59	ND58	ND57	ND56	ND55	ND54	ND53	ND52	ND51	ND50	ND49	ND48
rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ND47	ND46	ND45	ND44	ND43	ND42	ND41	ND40	ND39	ND38	ND37	ND36	ND35	ND34	ND33	ND32
rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1

Bit Field	Name	Description
31:0	ND[63:32]	New Data[63:32] This register is the New Data flag for receive buffers 32 to 63. When a receive buffer is updated with a received frame, the corresponding flag is set to 1. The flag remains set to 1 until cleared by the host. Writing a 1 to the corresponding bit clears the flag. Writing "0" has no effect. The register is cleared on hardware reset. 0: Receive buffer not updated 1: New message received in receive buffer

25.5.27 FDCAN Receive FIFO 0 Configuration Register (FDCAN_RXF0C)

Address offset: 0xA0

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
F0OM	F0WM[6:0]							Reserved	F0S[6:0]						
rw	rw								rw						
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

F0SA[13:0]	Reserved
rw	

Bit Field	Name	Description
31	F0OM	Rx FIFO 0 Operation mode 0: Receive FIFO 0 in blocking mode 1: Receive FIFO 0 in overwrite mode
30:24	F0WM[6:0]	0: Disable watermark interrupt 1-64: Level of receive FIFO 0 watermark interrupt (FDCAN_IR.RF0W) >64: Disable watermark interrupt These bits are write-protected and can only be accessed for writing when FDCAN_CCCR.CCE and FDCAN_CCCR.INIT are both 1.
23	Reserved	Reserved, must be kept at reset value.
22:16	F0S[6:0]	Rx FIFO 0 Size 0: No receive FIFO 0 1-64: Number of elements in receive FIFO 0 >64: Values greater than 64 will be interpreted as 64 The index range of receive FIFO 0 elements is from 0 to F0S-1.
15:2	F0SA[13:0]	Rx FIFO 0 Start Address The offset of the start address of the receive FIFO 0 storage area in the message RAM relative to the start address of the message RAM, in words (32-bit units).)
1:0	Reserved	Reserved, must be kept at reset value.

25.5.28 FDCAN Receive FIFO 0 Status Register (FDCAN_RXF0S)

Address offset: 0xA4

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved						RF0L	F0F	Reserved		F0PI[5:0]					
						r	r			r					

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		F0GI[5:0]						Reserved	F0FL[6:0]						
		r							r						

Bit Field	Name	Description
31:26	Reserved	Reserved, must be kept at reset value.
25	RF0L	Rx FIFO 0 Message Lost When FDCAN_IR.RF0L is reset, this bit is also reset. 0: No message lost in receive FIFO 0 1: Message lost in receive FIFO 0, also set to 1 when attempting to write to a receive FIFO 0 of size zero
24	F0F	Rx FIFO 0 Full

Bit Field	Name	Description
		0: Receive FIFO 0 not full 1: Receive FIFO 0 full
23:22	Reserved	Reserved, must be kept at reset value.
21:16	F0PI[5:0]	Rx FIFO 0 Put Index Receive FIFO 0 write index pointer, range from 0 to 63.
15:14	Reserved	Reserved, must be kept at reset value.
13:8	F0GI[5:0]	Rx FIFO 0 Get Index Receive FIFO 0 write index pointer, range from 0 to 63.
7	Reserved	Reserved, must be kept at reset value.
6:0	F0FL[6:0]	Rx FIFO 0 Fill Level Receive FIFO 0 write index pointer, range from 0 to 64.

25.5.29 FDCAN Receive FIFO 0 Acknowledge Register (FDCAN_RXF0A)

Address offset: 0xA8

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										F0AI[5:0]					
										rw					

Bit Field	Name	Description
31:6	Reserved	Reserved, must be kept at reset value.
5:0	F0AI[5:0]	Rx FIFO 0 Acknowledge Index After the host reads a message or message sequence from receive FIFO 0, the host must write the buffer index of the last read element to F0AI. This operation sets the receive FIFO 0 get index FDCAN_RXF0S.F0GI to F0AI + 1 and updates the FIFO 0 fill level FDCAN_RXF0S.F0FL.

25.5.30 FDCAN Receive Buffer Configuration Register (FDCAN_RXBC)

Address offset: 0xAC

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RBSA[13:0]														Reserved	
rw															

Bit Field	Name	Description
31:16	Reserved	Reserved, must be kept at reset value.
15:2	RBSA[13:0]	Rx Buffer Start Address Configures the start address (32-bit word address) of the dedicated receive buffer section in the message RAM. Also used for referencing debug messages A, B, C. These bits are write-protected and can only be accessed for writing when FDCAN_CCCR.CCE and FDCAN_CCCR.INIT are both 1.
1:0	Reserved	Reserved, must be kept at reset value.

25.5.31 FDCAN Receive FIFO 1 Configuration Register (FDCAN_RXF1C)

Address offset: 0xB0

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
F1OM	F1WM[6:0]							Reserved	F1S[6:0]						
rw	rw								rw						

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
F1SA[13:0]														Reserved	
rw															

Bit Field	Name	Description
31	F1OM	Rx FIFO 1 Operation mode 0: Receive FIFO 1 in blocking mode 1: Receive FIFO 1 in overwrite mode
30:24	F1WM[6:0]	0: Disable watermark interrupt 1-64: Level of receive FIFO 1 watermark interrupt (FDCAN_IR.RF1W) >64: Disable watermark interrupt These bits are write-protected and can only be accessed for writing when FDCAN_CCCR.CCE and FDCAN_CCCR.INIT are both 1.
23	Reserved	Reserved, must be kept at reset value.
22:16	F1S[6:0]	Rx FIFO 1 Size 0: No receive FIFO 1 1-64: Number of elements in receive FIFO 1 >64: Values greater than 64 will be interpreted as 64 The index range of receive FIFO 1 elements is from 0 to F1S-1.
15:2	F1SA	Rx FIFO 1 Start Address The offset of the start address of the receive FIFO 1 storage area in the message RAM relative to the start address of the message RAM, in words (32-bit units).)

Bit Field	Name	Description
1:0	Reserved	Reserved, must be kept at reset value.

25.5.32 FDCAN Receive FIFO 1 Status Register (FDCAN_RXF1S)

Address offset: 0xB4

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DMS[1:0]		Reserved				RF1L	F1F	Reserved		F1PI[5:0]					
r						r	r			r					

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		F1GI[5:0]						Reserved	F1FL[6:0]						
		r							r						

Bit Field	Name	Description
31:30	DMS[1:0]	Debug Message Status 00: Idle state, waiting to receive debug message 01: Debug message A received 10: Debug messages A and B received 11: Debug messages A, B, and C received
29:26	Reserved	Reserved, must be kept at reset value.
25	RF1L	Rx FIFO 1 Message Lost When FDCAN_IR.RF1L is reset, this bit is also reset. 0: No message lost in receive FIFO 1 1: Message lost in receive FIFO 1, also set to 1 when attempting to write to a receive FIFO 1 of size zero
24	F1F	Rx FIFO 1 Full 0: Receive FIFO 1 not full 1: Receive FIFO 1 full
23:22	Reserved	Reserved, must be kept at reset value.
21:16	F1PI[5:0]	Rx FIFO 1 Put Index Receive FIFO 1 write index pointer, range from 0 to 63.
15:14	Reserved	Reserved, must be kept at reset value.
13:8	F1GI[5:0]	Rx FIFO 1 Get Index Receive FIFO 1 write index pointer, range from 0 to 63.
7	Reserved	Reserved, must be kept at reset value.
6:0	F1FL[6:0]	Rx FIFO 1 Fill Level Receive FIFO 1 write index pointer, range from 0 to 64.

25.5.33 FDCAN Receive FIFO 1 Acknowledge Register (FDCAN_RXF1A)

Address offset: 0xB8

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										F1AI[5:0]					
										rw					

Bit Field	Name	Description
31:6	Reserved	Reserved, must be kept at reset value.
5:0	F1AI[5:0]	Rx FIFO 1 Acknowledge Index After the host reads a message or message sequence from receive FIFO 1, the host must write the buffer index of the last read element to F1AI. This operation sets the receive FIFO 1 get index FDCAN_RXF1S.F1GI to F1AI + 1 and updates the FIFO 1 fill level FDCAN_RXF1S.F1FL.

25.5.34 FDCAN Receive Buffer/FIFO Element Size Configuration Register (FDCAN_RXESC)

Configures the number of data field bytes for receive buffer and receive FIFO elements. Data field sizes greater than 8 bytes are only used for CAN FD.

Address offset: 0xBC

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved					RBDS[2:0]			Reserved	F1DS[2:0]			Reserved	F0DS[2:0]		
					rw				rw				rw		

Bit Field	Name	Description
31:11	Reserved	Reserved, must be kept at reset value.
10:8	RBDS[2:0]	Rx Buffer Data Field Size 000: 8-byte data field 001: 12-byte data field 010: 16-byte data field 011: 20-byte data field 100: 24-byte data field 101: 32-byte data field

Bit Field	Name	Description
		110: 48-byte data field 111: 64-byte data field
7	Reserved	Reserved, must be kept at reset value.
6:4	F1DS[2:0]	Rx FIFO 0 Data Field Size 000: 8-byte data field 001: 12-byte data field 010: 16-byte data field 011: 20-byte data field 100: 24-byte data field 101: 32-byte data field 110: 48-byte data field 111: 64-byte data field
3	Reserved	Reserved, must be kept at reset value.
2:0	F0DS[2:0]	Rx FIFO 1 Data Field Size 000: 8-byte data field 001: 12-byte data field 010: 16-byte data field 011: 20-byte data field 100: 24-byte data field 101: 32-byte data field 110: 48-byte data field 111: 64-byte data field

Note: If the data field size of the received CAN frame exceeds the data field size configured for the Rx buffer or Rx FIFO, only the number of bytes configured by RXESC will be stored in the Rx buffer or Rx FIFO element, and the rest will be ignored.

25.5.35 FDCAN Transmit Buffer Configuration Register (FDCAN_TXBC)

Address offset: 0xC0

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved	TFQM	TFQS[5:0]						Reserved		NDTB[5:0]					
	rw	rw								rw					

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TBSA[13:0]														Reserved	
rw															

Bit Field	Name	Description
31	Reserved	Reserved, must be kept at reset value.
30	TFQM	Tx FIFO/Queue Mode

Bit Field	Name	Description
		0: Transmit FIFO mode 1: Transmit queue mode These bits are write-protected and can only be accessed for writing when FDCAN_CCCR.CCE and FDCAN_CCCR.INIT are both 1.
29:24	TFQS[5:0]	Transmit FIFO/Queue Size 0: No transmit FIFO/queue 1-32: Number of transmit buffers used for transmit FIFO/queue >32: Values greater than 32 will be interpreted as 32 These bits are write-protected and can only be accessed for writing when FDCAN_CCCR.CCE and FDCAN_CCCR.INIT are both 1.
23:22	Reserved	Reserved, must be kept at reset value.
21:16	NDTB[5:0]	Number of Dedicated Transmit Buffers 0: No dedicated transmit buffer 1-32: Number of dedicated transmit buffers >32: Values greater than 32 will be interpreted as 32 These bits are write-protected and can only be accessed for writing when FDCAN_CCCR.CCE and FDCAN_CCCR.INIT are both 1.
15:2	TBSA[13:0]	Tx Buffer Start Address Start address (32-bit word address) of the transmit buffer section in the message RAM. These bits are write-protected and can only be accessed for writing when FDCAN_CCCR.CCE and FDCAN_CCCR.INIT are both 1.
1:0	Reserved	Reserved, must be kept at reset value.

Note: The sum of TFQS and NDTB cannot exceed 32. FDCAN does not check for configuration errors. The transmit buffer section in the message RAM starts from the dedicated transmit buffer.

25.5.36 FDCAN Transmit FIFO/Queue Status Register (FDCAN_TXFQS)

The transmit FIFO/queue status is related to the transmit requests in FDCAN_TXBRP. Therefore, adding/canceling transmit requests may be delayed due to an ongoing transmit scan (FDCAN_TXBRP has not been updated yet).

Address offset: 0xC4

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved										TFQF	TFQPI[4:0]				
										r	r				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			TFGI[4:0]					Reserved		TFFL[5:0]					
			r							r					

Bit Field	Name	Description
31:22	Reserved	Reserved, must be kept at reset value.
21	TFQF	Tx FIFO/Queue Full 0: Transmit FIFO/queue not full 1: Transmit FIFO/queue full
20:16	TFQPI[4:0]	Tx FIFO/Queue Put Index Transmit FIFO/queue write index pointer, range from 0 to 31
15:13	Reserved	Reserved, must be kept at reset value.
12:8	TFGI[4:0]	Tx FIFO Get Index Transmit FIFO read index pointer, range from 0 to 31. If configured as a transmit queue (FDCAN_TXBC.TFQM = "1"), the readout value is always zero.
7:6	Reserved	Reserved, must be kept at reset value.
5:0	TFFL[5:0]	Tx FIFO Free Level Number of consecutive free transmit FIFO elements starting from TFGI, range from 0 to 32. If transmit queue operation is configured (FDCAN_TXBC.TFQM = "1"), the readout value is always zero.

Note: In a hybrid configuration combining dedicated transmit buffers with transmit FIFO or transmit queue, the write and get index pointers indicate the transmit buffer index starting from the first dedicated transmit buffer.

For example: In a hybrid configuration combining 12 dedicated transmit buffers with a transmit FIFO containing 20 buffers, a write index of 15 would point to the fourth transmit buffer in the transmit FIFO.

25.5.37 FDCAN Transmit Buffer Element Size Configuration Register (FDCAN_TXESC)

Configures the number of data segment bytes for transmit buffer elements. Data segment sizes greater than 8 bytes are only used for CAN FD.

Address offset: 0xC8

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													TBDS[2:0]		
													rw		

Bit Field	Name	Description
31:3	Reserved	Reserved, must be kept at reset value.
2:0	TBDS[2:0]	Tx Buffer Data Field Size 000: 8-byte data field

Bit Field	Name	Description
		001: 12-byte data field 010: 16-byte data field 011: 20-byte data field 100: 24-byte data field 101: 32-byte data field 110: 48-byte data field 111: 64-byte data field

25.5.38 FDCAN Transmit Buffer Request Pending Register (FDCAN_TXBRP)

Address offset: 0xCC

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TRP31	TRP30	TRP29	TRP28	TRP27	TRP26	TRP25	TRP24	TRP23	TRP22	TRP21	TRP20	TRP19	TRP18	TRP17	TRP16
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRP15	TRP14	TRRP13	TRP112	TRP11	TRP10	TRP9	TRP8	TRP7	TRP6	TRP5	TRP4	TRP3	TRP2	TRP1	TRP0
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

Bit Field	Name	Description
31:0	TRP[31:0]	Transmission Request Pending Each transmit buffer has its own transmit request pending bit, set to 1 by the FDCAN_TXBAR register. The corresponding bit is reset once the requested transmission is completed or canceled via FDCAN_TXBCR. FDCAN_TXBRP will only set the position corresponding to the valid transmit buffer configured by FDCAN_TXBC to 1. Once a position in FDCAN_TXBRP is set to 1, a transmit scan is initiated to check the transmit request with the highest priority (the transmit buffer with the smallest message ID). Canceling a request will reset the corresponding transmit request pending bit in the FDCAN_TXBRP register. If the transmission has already started when the request is canceled, the reset operation will be performed at the end of the transmission, regardless of its success. Once the corresponding FDCAN_TXBRP bit is reset, the cancel request bit will also be immediately reset. A request cancellation will be indicated as successful by setting the FDCAN_TXBCF bit to 1 under the following conditions: - Successful transmission, with the corresponding FDCAN_TXBTO bit set to 1 - Transmission was canceled before it started - Transmission was aborted due to arbitration loss

Bit Field	Name	Description
		- Error occurred during transmission In DAR mode, all transmissions are automatically canceled after a transmission failure. For all unsuccessful transmissions, the corresponding FDCAN_TXBCF bit is set to 1. 0: No transmit request pending 1: Transmit request pending

Note: If an FDCAN_TXBRP bit is set to 1 during an ongoing transmit scan, the current transmit scan will ignore this bit. If a request to cancel such a transmit buffer is made, the request will be immediately canceled, and the corresponding FDCAN_TXBRP bit will be reset.

25.5.39 FDCAN Transmit Buffer Add Request Register (TXBAR)

Address offset: 0xD0

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AR31	AR30	AR29	AR28	AR27	AR26	AR25	AR24	AR23	AR22	AR21	AR20	AR19	AR18	AR17	AR16
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AR15	AR14	AR13	AR12	AR11	AR10	AR9	AR8	AR7	AR6	AR5	AR4	AR3	AR2	AR1	AR0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit Field	Name	Description
31:0	AR[31:0]	Add Request Each transmit buffer has its own add request bit. Writing a 1 will set the corresponding add request bit to 1, while writing a 0 has no effect. A single write operation by the host to TXBAR can add transmit requests for multiple transmit buffers, but only the valid transmit buffer configured by FDCAN_TXBC will be set to 1. If no transmit scan is running, these bits will be immediately reset; otherwise, they will remain set to 1 until the transmit scan process is completed. 0: No transmit request added 1: Transmit request added

Note: If a transmit buffer for which the transmit request bit is already set to 1 receives an add request, the duplicate request will be ignored.

25.5.40 FDCAN Transmit Buffer Cancellation Request Register (FDCAN_TXBCR)

Address offset: 0xD4

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CR31	CR30	CR29	CR28	CR27	CR26	CR25	CR24	CR23	CR22	CR21	CR20	CR19	CR18	CR17	CR16
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CR15	CR14	CR13	CR12	CR11	CR10	CR9	CR8	CR7	CR6	CR5	CR4	CR3	CR2	CR1	CR0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit Field	Name	Description
31:0	CR[31:0]	Cancellation Request Each transmit buffer has its own cancel request bit. Writing a 1 will set the corresponding cancel request bit to 1, while writing a 0 has no effect. A single write operation by the host to FDCAN_TXBCR can set cancel requests for multiple transmit buffers, but only the valid transmit buffer configured by FDCAN_TXBC will be set to 1. These bits will remain set to 1 until the corresponding FDCAN_TXBRP bit is reset. 0: No cancel request 1: Cancel requested

25.5.41 FDCAN Transmit Buffer Transmission Occurred Register (FDCAN_TXBTO)

Address offset: 0xD8

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TO31	TO30	TO29	TO28	TO27	TO26	TO25	TO24	TO23	TO22	TO21	TO20	TO19	TO18	TO17	TO16
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TO15	TO14	TO13	TO12	TO11	TO10	TO9	TO8	TO7	TO6	TO5	TO4	TO3	TO2	TO1	TO0
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

Bit Field	Name	Description
31:0	TO[31:0]	Transmission Occurred Each transmit buffer has its own transmission occurred bit. When the corresponding FDCAN_TXBRP bit is cleared after a successful transmission, the respective bit in this register is set to 1. These bits are reset when a new transmission is requested by writing a 1 to the corresponding bit in FDCAN_TXBAR. 0: No transmission occurred 1: Transmission occurred

25.5.42 FDCAN Transmit Buffer Cancellation Finished Register (FDCAN_TXBCF)

Address offset: 0xDC

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CF31	CF30	CF29	CF28	CF27	CF26	CF25	CF24	CF23	CF22	CF21	CF20	CF19	CF18	CF17	CF16
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CF15	CF14	CF13	CF12	CF11	CF10	CF9	CF8	CF7	CF6	CF5	CF4	CF3	CF2	CF1	CF0
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

Bit Field	Name	Description
31:0	CF[31:0]	Cancellation Finished Each transmit buffer has its own cancellation finished bit. When the corresponding FDCAN_TXBRP bit is cleared after a cancel request through the respective FDCAN_TXBCR bit, the corresponding bit in this register is set to 1. If the corresponding bit in this register is not set to 1 when a cancel request is made, the bit in this register is immediately set to 1. When a 1 is written to the corresponding bit in the FDCAN_TXBAR register to request a new transmission, the bit in this register is reset. 0: No transmit buffer cancellation 1: Transmit buffer cancellation finished

25.5.43 FDCAN Transmit Buffer Transmission Interrupt Enable Register (FDCAN_TXBTIE)

Address offset: 0xE0

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TIE31	TIE30	TIE29	TIE28	TIE27	TIE26	TIE25	TIE24	TIE23	TIE22	TIE21	TIE20	TIE19	TIE18	TIE17	TIE16
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIE15	TIE14	TIE13	TIE12	TIE11	TIE10	TIE9	TIE8	TIE7	TIE6	TIE5	TIE4	TIE3	TIE2	TIE1	TIE0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit Field	Name	Description
31:0	TIE[31:0]	Transmission Interrupt Enable Each transmit buffer has its own transmission interrupt enable bit.

Bit Field	Name	Description
		0: Disable transmission interrupt 1: Enable transmission interrupt

25.5.44 FDCAN Transmit Buffer Cancellation Finished Interrupt Enable Register (FDCAN_TXBCIE)

Address offset: 0xE4

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CFIE31	CFIE30	CFIE29	CFIE28	CFIE27	CFIE26	CFIE25	CFIE24	CFIE23	CFIE22	CFIE21	CFIE20	CFIE19	CFIE18	CFIE17	CFIE16
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CFIE15	CFIE14	CFIE13	CFIE12	CFIE11	CFIE10	CFIE9	CFIE8	CFIE7	CFIE6	CFIE5	CFIE4	CFIE3	CFIE2	CFIE1	CFIE0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit Field	Name	Description
31:0	CFIE[31:0]	Cancellation Finished Interrupt Enable Each transmit buffer has its own cancel completion interrupt enable bit 0: Disable cancel completion interrupt 1: Enable cancel completion interrupt

25.5.45 FDCAN Transmit Event FIFO Configuration Register (FDCAN_TXEFC)

Address offset: 0xF0

Reset value: 0x0000 0000

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="2">Reserved</td><td colspan="6">EFWM[5:0]</td><td colspan="2">Reserved</td><td colspan="6">EFS[5:0]</td></tr>
<tr><td colspan="2"></td><td colspan="6">rw</td><td colspan="2"></td><td colspan="6">rw</td></tr>
</table>

<table>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="14">EFSA[13:0]</td><td colspan="2">Reserved</td></tr>
<tr><td colspan="14">rw</td><td colspan="2"></td></tr>
</table>

Bit Field	Name	Description
31:30	Reserved	Reserved, must be kept at reset value.
29:24	EFWM[5:0]	Event FIFO Watermark 0: Disable watermark interrupt 1-32: Level of transmit event FIFO watermark interrupt (FDCAN_IR.TEFW)

Bit Field	Name	Description
		>32: Disable watermark interrupt These bits are write-protected and can only be accessed for writing when FDCAN_CCCR.CCE and FDCAN_CCCR.INIT are both 1.
23:22	Reserved	Reserved, must be kept at reset value.
21:16	EFS[5:0]	Event FIFO Size 0: Disable transmit event FIFO 1-32: Number of transmit event FIFO elements >32: Values greater than 32 will be interpreted as 32 The index of transmit event FIFO elements is from 0 to EFS – 1 These bits are write-protected and can only be accessed for writing when FDCAN_CCCR.CCE and FDCAN_CCCR.INIT are both 1.
15:2	EFSA[13:0]	Event FIFO Start Address The offset of the start address of the transmit event FIFO storage area in the message RAM relative to the start address of the message RAM, in words (32-bit units). These bits are write-protected and can only be accessed for writing when FDCAN_CCCR.CCE and FDCAN_CCCR.INIT are both 1.
1:0	Reserved	Reserved, must be kept at reset value.

25.5.46 FDCAN Transmit Event FIFO Status Register (FDCAN_TXEFS)

Address offset: 0xF4

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved						TEFL	EFF	Reserved			EFPI[4:0]				
						r	r				r				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			EFGI[4:0]					Reserved		EFFL[5:0]					
			r							r					

Bit Field	Name	Description
31:26	Reserved	Reserved, must be kept at reset value.
25	TEFL	Tx Event FIFO Element Lost When FDCAN_IR.TEFL is reset, this bit is also reset. 0: No transmit event FIFO element lost 1: Transmit event FIFO element lost, also set to 1 when attempting to write to a transmit event FIFO of size zero
24	EFF	Event FIFO Full 0: Transmit event FIFO not full 1: Transmit event FIFO full

Bit Field	Name	Description
23:21	Reserved	Reserved, must be kept at reset value.
20:16	EFPI[4:0]	Event FIFO Put Index Transmit event FIFO put index pointer,range from 0 to 31
15:13	Reserved	Reserved, must be kept at reset value.
12:8	EFGI[4:0]	Event FIFO Get Index Transmit event FIFO read index pointer, range from 0 to 31
7:6	Reserved	Reserved, must be kept at reset value.
5:0	EFFL[5:0]	Event FIFO Fill Level Number of elements stored in the transmit event FIFO, range from 0 to 31

25.5.47 FDCAN Transmit Event FIFO Acknowledge Register (FDCAN_TXEFA)

Address offset: 0xF8

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved											EFAI[4:0]				
											rw				

Bit Field	Name	Description
31:5	Reserved	Reserved, must be kept at reset value.
4:0	EFAI[4:0]	Event FIFO Acknowledge Index Once the host reads an element or a sequence of elements from the transmit event FIFO, it must write the index value of the last read element to EFAI. This action sets the transmit event FIFO get index FDCAN_TXEFS.EFGI to EFAI + 1 and updates the FIFO fill level FDCAN_TXEFS.EFFL.

25.5.48 FDCAN External Trigger Selection Register (FDCAN_TTSS)

Address offset: 0x144

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved							TS_EN	TS_SEL[2:0]			Reserved				
							rw		rw						

Bit Field	Name	Description
31:9	Reserved	Reserved, must be kept at reset value.
8	TS_EN	External timestamp vector enable 0: Disable external timestamp 1: Enable external timestamp
7:5	TS_SEL[2:0]	Select external timestamp's clock(PCLK1) divided 000: Timestamp clock at 4x frequency 001: Timestamp clock at 8x frequency 010: Timestamp clock at 16x frequency 011: Timestamp clock at 32x frequency 100: Timestamp clock at 64x frequency 101: Timestamp clock at 128x frequency 110: Timestamp clock at 256x frequency 111: Timestamp clock at 512x frequency
4:0	Reserved	Reserved, must be kept at reset value.

26 Serial Peripheral Interface/Inter-IC Sound (SPI/ I2S)

26.1 Introduction

The serial peripheral interface (SPI) allows half/ full-duplex, synchronous, serial communication with external devices. The interface can be configured as the master and in this case it provides the communication clock (SCK) to the external slave device. The interface is also capable of operating in multimaster configuration.

It may be used for a variety of purposes, including simplex synchronous transfers on two lines with a possible bidirectional data line or reliable communication using CRC checking.

The I2S is also a synchronous serial communication interface. It supports four different audio standards including the I2S Philips standard, the MSB- and LSB-justified standards, and the PCM standard. It can operate as a slave or a master device in full-duplex mode or in half-duplex mode. Master clock can be provided by the interface to an external slave component when the I2S is configured as the communication master.

26.2 Main Features

26.2.1 SPI Features

- Full-duplex and half-duplex synchronous mode.
- Support master mode, slave mode and multi-master mode.
- Supports 8-bit or 16-bit data frame format.
- Data bit sequence programmable.
- NSS management by hardware or software.
- Clock polarity and phase programmable.
- Transmitting and receiving support hardware CRC calculation and check.
- DMA capability for transmission and reception
- Supports FIFO mode, with a depth of 8 for both the transmit FIFO and receive FIFO.

26.2.2 I^{2}S Features

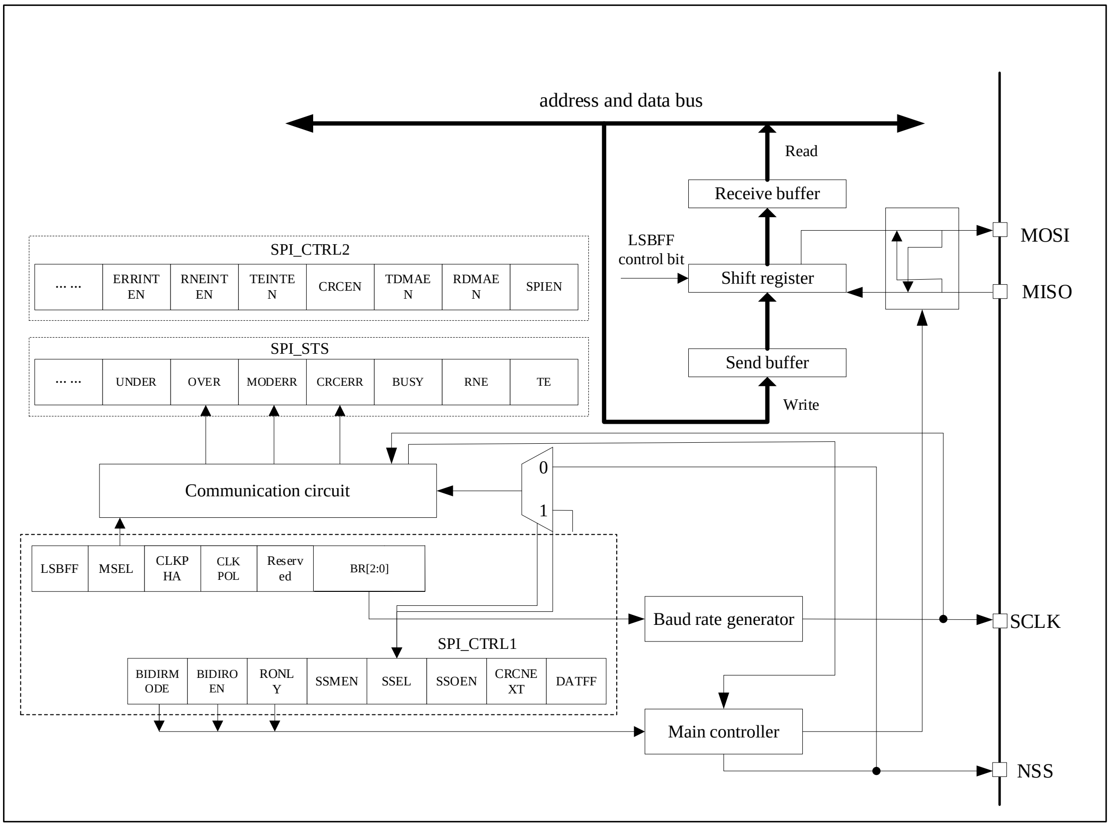
- Full-duplex mode and half-duplex synchronous mode.
- Supports master mode and slave mode operation.
- Four audio standards are supported: Philips I2S standard, MSB alignment standard, LSB alignment standard and PCM standard.
- The audio sampling frequency from 8kHz to 192kHz can be configured.
- Supports 16-bit, 24-bit or 32-bit data length and data frame format (configured according to requirements).
- Steady state clock polarity programmable.

- The data direction is always MSB first.
- DMA capability for transmission and reception
- Supports PLL, HSI, SYSCLK, and external clock.

26.3 SPI Function Description

26.3.1 General Description

Figure 26-1 SPI Block Diagram



Usually, the SPI is connected to external devices through four pins:

- SCLK: serial clock pin. Serial clock signal is output from the SCLK pin of master device and input to SCLK pin of slave device.
- MISO: master input/slave output pin. Data is received from the MISO pin of master device and send by the MISO pin of slave device.
- MOSI: master output/slave input pin. Data is transmitted by the MOSI pin of master device and received from the MOSI pin of slave device.
- NSS: Slave select. There are two types of NSS pin, internal pin and external pin. If the internal pin detects a high level, SPI works in the master mode. Conversely, SPI works in the slave mode. Users can use a standard I/O pin of the master device to control the NSS pin of the slave device.

Software NSS mode

The software slave device management is enabled when SPI_CTRL1.SSMEN = 1.

The NSS pin remains free in software NSS mode. In this mode the internal NSS signal level is driven by writing the SPI_CTRL1.SSEL bit (set SPI_CTRL1.SSEL=1 in master mode and set SPI_CTRL1.SSEL = 0 in slave mode).

Hardware NSS mode

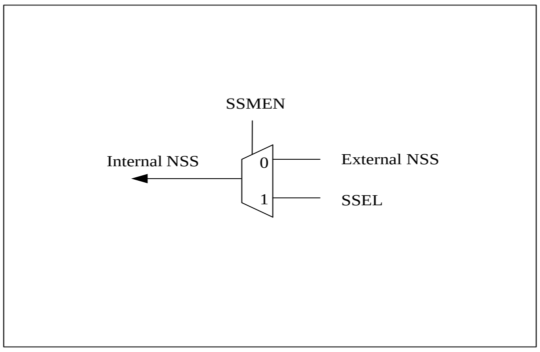
The software slave device management is disabled when SPI_CTRL1.SSMEN = 0.

NSS input mode: The NSS output of the master device is disabled (SPI_CTRL1.MSEL = 1, SPI_CTRL1.SSOEN = 0), allowing operation in multi-master mode. The master should connect NSS pin to the high level and the slave should connect NSS pin to the low level during the entire data frame transfer. When configured in master mode with NSS configured as an input (SPI_CTRL1.MSEL = 1, SPI_CTRL1.SSOEN = 0) and if NSS is pulled low, the SPI enters the master mode fault state.

NSS output mode: NSS output of the master device is enable (SPI_CTRL1.MSEL = 1, SPI_CTRL1.SSOEN = 1). SPI as the master device must pull the NSS pin to low level, all device which connected to the master device and set to NSS hardware mode, will detect low level and enter the slave mode automatically.

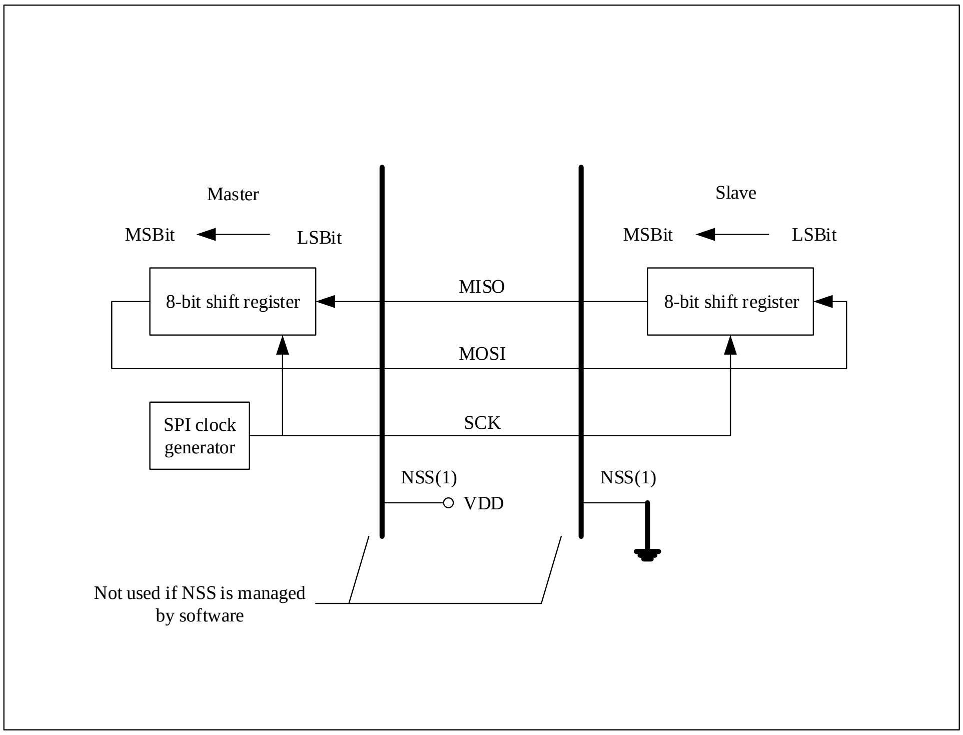
Note: The choice of software mode or hardware mode depends on whether NSS control is needed in the communication protocol. If not, you can choose the software mode, and release a GPIO pin for other purposes.

Figure 26-2 Selective Management of Hardware/Software



The following figure is an example of the interconnection of single master and single slave devices：

Figure 26-3 SPI Master and Slave Applications



Note: $^{(1)}$NSS pin is set as input

SPI is a ring bus structure. The master device outputs a synchronous clock signal through the SCK pin, the MOSI pin of the master device is connected to the MOSI pin of the slave device, and the MISO pin of the master device is connected to the MISO pin of the slave device, so that data can be transferred between devices. Continuous data transfer between master and slave, sending data to slave through MOSI pin and slave sending data to master through MISO pin.

SPI timing mode

User can select the clock edge of data capture by setting SPI_CTRL1.CLKPOL bit and SPI_CTRL1.CLKPHA bit.

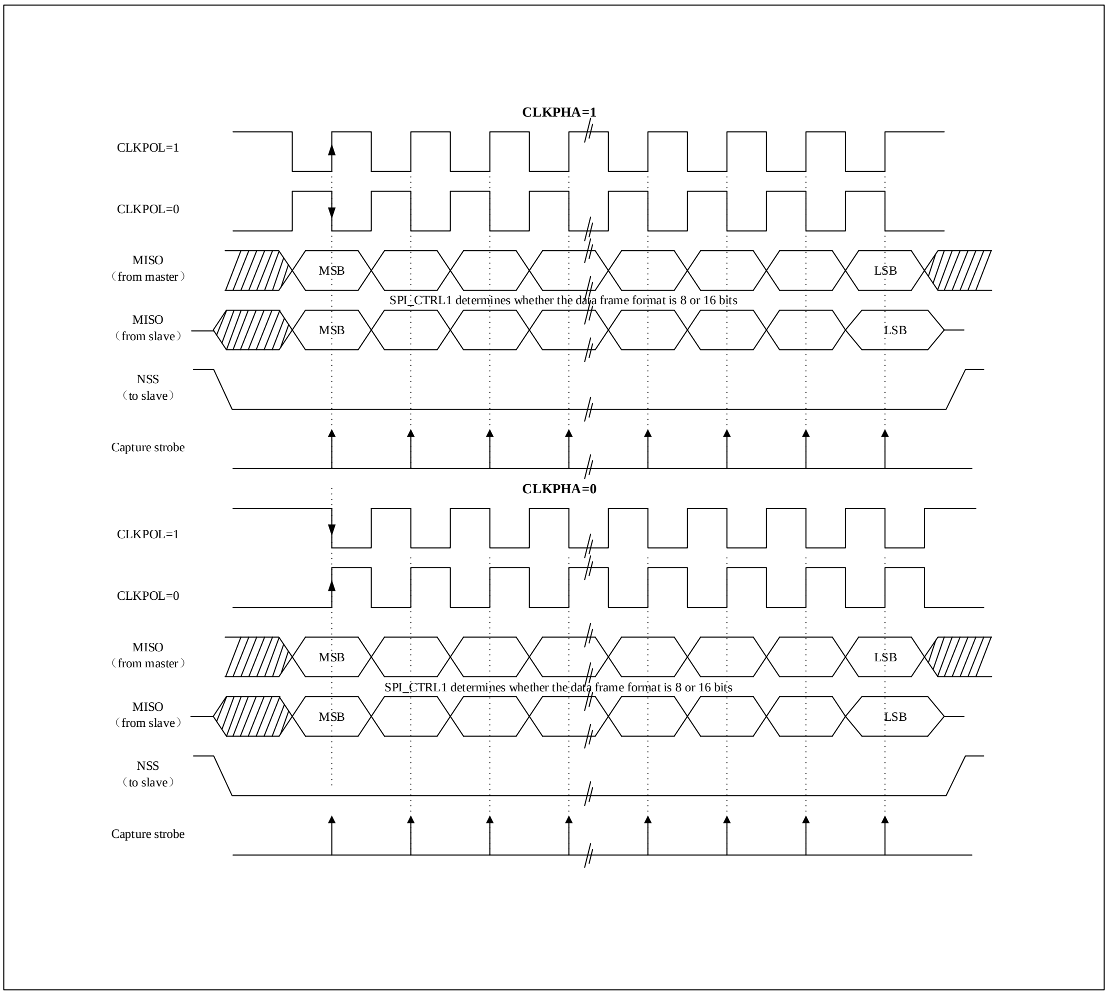
- When CLKPOL = 0, CLKPHA = 0, the SCLK pin will keep low in idle state, and the data will be sampled at the first edge, which is rising edge.
- When CLKPOL = 0, CLKPHA = 1, the SCLK pin will keep low in idle state, and the data will be sampled at the second edge, which is falling edge.
- When CLKPOL = 1, CLKPHA = 0, the SCLK pin will keep high in idle state, and the data will be sampled at the first edge, which is falling edge.
- When CLKPOL = 1, CLKPHA = 1, the SCLK pin will keep high in idle state, and the data will be sampled at the second edge, which is rising edge.

Regardless of the timing mode used, the master and slave configuration must be the same.

Figure 26-4 Data Clock Timing Diagram

is the combination timing of four CLKPHA and CLKPOL bits transmitted by SPI when the SPI_CTRL1.LSBFF = 0.

Figure 26-4 Data Clock Timing Diagram



Data format

User can selects the data order by setting the SPI_CTRL1.LSBFF bit. When SPI_CTRL1.LSBFF = 0, SPI will send the high-order data (MSB) first; When SPI_CTRL1.LSBFF = 1, SPI will send low-order data (LSB) first.

User can selects the data frame by setting the SPI_CTRL1.DATFF bit.

26.3.2 SPI Operating Mode

- Master full duplex mode (SPI_CTRL1.MSEL = 1, SPI_CTRL1.BIDIRMODE = 0, SPI_CTRL1.RONLY = 0)

After the first data is written to the SPI_DAT register, the transmission will start. When the first bit of the data is sent, the data bytes are loaded from the data register into the shift register in parallel, and then according to the configuration of the SPI_CTRL1.LSBFF bit, the data bits follow the MSB or LSB order are serially shifted to the MOSI pin. At the same time, the data receives on the MISO pin is serially shifted into the shift register in the same order and then loaded into the SPI_DAT register in parallel. The software operation process is as follows:

– Set SPI_CTRL2.SPIEN = 1, Enable SPI module.

– Write the first data to be sent into SPI_DAT register (this operation will clear SPI_STS.TE bit).

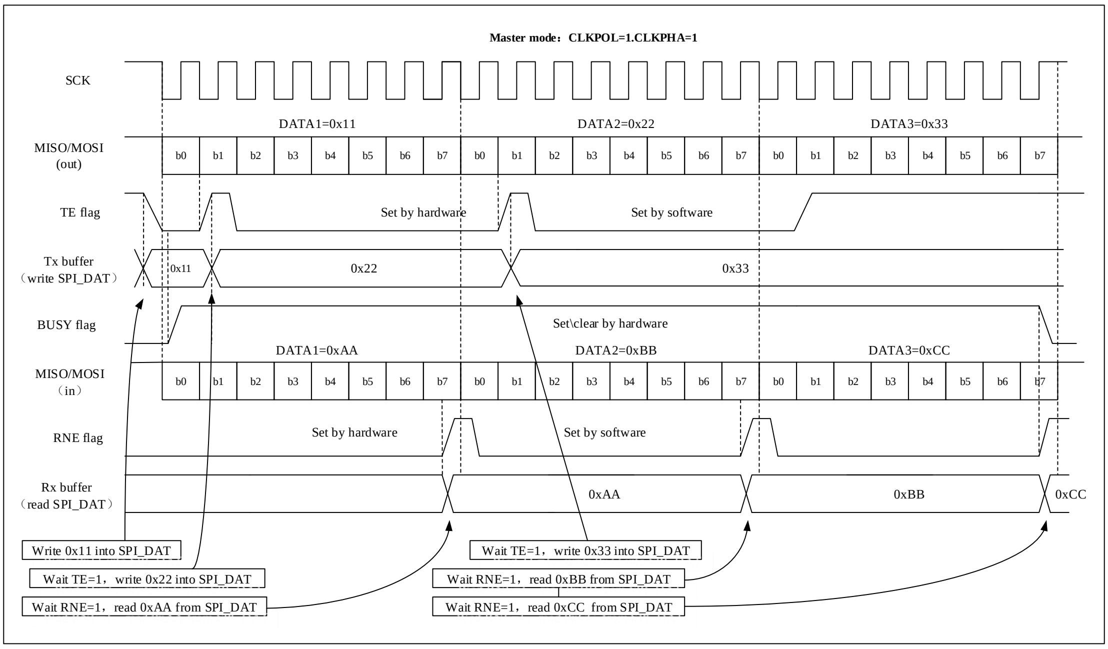
– Wait for SPI_STS.TE bit to be set to '1', and write the second data to be sent into SPI_DAT. Wait for SPI_STS.RNE bit to be set to '1', read SPI_DAT to get the first received data, and the SPI_STS.RNE bit will be cleared by hardware while reading SPI_DAT. Repeat the above operation, sending subsequent data and receiving n-1 data at the same time;

– Wait for SPI_STS.RNE bit to be set to '1' to receive the last data;

– Wait for SPI_STS.TE to be set to '1', then wait for SPI_STS.BUSY bit to be cleared and turn off SPI module.

The process of data transmitting and data receiving can also be implemented in the interrupt handler generated by the rising edge of the SPI_STS.RNE or SPI_STS.TE flag.

Figure 26-5 TE/RNE/BUSY Behavior in Master / Full-Duplex Mode(BIDIRMODE = 0, RONLY = 0) during Continuous Transmission



- Master two-wire unidirectional transmit-only mode (SPI_CTRL1.MSEL=1, SPI_CTRL1.BIDIRMODE = 0, SPI_CTRL1.RONLY = 0)

Master two-wire unidirectional transmit-only mode is similar to master full-duplex mode. The difference is that this mode will not read the received data, so the SPI_STS.OVER bit will be set to '1', and the software will ignore it. The software operation process is as follows:

- Set SPI_CTRL1.SPIEN = 1 to enable SPI module.

- Write the first data to be sent into SPI_DAT register (this operation will clear SPI_STS.TE bit).

- Wait for SPI_STS.TE bit to be set to '1', and write the second data to be sent into SPI_DAT. Repeat this operation to send subsequent data;

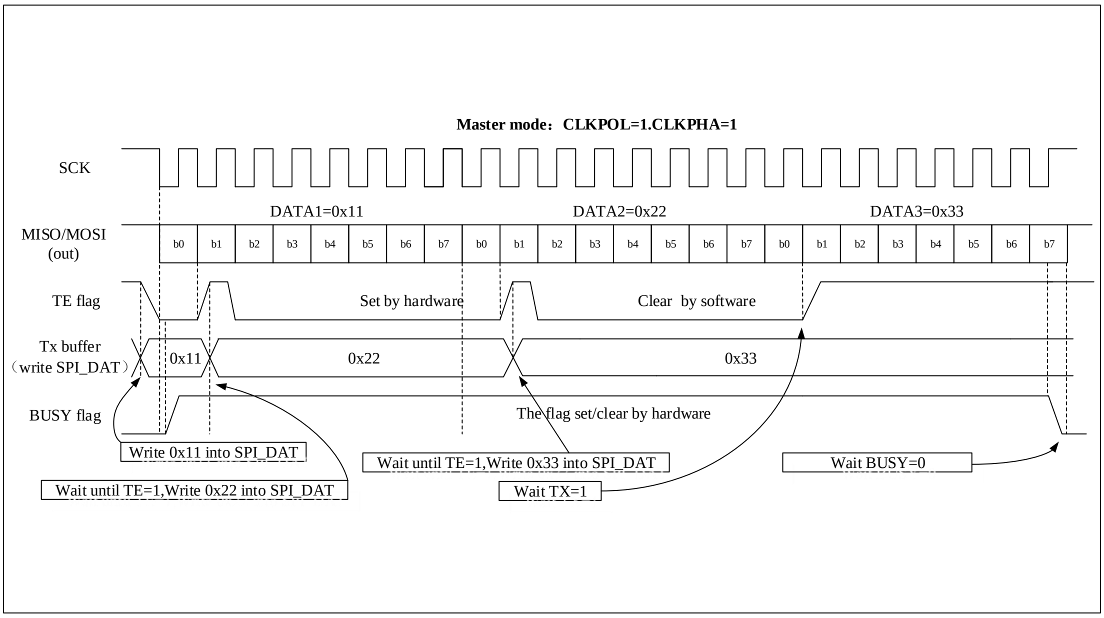
- After writing the last data to SPI_DAT, wait for SPI_STS.TE bit to set '1'; then wait for SPI_STS.BUSY bit to be cleared to complete the transmission of all data. This process can also be implemented in the interrupt handler triggered by the rising edge of the SPI_STS.TE flag.

Note:

1) *For discontinuous transfers, there is a delay of 2 APB clock cycles between writing to the SPI_DAT register and setting the SPI_STS.BUSY bit. Therefore, in transmit-only mode, after writing the last data, it is advisable to wait for SPI_STS.TE to be 1 first, and then wait for SPI_STS.BUSY to be 0.*

2) *In transmit-only mode, after transmitting 2 data, the OVER bit in the SPI_STS register will be set to '1' since no received data is read out. (Note: Software does not need to pay attention to this OVER flag)*

Figure 26-6 TXE/BSY Behavior in Master Transmit-Only Mode(BIDIRMODE = 0, RONLY = 0) during Continuous Transmissions

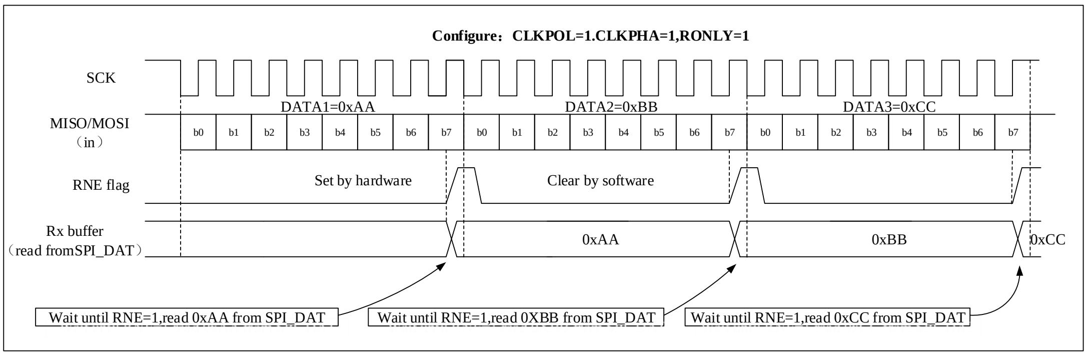

Master mode：CLKPOL=1.CLKPHA=1
SCK
DATA1=0x11
DATA2=0x22
DATA3=0x33
MISO/MOSI (out)
b0 b1 b2 b3 b4 b5 b6 b7 b0 b1 b2 b3 b4 b5 b6 b7 b0 b1 b2 b3 b4 b5 b6 b7
TE flag
Set by hardware
Clear by software
Tx buffer（write SPI_DAT）
0x11
0x22
0x33
BUSY flag
The flag set/clear by hardware
Write 0x11 into SPI_DAT
Wait until TE=1,Write 0x22 into SPI_DAT
Wait until TE=1,Write 0x33 into SPI_DAT
Wait TX=1
Wait BUSY=0

- Master two-wire unidirectional transmit-only mode (SPI_CTRL1.MSEL = 1, SPI_CTRL1.BIDIRMODE = 0, SPI_CTRL1.RONLY = 1)

When SPI_CTRL1.SPIEN = 1, the receiving process starts. The data bits from the MISO pin are sequentially shifted into the shift register and then loaded into the SPI_DAT register(receive buffer) in parallel. The software operation process is as follows:

- Set SPI_CTRL1.RONLY = 1 to enable the receive-only mode.
- In master mode, setting the SPI_CTRL2.SPIEN bit to 1 enables the SPI module, and the SCLK signal is generated immediately. Data is continuously received until the SPI is disabled (SPI_CTRL2.SPIEN = 0). In slave mode, when the master device drives the NSS signal low and generates SCLK, data is continuously received.
- Wait for SPI_STS.RNE bit to be set to '1', read the SPI_DAT register to get the received data, and the SPI_STS.RNE bit will be cleared by hardware while reading SPI_DAT register. Repeat this operation to receive all data.The process of data receiving can also be implemented in the interrupt handler generated by the rising edge of the RNE flag (SPI_STS.RNE).

Figure 26-7 RNE Behavior in Receive-Only Mode duringContinuous Transmissions (BIDIRMODE = 0 , RONLY = 1)


Configure：CLKPOL=1.CLKPHA=1,RONLY=1
SCK
MISO/MOSI
（in）
DATA1=0xAA
DATA2=0xBB
DATA3=0xCC
b0
b1
b2
b3
b4
b5
b6
b7
RNE flag
Set by hardware
Clear by software
Rx buffer
（read fromSPI_DAT）
0xAA
0xBB
0xCC
Wait until RNE=1,read 0xAA from SPI_DAT
Wait until RNE=1,read 0XBB from SPI_DAT
Wait until RNE=1,read 0xCC from SPI_DAT

- Master one-wire bidirectional transmit mode (SPI_CTRL1.MSEL = 1, SPI_CTRL1.BIDIRMODE = 1, SPI_CTRL1.BIDIROEN = 1, SPI_CTRL1.RONLY = 0)

After the data is written to the SPI_DAT register (send buffer), the transmission process starts. This mode does not receive data. At the same time as the first data bit is sent, the data to be sent is loaded into the shift register in parallel, and then according to the configuration of the SPI_CTRL1.LSBFF bit, the SPI serially shifts the data bits to the MOSI pin in MSB or LSB order.

The software operation flow of the master one-wire bidirectional transmit mode is the same as that of the transmit-only mode. The difference is that before enabling the SPI module, both the SPI_CTRL1.BIDIMODE and SPI_CTRL1.BIDIROEN bits need to be set to '1' in the SPI_CTRL1 register.

- **Master one-wire bidirectional receive mode (SPI_CTRL1.MSEL = 1, SPI_CTRL1.BIDIRMODE = 1, SPI_CTRL1.BIDIROEN = 0, SPI_CTRL1.RONLY = 0)**

When SPI_CTRL1.SPIEN = 1 , the receiving process starts. There is no data output in this mode, the received data bits are sequentially and serially shifted into the shift register, and then loaded into the SPI_DAT register (receive buffer) in parallel

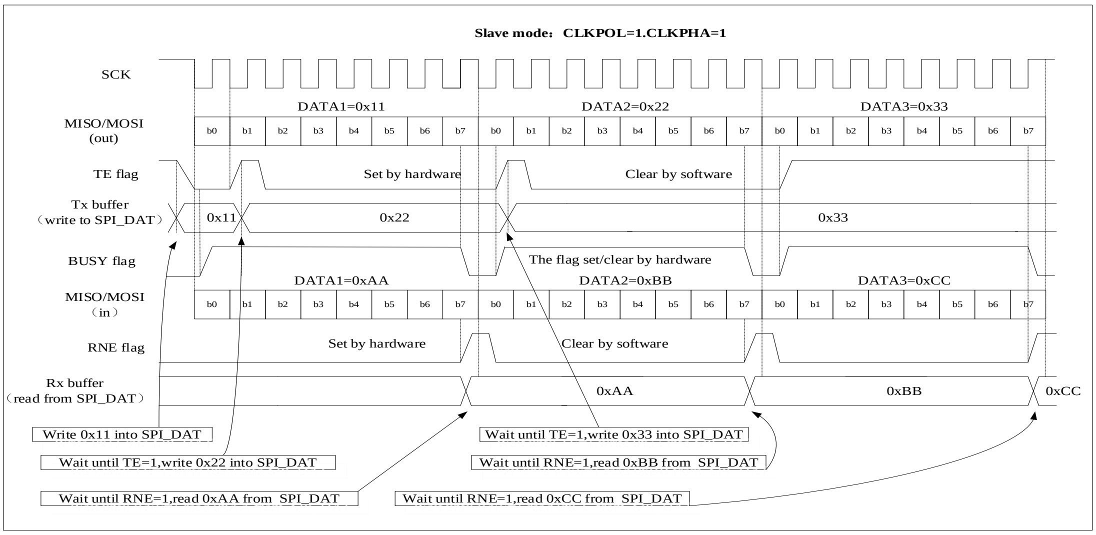
The software operation flow of the master one-wire bidirectional receive mode is the same as that of the receive-only mode. The difference is that before enabling the SPI module, the SPI_CTRL1.BIDIMODE bit needs to be set to '1' and the SPI_CTRL1.BIDIROEN bit needs to be set to '0' in the SPI_CTRL1 register.

- **Slave full duplex mode (SPI_CTRL1.MSEL = 0, SPI_CTRL1.BIDIRMODE = 0, SPI_CTRL1.RONLY = 0)**

The data transfer process begins when the slave device receives the first clock edge.The data serially moves into the shift register.

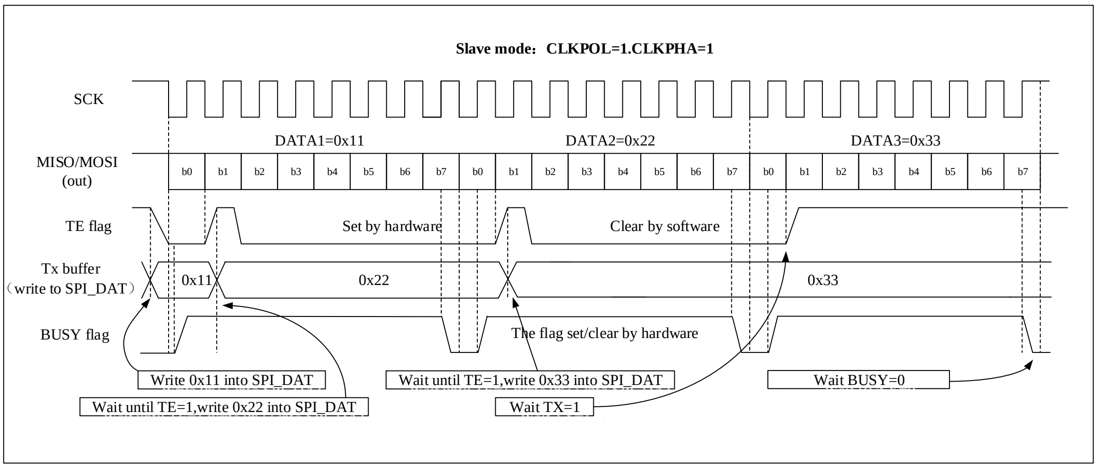
At the same time, while transmitting the first data bit, the data in the transmit buffer is parallel transferred to the shift register, and then serially sent out to the MISO pin. Software must ensure that the data to be transmitted is written into the slave transmit buffer before the SPI master device starts data transfer.

Figure 26-8 TE/RNE/BUSY Behavior in Slave / Full-Duplex Mode during Continuous Transmissions


Slave mode：CLKPOL=1.CLKPHA=1
SCK
MISO/MOSI (out)
DATA1=0x11
DATA2=0x22
DATA3=0x33
b0 b1 b2 b3 b4 b5 b6 b7
TE flag
Set by hardware
Clear by software
Tx buffer（write to SPI_DAT）
0x11
0x22
0x33
BUSY flag
The flag set/clear by hardware
MISO/MOSI（in）
DATA1=0xAA
DATA2=0xBB
DATA3=0xCC
RNE flag
Set by hardware
Clear by software
Rx buffer（read from SPI_DAT）
0xAA
0xBB
0xCC
Write 0x11 into SPI_DAT
Wait until TE=1,write 0x22 into SPI_DAT
Wait until RNE=1,read 0xAA from SPI_DAT
Wait until TE=1,write 0x33 into SPI_DAT
Wait until RNE=1,read 0xBB from SPI_DAT
Wait until RNE=1,read 0xCC from SPI_DAT

- **Slave two-wire unidirectional transmit-only mode (SPI_CTRL1.MSEL = 0, SPI_CTRL1.BIDIRMODE = 0 and SPI_CTRL1.RONLY = 0)**

Figure 26-9 TE/BUSY Behavior in Slave Transmit-Only Mode in Case of Continuous Transfers



- **Slave two-wire unidirectional receive-only mode (SPI_CTRL1.MSEL = 0, SPI_CTRL1.BIDIRMODE = 0 and SPI_CTRL1.RONLY = 1)**

The data receiving process begins when the slave device receives the clock signal and the first data bit from the MOSI pin. The received data bits are sequentially and consecutively shifted serially into an shift register and then loaded into the SPI_DAT register (receive buffer) in parallel.

- **Slave one-wire bidirectional transmit mode (SPI_CTRL1.MSEL = 0, SPI_CTRL1.BIDIRMODE = 1 and SPI_CTRL1.BIDIROEN = 1)**

When the slave device receives the first edge of the clock signal, the transmitting process starts. No data is received in this mode. Simultaneously with the transmission of the first data bit to the MISO pin, the data to be transmitted from the transmit buffer is parallel transferred to the shift register and then serially sent out to the MISO pin. Software must ensure that the data to be transmitted is written into the slave transmit register before the SPI master device starts data transfer.

- **Slave one-wire bidirectional receive mode (SPI_CTRL1.MSEL = 0, SPI_CTRL1.BIDIRMODE = 1 and SPI_CTRL1.BIDIROEN = 0)**

Data receiving begins when the slave device receives the first clock edge and a data bit from the MOSI pin. There is no data output in this mode, the received data bits are sequentially and consecutively shifted serially into an shift register, and then loaded into the SPI_DAT register (receive buffer) in parallel.

Note: The software operation process of the slave can refer to the master.

SPI initialization process

1) The baud rate of serial clock is defined by the SPI_CTRL1.BR[2:0] bits (this step is ignored if it is working in slave mode).
2) Select SPI_CTRL1.CLKPOL bit and SPI_CTRL1.CLKPHA bit to define the phase relationship between data transmission and serial clock.
3) Set SPI_CTRL1.DATFF bit to define 8-bit or 16-bit data frame format.
4) Configure the SPI_CTRL1.LSBFF bit to define the frame format.
5) Configure the NSS mode as described above for the NSS function.
6) Run mode is configured by SPI_CTRL1.MSEL bit, SPI_CTRL1.BIDIRMODE bit, SPI_CTRL1.BIDIROEN bit and SPI_CTRL1.RONLY bit.
7) Set the SPI_CTRL1.SPIEN = 1 to enable SPI.

Basic SPI send and receive processing

When SPI sends a data frame, it firstly loads the data frame from the data buffer into the shift register, and then starts to send the loaded data. When the data is transferred from the transmit buffer to the shift register, the transmit buffer empty flag is set (SPI_STS.TE = 1), and the next data can be loaded into the send buffer; if the TEINTEN bit is set (SPI_CTRL2.TEINTEN = 1), an interrupt will be generated; writing data to the SPI_DAT register will clear the SPI_STS.TE bit.

At the last edge of the sampling clock, when the data is transferred from the shift register to the receive buffer, the receive buffer non-empty flag is set (SPI_STS.RNE = 1), at this time the data is ready and can be read from the SPI_DAT register; if the receive buffer non-empty interrupt is enabled (SPI_CTRL2.RNEINTEN = 1), an interrupt will be generated; the SPI_STS.RNE bit can be cleared by reading the SPI_DAT register data.

In master mode, the transmitting process starts when data is written to the send buffer. If the next data has been written into the SPI_DAT register before the current data frame sending is completed, the continuous sending function can be achieved.

In slave mode, the NSS pin is low, and when the first clock edge arrives, the transmission process begins. In order to avoid accidental data transmission, software must write data to the transmit buffer before data transmission (it is recommended to enable the SPI module before the master sends the clock).

In some configurations, when the last data is sent, the BUSY flag (SPI_STS.BUSY) can be used to wait for the end of the data sending.

Continuous and discontinuous transmission.

When sending data in master mode, if the software is fast enough to detect each TE (SPI_STS.TE) rising edge (or TE interrupt), and the data is written to the SPI_DAT register immediately before the end of the ongoing transmission. At this time, the SPI clock remains continuous between the transmission of data items, and the SPI_STS.BUSY bit will not be cleared, continuous communication can be achieved.

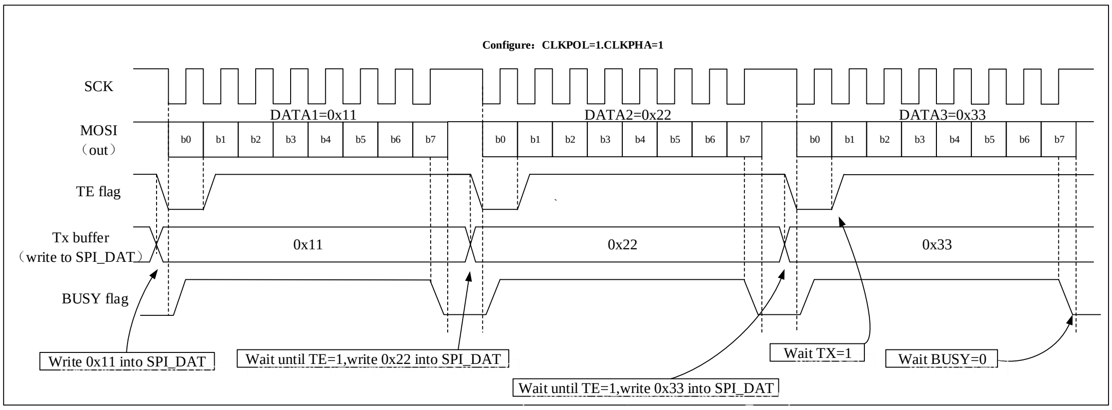
If the software is not fast enough, it will result in discontinuous communication; in this case, the SPI_STS.BUSY bit is cleared between the transmission of each data items (refer to Figure 26-10 below).

In master receive-only mode (SPI_CTRL1.RONLY = 1), communication is always continuous and the BUSY flag (SPI_STS.BUSY) is always high.

In slave mode, the continuity of communication is determined by the SPI master device. However, even if the communication is continuous, the BUSY flag (SPI_STS.BUSY) will be low for at least one SPI clock cycle between each data item.

Note: If you want the NSS to toggle with each frame of data, i.e., pulled low during frame transmission and pulled high after frame transmission, the user can enable the SPI before frame transmission and disable it after transmission is completed.

Figure 26-10 TE/BUSY Behavior in Non-Continuous Transmission (BIDIRMODE = 0 and RONLY = 0).


Configure：CLKPOL=1.CLKPHA=1
SCK
MOSI （out）
DATA1=0x11
DATA2=0x22
DATA3=0x33
b0
b1
b2
b3
b4
b5
b6
b7
TE flag
Tx buffer （write to SPI_DAT）
0x11
0x22
0x33
BUSY flag
Write 0x11 into SPI_DAT
Wait until TE=1,write 0x22 into SPI_DAT
Wait until TE=1,write 0x33 into SPI_DAT
Wait TX=1
Wait BUSY=0

26.3.3 Status Flag

The SPI_STS register has 3 flag bits to monitor the status of the SPI:

Send buffer empty flag bit (TE)

When the send buffer is empty, the TE flag (SPI_STS.TE) is set to 1, which means that new data can be written into the SPI_DAT register. When the send buffer is not empty, the hardware will clear this flag to 0.

Receive buffer non-empty flag bit (RNE)

When the receive buffer is not empty, the RNE flag (SPI_STS.RNE) is set to 1, so the user knows that there is data in the receive buffer. After reading the SPI_DAT register, the hardware will set this flag to 0.

BUSY flag bit (BUSY)

When the transmission starts, the hardware sets the BUSY flag (SPI_STS.BUSY) to 1, and after the transmission ends, the hardware sets the BUSY flag to 0.

Only when the device is in the master one-wire bidirectional receive mode, the BUSY flag (SPI_STS.BUSY) will be set to 0 when the communication is in progress.

The BUSY flag (SPI_STS.BUSY) will be cleared to 0 in the following cases:

- End of transmission (except for continuous communication in master mode);
- Disable the SPI module (SPI_CTRL1.SPIEN = 0);
- The master mode error occurs (SPI_STS.MODERR = 1)

When the communication is discontinuous: the BUSY flag (SPI_STS.BUSY) is cleared to '0' between the transmission of each data item.

When communication is continuous: in master mode, the BUSY flag (SPI_STS.BUSY) remains high during the entire transfer process; In slave mode, the BUSY flag (SPI_STS.BUSY) will be low for 1 SPI clock cycle between each data item transfer. So do not use the BUSY flag to handle the sending and receiving of each data item.

26.3.4 Disabling SPI

Clearing the SPI_CTRL2.SPIEN bit will disable the SPI.

In certain configurations, disabling the SPI module before the transmission is completed may result in the current transmission being corrupted, and the SPI_STS.BUSY flag becoming unreliable.

To avoid this situation, it is recommended to follow the steps below when disabling the SPI module:

Master or slave full duplex mode(SPI_CTRL1.BIDIMODE=0，SPI_CTRL1.RONLY=0)

1) Wait for the RNE flag (SPI_STS.RNE) to be set to 1 and the last byte to be received;
2) Wait for the TE flag (SPI_STS.TE) to be set to 1;
3) Wait for the BUSY flag (SPI_STS.BUSY) to be cleared to 0;
4) Turn off the SPI module (SPI_CTRL1.SPIEN = 0).

Two-wire unidirectional transmit-only mode(SPI_CTRL1.BIDIMODE=0，SPI_CTRL1.RONLY=0) or one-wire bidirectional transmit mode(SPI_CTRL1.BIDIMODE=1，SPI_CTRL1.BIDIROEN=1) for master or slave

1) After writing the last byte to the SPI_DAT register, wait for the TE flag (SPI_STS.TE) to be set to 1;
2) Wait for the BUSY flag (SPI_STS.BUSY) to be cleared to 0;
3) Disable the SPI module (SPI_CTRL1.SPIEN = 0).

Note: In master mode with unidirectional transmit-only mode, the SPI_STS.BUSY flag remains low during the transmission process.

Two-wire unidirectional receive-only mode(SPI_CTRL1.MSEL=1，SPI_CTRL1.BIDIMODE=0，SPI_CTRL1.RONLY=1) or one-wire bidirectional receive mode(SPI_CTRL1.MSEL=1，SPI_CTRL1.BIDIMODE=1，SPI_CTRL1.BIDIROEN=0) for master

This scenario needs to be handled specifically to ensure that the SPI does not initiate a new transmission:

1) Wait for the penultimate RNE (SPI_STS.RNE) to be set to 1;

2) Before closing the SPI module (SPI_CTRL1.SPIEN = 0), wait for 1 SPI clock cycle (using software delay);

3) Wait for the last RNE (SPI_STS.RNE) to be set before entering shutdown mode (or disabling the SPI module clock).

Two-wire unidirectional receive-only mode(SPI_CTRL1.MSEL=0 ， SPI_CTRL1.BIDIMODE=0 ， SPI_CTRL1.RONLY=1) or one-wire bidirectional receive mode(SPI_CTRL1.MSEL=0 ， SPI_CTRL1.BIDIMODE=1，SPI_CTRL1.BIDIROEN=0) for slave

1) The SPI module can be disabled at any time (SPI_CTRL1.SPIEN = 0), and after the current transfer is over, the SPI module will be disabled;

2) If you want to enter the shutdown mode, you must wait for the BUSY flag (SPI_STS.BUSY) to be set to 0 before entering the shutdown mode (or disable the SPI module clock).

26.3.5 SPI Communication Using DMA

To achieve maximum communication speed, it is necessary to promptly fill the SPI transmit buffer with data and also read data from the receive buffer in a timely manner to prevent overflow. To facilitate high-speed data transfer, SPI implements a simple request/response DMA mechanism.

When the corresponding enable bit on the SPI_CTRL2 register is set, the SPI module can issue DMA transfer requests. The transmit buffer and receive buffer also have their own DMA requests.

- When the send buffer DMA is enabled (SPI_CTRL2.TDMAEN = 1), each time the TE flag (SPI_STS.TE) bit is 1, a DMA request will be generated, and the DMA will automatically write the data to the SPI_DAT register, which will clear the TE flag (SPI_STS.TE) bit.
- When the receive buffer DMA is enabled (SPI_CTRL2.RDMAEN = 1), each time the RNE flag (SPI_STS.RNE) bit is set to 1, a DMA request will be generated, and the DMA will automatically read the SPI_DAT register, which will clear the RNE flag (SPI_STS.RNE) bit.

When the SPI is only used for transmitting data, only the transmit DMA channel of the SPI needs to be enabled (SPI_CTRL2.TDMAEN = 1). At this time, since the received data has not been read, the OVER flag is set to '1' (software does not need to pay attention to this flag).

When the SPI is only used for receiving data, only the receive DMA channel of the SPI needs to be enabled (SPI_CTRL2.RDMAEN = 1).

In transmit mode, after DMA has sent all the data to be sent (DMA_INTSTS.TXCF = 1), BUSY flag (SPI_STS.BUSY) can monitor to confirm whether SPI communication is over, which can avoid destroying the transmission of the last data when the SPI is turned off or enters the shutdown mode. Therefore, the software needs to wait for the TE flag (SPI_STS.TE) bit to be set to 1, and wait for the BUSY flag (SPI_STS.BUSY) bit to be set to 0.

Note: In discontinuous transfers, there is a delay of 2 APB clock cycles between writing data to the SPI_DAT register and the SPI_STS.BUSY bit being set to '1'. Therefore, after writing the last data, it is necessary to wait for TE=1 first and then wait for SPI_STS.BUSY=0.

Figure 26-11 Transmition Using DMA

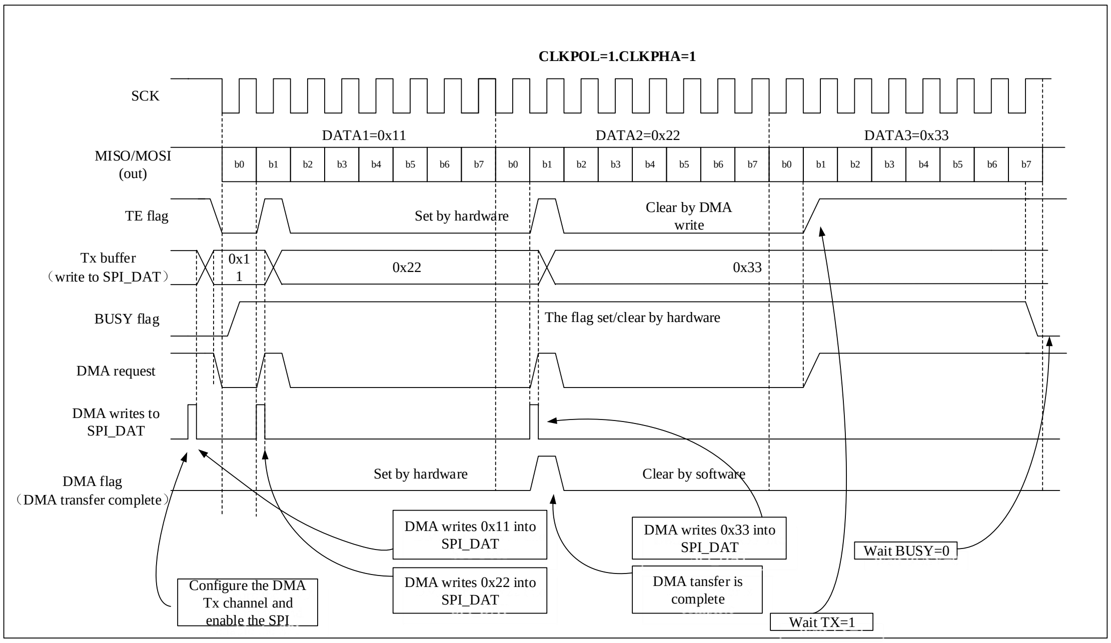
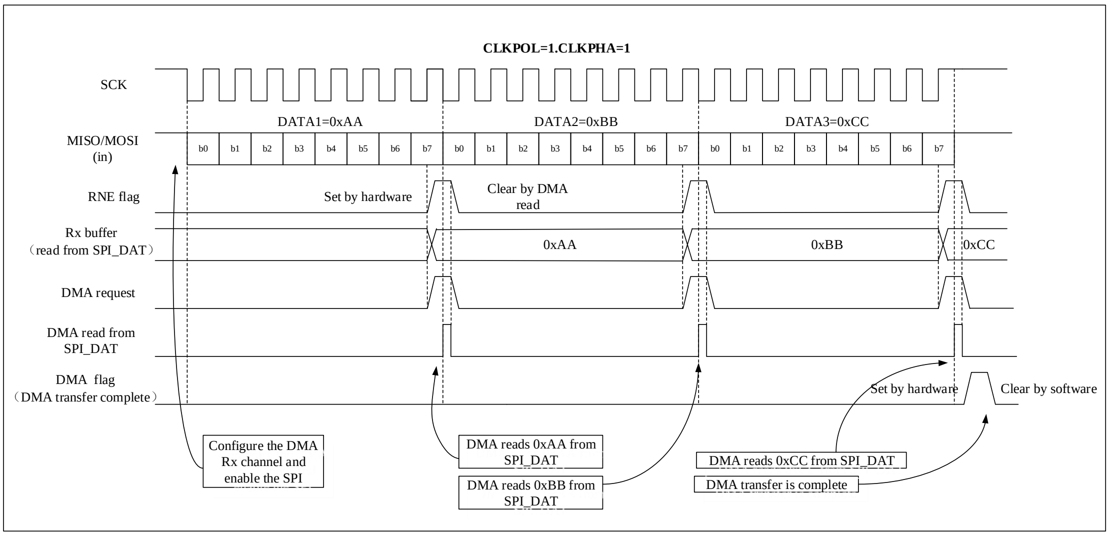


Figure 26-12 Reception Using DMA



26.3.6 CRC Calculation

SPI can enable CRC calculation through follow steps:

- Write the polynomial into the SPI_CRCPOLY register.
- Enable CRC calculation by setting the SPI_CTRL2.CRCEN bit, which will also clear the SPI_CRCRDAT and SPI_CRCTDAT registers.
- Set the SPI_CTRL2.SPIEN bit to enable the SPI module.
- Start and continue transmitting data until only the last byte or half-word remains in the current communication.
- After the last data is written into the transmit buffer, set the SPI_CTRL1.CRCNEXT = 1, which indicates that

the hardware will start transmitting the CRC value (SPI_CRCTDAT value) after transmitting the data. When the CRC is sent, the CRC calculation will stop.

- When the last byte or half-word is sent, the SPI sends the CRC value, and the SPI_CTRL1.CRCNEXT bit is cleared. Similarly, the received CRC is compared with the SPI_CRCRDAT value. If they do not match, the SPI_STS.CRCERR flag is set. If SPI_CTRL2.ERRINTEN is set, an interrupt is generated.

Note: When the SPI module is in slave mode, please ensure that CRC calculation is enabled only after the clock is stable to avoid incorrect CRC calculation results. As long as the SPI_CTRL2.CRCEN bit is set and there is an input clock on the SCK pin, CRC calculation will be performed regardless of the state of the SPI_CTRL2.SPIEN bit.

When the SPI clock frequency is high, users must be cautious when sending CRC. During CRC transmission, the CPU usage should be minimized as much as possible. To avoid errors when receiving the final data and CRC, function calls should be disabled during the CRC sending process. Setting the CRCNEXT bit must be completed before sending/receiving the last piece of data.

When the SPI clock frequency is high, as CPU operations can impact the SPI bandwidth, it is recommended to use DMA mode to avoid slowing down the SPI frequency.

When the SPI is configured as a slave and NSS hardware mode is used, the NSS pin should remain low during both data and CRC transmission. When the SPI is configured as a slave and using CRC functionality, the CRC calculation will still be performed even if the NSS pin is high (CRC calculation will continue if there are clock pulses on the SCK pin while the NSS signal is high. For example, this can occur when the master device alternates communication between multiple slave devices. Setting the SPI_CTRL2[13] bit can prevent incorrect CRC operation in this scenario).

In order to keep the synchronization of the next CRC calculation results between the master and slave devices when transitioning from deselecting one slave device (NSS signal high) to selecting a slave device (NSS signal low), the CRC values at both ends (master and slave) should be cleared.

Clear the CRC values according to the following steps:

1) Disable the SPI module (SPI_CTRL2.SPIEN=0)

2) Clear the SPI_CTRL2.CRCEN bit to '0'

3) Set the SPI_CTRL2.CRCEN bit to '1'

4) Enable the SPI module (SPI_CTRL2.SPIEN=1)

Transmit the CRC

In non-FIFO mode:

- In CPU mode, the user needs to set the SPI_CTRL1.CRCNEXT bit to 1 before transmitting the last frame of data.
- In DMA mode, the hardware automatically transmits the CRC check value.

In FIFO mode:

- In CPU mode, if it is receive-only mode, the CRC will automatically be transmitted based on the value configured in the SPI_TRANS_NUM register by the user.

- In CPU mode, except for receive-only mode, the user needs to configure the SPI_CTRL1.CRCNEXT bit to 1 before transmitting the last data to transmit the CRC value.
- In DMA mode, the hardware automatically transmits the CRC check value.

Note: In practical applications, if software-calculated CRC needs to match the hardware-calculated result, the following settings need to be correctly configured:

CRC8/CRC16

- width:8/16
- Polynomial (POLY): Configurable through the SPI_CRCPOLY register.
- Initial value (INIT): 0x0000
- Input data reverse (REFIN): No
- Output data reverse (REFOUT): No

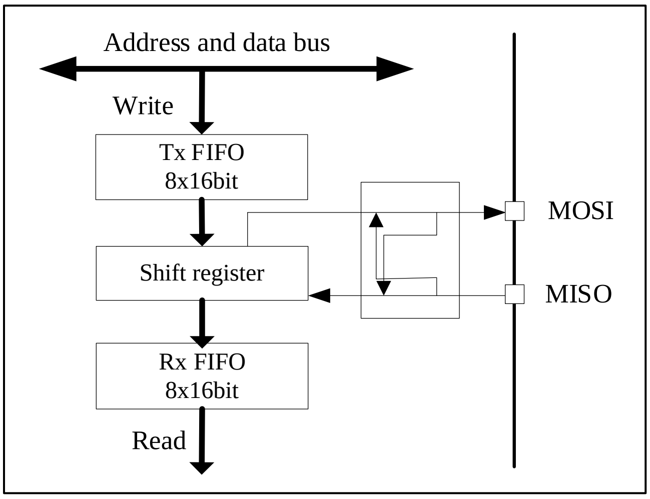
26.3.7 FIFO

Support FIFO function.

When the data frame format is configured as 8-bit, the TX_FIFO size is 8 x 8 bits, and the RX_FIFO size is 8 x 8 bits.

When the data frame format is configured as 16-bit, the TX_FIFO size is 8 x 16 bits, and the RX_FIFO size is 8 x 16 bits.

Figure 26-26-13 Reception Using DMA



When setting SPI_CTRL2.FIFOEN to ‘1’, the SPI will enable the FIFO function.

Transmit: When the CPU writes data to the SPI_DAT register, the transmitted data will be written to the TX_FIFO, and serially output to the interface through TX_SHIFT.

Receive: Receive data serially from the interface and transfer it to the RX_FIFO through RX_SHIFT.

Note: In FIFO mode, data is read by accessing the SPI_RX_FIFO register, while in non-FIFO mode, data is read by accessing the SPI_DAT register. Both FIFO and non-FIFO modes involve writing data by accessing the SPI_DAT *register.*

To enhance data control flexibility, flags for TX_FIFO half-empty, TX_FIFO full, RX_FIFO half-full, and RX_FIFO full have been added. Users can access the SPI_FIFO_CNT register to view the effective number of data in the FIFO for better compatibility with various scenarios.

26.3.8 Error Flag

Master mode fault (MODERR)

The following two conditions will cause the master mode fault:

- In NSS pin hardware management mode, the master device NSS pin is pulled low;
- In NSS pin software management mode, the SPI_CTRL1.SSEL bit is set to 0.

The master mode fault has the following impacts on the SPI device:

- When a master mode failure error occurs, the SPI_STS.MODERR bit is set to 1. An interrupt is generated if the user enables the corresponding interrupt (SPI_CTRL2.ERRINTEN = 1).
- The SPI_CTRL1.SPIEN bit and SPI_CTRL1.MSEL bit will be write protected and both are cleared by hardware. SPI is disabled and forced into slave mode
- The SPI_CTRL1.MSEL bit is cleared to '0', forcing the device into slave mode.

The following steps are used to clear the SPI_STS.MODERR bit:

1) Perform a read or write operation to the SPI_STS register
2) Writes to the SPI_CTRL1 register.

In systems with multiple MCUs, to avoid conflicts with multiple slave devices, the NSS pin of the master device must be pulled high before clearing the SPI_STS.MODERR bit. After clearing it, the SPI_CTRL2.SPIEN and SPI_CTRL1.MSEL bits can be restored to their reset state.

Forsecurity reasons, when the SPI_STS.MODERR bit is '1', the hardware does not allow the setting of the SPI_CTRL2.SPIEN and SPI_CTRL1.MSEL bits.

Normally, the SPI_STS.MODERR bit of the slave cannot be set to 1. However, in a multi-master configuration, the slave's SPI_STS.MODERR bit may be set to 1. In this case, the SPI_STS.MODERR bit indicates that there is a multi-master collision. The interrupt routine can perform a reset or return to the default state to recover from an error state.

Overflow Error (OVER)

In SPI non-FIFO mode, When the SPI_STS.RNE bit is set to 1, but there is still data sent into the receive buffer, an overflow error will occur. In SPI FIFO mode, if data is written to the TX FIFO when it is already full, this flag will be set to 1.

When an overflow error occurs:

- The OVER bit is set to "1", when the SPI_CTRL2.ERRINTEN bit is set to 1, an interrupt is generated.

For SPI1/SPI4/SPI5/SPI6, when OVER is 1, reading the SPI_DAT register (twice) and then reading the SPI_STS register will clear OVER. For SPI2_I2S2/SPI3_I2S3, when OVER is 1, reading the SPI_DAT register (once) and then the SPI_STS register will clear OVER.

Underflow Error (UNDER)

In SPI non-FIFO mode, this flag is not used; in SPI FIFO mode, attempting to read the RX FIFO when it is empty will cause this flag to be set to 1. Reading the SPI_STS status register in software will clear this bit.

CRC Error (CRCERR)

The CRC error flag is used to check the validity of the received data. A CRC error occurs when the received CRC value does not match the SPI_CRCRDAT value. At this time, the SPI_STS.CRCERR flag bit is set to '1', and an interrupt will be generated if the user enables the corresponding interrupt (SPI_CTRL2.ERRINTEN = 1).

26.3.9 SPI Interrupt

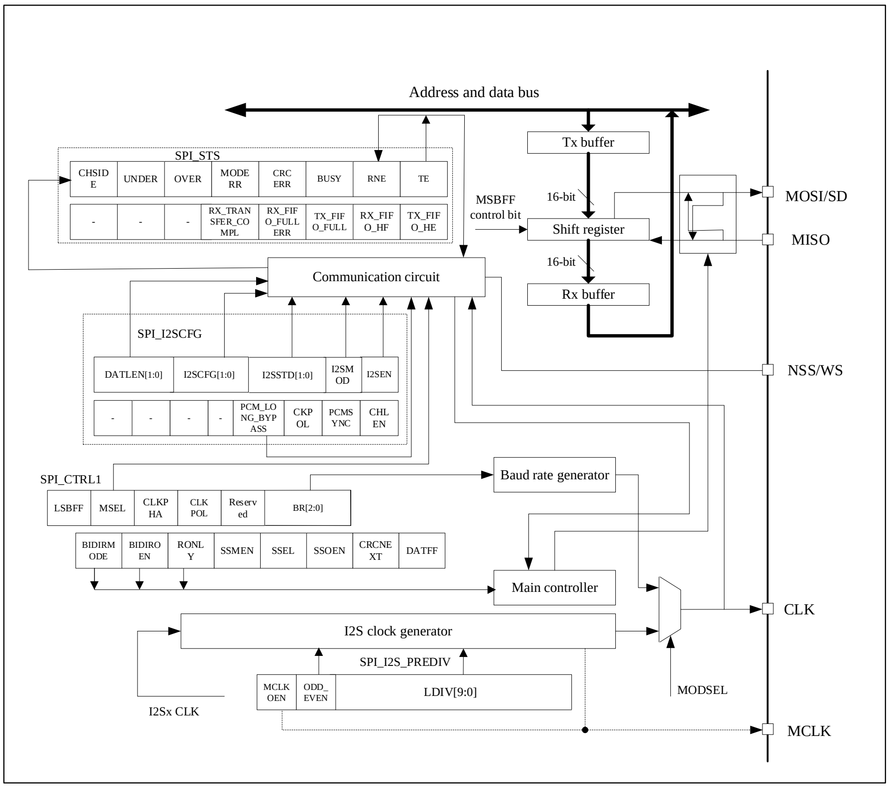
Table 26-26-1 SPI Interrupt Request

Interrupt event	Event flag	Enable Control bit
transmit buffer empty flag	TE	TEINTEN
receive buffer not empty flag	RNE	RNEINTEN
master mode fault event	MODERR	ERRINTEN
Overflow error	OVER	
CRC error	CRCERR	
transmit FIFO half-empty flag	TXFIFHE	TXFHEINTEN
receive FIFO half-full flag	RXFIFHF	RXFHFINTEN
receive FIFO full flag	RXFIFFU	RXFFUINTEN
transfer completion flag (receive-only mode)	RXTSCP	RXCPINTEN

26.4 I2S Function Description

The block diagram of I2S is shown in the figure below:

Figure 26-26-14 I2S Block Diagram



The I2S interface uses the same pins, flags and interrupts as the SPI interface. Setting the SPI_I2S_CFGR. I2SMOD= 1 selects the I2S audio interface.

I2S has a total of 4 pins, 3 of which are shared with SPI:

- SD: serial data (shared with MOSI pin), used for data send and receive (independently mapped).
- SD_1: used only in I2S full-duplex mode for data transmission and reception.
- SCK:serial clock (mapped to the CLK pin), acts as a clock signal output in master mode and as an input in slave mode. In some external audio devices that require a master clock, an additional pin can be used to output the clock.
- WS: Word Select (mapped to the NSS pin), used as a data control signal output in master mode and as an input in slave mode.
- MCK: Master Clock (independently mapped), when I2S is configured as master mode, it is used as an additional clock signal output pin when the MCLKOEN bit of the SPI_I2S_PREDIV register is set to '1'. The frequency of the output clock signal is preset to 256 × Fs, where Fs is the sampling frequency of the audio signal.

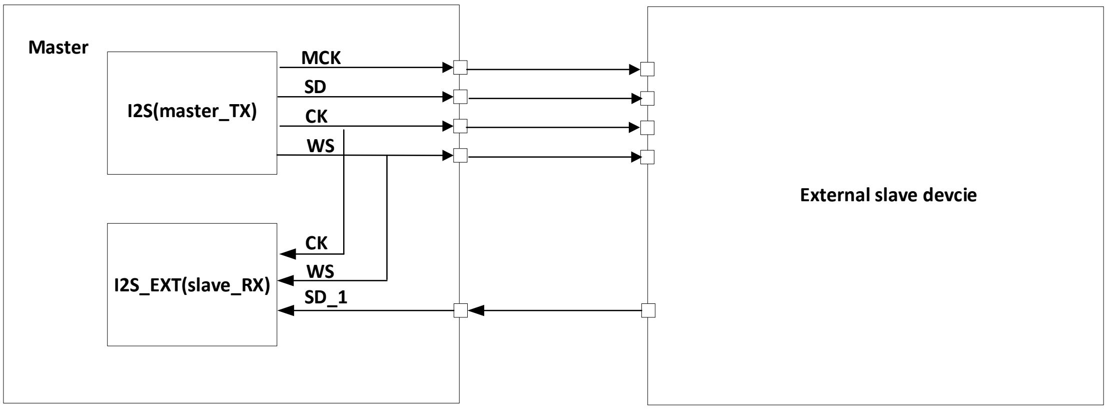
In master mode, I2S uses its own clock generator to generate clock signals for communication, and this clock generator is also the clock source of the master clock output

26.4.1 I2S Full-dDuplex Function

I2S can be combined with the I2SEXT module to support I2S full-duplex mode. When the I2S and I2SEXT are combined to achieve full-duplex functionality, the I2S can be configured as master or slave mode, while the I2SEXT can only be configured as slave mode. I2SEXT shares the CK and WS of I2S, and there are several corresponding configurations:

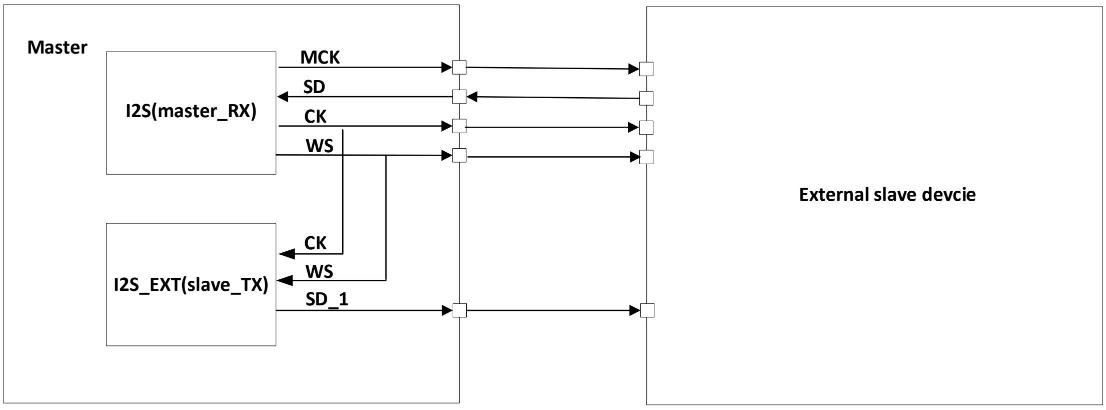
1) I2S configured as master transmit mode (I2SCFG=10), I2S_EXT configured as slave receive mode (I2SCFG=01):

Figure 26-15 I2S master transmit (I2SCFG=10), I2S_EXT slave receive (I2SCFG=01)



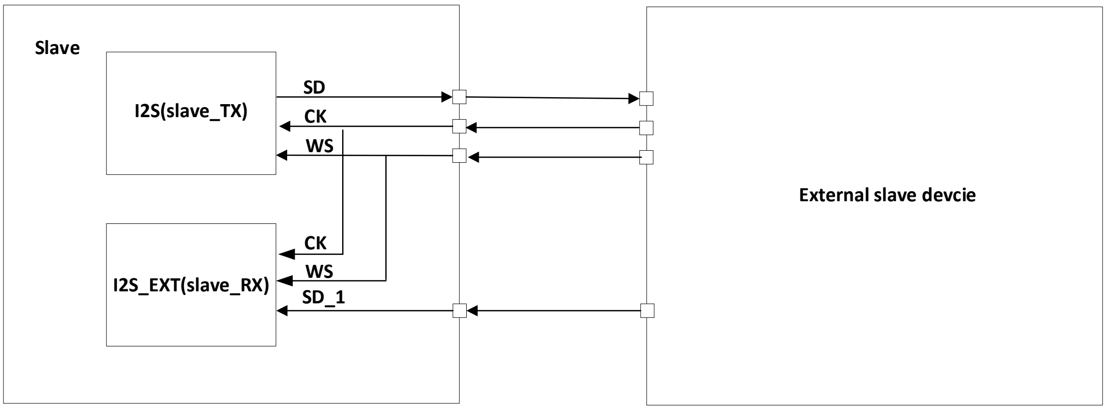
2) I2S configured as master receive mode (I2SCFG=11), I2S_EXT configured as slave transmit mode (I2SCFG=00):

Figure 26-16 I2S master receive (I2SCFG=11), I2S_EXT slave transmit (I2SCFG=00)



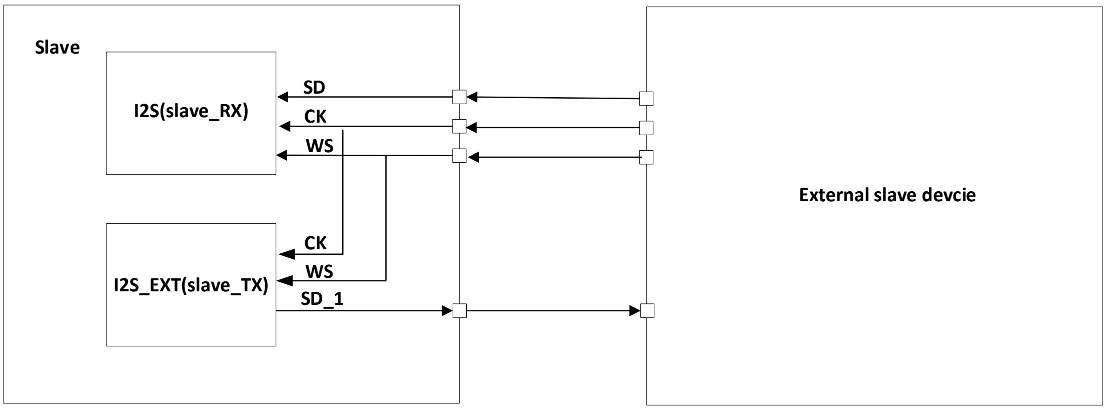
3) I2S configured as slave transmit mode (I2SCFG=00), I2S_EXT configured as slave receive mode (I2SCFG=01):

Figure 26-17 I2S slave transmit mode (I2SCFG=00), I2S_EXT slave receive (I2SCFG=01)



4) I2S configured as slave receive mode (I2SCFG=01), I2S_EXT configured as slave transmit mode (I2SCFG=00):

Figure 26-18 I2S slave receive (I2SCFG=01), I2S_EXT slave transmit (I2SCFG=00)



26.4.2 Supported Audio Protocols

Both the I2S and I2S_EXT modules have a 16-bit data register for transmitting or receiving data. When software writes data to the data register, it writes the corresponding data based on the current channel in the transmission. Similarly, when reading from the register, the received data can be identified for each channel by checking the CHSIDE bit in the register. The left channel always transmit data before the right channel (the CHSIDE bit is not meaningful in PCM protocol).

By setting the SPI_I2S_CFGR.DATLEN bits, the user can set the length of the data to be transmitted, and set the data bit width of the channel by setting the SPI_I2S_CFGR.CHLEN bits. There are 4 data formats for sending data as follows:

- 16-bit data is packed into 16-bit data frame
- 16-bit data is packed into 32-bit data frame(The first 16 bits are valid data, and the last 16 bits are set to 0 by hardware).
- 24-bit data is packed into 32-bit data frame(The first 24 bits are valid data, and the last 8 bits are set to 0 by hardware).
- 32-bit data is packed into 32-bit data frame

I2S uses the same SPI_DAT register as SPI to transmit and receive 16-bit wide data. If I2S needs to transmit or receive 24-bit or 32-bit wide data, the CPU needs to read or write the SPI_DAT register twice. On the other hand, when I2S transmits or receives 16-bit wide data, the CPU only needs to read or write the SPI_DAT register once.

Regardless of which data format and communication standard is used, I2S always sends the data high-order bit (MSB) first.

26.4.2.1 I2S Philips standard

Using the I2S Philips standard, the device that transmits data on the falling edge of the clock, and the device that receives data on the rising edge of the clock. The WS signal should be valid one clock before the first data bit (MSB) is sent and will change on the falling edge of the clock signal.

Figure 26-26-19 I2S Philips Protocol Waveform (16/32-Bit Full Precision, CLKPOL = 0)

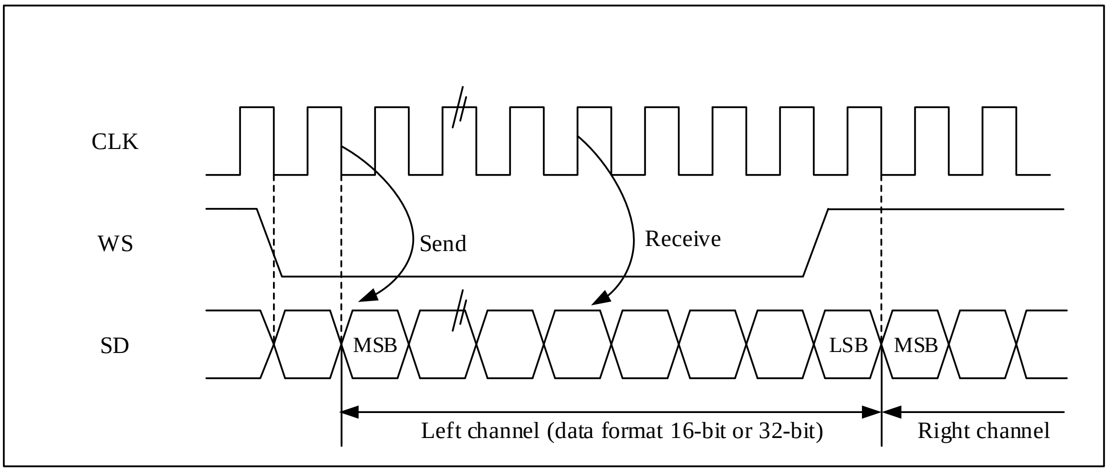
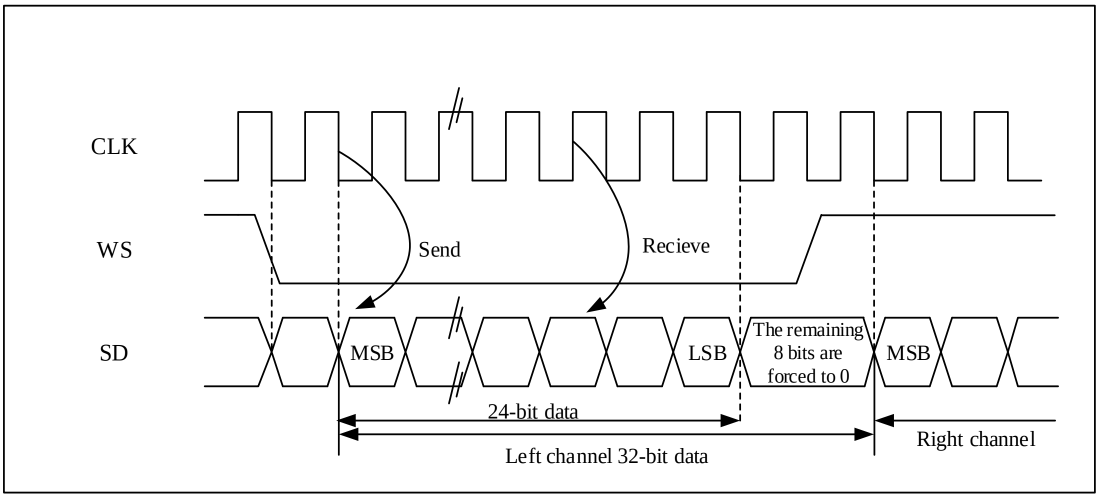
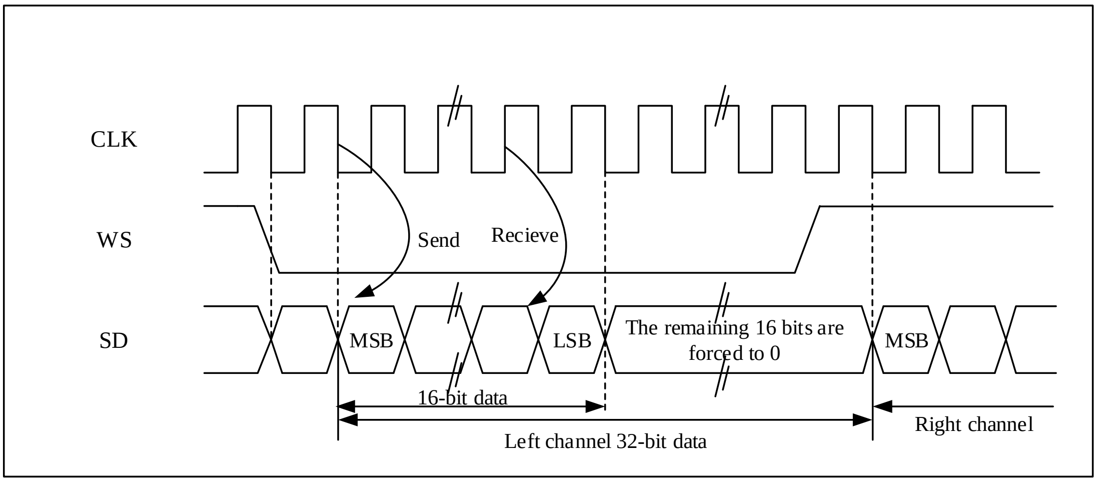


Figure 26-26-20 I2S Philips Protocol Standard Waveform (24-Bit Frame, CLKPOL = 0)



If the 24-bit data needs to be packaged into 32-bit data frame format, the CPU needs to read or write the SPI_DAT register twice during each frame of data transmission. For example, if the user sends 24-bit data 0x95AA66, the CPU will first write 0x95AA into the SPI_DAT register, and then write 0x66XX into the SPI_DAT register (only the upper 8-bit data is valid, the lower 8-bit data is meaningless and can be any value); if the user receives 24-bit data 0x95AA66, the CPU will first read the SPI_DAT register to get 0x95AA, and then read the SPI_DAT register to get 0x6600 (only the upper 8-bit data is valid, and the lower 8-bit data is always 0).

Figure 26-26-21 I2S Philips Protocol Standard Waveform (16-Bit Extended to 32-Bit Packet Frame, CLKPOL = 0)



If 16-bit data needs to be packed into 32-bit data frame format, the CPU only needs to read or write the SPI_DAT register once for each frame of data transmission. The lower 16 bits of data for expansion to 32 bits are always set to 0x0000. For example, if the user sends or receives 16-bit data 0x89C1 (extended to 32-bit data is 0x89C10000). In the process of sending data, the upper 16-bit half word (0x89C1) needs to be written into the SPI_DAT register; the user can write new data until the SPI_STS.TE bit is set. An interrupt is generated if the user enables the corresponding interrupt. The sending is performed by hardware, even if the last 16 bits (0x0000) are not sent, the hardware will set the TE (SPI_STS.TE) bit to 1 and the corresponding interrupt will be generated. In the process of receiving data, the RNE flag (SPI_STS.RNE) will be set to 1 after each time the device receives the upper 16-bit halfword (0x89C1). An interrupt is generated if the user enables the corresponding interrupt. In this way, there is more time between 2 reads and writes, which can prevent underflow or overflow from happening.

26.4.2.2 MSB alignment standard

In the MSB alignment standard, the device transmits the data on the falling edge of the clock, and the device receives the data on the rising edge of the clock. The WS signal and the first data bit (MSB) are generated simultaneously.

The standard data receiving and transmitting processing mode is the same as I2S Philips standard.

Figure 26-26-22 The MSB is Aligned With 16-Bit or 32-Bit Full Precision, CLKPOL = 0

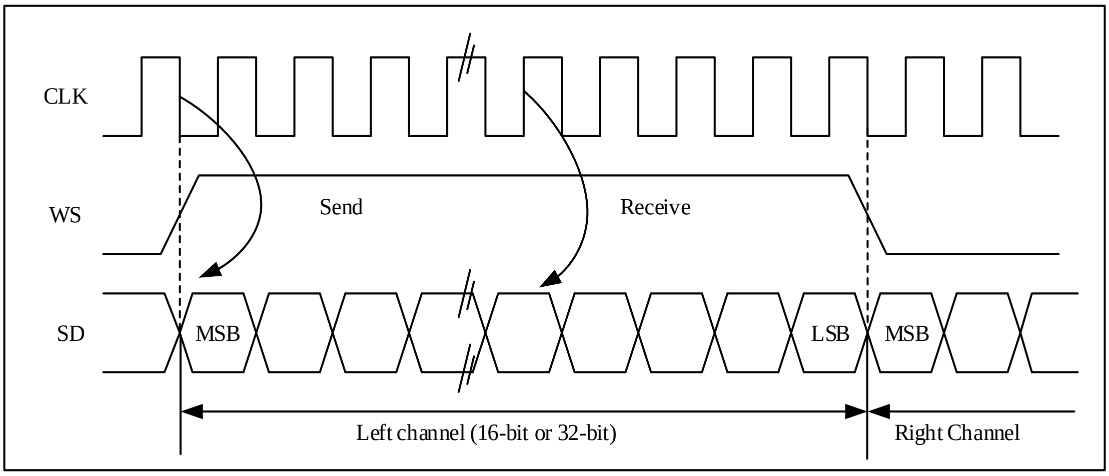


Figure 26-26-23 MSB Aligns 24-Bit Data, CLKPOL = 0

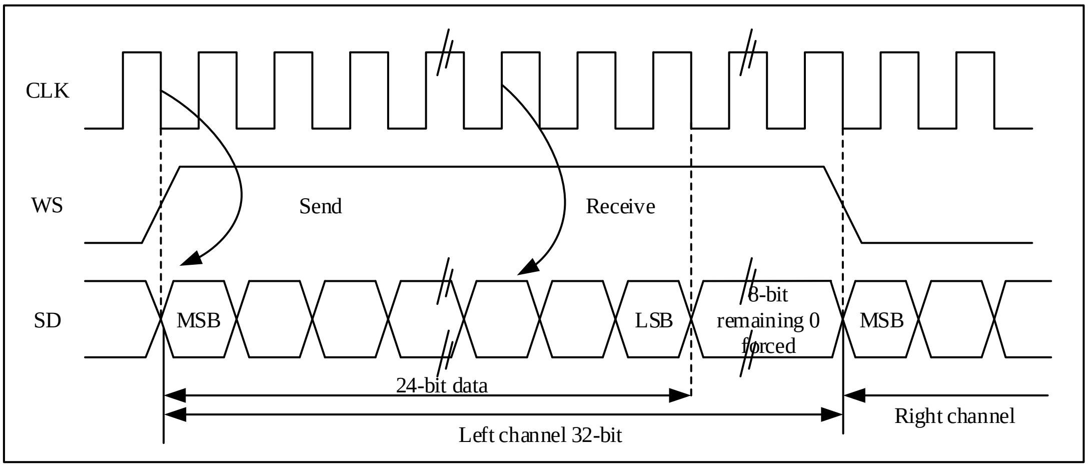
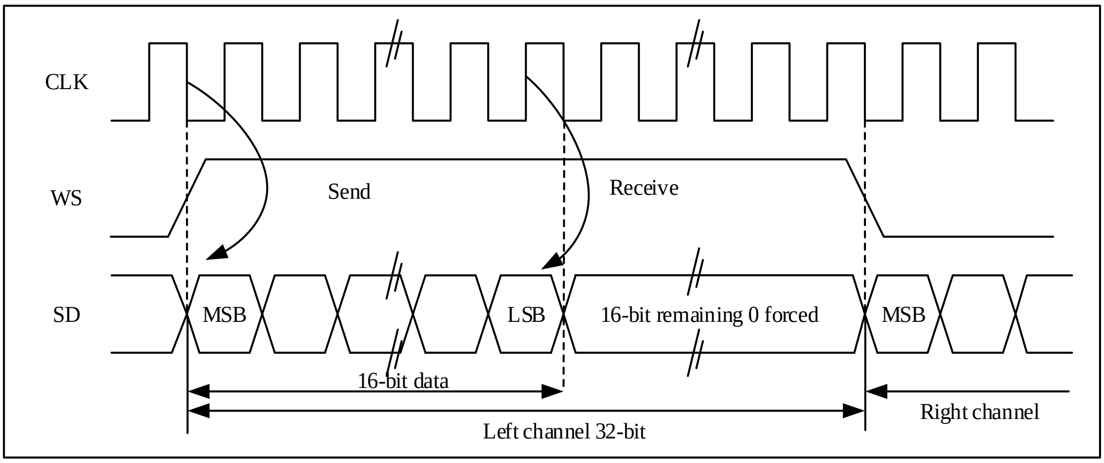


Figure 26-26-24 MSB-Aligned 16-Bit Data Is Extended to 32-Bit Packet Frame, CLKPOL = 0



26.4.2.3 LSB alignment standard

In 16-bit or 32-bit full-precision frame format, LSB alignment standard is the same as MSB alignment standard.

Figure 26-26-25 LSB Alignment 16-Bit or 32-Bit Full Precision, CLKPOL = 0

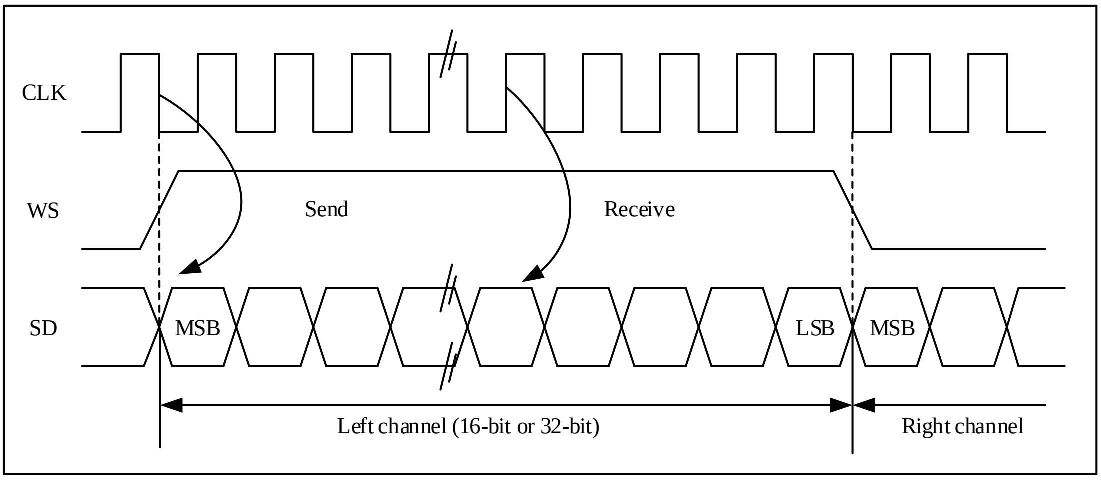
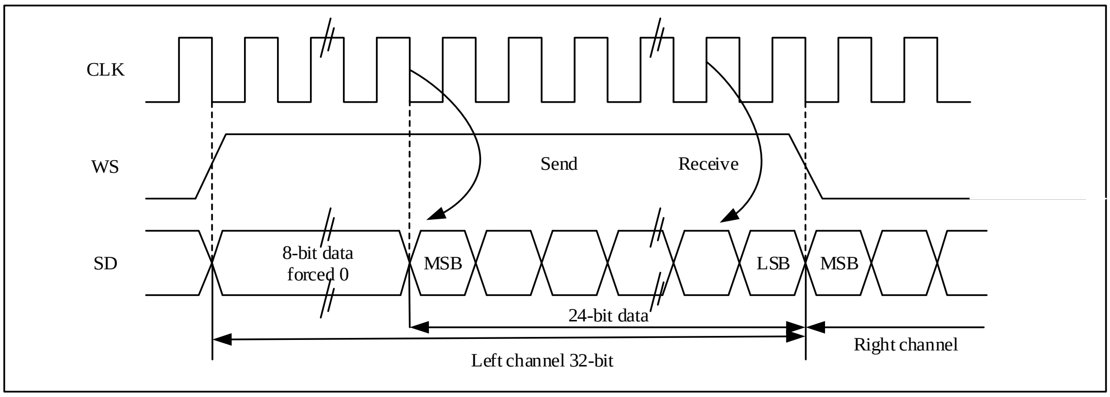
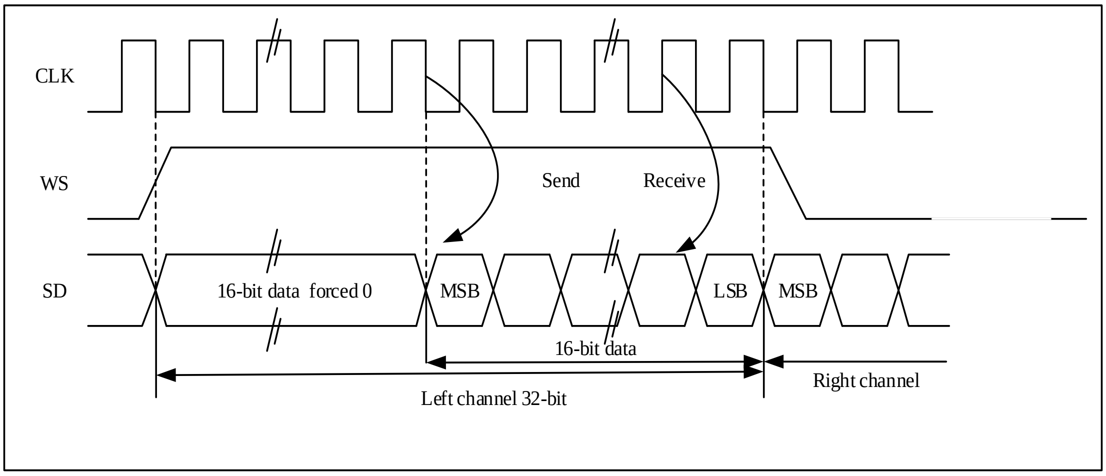


Figure 26-26-26 LSB Aligns 24-Bit Data, CLKPOL = 0



If the 24-bit data needs to be packed into the 32-bit data frame format, the CPU needs to read or write the SPI_DAT register twice during each frame of data transmission. For example, if the user sends 24-bit data 0x95AA66, the CPU will first write 0xXX95 (only the lower 8-bit data is valid, the upper 8-bit data is meaningless and can be any value) into the SPI_DAT register, and then write 0xAA66 into the SPI_DAT register. If the user receives 24-bit data 0x95AA66, the CPU will first read the SPI_DAT register to get 0x0095 (only the lower 8 bits are valid, the upper 8 bits are always 0), and then read the SPI_DAT register to get 0xAA66.

Figure 26-26-27 LSB Aligned 16-Bit Data Is Extended to 32-Bit Packet Frame, CLKPOL = 0



If the 16-bit data needs to be packaged into a 32-bit data frame format, the CPU only needs to read or write the SPI_DAT register once for each frame of data transmission. The upper 16 bits of extended to 32 bits data are set to 0x0000 by hardware, if the user sends or receives 16-bit data 0x89C1 (extended to 32-bit data is 0x000089C1). In the process of sending data, the upper 16-bit halfword (0x0000) needs to be written to the SPI_DAT register first; once the valid data starts to be send, the next TE (SPI_STS.TE) event will be generated. In the process of receiving data, once the device receives valid data, the RNE (SPI_STS.RNE) event will be generated. In this way, there is more time between 2 reads and writes, which can prevent underflow or overflow from happening.

26.4.2.4 PCM standard

In the PCM standard, there are two frame structures, short frame and long frame. The user can select the frame structure by setting the SPI_I2S_CFGR.PCMSYNCbits. The WS signal indicates frame synchronization information. The WS signal for synchronizing long frames is 13 bits effective; the WS signal length for synchronizing short frames is 1 bit.

The standard data receiving and transmitting processing mode is the same as I^{2}S Philips standard.

Figure 26-26-28 PCM Standard Waveform (16 Bits)

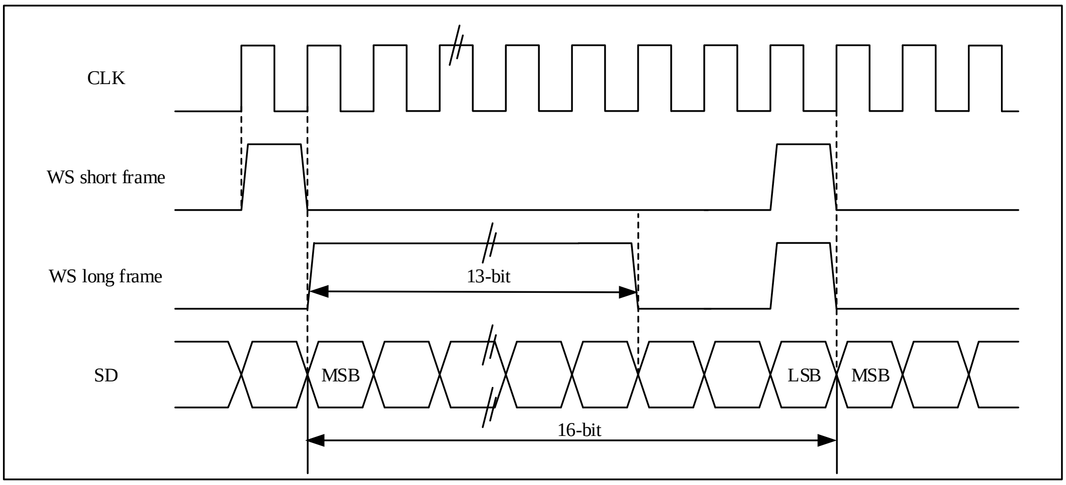
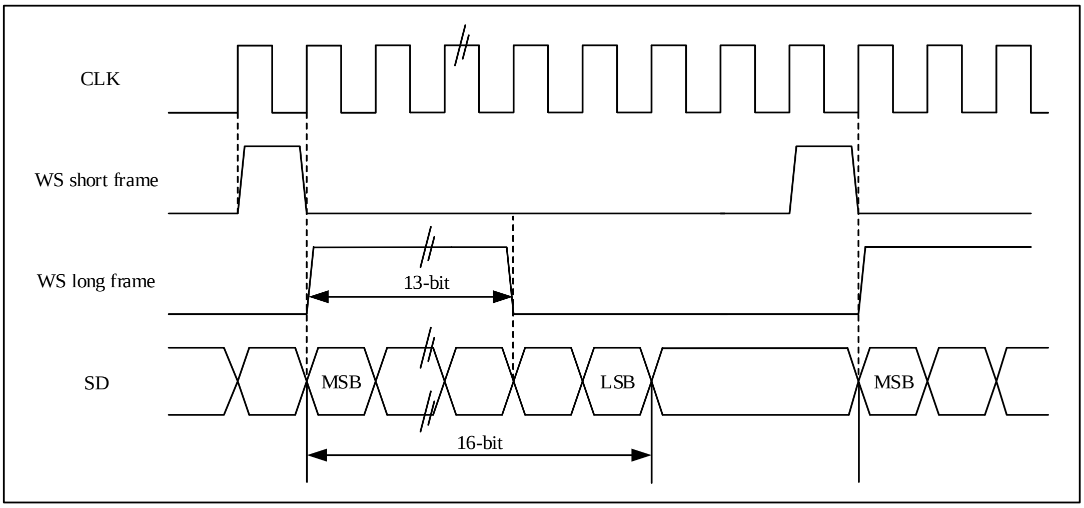


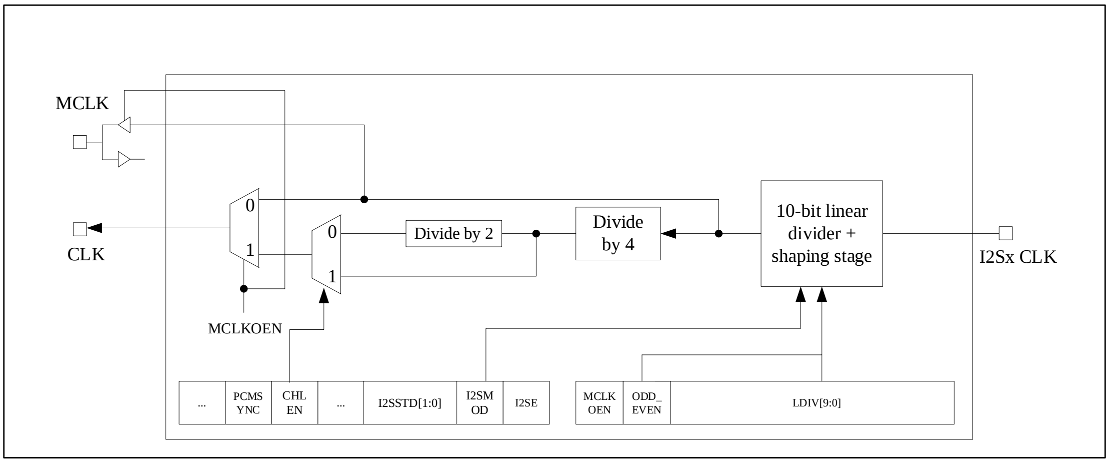
Figure 26-26-29 PCM Standard Waveform (16-Bit Extended to 32-Bit Packet Frame)



26.4.3 Clock Generator

In the master mode, the linear divider needs to be set correctly in order to obtain the desired audio frequency.

Figure 26-26-30 I2S Clock Generator Structure



Note: The clock source for I2SxCLK can be an independent SHRTPLL, HSI, SYSYCLK, or an external clock input.

The bit rate of I2S determines the data flow on the I2S data line and the frequency of the I2S clock signal.

I2S bit rate = number of bits per channel × number of channels × audio sampling frequency

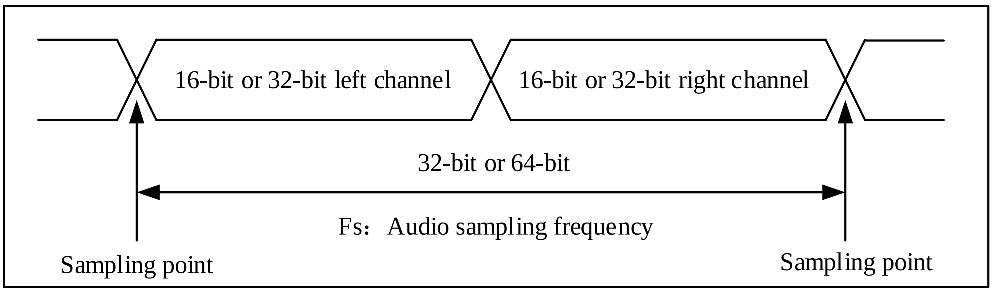
For a signal with left and right channels and 16-bit audio, the I2S bit rate is calculated as:

I2S bit rate = 16 × 2 × FS

If the packet length is 32 bits, there are:

I2S bit rate = 32 × 2 × FS

Figure 26-26-31 Audio Sampling Frequency Definition


16-bit or 32-bit left channel
16-bit or 32-bit right channel
32-bit or 64-bit
Fs：Audio sampling frequency
Sampling point
Sampling point

The sampling signal frequency of the audio can be set by setting the SPI_I2S_PREDIV.ODDEVEN bit and the SPI_I2S_PREDIV.LDIV[7:0] bits. Audio can be sampled at 96kHz, 48kHz, 44.1kHz, 32kHz, 22.05kHz, 16kHz, 11.025kHz, or 8kHz (or any value within this range). Set the linear divider according to the following formula:

When MCLKOEN = 1 and CHLEN= 0, $F_S = I^2Sx\ \ CLK/[(16 \times 2) \times ((2 \times LDIV) + ODD\_EVEN) \times 8]$

When MCLKOEN = 1 and CHLEN = 1, $F_S = I^2Sx\ \ CLK/[(32 \times 2) \times ((2 \times LDIV) + ODD\_EVEN) \times 4]$

When MCLKOEN = 0 and CHLEN = 0, $F_S = I^2Sx\ \ CLK/[(16 \times 2) \times ((2 \times LDIV) + ODD\_EVEN)]$

When MCLKOEN = 0 and CHLEN = 1, $F_S = I^2Sx\ \ CLK/[(32 \times 2) \times ((2 \times LDIV) + ODD\_EVEN)]$

The exact audio frequency can be obtained by referring to the clock configuration in the table below.

Table 26-26-2 Use The Standard 8mhz HSE Clock to Get Accurate Audio Frequency.

SYSCLK（MHz）	I^{2}S_LDIV		I^{2}S_ODDEVEN		MCLK	Target Fs(Hz)	Real FS (Hz)		Error	
	16 bits	32 bits	16 bits	32 bits			16 bits	32 bits	16 bits	32 bits
240	19	10	1	0	without	192000	192307.7	187500	0.16%	2.34%
240	39	19	0	1	without	96000	96153.85	96153.85	0.16%	0.16%
240	78	39	0	0	without	48000	48076.92	48076.92	0.16%	0.16%
240	85	42	0	1	without	44100	44117.65	44117.65	0.04%	0.04%
240	117	58	0	1	without	32000	32051.28	32051.28	0.16%	0.16%
240	170	85	0	0	without	22050	22058.82	22058.82	0.04%	0.04%
240	234	117	1	0	without	16000	15991.47	16025.64	0.05%	0.16%
240	340	170	0	0	without	11025	11029.41	11029.41	0.04%	0.04%
240	469	234	0	1	without	8000	7995.736	7995.736	0.05%	0.05%
240	2	2	1	1	yes	192000	187500	187500	2.34%	2.34%
240	5	5	0	0	yes	96000	93750	93750	2.34%	2.34%
240	10	10	0	0	yes	48000	46875	46875	2.34%	2.34%
240	10	10	1	1	yes	44100	44642.86	44642.86	1.23%	1.23%
240	14	14	1	1	yes	32000	32327.59	32327.59	1.02%	1.02%
240	21	21	1	1	yes	22050	21802.33	21802.33	1.12%	1.12%
240	29	29	1	1	yes	16000	15889.83	15889.83	0.69%	0.69%
240	42	42	1	1	yes	11025	11029.41	11029.41	0.04%	0.04%
240	58	58	1	1	yes	8000	8012.821	8012.821	0.16%	0.16%

Note: Using SHRPLL as the clock source can achieve higher precision.

26.4.4 I^{2}S Transmission And Reception Sequence

There are several configuration modes corresponding to master full-duplex mode:

- I2S configured as master transmit mode (I2SCFG=10), I2S_EXT configured as slave receive mode (I2SCFG=01)
- I2S configured as master receive mode (I2SCFG=10), I2S_EXT configured as slave transmit mode (I2SCFG=01)

Setting I2S to operate in master mode, the serial clock is output by the CK pin, and the word select signal is generated by the WS pin. The output of the master clock (MCK) can be selected or deselected by setting the SPI_I2S_PREDIV.MCLKOEN register bit.

I^{2}S initialization sequences:

1) Set the SPI_I2S_PREDIV.LDIV[9:0] and SPI_I2S_PREDIV.ODDEVEN to define the serial clock baud rate that matches the audio sampling frequency.

2) The user can set the CLKPOL bit to define the polarity of the communication clock when idle.If the user needs the master device to provide the main clock MCLK to the external DAC/ADC audio device, set the SPI_I2S_PREDIV.MCLKOEN = 1.

3) Set the SPI_I2S_CFGR.I2SMOD bit to '1' to enable the I2S function, set the SPI_I2S_CFGR.I2SSTD[1:0] and SPI_I2S_CFGR.PCMSYNC bits to select the I2S standard to be used, set the SPI_I2S_CFGR.CHLEN to choose

the data bit length for each channel. Also, set the SPI_I2S_CFGR.I2SCFG[1:0] to select the I2S master mode and direction (transmit or receive).

4) If needed, the desired interrupt and DMA functions can be enabled by setting the SPI_CTRL2 register.

5) Set the SPI_I2S_CFGR.I2SE register bit to '1'.

6) The WS and CK pins need to be configured as output mode. If the SPI_I2S_PREDIV.MCLKOEN bit is set to '1', the MCK pin also needs to be configured as output mode.

26.4.5 I2S Master Mode

26.4.5.1 Transmission Process in Master Mode

When writing one half-word (16 bits) of data to the transmit buffer, the transmission process begins.

Assuming the first data written to the transmit buffer corresponds to the left channel data. When the data is moved from the transmit buffer to the shift register, the TE flag is set to '1'. At this point, the corresponding right channel data needs to be written to the transmit buffer. The CHSIDE flag indicates which channel the current data to be transmitted corresponds to. The value of the CHSIDE flag is updated when TE is '1', so it is meaningful when TE is '1'. A complete data frame is considered complete only after both left and right channel data have been transmitted. It is not permissible to transmit only part of a data frame, such as data from only the left channel.

As the first bit of data is sent out, the half-word data is transmitted in parallel to the 16-bit shift register, and the subsequent bits are then sent out from the I2S_SD/I2S_EXT_SD pins in order, starting with the most significant bit. Each time data is transferred from the transmit buffer to the shift register, the TE flag is set to '1', and if the SPI_CTRL2.TEINTEN bit is set to '1', an interrupt is generated.

To ensure continuous audio data transmission, it is recommended to write the next data to be transmitted to the SPI_DAT register before the current transmission is completed. It is recommended to wait for the SPI_STS.TE=1 and SPI_STS.BUSY=0 flags before disabling the I2S function, and then clear the SPI_I2S_CFGR.I2SE bit to '0'.

26.4.5.2 Reception Process in Master Mode

Select the master receive mode by configuring SPI_I2S_CFGR[1:0].

Regardless of the data and channel length, audio data is always received in packets of 16 bits. When the data is transferred from the shift register to the receive buffer, the RNE flag bit is set to 1, at this time, the data is ready and can be read from the SPI_DAT register. If the SPI_CTRL2.RNEINTEN bit is set to 1, an interrupt will be generated. According to the configured data and channel length, the received audio data will need to be transferred to the receive buffer once or twice.

Reading the SPI_DAT register to clear the RNE flag. The CHSIDE is updated after each reception. Its value depends on the WS signal generated by the I2S unit. If the previously received data is not read, new data is received again, an overflow occurs, and the SPI_STS.OVER flag is set to 1. If the SPI_CTRL2.ERRINTEN bit is set to 1, an interrupt is generated to indicate that an error has occurred.

To disable the I2S module, special operations need to be performed to ensure that the I2S module can complete the transmission cycle properly without starting new data transfers. The operation process is related to data configuration, channel length, and the audio protocol:

- Data length is 16 bits, channel length is 32 bits (DATLEN=00, CHLEN=1), LSB alignment standard

(I2SSTD=10).

1) Wait for the penultimate RNE flag (SPI_STS.RNE) bit to be set to' 1'.

2) Software delay, waiting for 17 I2S clock cycles.

3) Disable I2S (SPI_I2S_CFGR.I2SEN = 0).

- Data length is 16 bits, the channel length is 32 bits (DATLEN=00, CHLEN=1), the MSB alignment standard (I2SSTD=01), I2S Philips standard (I2SSTD=00) or PCM standard (I2SSTD= = 11)

1) Wait for the last RNE flag (SPI_STS.RNE) bit to be set to' 1'.

2) Software delay, waiting for 1 I2S clock cycle.

3) Turn off I2S (SPI_I2S_CFGR.I2SEN = 0).

- For all other combinations of DATLEN and CHLEN, and any audio mode selected by I2SSTD, the I2S can be disabled in the following steps:

1) Wait for the penultimate RNE flag (SPI_STS.RNE) bit to be set to' 1'.

2) Software delay, waiting for 1 I2S clock cycle.

3) Disable I2S (SPI_I2S_CFGR.I2SEN = 0).

26.4.6 I2S Slave Mode(Full Duplex)

There are several configuration modes corresponding to slave full-duplex mode:

1) I2S configured as slave transmit mode (SPI_I2S_CFGR.I2SCFG=00), I2S_EXT configured as slave receive mode (I2S_CFGR.I2SCFG=01).

2) I2S configured as slave receive mode (SPI_I2S_CFGR I2SCFG=01), I2S_EXT configured as slave transmit mode (I2S_CFGR.I2SCFG=00).

In slave mode, I2S/I2S_EXT can be set to transmit and receive modes. The configuration process for slave mode is similar to that of master mode. In slave mode, there is no need for the I2S interface to provide a clock. The clock signal and WS signal are provided by the external master I2S device, connected to the respective pins. Therefore, the user does not need to configure the clock.

The configuration steps are as follows:

1) Set the SPI_I2S_CFGR.I2SMOD bit to 1 to enable the I2S module. Select the I2S standard by configuring I2SSTD[1:0]. Choose the length of the data to betransferred by configuring DATLEN[1:0]. Configure CHLEN to select the data bit length for each channel. Configure the SPI_I2S_CFGR.I2SCFG[1:0] to choose the data direction (transmit or receive) for the I2S slave mode.

2) If needed, the desired interrupt and DMA functions can be enabled by setting the SPI_CTRL2 register.

3) The SPI_I2S_CFGR.I2SE bit must be set to '1'.

26.4.6.1 Transmission Process in Slave Mode

The transmitting process begins when an external master sends a clock signal, and when a WS signal requires data transmission. Only when the slave device is enabled and the data has been written to the I2S data register, the external master device can start communication.

For I2S MSB alignment and LSB alignment modes, the first data item written to the data register corresponds to the left channel data. When communication begins, the data is transferred from the transmit buffer to the shift register, and then the TE flag is set to '1'; at this point, the corresponding data item for the right channel should be written to the I2S data register.

The SPI_STS.CHSIDE flag indicates which channel the currently transmitted data corresponds to. Compared with the master mode transmitting process, in the slave mode, CHSIDE depends on the WS signal of the external master I2S device (WS signal is 1 means the left channel). This means that the I2S slave needs to be ready with the first data to be sent before receiving the clock signal generated by the master.

As the first bit of data is transmitted, the half-word data is transmitted in parallel through the I2S internal bus to the 16-bit shift register, and then the remaining bits are sent out from the I2S_SD/I2S_EXT_SD pins in order, starting with the most significant bit. Each time data is transferred from the transmit buffer to the shift register, the TE flag is set to '1', and if the SPI_CTRL2.TEINTEN bit is set to '1', an interrupt is generated.

Before writing data to the transmit buffer, ensure that the TE flag is set to '1'. To ensure continuous audio data transmission, it is recommended to write the next data to be transmitted to the SPI_DAT register before the current transmission is completed.

Before writing data to the transmit buffer, ensure that the TE flag is set to '1'. To ensure continuous audio data transmission, it is recommended to write the next data to be transmitted to the SPI_DAT register before the current transmission is completed. If the new data is not written to the SPI_DAT register before the first clock edge representing the next data transfer arrives, the underrun flag will be set to '1'. It indicates a software data transmission error. If the SPI_CTRL2.ERRINTEN bit is set to '1', an interrupt will be generated when the UNDER flag in the SPI_STS register is high. It is recommended to close the I2S at this point, and then restart sending data from the left channel.

26.4.6.2 Reception Process in Slave Mode

Select the slave receive mode by configuring I2SCFG[1:0].

Regardless of the data and channel length, audio data is always received in packets of 16 bits. When the data is transferred from the shift register to the receive buffer, the RNE flag bit is set to 1, at this time, the data is ready and can be read from the SPI_DAT register. If the SPI_CTRL2.RNEINTEN bit is set to 1, an interrupt will be generated. According to the configured data and channel length, the received audio data will need to be transferred to the receive buffer once or twice.

The CHSIDE is updated after each reception. Its value depends on the WS signal generated by the master device. Reading the SPI_DAT register to clear the RNE flag. If the previously received data is not read, new data is received again, an overflow occurs, and the SPI_STS.OVER flag is set to 1. If the SPI_CTRL2.ERRINTEN bit is set to 1, an interrupt is generated to indicate that an error has occurred. When disabling the I2S function, the SPI_I2S_CFGR.I2SE bit should be cleared to 0 at the last RNE=1.

26.4.7 Status Flag

There are the following 4 flag bits in the SPI_STS register for monitoring the status of the I2S bus.

BUSY flag (BUSY)

The SPI_STS.BUSY flag is set and cleared by hardware (writing to this bit has no effect), indicating the status of the I2S communication. When this bit is '1', it indicates that I2S communication is in progress, with one exception: in master receive mode (I2SCFG=11), the SPI_STS.BUSY flag is always low during reception.

Before disabling the I2S module in software, you can use the SPI_STS.BUSY flag to check if the transmission has ended. This can help avoid disrupting the final transmission.

When the transmission starts, the SPI_STS.BUSY flag is set to '1', unless the I2S module is in master receive mode.

The flag is cleared under the following conditions:

- When the transmission ends (except in master transmit mode, where communication is continuous).
- When the I2S module is disabled.

When communication is continuous:

- In master transmit mode, the SPI_STS.BUSY flag remains high throughout the entire transmission.
- In slave mode, the SPI_STS.BUSY flag transitions low for one I2S clock cycle between each data item transfer.

TX buffer empty flag (TE)

When the send buffer is empty, this flag is set to 1, indicating that new data can be written into the SPI_DAT register. When the send buffer is not empty, this flag is cleared to 0.

RX buffer not empty flag (RNE)

When the receive buffer is not empty, this flag is set to 1, indicating that valid data has been received into the receive buffer. When reading the SPI_DAT register, this flag is set to 0.

Channel Side flag (CHSIDE)

In transmit mode, the flag is updated when the TE flag (SPI_STS.TE) is set, indicating the channel where the data sent on the I2S_SD/I2S_EXT_SD pin is located. If an underflow error occurs in transmit mode, the value of this flag is invalid, and the I2S needs to be disabled and enabled before communication can resume. In receive mode, the flag is updated when the RNE flag (SPI_STS.RNE) is set. If an overflow (SPI_STS.OVER) error occurs in receive mode, this flag is invalid, and the I2S needs to be disabled and enabled before communication can resume Under the PCM standard, this flag has no meaning.

26.4.8 Error Flag

The SPI_STS register has 2 error flag bits.

Overflow flag (OVER)

If new data is received before the previous data is read, causing an overflow, the flag is set to '1'. If the SPI_CTRL2.ERRINTEN bit is set to '1', an interrupt is generated to indicate an error has occurred. In this case, the contents of the receive buffer are not updated with the new data from the transmitting device. Reading the SPI_DAT register will return the last correctly received data. All other 16-bit data sent by the transmitting device after the overflow occurred will be lost. Reading the SPI_DAT register first and then the SPI_STS register can clear this flag.

Underflow flag (UDER)

In slave send mode, when the first clock edge of sending data arrives, if the transmit buffer is still empty, the UNDER flag (SPI_STS.UNDER) is set to 1. If the SPI_CTRL2.ERRINTEN bit is set to '1', an interrupt will be generated. This flag can be cleared by reading the SPI_STS register.

26.4.9 I^{2}S Interrupt

The following table lists all I^{2}S interrupts.

Table 26-26-3 I2S Interrupt Request

Interrupt Event	Event Flag Bit	Enable Control Bit
Send buffer empty flag	TE	TEINTEN
Receive buffer non empty flag	RNE	RNEINTEN
Underflow flag bit	UNDER	ERRINTEN

26.4.10 DMA Function

To achieve the maximum communication frequency, it is necessary to promptly write data to the I2S transmit buffer and also read the data from the receive buffer in a timely manner to prevent overflow. To facilitate high-speed data transfer, I2S has implemented a simple request/response DMA mechanism.

When the corresponding enable bit on the SPI_CTRL2 register is set, the I2S module can generate DMA transfer requests. Both the transmit buffer and receive buffer also have their own DMA requests.

- During transmission, a DMA request is generated each time TE is set to '1', and the DMA controller writes data to the SPI_DAT register, clearing the TE flag in the process.
- During reception, a DMA request is generated each time RNE is set to '1', and the DMA controller reads data from the SPI_DAT register, clearing the RNE flag in the process.

When only using I2S/I2S_EXT to transmit data, only the transmit DMA channel of I2S/I2S_EXT needs to be enabled. In this case, since no received data is being read, OVER is set to '1'. When only using I2S/I2S_EXT to receive data, only the receive DMA channel of I2S/I2S_EXT needs to be enabled.

26.5 SPI and I2S Register

26.5.1 SPI Register Overview

Table 26-26-4 SPI Register Overview

Offset	Register	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	SPI_CTRL1	BIDIMODE	BIDIROEN	RONLY	SSMEN	SSEL	SSOEN	CRCNEXT	DATFF	LSBFF	MSEL	CLKPHA	CLKPOL	Reserved	BR[2:0]		
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0		0	0	0

Offset	Register	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x04	SPI_CTRL2	Reserved	RXCPINTEN	CRCNMISEN	RXFFUINTE	RXFHFINTE	TXFHEINTE	FIFOCLR	FIFOEN	SS_POL	ERRINTEN	RNEINTEN	TEINTEN	CRCEN	TDMAEN	RDMAEN	SPIEN
	Reset value		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x08	SPI_STS	Reserved			RXTSCP	RXFIFFU	TXFIFFU	RXFIFHF	TXFIFHE	CHSIDE	UNDER	OVER	MODERR	CRCERR	BUSY	RNE	TE
	Reset value				0	0	0	0	1	0	0	0	0	0	0	0	1
0x0C	SPI_DAT	DAT[15:0]															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x10	SPI_CRCTDAT	CRCTDAT[15:0]															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x14	SPI_CRCRDAT	CRCRDAT[15:0]															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x18	SPI_CRCPOLY	CRCPOLY [15:0]															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
0x1C	SPI_I2S_CFGR	Reserved				PCMBYPASS	CLKPOL	PCMSYNC	CHLEN	DATLEN[1:0]		I2SCFG[1:0]		I2SSTD[1:0]		I2SMOD	I2SE
	Reset value					0	0	0	0	0	0	0	0	0	0	0	0
0x20	SPI_I2S_PREDIV	Reserved				MCLKOEN	ODDEVEN	LDIV[9:0]									
	Reset value					0	0	0	0	0	0	0	0	0	0	1	0
0x24	SPI_RX_FIFO	RXFIFDAT[15:0]															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x28	SPI_FIFO_NUM	Reserved									RXFBSNUM[2:0]			Reserved	TXFBSNUM[2:0]		
	Reset value										1	0	0		1	0	0
0x30	SPI_FIFO_CNT	Reserved								RXFIFCNT[3:0]				TXFIFCNT[3:0]			

Offset	Register	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reset value									0	0	0	0	0	0	0	0
0x34	SPI_TRANS_NUM	TRANSNUM[15:0]															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x38	SPI_CR3	Reserved												DELAYTIME[3:0]			
	Reset value													0	0	0	0
0x204	I2S_CTRL2	Reserved									ERRINTEN	RNEINTEN	TEINTEN	Reserved	TDMAEN	RDMAEN	Reserved
	Reset value										0	0	0		0	0	
0x208	I2S_STS	Reserved								CHSIDE	UNDER	OVER	Reserved		BUSY	RNE	TE
	Reset value									0	0	0			0	0	1
0x20C	I2S_DAT	DAT[15:0]															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x21C	I2S_CFGR	Reserved					CLKPOL	PCMSYNC	CHLEN	DATLEN[1:0]		I2SCFG[1:0]		I2SSTD[1:0]		I2SMOD	I2SE
	Reset value						0	0	0	0	0	0	0	0	0	0	0

26.5.2 SPI Control Register 1 (SPI_CTRL1) (Not Used in I2S Mode)

Address: 0x00

Reset value: 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BIDIRMODE	BIDIROEN	RONLY	SSMEN	SSEL	SSOEN	CRCNEXT	DATFF	LSBFF	MSEL	CLKPHA	CLKPOL	Reserved	BR[2:0]		
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw		rw		

Bit Field	Name	Description
15	BIDIRMODE	Bidirectional data mode enable 0: Select the "two-wire one-way" mode. 1: Select the "one-wire bidirectional " mode.

Bit Field	Name	Description
		Note: Not used in I^{2}S mode.
14	BIDIROEN	Output enable in bidirectional mode 0: Output disable (receive-only mode). 1: Output enabled (send-only mode). In master mode, the "one-wire" data line is the MOSI pin, and in slave mode, the "one-wire" data line is the MISO pin. *Note: Not used in I^{2}S mode.*
13	RONLY	Only receive mode This bit, together with the SPI_CTRL1.BIDIRMODE bit, determines the transfer direction in two-wire one-way mode. In the application scenario of multiple slave devices, this bit is only set to 1 by the unaccessed slave device, and only the accessed slave device can output, so as to avoid data line conflicts. 0: Full duplex (sending mode and receiving mode). 1: Disable output (receive-only mode). *Note: Not used in I^{2}S mode.*
12	SSMEN	Software slave device management When the SPI_CTRL1.SSMEN bit is set to 1, the NSS pin level is determined by the value of the SPI_CTRL1.SSEL bit. 0: Disable software slave device management. 1: Enable software slave device management. *Note: Not used in I^{2}S mode.*
11	SSEL	Internal slave device selection This bit only has meaning when the SPI_CTRL1.SSMEN bit is set. It determines the NSS level, and I/O operations on the NSS pin have no effect. *Note: Not used in I^{2}S mode.*
10	SSOEN	NSS output enable 0: Disable NSS output in master mode, the device can work in multi-master mode. 1: When the device is enabled, enable NSS output in the master mode, the device cannot work in the multi-master device mode. *Note: Not used in I^{2}S mode.*
9	CRCNEXT	Send CRC next 0: The next sent value comes from the send buffer. 1: The next send value comes from the CRC register. *Note: This bit should be set immediately after the last data is written in SPI_DAT register during transmission. This bit should be set immediately after receiving the second-to-last data during reception.*

Bit Field	Name	Description
		Note: Not used in I^{2}S mode.
8	DATFF	Data frame format 0: 8-bit data frame format is used for sending/receiving. 1: 16-bit data frame format is used for sending/receiving. *Note: This bit can only be written when SPI is disabled (SPI_CTRL1.SPIEN = 0), otherwise an error will occur.* *Note: Not used in I^{2}S mode.*
7	LSBFF	Frame format 0: Send MSB first. 1: Send LSB first. *Note: This bit cannot be changed during communication.* *Note: Not used in I^{2}S mode.*
6	MSEL	Master device selection 0: Configure as the slave device. 1: Configure as the master device. *Note: This bit cannot be changed during communication.* *Note: Not used in I^{2}S mode.*
5	CLKPHA	Clock phase 0: Data is sampled on the first clock edge. 1: Data is sampled on the second clock edge. *Note: This bit cannot be changed during communication.* *Note: Not used in I^{2}S mode.*
4	CLKPOL	Clock phase 0: Data is sampled on the first clock edge. 1: Data is sampled on the second clock edge. *Note: This bit cannot be changed during communication.* *Note: Not used in I^{2}S mode.*
3	Reserved	Reserved, the reset value must be maintained.
2:0	BR[2:0]	Baud rate control 000: fPCLK/2 001: fPCLK/4 010: fPCLK/8 011: fPCLK/16 100: fPCLK/32 101: fPCLK/64 110: fPCLK/128

Bit Field	Name	Description
		111: fPCLK/256 *Note: This bit cannot be changed during communication.* *Note: Not used in I²S mode.*

26.5.3 SPI Control Register 2 (SPI_CTRL2)

Address: 0x04

Reset value: 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	RXCPINTEN	CRCNMISEN	RXFFUINTEN	RXFHFINTEN	TXFHEINTEN	FIFOCLR	FIFOEN	SS_POL	ERRINTEN	RNEINTEN	TEINTEN	CRCEN	TDMAEN	RDMAEN	SPIEN
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit Field	Name	Descption
15	Reserved	Reserved, the reset value must be maintained.
14	RXCPINTEN	In receive-only mode, enable the transfer complete interrupt. 0：enable the transfer complete interrupt 1：disable the transfer complete interrupt
13	CRCNMISEN	When NSS becomes invalid, whether the CRC stop calculating immediately 0：As long as there is a clock, the CRC calculation continues. 1：CRC stop calculating immediately when NSS becomes invalid
12	RXFFUINTEN	Enable RX FIFO full interrupt. This bit is only used in FIFO mode. 0：disable RX FIFO full interrupt 1：enable RX FIFO full interrupt
11	RXFHFINTEN	Enable RX FIFO half-full interrupt. This bit is only used in FIFO mode. 0：disable RX FIFO half-full interrupt 1：enable RX FIFO half-full interrupt
10	TXFHEINTEN	Enable TX FIFO half-empty interrupt. This bit is only used in FIFO mode.

Bit Field	Name	Descption
		0：disable TX FIFO half-empty interrupt. 1：Enable TX FIFO half-empty interrupt.
9	FIFOCLR	FIFO clear bit This bit is only used in FIFO mode. 0：not clear FIFO 1：clear FIFO
8	FIFOEN	Enable FIFO 0：disable FIFO 1：enable FIFO
7	SS_POL	NSS polarity control 0：NSS valid level is low. 1：NSS valid level is high。
6	ERRINTEN	Error interrupt enable When an error (SPI_STS.CRCERR, SPI_STS.OVER, SPI_STS.UNDER, SPI_STS.MODERR) is generated, this bit controls whether an interrupt is generated 0: Disable error interrupt. 1: Enable error interrupt.
5	RNEINTEN	Receive buffer non-empty interrupt enable 0: Disable RNE interrupt. 1: Enable RNE interrupt, and generate interrupt request when RNE flag (SPI_STS.RNE) is set to ‘1’.
4	TEINTEN	Transmit buffer empty interrupt enable 0: Disable TE interrupt. 1: Enable TE interrupt, and interrupt request is generated when TE flag (SPI_STS.TE) is set to '1'.
3	CRCEN	Hardware CRC check enable 0: Disable CRC calculation. 1: Enable CRC calculation. *Note: This bit can only be written when SPI is disabled (SPI_CTRL1.SPIEN = 0), otherwise an error will occur.* This bit can only be used in full duplex mode. *Note: Not used in I^{2}S mode.*
2	TDMAEN	Transmit buffer DMA enable

Bit Field	Name	Descption
		When this bit is set, a DMA request is issued as soon as the TE flag (SPI_STS.TE) is set 0: Disable send buffer DMA. 1: Enable send buffer DMA.
1	RDMAEN	Receive buffer DMA enable When this bit is set, a DMA request is issued as soon as the RNE flag (SPI_STS.RNE) is set 0: Disable receive buffer DMA. 1: Enable receive buffer DMA.
0	SPIEN	SPI enable 0: Disable SPI device. 1: Enable the SPI device. *Note: Not used in I^{2}S mode.* *Note: When turning off the SPI device, please follow paragraph 26.3.4 Section's procedure operation.*

26.5.4 SPI Status Register (SPI_STS)

Address: 0x08

Reset value: 0x0101

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			RXTSCP	RXFIFFU	TXFIFFU	RXFIFHF	TXFIFHE	CHSIDE	UNDER	OVER	MODERR	CRCERR	BUSY	RNE	TE
			ro	ro	ro	ro	ro	ro	ro	ro	ro	w0c	ro	ro	ro

Bit Field	Name	Description
15:13	Reserved	Reserved, the reset value must be maintained.。
12	RXTSCP	In receive-only mode, the transfer completion status. 0：The current transfer is not yet completed. 1：The current transfer has been completed.
11	RXFIFFU	RX FIFO full flag This bit is only used in FIFO mode. 0：RX FIFO is not full. 1：RX FIFO is full.
10	TXFIFFU	TX FIFO full flag

Bit Field	Name	Description
		This bit is only used in FIFO mode. 0：TX FIFO is not full 1：TX FIFO is full
9	RXFIFHF	RX FIFO half-full flag This bit is only used in FIFO mode. 0：RX FIFO is not half-full 1：RX FIFO is half-full
8	TXFIFHE	TX FIFO half-empty flag This bit is only used in FIFO mode. 0：TX FIFO is not half-empty 1：TX FIFO is half-empty
7	CHSIDE	Channel 0: The left channel needs to be sent or received. 1: The right channel needs to be sent or received. *Note: not used in SPI mode. No meaning in PCM mode.*
6	UNDER	Underflow flag 0: No underflow occurred. 1: Underflow occurred. This bit is set by hardware and cleared according to the sequence of software operations. In non-FIFO mode of SPI, this flag is not used. In FIFO mode of SPI, attempting to read the RX FIFO when it is empty will set this flag. In I2S slave transmit mode, if the first clock edge of data transfer arrives and new data has not been written to the SPI_DR register, this flag will be set to '1'. *Note: The UNDER flag of SPI2/3 is invalid. The RNE flag can be used to determine whether there is data in rx FIFO*
5	OVER	Overflow flag 0: No overflow error. 1: An overflow error occurred. This bit is set by hardware, and reading the data register first and then reading the status register will clear it. In SPI non-FIFO mode/I2S mode, this flag being set to 1 indicates an overflow in the receive data register. In SPI FIFO mode, attempting to write data to the TX FIFO when it is already full will set this

Bit Field	Name	Description
		flag.
4	MODERR	Mode error 0: No mode error. 1: A mode error occurred. *Note:This bit is set by hardware and cleared according to the sequence of software operations.* For detailed information on software sequences, refer to the error flag section. *It is not used in I2S mode.*
3	CRCERR	CRC error flag 0: The received CRC value matches the value the SPI_CRCRDAT register value. 1: The received CRC value does not match the SPI_CRCRDAT register value. *Note: this bit is set by hardware and cleared by software by writing 0.* *Note: Not used in I^2S mode.*
2	BUSY	Busy flag 0: SPI is not busy. 1: SPI is busy communicating or the send buffer is not empty. This bit is set or reset by hardware.
1	RNE	Receive buffer is not empty 0: The receive buffer is empty. 1: The receive buffer is not empty.
0	TE	The send buffer is empty 0: The send buffer is not empty. 1: The send buffer is empty.

26.5.5 SPI Data Register (SPI_DAT)

Address: 0x0C

Reset value: 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DAT[15:0]															
rw															

Bit Field	Name	Description
15:0	DAT[15:0]	Data register

Bit Field	Name	Description
		Data to be sent or received In FIFO mode, this register is the transmit data register. In non-FIFO mode, this register is the transmit/receive data register. The data register corresponds to two buffers: one for write (transmit buffer); The other is for read (receive buffer).Write operation writes data to transmit buffer; The read operation will return the data in the receive buffer. Note on SPI mode: According to the selection of the data frame format by the SPI_CTRL1.DATFF bit, the data transmitting and receiving can be 8-bit or 16-bit. To ensure correct operation, the data frame format needs to be determined before enabling the SPI. For 8-bit data, the buffer is 8-bit, and only SPI_DAT[7:0] is used when transmitting and receiving. When receiving, SPI_DAT[15:8] is forced to 0. For 16-bit data, the buffer is 16-bit, and the entire data register is used when transmitting and receiving, that is, SPI_DAT[15:0].

26.5.6 SPI Transmit CRC Register（SPI_CRCTDAT）

Address offset：0x10

Reset value：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CRCTDAT[15:0]															
ro															

Bit Field	Name	Description
15:0	CRCTDAT[15:0]	Transmit CRC register When CRC calculation is enabled, CRCTDAT[15:0] contains the CRC value calculated by the bytes sent subsequently. This register is reset when ‘1’ is written to the SPI_CTRL1.CRCEN bit. The CRC calculation uses the polynomial in SPI_CRCPOLY. When the data frame format is set to 8 bits, only the lower 8 bits participate in the calculation and follow the CRC8 standard; when the data frame format is 16 bits, all 16 bits in the register participate in the calculation and follow the CRC16 standard. *Note: reading this register when the BUSY flag (SPI_STS.BUSY) is '1' may read incorrect values.* *Note: not used in I^2s mode.*

26.5.7 SPI Receive CRC Register（SPI_CRCRDAT）

Address offset: 0x14

Reset value: 0x0000

<table>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="16">CRCRDAT[15:0]</td></tr>
<tr><td colspan="16">ro</td></tr>
</table>

Bit Field	Name	Description
15:0	CRCRDAT[15:0]	Receive CRC register When CRC calculation is enabled, CRCRDAT[15:0] will contain the calculated CRC value of subsequent received bytes. This register is reset when ‘1’ is written to the SPI_CTRL1.CRCEN bit. The CRC calculation uses the polynomial in SPI_CRCPOLY. When the data frame format is set to 8 bits, only the lower 8 bits participate in the calculation and follow the CRC8 standard; when the data frame format is 16 bits, all 16 bits in the register participate in the calculation and follow the CRC16 standard. *Note: reading this register when the BUSY flag (SPI_STS.BUSY) is '1' may read incorrect values.* *Note: not used in I^2s mode.*

26.5.8 SPI CRC Polynomial Register（SPI_CRCPOLY）

Address offset：0x18

Reset value：0x0007

<table>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="16">CRCPOLY[15:0]</td></tr>
<tr><td colspan="16">rw</td></tr>
</table>

Bit Field	Name	Description
15:0	CRCPOLY[15:0]	CRC polynomial register This register contains the polynomial used for the CRC calculation. The reset value is 0x0007, other values can be set according to the application. *Note: not used in I^2s mode.*

26.5.9 SPI_I2S Configuration Register（SPI_I2S_CFGR）

Address offset：0x1C

Reset value：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				PCMBYPASS	CLKPOL	PCMSYNC	CHLEN	DATLEN[1:0]		I2SCFG[1:0]		I2SSTD[1:0]		I2SMOD	I2SE
				rw	rw	rw	rw	rw		rw		rw		rw	rw

Bit Field	Name	Description
15:12	Reserved	Reserved, the reset value must be maintained.
11	PCMBYPASS	pcm long for 13bit is bypass 1:no bypass 0:bypass
10	CLKPOL	Static clock polarity 0: I2S clock static state is low level. 1: I2S clock static state is high level. *Note: For correct operation, this bit can only be set when I^2S is disabledf.* *Note: not used in SPI mode.*
9	PCMSYNC	PCM frame synchronization 0: Short frame synchronization. 1: Long frame synchronization. *Note: This bit is only valid when I2SSTD = 11 (used by the PCM standard)。* *Note: not used in SPI mode.*
8	CHLEN	Channel length (number of data bits per audio channel) 0: 16 bits wide; 1: 32 bits wide. *Note: This bit is only valid when* DATLEN = 00， otherwise the channel length is fixed to 32 bits by hardware. *Note: For correct operation, this bit can only be set when I^2S is disabled.* *Note: not used in SPI mode.*

7:6	DATLEN[1:0]	Length of data to be transmitted
		00: 16-bit data length.
		01: 24-bit data length;
		10: 32-bit data length;

		11: Not allowed. *Note: For correct operation, this bit can only be set when I^{2}S is disabled.*
5:4	I2SCFG[1:0]	I^{2}S mode setting 00: Configured as slave transmit mode. 01: Configured as slave receive mode. 10: Configured as Master transmit mode. 11: Configured as Master receive mode. *Note: This bit can only be set when I^{2}S is turned off.* *Note: not used in SPI mode.*
3:2	I2SSTD[1:0]	Selection of I^{2}S standard 00: I^{2}S Philips standard. 01: High byte alignment standard (left alignment). 10: Low byte alignment standard (right alignment). 11: PCM standard. *Note: For correct operation, this bit can only be set when I^{2}S is disabled.* *Not used in SPI mode.*
1	I2SMOD	I^{2}S mode selection 0: Select SPI mode. 1: Select I^{2}S mode. *Note: this bit can only be set when SPI or* I^{2}S *is disabled.*
0	I2SE	I^{2}S enable 0: Disable I^{2}S. 1: Enable I^{2}S. *Note: not used in SPI mode.*

26.5.10 SPI_I^{2}S Prescaler Register (SPI_I2S_PREDIV)

Address offset：0x20

Reset value：0x0002

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				MCLKOEN	ODDEVEN	LDIV[9:0]									
				rw	rw	rw									

Bit Field	Name	Description
15:12	Reserved	Reserved, the reset value must be maintained.

11	MCLKOEN	Master clock output enable 0: Disable master clock output. 1: Enable master clock output. *Note: For correct operation, this bit can only be set when I^{2}S is disabled* *Note: not used in SPI mode.*
10	ODDEVEN	Coefficient prescaler 0: actual frequency division coefficient = LDIV ×2. 1: actual frequency division coefficient = (LDIV × 2) + 1. See Section 26.4.3 for details. *Note: For correct operation, this bit can only be set when I^{2}S is disabled. Use this bit only in I^{2}S master mode.* *Not used in SPI mode.*
9:0	LDIV[9:0]	I^{2}S linear prescaler Setting LDIV [9:0] = 0 or LDIV [9:0] = 1 is prohibited. *Note: For correct operation, this bit can only be set when I^{2}S is disabled. Use this bit only in I^{2}S master mode.* *Not used in SPI mode.*

26.5.11 SPI Receive FIFO（SPI_RX_FIFO）

Address offset：0x24

Reset value0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RXFIFDAT[15:0]															
rw															

Bit Field	Name	Description
15:0	RXFIFDAT[15:0]	In FIFO mode, it is the receive FIFO (read receive FIFO data by accessing this register). In non-FIFO mode, this register is not used.

26.5.12 SPI FIFO Configuration register（SPI_FIFO_NUM）

Address offset：0x28

Reset value：0x0044

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

<table>
<tr><td colspan="9">Reserved</td><td colspan="3">RXFBSNUM[2:0]</td><td>Reserved</td><td colspan="3">TXFBSNUM[2:0]</td></tr>
<tr><td colspan="9"></td><td colspan="3">rw</td><td></td><td colspan="3">rw</td></tr>
</table>

Bit Field	Name	Description
15:7	Reserved	Reserved, the reset value must be maintained.
6:4	RXFBSNUM[2:0]	In FIFO mode, configuration of the RX FIFO half-full threshold When the number of valid data entries in the RX FIFO reaches this configured value, SPI_STS.RXFIFHF will be set to 1. When conducting DMA communication, the number of data items per burst should match the value in this register. This register is not used in non-FIFFO mode.
3	Reserved	Reserved, the reset value must be maintained.
2:0	TXFBSNUM[2:0]	In FIFO mode, configuration of the TX FIFO half-empty threshold When the number of valid data entries in the TX FIFO reaches this configured value, SPI_STS.TXFIFHE will be set to 1. When conducting DMA communication, the number of data items per burst should match the value in this register. This register is not used in non-FIFFO mode.

26.5.13 SPI FIFO Count Register（SPI_FIFO_CNT）

Address offset：0x30

Reset value：0x0000

<table>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="8">Reserved</td><td colspan="4">RXFIFCNT[3:0]</td><td colspan="4">TXFIFCNT[3:0]</td></tr>
<tr><td colspan="8"></td><td colspan="4">ro</td><td colspan="4">ro</td></tr>
</table>

Bit Field	Name	Description
15:8	Reserved	Reserved, the reset value must be maintained.
7:4	RXFIFCNT[3:0]	In FIFO mode, the number of valid data entries in the RX FIFO
3:0	TXFIFCNT[3:0]	In FIFO mode, the number of valid data entries in the TX FIFO

26.5.14 SPI Transfer Number Configuration Register（SPI_TRANS_NUM）

Address offset：0x34

Reset value：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRANSNUM[15:0]															
rw															

Bit Field	Name	Description
15:0	TRANSNUM[15:0]	When in FIFO mode and CRC function is enabled, this register is configured for the total number of data to be transferred. In CPU mode, if the receive-only mode is enabled, by configuring this register, the user does not need to set the CRCNEXT bit by software, as this operation is performed by hardware.

26.5.15 SPI RX Sample Delay Register（SPI_CR3）

Address offset：0x38

Reset value：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												DELAYTIME[3:0]			
												rw			

Bit Field	Name	Description
15:4	Reserved	Reserved, the reset value must be maintained.
3:0	DELAYTIME[3:0]	Configuration of SPI master sampling delay time, after setting the value of this register, the received data will be sampled using the delayed clock. 4'b0000:bypass, no delay processing for the sampling clock. 4'b0001:Sample the data on the MISO port after a delay of 1/2 Tapb_clk cycle. 4'b0010:Sample the data on the MISO port after a delay of 1 Tapb_clk cycle. 4'b0011:Sample the data on the MISO port after a delay of 3/2 Tapb_clk cycle. 4'b0100:Sample the data on the MISO port after a delay of 2 Tapb_clk cycle. 4'b0101:Sample the data on the MISO port after a delay of 5/2 Tapb_clk cycle.

		4'b0110:Sample the data on the MISO port after a delay of 3 Tapb_clk cycle. 4'b0111:Sample the data on the MISO port after a delay of 7/2 Tapb_clk cycle. 4'1000:Sample the data on the MISO port after a delay of 4 Tapb_clk cycle. 4'b1001:Sample the data on the MISO port after a delay of 9/2 Tapb_clk cycle. 4'b1010:Sample the data on the MISO port after a delay of 5 Tapb_clk cycle. 4'b1011:Sample the data on the MISO port after a delay of 11/2 Tapb_clk cycle. 4'b1100:Sample the data on the MISO port after a delay of 6 Tapb_clk cycle. 4'b1101:Sample the data on the MISO port after a delay of 13/2 Tapb_clk cycle. 4'b1110:Sample the data on the MISO port after a delay of 7 Tapb_clk cycle. 4'b1111:Sample the data on the MISO port after a delay of 1 Tapb_clk cycle. Note: Tapb_clk is the APB clock cycle.This register can only be configured in SPI master full-duplex and SPI master receive modes. Configuring this bit in other SPI modes is invalid; this register is not used in I2S mode.

26.5.16 I2S_EXT Control Register（I2S_CTRL2）

Address offset：0x204

Reset value：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									ERRINTEN	RNEINTEN	TEINTEN	Reserved	TDMAEN	RDMAEN	Reserved
									rw	rw	rw		rw	rw	

Bit Field	Name	Description
15:7	Reserved	Reserved, the reset value must be maintained.
6	ERRINTEN	Error interrupt enable When an error is generated, this bit controls whether an interrupt is generated 0: Disable error interrupt. 1: Enable error interrupt.
5	RNEINTEN	Receive buffer non-empty interrupt enable 0: Disable RNE interrupt. 1: Enable RNE interrupt, and generate interrupt request when RNE flag is set to ‘1’.

4	TEINTEN	Transmit buffer empty interrupt enable 0: Disable TE interrupt. 1: Enable TE interrupt, and interrupt request is generated when TE flag is set to '1'.
3	Reserved	Reserved, the reset value must be maintained.
2	TDMAEN	Transmit buffer DMA enable When this bit is set, a DMA request is issued as soon as the TE flag is set 0: Disable Transmi buffer DMA. 1: Enable Transmibuffer DMA.
1	RDMAEN	Receive buffer DMA enable When this bit is set, a DMA request is issued as soon as the RNE flag is set 0: Disable receive buffer DMA. 1: Enable receive buffer DMA.
0	Reserved	Reserved, the reset value must be maintained.

26.5.17 I2S_EXT Status Register （I2S_STS）

Address offset：0x208

Reset value：0x0001

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								CHSIDE	UNDER	OVER	Reserved		BUSY	RNE	TE
								ro	ro	ro			ro	ro	ro

Bit Field	Name	Descption
15:8	Reserved	Reserved, the reset value must be maintained.
7	CHSIDE	Channel 0: The left channel needs to be sent or received. 1: The right channel needs to be sent or received. *Note: not used in SPI mode. No meaning in PCM mode.*
6	UNDER	Underflow flag 0: No underflow occurred. 1: Underflow occurred. In I2S_EXT slave transmit mode, if the first clock edge of data transfer arrives before new data is written to the data register, this bit will be set to 1. This flag is set to 1 by hardware, and reading the I2S_STS status register in

Bit Field	Name	Descption
		software will clear it to 0.
5	OVER	Overflow flag 0: No overflow error. 1: An overflow error occurred. This bit is set by hardware, and software must first read the data register, then read the I2S_STS status register to clear it.
4:3	Reserved	Reserved, the reset value must be maintained.
2	BUSY	Busy flag 0: I2S is not busy. 1: I2S is busy communicating or the transmit buffer is not empty. This bit is set or reset by hardware.
1	RNE	Receive buffer is not empty 0: The receive buffer is empty. 1: The receive buffer is not empty.
0	TE	The transmit buffer is empty 0: The transmit buffer is not empty. 1: The transmit buffer is empty.

26.5.18 I2S_EXT Data Register（I2S_DAT）

Address offset：0x20C

Reset value：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DAT[15:0]															
rw															

Bit Field	Name	Description
15:0	DAT[15:0]	Data register Data to be sent or received The data register corresponds to two buffers: one for write (transmit buffer); The other is for read (receive buffer).Write operation writes data to transmit buffer; The read operation will return the data in the receive buffer.

26.5.19 I2S_EXT Configuration Register（I2S_CFGR）

Address offset：0x21C

Reset value：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved					CLKPOL	PCMSYNC	CHLEN	DATLEN[1:0]		I2SCFG[1:0]		I2SSTD[1:0]		I2SMOD	I2SE
					rw	rw	rw	rw		rw		rw		rw	rw

Bit Field	Name	Description
15:11	Reserved	Reserved, the reset value must be maintained.
10	CLKPOL	Static clock polarity 0: I2S clock static state is low level. 1: I2S clock static state is high level. *Note: For correct operation, this bit can only be set when I2S_CFGR.I2SE=0.* *Note: not used in SPI mode.*
9	PCMSYNC	PCM frame synchronization 0: Short frame synchronization. 1: Long frame synchronization. *Note: This bit is only valid when I2S_CFGR.I2SSTD = 11 (used by the PCM standard).* *Note: not used in SPI mode.*
8	CHLEN	Channel length (number of data bits per audio channel) 0: 16 bits wide; 1: 32 bits wide. Writing to this bit is valid only when I2S_CFGR.DATLEN = 00, otherwise the channel length is fixed to 32 bits by hardware. *Note: For correct operation, this bit can only be set when I2S_CFGR.I2SE=0..* *Note: not used in SPI mode.*
7:6	DATLEN[1:0]	Length of data to be transmitted 00: 16-bit data length. 01: 24-bit data length; 10: 32-bit data length; 11: Not allowed. *Note: For correct operation, this bit can only be set when I2S_CFGR.I2SE=0..*

5:4	I2SCFG[1:0]	I^{2}S mode setting 00: Configured as slave transmit mode. 01: Configured as slave receive mode. 10: Configured as Master transmit mode. 11: Configured as Master receive mode. *Note: this bit can only be configured as 2'b00 or 2'b01.This bit can only be set when I2S_CFGR.I2SE=0.* *Note: not used in SPI mode.*
3:2	I2SSTD[1:0]	Selection of I^{2}S standard 00: I^{2}S Philips standard. 01: High byte alignment standard (left alignment). 10: Low byte alignment standard (right alignment). 11: PCM standard. *Note:this bit can only be set when I2S_CFGR.I2SE=0.* *Note: not used in SPI mode.*
1	I2SMOD	I^{2}S mode selection 0: Select SPI mode. 1: Select I^{2}S mode. *Note: this bit can only be configured as 1’b1. This bit can only be set when I2S_CFGR.I2SE=0.*
0	I2SE	I^{2}S enable 0: Disable I^{2}S. 1: Enable I^{2}S. *Note: not used in SPI mode.*

27 Real-Time Clock (RTC)

27.1 Introduction

- The real-time clock (RTC) is an independent BCD timer/counter.
- Daylight saving time compensation supported by software.
- A periodic automatic programmable wakeup timer.
- Two 32-bit registers contain the seconds, minutes, hours, day (day of week), date (day of month), month, and year.
- Independent 32-bit register contain sub-seconds value.
- Two programmable alarms.
- Two 32-bit registers contain two programmable alarms seconds, minutes, hours, day (day of week), and date (day of month).
- Two 32-bit registers contain two programmable alarms sub-seconds.
- Digital calibration function.
- Reference clock detection: A more accurate external clock source (50 or 60Hz) can be used to improve calendar accuracy.
- Three configurable filtering and internal pull-up tamper detection events.
- Time-Stamp function.
- 20 backup registers, capable of maintaining data in low-power mode.
- Multiple Wakeup sources of Interrupt/Event. These include Alarm A, Alarm B, Wakeup Timer, Time-Stamp.
- After RTC is enabled by the RCC register and voltage remains in the operating range, RTC will not stop timing in any mode (include RUN mode, SLEEP mode, STOP0 mode, VBAT mode and STANDBY mode).
- RTC provides a variety of ways to wakeup from all low-power modes (SLEEP mode, STOP0 mode and STANDBY mode).

27.1.1 Main Features

Table 27-27-1 RTC Feature Support

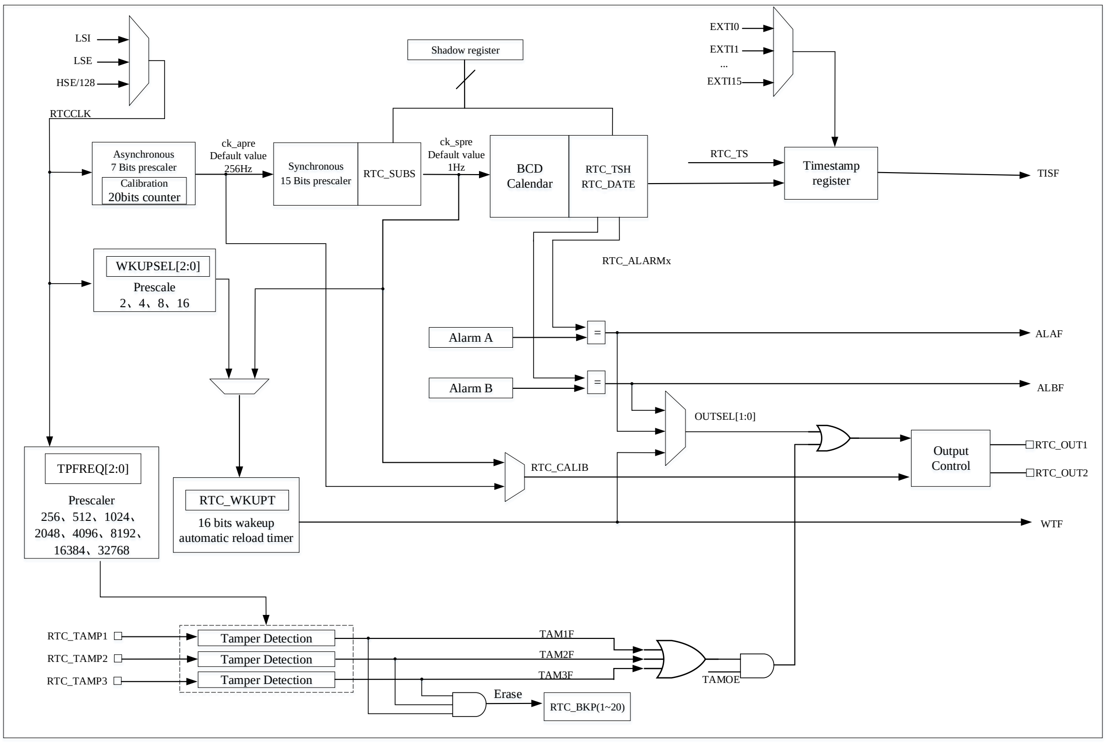
Main Function	Description
Clock	RTC clock can be selected from LSI, LSE or HSE/128 respectively
Reset	The APB interface is reset by the system. Some registers synchronized through APB in the RTC module will be reset. The following registers will be cleared when the system is reset.

Main Function	Description
	● RTC_SUBS ● RTC_TSH ● RTC_DATA ● RTC_INITSTS(some bits) RTC core is reset by backup domain reset. Reset the RTC, and retain the contents of some registers in low-power mode, including: ● RTC_CTRL ● RTC_PRE ● RTC_CALIB ● RTC_SCTRL ● RTC_TSSS, RTC_TST and RTC_TSD ● RTC_TMPCFG ● RTC_WKUPT ● RTC_ALRMASS/RTC_ALRMA ● RTC_ALRMBSS/RTC_ALRMB ● RTC_OPT ● RTC_BKP(1~20)
Calendar	Calendar consists of sub second, second, minute, hour (12 or 24 format), day (day of the week), date, month and year. These data are stored in the shadow register of APB module.
Wakeup Timer	Output 'RTC_OUT' can be configured to transmit wakeup events to GPIO. At the same time, it also can be configured as an interrupt/event to wake up the system from SLEEP, STOP0 ,STANDBY modes.
Alarm	'RTC_OUT' is configured to output to GPIO, and can also wake up the CPU or trigger PWR to wake up from SLEEP, STOP0, and STANDBY modes when a match occurs.
Tamper	The three tamper detection logics are a source of system wake-up. If an tamper event occurs on any of the input lines, when enabled, tamper events also result in the deletion of backup registers. It is also a source for hardware triggering of the LP timer.
Timestamp	GPIO events can trigger the timestamp saving function. It is a source of wake-up from low-power mode. Additionally, tamper events can be a source of timestamp events.
Interrupts/events	Alarm A/Alarm B interrupt/event Wakeup interrupt/event Timestamp interrupt/event Tamper interrupt/event
Backup registers	20 backup registers

27.2 RTC Function Description

27.2.1 RTC Block Diagram

Figure 27-27-1 RTC Block Diagram



RTC includes the following functions:

- Alarm A and Alarm B event/interrupt
- Timestamp event/interrupt
- Tamper event/interrupt
- 20 32-bit backup registers
- RTC output functions
  - 256 Hz or 1Hz clock output (LSE frequency is 32.768 kHz).
  - Alarm clock output (polarity configurable), Alarm A and Alarm B are optional.
  - Auto wakeup output (polarity configurable).
- RTC input functions:
  - Timestamp event detection
  - 50 or 60Hz reference clock input
  - Tamper event detection
- Control PC13/PB2 by configuring output register:

- Set RTC_OPT.TYPE bit to configure open-drain/push-pull output of PC13/PB2

27.2.2 GPIOs Controlled by RTC

Timestamp input come from IOM (mapped to PC13, PC14 and PC15) or EXTI module, if EXTI module is needed to start, please refer to the timestamp trigger source selection register (EXTI_TS_SEL) for details.

RTC_OUT (Alarm, Wakeup event, Tamper event or calibration output (256Hz or 1Hz)) is mapped to PC13 and PB2. Regardless of the configuration of PC13 and PB2 GPIO pins, the pin configuration of PC13 and PB2 is controlled by RTC as output.

Note: When RTC_CTRL.OUTSEL[1:0] is not configured to 0, even if RTC_CTRL.OUT2EN is configured to 0 (i.e., the output of the RTC_OUT2 pin is prohibited), at this time, the PB2 pin can not be used for common pins or peripheral multiplexing functions. For details, please refer to the errata manual.

The PC13 pin is used as the TAMPER1 tamper detection pin, the PA0 pin is used as the TAMPER2 tamper detection pin, and the PE6 pin is used as the TAMPER3 tamper detection pin.

PA1 or PB15 can be used as the RTC_REFCLKIN reference clock input pin.

27.2.3 RTC Register Write Protection

The PWR_CTRL.DBKP bit (refer to the power control register (PWR_CTRL)) is cleared by default, so PWR_CTRL.DBKP must be set to 1 to enable RTC register write functionality. Once the backup domain is reset, all RTC write-protected registers will have write protection enabled. All RTC write-protected registers need to be unlocked by following the steps below:

- Write “0xCA” into RTC_WRP register.
- Write “0x53” into RTC_WRP register.

After unlocking these registers, writing an incorrect key or resetting the RTC by software or power cycling will re-enable the write protection.The unlocking mechanism only checks the write operation to the RTC_WRP register. During the unlocking process, before unlocking, and after unlocking, writing to other registers will not affect the unlocking result.

After the backup domain reset, all configurable RTC registers and backup registers RTC_BKP (20) are write-protected.

27.2.4 RTC Clock And Prescaler

RTC clock source:

- LSE clock
- LSI clock
- HSE/128 clock

To reduce power consumption, the prescaler is divided into an asynchronous prescaler and a synchronous prescaler.

If both prescalers are used simultaneously, it is recommended to set the value of the asynchronous prescaler as large as possible.

By configuring RTC_PRE.DIVA[6:0] and RTC_PRE.DIVS[14:0], the user can obtain a ck_spre clock of 1Hz.

- A 7-bit asynchronous prescaler which is given by RTC_PRE.DIVA[6:0] bits
- A 15-bit synchronous prescaler which is given by RTC_PRE.DIVS[14:0] bits

The formula for f_{ck_apre} and f_{ck_spre} are given below:

$$f_{\text{ck\_apre}} = \frac{f_{RTCCLK}}{RTC\_PRE.DIVA[6:0]+1}$$

$$f_{\text{ck\_spre}} = \frac{f_{RTCCLK}}{(RTC\_PRE.DIVS[14:0]+1)*(RTC\_PRE.DIVA[6:0]+1)}$$

The ck_apre clock is used to driven RTC_SUBS sub-second down counter. When it reaches 0, RTC_SUBS is reloaded with the value of RTC_PRE.DIVS[14:0].

27.2.5 RTC Calendar

There are three shadow registers, they are RTC_DATE, RTC_TSH and RTC_SUBS. The RTC time and date registers can be accessed through the shadow registers. It is also possible to access them directly to avoid the synchronization waiting time. The three shadow registers are as follow:

- RTC_DATE: set and read date
- RTC_TSH: set and read time
- RTC_SUBS: read sub-second

Every two RTCCLK cycles, the current calendar value is copied to the shadow registers, and the RTC_INITSTS.RSYF bit is set to 1. This process is not executed in low-power (stop and standby) modes. When exiting these modes, the shadow registers update their values after 2 RTCCLK cycles.

By default, when user try to access the calendar register, it will access the contents of the shadow register instead. User can access the calendar register directly by setting the RTC_CTRL.BYPS bit.

When RTC_CTRL.BYPS=0, calendar values are from shadow registers, when reading RTC_SUBS, RTC_TSH or RTC_DATE register, it is necessary to make ensure the frequency of APB1 clock (f_{APB1}) is at least 7 times the frequency of RTC clock (f_{RTCCLK}), and APB1 clock frequency lower than RTC clock frequency is not allowed in any case. System reset will reset shadow registers.

27.2.6 Calendar Initialization And Configuration

The value of prescaler and calendar can be initialized by the following steps:

- Set the RTC_INITSTS.INITM bit to 1 to enter the initialization mode, and then wait for the RTC_INITSTS.INITF bit to be set to 1.
- Set RTC_PRE.DIVS[14:0] and RTC_PRE.DIVA[6:0] value.

- Write the initial calendar values include time and date into the shadow registers (RTC_TSH and RTC_DATE) and configure the time format (12 or 24 hours) through the RTC_CTRL.HFMT bit.
- Exit initialization mode by clearing the RTC_INITSTS.INITM bit.

The values of calendar counter will automatically loaded from shadow registers after 4 RTCCLK clock cycles, then the calendar counter restarts.

27.2.7 Calendar Reading

1. Reading calendar value when RTC_CTRL.BYPS=0

Calendar value is read from shadow registers if RTC_CTRL.BYPS=0. In order to read RTC calendar registers (RTC_SUBS, RTC_TSH and RTC_DATE) correctly, APB1 clock frequency must be set greater than 7 times of RTC clock frequency. In any case, APB1 clock frequency must not be less than RTC clock frequency.

If APB1 clock frequency is not equal to or greater than 7 times of RTC clock frequency, refer to the following process to read calendar value.

- Read the data of RTC_SUBS, RTC_TSH and RTC_DATE twice.
- Compare the data read on two occasions, if they are equal, the read data can be considered correct; if they are not equal, read the data for the third time.
- The third time read data can be considered correctly.
- To ensure that the calendar is read from the same time point, the value in the RTC_DATE register is locked when reading RTC_SUBS or RTC_TSH. Until RTC_DATE is read, it will not be updated in real-time, so after reading RTC_SUBS or RTC_TSH, RTC_DATE needs to be read again.

Shadow registers (RTC_SUBS, RTC_TSH and RTC_DATE) are updated every two RTCCLK cycles. If user want to read calendar value in a short time(less than two RTCCLK cycles), RTC_INITSTS.RSYF bit must be cleared by software after the first time read.

In some cases, it is necessary to wait until RTC_INITSTS.RSYF bit is set to 1 before read calendar value.

- After waking up from the low power modes (STANDBY mode), clear RTC_INITSTS.RSYF bit, then wait for the RTC_INITSTS.RSYF bit to be set to 1 again..
- System reset.
- Calendar complete initialization.
- Calendar complete synchronization.

2. Reading calendar value when RTC_CTRL.BYPS=1

If RTC_CTRL.BYPS=1, read the calendar value directly from the calendar counter. The advantage of this configuration is that there is no delay in reading the calendar value after waking up from low-power mode. The disadvantage is that the data from RTC_SUBS, RTC_TSH, and RTC_DATE may not be from the same moment.

To ensure the correctness of read calendar value, it is necessary to read RTC_SUBS, RTC_TSH and RTC_DATE twice, the data read on two occasions should then be compared, and if they are equal, the read data is considered correct.

27.2.8 Calibration Clock Output

When RTC_CTRL.COEN is set to 1, the PC13 or PB2 pins will output the calibration clock. If RTC_CTRL.CALOSEL=0 and RTC_PRE.DIVA[6:0]=0x7F, the frequency of RTC_CALIB is fRTCCLK / RTC_PRE.DIVA[6:0].When the frequency of RTCCLK is 32.768 kHz, the calibration output is 256Hz. Due to slight jitter on the falling edge, it is recommended to use the rising edge.

When RTC_CTRL.CALOSEL=1 and "RTC_PRE.DIVS[14:0]+1" is a non-zero integer multiple of 256, the frequency of RTC_CALIB is given by the formula f_{RTCCLK}/(256 * (DIVA+1)). When the frequency of RTCCLK is 32.768 kHz and RTC_PRE.DIVA[6:0]=0x7F, the calibration output is 1Hz.

Notes:

(1) When selecting RTC_CALIB as the output, the RTC_OUT pin (PC13/PB2) is automatically configured as an output.

(2) The duty cycle of the RTC_OUT output at 256Hz or 1Hz is fixed at 50% (+-20%).

27.2.9 Programmable Alarm

RTC has 2 programmable alarms: Alarm A and Alarm B.

RTC alarm can be enabled or disabled by configuring RTC_CTRL.ALxEN bit. If the Alarm value matches the calendar value, the RTC_INITSTS.ALxF flag is set to 1. If RTC_CTRL.ALxIEN is enabled, any calendar field can be selected to trigger the alarm interrupt.

Alarm Output: After configuring RTC_CTRL.OUTSEL[1:0], Alarm A or Alarm B can be mapped to the RTC_ALxRM output, and the output polarity can be configured by the RTC_CTRL.OPOL bit.

Note: When the second field is selected (RTC_ALARMx.MASK1 bit reset), RTC_PRE.DIVS[14:0] must be greater than 3 to ensure correct operation.

27.2.10 Alarm Configuration

Alarm A and Alarm B should be configured in the following below:

- Disable Alarm A/Alarm B by clearing RTC_CTRL.ALAEN/RTC_CTRL.ALBEN bit.
- Configure the Alarm x registers (RTC_ALRMxSS/RTC_ALARMx)
- Enable the Alarm A/Alarm B interrupt by setting the RTC_CTRL.ALAIEN/RTC_CTRL.ALBIEN bit to 1 (add this step as needed).
- Enable Alarm A/Alarm B by setting the RTC_CTRL.ALAEN/RTC_CTRL.ALBEN bit to 1.

27.2.11 Alarm Output

When RTC_CTRL.OUTSEL[1:0] != 0, the RTC_ALARM output function is enabled. Depending on the value of RTC_CTRL.OUTSEL[1:0], select Alarm A output, Alarm B output, or wakeup output.

The RTC_CTRL.OPOL bit controls the polarity of Alarm A, Alarm B, or wakeup output.

The RTC_OPT.TYPE bit controls the RTC_ALARM pin open-drain or push-pull output.

When selecting RTC_ALARM output, the RTC_OUT pin (PC13/PB2) is automatically configured as an output.

27.2.12 Periodic Automatic Wakeup

A 16-bit programmable auto-reload counter can generate a periodic wakeup flag upon reaching 0. It can also extend the range of the wakeup timer to 17 bits. Enabling the periodic auto-wakeup function can be done by setting RTC_CTRL.WTEN

There are two wake-up input clock sources can be selected:

- RTC clock (RTCCLK) divided by 2/4/8/16.

  Assume RTCCLK comes from LSE (32.768KHz), wake-up interrupt period can be configured range from 122us to 32s under the resolution down to 61us.

- Internal clock ck_spre.

  Assume ck_spre frequency is 1Hz, the available wake-up time range from 2s to 18h, and the resolution is 1 second.

  - When RTC_CTRL.WKUPSEL [2:0] = 10x, the period is range from 1s to 18h.
  - When RTC_CTRL.WKUPSEL [2:0] = 11x, the period is range from 18h to 36h.

When the RTC_CTRL.WTEN bit is set to 1, the countdown counter is running, and when it reaches 0, the RTC_INITSTS.WTF bit is set to 1. By setting the RTC_CTRL.WTIEN bit to 1, the device can exit low-power mode when the periodic wakeup interrupt is triggered and enabled.

Periodic Wakeup Output: When RTC_CTRL.OUTSEL[1:0] selects the periodic wakeup, the RTC_OUT pin (PC13/PB2) is automatically configured as an output, and the output polarity can be configured by the RTC_CTRL.OPOL bit.

27.2.13 Wakeup Timer Configuration

The configuration of the automatic reload value for the wakeup timer is as follows:

- Disable the wakeup timer by clearing RTC_CTRL.WTEN, then wait for the RTC_INITSTS.WTWF flag to be set to 1.
- Select the wakeup timer clock by setting RTC_CTRL.WKUPSEL[2:0].
- Configure the wake-up automatic reload value by setting RTC_WKUPT.WKUPT[15:0] bits.
- Enable Wakeup interrupt by setting RTC_CTRL.WTIEN bit(this step can be selected as needed)
- Enable wakeup timer by setting RTC_CTRL.WTEN bit

27.2.14 Timestamp Function

Timestamp can be enabled by setting RTC_CTRL.TSEN bit to 1. When a timestamp event is detected on the RTC_TS pin, the calendar values of the event will be stored in the timestamp register (RTC_TSSS, RTC_TST, RTC_TSD), and RTC_INITSTS.TISF is set to 1. If RTC_CTRL.TSIEN is set to 1, a timestamp event can generate an interrupt. If a new timestamp event is detected while RTC_INITSTS.TISF is already set to 1, the hardware will set the RTC_INITSTS.TISOVF flag to 1. The timestamp registers (RTC_TST and RTC_TSD) will continue to hold the values of the previous event, meaning that the data in the timestamp registers (RTC_TST and RTC_TSD) will not change when RTC_INITSTS.TISF=1.

After the timestamp event caused by the synchronization process occurs again, RTC_INITSTS.TISF is set to 1 in 2 RTC_CLK cycles. There is no delay in the generation of RTC_INITSTS.TISOVF. This means that if two timestamp events are very close, this can cause RTC_INITSTS.TISOVF to be "1" and RTC_INITSTS.TISF to be "0". Therefore, after detecting RTC_INITSTS.TISF as "1", check the RTC_INITSTS.TISOVF bit. When RTC_TMPCFG.TPTS bit is set to 1, tamper events can trigger timestamp events.

If timestamp events are enabled, the timestamp will capture the read calendar in the timestamp registers.Timestamp events can be generated on any of the 16 GPIO ports selected by EXTI. The GPIO pins in each port are selected by setting the corresponding EXTI_TS_SEL.TSSEL[3:0] bits.

27.2.15 Tamper Detection

There are three tamper detection pins, RTC_TAMP1 pin is PC13, RTC_TAMP2 pin is PA0, and RTC_TAMP3 pin is PE6. The RTC_TAMPx pins can be used as input pins for tamper event detection. There are two detection modes, edge detection mode and level detection mode with configurable filtering function.

When a RTC_TAMPx event is detected, if RTC_TMPCFG.TPxNOE=0, the RTC_BKP(1~20) registers will be cleared.

Tamper Detection Initialization

There are three tamper detection pins, each pin can be configured independently. Users need to configure tamper detection before setting the RTC_TMPCFG.TPxEN bit. When tamper detection is enabled and a tamper event is detected, if RTC_TMPCFG.TPxINTEN is set to 1, the tamper event can generate an interrupt and the RTC_INITSTS.TAMxF bit will be set to 1.

When the RTC_INITSTS.TAMxF bit is set to 1, new tamper events on the same pin cannot be detected.

Timestamp on tamper event

When RTC_TMPCFG.TPTS bit is set to 1, any tamper event can trigger a timestamp event, and the RTC_INITSTS.TISF bit and RTC_INITSTS.TISOVF bit will be set for normal timestamp events.

Edege detection on tamper inputs

When the RTC_TMPCFG.TPFLT[1:0] bit is set to 0, tamper detection is set to edge detection, with the rising or falling edge controlled by the RTC_TMPCFG.TPxTRG bit. When the corresponding edge is detected, the RTC_TAMPx pin will generate a tamper detection event.

Since RTC_BKP(120) can be reset when a tamper event is detected, it is necessary to ensure that tamper event detection and writing to RTC_BKP(120) do not occur simultaneously. It is recommended to enable the tamper detection function after writing to RTC_BKP(1~20).

Note: When selecting edge triggering, external pull-up or pull-down is required

Level detection with filtering on RTC_TAMPx input

When RTC_TMPCFG.TPFLT[1:0] bits are set to 1/2/3, tamper detection is set to level detection. The value of RTC_TMPCFG.TPFLT[1:0] determines the number of samples.

Before each sampling, pre-charging can be performed through the internal pull-up resistor of the tamper pin, and the pre-charging time is controlled by the RTC_TMPCFG.TPPRCH[1:0] bits. When RTC_TMPCFG.TPPUDIS is set to 1, pre-charging will be disabled.

Using RTC_TMPCFG.TPFREQ[2:0] to determine the sampling frequency of level detection can optimize the best balance between tamper detection delay and pull-up power consumption.

27.2.16 Daylight Saving Time Configuration

Daylight saving time function can be controlled through RTC_CTRL.SU1H, RTC_CTRL.AD1H, and RTC_CTRL.BAKP bits. Calendar will subtract one hour when set RTC_CTRL.SU1H bit to 1, and add one hour when set RTC_CTRL.AD1H to 1. RTC_CTRL.BAKP can be used to memorize this adjustment.

Setting the RTC_CTRL.SU1H bit to 1 will subtract one hour for the calendar, while setting the RTC_CTRL.AD1H to 1 will add one hour. The RTC_CTRL.BAKP bit can be used to remember or not remember this adjustment.

27.2.17 RTC Sub-Second Register Shift Operation

When the value of calendar has a sub-second deviation compared to the external precision clock, the shift function can be used to improve the precision of calendar.

The calendar can be controlled to delay or advance by 1 second using the RTC_SCTRL.AD1S and RTC_SCTRL.SUBF[14:0] bits. The adjustment resolution is 1/(RTC_PRE.DIVS[14:0]+1), where a higher value of RTC_PRE.DIVS[14:0] indicates a higher resolution. To keep the synchronized prescaler output at 1Hz, a higher value of RTC_PRE.DIVS[14:0] means a lower value of RTC_PRE.DIVA[6:0], resulting in higher power consumption

Note: Before starting the shift operation, the user must check if the RTC_SUBS.SS[15] bit is 0

Whenever writing to the RTC_SCTRL register, the hardware sets the RTC_INITSTS.SHOPF flag, indicating that the shift operation is in a suspended state. Once the shift operation is completed, this bit is cleared by the hardware.

27.2.18 RTC Digital Clock Precision Calibration

Digital precision calibration is achieved by adjusting the number of RTC clock pulses in the calibration period. Digital precision calibration resolution is 0.954 PPM with the range from -487.1 PPM to +488.5 PPM.

When the input frequency is 32768 Hz, calibration period can be configured as $2^{20}/2^{19}/2^{18}$ RTCCLK cycles or 32/16/8 seconds. The precision calibration register (RTC_CALIB) indicates that there has RTC_CALIB.CM[8:0] RTCCLK clock cycles will be reduced during the specified period.

The value of RTC_CALIB.CM[8:0] represents the number of RTCCLK pulses to be reduced within a specified period. RTC_CALIB.CP can be used to increase by 488.5PPM, where every 2^{11} RTCCLK periods will insert an RTCCLK pulse

When using RTC_CALIB.CM[8:0] and RTC_CALIB.CP in combination, it can increase cycles range from -511 to +512 RTCCLK cycles, and the calibration range from -487.1 ppm to +488.5 ppm, with the resolution is about 0.954 ppm.

The valid calibrated frequency (f_{CAL}) can be calculated by using the formula given below:

$$f_{CAL} = f_{RTCCLK} * (1 + \frac{RTC\_CALIB.CP*512-RTC\_CALIB.CM[8:0]}{2^{n}+ RTC\_CALIB.CM[8:0] - RTC\_CALIB.CP * 512})$$

Note:n=20/19/18

Calibrated when RTC_PRE .DIVA[6:0]<3

When the asynchronous prescaler value (RTC_PRE.DIVA[6:0]) is less than 3, the RTC_CALIB.CP cannot be programmed to 1, and RTC_CALIB.CP value will be ignored if the it has been set to 1.

Assume the frequency of RTCCLK is 32768Hz, when RTC_PRE .DIVA[6:0]<3, the value of RTC_PRE.DIVS[14:0] should be decrease.:

- When RTC_PRE .DIVA[6:0] =2, RTC_PRE.DIVS[14:0]=8189.
- When RTC_PRE .DIVA[6:0] =1, RTC_PRE.DIVS[14:0]=16379.
- When RTC_PRE .DIVA[6:0] =0, RTC_PRE.DIVS[14:0]=32759.

The valid calibrated frequency (f_{CAL}) can be calculated by using the formula given below:

$$f_{CAL} = f_{RTCCLK} * \left(1 + \frac{256-RTC\_CALIB.CM[8:0]}{2^{n}+ RTC\_CALIB.CM[8:0]- 265}\right)$$

Verify RTC calibration

RTC outputs waveform(1Hz) for measuring and verifying RTC precision.

Up to 2 RTCCLK cycles measurement error may occur when measure the RTC frequency in a limit measurement period. If the measurement period is the same as calibration period, the error can be eliminated.

- The calibration period is 32 seconds (default).

  Using an accurate 32-second period to measure the 1Hz calibration output can ensure that the measurement error is within 0.447ppm (0.5 RTCCLK cycles within 32 seconds).

- The calibration period is 16 seconds.

  Using an accurate 16-second period to measure the 1Hz calibration output can ensure that the measurement error is within 0.954ppm (0.5 RTCCLK cycles within 16 seconds).

- The calibration period is 8 seconds.

  Using an accurate 8-second period to measure the 1Hz calibration output can ensure that the measurement error is within 1.907ppm (0.5 RTCCLK cycles within 8 seconds).

Dynamic recalibration

When RTC_INITSTS.INITF=0, RTC_CALIB register can be updated through following steps:

- Wait RTC_INITSTS.RECPF=0.
- A new value is written to the RTC_CALIB, then RTC_INITSTS.RECPF is automatically set to 1.
- The new calibration settings will take effect within 3 ck_apre cycles after a data write to the RTC_CALIB.

27.2.19 RTC Low Power Mode

The operating state of RTC in low power mode:

Lower Power Mode	RTC Operating State	Exit Low Power Mode
SLEEP	Normal operation	Alarm A, Alarm B, Periodic Wakeup, tamper enent and Timestamp event
STOP0	Normal operation when RTC clock source is LSE or LSI	Alarm A, Alarm B, Periodic Wakeup, tamper enent and Timestamp event
STANDBY	Normal operation when RTC clock source is LSE or LSI	Alarm A, Alarm B, Periodic Wakeup, tamper enent and Timestamp event

27.2.20 RTC_OUT1 (PC13) And RTC_OUT2 (PB2) Mapping

Table 27-2 RTC OUT Mapping

RTC_CTRL				RTC_OUT1(PC13)	RTC_OUT2(PB2)
TAMPOE	OUTSEL[1:0]	COEN	OUT2EN		
0	00	1	0	CALIB	-
1	00	Don’t care		TAMPER	-
0	01	Don’t care		ALARMA	-
0	10	Don’t care		ALARMB	-
0	11	Don’t care		WAKEUP	
1	01/10/11	Don’t care		TAMPER_ALARM[1]	-
0	00	1	1	-	CALIB
1	00	0		-	TAMER
0	01	0		-	ALARMA
0	10	0		-	ALARMB
0	11	0		-	WAKEUP
1	01/10/11	0		-	TAMPER_ALARM[1]
1	00	1		TAMPER	CALIB
0	01	1		ALARMA	CALIB
0	10	1		ALARMB	CALIB
0	11	1		WAKUP	CALIB
1	01/10/11	1		TAMPER_ALARM[1]	CALIB

1. "TAMPER_ALARM" indicates the output of TAMPER and either ALARMA, ALARMB, or WAKEUP.

27.3 RTC Registers

27.3.1 RTC Register Overview

Table 27-3 RTC Register Overview

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
000h	RTC_INITSTS	Reserved															RECPF	TAM3F	TAM2F	TAM1F	TISOVF	TISF	WTF	ALBF	ALAF	INITM	INITF	RSYF	INITSF	SHOPF	WTWF	ALBWF	ALAWF
	Reset Value																0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
004h	RTC_CTRL	Reserved						TAMPOE	OUT2EN	COEN	OUTSEL[1:0]		OPOL	CALOSEL	BAKP	SU1H	AD1H	TSIEN	WTIEN	ALBIEN	ALAIEN	TSEN	WTEN	ALBEN	ALAEN	Reserved	HFMT	BYPS	REFCLKEN	TEDGE	WKUPSEL[2:0]		
	Reset Value							0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0
008h	RTC_TSH	Reserved									APM	HOT[1:0]		HOU[3:0]				Reserved	MIT[2:0]			MIU[3:0]				Reserved	SCT[2:0]			SCU[3:0]			
	Reset Value										0	0	0	0	0	0	0		0	0	0	0	0	0	0		0	0	0	0	0	0	0
00Ch	RTC_DATE	Reserved								YRT[3:0]				YRU[3:0]				WDU[2:0]			MOT	MOU[3:0]				Reserved		DAT[1:0]		DAU[3:0]			
	Reset Value									0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1			0	0	0	0	0	1
010h	RTC_WRP	Reserved																								PKEY[7:0]							
	Reset Value																									0	0	0	0	0	0	0	0
014h	RTC_SCTRL	ADIS	Reserved																SUBF[14:0]														
	Reset Value	0																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
018h	RTC_SUBS	Reserved																SS[15:0]															
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
01Ch	RTC_TST	Reserved									APM	HOT[1:0]		HOU[3:0]				Reserved	MIT[2:0]			MIU[3:0]				Reserved	SET[2:0]			SEU[3:0]			
	Reset Value										0	0	0	0	0	0	0		0	0	0	0	0	0	0		0	0	0	0	0	0	0
020h	RTC_ALARMA	MASK4	WKDSEL	DTT[1:0]		DTU[3:0]				MASK3	APM	HOT[1:0]		HOU[3:0]				MASK2	MIT[2:0]			MIU[3:0]				MASK1	SET[2:0]			SEU[3:0]			
	Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
024h	RTC_PRE	Reserved									DIVA[6:0]							Reserved	DIVS[14:0]														
	Reset Value										1	1	1	1	1	1	1		0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
028h	RTC_ALARMB	MASK4	WKDSEL	DTT[1:0]		DTU[3:0]				MASK3	APM	HOT[1:0]		HOU[3:0]				MASK2	MIT[2:0]			MIU[3:0]				MASK1	SET[2:0]			SEU[3:0]			
	Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
02Ch	RTC_WKUPT	Reserved																WKUPT[15:0]															
	Reset Value																	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
030h	RTC_TMPCFG	Reserved				TP3MF	TP3NOE	TP3INTEN	TP2MF	TP2NOE	TP2INTEN	TP1MF	TP1NOE	TP1INTEN	TPPUDIS	TPPRCH[1:0]		Reserved		TPFLT[1:0]		Reserved	TPFREQ[2:0]			TPTS	TP3TRG	TP3EN	TP2TRG	TP2EN	TPINTEN	TP1TRG	TP1EN
	Reset Value					0	0	0	0	0	0	0	0	0	0	0	0			0	0		0	0	0	0	0	0	0	0	0	0	0
034h	RTC_ALRMASS	Reserved												MASKSSA[3:0]				Reserved	SSV[14:0]														
	Reset Value													0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
038h	RTC_OPT	Reserved																															TYPE
	Reset Value																																0
03Ch	RTC_ALRMBSS	Reserved												MASKSSB[3:0]				Reserved	SSV[14:0]														
	Reset Value													0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
040h	RTC_CALIB	Reserved																				CP	CW8	CW16	CM[8:0]								
	Reset Value																					0	0	0	0	0	0	0	0	0	0	0	0
044h	RTC_TSSS	Reserved																SSE[15:0]															
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
048h	RTC_TSD	Reserved								YRT[3:0]				YRU[3:0]				MOT	WDU[2:0]			MOU[3:0]				Reserved		DAT[1:0]		DAU[3:0]			
	Reset Value									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			0	0	0	0	0	0
04Ch ~ 098h	RTC_BKPx	BF[31:0]																															
	Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

27.3.2 RTC Initial Status Register (RTC_INITSTS)

Address offset: 0x00

Reset value: 0x0000 0007

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															RECPF
															r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAM3F	TAM2F	TAM1F	TISOVF	TISF	WTF	ALBF	ALAF	INITM	INITF	RSYF	INITSF	SHOPF	WTWF	ALBWF	ALAWF
rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rw	r	rc_w0	r	r	r	r	r

Bit Field	Name	Description
31:17	Reserved	Reserved, the reset value must be maintained
16	RECPF	Recalibration pending flag The RECPF status flag is automatically set to ‘1’ when software writes to the RTC_CALIB register, indicating that the RTC_CALIB register is blocked. After the new calibration settings are processed, this bit returns to ‘0’.
15	TAM3F	RTC_TAMP3 detection flag When a tamper event is detected on the RTC_TAMP3 input, the hardware sets this flag. Writing 0 through software clears it.
14	TAM2F	RTC_TAMP2 detection flag. When a tamper event is detected on the RTC_TAMP2 input, the hardware sets this flag. Writing 0 through software clears it
13	TAM1F	RTC_TAMP1 detection flag. When a tamper event is detected on the RTC_TAMP1 input, the hardware sets this flag. Writing 0 through software clears it.
12	TISOVF	The time-stamp overflow flag When a timestamp event occurs while the TISF bit has already been set to 1, the hardware sets this flag to 1. It is recommended to check and clear the TISOVF bit after clearing the TISF bit. Otherwise, if the timestamp event occurs just before clearing the TISF bit, the overflow event might be missed
11	TISF	Time-stamp flag This flag is set to ‘1’ by hardware when a time-stamp event occurs. This flag can be cleared by software by writing 0
10	WTF	Wake up timer flag When the wake-up auto-reload counter reaches 0, the hardware sets this flag. This flag is cleared by software by writing 0. Before the WTF is set to 1 again, this flag must be cleared by software for at least 1.5 RTCCLK cycles
9	ALBF	Alarm B flag This flag is set to ‘1’ by hardware when the time/date registers(RTC_TSH and RTC_DATE) value match the Alarm B register(RTC_ALARMB) values. This flag can be cleared by software by writing 0

Bit Field	Name	Description
8	ALAF	Alarm A flag This flag is set to ‘1’ by hardware when the time/date registers(RTC_TSH and RTC_DATE) value match the Alarm A register(RTC_ALARMA) values. This flag can be cleared by software by writing 0
7	INITM	Enter Initialization mode 0: Free operating mode 1: Enter initialization mode and set calendar time value, date value, and prescale value.
6	INITF	Initialization flag RTC is in initialization state when this bit is ‘1’, and calendar time, date and prescale value can be updated. 0: Calendar register can not be updated 1: Calendar register can be updated
5	RSYF	Register synchronization flag This flag is set to ‘1’ by hardware when the calendar value are copied into the shadow registers. This bit is cleared by hardware when in initialization mode, while a shift operation is pending (SHOPF = 1), or when in bypass shadow register mode (RTC_CTRL.BYPS = 1). This bit can also be cleared by software. It is cleared either by software or by hardware in initialization mode. 0: Calendar shadow register not yet synchronized 1: Calendar shadow register synchronized
4	INITSF	Initialization status flag This flag is set to ‘1’ by hardware when the calendar year field is different from 0 (which is the RTC domain reset state). 0: Calendar has not been initialized 1: Calendar has been initialized
3	SHOPF	Shift operation pending flag This flag is set to ‘1’ by hardware as soon as a shift operation is initiated by a write to the RTC_SCTRL register. It is cleared by hardware when the corresponding shift operation has been completed, note that writing to the SHOPF bit has no effect. 0: No shift operation is pending

Bit Field	Name	Description
		1: A shift operation is pending
2	WTWF	Wakeup timer write flag 0: Wakeup timer configuration update is not allowed 1: Wakeup timer configuration update is allowed
1	ALBWF	Alarm B write flag This flag is set to ‘1’ by hardware when Alarm B values can be changed, after the RTC_CTRL.ALBEN bit has been set to 0. 0: Alarm B update is not allowed 1: Alarm B update is allowed
0	ALAWF	Alarm A write flag. This flag is set to ‘1’ by hardware when Alarm A values can be changed, after the RTC_CTRL.ALAEN bit has been set to 0. 0: Alarm A update is not allowed 1: Alarm A update is allowed

27.3.3 RTC Control Register (RTC_CTRL)

Address offset: 0x04

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved						TAMPOE	OUT2EN	COEN	OUTSEL[1:0]		OPOL	CALOSEL	BAKP	SU1H	AD1H
						rw	rw	rw	rw		rw	rw	rw	w	w

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSIEN	WTIEN	ALBIEN	ALAIEN	TSEN	WTEN	ALBEN	ALAEN	Reserved	HFMT	BYPS	REF	TEDGE	WKUPSEL[2:0]		
rw	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw	rw	rw		

Bit Field	Name	Description
31:26	Reserved	Reserved, the reset value must be maintained
25	TAMPOE	Tamper output enable 0: Disable Tamper output

Bit Field	Name	Description
		1: Enable Tamper output
24	OUT2EN	RTC_OUT2 output enable 0: Disable RTC_OUT2 output 1: Enable RTC_OUT2 output
23	COEN	Calibration output enable 0: Disable calibration output 1: Enable calibration output
22:21	OUTSEL[1:0]	Output selection These bits are used to select the alarm/wakeup output 00: Disable output 01: Enable Alarm A output 10: Enable Alarm B output 11: Enable Wakeup output
20	OPOL	Output polarity bit This bit is used to configure the polarity of output. 0: When the ALAF/ALBF/WTF flag is set to 1 (depending on OUTSEL [1:0]), the pin outputs a high level 1: When the ALAF/ALBF/WTF flag is set to 1 (depending on OUTSEL [1:0]), the pin outputs a low level
19	CALOSEL	Calibration output selection When COEN=1, this bit selects which signal is output on RTC_CALIB Under the conditions of RTCCLK at 32.768 kHz and default prescaler values (RTC_PRE.DIVA[6:0]=127 and RTC_PRE.DIVS[14:0]=255), these frequencies are valid. 0: Calibration output is 256 Hz 1: Calibration output is 1 Hz
18	BAKP	This bit can be written by the user to remember if daylight saving time changes have been applied.
17	SU1H	Subtract 1 hour (winter time change) 1 hour will be subtracted to the calendar time when the current hour value is not 0. This bit

Bit Field	Name	Description
		is always read as 0. Setting this bit is invalid when the current hour is 0 0: No effect. 1: Subtracts 1 hour to the current time. This can be used to change the outdoor initialization mode in winter.
16	AD1H	Add 1 hour (summer time change) When this bit is set, 1 hour can be added to the calendar time. This bit is always read as. 0: No effect. 1: Adds 1 hour to the current time. This can be used to change the outdoor initialization mode in summer.
15	TSIEN	Timestamp interrupt enable bit. 0: Disable Timestamp interrupt 1: Enable Timestamp interrupt
14	WTIEN	Wakeup timer interrupt enable 0: Disable wakeup timer interrupt. 1: Enable wakeup timer interrupt.
13	ALBIEN	Alarm B interrupt enable 0: Disable Alarm B interrupt 1: Enable Alarm B Interrupt
12	ALAIEN	Alarm A interrupt enable 0: Disable Alarm A interrupt 1: Enable Alarm A interrupt
11	TSEN	Timestamp enable 0: Disable timestamp 1: Enable timestamp
10	WTEN	Wakeup timer enable 0: Disable wakeup timer 1: Enable wakeup timer
9	ALBEN	Alarm B enable

Bit Field	Name	Description
		0: Disable Alarm B 1: Enable Alarm B
8	ALAEN	Alarm A enable 0: Disable Alarm A 1: Enable Alarm A
7	Reserved	Reserved, the reset value must be maintained
6	HFMT	Hour format bit 0: 24 hour format 1: Am/PM format
5	BYPS	Bypass values from the shadow registers 0: Calendar values(when reading from RTC_SUBS、RTC_TSH and RTC_DATE) are copied from the shadow registers, which are refreshed every two RTCCLK cycles. 1: Calendar values(when reading from RTC_SUBS、RTC_TSH and RTC_DATE) are copied directly from the calendar counters. *Note: If the frequency of the APB1 clock falls below seven times the frequency of RTCCLK, RTC_CTRL.BYPS bit must be set to '1'*
4	REFCLKEN	RTC_REFIN reference clock detection bit (50 or 60Hz). 0: Disable RTC_REFIN detection 1: Enable RTC_REFIN detection. *Note: DIVS must be 0x00FF*
3	TEDGE	Time-stamp event active edge 0: Generate a timestamp event on the rising edge of the RTC_TS input 1: Generate a timestamp event on the falling edge of the RTC_TS input Note: When changing TEDGE, RTC_CTRL.TSEN must be reset to avoid accidentally setting RTC_INITSTS.TISF to 1
2:0	WKUPSEL[2:0]	Wakeup clock selection 000: RTC clock is divided by 16 001: RTC clock is divided by 8 010: RTC clock is divided by 4 011: RTC clock is divided by 2

Bit Field	Name	Description
		10x: ck_spre (usually 1Hz) clock is selected 11x: ck_spre (usually 1Hz) clock is selected and configure the wake-up timer counter value to 2^{16}

27.3.4 RTC Calendar Time Register (RTC_TSH)

Address offset: 0x08

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved									APM	HOT[1:0]		HOU[3:0]			
									rw	rw		rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	MIT[2:0]			MIU[2:0]				Reserved	SCT[2:0]			SCU[3:0]			
	rw			rw					rw			rw			

Bit Field	Name	Description
31:23	Reserved	Reserved, the reset value must be maintained
22	APM	AM/PM format. 0:AM format or 24-hour format 1:PM format
21:20	HOT[1:0]	Describes the hour tens value in BCD format
19:16	HOU[3:0]	Describes the hour units value in BCD format
15	Reserved	Reserved, the reset value must be maintained
14:12	MIT [2: 0]	Describes the minute tens value in BCD format
11:8	MIU[3:0]	Describes the minute units value in BCD format
7	Reserved	Reserved, the reset value must be maintained
6:4	SCT[2:0]	Describes the second tens value in BCD format
3:0	SCU[3:0]	Describes the second units value in BCD format

27.3.5 RTC Calendar Date Register (RTC_DATE)

Address offset: 0x0c

Reset value: 0x0000 2101

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								YRT[3:0]				YRU[3:0]			
								rw				rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WDU[2:0]			MOT	MOU[2:0]				Reserved		DAT[1:0]		DAU[3:0]			
rw			rw	rw						rw		rw			

Bit Field	Name	Description
31:24	Reserved	Reserved, the reset value must be maintained
23:20	YRT[3:0]	Describes the year tens value in BCD format
19:16	YRU[3:0]	Describes the year units value in BCD format
15:13	WDU[2:0]	Describes which Week day 000: Forbidden 001: Monday ... 111: Sunday
12	MOT	Describes the month tens value in BCD format
11:8	MOU[3:0]	Describes the month units value in BCD format
7:6	Reserved	Reserved, the reset value must be maintained
5:4	DAT[1:0]	Describes the date tens value in BCD format
3:0	DAU[3:0]	Describes the date units value in BCD format

27.3.6 RTC Write Protection register (RTC_WRP)

Address offset: 0x10

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								PKEY[7:0]							
								w							

Bit Field	Name	Description
31:8	Reserved	Reserved, the reset value must be maintained
7:0	PKEY[7:0]	Write protection key Reading this byte always returns 0x00. For detail on how to unlock RTC register write protection, see chapter RTC write protection register.

27.3.7 RTC Shift Control Register (RTC_ SCTRL)

Address offset: 0x14

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AD1S	Reserved														
w															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	SUBF[14:0]														
	w														

Bit Field	Name	Description
31	AD1S	Add one second 0: No impact. 1: Subtract one second to the clock/calendar This bit can only be written and read as zero. Writing to this bit does not have an impact when RTC_INITSTS.SHOPF=1.
30:15	Reserved	Reserved, the reset value must be maintained
14:0	ADFS[14:0]	Subtract a fraction of a second These bits can only be written and read as zero. Writing to these bits has no effect

Bit Field	Name	Description
		when RTC_INITSTS.SHOPF=1. The value written to SUBF[14:0] is added to the synchronized prescaler counter, causing a delay in the clock: delay(seconds)= (SUBF[14:0]+1) / (DIVS[14:0] + 1) The AD1S bit can be used together with the SUBF[14:0] bits "Advance (seconds)= (1 - ((SUBF[14:0]+1) / (DIVS[14:0] + 1))) *Note: The RTC_INITSTS.RSYF bit will be cleared when writing to SUBF[14:0]. When RTC_INITSTS.RSYF=1, the shadow registers have been updated to the shifted time.*

27.3.8 RTC Sub-second Register (RTC_SUBS)

Address offset: 0x18

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SS[15:0]															
r															

Bit Field	Name	Description
31:16	Reserved	Reserved, the reset value must be maintained
15:0	SS[15:0]	Sub-second value. The value is the counter value of synchronous prescaler. This sub-second value is calculated by the below formula: Sub-second value = (RTC_PRE.DIVS[14:0]-SS)/(RTC_PRE.DIVS[14:0]+1) *Note: SS[15:0] can be larger than RTC_PRE.DIVS[14:0] only after the shift operation is completed. In this case, the correct time/date is one second slower than the time/date indicated by RTC_TSH/RTC_DATE.*

27.3.9 RTC Timestamp Time Register (RTC_TST)

Address offset: 0x1C

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Rserved									APM	HOT[1:0]		HOU[3:0]			
									r	r		r			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Rserved	MIT[2:0]			MIU[3:0]				Rserved	SET[2:0]			SEU[3:0]			
	r			r					r			r			

Bit Field	Name	Description
31:23	Reserved	Reserved, the reset value must be maintained
22	APM	AM/PM notation 0: AM or 24-hour clock 1: PM
21:20	HOT[1:0]	Describes the hour tens value in BCD format
19:16	HOU[3:0]	Describes the hour units value in BCD format
15	Reserved	Reserved, the reset value must be maintained
14:12	MIT[2:0]	Describes the minute tens value in BCD format
11:8	MIU[3:0]	Describes the minute units value in BCD format
7	Reserved	Reserved, the reset value must be maintained
6:4	SET[2:0]	Describes the second tens value in BCD format
3:0	SEU[3:0]	Describes the second units value in BCD format

27.3.10 RTC Alarm A Register (RTC_ALARMA)

Address offset: 0x20

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MASK4	WKDSEL	DTT[1:0]		DTU[3:0]				MASK3	APM	HOT[1:0]		HOU[3:0]			
rw	rw	rw		rw				rw	rw	rw		rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MASK2	MIT[2:0]			MIU[3:0]				MASK1	SET[2:0]			SEU[3:0]			
rw	rw			rw				rw	rw			rw			

Bit Field	Name	Description
31	MASK4	Alarm date mask 0: Date/day match 1: Date/day not match
30	WKDSEL	Week day selection 0: DTU[3:0] represents the date units 1: DTU[3:0] represents week day only. DTT[1:0] is not considered
29:28	DTT[1:0]	Describes the date tens value in BCD format
27:24	DTU[3:0]	Describes the date units value in BCD format
23	MASK3	Alarm hours mask 0: Hours match 1: Hours not match
22	APM	AM/PM notation 0: AM or 24 hours format 1: PM format
21:20	HOT[1:0]	Describes the hour tens value in BCD format
19:16	HOU[3:0]	Describes the hour units value in BCD format
15	MASK2	Alarm minutes mask 0: Minutes match 1: Minutes not match
14:12	MIT[2:0]	Describes the minute tens value in BCD format
11:8	MIU[3:0]	Describes the minute units value in BCD format
7	MASK1	Alarm seconds mask 0: Seconds match 1: Seconds not match
6:4	SET[2:0]	Describes the second tens value in BCD format

Bit Field	Name	Description
3:0	SEU[3:0]	Describes the second units value in BCD format

27.3.11 RTC Prescaler Register (RTC_PRE)

Address offset: 0x24

Reset value: 0x007F 00FF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved									DIVA[6:0]						
												rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	DIVS[14:0]														
								rw							

Bit Field	Name	Description
31:23	Reserved	Reserved, the reset value must be maintained
22:16	DIVA[6:0]	Asynchronous prescaler factor f_{ck_apre} = RTCCLK/(DIVA[6:0]+1)
15	Reserved	Reserved, the reset value must be maintained
14:0	DIVS[14:0]	Synchronous prescaler factor $f_{ck\_spre} = f_{ck\_apre}$/(DIVS[14:0]+1)

27.3.12 RTC Alarm B Register (RTC_ALARMB)

Address offset: 0x28

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MASK4	WKDSEL	DTT[1:0]		DTU[3:0]				MASK3	APM	HOT[1:0]		HOU[3:0]			
rw	rw	rw		rw				rw	rw	rw		rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MASK2	MIT[2:0]			MIU[3:0]				MASK1	SET[2:0]			SEU[3:0]			
rw	rw			rw				rw	rw			rw			

Bit Field	Name	Description
31	MASK4	Alarm date mask 0: Date/day match 1: Date/day not match
30	WKDSEL	Week day selection 0: DTU[3:0] represents the date units 1: DTU[3:0] represents week day only. DTT[1:0] is not considered
29:28	DTT[1:0]	Describes the date tens value in BCD format
27:24	DTU[3:0]	Describes the date units value in BCD format
23	MASK3	Alarm hours mask 0: Hours match 1: Hours not match
22	APM	AM/PM notation 0: AM or 24 hours format 1: PM format
21:20	HOT[1:0]	Describes the hour tens value in BCD format
19:16	HOU[3:0]	Describes the hour units value in BCD format
15	MASK2	Alarm minutes mask 0: Minutes match 1: Minutes not match
14:12	MIT[2:0]	Describes the minute tens value in BCD format
11:8	MIU[3:0]	Describes the minute units value in BCD format
7	MASK1	Alarm seconds mask 0: Seconds match 1: Seconds not match
6:4	SET[2:0]	Describes the second tens value in BCD format
3:0	SEU[3:0]	Describes the second units value in BCD format

27.3.13 RTC Wakeup Timer Register (RTC_WKUPT)

Address offset: 0x2C

Reset value: 0x0000 FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WKUPT[15:0]															
rw															

Bit Field	Name	Description
31:16	Reserved	Reserved, the reset value must be maintained
15:0	WKUPT[15:0]	Wake up auto-reload value bits When RTC_CTRL.WTEN=1, set the RTC_INITSTS.WTF flag every N ck_wut cycles. When RTC_CTRL.WKUPSEL[2:0]=0b10x, N = WKUPT[15:0]+ 1; When RTC_CTRL.WKUPSEL[2:0]=0b11x, N = (WKUPT[15:0]+ 1)+65536. *Note:* *This register change (such as the second setting or later Settings) needs to be changed in the wakeup interrupt, otherwise the changed settings will not take effect immediately, but will take effect after the next wakeup;* *In particular, when RTC_CTRL.WKUPSEL[2:0] is set to 010, the modified setting does not take effect immediately, but will take effect after wake up in the next cycle.*

27.3.14 RTC Tamper Configuration Register（RTC_ TMPCFG）

Address offset：0x30

Reset value：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				TP3MF	TP3NOE	TP3INT	TP2MF	TP2NOE	TP2INT	TP1MF	TP1NOE	TP1INT	TPPUDIS	TPPRCH[1:0]	
				rw		rw	rw		rw	rw		rw	rw	rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		TPFLT[1:0]		Reserved	TPFREQ[2:0]			TPTS	TP3TRG	TP3EN	TP2TRG	TP2EN	TPINTEN	TP1TRG	TP1EN

rw rw rw rw rw rw rw rw rw rw

Bit Field	Name	Description
31:28	Reserved	Reserved, the reset value must be maintained
27	TP3MF	Tamper 3 mask flag. 0: Do not mask Tamper 3 events. 1: Mask Tamper 3 events. *Note: When TP3MF is set, Tamper 3 interrupt must not be enabled.*
26	TP3NOE	Tamper 3 non-erase bit. 0: Tamper 3 event erases backup registers. 1: Tamper 3 event does not erase backup registers
25	TP3INTEN	Tamper 3 interrupt enable bit. 0: Disables Tamper 3 interrupt when TPINTEN = 0. 1: Enables Tamper 3 interrupt.
24	TP2MF	Tamper 2 mask flag. 0: Do not mask Tamper 2 events. 1: Mask Tamper 2 events. *Note: When TP2MF is set, Tamper 2 interrupt must not be enabled*
23	TP2NOE	Tamper 2 non-erase bit. 0: Tamper 2 event erases backup registers. 1: Tamper 2 event does not erase backup registers
22	TP2INTEN	Tamper 2 interrupt enable bit. 0: Disables Tamper 2 interrupt when TPINTEN = 0. 1: Enables Tamper 2 interrupt.
21	TP1MF	Tamper 1 mask flag. 0: Do not mask Tamper 1 events. 1: Mask Tamper 1 events. *Note: When TP1MF is set, Tamper 1 interrupt must not be enabled*
20	TP1NOE	Tamper 1 non-erase bit.

		0: Tamper 1 event erases backup registers. 1: Tamper 1 event does not erase backup registers
19	TP1INTEN	Tamper 1 interrupt enable bit. 0: Disables Tamper 1 interrupt when TPINTEN = 0. 1: Enables Tamper 1 interrupt
18	TPPUDIS	RTC_TAMPx pull-up disable bit. 0: Enable pre-charging of RTC_TAMPx pin before each sampling. 1: Disable pre-charging of RTC_TAMPx pin
17:16	TPPRCH[1:0]	RTC_TAMPx pre-charging duration. These bits determine the pre-charging time before each sampling. 0x0: 1 RTCCLK period 0x1: 2 RTCCLK periods 0x2: 4 RTCCLK periods 0x3: 8 RTCCLK periods
15:14	Reserved	Reserved, the reset value must be maintained
13:12	TPFLT[1:0]	RTC_TAMPx filter count. These bits determine the number of consecutive samples on a valid level. 0x0: Trigger tamper event after 1 sample on a valid level 0x1: Trigger tamper event after 2 consecutive samples on a valid level 0x2: Trigger tamper event after 4 consecutive samples on a valid level 0x3: Trigger tamper event after 8 consecutive samples on a valid level
11	Reserved	Reserved, the reset value must be maintained
10:8	TPFREQ[2:0]	Tamper sampling frequency. This bit determines the frequency at which each RTC_TAMPx input is sampled. 0x0: Sample once every 32768 RTCCLKs (frequency of 1 Hz when RTCCLK = 32.768 KHz) 0x1: Sample once every 16384 RTCCLKs 0x2: Sample once every 8192 RTCCLKs 0x3: Sample once every 4096 RTCCLKs 0x4: Sample once every 2048 RTCCLKs

		0x5: Sample once every 1024 RTCCLKs 0x6: Sample once every 512 RTCCLKs 0x7: Sample once every 256 RTCCLKs
7	TPTS	Enable timestamp on tamper detection event. 0: Do not save timestamp on tamper detection event 1: Save timestamp on tamper detection event TPTS remains valid even if RTC_CTRL.TSEN=0
6	TP3TRG	Tamper 3 event trigger mode. If TPFLT[1:0] != 00, Tamper detection is in level mode: 0: Low level triggers Tamper detection event. 1: High level triggers Tamper detection event. If TPFLT[1:0] = 00, Tamper detection is in edge mode: 0: Rising edge triggers Tamper detection event. 1: Falling edge triggers Tamper detection event
5	TP3EN	RTC_TAMP3 detection enable bit. 0: Disable RTC_TAMP3 input detection 1: Enable RTC_TAMP3 input detection
4	TP2TRG	Tamper 2 event trigger mode. If TPFLT[1:0] != 00, tamper detection is in level mode: 0: Low level triggers tamper detection event. 1: High level triggers tamper detection event. If TPFLT[1:0] = 00, tamper detection is in edge mode: 0: Rising edge triggers tamper detection event. 1: Falling edge triggers tamper detection event
3	TP2EN	RTC_TAMP2 detection enable bit. 0: Disable RTC_TAMP2 input detection 1: Enable RTC_TAMP2 input detection
2	TPINTEN	Tamper event interrupt enable. 0: Disable tamper interrupt

		1: Enable tamper interrupt *Note: This bit enables interrupts for all tamper pin events, regardless of the TPxINTEN level. If this bit is cleared, each tamper event interrupt can be individually enabled by setting TPxINTEN*
1	TP1TRG	Tamper 1 event trigger mode. If TPFLT[1:0] != 00, tamper detection is in level mode: 0: Low level triggers tamper detection event. 1: High level triggers tamper detection event. If TPFLT[1:0] = 00, tamper detection is in edge mode: 0: Rising edge triggers tamper detection event. 1: Falling edge triggers tamper detection event.
0	TP1EN	RTC_TAMP1 detection enable bit. 0: Disable RTC_TAMP1 input detection 1: Enable RTC_TAMP1 input detection

27.3.15 RTC Alarm A Sub-Second Register (RTC_ ALRMASS)

Address offset: 0x34

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved												MASKSSB[3:0]			
												rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	SSV[14:0]														
	rw														

Bit Field	**Name**	**Description**
31:20	Reserved	Reserved, the reset value must be maintained
19:16	MASKSSB[3:0]	Mask the most significant bit from this bits. 0x0: No comparison on sub seconds for Alarm. The alarm is set when the seconds unit is incremented (assuming that the rest of the fields match). 0x1: Only SSV[0] is compared and other bits are not compared. 0x2: Only SSV[1:0] are compared and other bits are not compared.

Bit Field	Name	Description
		0x3: Only SSV[2:0] are compared and other bits are not compared. ... 0xC: Only SSV[11:0] are compared and other bits are not compared. 0xD: Only SSV[12:0] are compared and other bits are not compared. 0xE: Only SSV[13:0] are compared and other bits are not compared. 0xF: SSV[14:0] are compared Synchronization counter RTC_SUBS.SS[15] bit is never compared.
15	Reserved	Reserved, the reset value must be maintained
14:0	SSV[14:0]	Sub seconds value This value is compared with the synchronous prescaler counter RTC_SUBS.SS[14:0], and bit number of compared is controlled by MASKSSB[3:0].

27.3.16 RTC Option Register (RTC_ OPT)

Address offset: 0x38

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															TYPE
															rw

Bit Field	Name	Description
31:1	Reserved	Reserved, the reset value must be maintained
0	TYPE	RTC_ALARM output type on PC13 0: Open-drain output 1: Push-pull output

27.3.17 RTC Alarm B Sub-Second Register (RTC_ ALRMBSS)

Address offset: 0x3C

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved												MASKSSB[3:0]			
												rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	SSV[14:0]														
	rw														

Bit Field	Name	Description
31:20	Reserved	Reserved, the reset value must be maintained
19:16	MASKSSB[3:0]	Mask the most significant bit from this bits. 0x0: No comparison on sub seconds for Alarm. The alarm is set when the seconds unit is incremented (assuming that the rest of the fields match). 0x1: Only SSV[0] is compared and other bits are not compared. 0x2: Only SSV[1:0] are compared and other bits are not compared. 0x3: Only SSV[2:0] are compared and other bits are not compared. ... 0xC: Only SSV[11:0] are compared and other bits are not compared. 0xD: Only SSV[12:0] are compared and other bits are not compared. 0xE: Only SSV[13:0] are compared and other bits are not compared. 0xF: SSV[14:0] are compared Synchronization counter RTC_SUBS.SS[15] bit is never compared.
15	Reserved	Reserved, the reset value must be maintained
14:0	SSV[14:0]	Sub seconds value This value is compared with the synchronous prescaler counter RTC_SUBS.SS[14:0], and bit number of compared is controlled by MASKSSB[3:0].

27.3.18 RTC Calibration Register (RTC_CALIB)

Address offset: 0x3C

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				CP	CW8	CW16	CM[8:0]								
				rw	rw	rw	rw								

Bit Field	**Name**	**Description**
31:12	Reserved	Reserved, the reset value must be maintained
11	CP	Increase frequency of RTC by 488.5 ppm This feature is intended to be used along with CM[8:0]. When frequency of RTCCLK is 32768 Hz, the number of RTCCLK pulses added during a 32-second window is ((512 * CP) – CM[8:0]). 0: No add pulse. 1: One RTCCLK pulse is inserted every 2^{11} pulses.
10	CW8	Select an 8-second calibration cycle period 0: Not effect. 1: Select an 8-second calibration period. When CW8 is set to ‘1’, the 8-second calibration cycle period is selected. *Note: when CW8 = 1, CM[1:0] will always be' 00'*
9	CW16	To select a 16-second calibration cycle period 0: Not effect. 1: Select a calibration period of 16 seconds. If CW8 = 1, this bit cannot be set to 1. *Note: when CW16 = 1, CM[0] will always be '0'*
8:0	CM[8:0]	Negative calibration bits The number of mask pulse out of 2^{20} RTCCLK pulses. This effectively decreases the frequency of the calendar with a resolution of 0.9537 ppm.

27.3.19 RTC Timestamp Sub-Second Register (RTC_TSSS)

Address offset: 0x44

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSE[15:0]															
r															

Bit Field	Name	Description
31:16	Reserved	Reserved, the reset value must be maintained
15:0	SSE[15:0]	Sub second value SSE[15:0] is the value in the synchronous prescaler counter. The fraction of a second is provided by the formula below: Second fraction = (RTC_PRE.DIVS[14:0] – SSE[15:0]) / (RTC_PRE.DIVS[14:0] + 1) *Note: SSE[15:0] can be larger than RTC_PRE.DIVS[14:0] only after a shift operation. In that case, the correct time/date is one second less than as indicated by RTC_TSH/RTC_DATE.*

27.3.20 RTC Timestamp Date Register (RTC_TSD)

Address offset: 0x48

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MOT	WDU[2:0]			MOU[2:0]				Reserved		DAT[1:0]		DAU[3:0]			
r	r			r						r		r			

Bit Field	Name	Description
31:16	Reserved	Reserved, the reset value must be maintained
15	MOT	Describes the month tens value in BCD format
14:12	WDU[2:0]	Describes which Week day 000: Forbidden 001: Monday ... 111: Sunday
11:8	MOU[3:0]	Describes the month units value in BCD format
7:6	Reserved	Reserved, the reset value must be maintained
5:4	DAT[1:0]	Describes the date tens value in BCD format
3:0	DAU[3:0]	Describes the date units value in BCD format

27.3.21 RTC Backup Register(RTC_ BKP(1~20))

Address offset：0x4C-0x98

Reset value：0x0000 0000

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="16">BF[31:16]</td></tr>
<tr><td colspan="16">rw</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="16">BF[15:0]</td></tr>
<tr><td colspan="16">rw</td></tr>
</table>

Bit Field	Name	Description
31:0	BF[31:0]	Backup data These registers can be read and written by software. These registers are powered by BKR when MR is switched off, so they are not reset when the system resets, and the contents of the registers remain valid when the device is operating in low-power mode. If RTC_TMPCFG.TPxNOE=0, these registers are reset when an tamper

		event is detected

28 Expanded Serial Peripheral Interface（xSPI）

28.1 Introduction

xSPI can be used as a single/dual/quad/octal SPI peripheral communication interface. It can operate in indirect and memory-mapped modes.

Indirect mode: Perform all operations using xSPI registers.

Memory-mapped mode: External flash is mapped to the microcontroller's address space, and the system treats it as internal storage space.

28.2 Main Features

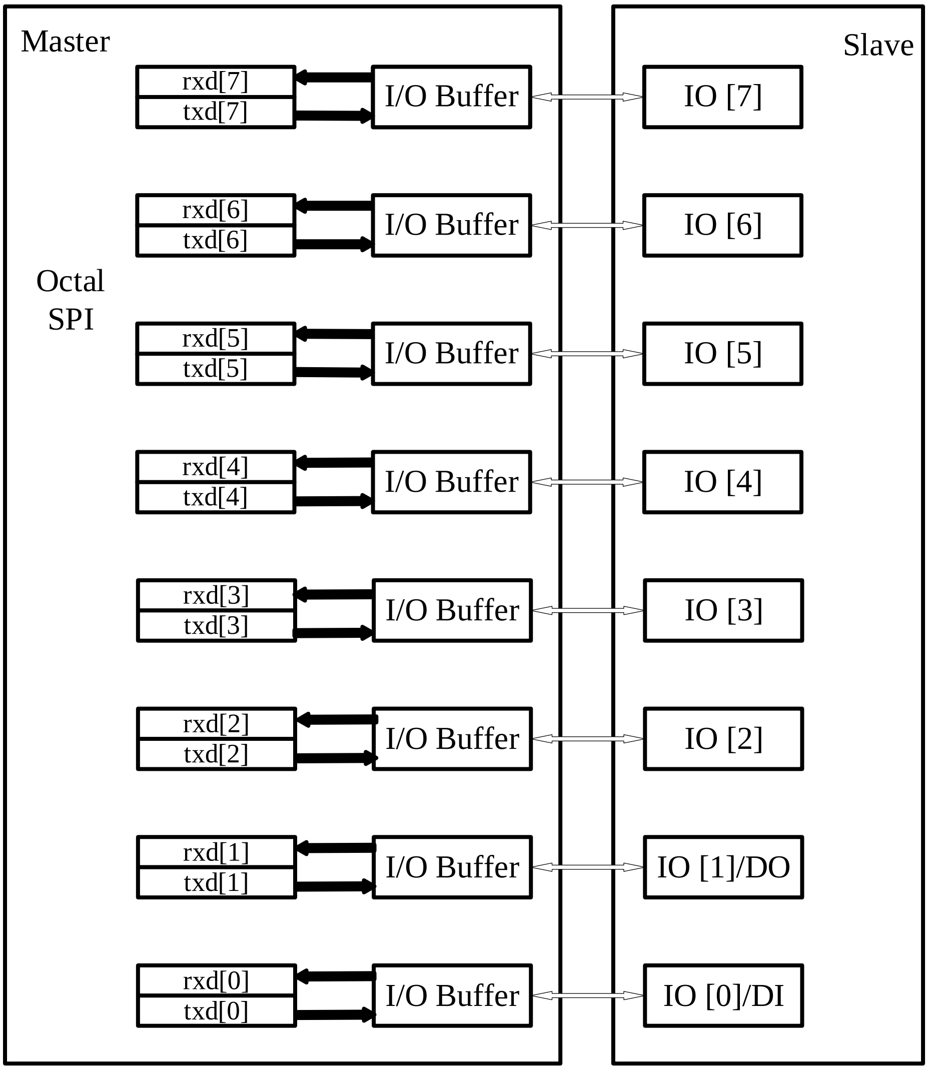
- Data communication supports single/dual/quad/octal mode
- Supports Single SPI/Normal SPI, DUAL SPI, QUAD SPI, Dual-QUAD, OCTAL SPI modes
- Supports Motorola SPI:
  - Standard/Dual/Quad/Octal SPI
- Supports Single Data Rate (SDR) and Double Data Rate (DDR) modes
- Supports read data strobe, supports data masking feature
- Supports clock stretching feature
- In indirect mode and memory-mapped mode, the frame format and opcodes can be configured through software
- Internally integrated FIFO for transmitting and receiving data
- Supports 8/16/32-bit data access
- Dedicated 16-word TX FIFO and 16-word RX FIFO
- DMA capability for transmission and reception
- Supports XIP read/write, supports communication with serial NOR FLASH
  - Supports continuous transmission in XIP mode
  - Supports data prefetch operation in XIP mode
- Supports automatic decryption when executing code through XSPI peripheral, i.e., storing code in ciphertext in XSPI peripheral, reading ciphertext during code execution and automatically decrypting into plaintext for CPU execution, without affecting the access speed to external storage. Decryption can be enabled/disabled by software control, with the root key stored in the NVR area, inaccessible to the user
- Supports communication with serial NAND FLASH and PSRAM
- xSPI performs read and write access to external storage. After xSPI initialization, no additional configuration of xSPI is required between writing to and reading from external storage, or between reading from and writing to external storage. Direct memory access (through SRAM address) is used for reading and writing to external storage.

- In master mode, it supports 2 external chip select output controls. In slave mode, it supports 1 chip select input. In master mode, all IOs multiplexed as chip select outputs can be multiplexed as chip select inputs in slave mode
- Supports multi-master arbitration function

Note: Mode bits phase is only used in XIP mode

28.3 xSPI Function Description

Figure 28-1 Octal SPI Block Diagram



28.3.1 Dual Data-Rate (DDR) Support

Enabling the DDR mode bit will enable double-rate transmission for the corresponding communication phases in 2/4/8-wire mode of SPI, which can improve throughput. Data is transferred on both the positive and negative edges of the clock.

The DDR mode supports the following modes of the SPI protocol:

- Mode0: When the default serial clock phase and default serial clock polarity are not enabled (SCPH= 0 &

SCPOL = 0)

- Mode3: When the default serial clock phase and default serial clock polarity are enabled (SCPH = 1& SCPOL = 1)

28.3.1.1 Description of Dual Data-Rate (DDR)

DDR commands enable data to be transferred on both edges of clock. Following are the different types of DDR commands:

- Address and data are transmitted (or received in case of data) in DDR format, while instruction is transmitted in standard format.
- Instruction, address, and data are all transmitted or received in DDR format

28.3.1.2 Transmitting Data in DDR Mode

In DDR mode, data is transmitted on both edges so that it is difficult to sample data correctly. xSPI determines the edge on which the data should be transmitted by configuring DDR_TXDE. This ensures that the receiver is able to get a stable data while sampling.

DWC_ssi sends data with respect to baud clock, which is an integral multiple of the internal clock (XSPI_clk * BAUDR). The data needs to be transmitted within half clock cycle (BAUDR/2), therefore the maximum value for DDR_TXDE register is equal to [(BAUDR/2)-1] .Data is driven before XSPI_clk *((BAUD/2)-TXD_DRIVE_EDGE) before the sampling edge. If the programmed value of DDR_TXDE is 0 then data is transmitted edge-aligned with respect to SCK (baud clock). If the programmed value of DDR_TXDE register is set to 1 then the data is transmitted two XSPI_clk before the edge of SCK.

Note: The baud rate value in DDR mode should be a multiple of 4

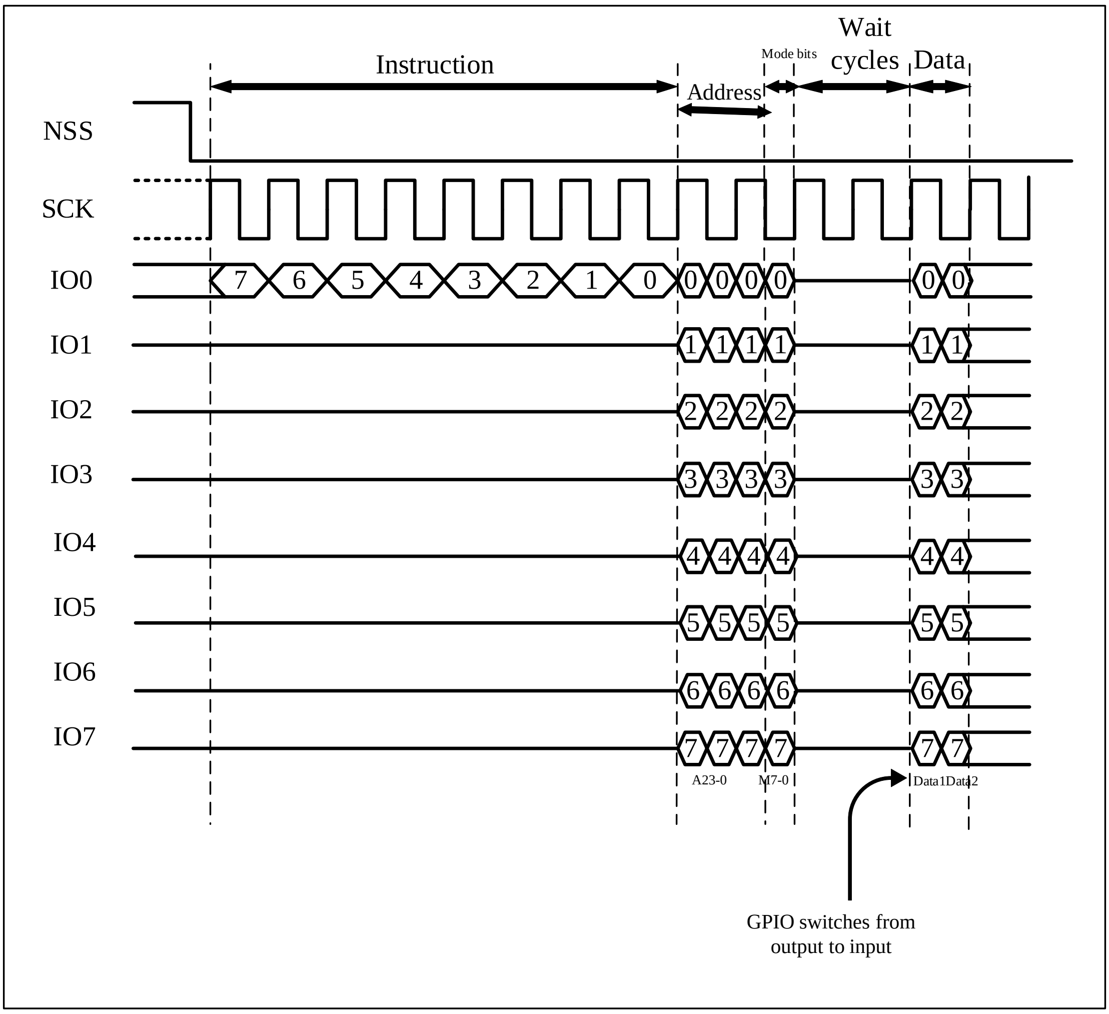
28.3.2 xSPI Command Sequence

XSPI_clk is derived from the AHB clock, BAUDR is the divisor, and SCK is the xSPI output clock.

xSPI communicates with external devices through commands. Each command includes five phases: Instruction phase, Address phase, Mode bits phase, Wait cycles phase, and Data phase. Any phase can be skipped, but at least one of the Instruction phase, Address phase, Mode bits phase, or Data phase must be retained.

Note: Mode bits are only supported in XIP mode. After mode bit support is enabled, mode bits occur immediately after the address phase is completed and follow the same rules as for the address bits. The length of the mode bits can be programmed as 2, 4, 8, 16 (valid after enabling the XIP mode bit phase)

Figure 28-2 Octal SPI Command Sequence



28.3.3 XIP

XIP operations are supported in only in Dual, Quad, or Octal enhanced SPI modes of operation, and so, the XSPI_CTRL0.SPIFRF bit must not be programmed to 0. Typically, an XIP operation consists of an address phase and a data phase.

When XSPI_XIP_CTRL field of the DFSHC register is set to 0, AHB control signals are used to derive the values of the data frame size and the number of data frames to be fetched by the device. hsize signal is used to get the value of data frame size for the transfer. Table 28-2 shows the data frame size mapping from the HSIXZE value.

Table 28-1 Hsize to Data Frame Size Decode

HSIZE	Data frame Size (DFS)
3’b000	8
3’b001	16

>=3’b010	32

The number of data frames to be fetched is derived from the HBURST signal. Table 28-3 shows the decode for the number for data frames decoded from the HBURST value.

Table 28-2 HBURST to Number of Data Frames(XSPI_XIP_CTRL. DFSHC=0)

HBURST	Type	Number of Data Frames (NDF)
000	Single	1
001	INCR	Data is fetched until the burst completes
010	WRAP4	4
011	INCR4	4
100	WRAP8	8
101	INCR8	8
110	WRAP16	16
111	INCR16	16

28.3.3.1 XIP Transfers

When the XSPI_XIP_CTRL. XIPCTEN bit is set to 1, continuous transmission in XIP mode is enabled (reducing the number of AHB interface requests).

The programming flow to set-up an XIP transfer is as follows:

1) Set the SPI frame format value in XSPI_CTRL0 register.

2) XIP read operation requires configuring XSPI_XIP_CTRL.ADDRL [3:0] and XSPI_XIP_CTRL.WAITCYCLES[4:0]

3) XIP write operation requires configuring XSPI_XIP_WRITE_CTRL.WRADDRL and XSPI_XIP_WRITE_CTRL.XIPWRWCY[4:0].

XIP instruction phase

The instruction phase can also be included in XIP transfer by using XSPI_XIP_CTRL field of the XIPINSTEN register. In this case, the following registers must be set:

1) Configure XSPI_ENH_CTRL0.INSTL to set the length of the instruction.

2) Write the instruction opcodes in the XSPI_XIP_WRITE_INCR_INST and XSPI_XIP_WRITE_WRAP_INST registers. After the programming is complete, you can initiate a read transaction through the AHB interface that is transferred to the SPI peripheral using programmed values.

Select the instruction opcode from different registers based on the AHB transfer type and read/write mode (INCR type, WRAP type). INCR type, WRAP type are burst transfer types

xSPI first receives all the data and then transmits on the AHB interface. Once the required amount of data is fetched for a particular burst, the slave is de-selected.

- Fixed INCR/WRAP Burst Transfer: When xSPI receives a FIXED burst request, it fetchs only fixed amount of data from SPI device. The number of data frames (NDF) field is determined by HBURST and data frame size is derived from the hsize signal. For the WRAP request, the device must send the correct data and Xspi forwards the data on the AHB interface.
- Undefined Incrementing Burst (INCR): In this case, xSPI continues to fetch the data from the device unless end of burst (IDLE transfer) is detected on slave interface. xSPI fetches s maximum of 1 KB data from SPI device for this transfer type.

28.3.3.2 AHB WAIT Transfers

During XIP transfers, AHB master may insert wait states in the transfer. In this condition, xSPI does not break the transfer on SPI interface and continues to fetch the data from SPI slave until all the required data frames are fetched for the ongoing burst. This intermediate data is stored in xSPI receive FIFO and is transmitted to AHB interface after WAIT cycles are completed.

For example, after the master receives D-1 data, AHB inserts BUSY cycles in the transfer. During this time, no data is sent to AHB master. After AHB resumes the AHB transfer, xSPI starts sending.

As intermediate data frames are stored in RX FIFO, it may cause FIFO overflow condition during the transfer, for example if the FIFO depth is set to 8 and AHB interface is performing a burst transfer of 16.This may lead to data mismatch in the system. Therefore, it has to be taken into consideration while choosing RX FIFO depth. If RX FIFO overflow interrupt is detected by the software during XIP transfer, it must re-attempt the data transfer to avoid data corruption.

28.4 Configuration Process

28.4.1 xSPI Indirect Mode

In indirect mode, commands are transferred by configuring xSPI registers, and data is transferred through read and write data registers, following the same manner as with other communication peripherals.

When XSPI_CTRL0.TMOD[1:0] = 00, it is in transmit & receive mode, where both transmitting and receiving data are valid. The data transfer process continues until the transmit FIFO is empty. Data received from external devices is stored in the Receive FIFO, which can be accessed by the master processor.

Note: transmit & receive mode can only be used in standard SPI mode (XSPI_CTRL0.SPIFRF[1:0] = 00).

When XSPI_CTRL0.TMOD[1:0] = 01, the xSPI is in indirect transmit mode, where the data to be transmitted is written to the XSPI_DATx register, and it is sent to the flash memory during the data transmission phase.

When XSPI_CTRL0.TMOD[1:0] = 10, the xSPI is in indirect receive mode, where data is received from the flash memory during the data reception phase, and the received data is obtained by reading the XSPI_DATx register.

When XSPI_CTRL0.TMOD[1:0] = 11, the xSPI is in EEPROM read mode, and can transmit the opcode/address to the EEPROM device

Note: EEPROM read mode is only supported in standard SPI mode (XSPI_CTRL0.SPIFRF[1:0] = 00). The number of bytes to be read is configured in XSPI_CTRL1.NDF[15:0].

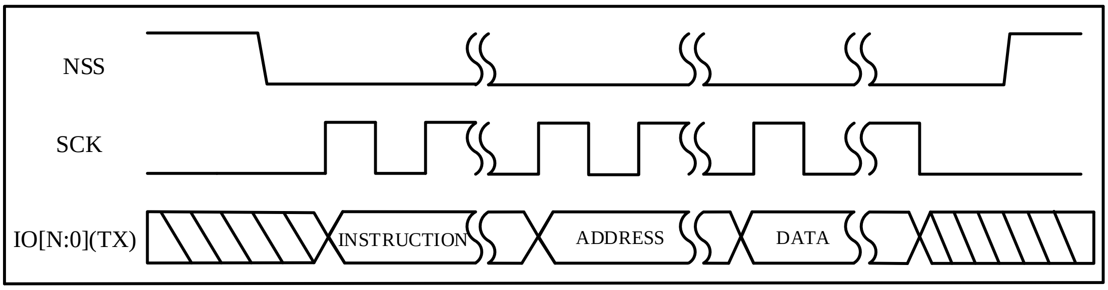
28.4.2 Write Operation in xSPI Indirect Mode

1) Specify the format (standard/dual/quad/octal mode) by configuring XSPI_CTRL0.SPIFRF[1:0]
2) Specifies data length by configuring XSPI_CTRL0.DFS[4:0]
3) Specify the address length (4bit to 60bit, with the option to skip the Address phase) by configuring XSPI_ENH_CTRL0.ADDRLEN[3:0].
4) Specify the instruction length (4bit, 8bit, 16bit, with the option to skip the Instruction phase) by configuring XSPI_ENH_CTRL0.INSTL[1:0].

Note: An instruction takes one FIFO location and address can take more than one FIFO locations. Both the instruction and address must be programmed in the data register (DR).

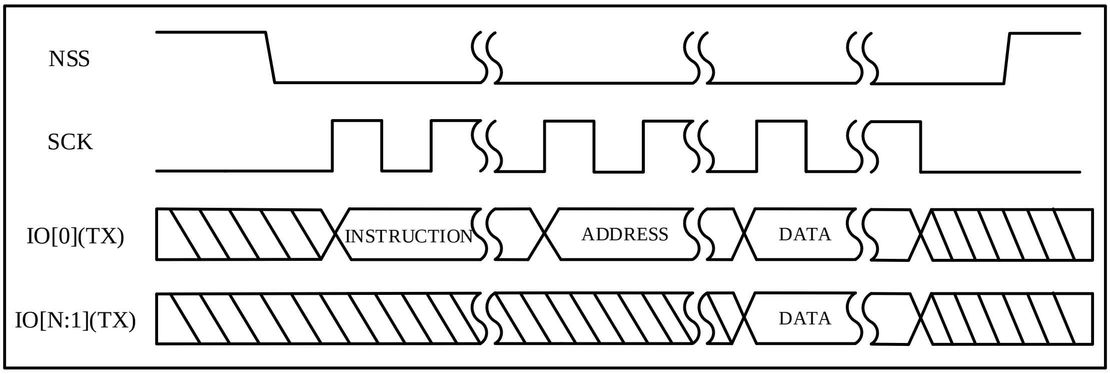
The write operation can be divided into three stages: Instruction phase, Address phase, Data phase.

- Typical Write Operation Timing



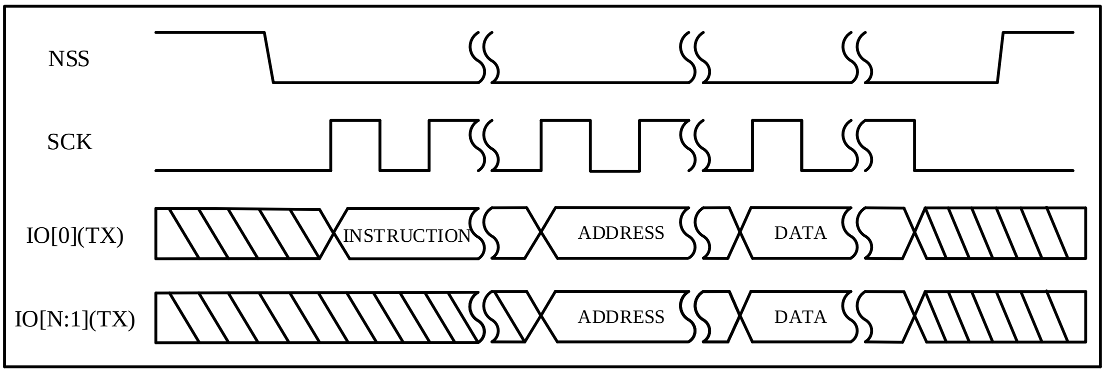
In quad SPI mode with N=3, for a single write operation, the instruction and address are transmitted only once, followed by transmitting the data frame stored in the XSPI_DATx register until the send FIFO is empty.

- Instruction and Address Transmitted in Standard SPI Format



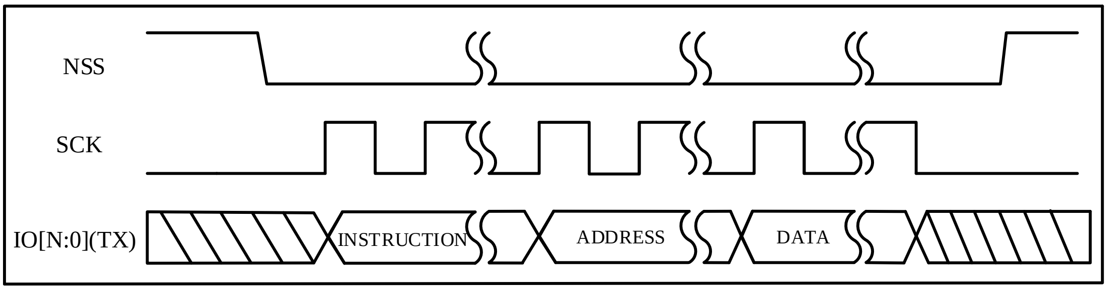
XSPI_ENH_CTRL0.TRANSTYPE[1:0] should be configured as 0. XSPI_CTRL0.SPIFRF[1:0] is configured as 0x02 (quad SPI mode, N=3). XSPI_CTRL0.SPIFRF[1:0] is configured as 0x01 (dual SPI mode, N=1).

- Instruction transmitted in standard and address transmitted in the format configured by CTRL0.SPIFRF



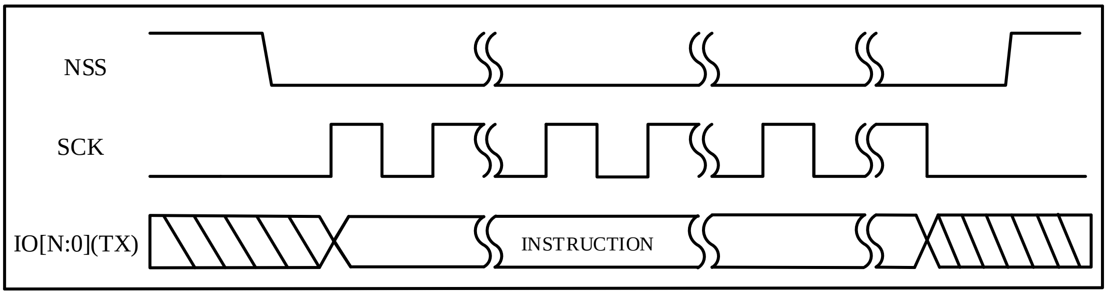
XSPI_ENH_CTRL0.TRANSTYPE[1:0] should be configured as 0x01. XSPI_CTRL0.SPIFRF[1:0] is configured as 0x02 (quad SPI mode, N=3). XSPI_CTRL0.SPIFRF[1:0] is configured as 0x01 (dual SPI mode, N=1).

- Instruction and address both transmitted in the format configured by CTRL0.SPIFRF



XSPI_ENH_CTRL0.TRANSTYPE[1:0] should be configured as 0x02. XSPI_CTRL0.SPIFRF[1:0] is configured as 0x02 (quad SPI mode, N=3). XSPI_CTRL0.SPIFRF[1:0] is configured as 0x01 (dual SPI mode, N=1).

- Instruction only Transfer in xSPI Format

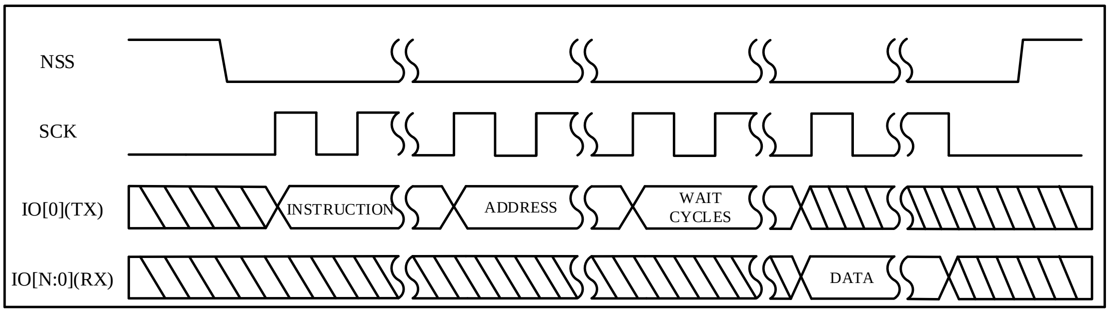


Irrelevant to the configuration of XSPI_ENH_CTRL0.TRANSTYPE[1:0]. XSPI_CTRL0.SPIFRF[1:0] is configured as 0x02 (quad SPI mode, N=3). XSPI_CTRL0.SPIFRF[1:0] is configured as 0x01 (dual SPI mode, N=1).

28.4.3 Read Operation in xSPI Indirect Mode

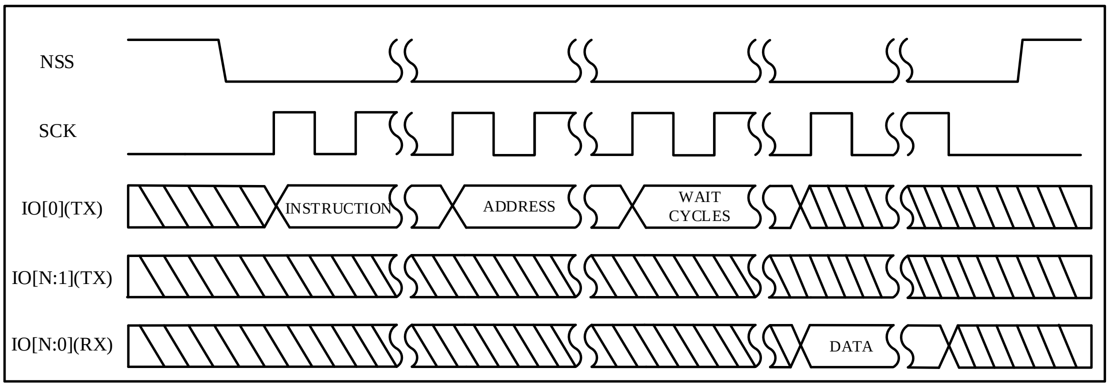
For a READ operation, xSPI sends instruction and control data once and waits until it receives NDF(XSPI_CTRL1[15:0]) number of data frames and then de-asserts slave select signal. The read operation can be divided into four stages: Instruction phase, Address phase, Wait cycles phase, and Data phase.

- Typical Read Operation Timing



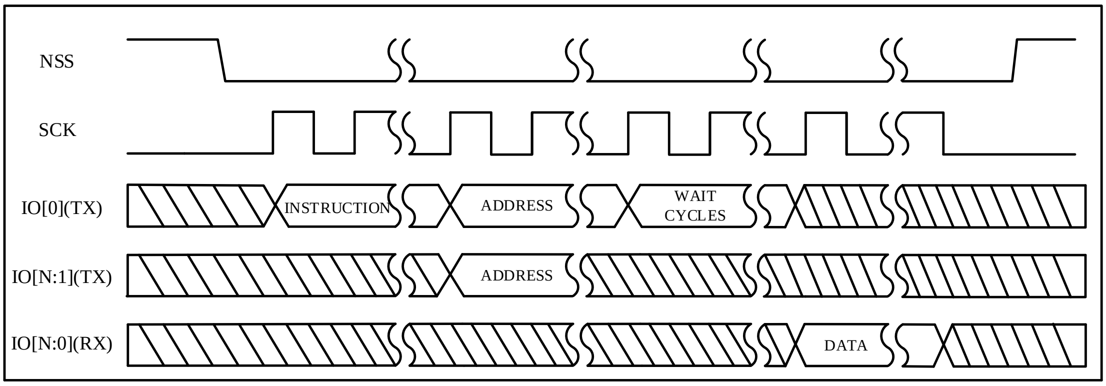
The received data will be transferred in the format configured by XSPI_CTRL0.SPIFRF[1:0] (configured as 0x02 in quad SPI mode, N=3).

- Instruction and address both transmitted in standard SPI format



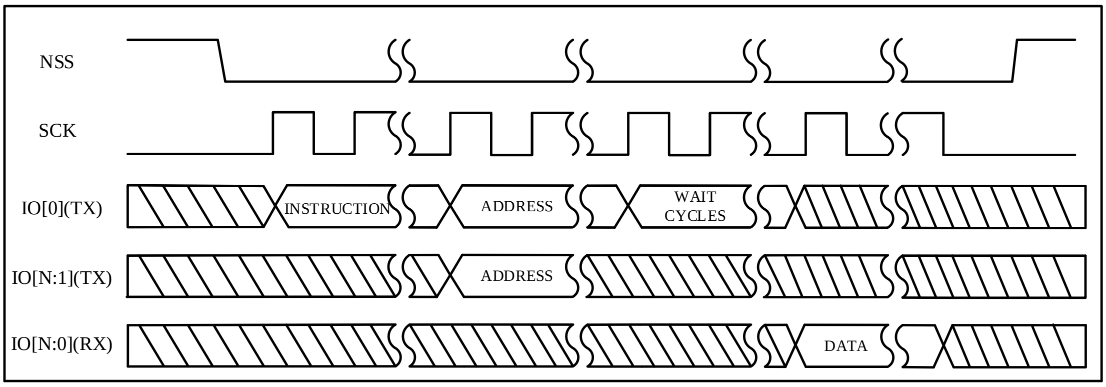
XSPI_ENH_CTRL0.TRANSTYPE[1:0] should be configured as 0x0. XSPI_ENH_CTRL0.WAITCYCLES[4:0] configures the number of WAIT cycles. XSPI_CTRL0.SPIFRF[1:0] is configured as 0x02 (quad SPI mode, N=3). XSPI_CTRL0.SPIFRF[1:0] is configured as 0x01 (dual SPI mode, N=1).

- Instruction transmitted in standard and address transmitted in the format configured by CTRL0.SPIFRF



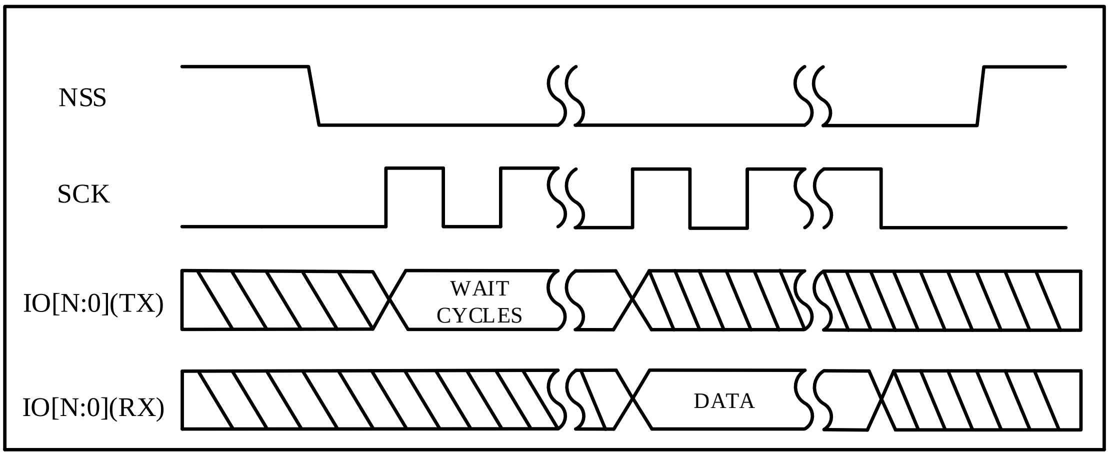
XSPI_ENH_CTRL0.TRANSTYPE[1:0] should be configured as 0x1. XSPI_ENH_CTRL0.WAITCYCLES[4:0] configures the number of WAIT cycles. XSPI_CTRL0.SPIFRF[1:0] is configured as 0x02 (quad SPI mode, N=3). XSPI_CTRL0.SPIFRF[1:0] is configured as 0x01 (dual SPI mode, N=1).

- Instruction and address both transmitted in the format configured by CTRL0.SPIFRF



XSPI_ENH_CTRL0.TRANSTYPE[1:0] should be configured as 0x2. XSPI_ENH_CTRL0.WAITCYCLES[4:0] configures the number of WAIT cycles. XSPI_CTRL0.SPIFRF[1:0] is configured as 0x02 (quad SPI mode, N=3). XSPI_CTRL0.SPIFRF[1:0] is configured as 0x01 (dual SPI mode, N=1).

- No Instruction and No Address READ Transfer



XSPI_ENH_CTRL0.ADDRLEN[3:0] is configured as 0, XSPI_ENH_CTRL0.INSTL[1:0] is configured as 0. XSPI_ENH_CTRL0.WAITCYCLES[4:0] configures the number of WAIT cycles. XSPI_CTRL0.SPIFRF[1:0] is configured as 0x02 (quad SPI mode, N=3). XSPI_CTRL0.SPIFRF[1:0] is configured as 0x01 (dual SPI mode, N=1).

28.5 xSPI Registers

28.5.1 xSPI Register Overview

Table 28-3XSPI Register Overview

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
000h	XSPI_CTRL0	MST	Reserved					DWSEN	Reserved	SPIFRF[1:0]		Reserved		CFS[3:0]				Reserved	SSTE	SRL	SLVOE	TMOD[1:0]		SCPOL	SCPH	FRF[1:0]		Reserved	DFS[4:0]				
	Reset Value	1						0		1	0			0	0	0	0		1	0	0	0	1	0	0	0	0		0	0	1	1	1
004h	XSPI_CTRL1	Reserved																NDF[15:0]															
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
008h	XSPI_EN	Reserved																															XSPIEN

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reset Value																																0
00Ch	XSPI_MW_CTRL	Reserved																												MHSEN	MCDIR	MWMOD
	Reset Value																														0	0	0
010h	XSPI_SLAVE_EN	Reserved																													SEN[1:0]	
	Reset Value																															0	0
014h	XSPI_BAUD	Reserved															CLKDIV[14:0]														Reserved
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
018h	XSPI_TXFT	Reserved											TXFTST[3:0]			Reserved											TXFTTEI[3:0]		
	Reset Value													0	0	0	0												0	0	0	0
01Ch	XSPI_RXFT	Reserved																											RXFTTFI[3:0]		
	Reset Value																													0	0	0	0
020h	XSPI_TXFN	Reserved																										TXFN[4:0]			
	Reset Value																												0	0	0	0	0
024h	XSPI_RXFN	Reserved																										RXFN[4:0]			
	Reset Value																												0	0	0	0	0
028h	XSPI_STS	CMPLTDDF[16:0]																Reserved							DCERR	TXE	RXFF	RXFNE	TXFE	TXFNF	BUSY
	Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0								0	0	0	0	1	1	0
02Ch	XSPI_IMASK	Reserved																							TXUIM	XRXOIM	MMCIM	RXFFIM	RXFOIM	RXFUIM	TXFOIM	TXFEIM
	Reset Value																									1	1	1	1	1	1	1	1
030h	XSPI_ISTS	Reserved																							TXUIS	XRXOIS	MMCIS	RXFFIS	RXFOIS	RXFUIS	TXFOIS	TXFEIS
	Reset Value																									0	0	0	0	0	0	0	0
034h	XSPI_RISTS	Reserved																							TXUIR	XRXORIS	MMCRIS	RXFFRIS	RXFORIS	RXFURIS	TXFORIS	TXFERIS
	Reset Value																									0	0	0	0	0	0	0	0
038h	XSPI_TXFOI_CLR	Reserved																														TXEICR
	Reset Value																																0
03Ch	XSPI_RXFOI_CLR	Reserved																														RXFOIC
	Reset Value																																0
040h	XSPI_RXFUI_CLR	Reserved																														RXFUIC
	Reset Value																																0
044h	XSPI_MMC_CLR	Reserved																														MMCIC
	Reset Value																																0
048h	XSPI_ICLR	Reserved																														INTC
	Reset Value																																0

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
04Ch	XSPI_DMA_CTRL	Reserved																														TXDMAEN	RXDMAEN
	Reset Value																															0	0
050h	XSPI_DMATDL_CTRL	Reserved																												DMATDL[3:0]			
	Reset Value																													0	0	0	0
054h	XSPI_DMARDL_CTRL	Reserved																												DMARDL[3:0]			
	Reset Value																													0	0	0	0
058h	XSPI_IDR	IDCODE[31:0]																															
	Reset Value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	1	1	0	1	1	0	0	0	1	0	0	0	0	0	1
05Ch	XSPI_VERSION_ID	VERSION [31:0]																															
	Reset Value	0	0	1	1	0	0	0	1	0	0	1	1	0	0	0	0	0	0	1	1	0	0	1	1	0	0	1	0	1	0	1	0
060h+004h*x(x=0～31)	XSPI_DATx	DATx[31:0]																															
	Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0F0h	XSPI_RS_DELAY	Reserved															SES	Reserved								SDCN[7:0]							
	Reset Value																0									0	0	0	0	0	0	0	0
0F4h	XSPI_ENH_CTRL0	Reserved	CLKSTREN	Reserved					SPIDMEN	Reserved					SPIRXDSEN	WRINDDREN	WRSPIDDREN	WAITCYCLES[4:0]					Reserved	INSTL[1:0]		Reserved		ADDRL[3:0]				TRANSTYPE[1:0]	
	Reset Value		0						0						0	0	0	0	0	0	0	0		1	0			0	0	0	0	0	0
0F8h	XSPI_DDR_TXDE	Reserved																								TXDE[7:0]							
	Reset Value																									0	0	0	0	0	0	0	0
0FCh	XSPI_XIP_MODE	Reserved																XIPMDBITS[15:0]															
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
100h	XSPI_XIP_INCR_TOC	Reserved																ITOC[15:0]															
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
104h	XSPI_XIP_WRAP_TOC	Reserved																WTOC[15:0]															
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
108h	XSPI_XIP_CTRL	Reserved		XIPPREEN	Reserved	XIPMBL[1:0]		RXDSSIGEN	XIPHYPEEN	XIPCTEN	XIPINSTEN	RXDSEN	WRINDDREN	DDREN	DFSHC	WAITCYCLES[4:0]					MDBITSEN	Reserved	INSTL[1:0]		Reserved	ADDRL[3:0]				TRANSTYPE[1:0]		FRF[1:0]	
	Reset Value			0		1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		1	0		0	0	0	0	0	0	1	0
10Ch	XSPI_XIP_SLAVE_EN	Reserved																														SEN[1:0]	
	Reset Value																															0	0
110h	XSPI_XIP_RXFOI_CLR	Reserved																															XRXFOIC
	Reset Value																																0
114h	XSPI_XIP_TOUT	Reserved																								XTOUT[7:0]							
	Reset Value																									0	0	0	0	0	0	0	0
140h	XSPI_XIP_WRITE_INCR_INST	Reserved																INCRWRINST[15:0]															
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
144h	XSPI_XIP_WRITE_INST	Reserved																WRAPWRINST[15:0]															
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
148h	XSPI_XIP_WRITE_CTRL	Reserved											XIPWRWCY[4:0]					Reserved				WRINDDREN	WRSPIDDREN	WRINSTL[1:0]		WRADDRL[3:0]				WRTRTYPE[1:0]		WRFRF[1:0]	
	Reset Value												0	0	0	0	0					0	0	0	0	0	1	1	1	0	0	1	0

28.5.2 xSPI Control Register0（XSPI_CTRL0）

Note: This register cannot be written to when XSPI_EN.XSPIEN = 1.

Address offset: 0x00

Reset value: 0x8080 4007

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MST	Reserved					DWSEN	Reserved	SPIFRF[1:0]		Reserved		CFS[3:0]			
rw						ro		rw				rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	SSTE	SRL	SLVOE	TMOD[1:0]		SCPOL	SCPH	FRF[1:0]		Reserved	DFS[4:0]				
	rw	rw	rw	rw		rw	rw	rw			rw				

Bit Field	Name	Description
31	MST	This field selects if xSPI is working in Master or Slave mode 00：xSPI is master 01：xSPI is slave
30:26	Reserved	Reserved, the reset value must be maintained
25	DWSEN	Enable Dynamic wait states in SPI mode of operation. 00：Disable SPI Dynamic Wait State 01：Enable SPI Dynamic Wait State This field is only applicable when *XSPI_CTRL0.FRF* is set to 0 (Motorola SPI Frame Format).
24	Reserved	Reserved, the reset value must be maintained
23:22	SPIFRF[1:0]	SPI Frame Format

Bit Field	Name	Description
		Selects data frame format for Transmitting/Receiving the data. 00：Standard SPI Format； 01：Dual SPI Format； 10：Quad SPI Format； 11：Octal SPI Format。
21:20	Reserved	Reserved, the reset value must be maintained
19:16	CFS[1:0]	Control Frame Size. Selects the length of the control word for the Microwire frame format 0000：01-bit Control Word； 0001：02-bit Control Word； 0010：03-bit Control Word； 0011：04-bit Control Word； …… 1110：15-bit Control Word； 1111：16-bit Control Word。
15	Reserved	Reserved, the reset value must be maintained
14	SSTE	Slave Select Toggle Enable While operating in SPI mode with clock phase (XSPI_CTRL0.SCPH) set to 0, this register controls the behavior of the slave select line (NSS) between data frames. 0: NSS will stay low and clock will run continuously for the duration of the transfer 1: NSS line will toggle between consecutive data frames, with the serial clock being held to its default value while NSS is high.
13	SRL	Shift Register Loop. Used for testing purposes only. When internally active,connects the transmit shift register output to the receive shift register input. Can be used in both serial-slave and serial-master modes. When the xSPI is configured as a slave in loopback mode, the chip select and clock signals must be provided by an external source. 00：Normal mode operation； 01：Test Mode Operation。

Bit Field	Name	Description
12	SLVOE	Slave Output Enable。 Relevant only when the xSPI is configured as aserial-slave device. When configured as a serial master, this bit field has no functionality. when the master transmits in broadcast mode, if this device does not need to respond to the data, disable the output by setting this bit to 1 by software. 0：Slave Output is enabled 1：Slave Output is disabled
11:10	TMOD[1:0]	Transfer Mode。 00：Transmit & Receive. Not Applicable in enhanced SPI operating mode 01：Transmit only mode. Or Write in enhanced SPI operating mode 10：Receive only mode. Or Read in enhanced SPI operating mode； 11：EEPROM Read mode. Not Applicable in enhanced SPI operating mode。
9	SCPOL	Serial Clock Polarity. Valid when the frame format (FRF) is set to Motorola SPI. Used to select the polarity of the inactive serial clock, which is held inactive when the xSPI master is not actively transferring data on the serial bus. 0：Inactive state of serial clock is low 1：Inactive state of serial clock is high
8	SCPH	Serial Clock Phase。 Valid when the frame format (FRF) is set to Motorola SPI. 0: Sample data on the first clock edge 1: Sample data on the second clock edge. *Note: In transmit mode, data is sent out half a SCK cycle ahead of time*
7:6	FRF[1:0]	Frame Format Selects which serial protocol transfers the data. 00：Motorola SPI Frame Format； 01：Texas Instruments SSP Frame Format； 10：National Semiconductors Microwire Frame Format； 11：reserved。

Bit Field	Name	Description
5	Reserved	Reserved, the reset value must be maintained
4:0	DFS[4:0]	Data Frame Size When the data frame size is programmed to be less than 32 bits, the receive data is automatically right-justified by the receive logic, with the upper bits of the receive FIFO zero-padded.。 You must right-justify transmit data before writing into the transmit FIFO. The transmit logic ignores the upper unused bits when transmitting the data。 0x0/0x01/0x02：Reserved 0x03：4bit 0x04：5bit 0x05：6bit …… 0x1D：30bit 0x1E：31bit 0x1F：32bit *Notes：* • *DFS value must be a multiple of 2 if XSPI_CTRL0.SPIFRF = 01* • *DFS value must be multiple of 4 if XSPI_CTRL0.SPIFRF = 10* • DFS value must be multiple of 8 if *XSPI_CTRL0.SPIFRF* = 11

28.5.3 xSPI Control Register1（XSPI_CTRL1）

Address offset: 0x04

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NDF[15:0]															
rw															

Bit Field	Name	Description
31:16	Reserved	Reserved, the reset value must be maintained
15:0	NDF[15:0]	Number of Data Frames receive mode： When XSPI_CTRL0.TMOD[1:0] = 10 or XSPI_CTRL0.TMOD[1:0] = 11， this register field sets the number of data frames to be continuously received by the xSPI. The xSPI continues to receive serial data until the number of data frames received is equal to this register value plus 1, which enables you to receive up to 64 KB of data in a continuous transfer. Transmit mode： when XSPI_ENH_CTRL0.CLKSTREN=1 and XSPI_CTRL0.TMOD[1:0] = 01， this register field sets the number of data frames to be continously transmitted by xSPI. If the Transmit FIFO goes empty in-between, xSPI masks the serial clock and wait for rest of the data until the programmed amount of frames are transferred successfully. When xSPI is configured as a serial slave, this register is invalid

28.5.4 xSPI Enable Register（XSPI_EN）

Address offset: 0x08

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															XSPIEN
															rw

Bit Field	Name	Description
31:1	Reserved	Reserved, the reset value must be maintained
0	XSPIEN	xSPI enable When disabled,all serial transfers are halted immediately. Transmit and receive FIFO buffers are cleared when the device is disabled.It is impossible to program

Bit Field	Name	Description
		some of the xSPI control registers when enabled. 0：Disable xSPI 1：Enable xSPI

28.5.5 xSPI MW Control Register（XSPI_MW_CTRL）

Note: This register cannot be written to when XSPI_EN.XSPIEN = 1.

Address offset：0x0C

Reset value：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													MHSEN	MCDIR	MWMOD
													rw	rw	rw

Bit Field	Name	Description
31:3	Reserved	Reserved, the reset value must be maintained
2	MHSEN	Microwire handshaking。 Relevant only when the xSPI is configured as a serial-master device. 0：handshaking interface is disabled。 1：handshaking interface is enabled，When enabled, the xSPI checks for a ready status from the target slave, after the transfer of the last data/control bit,before clearing the XSPI_STS.BUSY status in the SR register.
1	MCDIR	Microwire Control.。 01：xSPI transmits data 00：xSPI receives data
0	MWMOD	Microwire transfer mode 0：Non-sequential transfer，there must be a control word for each data word that is transmitted or received； 1：Sequential Transfer，only one control word is needed to transmit or receive a block of data words.

28.5.6 xSPI Slave Enable Register（XSPI_SLAVE_EN）

Note: The XSPI_SLAVE_EN register will only be enabled after enabling XSPI_EN.XSPIEN for enabling the individual slave select output lines。

Address offset：0x10

Reset value：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved														SEN[1:0]	
														rw	

Bit Field	Name	Description
31:2	Reserved	Reserved, the reset value must be maintained
1:0	SEN[1:0]	Slave Select Enable Flag Each bit in this register corresponds to a slave select line from the xSPI master. When a bit in this register is set to 1, the corresponding slave select line from the master is activated when a serial transfer begins. Setting or clearing bits in this register have no effect on the corresponding slave select outputs until a transfer is started.。 When not operating in broadcast mode, only one bit in this field should be set.。

28.5.7 xSPI Baud Rate Select（XSPI_BAUD）

Address offset：0x14

Reset value：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CLKDIV [14:0]															Reserved

rw

Bit Field	Name	Description
31:16	Reserved	Reserved, the reset value must be maintained
15:1	CLKDIV [14:0]	Clock Divider. If the value is 0, the serial output clock is disabled, If the value is not 0, the frequency of the serial output clock (FSCK) is derived from the following equation: FSCK = f_{HCLK} / (CLKDIV [14:0]*2)
0	Reserved	Reserved, the reset value must be maintained

28.5.8 xSPI Transmit FIFO Threshold Level Register(XSPI_TXFT)

Address offset：0x18

Reset value：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved												TXFTST[3:0]			
												rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												TXFTTEI[3:0]			
												rw			

Bit Field	Name	Description
31:20	Reserved	Reserved, the reset value must be maintained
19:16	TXFTST[3:0]	Transfer start FIFO level.。 Used to control the level of entries in transmit FIFO above which transfer will start on serial line.
15:4	Reserved	Reserved, the reset value must be maintained
3:0	TXFTTEI[3:0]	Transmit FIFO Threshold.。 When the number of transmit FIFO entries is less than or equal to this value, the transmit FIFO empty interrupt is triggered.。

28.5.9 xSPI Receive FIFO Threshold Level Register(XSPI_RXFT)

Address offset：0x1C

Reset value：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												RXFTTFI[3:0]			
												rw			

Bit Field	Name	Description
31:4	Reserved	Reserved, the reset value must be maintained
3:0	RXFTTFI[3:0]	Receive FIFO Threshold。 When the number of receive FIFO entries is greater than or equal to this value + 1, the receive FIFO full interrupt is triggered.。

28.5.10 xSPI Transmit FIFO Count Register（XSPI_TXFN）

Address offset：0x20

Reset value：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved											TXFN[4:0]				
											ro				

Bit Field	Name	Description
31:5	Reserved	Reserved, the reset value must be maintained
4:0	TXFN[4:0]	Contains the number of valid data entries in the transmit FIFO.

28.5.11 xSPI Receive FIFO Count Register（XSPI_RXFN）

Address offset：0x24

Reset value：0x0000 0000

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="16">Reserved</td></tr>
</table>

<table>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="11">Reserved</td><td colspan="5">RXFN[4:0]</td></tr>
<tr><td colspan="11"></td><td colspan="5">ro</td></tr>
</table>

Bit Field	Name	Description
31:5	Reserved	Reserved, the reset value must be maintained
4:0	RXFN[4:0]	Contains the number of valid data entries in the receive FIFO

28.5.12 xSPI Status Register（XSPI_STS）

Address offset：0x28

Reset value：0x0000 0006

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="16">CMPLTDDF[16:0]</td></tr>
<tr><td colspan="16">ro</td></tr>
</table>

<table>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td></td><td colspan="8">Reserved</td><td>DCERR</td><td>TXE</td><td>RXFF</td><td>RXFNE</td><td>TXFE</td><td>TXFNF</td><td>BUSY</td></tr>
<tr><td>ro</td><td colspan="8"></td><td>ro</td><td>ro</td><td>ro</td><td>ro</td><td>ro</td><td>ro</td><td>ro</td></tr>
</table>

Bit Field	Name	Description
31:15	CMPLTDDF[16:0]	Completed data frames This field indicates total data frames transferred in the previous internal DMA transfer
14:7	Reserved	Reserved, the reset value must be maintained
6	DCERR	Data collision error hen the xSPI is configured as a master device. This bit will be set if chip select

Bit Field	Name	Description
		input is asserted by other master, when the xSPI master is in the middle of the transfer. This informs the processor that the last data transfer was halted before completion. This bit is cleared when read 0 ： No error 1 ： Transmit data collision error
5	TXE	Transmission error。 0：No error 1：Transmission error Set if the transmit FIFO is empty when a transfer is started. This bit can be set only when the xSPI is configured as a slave device. Data from the previous transmission is resent on the txd line. This bit is cleared when read.
4	RXFF	Receive FIFO full. 0：Rx FIFO is not full 1：Rx FIFO is full When the receive FIFO is completely full, this bit is set. When the receive FIFO contains one or more empty location, this bit is cleared.
3	RXFNE	Receive FIFO not empty 0：Receive FIFO is empty 1：Receive FIFO is not empty Set when the receive FIFO contains one or more entries and is cleared when the receive FIFO is empty. This bit can be polled by software to completely empty the receive FIFO。
2	TXFE	Transmit FIFO empty 0：Transmit FIFO is not empty 1：Transmit FIFO is empty When the transmit FIFO is completely empty, this bit is set. When the transmit FIFO contains one or more valid entries, this bit is cleared. This bit field does not request an interrupt.
1	TXFNF	Transmit FIFO not full 0：Tx FIFO is full 1：Tx FIFO is not Full

Bit Field	Name	Description
		Set when the transmit FIFO contains one or more empty locations, and is cleared when the FIFO is full.
0	BUSY	xSPI busy flag. 0：xSPI is idle or disabled 1：xSPI is actively transferring data When set, indicates that a serial transfer is in progress; when cleared indicates that the xSPI is idle or disabled.

28.5.13 xSPI Interrupt Mask Register（XSPI_IMASK）

Address offset：0x2C

Reset value：0x0000 00FF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								TXUIM	XRXOIM	MMCIM	RXFFIM	RXFOIM	RXFUIM	TXFOIM	TXFEIM
								rw	rw	rw	rw	rw	rw	rw	rw

Bit Field	Name	Description
31:8	Reserved	Reserved, the reset value must be maintained
7	TXUIM	Transmit FIFO underflow interrupt mask 0：transmit FIFO underflow interrupt is masked 1：transmit FIFO underflow interrupt is not masked
6	XRXOIM	XIP receive FIFO overflow interrupt mask 0：XIP receive FIFO overflow interrupt is masked 1：XIP receive FIFO overflow interrupt is not masked
5	MMCIM	Multi-master contention interrupt mask. 0：multi-master contention interrupt is masked 1：multi-master contention interrupt is not masked

Bit Field	Name	Description
		Note：This bit field is not present if the xSPI is configured as a serial-master device
4	RXFFIM	Receive FIFO full interrupt mask 0：receive FIFO full interrupt is masked 1：receive FIFO full interrupt is not masked
3	RXFOIM	Receive FIFO overflow Interrupt mask 0：receive FIFO overflow interrupt is masked 1：receive FIFO overflow interrupt is not masked
2	RXFUIM	Receive FIFO underflow interrupt mask 0：receive FIFO underflow is masked 1：receive FIFO underflow is not masked
1	TXFOIM	Transmit FIFO overflow interrupt mask 0：transmit FIFO overflow interrupt is masked 1：transmit FIFO overflow interrupt is not masked
0	TXFEIM	Transmit FIFO empty interrupt mask 0：transmit FIFO empty interrupt is masked 1：transmit FIFO empty interrupt is not masked

28.5.14 xSPI Interrupt Status Register（XSPI_ISTS）

Address offset：0x30

Reset value：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								TXUIS	XRXOIS	MMCIS	RXFFIS	RXFOIS	RXFUIS	TXFOIS	TXFEIS
								ro	ro	ro	ro	ro	ro	ro	ro

Bit Field	Name	Description
31:8	Reserved	Reserved, the reset value must be maintained
7	TXUIS	Transmit FIFO underflow interrupt status 0：transmit FIFO underflow interrupt is not active after masking 1：transmit FIFO underflow interrupt is active after masking
6	XRXOIS	xip receive fifo overflow interrupt status 0：xip receive fifo overflow interrupt is not active after masking 1：xip receive fifo overflow interrupt is active after masking
5	MMCIS	multi-master contention interrupt status. 0：multi-master contention interrupt is not active after masking 1：multi-master contention interrupt is active after masking *Note: This bit field is not present if the xSPI is configured as a serial-slave device*
4	RXFFIS	Receive FIFO full interrupt status 0：receive FIFO full interrupt is not active after masking 1：receive FIFO full interrupt is active after masking
3	RXFOIS	Receive FIFO overflow interrupt status 0：receive FIFO overflow interrupt is not active after masking 1：receive FIFO overflow interrupt is active after masking
2	RXFUIS	Receive FIFO underflow interrupt status 0：receive FIFO underflow interrupt is not active after masking 1：receive FIFO underflow interrupt is active after masking
1	TXFOIS	Transmit FIFO overflow interrupt status 0：transmit FIFO overflow interrupt is not active after masking 1：transmit FIFO overflow interrupt is active after masking
0	TXFEIS	Transmit FIFO empty interrupt status 0：transmit FIFO empty interrupt is not active after masking 1：transmit FIFO empty interrupt is active after masking

28.5.15 xSPI Raw Interrupt Status Register（XSPI_RISTS）

Address offset：0x34

Reset value：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								TXUIR	XRXORIS	MMCRIS	RXFFRIS	RXFORIS	RXFURIS	TXFORIS	TXFERIS
								ro	ro	ro	ro	ro	ro	ro	ro

Bit Field	Name	Description
31:8	Reserved	Reserved, the reset value must be maintained
7	TXUIR	Transmit FIFO underflow interrupt raw status 0：transmit FIFO underflow interrupt is not active prior masking 1：transmit FIFO underflow interrupt is active prior masking
6	XRXORIS	XIP receive FIFO overflow raw interrupt status 0：XIP receive FIFO overflow raw interrupt is not active prior masking 1：XIP receive FIFO overflow raw interrupt is active prior masking
5	MMCRIS	Multi-master contention raw interrupt status 0：multi-master contention raw interrupt is not active prior masking 1：multi-master contention raw interrupt is active prior masking
4	RXFFRIS	Receive FIFO full raw interrupt status 0：receive FIFO full raw interrupt is not active prior masking 1：receive FIFO full raw interrupt is active prior masking
3	RXFORIS	Receive FIFO overflow raw interrupt status 0：receive FIFO overflow raw interrupt is not active prior masking 1：receive FIFO overflow raw interrupt is active prior masking
2	RXFURIS	Receive FIFO underflow raw interrupt status 0：receive FIFO underflow raw interrupt is not active prior masking

Bit Field	Name	Description
		1：receive FIFO underflow raw interrupt is active prior masking
1	TXFORIS	Transmit FIFO overflow raw interrupt status 0：transmit FIFO overflow raw interrupt is not active prior masking 1：transmit FIFO overflow raw interrupt is active prior masking
0	TXFERIS	Transmit FIFO empty raw interrupt status 0：transmit FIFO empty raw interrupt is not active prior masking 1：transmit FIFO empty raw interrupt is active prior masking

28.5.16 xSPI Transmit FIFO Error Interrupt Clear Registers （XSPI_TXEICR_CLR）

Address offset：0x38

Reset value：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															TXEICR
															rc

Bit Field	Name	Description
31:1	Reserved	Reserved, the reset value must be maintained
0	TXEICR	Clear transmit FIFO overflow/underflow interrupt. This register reflects the status of the interrupt. A read from this register clears the transmit FIFO overflow/underflow interrupt, writing has no effect

28.5.17 xSPI Receive FIFO Overflow Interrupt Clear Register(XSPI_RXFOI_CLR）

Address offset：0x3C

Reset value：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															RXFOIC
															rc

Bit Field	Name	Description
31:1	Reserved	Reserved, the reset value must be maintained
0	RXFOIC	Clear receive FIFO overflow interrupt. This register reflects the status of the interrupt. A read from this register clears the receive FIFO overflow interrupt, writing has no effect.

28.5.18 xSPI Receive FIFO Underflow Interrupt Clear Register （XSPI_RXFUI_CLR）

Address offset：0x40

Reset value：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															RXFUIC
															rc

Bit Field	Name	Description
31:1	Reserved	Reserved, the reset value must be maintained
0	RXFUIC	Clear receive FIFO underflow interrupt This register reflects the status of the interrupt. A read from this register clears the receive FIFO underflow interrupt, writing has no effect.

28.5.19 xSPI Multi-Master Interrupt Clear Register（XSPI_MMC_CLR）

Address offset：0x44

Reset value：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															MMCIC
															rc

Bit Field	Name	Description
31:1	Reserved	Reserved, the reset value must be maintained
0	MMCIC	Clear multi-master contention interrupt. This register reflects the status of the interrupt. A read from this register clears the multi-master contention interrupt, writing has no effect.

28.5.20 xSPI Interrupt Clear Register（XSPI_ICLR）

Address offset：0x48

Reset value：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															INTC
															rc

Bit Field	Name	Description
31:1	Reserved	Reserved, the reset value must be maintained
0	INTC	Clear Interrupts. This register is set if any of the interrupts below are active. A read clears the transmit

Bit Field	Name	Description
		FIFO overflow/receive FIFO overflow/receive FIFO underflow/ multi-master contention interrupts. Writing to this register has no effect.

28.5.21 xSPI DMA Control Register（XSPI_DMA_CTRL）

Address offset：0x4C

Reset value：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved														TXDMAEN	RXDMAEN
														rw	rw

Bit Field	Name	Description
31:2	Reserved	Reserved, the reset value must be maintained
1	TXDMAEN	Transmit DMA Enable. This bit enables/disables the transmit FIFO DMA channel. 0：Transmit DMA disabled 1：Transmit DMA enabled *Note: Configure when XSPI_EN.XSPIEN is disabled*
0	RXDMAEN	Receive DMA Enable. This bit enables/disables the receive FIFO DMA channel. 0：Receive DMA disabled 1：Receive DMA enabled *Note: Configure when XSPI_EN.XSPIEN is disabled*

28.5.22 xSPI DMA Transmit Data Level Control Register （XSPI_DMATDL_CTRL）

Address offset：0x50

Reset value：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

Reserved

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												DMATDL[3:0]			
												rw			

Bit Field	Name	Description
31:4	Reserved	Reserved, the reset value must be maintained
3:0	DMATDL[3:0]	Transmit Data Level. This bit field controls the level at which a DMA request is made by the transmit logic. It is equal to the watermark level, that is, the transmit FIFO request signal is generated when the number of valid data entries in the transmit FIFO is equal to or below this field value, and TDMAE = 1. *Note: The configuration value needs to be greater than or equal to XSPI_TXFT.TXFTST*

28.5.23 xSPI DMA Receive Data Level Control Register （XSPI_DMARDL_CTRL）

Address offset：0x54

Reset value：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												DMARDL[3:0]			
												rw			

Bit Field	Name	Description
31:4	Reserved	Reserved, the reset value must be maintained
3:0	DMARDL[3:0]	Receive Data Level. This bit field controls the level at which a DMA request is made by the receive logic. The watermark level = DMARDL+1, that is, the receive FIFO request is generated when the number of valid data entries in the receive FIFO is equal to or above this field value + 1, and RDMAE=1.

28.5.24 xSPI Identification Register（XSPI_IDR）

Address offset：0x58

Reset value：0xFFFF 3621

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IDCODE[31:0]															
ro															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IDCODE[31:0]															
ro															

Bit Field	Name	Description
31:0	IDCODE[31:0]	Identification code. The register contains the peripheral's identification code, which is written into the register at configuration time

28.5.25 xSPI Component Version Register（XSPI_VERSION_ID）

Address offset：0x5C

Reset value：0x3130 332A

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
VERSION [31:0]															
ro															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VERSION [31:0]															
ro															

Bit Field	Name	Description
31:0	VERSION [31:0]	Contains the hex representation of the Synopsys component version. Consists of ASCII value for each number in the version, followed by *. For example 31_30_33_2A represents the version 1.03*.

28.5.26 xSPI Data Register（XSPI_DATx）

Address offset：0x60+0x04*x （x=0~31）

Reset value：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATx[31:0]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATx[31:0]															
rw															

Bit Field	Name	Description

31:0	DATx[31:0]	Data Register. When writing to this register, you must right-justify the data. Read data are automatically right-justified.
		Read = Receive FIFO buffer
		Write = Transmit FIFO buffer.
		*Note: There are a total of 36 register addresses at 0x60+x*0x4(x=0-31).*

28.5.27 xSPI RX Sample Delay Register（XSPI_RX_DELAY）

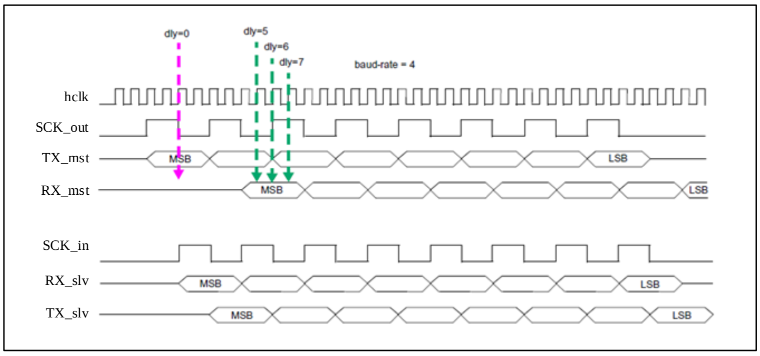
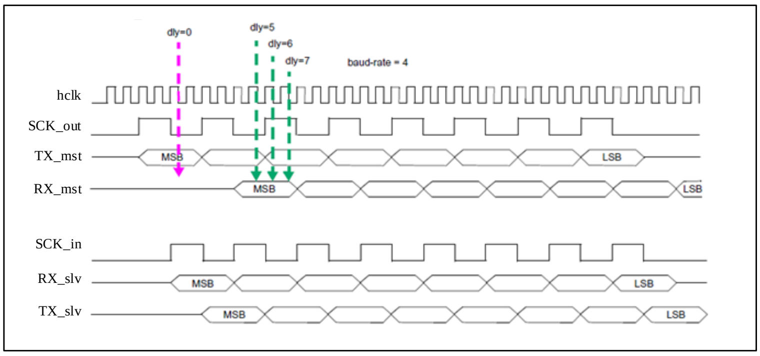
Address offset：0xF0

Reset value：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															SES
															rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								SDCN[7:0]							
								rw							

Bit Field	Name	Description
31:17	Reserved	Reserved, the reset value must be maintained

Bit Field	Name	Description
16	SES	Master Receive data sampling edge. This register is used to decide the sampling edge for RX signal with hclk. 0：positive edge of hclkwill be used to sample the incoming data 1：negative edge of hclk will be used to sample the incoming data
15:8	Reserved	Reserved, the reset value must be maintained
7:0	SDCN[7:0]	Master Receive data (rxd) sample delay. This register is used to delay the sample of the rxd input port. Each value represents a single clock delay on the sample of rxd. If SES=0: Sample delay = SDCN[7:0]*T$_{hclk}$  dly=0 dly=5 dly=6 dly=7 baud-rate = 4 hclk SCK_out TX_mst MSB LSB RX_mst MSB LSB SCK_in RX_slv MSB LSB TX_slv MSB LSB If SES=1： Sample delay = (SDCN[7:0]+0.5)*T$_{hclk}$  dly=0 dly=5 dly=6 dly=7 baud-rate = 4 hclk SCK_out TX_mst MSB LSB RX_mst MSB LSB SCK_in RX_slv MSB LSB TX_slv MSB LSB

28.5.28 xSPI Enhanced SPI Mode Control Register（XSPI_ENH_CTRL0）

This register is used to control the serial data transfer in enhanced SPI mode of operation. This register is only effective when XSPI_CTRL0.SPIFRF ≠ 00. It cannot be written to after enabling XSPI_EN register.

Address offset：0xF4

Reset value：0x0000 0200

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved	CLKSTREN	Reserved					SPIDMEN	Reserved					SPIRXDSEN	WRINDDREN	WRSPIDDREN
	rw						rw						rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WAITCYCLES[4:0]					Reserved	INSTL[1:0]		Reserved		ADDRLEN[3:0]				TRANSTYPE[1:0]	
rw						rw				rw				rw	

Bit Field	Name	Description
31	Reserved	Reserved, the reset value must be maintained
30	CLKSTREN	Enables clock stretching capability in SPI transfers. In case of write, if the FIFO becomes empty xSPI will stretch the clock until FIFO has enough data to continue the transfer. In case of read, if the receive FIFO becomes full xSPI will stop the clock until data has been read from the FIFO.
29:25	Reserved	Reserved, the reset value must be maintained
24	SPIDMEN	SPI data mask enable bit When this bit is enabled, the data mask signal is used to mask the data on the txd data line. 0：Disable data mask 1：Enable data mask
23:19	Reserved	Reserved, the reset value must be maintained
18	SPIRXDSEN	Read data strobe enable bit. Once this bit is set to 1 xSPI will use Read data strobe (rxds) to capture read data in DDR mode. 0：Disable read data strobe 1：Enable read data strobe
17	WRINDDREN	Instruction DDR enable bit This will enable dual-data rate transfer for Instruction phase. 0：disable dual-data rate transfer for Instruction phase.

Bit Field	Name	Description
		1：enable dual-data rate transfer for Instruction phase.
16	WRSPIDDREN	SPI DDR enable bit. This will enable dual-data rate transfers in Dual/Quad/Octal frame formats of xSPI. 0：disable dual-data rate transfers 1：enable dual-data rate transfers
15:11	WAITCYCLES[4:0]	Wait cycles in Dual/Quad/Octal mode between control frames transmit and data reception. Specified as number of SPI clock cycles.
10	Reserved	Reserved, the reset value must be maintained
9:8	INSTL[1:0]	Dual/Quad/Octal mode instruction length in bits. 00：No Instruction 01：4bit 10：8 bit 11：16 bit
7:6	Reserved	Reserved, the reset value must be maintained
5:2	ADDRLEN[3:0]	This bit defines length of address to be transmitted. Only after this much bits are programmed in to the FIFO the transfer can begin. 0x0：No address 0x1：4bit 0x2：8bit 0x3：12bit …… 0xD：52bit 0xE：56bit 0xF：60bit
1:0	TRANSTYPE[1:0]	Address and instruction transfer format. 00：Instruction and address will be sent in standard SPI Mode. 01：Instruction will be sent in standard SPI Mode and Address will be sent in the mode specified by XSPI_CTRL0.SPIFRF 10：Both Instruction and Address will be sent in the mode specified by

Bit Field	Name	Description
		XSPI_CTRL0.SPIFRF. 11：Reserved

28.5.29 xSPI Transmit Drive Edge Register（XSPI_DDR_TXDE）

Address offset：0xF8

Reset value：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								TXDE[7:0]							
								rw							

Bit Field	Name	Description
31:8	Reserved	Reserved, the reset value must be maintained
7:0	TXDE[7:0]	TXD drive edge register which decided the driving edge of transmit data. The maximum value of this register is = (XSPI_BAUD/2) - 1。

28.5.30 xSPI XIP Mode Bits Register（XSPI_XIP_MODE）

Address offset：0xFC

Reset value：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
XIPMDBITS[15:0]															
rw															

Bit Field	Name	Description
31:16	Reserved	Reserved, the reset value must be maintained
15:0	XIPMDBITS[15:0]	XIP mode bits to be sent after address phase of XIP transfer.

28.5.31 xSPI XIP INCR Transfer Opcode Register（XSPI_XIP_INCR_TOC）

Address offset：0x100

Reset value：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ITOC[15:0]															
rw															

Bit Field	Name	Description
31:16	Reserved	Reserved, the reset value must be maintained
15:0	ITOC[15:0]	XIP INCR transfer opcode. When XSPI_XIP_CTRL.XIPINSTEN=1，xSPI sends instruction for all XIP transfers, this register field stores the instruction op-code to be sent when an INCR type transfer is requested on AHB bus. The number of bits to be send in instruction phase is determined by XSPI_ENH_CTRL0.INSTL field.

28.5.32 xSPI XIP WRAP Transfer Opcode Register(XSPI_XIP_WRAP_TOC)

Address offset：0x104

Reset value：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WTOC[15:0]															
rw															

Bit Field	Name	Description
31:16	Reserved	Reserved, the reset value must be maintained
15:0	WTOC[15:0]	XIP WRAP transfer opcode. When XSPI_XIP_CTRL.XIPINSTEN=1，xSPI sends instruction for all XIP transfers, this register field stores the instruction op-code to be sent when an WRAP type transfer is requested on AHB bus. The number of bits to be send in instruction phase is determined by XSPI_ENH_CTRL0.INSTL field.

28.5.33 xSPI XIP Control Register（XSPI_XIP_CTRL）

Address offset：0x108

Reset value：0x0800 0402

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		XIPPREEN	Reserved	XIPMBL[1:0]		RXDSSIGEN	XIPHYPEEN	XIPCTEN	XIPINSTEN	RXDSEN	WRINDDREN	DDREN	DFSHC	WAITCYCLES[4:0]	
		rw		rw		ro	ro	rw	rw	rw	rw	rw	rw	rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WAITCYCLES[4:0]			MDBITSEN	Reserved	INSTL[1:0]		Reserved	ADDRL[3:0]				TRANSTYPE[1:0]		FRF[1:0]	
rw					rw			rw				rw		rw	

Bit Field	Name	Description
31:30	Reserved	Reserved, the reset value must be maintained
29	XIPPREEN	Enables XIP pre-fetch functionality in xSPI. Once enabled xSPI will pre-fetch data frames from next contigous location, to reduce the latency for the upcoming contiguous transfer. If the next XIP request is not contigous then pre-fetched bits will be discarded. 0：Disable XIP pre-fetch functionality 1：Enable XIP pre-fetch functionality
28	Reserved	Reserved, the reset value must be maintained
27:26	XIPMBL[1:0]	XIP mode bits length. Sets the length of mode bits in XIP mode of operation. 00：mode bits length equal to 2 01：mode bits length equal to 4 10：mode bits length equal to 8

Bit Field	Name	Description
		11：mode bits length equal to 16 These bits are valid only when XSPI_XIP_CTRL.MDBITSEN is set to 1.
25	RXDSSIGEN	Enable rxds signaling during address and command phase of hyperbus transfer. If the rxds signal is set to 1 during the CA phase of transfer, xSPI transmits (2* WAITCYCLES-1) wait cycles after the address phase is complete.。 0：disable rxds signaling 1：enable rxds signaling
24	XIPHYPEEN	xSPI hyperbus frame format enable for XIP transfers. This field is effective only when XSPI_CTRL0.FRF is set to 0x00. 0：disable xSPI hyperbus frame format 1：enable xSPI hyperbus frame format
23	XIPCTEN	Enable continuous transfer in XIP mode. If this bit is set to 1 then continuous transfer mode in XIP will be enabled, in this mode xSPI will keep slave selected until a non-XIP transfer is detected on the AHB interface. 0：disable continuous transfer in XIP mode 1：enable continuous transfer in XIP mode
22	XIPINSTEN	XIP instruction enable bit. If this bit is set to 1 then XIP transfers will also have instruction phase. The instruction op-codes will be chosen from XSPI_XIP_INCR_TOC or XSPI_XIP_WRAP_TOC registers bases on AHB transfer type。 0：disable XIP instruction phase 1：enable XIP instruction phase
21	RXDSEN	Read data strobe enable bit. Once this bit is set to 1 xSPI will use Read data strobe (rxds) to capture read data in DDR mode. 0：disable read data strobe 1：enable read data strobe
20	WRINDDREN	Instruction DDR enable bit. This will enable dual-data rate transfer for Instruction phase. 0：disable instruction DDR transfer 1：enable instruction DDR transfer

Bit Field	Name	Description
19	DDREN	xSPI DDR Enable bit. This will enable dual-data rate transfers in Dual/Quad/Octal frame formats of xSPI
18	DFSHC	Fix DFS for XIP transfers. 0：data frame size for XIP transfers will be determined by HSIZE 1：data frame size for XIP transfers will be fixed to the programmed value in XSPI_CTRL0.DFS.
17:13	WAITCYCLES[4:0]	Wait cycles in Dual/Quad/Octal mode between control frames transmit and data reception. Specified as number of SPI clock cycles.
12	MDBITSEN	Mode bits enable in XIP mode. If this bit is set to 1, then in XIP mode of operation xSPI will insert mode bits after the address phase. These bits are set in register XSPI_XIP_MODE register。 0：disable mode bits 1：enable mode bits
11	Reserved	Reserved, the reset value must be maintained
10:9	INSTL[1:0]	Dual/Quad/Octal mode instruction length in bits 00：No Instruction 01：4bit instruction length 10：8 bit instruction length 11：16 bit instruction length
8	Reserved	Reserved, the reset value must be maintained
7:4	ADDRL[3:0]	This bit defines length of address to be transmitted. Only after this much bits are programmed in to the FIFO the transfer can begin. 0x0：no address 0x1：4bit address length 0x2：8bit address length 0x3：12bit address length …… 0xE：56bit address length 0xF：60bit address length

Bit Field	Name	Description
3:2	TRANSTYPE[1:0]	Address and instruction transfer format. 00：Instruction and address will be sent in Standard SPI Mode. 01：Instruction will be sent in standard SPI mode and address will be sent in the mode specified by XSPI_XIP_CTRL.FRF[1:0]. 10：Both instruction and address will be sent in the mode specified by XSPI_XIP_CTRL.FRF[1:0] 11：reserved
1:0	FRF[1:0]	SPI frame format 00：reserved 01：dual SPI format 10：quard SPI format 11：octal SPI format

28.5.34 xSPI XIP Slave Enable Register（XSPI_XIP_SLAVE_EN）

This register is valid only when the XSPI_EN.XSPIEN is enabled，the register enables the individual slave select output lines from the xSPI master for XIP mode of operation

Address offset:0x10C

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved														SEN[1:0]	
														rw	

Bit Field	Name	Description
31:2	Reserved	Reserved, the reset value must be maintained
1:0	SEN[1:0]	Slave Select Enable Flag. Each bit in this register corresponds to a slave select line from the xSPI master. When a bit in this register is set to 1, the corresponding slave select line from the master is activated

Bit Field	Name	Description
		when a XIP transfer begins. It should be noted that setting or clearing bits in this register have no effect on the corresponding slave select outputs until a XIP transfer is started. Before beginning a transfer, you should enable the bit in this register that corresponds to the slave device with which the master wants to communicate. When not operating in broadcast mode, only one bit in this field should be set.

28.5.35 xSPI XIP Receive FIFO Overflow Interrupt Clear Register （XSPI_XIP_RXFOI_CLR）

Address offset：0x110

Reset value：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															XRXFOIC
															rc

Bit Field	Name	Description
31:1	Reserved	Reserved, the reset value must be maintained
0	XRXFOIC	Clear XIP receive fifo overflow interrupt. This register reflects the status of the interrupt. A read from this register clears the XIP receive fifo overflow interrupt, writing has no effect.

28.5.36 xSPI XIP Time Out Register for Continuous Transfers （XSPI_XIP_TOUT）

Address offset：0x114

Reset value：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								XTOUT[7:0]							
								rw							

Bit Field	Name	Description
31:8	Reserved	Reserved, the reset value must be maintained
7:0	XTOUT[7:0]	XIP time out value in terms of hclk. Once slave is selected in continuous XIP mode this counter will be used to de-select the slave if there is no request for the time specified in the counter.

28.5.37 xSPI XIP Write INCR transfer opcode Register （XSPI_XIP_WRITE_INCR_INST）

Address offset：0x140

Reset value：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INCRWRINST[15:0]															
rw															

Bit Field	Name	Description
31:16	Reserved	Reserved, the reset value must be maintained
15:0	INCRWRINST[15:0]	XIP Write INCR transfer opcode. When XIP_WRITE_CTRL.WRINSTL is not equal to 0, xSPI sends instruction for all XIP write transfers, this register field stores the instruction op-code to be sent when an INCR type XIP write transfer is requested on AHB bus. The number of bits to be send in instruction phase is determined by XIP_WRITE_CTRL.WRINSTL field.

28.5.38 xSPI XIP Write Wrap Transfer Opcode Register （XSPI_XIP_WRITE_WRAP_INST）

Address offset：0x144

Reset value：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WRAPWRINST[15:0]															
rw															

Bit Field	Name	Description
31:16	Reserved	Reserved, the reset value must be maintained
15:0	WRAPWRINST[15:0]	XIP Write WRAP transfer opcode. When XIP_WRITE_CTRL.WRINSTL is not equal to 0, xSPI sends instruction for all XIP write transfers, this register field stores the instruction op-code to be sent when an WRAP type XIP write transfer is requested on AHB bus. The number of bits to be send in instruction phase is determined by XIP_WRITE_CTRL.WRINSTL field.

28.5.39 xSPI XIP Write Control Register（XSPI_XIP_WRITE_CTRL）

Address offset：0x148

Reset value：0x0000 0072

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved											XIPWRWCY[4:0]				
											ro				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				WRINDDREN	WRSPIDDREN	WRINSTL[1:0]		WRADDRL[3:0]				WRTRTYPE[1:0]		WRFRF[1:0]	
				rw	rw	rw		rw				rw		rw	

Bit Field	Name	Description
31:21	Reserved	Reserved, the reset value must be maintained
20:16	XIPWRWCY[4:0]	Wait cycles in dual/quad/octal mode between control frames transmit and data reception. Specified as number of SPI clock cycles.
15:12	Reserved	Reserved, the reset value must be maintained
11	WRINDDREN	Instruction DDR Enable bit. This will enable dual-data rate transfer for Instruction phase
10	WRSPIDDREN	SPI DDR enable bit. This will enable dual-data rate transfers in dual/quad/octal frame formats of SPI
9:8	WRINSTL[1:0]	dual/quad/octal mode instruction length in bits. 00：no instruction 01：4 bit instruction length 10：8 bit instruction length 11：16 bit instruction length
7:4	WRADDRL[3:0]	This bit defines length of address to be transmitted. Only after this much bits are programmed in to the FIFO the transfer can begin. 0000：reserved 0001：4 bit address length 0010：8 bit address length 0011：12 bit address length 0100：16 bit address length 0101：20 bit address length 0110：24 bit address length 0111：28 bit address length 1000：32 bit address length
3:2	WRTRTYPE[1:0]	Address and instruction transfer format. 00：Instruction and address will be sent in standard SPI Mode. 01：Instruction will be sent in standard SPI mode and address will be sent in the mode specified by XIP_WRITE_CTRL.WRFRF 10：Both instruction and address will be sent in the mode specified by XIP_WRITE_CTRL.WRFRF 11：reserved

Bit Field	Name	Description
1:0	WRFRF[1:0]	SPI frame format 00：reserved 01：dual SPI format 10：quard SPI format 11：octal SPI format

29 Secure Algorithm Co-processor (SAC)

The device features an embedded algorithm hardware acceleration engine, it supports a variety of international algorithms hash cryptographic algorithm acceleration, which can greatly improve the encryption and decryption speed compared with pure software algorithms.

The hardware supports the following algorithms :

- Supports DES symmetric algorithm
  - DES and 3DES encryption and decryption operations are supported
  - TDES supports 2KEY and 3KEY modes.
  - Support CBC and ECB modes
- Supports the SYMMETRIC AES algorithm
  - 128bit/192bit/ 256bit key length is supported
  - Supports CBC, ECB and CTR modes
- Supports SM4 symmetric algorithm
  - Supports CBC, ECB modes
- Supports SHA hash algorithm
  - Supports SHA1/SHA224/SHA256
- Supports MD5 digest algorithm
- Supports SM3 hash algorithm.
- Support for random number generation

Note: 1. The SAC module operating clock is up to 180MHz, the HCLK frequency must not exceed 180MHz to avoid SAC module arithmetic anomalies, and RCC_CFG2.HCLKPRES must be configured to 0 (no crossover frequency).

2.For performance and use of cryptographic algorithms, please contact Nsing Technologies sales representativesl.

30 Digital Video Peripheral（DVP）

DVP is a flexible and powerful CMOS optical sensor interface that can easily meet customers' image acquisition requirements, and the entire acquisition process does not require CPU intervention.

DVP is capable of receiving high-speed data from traditional or ITU-R BT.656 format CMOS image sensors, and supports their data formats: YCbCr422 and RGB565 progressive, as well as compressed data (JPEG).

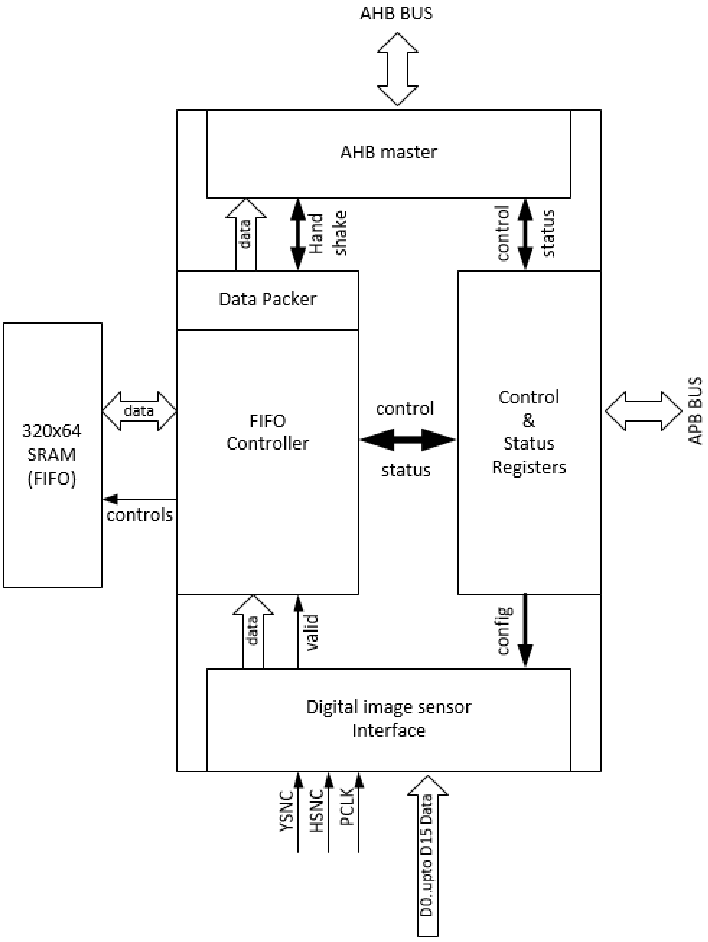
30.1 Main Features

- Hardware acquisition method
- Pure input interfaces
- Support 8, 10, 12, and 16-bit traditional synchronous parallel interface
- Support 8-bit and 10-bit ITU-R BT.656 video format
- Support 8-bit and 16-bit data port for YCbCr, YUV and RGB data format
- Support 8-, 10-, and 16-bit data port for Bayer data format
- Support clock output (through MCO output, typical value 48MHz) to provide clock to external CMOS optical sensor
- The polarities of the input pixel clock (DVP_PCLK), field synchronization signal (DVP_VSYNC) and line synchronization signal (DVP_HSYNC) can be configured independently.
- 16x4 bytes FIFO for receiving pixel data
- Support Receive FIFO overflow protection
- Support DMA function, the entire image acquisition process does not require CPU intervention.
- The image size for acquisition must be a multiple of 4 bytes.
- Supports hardware inversion of the acquired image data.
- Support up to 1280 * 720 @30Hz
- Support hardware cropping
- Support multiple data formats:
  - YCbCr422 progressive video
  - RGB565 progressive video

- Compressed data (JPEG)

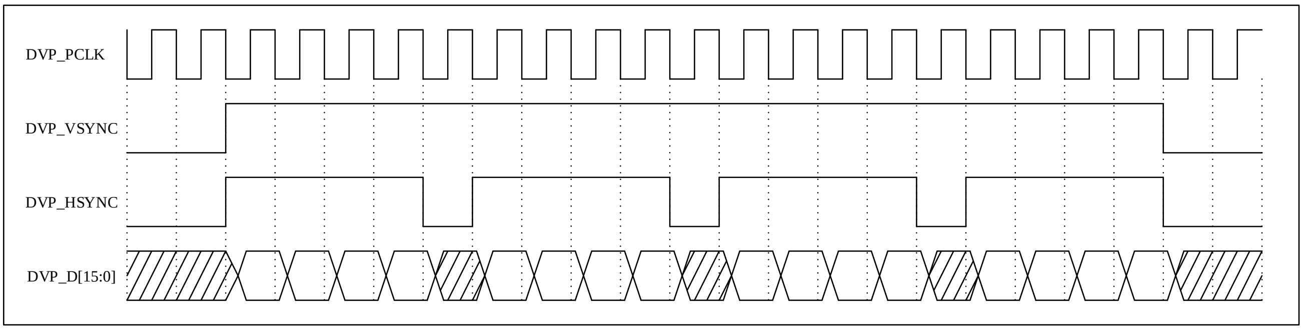
30.2 DVP Block Diagram

Figure 30-1 DVP Block Diagram



30.3 DVP Timing

Figure 30-2 Example of DVP Interface Timing



As shown in the above figure:

- DVP_PCLK is the pixel clock, capturing 2 bytes (16 bits) of valid data per clock cycle.

- DVP_VSYNC is the vertical synchronization (frame synchronization) signal, active high.
- DVP_HSYNC is the horizontal synchronization signal, active high.
- When both DVP_VSYNC and DVP_HSYNC are high, the captured data is valid.
- There is at least one pixel clock cycle interval between every two lines.
- According to the timing diagram above, users need to configure DVP_VSYNC and DVP_HSYNC as active high in the DVP module, and capture on the falling edge of DVP_PCLK to correctly receive data.
- DVP data is only valid when the DVP port enable bit (DVP_CTRL.DVPEN) is set to 1, and the enable bit must be set to 1 at least 4 pixel clock cycles before the DVP_VSYNC active signal (high level) to prevent the current frame from being discarded.

Note: In the above diagram, the DVP_VSYNC and DVP_HSYNC signals are active high, but in actual applications, they may be active low. The signal polarity needs to be configured in the DVP module according to the actual situation.

30.4 DVP Clock

DVP has three clock inputs: AHB interface clock, APB interface clock, and image sensor interface clock. The AHB interface clock and APB interface clock are synchronous clocks. The image sensor interface clock is an asynchronous clock with respect to other clocks.

The AHB interface clock (hclk) must be equal to or faster than the pixel clock.

30.5 DVP Function Description

30.5.1 Configuration Process

1) Switch on the CMOS optical sensor clock, enable the relevant control ports (typically I2C interface), and configure the sensor parameters.
2) Configure the DVP port and parameters (e.g., capture mode, DMA, etc.).
3) Configure the DVP port enable bit (DVP_CTRL.DVPEN), and prepare to receive data.
4) Enable the CMOS sensor to start sending data.

30.5.2 Data Transmission And Synchronization

The DVP module can receive 8-bit, 10-bit, 12-bit, 14-bit, or 16-bit parallel pixel data. The effective data position in the interface can be configured based on the DBIT[2:0] setting in the DVP_PORTCFG register, as shown in the table below.

Table 30-1 DVP Interface Signals

Port	DVP_DBIT[2:0]	Description

DVP_D[7:0]	000 (8 bits)	pixel data bus, all bits are used to capture data
DVP_D[9:0]	001 (10 bits)	
DVP_D[11:0]	010 (12 bits)	
DVP_D[13:0]	011 (14 bits)	
DVP_D[15:0]	100 (16 bits)	
DVP_D[9:0]	101	pixel data bus, only [9:2] (8-bit) are used to capture data
DVP_D[11:0]	110	pixel data bus, only [11:2] (10-bit) are used to capture data
DVP_D[11:0]	111	pixel data bus, only [11:4] (8-bit) are used to capture data
DVP_VSYNC	N/A	vertical synchronization signal
DVP_HSYNC	N/A	horizontal synchronization signal
DVP_PCLK	N/A	pixel clock

The polarity of DVP_VSYNC, DVP_HSYNC, and DVP_PCLK can be configured through the DVP_PORTCFG register.

To integrate the transmitted data into an image, horizontal synchronization and vertical synchronization are required. The digital camera interface supports embedded synchronization or hardware (HSYNC and VSYNC) synchronization. When using embedded synchronization, the digital camera module ensures that the values 0x00 and 0xFF are used solely for synchronization (not in the data).

30.5.2.1 Hardware Synchronization

In hardware synchronization mode, the CMOS image camera can provide horizontal sync signals and vertical sync signals for synchronization at the receiving end. Setting DVP_PORTCFG.EMBSEN to 0 allows the DVP to receive in this synchronous mode. Each effective interval of the DVP_VSYNC signal represents a frame of data, and each effective interval of the DVP_HSYNC signal represents a line of data. The DVP_VSYNC signal is low active by default, and the DVP_HSYNC signal is high active by default. When both DVP_VSYNC and DVP_HSYNC are active, valid pixel data is captured at the rising edge of the pixel clock.

30.5.2.2 Embedded Synchronization

In embedded synchronization mode, pixel data synchronization does not depend on DVP_VSYNC and DVP_HSYNC, but rather inserts synchronization codes in the data stream. The DVP can receive in this synchronous mode. The synchronization code (0xFF0000XX) consists of 4 bytes of data, with the first 3 bytes fixed, where the first byte is all 1s, followed by all 0s for the second and third bytes. The content of the fourth byte data varies based on the synchronization information. Configuration of the HS_PTTN, HE_PTTN, VS_PTTN, and VE_PTTN registers in the DVP_EMSC register is required according to the synchronization code format and content used by the camera manufacturer. Embedded code synchronization has two modes, namely normal mode and Sony mode, with specific processes as shown in the diagram below.

Figure 30-3 Frame Format in Normal Mode

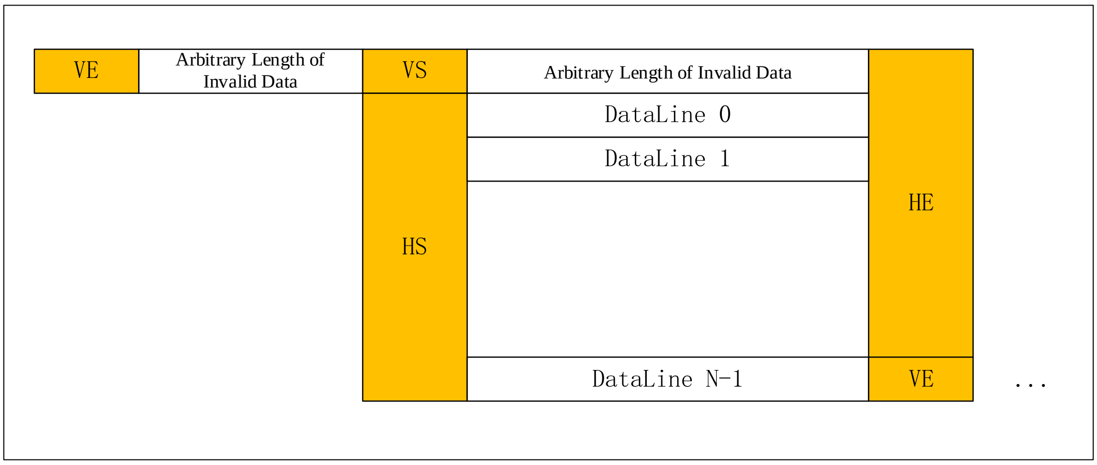
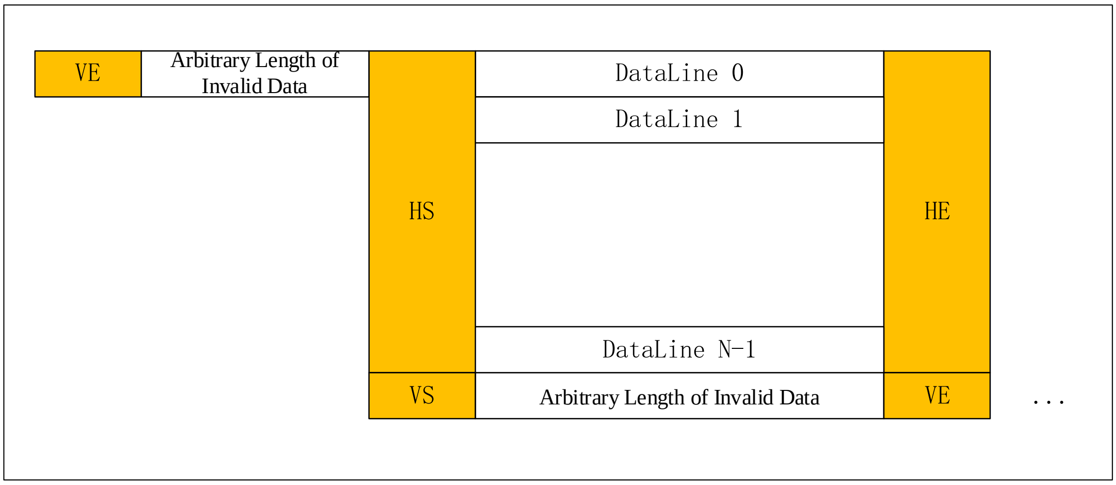

VE
Arbitrary Length of Invalid Data
VS
Arbitrary Length of Invalid Data
DataLine 0
DataLine 1
HS
HE
DataLine N-1
VE
. . .

Figure 30-3 Frame Format in Sony Mode


VE
Arbitrary Length of Invalid Data
DataLine 0
DataLine 1
HS
HE
DataLine N-1
VS
Arbitrary Length of Invalid Data
VE
. . .

Embedded synchronization mode only supports 8-bit, 10-bit, and 12-bit data modes.

30.5.3 DVP Capture Mode

There are two capture modes:one is continuous capturing mode, and the other one is snap-shot mode.While receiving an image frame, if FIFO overflow occurs, the overflow flag of the corresponding receiving frame buffer is set, and if a frame is skipped, the frame-skipping flag will get set too.

30.5.3.1 Snap-Shot Mode (Single Frame Capture)

Enable this mode by setting DVP_CTRL.CM register bit to “1”. In snap-shot mode, it depends on the number of the frame buffer getting configured. If only one frame buffer is in place, DVP will stop capturing a new frame after a frame is received. And at the same time, the hardware sets the DVP_INTSTS.M1TCF bit to "1" to indicate that the hardware has completed the current frame reception. For the DVP to start capturing a new frame again, the software has to clear DVP_INTSTS.M1TCF. Similarly, if two frame buffers are in place, the DVP will stop capturing frame after two frames are received, and the DVP_INTSTS.M1TCF and the DVP_INTSTS.M2TCF bits are pending for the software to clear them. The software must clear the flags in the order as they are received. Once the software clears one of the flags, the DVP starts capturing frame again.

30.5.3.2 Continuous Capturing Mode

In this mode, DVP will continue to capture image data and put it into frame memory regardless of whether the previous frame data is being read. It does not wait for the software to clear the completion Flag (DVP_INTSTS. MxRCF). If two frame buffers are set up, frames are received into these frame buffers in ping-pong fashion.

To enable this mode, make sure DVP_CTRL.CMregister bit to “0”.

30.5.4 DVP Crop Function

Set DVP_CTRL.CROPEN bit to "1" to enter the crop function. For this function, a user only needs to provide two coordinates to DVP, the start and end coordinates. The user must enter the coordinates first before setting the CROPEN bit. Each image pixel can be represented by 1 or 2 data bytes. If the image pixel needs two data bytes, then DVP_PORTCFG. PIXELDB bit has to be "1". DVP crop function is illustrated in Figure 30-5 below.

The user must enter the coordinates first before setting the CROPEN bit. Users needs to set the following registers:

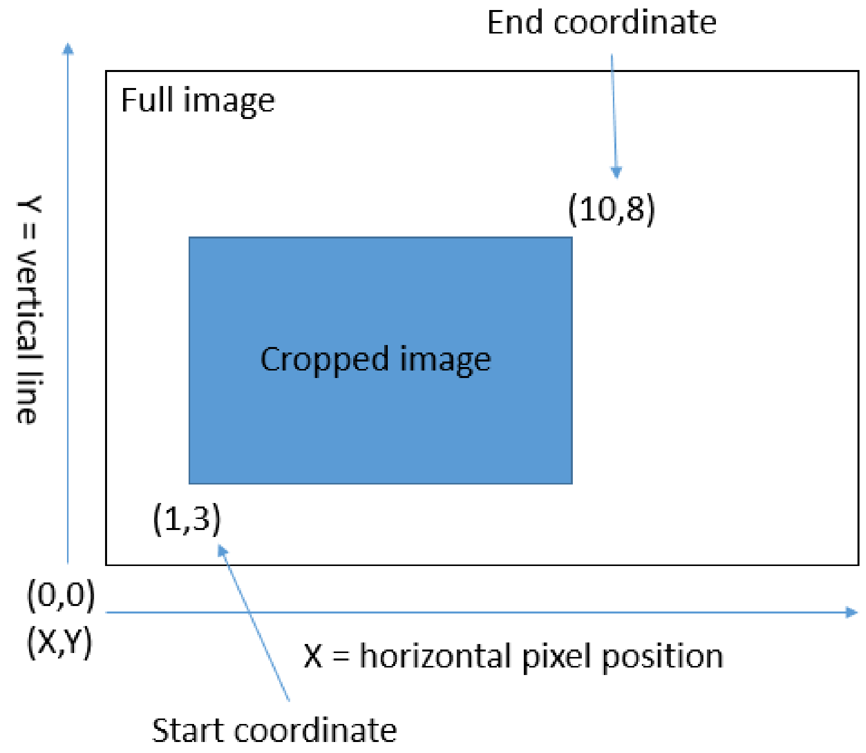
Start coordinate register: DVP_CSXY.CSTAX = 'd1

Start coordinate register: DVP_CSXY.CSTAY = 'd3

End coordinate register: DVP_CEXY.CENDX = 'd10

End coordinate register: DVP_CEXY.CENDY = 'd8

Figure 30-5 DVP Crop Function



30.5.5 DVP Frame Line Skipping Function

The frame line skipping function is controlled by two registers:the DVP_PORTCFG.HISKIP register controls the number of initial frame line skip. The DVP_PORTCFG.HRSKIP register controls line skip pattern.

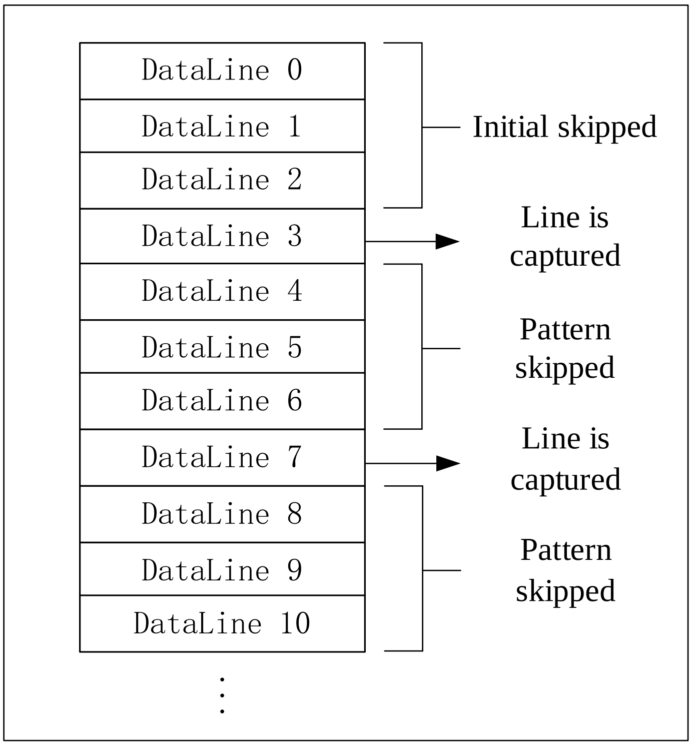
There are two types of line skipping pattern controlled by the DVP_PORTCFG.HRSKIP register:

1) capture all even or odd lines

2) capture a line followed by the number of skip lines.

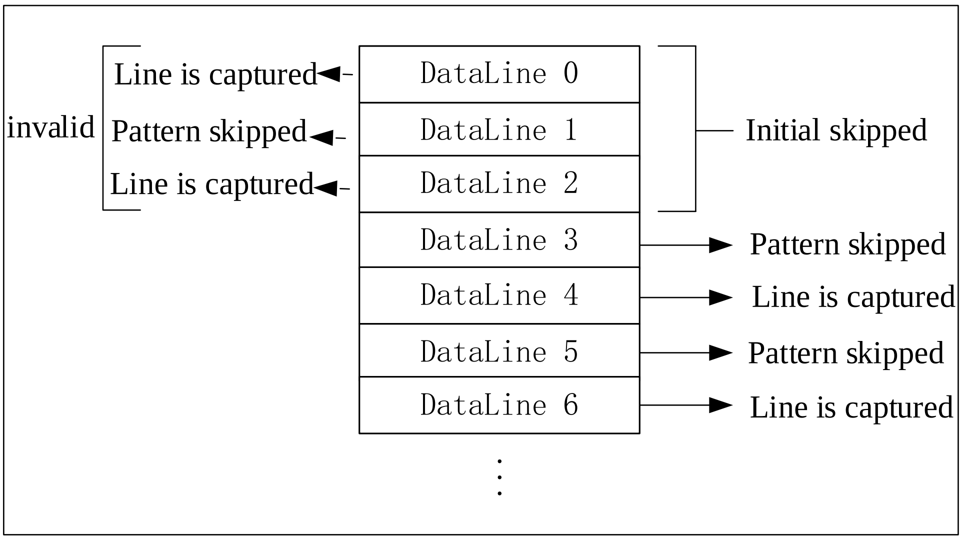
When these two registers are used in combination, there are two possible scenarios. For example, when configuring DVP_PORTCFG.HISKIP as 3 and DVP_PORTCFG.HRSKIP as 3, the actual situation is as shown in the diagram below.

Figure 30-6 Line Skip Function(HISKIP=3，HRSKIP=3)



When configuring DVP_PORTCFG.HISKIP as 3 and DVP_PORTCFG.HRSKIP as 14, the actual situation is as shown in the diagram below.

Figure 30-7 Line Skip Function(HISKIP=3，HRSKIP=14)



30.5.6 FIFO And DMA

The DVP module has a embedded DMA, which functions to transfer data from the FIFO to the corresponding frame buffer 1/2 when the data in the FIFO reaches a threshold. The TXBURSZ in the DVP_FIFOCFG register controls the amount of data transferred in a single DMA transfer, and burst transfer is used when the configured value is greater than 0. It is important to note that the configured amount of data for a single transfer must be greater than or equal to the FIFO transfer threshold. The M1ADDREN/M2ADDREN in the DVP_FIFOCFG register enable the frame buffer, and in conjunction with the information in the DVP_SMADDR1, DVP_SMADDR2, and DVP_FBS registers, the DMA can determine the address for transferring each data entry.

30.5.7 Interrupt

There are two registers related to interrupts: DVP_INTSTS and DVP_INTEN:

- DVP_INTEN is the interrupt enable register.
- DVP_INTSTS is the interrupt status register. Even if the interrupt is not enabled, the interrupt status will change, but the interrupt will not be reported to the system. The corresponding interrupt will only be reported when the corresponding interrupt enable bit in DVP_INTEN is enabled.
- Before the user wants to use a specific interrupt, they must first clear the corresponding flag in the DVP_INTSTS register (write 0 to clear) to avoid generating incorrect interrupts.

30.6 DVP Register

30.6.1 DVP Register Overview

Table 30-2 DVP Register Overview

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	DVP_CTRL	VBFLT [3:0]				Reserved																			CROPEN	Reserved						CM	DVPEN
	Reset value	1	0	0	0																				0							0	0
0x04	DVP_INTEN	Reserved												FMSIE	SERRIE	CERRIE	SKIPIE	Reserved							MOIE	AHBERRIE	Reserved	FOIE	Reserved	M2TCIE	M1TCIE	M2SIE	M1SIE
	Reset value													b	0	0	0								0	0		0		0	0	0	0
0x08	DVP_INTSTS	CROPENS	CMS	Reserved					DVPDIS	Reserved				FMSF	SERRF	CERRF	SKIPF	Reserved						M2OF	M1OF	AHBERR2F	AHBERR1F	M2FOF	M1FOF	M2TCF	M1TCF	M2SF	M1SF
	Reset value	0	0						1					0	0	0	0							0	0	0	0	0	0	0	0	0	0
0x0C	DVP_PORTCFG	HRSKIP[3:0]				HISKIP[3:0]				BMAP[7:0]								Reserved			PIXELDB	Reserved	DBIT[2:0]			EMBSEN	SLC	Reserved	DATRVS	DATINV	VSPOL	HSPOL	PCLKPOL
	Reset value	0	0	0	0	0	0	0	0	1	1	1	0	0	1	0	0				0		0	0	0	0	0		0	0	0	0	0
0x10	DVP_FIFOCFG	Reserved													TXBURSZ[2:0]			Reserved					TXFTH[1:0]			Reserved		EN1KBD	DISPMODE	Reserved		M2ADDREN	M1ADDREN
	Reset value														0	0	0						0	0	0			0	0			0	0

0x14	DVP_SMADDR1	MADDR[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x18	DVP_SMADDR2	MADDR[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x1C	DVP_FBS	Reserved										FMSIZE[21:0]																					
	Reset value											0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x20	DVP_FPBC1	PIXLC[21:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x24	DVP_FPBC2	PIXLC[21:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x28	DVP_CSXY	Reserved					CSTAY[10:0]											Reserved				CSTAX[11:0]											
	Reset value						0	0	0	0	0	0	0	0	0	0	0					0	0	0	0	0	0	0	0	0	0	0	0
0x2C	DVP_CEXY	Reserved					CENDY[10:0]											Reserved				CENDX[11:0]											
	Reset value						0	0	0	0	0	0	0	0	0	0	0					0	0	0	0	0	0	0	0	0	0	0	0
0x30	DVP_EMSC	HE_PTTN [7:0]								HS_PTTN [7:0]								VE_PTTN [7:0]								VS_PTTN [7:0]							
	Reset value	1	0	0	1	1	1	0	1	1	0	0	0	0	0	0	0	1	0	1	1	0	1	1	0	1	0	1	0	1	0	1	1
0x34	DVP_EMSCM	HE_MASK [7:0]								HS_MASK [7:0]								VE_MASK [7:0]								VS_MASK [7:0]							
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

30.6.2 DVP Control Register（DVP_CTRL）

Address Offset：0x00

Reset value：0x8000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

VBDP[3:0]	Reserved
rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved							CROPEN	Reserved						CM	DVPEN
							rw							rw	rw

Bit Field	Name	Description
31:28	VBFLT[3:0]	VSYNC blanking glitch filtering control. By default, it is set to 8. It means 8 consecutive sampling on VSYNC signal and all the samples are "1" which confirms the vertical blanking is detected. You can change the number of detection sampling of VSYNC to confirm the vertical blanking by programming this regiser, and it ranges from 0 to 15. Note: The software can only modify this bit when DVP_CTRL.DVPEN.
27:9	Reserved	Reserved, the reset value must be maintained
8	CROPEN	Cropping Feature Enable bit. 0: no cropping, normal mode. 1: cropping mode. Before enabling this function, the start and end of cropping coordinates need to be specified respectively in the register DVP_CSXY and DVP_CEXY.
7:2	Reserved	Reserved, the reset value must be maintained
1	CM	Capture Mode Control. 0: continuous capture mode. 1: single capture mode. In the continuous capture mode, and if it is configured to have two frame buffers the frames are continuously received into the two buffers in alternative manner. In this mode, the buffer does not wait for DVP_INTSTS.MxRCF(x=1,2) to be clear before it can be used to receive a new frame again. If there is a bus error occurred on one of the buffers the other buffer will continue to receive all the new frames until the error is removed then they resume to receive the frames alternatively. In the single capture mode, it is recommended to use single frame buffer. Single frame buffer enabled (M1): when the frame buffer is completed with a frame, i.e. DVP_INTSTS.M1TCF is set, it waits for DVP_INTSTS.M1TCF to be clear before it starts to capture a new frame again. Two frame buffers are enabled (M1 and M2): When the current frame buffer finishes transferring a frame and the DVP_INTSTS.MxRCF (x=1,2) of the other buffer is not set, the next frame will be transferred to the other buffer and the DVP_INTSTS.MxRCF (x=1,2) of that buffer will be set. If both DVP_INTSTS.MxRCF (x=1,2) flags are set to 1, the system will wait until one of the flags is cleared.
0	DVPEN	DVP_ENB

Bit Field	Name	Description
		DVP port enable bit. Write "1" to this bit which enables DVP function. Note: This is a Write-1-Only bit; writing "0" to this bit has no effect. Refer to the register, DVP_INTSTS.DVPDIS bit for disabling this function.

30.6.3 DVP Interrupt Enable Register（DVP_INTEN）

Address Offset：0x04

Reset value：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved												FMSIE	SERRIE	CERRIE	SKIPIE
												rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved							FBOIE	AHBERRIE	Reserved	FOIE	Reserved	M2TCIE	M1TCIE	M2SIE	M1SIE
							rw	rw		rw		rw	rw	rw	rw

Bit Field	Name	Description
31:20	Reserved	Reserved, the reset value must be maintained
19	FMSIE	Start of Frame detection interrupt Enable bit. 1: send an interrupt when start of frame (Vsync or equivalent vsync in embedded mode) is detected at the DVP port. 0: don't send an interrupt when start of frame is detected at the DVP port.
18	SERRIE	SYNC Code error sequence detected interrupt enable bit. 1: send an interrupt when a SYNC Code error sequence is detected. 0: don't send an interrupt when a SYNC Code error sequence is detected. Note: this is only applied to embedded synchronization mode.
17	CERRIE	SYNC Code mismatch detected interrupt enable bit. 1: send an interrupt when a SYNC Code mismatch is detected. 0: don't send an interrupt when a SYNC Code mismatch is detected. Note: This is only applied to embedded synchronization mode.
16	SKIPIE	Frame skipping interrupt enable bit. 1: send an interrupt when a frame skipping is detected. 0: don't send an interrupt when a frame skipping is detected. This occurs when the data FIFO is over flow or no more available frame buffer to receive a new frame. At this time, a new frame will be dropped and a frame skip flag will be set.

Bit Field	Name	Description
15:9	Reserved	Reserved, the reset value must be maintained
8	MOIE	Frame Buffer Over Flow interrupt enable bit. 1: send an interrupt when frame buffer overflow event occurred. 0: don't send an interrupt when frame buffer overflow event occurred.
7	AHBERRIE	AHB bus error response interrupt enable bit. 1: send an interrupt when the AHB master receives an error response while sending data to frame memory. 0: don't send an interrupt when the above situation is occurred.
6	Reserved	Reserved, the reset value must be maintained
5	FOIE	Data FIFO Over Flow interrupt enable it. 1: send an interrupt when data FIFO overflow occurs. 0: don't send an interrupt when data FIFO overflow occurs.
4	Reserved	Reserved, the reset value must be maintained
3	M2TCIE	Image data to frame buffer 2 transfer complete interrupt enable bit. 1: send an interrupt when data to frame buffer 2 is completed. 0: no interrupt is sent when data to frame buffer 2 is completed.
2	M1TCIE	Image data to Frame Buffer 1 transfer complete interrupt enable bit. 1: send an interrupt when data to frame buffer 1 is completed. 0: no interrupt is sent when data to frame buffer 1 is completed.
1	M2SIE	Frame Buffer 2 is started to receive image data interrupt enable bit. 1: send an interrupt when frame buffer 2 is starting to receive data 0: don't send an interrupt when frame buffer 2 is starting to receive data
0	M1SIE	Frame Buffer 1 is started to receive image data interrupt enable bit. 1: send an interrupt when frame buffer 1 is starting to receive data 0: don't send an interrupt when frame buffer 1 is starting to receive data

30.6.4 DVP Interrupt Flag and Status register（DVP_INTSTS）

Address Offset：0x08

Reset value：0x0100 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CROPENS	CM CTRLS	Reserved					DVPDIS	Reserved				SFDF	SERRF	CERRF	SKIPF
r	r						rs					rc_w1	rc_w1	rc_w1	rc_w1

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						M2OF	M1OF	AHBER R2F	AHBER R1F	M2FOF	M1FOF	M2TCF	M1TCF	M2SF	M1SF
						rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1

Bit Field	Name	Description
31	CROPENS	This is the DVP_CTRL.CROPEN shadow bit. It is read only. This bit is conveniently placed here for the software to access while it is accessing to other status bits. It will save the software to read the configuration register separately again.
30	CMS	This is the DVP_CTRL.CM shadow bit. It is read only. This bit is conveniently placed here for the software to access while it is accessing to other status bits. It will save the software to read the configuration register separately again.
29:25	Reserved	Reserved, the reset value must be maintained
24	DVPDIS	This is DVP port disable bit. Write "1" to this bit it disables DVP port. Write "0" to this bit no effect. Note: After writing "1" to disable DVP port it may not immediately stops; it depends on the state of the hardware at that time. Logic "1" will be returned to this bit when the hardware is completely stopped. To ensure the hardware is stopped the software should check this bit whether logic "1" is returned. When this bit returns to 0, it means that the DVP is in an active state.
23:20	Reserved	Reserved, the reset value must be maintained
19	FMSF	Start of frame detected flag. This bit is set to "1" when start of frame (Vsync or equivalent vsync in embedded mode) is detected. Write "0" to this bit no effect. Write "1" to this bit it clears this flag. Note: It starts to detect start of frame only after DVP_CTRL.DVPEN is set to '1'
18	SERRF	SYNC code sequence error detected flag. This bit is only used in embedded synchronization mode. This bit is set to "1" when a SYNC code sequence error is detected while it is in image active region, and it will not detect this error while it is in vertical blanking region. If it is set and its corresponding DVP_INTEN.SERRIE =1， then an interrupt will be sent. Write "1" to this bit to clear. Write "0" no effect.
17	CERRF	sync code mismatch error detected flag. This bit is only used in embedded synchronization mode. This bit is set to "1" when a SYNC code mismatch is detected while it is in image active region, and it will not detect this error while it is in vertical blanking region. If it is set and its corresponding DVP_INTEN.CERRIE=1， then an interrupt will be sent. Write "1" to this bit to clear. Write "0" no effect.
16	SKIPF	Frame skipping detected flag.

Bit Field	Name	Description
		This bit is set to "1" when there is no more FIFO space or frame buffer memory to receive a new frame. This bit is only valid when the DVP port is enabled(DVP_CTRL.DVPEN=1). Write "1" to this bit to clear. Write "0" no effect.
15:10	Reserved	Reserved, the reset value must be maintained
9	M2OF	Frame Buffer 2 Over Flow 0: no over flow event 1: over flow event detected When a frame data is exceeded the allocated frame buffer size this flag will be set to "1" and the data will be only written up to the frame buffer size. The remaining data will be dropped. Write "1" to this bit to clear. Write "0" no effect.
8	M1OF	Frame Buffer 1 Over Flow 0: no over flow event 1: over flow event detected When a frame data is exceeded the allocated frame buffer size this flag will be set to "1" and the data will be only written up to the frame buffer size. The remaining data will be dropped. Write "1" to this bit to clear. Write "0" no effect.
7	AHBERR2F	Receiving AHB bus error response on accessing to Frame Buffer 2 flag. This bit is set to "1" when the AHB master receives an error response from the bus while accessing Frame Buffer 2. If this occurs the Frame Buffer 2 will become unavailable until AHBERR2F is clear. Write "1" to this bit to clear. Write "0" no effect.
6	AHBERR1F	Receiving AHB bus error response on accessing to Frame Buffer 1 flag. This bit is set to "1" when the AHB master receives an error response from the bus while accessing Frame Buffer 1. If this occurs the Frame Buffer 1 will become unavailable until AHBERR1F is clear. Write "1" to this bit to clear. Write "0" no effect.
5	M2FOF	Frame Buffer 2 FIFO over flow flag. This bit is set to "1" when the FIFO overflow occurs while receiving frame data for Frame Buffer 2. Write "1" to this bit to clear. Write "0" no effect.
4	M1FOF	Frame Buffer 1 FIFO over flow flag. This bit is set to "1" when the FIFO overflow occurs while receiving frame data for Frame Buffer 1. Write "1" to this bit to clear. Write "0" no effect.
3	M2TCF	Data to Frame Buffer 2 completion flag. This bit is set to "1" when the Frame Buffer 2 is completed on receiving frame data. In single capture mode, this buffer only becomes available again when M2TCF is clear. Write "1" to this bit to clear. Write "0" no effect.

Bit Field	Name	Description
2	M1TCF	Data to Frame Buffer 1 completion flag. This bit is set to "1" when the Frame Buffer 1 is completed on receiving frame data. In single capture mode, this buffer only becomes available again when M1TCF M1_CMPL_FG is clear. Write "1" to this bit to clear. Write "0" no effect.
1	M2SF	Data to Frame Buffer 2 start flag. This bit is set to "1" when the Frame Buffer 2 starts receiving frame data. Write "1" to this bit to clear. Write "0" no effect.
0	M1SF	Data to Frame Buffer 1 start flag. This bit is set to "1" when the Frame Buffer 1 starts receiving frame data. Write "1" to this bit to clear. Write "0" no effect.

30.6.5 DVP Port Configuration Register（DVP_PORTCFG）

Address offset： 0x0C

Reset value： 0x00E4 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
HRSKIP[3:0]				HISKIP[3:0]				BMAP[7:0]							
rw				rw				rw							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			PIXELDB	Reserved	DBIT[2:0]			EMBSEN	SLC	Reserved	DATRVS	DATINV	VSPOL	HSPOL	PCLKPOL
			rw		rw				rw	rw	rw	rw	rw	rw	rw

Bit Field	Name	Description
31:28	HRSKIP[3:0]	Repeat Line Skip Pattern Control (include even/odd line capturing) There are two configurations for this register: 0000~1101： After capturing a line of data, skip a certain number of lines, then continue capturing the next line, and repeat this process. 1110、1111: Capture only odd or even lines. The specific situation is as follows: 0000： no skipping line。 0001： each captured line followed by one skip line. 0010： each captured line followed by two skip line. 0011： each captured line followed by three skip line. …..

Bit Field	Name	Description
		1101：each captured line followed by thirteen skip lines. 1110：only capture even lines and all odd lines are skipped (first line(row0) is captured, second line(row1) is dropped, etc). 1111：only capture odd lines and all even lines are skipped (first line(row0) is dropped, second line(row1) is captured, etc). The software can only modify this register when DVP_CTRL.DVPEN=0
27:24	HISKIP[3:0]	Initial frame line skip control This register controls the number of the horizontal lines skipped at the beginning of each frame. The number of skip lines is ranged from 1 to 15. For example, if you set this register to 3 then the first 3 lines of each frames will not be captured. The software can only modify this register when DVP_CTRL.DVPEN=0.
23:16	BMAP[7:0]	Received pixel data to Frame Buffer mapping. Every 4 bytes of data received from DVP port can be treated as a data group. Within a data group (4 bytes) their byte positions can be repositioned before writing to Frame Buffer. BMAP [1:0]：where the first received data byte will be repositioned. BMAP [3:2]：where the second received data byte will be repositioned BMAP [5:4]：where the third received data byte will be repositioned. BMAP [7:6]：where the fourth received data byte will be repositioned. For example, if you want to swap the first and the second data bytes then you need to program BMAP [1:0]=01 and BMAP [3:2]=00。In normal operation, you can reposition all four bytes in any order, and the all 2-bit fields should not have the same value. If you try to remap 2 bytes to the same byte position then the lower order 2-bit field will take precedence. If a byte position is not remapped then the byte data at that position is remainded unchanged. In normal case, byte re-mapping should only be used for 8-bit mode.
15:13	Reserved	Reserved, the reset value must be maintained
12	PIXELDB	2 bytes per pixel The software should set this bit to "1" when it is operated in 8-bit mode and it requires 2 bytes to describe a pixel, otherwise set this bit to "0". The software can only modify this bit when DVP_CTRL.DVPEN=0 *Note: This bit is only used for CROP mode. Only one byte per pixel and two bytes per pixel are supported for CROP mode.*
11	Reserved	Reserved, the reset value must be maintained
10:8	DBIT[2:0]	DVP data port mode. 000:8-bit physical data port, and data is captured on all data lines. 001:10-bit physical data port, and data is captured on all data lines. 010:12-bit physical data port, and data is captured on all data lines. 011:14-bit physical data port, and data is captured on all data lines. 100:16-bit physical data port, and data is captured on all data lines. 101: 10-bit physical data port, and data is captured on 8 data lines [9:2] only. 110: 12-bit physical data port, and data is captured on 10 data lines [11:2] only.

Bit Field	Name	Description
		111: 12-bit physical data port, and data is captured on 8 data lines [11:4] only. The software can only modify this bit when DVP_CTRL.DVPEN=0
7	EMBSEN	Embedded Synchronization Mode enable bit. 1: enable embedded synchronization mode. 0: use VSYNC and HSYNC to do synchronization The software can only modify this bit when DVP_CTRL.DVPEN=0
6	SLC	Embedded Synchronization Code Position. 1: the sync code is aligned to the least significant bit of the data lines. e.g.when DBIT=1, the sync code is at [7:0] DBIT=2, the sync code is at [7:0] DBIT=6, the sync code is at [9:2]. 0: the sync code is aligned to the most significant bit of the data lines, e.g. when DBIT=1, the sync code is at [9:2] DBIT=2, the sync code is at [11:4] DBIT=6, the sync code is at [11:4]. This register can only be modified when DVP_PORTCFG.EMBSEN=0
5	Reserved	Reserved, the reset value must be maintained
4	DATRVS	To reverse data bit order. 1: enable to reverse data bit order on the DVP port data. 0: remain original data bit order.
3	DATINV	Data bit inversion. 1: enable data bit inversion on the DVP port data. 0: data bit inversion is not enabled.
2	VSPOL	VSYNC polarity control. 1: when VSYNC=0 is vertical blanking at the DVP port. 0: when VSYNC=1 is vertical blanking at the DVP port. The software can only modify this bit when DVP_CTRL.DVPEN=0.
1	HSPOL	1: when HSYNC=1 is horizontal blanking at the DVP port. 0: when HSYNC=0 is horizontal blanking at the DVP port. The software can only modify this bit when DVP_CTRL.DVPEN=0
0	PCLKPOL	DVP port pixel clock polarity control. DVP port pixel clock polarity control.。 1: the data from the sensor is expected to be latched at the falling edge of the pixel clock. 0: the data from the sensor is expected to be latched at the rising edge of the pixel clock. The software can only modify this bit when DVP_CTRL.DVPEN=0.

30.6.6 DVP FIFO Configuration Register（DVP_FIFOCFG）

Address offset：0x10

Reset value：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved													TXBURSZ[2:0]		
														rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						TXFTH[1:0]		Reserved		EN1KBD	DISP	Reserved		M2ADDR EN	M1ADDR EN
						rw				rw	rw			rw	rw

Bit Field	Name	Description
31:19	Reserved	Reserved, the reset value must be maintained
18:16	TXBURSZ[2:0]	AHB master data burst setting. When FIFO data reaches transmit threshold the master initiates memory write burst cycle according to burst setting: 000：burst size is 4 bytes 001：burst size is 8 bytes 010：burst size is 12 bytes 011：burst size is 16 bytes 100：burst size is 20 bytes 101：burst size is 24 bytes 110：burst size is 28 bytes 111：burst size is 32 bytes Note: Make sure DVP_FIFOCFG.TXBURSZ setting is same or smaller than DVP_FIFOCFG .TXFTH setting, otherwise it may encounter FIFO under run issue.
15:10	Reserved	Reserved, the reset value must be maintained
9:8	TXFTH[1:0]	FIFO transmit threshold setting. When FIFO data reaches transmit threshold the master initiates memory write burst cycle according to burst setting: 00：transmit threshold is 4 bytes 01：transmit threshold is 8 bytes 10：transmit threshold is 16 bytes

Bit Field	Name	Description
		11：transmit threshold is 32 bytes Note: Make sure DVP_FIFOCFG.TXBURSZ setting is same or smaller than DVP_FIFOCFG .TXFTH setting, otherwise it may encounter FIFO under run issue.
7:6	Reserved	Reserved, the reset value must be maintained
5	EN1KBD	Bursting across 1K boundary Control 0: increamental bursting across 1K boundary is disabled 1: increamental bursting across 1K boundary is allowed/enabled Note: If this is disabled every increamental bursting across 1K boundary will be break up and send non-sequential transaction, once it acrosses 1K boundary, it will resume increamental sequential transaction again. This control bit is for debug purpose, in normal case should leave it as default setting, '0'.
4	DISPMODE	To show current frame buffer status in term of image data byte count or buffer address control. 1: buffer address is output on DVP_FPBC1 and DVP_FPBC2 registers 0：number of data bytes which have been sent to Frame buffer is output on DVP_FPBC1 and DVP_FPBC2 registers
3:2	Reserved	Reserved, the reset value must be maintained
1	M2ADDREN	Frame Buffer 2 enable control 1: Frame Buffer 2 is enabled. 0: Frame Buffer 2 is not enabled. Note: There is a restriction to enable Frame Buffer 2: it can only be enabled at the same time as Frame Buffer 1 is enabled. Frame Buffer 2 is not allowed to be used alone. If you just need one frame buffer then you have to use Frame Buffer 1. The software can only modify this bit when DVP_CTRL.DVPEN=0
0	M1ADDREN	Frame Buffer 1 enable control 1: Frame Buffer 1 is enabled. 0: Frame Buffer 1 is not enabled. The software can only modify this bit when DVP_CTRL.DVPEN=0 and DVP_FBS.FMSIZE has to be non-zero.

30.6.7 DVP Frame 1 Start Memory Address Register（DVP_SMADDR1）

Address offset：0x14

Reset value：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MADDR[31:0]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MADDR[31:0]															
rw															

Bit Field	Name	Description
31:0	MADDR[31:0]	Frame Buffer 1 Start Address. This address is only valid when DVP_FIFOCFG.M1ADDREN=1. This address has to be 32-bit align, which means MADDR[1:0] = 0. This register can only be modified when DVP_FIFOCFG.M1ADDREN=0.

30.6.8 DVP Frame 2 Start Memory Address Register（DVP_SMADDR2）

Address offset：0x18

Reset value：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MADDR[31:0]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MADDR[31:0]															
rw															

Bit Field	Name	Description
31:0	MADDR[31:0]	Frame Buffer 2 Start Address. This address is only valid when

Bit Field	Name	Description
		DVP_FIFOCFG.M2ADDREN=1. This address has to be 32-bit align, which means MADDR[1:0] = 0。 This register can only be modified when DVP_FIFOCFG.M2ADDREN=0.

30.6.9 DVP Frame Buffer Size Register（DVP_FBS）

Address offset：0x1C

Reset value：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved										FMSIZE[21:0]					
										rw					

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FMSIZE[21:0]															
rw															

Bit Field	Name	Description
31:22	Reserved	Reserved, the reset value must be maintained
21:0	FMSIZE[21:0]	Set the frame buffer size. This register is used to protect the hardware from writing frame data beyond this range.

30.6.10 DVP Frame 1 Pixel Byte Count Register（DVP_FPBC1）

Address offset：0x20

Reset value：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PIXLC[31:0]															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PIXLC[31:0]															
r															

Bit Field	Name	Description
31:0	PIXLC[31:0]	This register is read only. When DVP_FIFOCFG.DISPMODE = 0, it stores the data byte count of frame buffer 1. When DVP_FIFOCFG.DISPMODE = 1, it stores the address of frame buffer 1.

30.6.11 DVP Frame 2 Pixel Byte Count Register（DVP_FPBC2）

Address offset：0x24

Reset value：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PIXLC[31:0]															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PIXLC[31:0]															
r															

Bit Field	Name	Description
31:0	PIXLC[31:0]	This register is read only. When DVP_FIFOCFG.DISPMODE =0, it stores the data byte count of frame buffer 2. When DVP_FIFOCFG.DISPMODE =1, it stores the address of frame buffer 2.

30.6.12 DVP Crop Start XY Register（DVP_CSXY）

Address offset：0x28

Reset value：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved					CSTAY[10:0]										
					rw										

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Reserved	CSTAX[11:0]
	rw

Bit Field	Name	Description
31:27	Reserved	Reserved, the reset value must be maintained
26:16	CSTAY[10:0]	Y-coordinate of crop image start point. It is in term of vertical lines. This register will only be used when DVP_CTRL.CROPEN=1. *Note: The first pixel coordinate is (0,0).*
15:12	Reserved	Reserved, the reset value must be maintained
11:0	CSTAX[11:0]	X-coordinate of crop image start point. It is in term of image pixels. This register will only be used when DVP_CTRL.CROPEN=1. *Note: The first pixel coordinate is (0,0).*

30.6.13 DVP Crop End XY Register（DVP_CEXY）

Address offset：0x2C

Reset value：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved					CENDY[10:0]										
					rw										

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				CENDX[11:0]											
				rw											

Bit Field	Name	Description
31:27	Reserved	Reserved, the reset value must be maintained
26:16	CENDY[10:0]	Y-coordinate of crop image end point. It is in term of vertical lines. This register will only be used when DVP_CTRL.CROPEN=1 *Note: The first pixel coordinate is (0,0).*
15:12	Reserved	Reserved, the reset value must be maintained
11:0	CENDX[11:0]	X-coordinate of crop image end point. It is in term of image pixels. This register will only be used when DVP_CTRL.CROPEN=1. *Note: The first pixel coordinate is (0,0).*

30.6.14 DVP Embedded Synchronization Code Marker Register（DVP_EMSC）

Address offset： 0x30

Reset value： 0x9D80 B6AB

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
HE_PTTN [7:0]								HS_PTTN [7:0]							
rw								rw							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VE_PTTN [7:0]								VS_PTTN [7:0]							
rw								rw							

Bit Field	Name	Description
31:24	HE_PTTN [7:0]	Line valid ending synchronization code. This bit can configure the fourth byte data of the synchronization code in the embedded synchronization mode, serving as a feature value to send the line valid ending synchronization code to the DVP system. This register can only be modified when DVP_PORTCFG.EMBSEN =0
23:16	HS_PTTN [7:0]	Line valid starting synchronization code This bit can configure the fourth byte data of the synchronization code in the embedded synchronization mode, serving as a feature value to send the line valid starting synchronization code to the DVP system. This register can only be modified when DVP_PORTCFG.EMBSEN=0.
15:8	VE_PTTN [7:0]	Frame ending synchronization code This bit can configure the fourth byte data of the synchronization code in the embedded synchronization mode, serving as a feature value to send the frame ending synchronization code to the DVP system. This register can only be modified when DVP_PORTCFG.EMBSEN=0.
7:0	VS_PTTN [7:0]	Frame starting synchronization code This bit can configure the fourth byte data of the synchronization code in the embedded synchronization mode, serving as a feature value to send the frame starting synchronization code to the DVP system. This register can only be modified when DVP_PORTCFG.EMBSEN=0.

30.6.15 DVP Embedded Synchronization Code Mask Register（DVP_EMSCM）

Address offset:0x34

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
HE_MASK[7:0]								HS_MASK[7:0]							
rw								rw							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VE_MASK[7:0]								VS_MASK[7:0]							
rw								rw							

Bit Field	Name	Description
31:24	HE_MASK[7:0]	This is mask register for HE_PTTN register. If the corresponding HE_PTTN mask bits are set to "1" then these bits in HE_PTTN will be excluded from this pattern detection This register can only be modified when DVP_PORTCFG.EMBSEN=0.
23:16	HS_MASK[7:0]	This is mask register for HS_PTTN register. If the corresponding HS_PTTN mask bits are set to "1" then these bits in HS_PTTN will be excluded from this pattern detection. This register can only be modified when DVP_PORTCFG.EMBSEN=0.
15:8	VE_MASK[7:0]	This is mask register for VE_PTTN register. If the corresponding VE_PTTN mask bits are set to "1" then these bits in VE_PTTN will be excluded from this pattern detection. This register can only be modified when DVP_PORTCFG.EMBSEN=0.
7:0	VS_MASK[7:0]	This is mask register for VS_PTTN register. If the corresponding VS_PTTN mask bits are set to "1" then these bits in VS_PTTN will be excluded from this pattern detection. This register can only be modified when DVP_PORTCFG.EMBSEN=0.

31 Flexible External Memory Controller（FEMC）

31.1 FEMC Functional Description

The Flexible External Memory Controller (FEMC) is used to access various external memories, allowing for the convenient expansion of different types of high-capacity static memories based on application requirements. It can simultaneously expand multiple types of static memories without adding external interfaces. All external memories share the addresses, data, and control signals output by the FEMC controller, and FEMC distinguishes between different external devices using a unique chip-select signal. At any given time, FEMC can only access one external device. Its main function is:

- Converting the access signals sent by the core to the AHB bus into signals that comply with the communication rules of external storage devices, and delivering them to the corresponding pins of the external storage device.
- Adjusting signal width and timing to meet the timing requirements for accessing external devices.

Main features:

- Supports the expansion of the following devices:
  - SRAM
  - PSRAM
  - ROM
  - NOR Flash
  - NAND Flash（SLC）
  - LCD（8080/6800）
- Supports two NAND flash blocks, hardware 1-bit ECC can check up to 8K bytes of data
- Supports burst access mode for synchronous devices, such as NOR flash and PSRAM
- 8-bit or 16-bit data bus
- Each memory block has independent chip select control
- Supports various different devices through timing programming
- Depending on the data width of the external memory, automatically converts 32-bit AHB access requests into continuous 16-bit or 8-bit access requests to communicate with external 16-bit or 8-bit memory devices.
- PSRAM and SRAM devices support write enable and byte-select outputs.

Typically, during system reset or power-up, all FEMC registers that configure the type and characteristics of external memory should be set to ensure proper operation after power-up or reset. These configurations can also be changed at any time.

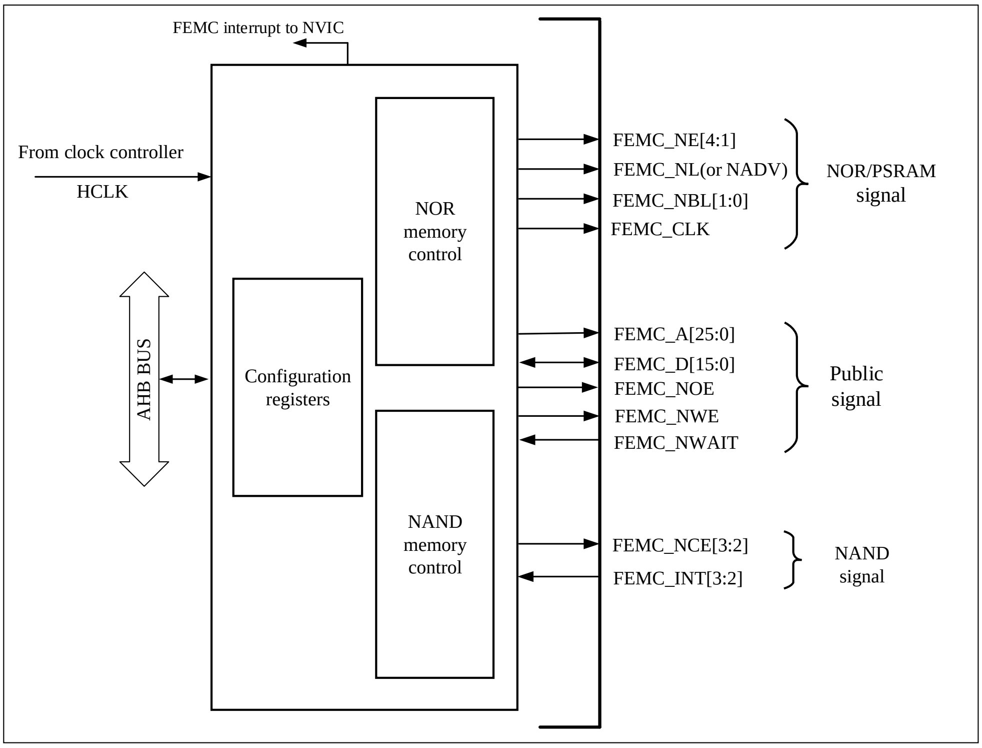
31.2 FEMC Block Diagram

FEMC consists of five parts:

- AHB interface
- FEMC configuration register
- NOR Flash and PSRAM controller
- NAND Flash controller
- External device interface

The FEMC block diagram is as follows:

Figure 31-1 FEMC Block Diagram



31.3 AHB Interface

The reference clock for FEMC is the AHB clock (HCLK). The AHB interface provides a channel for the internal CPU and other bus control devices to access external static memory. FEMC serves as a protocol conversion interface from the AHB bus to external devices, where 32-bit AHB read/write operations can be split into continuous 16-bit or 8-bit operations based on the selected data channel width of the external memory.

General operation rules:

During data transfer, the data width of the requested AHB operation can be 8-bit, 16-bit, or 32-bit, while the external device has a fixed data width. Therefore, the AHB data width and memory data width may not be the same. To ensure consistency in data transfer, FEMC read and write accesses need to adhere to the following rules:

- When the memory data width is equal to the data width of the AHB access, there are no data transfer consistency issues.
- When the memory data width is greater than the data width of the AHB access, FEMC will split the AHB operation into several continuous data width transfers to accommodate the data width of the external device.

- When the memory data width is less than the data width of the AHB access: If the external storage device has byte selection functionality, such as SRAM, ROM, PSRAM, it can be accessed by its byte channel BL[1:0]. Otherwise, write operations are disabled, and only read operations are allowed.

FEMC is configured by a set of registers. The FEMC register section provides detailed descriptions of the NOR flash, PSRAM controller registers, and NAND flash registers.

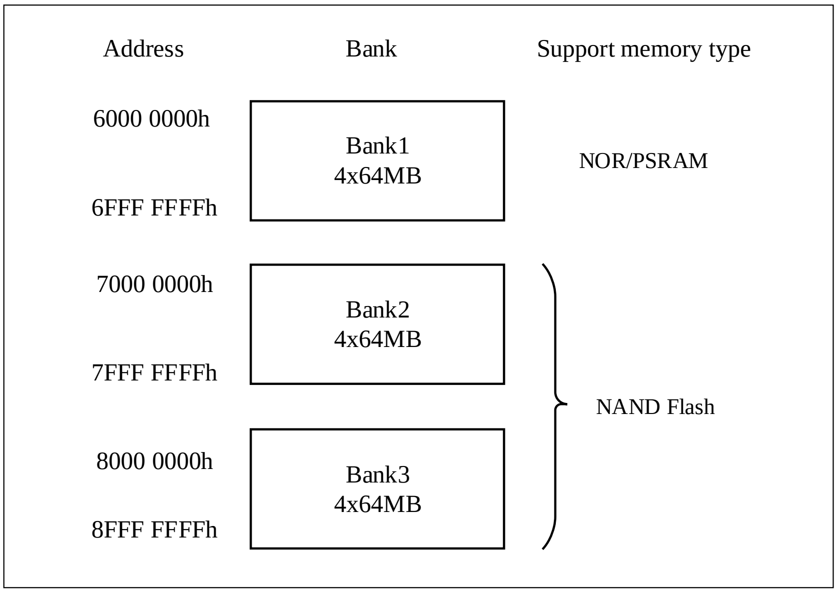
31.4 External Device Address Mapping

FEMC divides the external memory into three fixed-size banks, with Bank1 further divided into 4 regions, each occupying 64M bytes, totaling 256M bytes. Bank2 and Bank3 are each 256M in size and are divided into 2 sections, namely attribute storage space and common storage space, as shown in Figure 18-2.

- Bank1 is used to access NOR flash or SRAM/PSRAM storage devices. This bank is split into 4 NOR/PSRAM subbanks and has 4 dedicated chip selects, allowing Bank1 to access up to four external storage devices.
- Banks 2 and 3 are used to address NAND Flash devices, with each bank able to connect to one NAND flash.

The memory type on each bank or region can be independently configured, defined by the user in the corresponding configuration registers.

Figure 31-2 FEMC Memory Blocks



31.4.1 NOR/PSRAM Address Mapping

AHB address lines HADDR[27:26] are used for chip select signals for the four regions:

Table 31-1 NOR/PSRAM Bank Selection

HADDR[27:26][(1)]	Selected Bank
00	Bank1 NOR/PSRAM/SRAM 1
01	Bank 1 NOR/PSRAM/SRAM 2
10	Bank 1 NOR/PSRAM/SRAM 3

11	Bank 1 NOR/PSRAM/SRAM 4

Note: (1) HADDR is the internal AHB address line.

AHB address lines HADDR[25:0] contain the external memory address. HADDR is a byte address, and external memory access may not always be byte-aligned, leading to potential address inconsistencies. FEMC can adjust HADDR to accommodate the data width of the external memory. The specific rules are as follows:

Table 31-2 External memory address

Data Width	Address Issued to The Memory	Maximum Memory Capacity (Bits)
8-bit	HADDR[25:0] is connected to FEMC_A[25:0]	64Mbyte x 8 = 512Mbit
16-bit	HADDR[25:1] is connected to FEMC_A[24:0], HADDR[0] is not connected.	64Mbyte/2 x 16 = 512Mbit

1) If the external memory has an 8-bit data width and is byte-aligned, HADDR[25:0] is connected to FEMC_A[25:0], and FEMC_A[25:0] is connected to the external memory address lines.

2) If the external memory has a 16-bit data width and is half-word aligned, the byte address of HADDR needs to be converted to a half-word address before connecting to the external memory. Therefore, HADDR[25:1] is connected to FEMC_A[24:0]. FEMC_A[24:0] is connected to the external memory address lines.

3) Regardless of the width of the external memory (16-bit or 8-bit), FEMC_A[0] should always be connected to the external memory address line A[0].

Support for non-aligned access in NOR flash/PSRAM

By setting the corresponding registers, each NOR flash or PSRAM memory block can be configured to support non-aligned data access. Depending on whether the access is asynchronous or synchronous, the external memory must meet the following standards:

- Asynchronous Mode: In asynchronous mode, as long as each access has a correct address, non-aligned data access is fully supported.
- Synchronous Mode: In synchronous mode, FEMC only issues one address signal, and then the grouped data is transmitted sequentially through the clock.

Some NOR memories support linear non-aligned burst access, where a fixed number of data words can be read from consecutive addresses with a modulus N (typical N values are 8 or 16). In this case, the non-aligned access mode for the memory can be set to the same mode as AHB.

Note: If the non-aligned access mode of the external memory cannot be set to the same mode as AHB, the corresponding bit of the FEMC register should be configured to disable non-aligned access, and non-aligned access requests should be split into two consecutive access operations.

31.4.2 NAND Address Mapping

Bank2 and Bank3 are used to access NAND Flash, and each bank is further split into attribute access space and Common access space:

Table 31-3 Memory Mapping And Timing Registers

Start Address	End Address	FEMC Bank	Memory Space	Timing Register
0x8800 0000	0x8BFF FFFF	Bank3-NAND Flash	attribute	FEMC_NATTMEMTM3(0x8C)
0x8000 0000	0x83FF FFFF		common	FEMC_ NCMEMTM3(0x88)
0x7800 0000	0x7BFF FFFF	Bank2-NAND Flash	attribute	FEMC_NATTMEMTM2(0x6C)
0x7000 0000	0x73FF FFFF		common	FEMC_NCMEMTM2(0x68)

For NAND Flash memory, the common and attribute memory spaces(lower 256K bytes) are subdivided into three sections, as shown in Table 31-4:

Table 31-4 NAND Bank Selection

Section Name	HADDR[17:16]	Address Range	Capacity
Address section	1X	0x020000~0x03FFFF	128KB
Command section	01	0x010000~0x01FFFF	64KB
Data section	00	0x000000~0x00FFFF	64KB

AHB uses HADDR[17:16] to select one of the three regions above:

- HADDR[17:16] = 00 (The first 64K byte region of common/attribute space), selecting the data section
- HADDR[17:16] = 01 (The second 64K byte region of common/attribute space), selecting the command section
- HADDR[17:16] = 1x (The second 128K byte region of common/attribute space), selecting the address section

The application software uses the 3 sections to access the NAND Flash memory, The operation rules are as follows:

- To send commands to NAND Flash: software only needs to write the command to be sent to any memory location in the command section. During the command transfer, FEMC will enable the command latch enable signal (CLE), that is, FEMC_A[16] will be pulled high.
- To specify the address for NAND Flash operations: software only needs to write the address to any memory location in the address section. During the address transfer, FEMC will enable the address latch enable signal (ALE), that is, FEMC_A[17] will be high. Since a NAND address can have 4 or 5 bytes (depending on the actual memory capacity), continuous writes to the address section are needed to output the complete operation address.
- Reading/writing data: the software reads or writes the data value from or to any memory location in the data section.

When FEMC is in data transmission mode, software needs to write data to the data section. When FEMC is in data reception mode, software needs to read data from the data section. Since NAND Flash automatically increments its internal operation address, software does not need to modify the operation address during read/write operations.

31.5 NOR Flash And PSRAM Controller

The FEMC generates the appropriate signal timings to drive the following types of memories

- Asynchronous SRAM and ROM
  - 8-bit
  - 16-bit

- PSRAM（Cellular RAM）
  - Asynchronous mode
  - Burst mode for synchronous accesses
- NOR Flash
  - Asynchronous mode or Synchronous Burst Mode
  - Multiplexed or nonmultiplexed

FEMC generates a unique chip select signal NE[4:1] for each memory block in Bank1, used for chip selection among the 4 memory blocks, while all other signals (address, data, and control) are shared. In synchronous mode, FEMC provides a clock (CLK) to the selected external device, with a frequency that is a division of the HCLK. The size of each memory block is fixed at 64M bytes.

Each memory block has dedicated control registers (see Section 31.7.2 (NOR Flash and PSRAM Control Registers)).

In asynchronous mode, all controller output signals change on the rising edge of the internal AHB bus clock (HCLK).

In synchronous mode, all controller output data changes on the falling edge of the external memory clock(FEMC_CLK).

Programmable memory parameters include access timing (see table below), support for non-aligned data access, and wait period management (only for accessing PSRAM and NOR flash in burst mode).

Table 31-5 Programmable NOR/PSRAM Access Parameters

Parameter	Function	Access mode	Unit	Min.	Max
Address setup	Duration of the address setup phase	Asynchronous	AHB clock cycle(HCLK)	1	16
Address hold	Duration of the address hold phase	Asynchronous, multiplexed I/O	AHB clock cycle(HCLK)	1	16
Data setup	Duration of the data setup phase	Asynchronous	AHB clock cycle(HCLK)	2	256
Bus turn	Duration of the bus turnaround phase	Asynchronous or synchronous read	AHB clock cycle(HCLK)	1	16
Clock divide ratio	Ratio of memory access clock cycles (CLK) to AHB clock cycles	synchronous	AHB clock cycle(HCLK)	2	16
Data latency	Number of clock cycles to issue to the memory before the first data of the burst	synchronous	Memory clock cycle(HCLK)	2	17

31.5.1 External Memory Interface Signals

Tables 31-6, 31-7, and 31-8 list the typical signals associated with the NOR flash and PSRAM interfaces.

Note: Signals with the prefix "N" indicate active low signals.

31.5.1.1 NOR Flash

NOR flash memory is addressed in 16-bit words, with a maximum capacity of 64M bytes (26 address lines). The "N" in the signal indicates active low.

31.5.1.1.1 Non- Multiplexed Interface

Table 31-6 Nonmultiplexed I/O NOR flash

Signal Name	Input/Output	Function
CLK	output	Clock (used in synchronous burst mode)
A[25:0]	output	address bus
D[15:0]	input/ output	bidirectional data bus
NE[x]	output	chip select，x = 1,2,3,4
NOE	output	output enable
NWE	output	write enable
NL(=NADV)	output	Latch enable (some NOR Flash devices refer to this signal as Address Valid signal NADV）
NWAIT	input	NOR flash wait input signal to the FEMC

31.5.1.1.2 Multiplexed Interface

Table 31-7 Multiplexed I/O NOR flash

Signal Name	Input/Output	Function
CLK	output	Clock (used in synchronous burst mode)
A[25:16]	output	address bus
AD[15:0]	input/ output	16-bit multiplexed, bidirectional address/data bus
NE[x]	output	chip select，x=1,2,3,4
NOE	output	output enable
NWE	output	write enable
NL(=NADV)	output	Latch enable (some NOR Flash devices refer to this signal as Address Valid signal NADV）
NWAIT	input	NOR flash wait input signal to the FEMC

31.5.1.2 PSRAM

PSRAM memories are addressed in 16-bit words, with a maximum capacity of 64M bytes (26 address lines), using a non-multiplexed mode.

Table 31-8 Nonmultiplexed I/O PSRAM

Signal Name	Input/Output	Function
CLK	output	clock (used in synchronous burst mode)
A[25:0]	output	address bus
D[15:0]	input / output	bidirectional data bus
NE[x]	output	chip select，x = 1,2,3,4（called NCE by PSRAM (Cellular RAM i.e. CRAM)）)
NOE	output	output enable
NWE	output	write enable
NL(=NADV)	output	address valid (memory signal name: NADV)
NWAIT	input	PSRAM wait input signal to the FEMC
NBL[1]	output	upper byte enable (memory signal name: NUB)

NBL[0]	output	lowed byte enable (memory signal name: NLB)

31.5.2 Supported Memories And Operations

The table below lists the supported memories, access modes, and operation methods, with FEMC not supporting the shaded operation modes.

Table 31-9 Supported Memories And Transactions to NOR Flash/PSRAM

Device	Mode	Read/Write	AHB Data Size	Memory Data Size	Supported/Not Supported	Comments
NOR Flash（bus-multiplexed and non-bus-multiplexed）	Asynchronous	Read	8	16	Y	
	Asynchronous	Write	8	16	N	
	Asynchronous	Read	16	16	Y	
	Asynchronous	Write	16	16	Y	
	Asynchronous	Read	32	16	Y	Split into two FSMC accesses
	Asynchronous	Write	32	16	Y	Split into two FSMC accesses
	Asynchronous page	Read	-	16	N	Mode is not supported
	Synchronous	Read	8	16	N	
	Synchronous	Read	16	16	Y	
	Synchronous	Read	32	16	Y	
PSRAM（bus-multiplexed and non-bus-multiplexed）	Asynchronous	Read	8	16	Y	
	Asynchronous	Write	8	16	Y	Using byte signals NBL[1:0]
	Asynchronous	Read	16	16	Y	
	Asynchronous	Write	16	16	Y	
	Asynchronous	Read	32	16	Y	Split into two FSMC accesses
	Asynchronous	Write	32	16	Y	Split into two FSMC accesses
	Asynchronous page	Read	-	16	N	Mode is not supported
	Synchronous	Read	8	16	N	
	Synchronous	Read	16	16	Y	
	Synchronous	Read	32	16	Y	
	Synchronous	Write	8	16	Y	Using byte signals NBL[1:0]
	Synchronous	Write	16/32	16	Y	

SRAM and ROM	Asynchronous	Read	8/16	16	Y	Using byte signals NBL[1:0]
	Asynchronous	Write	8/16	16	Y	Using byte signals NBL[1:0]
	Asynchronous	Read	32	16	Y	Split into two FSMC accesses
	Asynchronous	Write	32	16	Y	Split into two FSMC accesses

31.5.3 General Timing Rules

31.5.3.1 Signals synchronization

- All controller output signals change on the rising edge of the internal clock (HCLK).
- In synchronous write mode (PSRAM), all output signals change on the rising edge of HCLK.

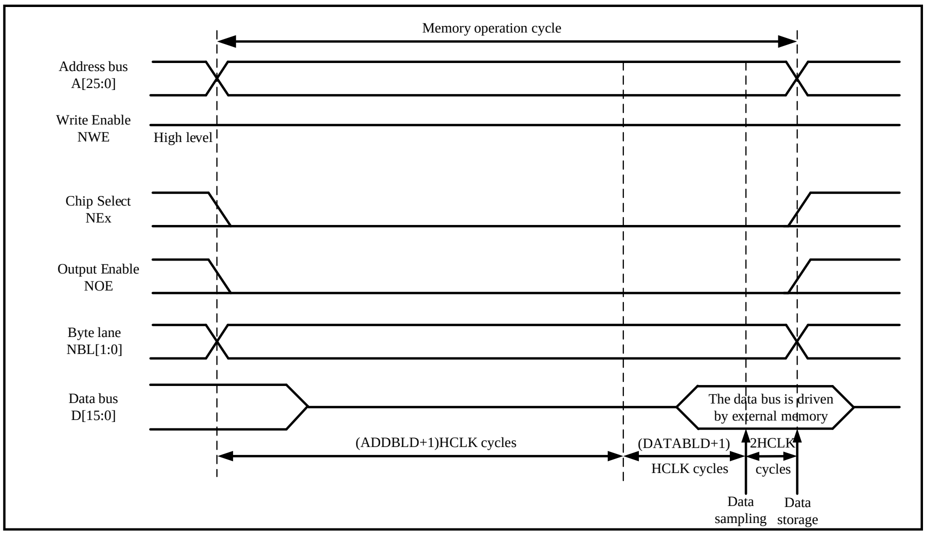
31.5.4 NOR Flash And PSRAM Controller Transactions

31.5.4.1 Asynchronous Static Memory (NOR Flash and PSRAM)

- All signals are synchronized by the internal clock(HCLK), but this clock is not issued to the memory.
- To guarantee that the memory data-hold timing constraint is met (the interval from chip select de-assertion to data transition, usually 0 ns min), FEMC always samples the data before de-asserting the chip select (NE) signals.
- Asynchronous memory operations: modes 1 and 2 are non-extended modes, A/B/C/D are extended modes, with separate timing control for read and write in extended mode, and the timing configuration for read and write is the same in non-extended mode.
- When extended mode is enabled, modes A, B, C, and D can be mixed for read and write operations (for example, allowing reads in mode A and writes in mode B).
- When extended mode is disabled (FEMC_SNCTRLx.EXTEN=0), the FEMC can operate in Mode1 or Mode2 as follows:
  - Mode1 is the default mode when accessing SRAM/PSRAM.
  - Mode2 is the default mode when accessing NOR Flash.

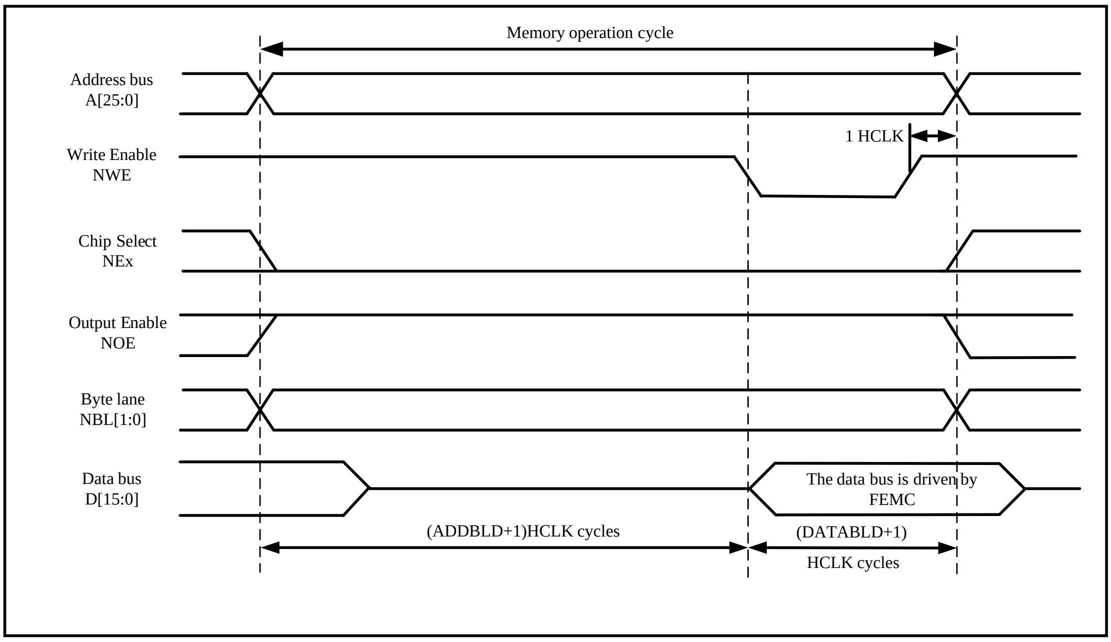
31.5.4.2 Mode1 - SRAM/CRAM

Figure 31-3 Mode1 Read Accesses



Note: ADDBLD: address setup time.   DATABLD: data setup time

Figure 31-4 Mode1 Write Accesses



Note: ADDBLD: address setup time.   DATABLD: data setup time

The one HCLK cycle at the end of the write transaction helps guarantee the address and data hold time after the NWE rising edge. Due to the presence of this one HCLK cycle, the value of DATABLD must be greater than 0 (DATABLD > 0).

31.5.4.2.1 Register Configuration in Mode1

Table 31-8 FEMC_SNCTRLx Register Configuration (Mode1)

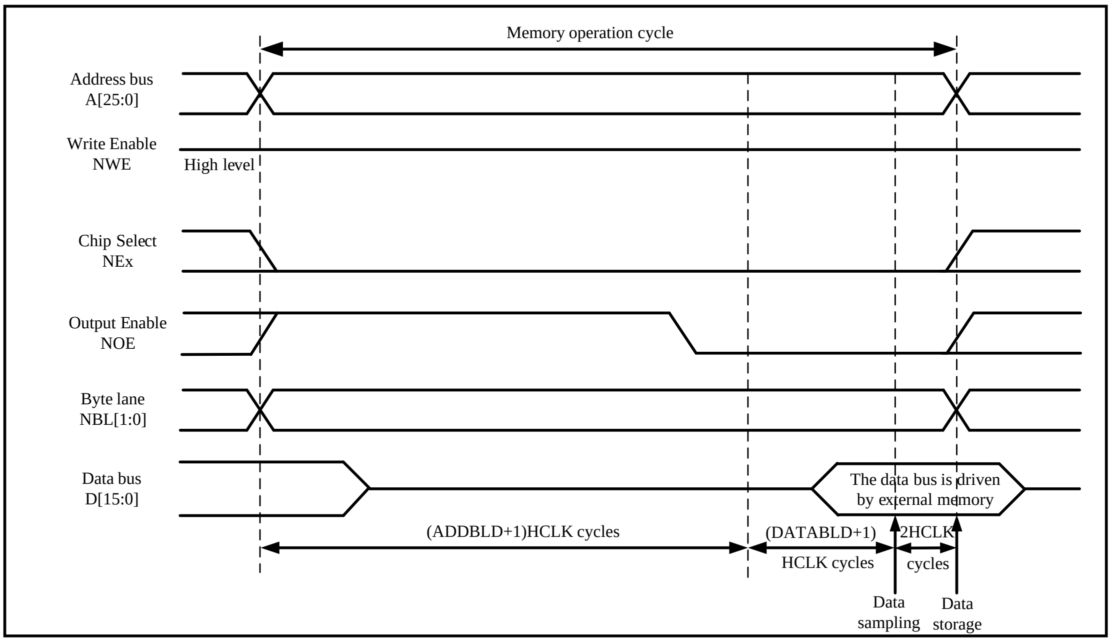
Bit Field	Name	Value to Set
31:20	Reserved	0x000
19	NADVDIR	0x0
18:16	PGSIZE[2:0]	0x0(invalid in asynchronous mode)
15	BURSTWREN	0x0(invalid in asynchronous mode)
14	WCFG	0x0(invalid in asynchronous mode)
13	WAITDIR	Only valid when bit 8 is set to '1'
12	WAITEN	0x0(invalid in asynchronous mode)
11	WRAPEN	0x0
10	BURSTEN	0x0
9	MUXEN	0x0
8	WAITASYNC	Set to 1 if the memory supports this feature, otherwise set to 0
7	EXTEN	0x0
6	ACCEN	0x0
5	WREN	As needed
4	MBEN	0x1
3:2	MDBW[1:0]	As needed
1:0	MTYPE[1:0]	As needed，exclude 2’b10 (NOR Flash)

Table 31-9 FEMC_SNTCFGx Register Configuration (Mode1)

Bit Field	Name	Value to Set
31:30	Reserved	0x0
29:28	ACCMODE[1:0]	Don’t care
27:24	CLKDIV[3:0]	Don’t care
23:20	BUSRECOVERY[3:0]	Time between NEx high to NEx low
19:16	DATAHLD[3:0]	Don’t care
15:8	DATABLD[7:0]	Duration of the second access phase, for write operations it is (DATABLD + 1) HCLK cycle, for read operations it is (DATABLD + 3) HCLK cycles. This field cannot be 0, it must be at least 1.
7:4	ADDHLD[3:0]	Don’t care
3:0	ADDBLD[3:0]	Duration of the first access phase, it is (ADDBLD+1) HCLK cycle.

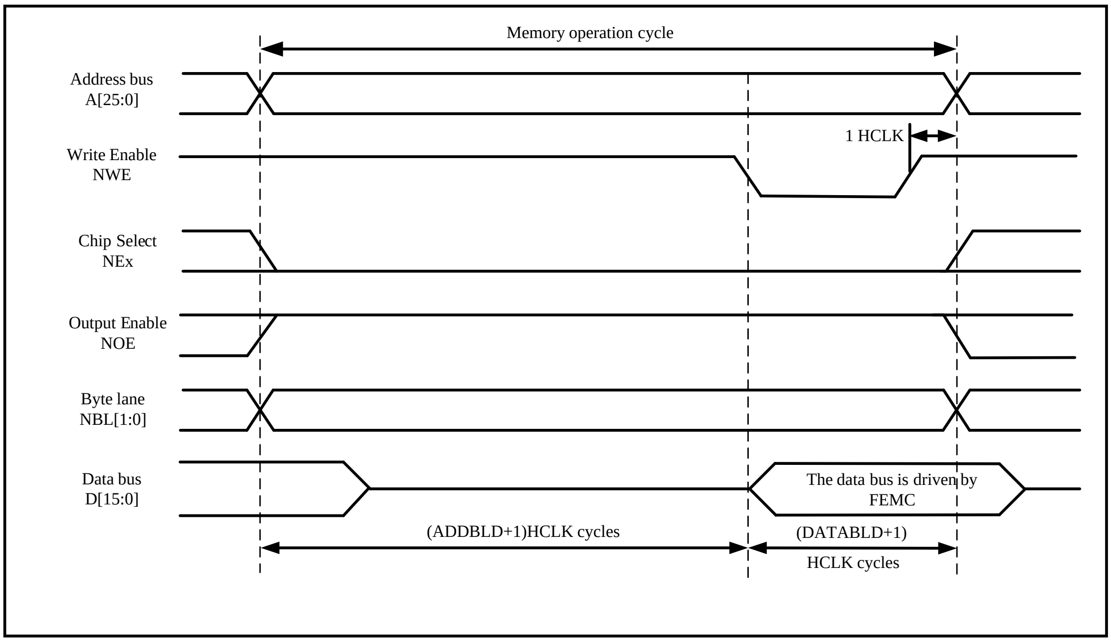
31.5.4.3 Mode A - SRAM/PSRAM(CRAM) OE Toggling

Figure 31-4 Mode A Read Accesses



Note: ADDBLD: address setup time.   DATABLD: data setup time

Figure 31-6 Mode A Write Accesses



Note: ADDBLD: address setup time.   DATABLD: data setup time

31.5.4.3.1 Register Configuration in Mode A

The difference between Mode A and Mode 1 is the write timing. When the registers of the two modes have the same timing configuration, Mode A has independent write and read timings.

Table 31-10 FEMC_SNCTRLx Register Configuration (Mode A)

Bit Field	Name	Value to Set
31:20	Reserved	0x000
19	NADVDIR	0x0
18:16	PGSIZE[2:0]	0x0(invalid in asynchronous mode)
15	BURSTWREN	0x0(invalid in asynchronous mode)
14	WCFG	0x0(invalid in asynchronous mode)
13	WAITDIR	Only valid when bit 8 is set to '1'
12	WAITEN	0x0(invalid in asynchronous mode)
11	WRAPEN	0x0
10	BURSTEN	0x0
9	MUXEN	0x0
8	WAITASYNC	Set to 1 if the memory supports this feature, otherwise set to 0
7	EXTEN	0x1
6	ACCEN	Don't care
5	WREN	As needed
4	MBEN	0x1
3:2	MDBW[1:0]	As needed
1:0	MTYPE[1:0]	As needed，exclude 2'b10(NOR flash)

Table 31-13 FEMC_SNTCFGx Register Configuration (Mode A)

Bit Field	Name	Value to Set
31:30	Reserved	0x0
29:28	ACCMODE[1:0]	Don't care
27:24	CLKDIV[3:0]	Don't care
23:20	BUSRECOVERY[3:0]	Time between NEx high to NEx low
19:16	DATAHLD[3:0]	Don't care
15:8	DATABLD[7:0]	Duration of the second phase of read access, it is (DATABLD + 3) HCLK cycle.
7:4	ADDHLD[3:0]	Don't care
3:0	ADDBLD[3:0]	Duration of the first phase of read access, it is (ADDBLD+1) HCLK cycle

Table 31-14 FEMC_SNWTCFGx Register Configuration (Mode A)

Bit Field	Name	Value to Set
31:30	Reserved	0x0
29:28	ACCMODE[1:0]	0x0
27:20	Reserved	Don't care
19:16	BUSRECOVERY[3:0]	Time between NEx high to NEx low
15:8	DATABLD[7:0]	Duration of the second phase of write access, it is (DATABLD+1) HCLK cycle.This field cannot be 0, it must be at least 1.
7:4	ADDHLD[3:0]	Don't care
3:0	ADDBLD[3:0]	Duration of the first phase of write access, it is (ADDBLD+1) HCLK cycle.

31.5.4.4 Mode 2/B – NOR Flash

Figure 31-5 Mode 2/B Read Accesses

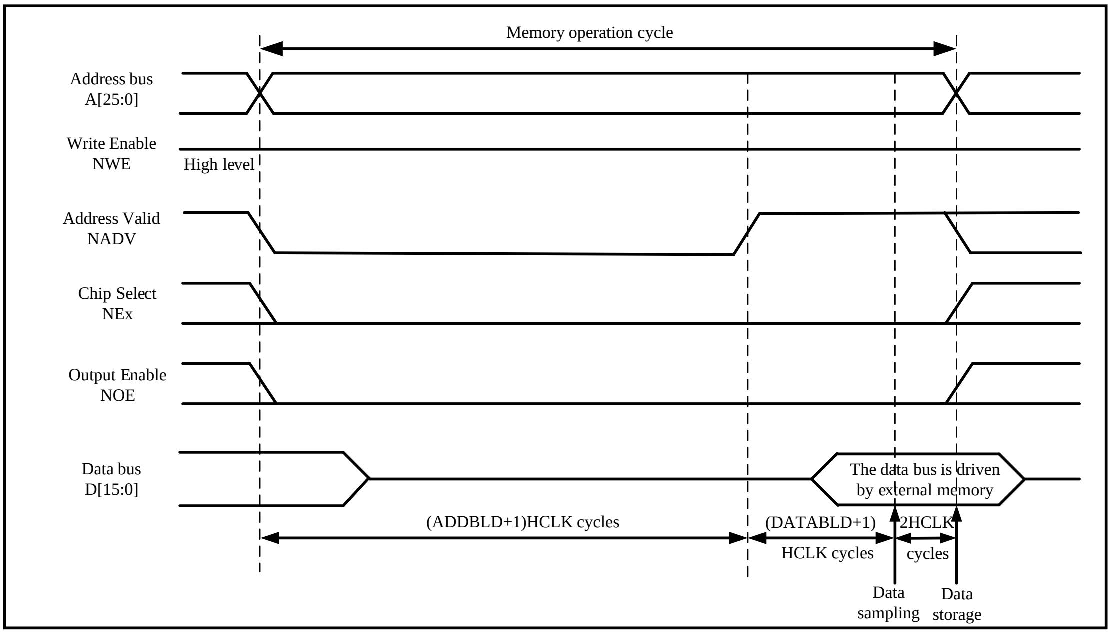


Figure 31-8 Mode 2 Write Accesses

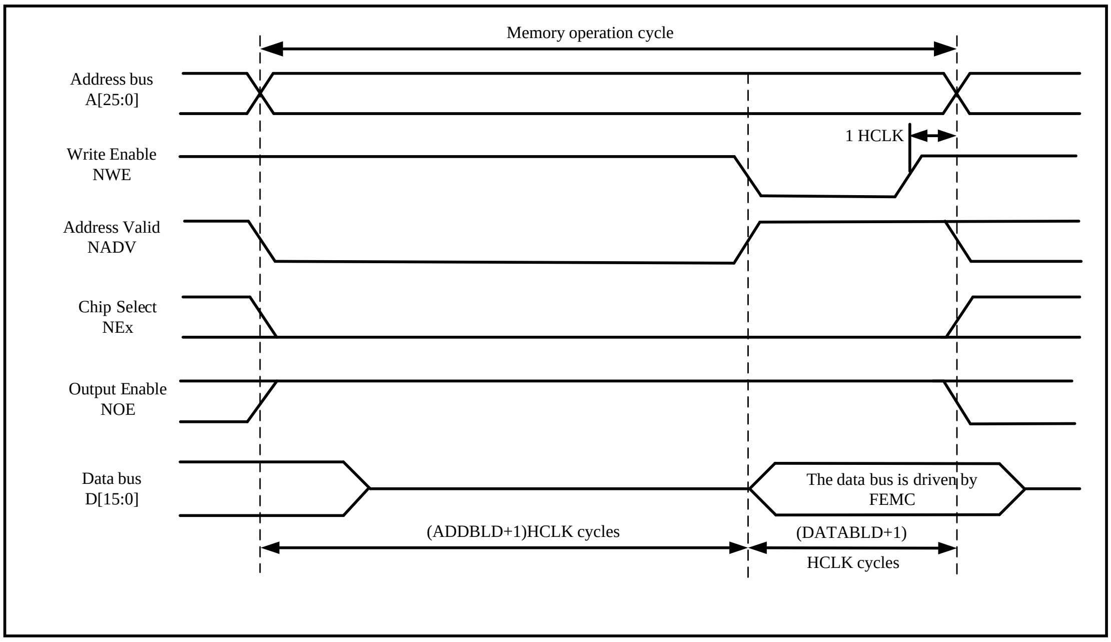
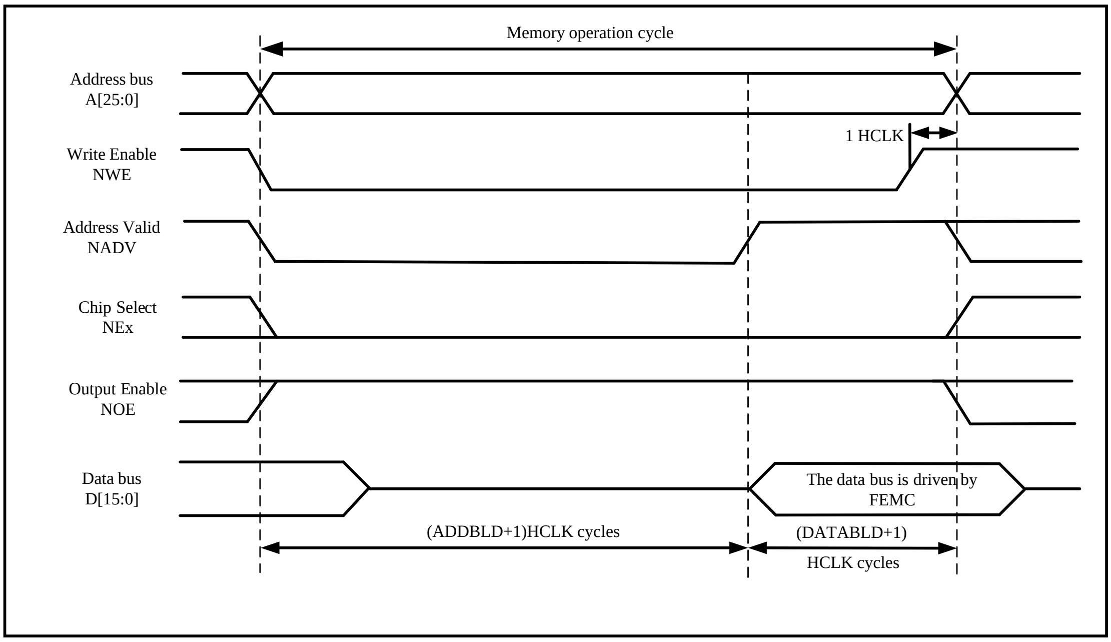


Figure 31-9 Mode B Write Accesses



Note: ADDBLD: address setup time.   DATABLD: data setup time

31.5.4.4.1 Register Configuration in Mode 2/B

In comparison to Mode 1, Mode 2/B differs in the change of NADV, and in the extended mode (Mode B), the read and write timings are independent of each other.

Table 31-15 FEMC_SNCTRLx Register Configuration(Mode 2/B)

Bit Field	Name	Value to Set
31:20	Reserved	0x000
19	NADVDIR	0x0
18:16	PGSIZE[2:0]	0x0(invalid in asynchronous mode)
15	BURSTWREN	0x0(invalid in asynchronous mode)
14	WCFG	0x0(invalid in asynchronous mode)
13	WAITDIR	Only valid when bit 8 is set to '1'
12	WAITEN	0x0(invalid in asynchronous mode)
11	WRAPEN	0x0
10	BURSTEN	0x0
9	MUXEN	0x0
8	WAITASYNC	Set to 1 if the memory supports this feature, otherwise set to 0
7	EXTEN	Mode2：0x0 ModeB：0x1
6	ACCEN	0x1
5	WREN	As needed
4	MBEN	0x1
3:2	MDBW[1:0]	As needed

1:0	MTYPE[1:0]	0x2（NOR Flash）

Table 31-11 FEMC_SNTCFGx Register Configuration(Mode 2/B)

Bit Field	Name	Value to Set
31:30	Reserved	0x0
29:28	ACCMODE[1:0]	If FEMC_SNCTRLx.EXTEN=1, this bit is set to 0x01.
27:24	CLKDIV[3:0]	Don't care
23:20	BUSRECOVERY[3:0]	Time between NEx high to NEx low
19:16	DATAHLD[3:0]	Don't care
15:8	DATABLD[7:0]	Duration of the second phase of read access, it is (DATABLD + 3) HCLK cycle. This field cannot be 0, it must be at least 1.
7:4	ADDHLD[3:0]	Don't care
3:0	ADDBLD[3:0]	Duration of the first phase of read access, it is (ADDBLD+1) HCLK cycle.

Table 31-17 FEMC_SNWTCFGx Register Configuration(Mode B)

Bit Field	Name	Value to Set
31:30	Reserved	0x0
29:28	ACCMODE[1:0]	If FEMC_SNCTRLx.EXTEN=1, this bit is set to 0x01.
27:20	Reserved	Don't care
19:16	BUSRECOVERY[3:0]	Time between NEx high to NEx low.
15:8	DATABLD[7:0]	Duration of the second phase of write access, it is (DATABLD+1) HCLK cycle. This field cannot be 0, it must be at least 1.
7:4	ADDHLD[3:0]	Don't care
3:0	ADDBLD[3:0]	Duration of the first phase of read access, it is (ADDBLD+1) HCLK cycle.

Note: FEMC_SNWTCFGx is only valid when the extended mode (Mode B) is set. Otherwise, the contents of this register are not valid.

31.5.4.5 Mode C – NOR Flash – OE Toggling

Figure 31-6 Mode C Read Accesses

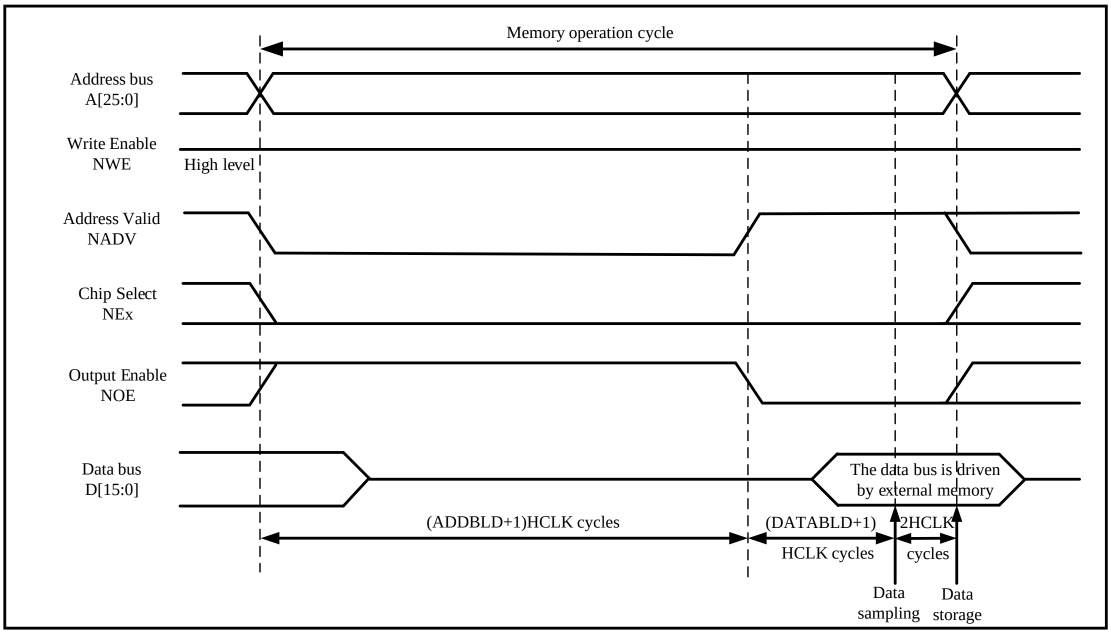
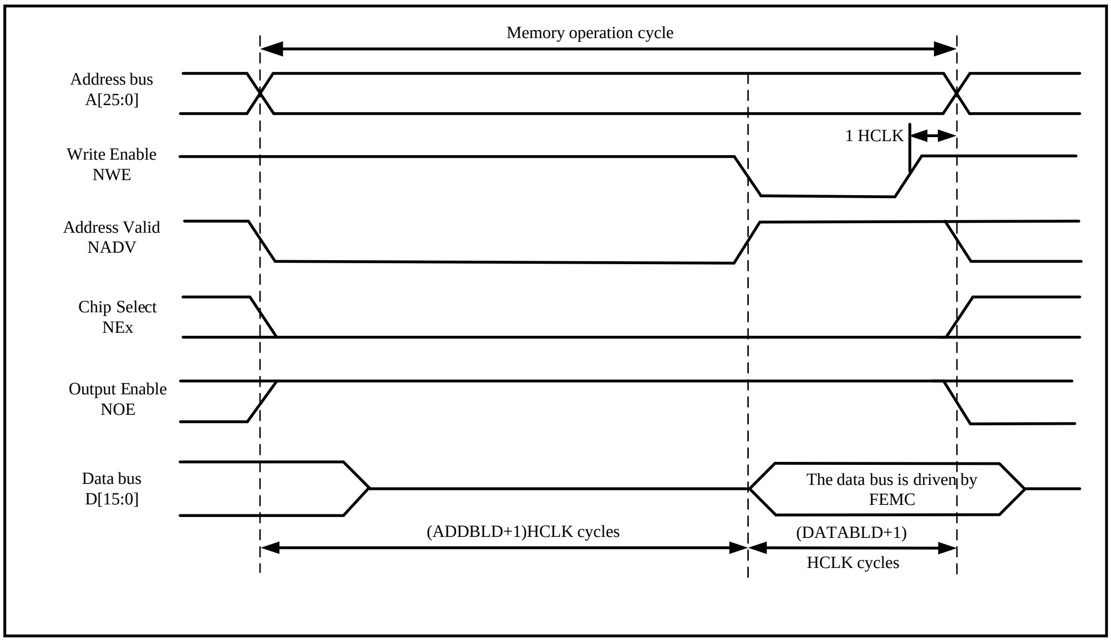


Figure 31-11 Mode C Write Accesses



Note: ADDBLD: address setup time.   DATABLD: data setup time

31.5.4.5.1 Register Configuration in Mode C

The difference between Mode C and Mode 1 is the write timing. When the registers of the two modes have the same timing configuration, Mode C has independent write and read timings.

Table 31-18 FEMC_SNCTRLx Register Configuration(Mode C)

Bit Field	Name	Value to Set
31:20	Reserved	0x000
19	NADVDIR	0x0
18:16	PGSIZE[2:0]	0x0(invalid in asynchronous mode)
15	BURSTWREN	0x0(invalid in asynchronous mode)
14	WCFG	0x0(invalid in asynchronous mode)
13	WAITDIR	Only valid when bit 8 is set to '1'
12	WAITEN	0x0(invalid in asynchronous mode)
11	WRAPEN	0x0
10	BURSTEN	0x0
9	MUXEN	0x0
8	WAITASYNC	Set to 1 if the memory supports this feature, otherwise set to 0
7	EXTEN	0x1
6	ACCEN	0x1
5	WREN	As needed
4	MBEN	0x1
3:2	MDBW[1:0]	As needed
1:0	MTYPE[1:0]	0x2（NOR Flash）

Table 31-19 FEMC_SNTCFGx Register Configuration(Mode C)

Bit Field	Name	Value to Set
31:30	Reserved	0x0
29:28	ACCMODE[1:0]	0x2
27:24	CLKDIV[3:0]	0x0
23:20	BUSRECOVERY[3:0]	Time between NEx high to NEx low.
19:16	DATAHLD[3:0]	Don’t care
15:8	DATABLD[7:0]	Duration of the second phase of read access, it is (DATABLD+3) HCLK cycle. This field cannot be 0, it must be at least 1.
7:4	ADDHLD[3:0]	0x0
3:0	ADDBLD[3:0]	Duration of the first phase of read access, it is (ADDBLD+1) HCLK cycle.

Table 31-20 FEMC_SNWTCFGx Register Configuration(Mode C)

Bit Field	Name	Value to Set
31:30	Reserved	0x0
29:28	ACCMODE[1:0]	0x2
27:20	Reserved	0x0
19:16	BUSRECOVERY[3:0]	Time between NEx high to NEx low.
15:8	DATABLD[7:0]	Duration of the second phase of write access, it is (DATABLD+1) HCLK cycle. This field cannot be 0, it must be at least 1.
7:4	ADDHLD[3:0]	0x0
3:0	ADDBLD[3:0]	Duration of the first phase of write access, it is (ADDBLD +1) HCLK cycle.

31.5.4.6 Mode D - Asynchronous Access With Extended Address

Figure 31-12 Mode D Read Accesses

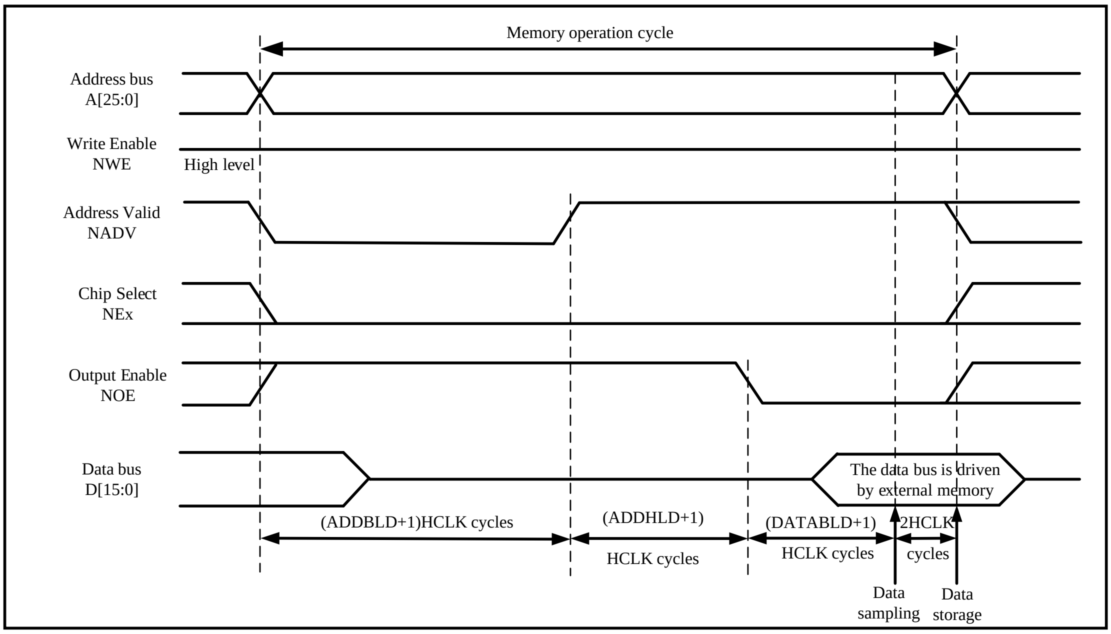
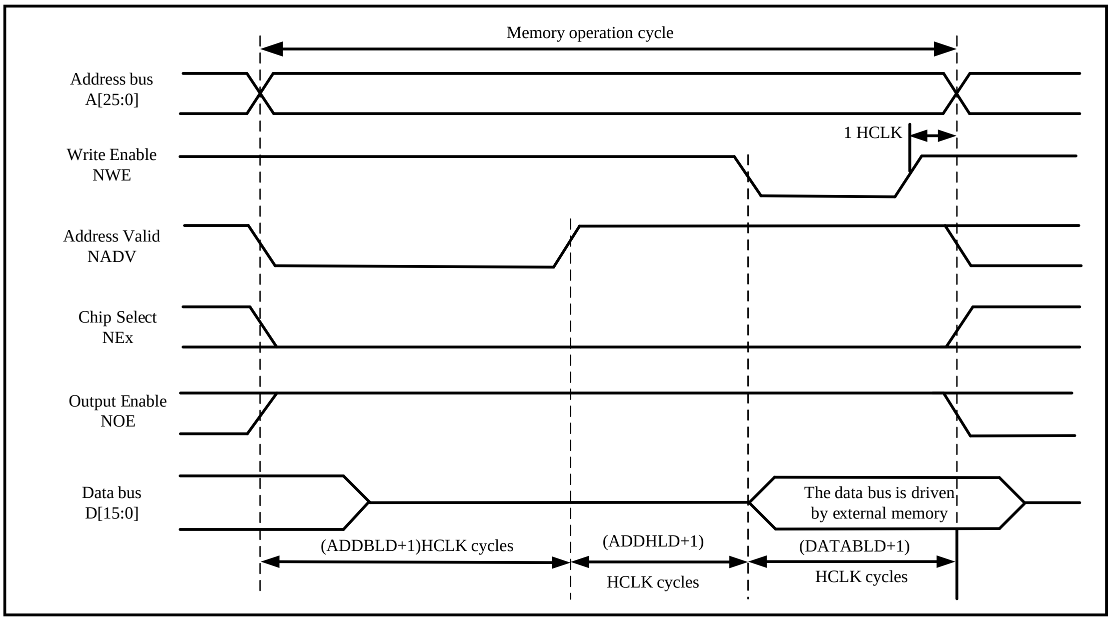


Figure 31-13 Mode D Write Accesses



Note: ADDBLD: address setup time. ADDHLD: address hold time. DATABLD: data setup time

31.5.4.6.1 Register Configuration in Mode D

The differences with mode1 are the toggling of NOE that goes on toggling after NADV changes and the independent read and write timings.

Table 31-21 FEMC_SNCTRLx Register Configuration(Mode D)

Bit Field	Name	Value to Set
31:20	Reserved	0x000
19	NADVDIR	0x0
18:16	PGSIZE[2:0]	0x0(invalid in asynchronous mode)
15	BURSTWREN	0x0(invalid in asynchronous mode)
14	WCFG	0x0(invalid in asynchronous mode)
13	WAITDIR	Only valid when bit 8 is set to '1'
12	WAITEN	0x0(invalid in asynchronous mode)
11	WRAPEN	0x0
10	BURSTEN	0x0
9	MUXEN	0x0
8	WAITASYNC	Set to 1 if the memory supports this feature, otherwise set to 0
7	EXTEN	0x1
6	ACCEN	Set according to memory support
5	WREN	As needed
4	MBEN	0x1
3:2	MDBW[1:0]	As needed
1:0	MTYPE[1:0]	Set according to memory support

Table 31-22 FEMC_SNTCFGx Register Configuration(Mode D)

Bit Field	Name	Value to Set
31:30	Reserved	0x0
29:28	ACCMODE[1:0]	0x3
27:24	CLKDIV[3:0]	0x0
23:20	BUSRECOVERY[3:0]	Time between NEx high to NEx low.
19:16	DATAHLD[3:0]	Don’t care
15:8	DATABLD[7:0]	Duration of the second phase of read access, it is (DATABLD+3) HCLK cycle. This field cannot be 0, it must be at least 1.
7:4	ADDHLD[3:0]	Duration of the address hold phase of the read access is (ADDHLD + 1) HCLK cycles.
3:0	ADDBLD[3:0]	Duration of the first phase of read access is (ADDBLD+1) HCLK cycle.

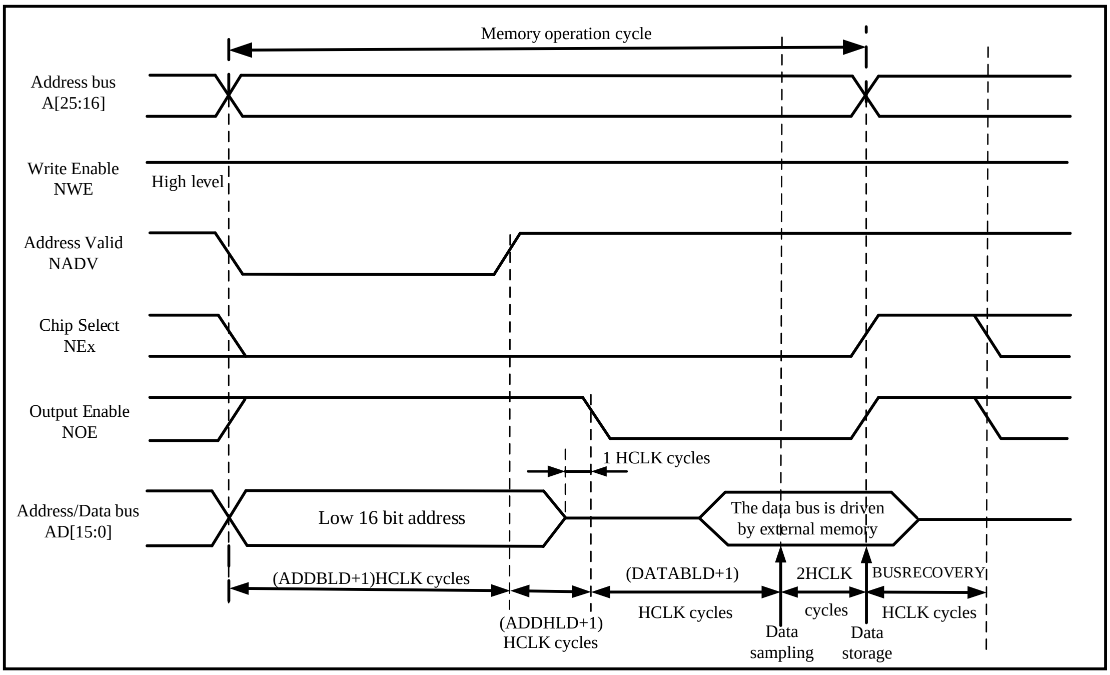
Table 31-23 FEMC_SNWTCFGx Register Configuration(Mode D)

Bit Field	Name	Value to Set
31:30	Reserved	0x0
29:28	ACCMODE[1:0]	0x3
27:20	Reserved	0x0
19:16	BUSRECOVERY[3:0]	Time between NEx high to NEx low.
15:8	DATABLD[7:0]	Duration of the second phase of write access is (DATABLD+1) HCLK cycle. This field cannot be 0, it must be at least 1.
7:4	ADDHLD[3:0]	Duration of the address hold phase of the write access is (ADDHLD + 1) HCLK

		cycles.
3:0	ADDBLD[3:0]	Duration of the first phase of write access is (DATABLD+1) HCLK cycle.

31.5.4.6.2 Muxed Mode-Multiplexed Asynchronous Access To NOR Flash Memory

Figure 31-7 Multiplexed Read Accesses

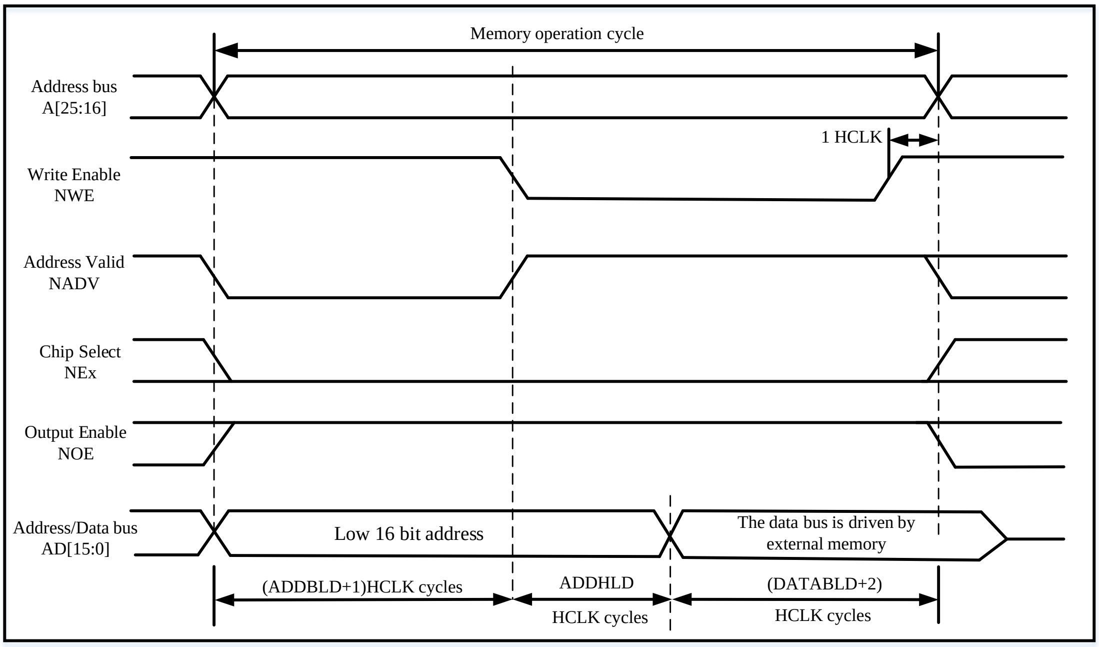


Notes:

1) *ADDBLD: address setup time.   DATABLD: data setup time   BUSRECOVERY: Bus recovery time*

2) *There is some overlap between the bus recovery delay (BUSRECOVERY+1) and the internal delay generated between two consecutive read accesses, so when BUSRECOVERY≤5, it will not affect the output timing.*

Figure 31-15 Multiplexed Write Accesses



Note: ADDBLD: address setup time. ADDHLD: address hold time. DATABLD: data setup time

31.5.4.6.3 Register Configuration in Multiplexed Mode

The difference between the multiplexed mode and mode D is that the low 16 bits of the address are driven on the data bus.

Table 31-24 FEMC_SNCTRLx Register Configuration(Mode D)

Bit Field	Name	Value to Set
31:20	Reserved	0x000
19	NADVDIR	0x0
18:16	PGSIZE[2:0]	0x0(invalid in asynchronous mode)
15	BURSTWREN	0x0(invalid in asynchronous mode)
14	WCFG	0x0(invalid in asynchronous mode)
13	WAITDIR	Only valid when bit 8 is set to '1'
12	WAITEN	0x0(invalid in asynchronous mode)
11	WRAPEN	0x0
10	BURSTEN	0x0
9	MUXEN	0x1
8	WAITASYNC	Set to 1 if the memory supports this feature, otherwise set to 0
7	EXTEN	0x1
6	ACCEN	0x1
5	WREN	As needed
4	MBEN	0x1
3:2	MDBW[1:0]	As needed
1:0	MTYPE[1:0]	0x2(NOR Flash)

Table 31-25 FEMC_SNTCFGx Register Configuration(Mode D)

Bit Field	Name	Value to Set
31:30	Reserved	0x0
29:28	ACCMODE[1:0]	0x0
27:24	CLKDIV[3:0]	0x0
23:20	BUSRECOVERY[3:0]	Time between NEx high to NEx low.
19:16	DATAHLD[3:0]	Don’t care
15:8	DATABLD[7:0]	Duration of the second phase of read access is (DATABLD+3) HCLK cycle, and the duration of the write access is (DATABLD+1) HCLK cycles.This field cannot be 0, it must be at least 1.
7:4	ADDHLD[3:0]	Duration of the address hold phase of the access is (ADDHLD + 1) HCLK cycles.
3:0	ADDBLD[3:0]	Duration of the first phase of access is (ADDHLD +1) HCLK cycle.

31.5.4.6.4 Wait Management in Asynchronous Accesses

Some asynchronous memories use the WAIT signal to indicate whether the memory is currently ready to transmit or receive data. FEMC supports this feature, and it can be enabled by setting FEMC_BANK_CRx.WAITASYNC.

In the second phase of the access (data setup phase, configure by the DATABLD bit), if the WAIT signal is active (the polarity is determined by the WAITDIR bit.), this phase is extended until the WAIT signal becomes inactive. Unlike the data setup phase, the address phase is not sensitive to the WAIT signal and will not be prolonged.

In the data setup phase, DATABLD needs to be configured to ensure that the WAIT signal meets the following conditions:

- For read access: the WAIT signal can be sampled 4 HCLK cycles before the data is sampled (or 6 HCLK cycles before NOE is deasserted), as shown in Figure 2-16.
- For write access: the WAIT signal can be sampled 4 HCLK cycles before NEW is deasserted, as shown in Figure 2-17.

Figure 31-16 Asynchronous Wait During A Read Access

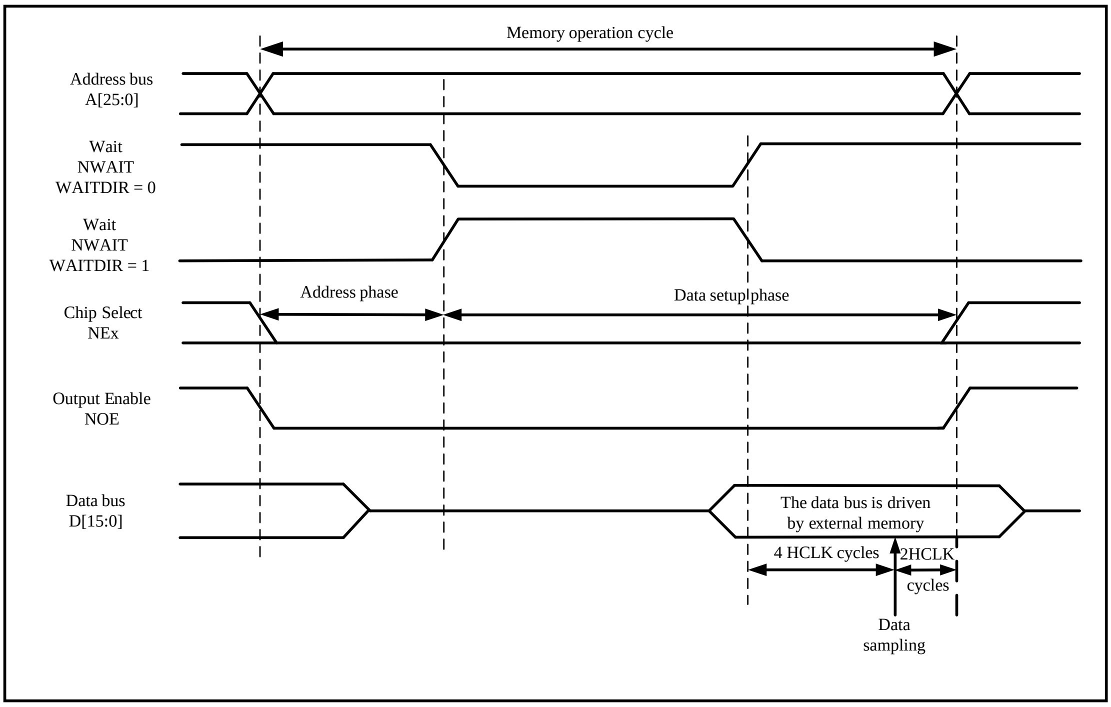
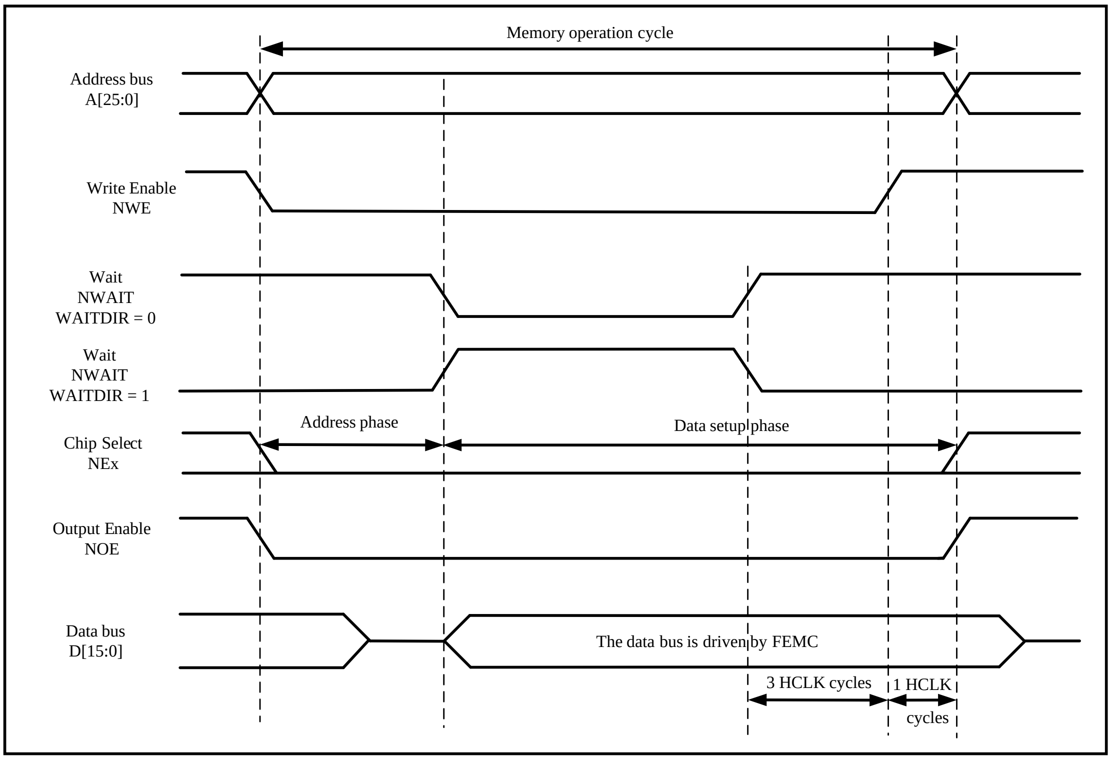


Figure 31-17 Asynchronous Wait During A Write Access



The setting of DATABLD needs to meet the following conditions:

- Align the moment when memory inserts WAIT with the moment when NOE/NEW toggles.

DATABLD ≥ (4× HCLK) + Tmax_wait_assertion_time

- Align the moment when memory inserts WAIT with the moment when NEx toggles(NOE/NEW does not toggle):
  - If max_wait_assertion_time address_phase hold_phase

    DATABLD ≥ (4× HCLK）+ Tmax_wait_assertion_time - Taddress_phase – Thold_phase

  - othersize

    DATABLD ≥ (4× HCLK)

The max_wait_assertion_time above is the maximum time that the memory inserts the WAIT signal once NEx/NOE/NEW goes low.

31.5.5 Synchronous Burst Read

In synchronous access mode, the relationship between the memory clock(CLK) and the system clock(HCLK) is: $CLK = \frac{HCLK}{CLKDIV+1}$

Where CLKDIV is the synchronous clock divider ratio, different values can be set by configuring the CLKDIV bit in the SRAM/NOR flash chip select timing register (FEMC_SNTCFG1/2/3/4).

NOR flash memory specify a minimum time from NADV being active to CLK rising. To meet this constraint, in the first internal clock cycle of synchronous access (before NADV assertion), FEMC will not output the clock to the memory. This ensures that the rising edge of the memory clock occurs in the middle of the low pulse of NADV.

31.5.5.1 Data Latency And NOR Flash Latency

The data hold time (DATAHLD) refers to the number of FEMC CLK cycles to wait before sampling data, and users can adjust the value of DATAHLD to match the value defined in the NOR flash configuration register. The FEMC data delay parameter does not include clock cycles when NADV is low. Its relationship with NOR flash delay is as follows:

- When the NOR flash delay does not include NADV, the NOR flash delay = DATAHLD + 2
- When the NOR flash delay includes NADV, the NOR flash delay = DATAHLD + 3

If the memory does not output the NWAIT signal during the data hold phase, FEMC must set its data hold time (DATAHLD) to be consistent with the data hold time at the memory end. Otherwise, data may be lost during the initial stage of accessing memory, or FEMC may not receive the correct data.

If the memory generates an NWAIT signal during the data hold phase, FEMC can set its data hold time (DATAHLD) to the minimum value. FEMC will sample the NWAIT signal and wait until the data is valid. FEMC will read the correct data after detecting that the memory has ended the hold phase.

31.5.5.2 Single-Burst-Transfer

When the selected bank is configured for synchronous burst mode, if only a single AHB burst transfer is required, and the AHB needs to transfer 16-bit data, FEMC will perform a burst transfer of length 1. If the AHB needs to transfer 32-bit data, FEMC will split it into two 16-bit transfers and perform a burst transfer of length 2. The chip select signal is deasserted when the last data transfer is completed.

31.5.5.3 Wait Management

For synchronous NOR flash burst access, after the programmed hold time (DATAHLD + 1 CLK clock cycle), the NWAIT signal needs to be detected.

- FEMC NWAIT Active Polarity:

  FEMC_SNCTRLx.WAITDIR = 1, NWAIT is active high

  FEMC_SNCTRLx.WAITDIR = 0, NWAIT is active low

- In synchronous burst mode, the NWAIT signal has two configurations:

  FEMC_SNCTRLx.WCFG= 1, when the NWAIT signal is active, the data in the current clock cycle is invalid.

  FEMC_SNCTRLx.WCFG= 0, when the NWAIT signal is active, the data in the next clock cycle is invalid, which is the default configuration after reset.

When NWAIT is detected at a valid level, FEMC will insert wait cycles until NWAIT becomes invalid.

When NWAIT becomes invalid, FEMC considers the data to be valid (FEMC_SNCTRLx.WCFG=1) or the data will be valid at the next clock edge (FEMC_SNCTRLx.WCFG=0).

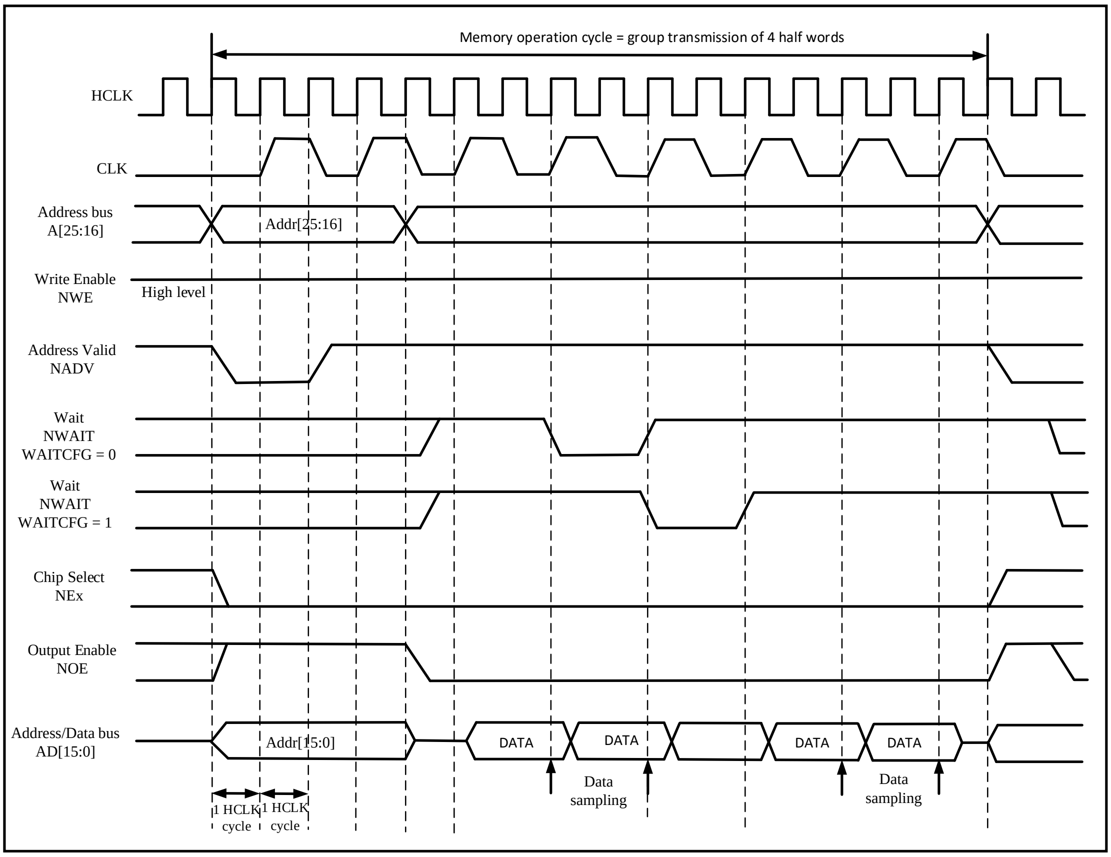
During the wait cycles when NWAIT is active, FEMC will continuously send clock pulses to the memory, keep the chip select signal and output enable active, while ignoring invalid data on the bus.

In burst transfer mode, the NWAIT signal of the NOR flash memory has 2 timing configurations:

- The flash memory inserts the NWAIT signal one data cycle before the wait state (default setting after reset).
- The flash memory inserts the NWAIT signal during the wait state.

By configuring the WCFG bit in the FEMC_SNCTRLx register, FEMC supports these 4 NOR flash wait state configurations on each chip select.

Figure 31-18 Synchronous Multiplexed Read Mode - NOR, PSRAM (CRAM)



The BL signal is not shown in the diagram. For NOR flash access, BL should be held high; for PSRAM (CRAM), BL should be held low.

31.5.5.3.1 Register Configuration in Synchronous Multiplexed Read Mode

Table 31-26 FEMC_SNCTRLx Register Configuration(Synchronous Mode)

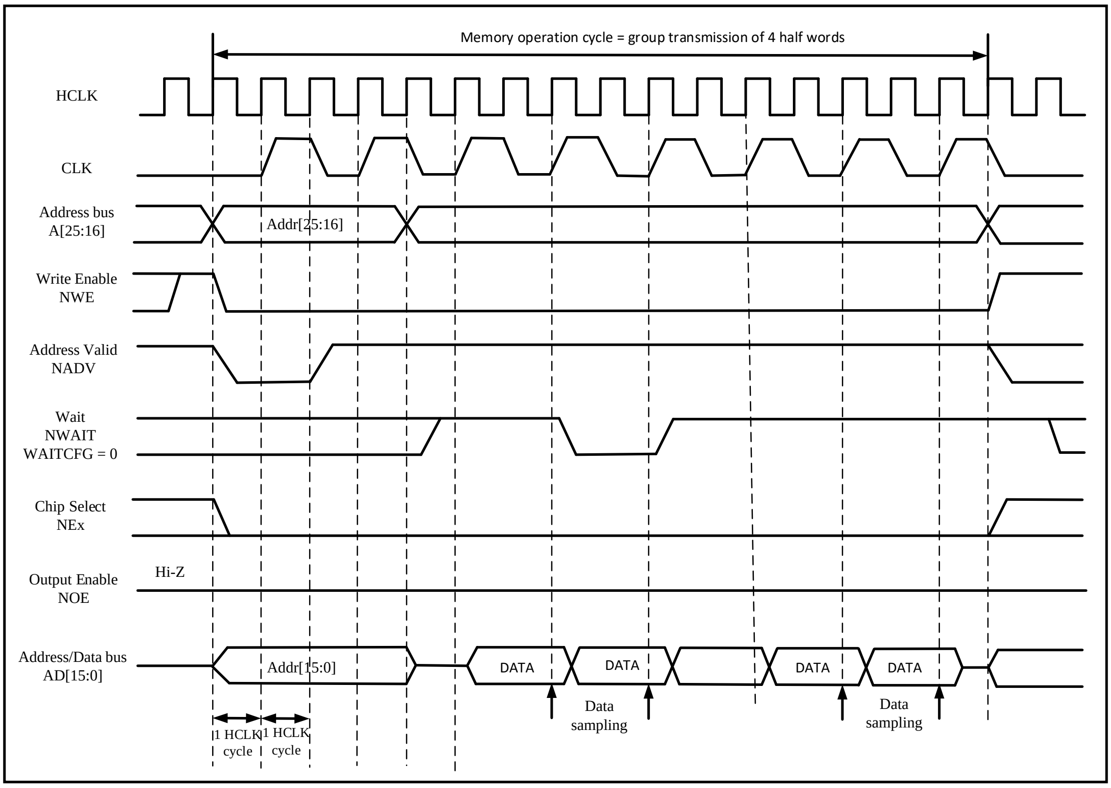
Bit Field	Name	Value to Set
31:20	Reserved	0x000
19	NADVDIR	0x0
18:16	PGSIZE[2:0]	As needed(CRAM1.5 is configured as 0x1)
15	BURSTWREN	Invalid in Synchronous read mode
14	WCFG	Set according to memory support
13	WAITDIR	Only valid when bit 8 is set to '1'
12	WAITEN	Set to 1 if the memory supports this feature, otherwise set to 0
11	WRAPEN	0x0
10	BURSTEN	0x0
9	MUXEN	0x0
8	WAITASYNC	0x0
7	EXTEN	0x0
6	ACCEN	0x0

5	WREN	Invalid in Synchronous read mode
4	MBEN	0x1
3:2	MDBW[1:0]	As needed
1:0	MTYPE[1:0]	As needed，0x1or 0x2

Table 31-27 FEMC_SNTCFGx Register Configuration (Synchronous Mode)

Bit Field	Name	Value to Set
31:30	Reserved	0x0
29:28	ACCMODE[1:0]	0x0
27:24	CLKDIV[3:0]	As needed
23:20	BUSRECOVERY[3:0]	Don’t care
19:16	DATAHLD[3:0]	Data latency
15:8	DATABLD[7:0]	Don’t care
7:4	ADDHLD[3:0]	Don’t care
3:0	ADDBLD[3:0]	Don’t care

Figure 31-19 Synchronous Multiplexed Write Mode - PSRAM (CRAM)



1. *The memory must generate the NWAIT signal one cycle in advance, while WAITCFG should be configured as 0.*

2. *The byte select BL outputs are not shown in the diagram, they are low when NEx is active*

31.5.5.3.2 Register Configuration in Synchronous Multiplexed Write Mode

Table 31-28 FEMC_SNCTRLx Register Configuration(Synchronous Mode)

Bit Field	Name	Value to Set
31:20	Reserved	0x000
19	NADVDIR	0x0
18:16	PGSIZE[2:0]	As needed(CRAM1.5 is configured as 0x1)
15	BURSTWREN	0x1
14	WCFG	0x0
13	WAITDIR	Only valid when bit 8 is set to '1'
12	WAITEN	Set to 1 if the memory supports this feature, otherwise set to 0
11	WRAPEN	0x0
10	BURSTEN	Invalid in Synchronous write mode
9	MUXEN	As needed
8	WAITASYNC	0x0
7	EXTEN	0x0
6	ACCEN	Set according to memory support
5	WREN	0x1
4	MBEN	0x1
3:2	MDBW[1:0]	As needed
1:0	MTYPE[1:0]	0x1

Table 31-29 FEMC_SNTCFGx Register Configuration(Synchronous Mode)

Bit Field	Name	Value to Set
31:30	Reserved	0x0
29:28	ACCMODE[1:0]	0x0
27:24	CLKDIV[3:0]	As needed
23:20	BUSRECOVERY[3:0]	Time between NEx high to NEx low.
19:16	DATAHLD[3:0]	Data latency
15:8	DATABLD[7:0]	Don’t care
7:4	ADDHLD[3:0]	Don’t care
3:0	ADDBLD[3:0]	Don’t care

31.6 NAND Flash Controller

The FEMC module supports 8-bit and 16-bit NAND Flash in Bank2 and Bank3. For each bank, FEMC provides independent registers to configure access timing and also includes an ECC calculation module to ensure data transmission and storage reliability.

Table 31-30 Programmable NAND

Parameters	Function	Operation Mode	Unit	Min.	Max.
Memory setup time	Number of (HCLK) clock cycles to establish the address before	read/write	AHB clock cycle （HCLKs	2	256

Parameters	Function	Operation Mode	Unit	Min.	Max.
	issuing the command				
Memory wait time	Minimum duration (in HCLK clock cycles) for issuing a command	read/write		2	255
Memory hold time	Number of (HCLK) clock cycles to hold the address after sending the command, also the data hold time during write access	read/write		2	255
memory data bus hold time in high-impedance state	The time to hold the data bus in high-impedance state after initiating a write access	write		1	255

31.6.1 External Memory Interface Singals

The table below lists the typical signal lines used for interfacing with NAND flash memory. The prefix 'N' indicates that the corresponding signal line is active low.

31.6.1.1 8-bit NAND Flash

Table 31-31 8-bit NAND Flash

FEMC Signal Name	I/O	Function
A[17]	O	NAND Flash address latch enable signal (ALE)
A[16]	O	NAND Flash command latch enable signal（CLE）
D[7:0]	I/O	8-bit multiplexed, bidirectional address/data bus
NCE[x]	O	chip select，x = 2,3
NOE	O	output enable
NWE	O	write enable
NWAIT	I	NAND Flash ready/busy signal, input to FEMC
INT[x]	I	Interrupt input， x = 2,3

FEMC can generate multiple address cycles as needed, theoretically FEMC does not limit the NAND capacity that can be accessed.

31.6.1.2 16-bit NAND Flash

Table 31-32 16-bit NAND Flash

FEMC Signal Name	I/O	Function
A[17]	O	NAND Flash address latch enable signal (ALE)
A[16]	O	NAND Flash command latch enable signal（CLE）

D[15:0]	I/O	16-bit multiplexed, bidirectional address/data bus
NCE[x]	O	chip select, x = 2,3
NOE	O	output enable（memory-side signal name: read enable, NRE）
NWE	O	write enable
NWAIT	I	NAND Flash ready/busy signal, input to FEMC
INT[x]	I	Interrupt input, x = 2,3

31.6.2 Supported Memories And Transactions for NAND Flash

The table below lists the supported devices, operation modes, and operation methods. Shaded areas in the table indicate that the corresponding operation method is not supported by the NAND flash.

Table 31-33 Supported Memories And Transactions

Device	Mode	R/W	AHB Data Size	Memory Data Size	Allowed/Not allowed	Comments
8-bit NAND	Asynchronous	R	8	8	Y	
	Asynchronous	W	8	8	Y	
	Asynchronous	R	16	8	Y	Split into two FEMC accesses
	Asynchronous	W	16	8	Y	Split into two FEMC accesses
	Asynchronous	R	32	8	Y	Split into four FEMC accesses
	Asynchronous	W	32	8	Y	Split into four FEMC accesses
16-bit NAND	Asynchronous	R	8	16	Y	
	Asynchronous	W	8	16	N	
	Asynchronous	R	16	16	Y	
	Asynchronous	W	16	16	Y	
	Asynchronous	R	32	16	Y	Split into two FEMC accesses
	Asynchronous	W	32	16	Y	Split into two FEMC accesses

31.6.3 NAND Flash Transfer Timing

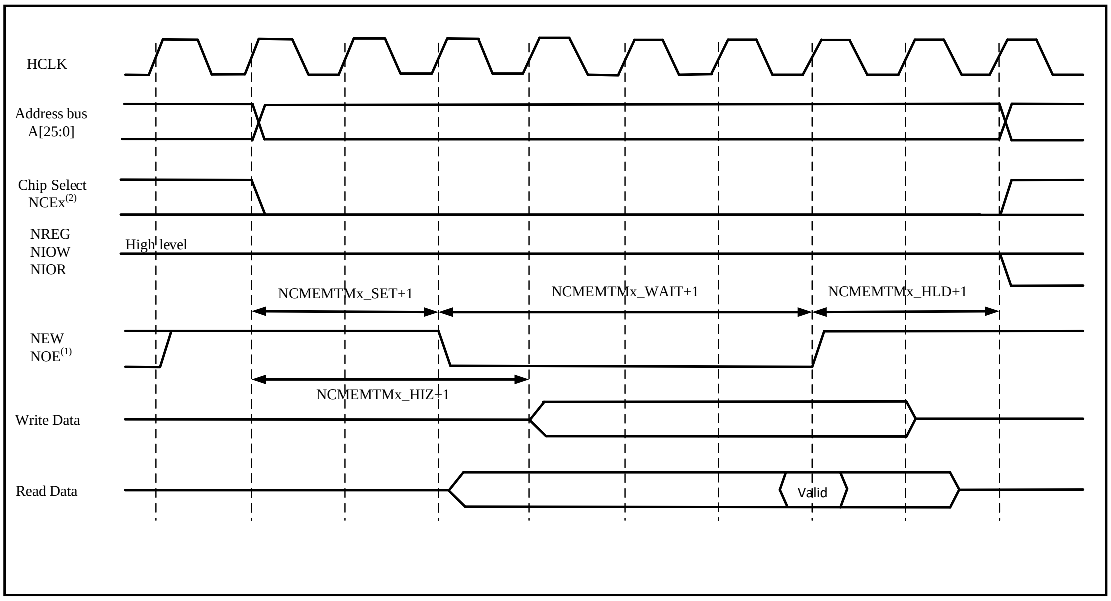
FEMC can generate appropriate timing signals for Nand Flash, and each Bank has corresponding registers to manage and control external memory. The register groups involved are as follows:

- Control registers: FEMC_NCTRLx
- Interrupt status registers: FEMC_NSTSx
- ECC registers: FEMC_ECCx
- Timing registers for common memory space: FEMC_NCMEMTM x

- Timing registers for attribute memory space: FEMC_NATTMEMTM x

Each timing control register contains three parameters to define the number of HCLK cycles for the three stages of NAND flash operation, as well as a parameter that defines the timing for when FEMC starts driving the data bus during write operations. These parameters can be configured according to user requirements and the characteristics of the external memory. The following diagram provides the timing parameter definitions for operations in the common memory space, with similar operations in the attribute memory space.

Figure 31-20 Common Memory Space Access Timing for NAND Flash



1. *During write operations, NOE is always held high (inactive), and during read operations, NWE is always held high (inactive).*
2. *As long as there is a request for NAND access, the NCEx signal goes low and remains low before accessing other memory blocks.*

31.6.4 NAND Flash Operations

The NAND flash's command latch enable (CLE) and address latch enable (ALE) signals are driven by FEMC's address signal lines. This means that when FEMC sends commands or addresses to the NAND Flash, it needs to use the command latch signal (A[16]) or address latch signal (A[17]). In other words, the CPU needs to perform write operations at specific addresses.

Steps for NAND Flash read operation:

1. Configure and enable the corresponding memory blocks using the FEMC_NCTRLx and FEMC_NCMEMTMx registers according to the characteristics of the NAND flash. For some NAND flashes, if pre-waiting function is required, the FEMC_NATTMEMTMx register may need to be configured (refer to section 31.6.5 - NAND Flash Pre-waiting Function). The bits that need to be configured include: BUSWID indicating the data bus width of the NAND flash, WAITEN = 1, BANKEN = 1, refer to the timing configuration of the FEMC_NCMEMTMx register.
2. Write the NAND Flash read data command to the common space, that is, during the low pulse of chip select and

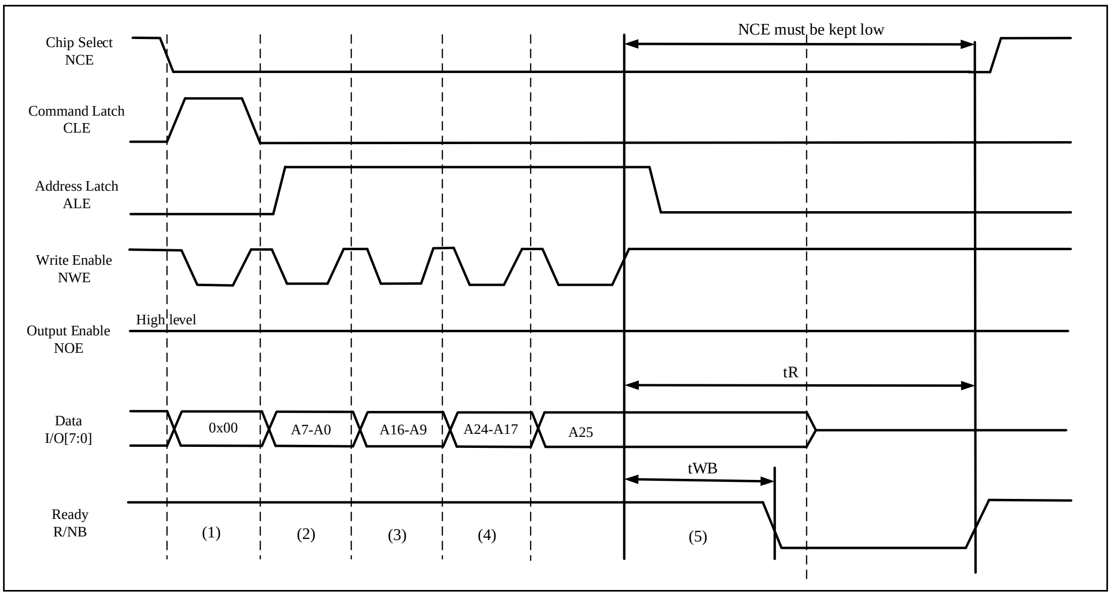
write signals (NCE and NWE), the CLE (A16) of the NAND flash becomes active (high), at this point the written bytes are recognized by the NAND flash as a command. Once the NAND flash latches this command, subsequent page read operations do not need to send the same command.

3. Write the starting address for the read operation to the common memory space (large-capacity NAND flash requires 4 bytes, while smaller capacity NAND flash may only require three bytes). That is, during the low pulse of chip select and write signals (NCE and NWE), the ALE (A17) of the NAND flash becomes active (high), at this point the written bytes are recognized by the NAND flash as the starting address of the read operation. By using the attribute memory space, FEMC can generate different timing to implement the pre-waiting function required by some NAND flashes (refer to section 31.6.5 - NAND Flash Pre-waiting Function).
4. Wait for the NAND flash to be ready (R/B signal goes high), during this waiting period, the NAND controller keeps the NCE signal continuously active (low).
5. Byte read operations can be performed in the common memory space, reading out the NAND flash storage page byte by byte (data field and spare field).
6. Without writing commands or addresses, the next page of the NAND flash can be read out in any of the following ways:
   a) Proceed with the operation as in step 5
   b) Return to step 3 to input a new address
   c) Return to step 2 to input a new command

31.6.5 NAND Flash Pre-waiting Function

Some NAND Flash devices require the controller to wait for the NAND Flash to be ready after inputting the last address byte, indicated by the R/B signal going low. Additionally, some NAND Flash devices sensitive to the FEMC chip select signal(NCE) require NCE to remain active before the NAND Flash is ready. See the diagram below:

Figure 31-21 Transactions for CE-Sensitive NAND Flash



1) CPU writes byte command 0x00 to the NAND's common space command area (0x7001 0000).

2) CPU writes operation address A7~A0 to the NAND's common space address area (0x7002 0000).

3) CPU writes operation address A16~A9 to the NAND's common space address area (0x7002 0000).

4) CPU writes operation address A24~A17 to the NAND's common space address area (0x7002 0000).

5) CPU writes NAND address A25 to the NAND's attribute space address area (0x7802 0000). At this point, FEMC uses the timing definition of FEMC_NATTMEMTM to perform the write operation, where FEMC_NATTMEMTx.HLD≥7 (here: (7+1)xHCLK=112ns > maximum value of tWB). This ensures that NCE remains low during the process of R/B going low and then high, a requirement only for CE-sensitive NAND Flash devices.

When such functionality is required, the value of FEMC_NMEMTMx.HLD can be configured to ensure the timing of t_{WB}. However, for any subsequent read or write operations of the NAND flash by the CPU, the FEMC controller will insert a hold delay between the rising edge of the NWE signal and the next operation, with a delay length of (FEMC_NMEMTMx.HLD+1) HCLK cycles.

To overcome this timing constraints, the value of FEMC_NATTMEMTMx.HLD is configured in the attribute memory space to conform to the t_{WB} timing, while keeping FEMC_NMEMTMx.HLD at its minimum value. In this case, the CPU must use the common memory space for all read and write operations of the NAND flash, and only write to the attribute memory space when writing the last byte of the NAND flash address.

31.6.6 ECC Calculation Method for NAND Flash

FEMC has a hardware module for ECC calculation in each of bank 2 and 3, which can reduce the software workload on the CPU when handling error correction codes. There are two identical registers (FEMC_ECC2, FEMC_ECC3) corresponding to bank 2 and 3, respectively.

The ECC algorithm implemented in FEMC can correct 1 bit error and detect 2 bit errors in every 256, 512, 1024, 2048, 4096, or 8192 bytes during read or write operations on the NAND flash. Users can choose the page size for ECC calculation based on ECCPGS in FEMC_NCTRLx.

When the NAND memory block is enabled, the ECC module monitors the data bus D[15:0] of the NAND flash as well as the NCE and NWE signals. Once the read or write operation of ECCPGS-sized bytes is completed, the hardware performs ECC calculation and stores the result in the FEMC_ECCx register. Software can retrieve the ECC value by reading the result from FEMC_ECCx. If ECC calculation needs to be restarted, the software must first clear the value in the FEMC_ECC register by setting ECCEN to 0 in FEMC_NCTRLx, and then set ECCEN to 1 to restart ECC calculation.

The ECC calculation rules are as follows:

- When accessing the NAND flash in Bank2 or Bank3, the data on the D[15:0] bus is latched and used for ECC calculation.
- When operations on the NAND flash occur at other addresses, the ECC circuitry does not perform any operation. Therefore, write operations for outputting NAND flash commands and addresses do not participate in ECC calculation.
- After the specified number of bytes have been written to or read from the NAND flash, the software must read

the FEMC_ECCx.ECC register to obtain the calculated ECC value. To calculate ECC again after reading the ECC value, the software must clear this register by setting ECCEN to '0' and then enable ECC calculation again by writing '1' to the ECCEN bit in the FEMC_NCTRL2/3 register.

When performing ECC calculation:

1) Enable the ECCEN bit in the FEMC_NCTRL2/3 register.
2) Write data to the NAND Flash, during the NAND write process, the ECC module calculates the ECC result.
3) Read the ECC result from the FEMC_NCTRL2/3 register and store it in a variable.
4) Clear ECCEN in the FEMC_NCTRL2/3 register, then re-enable ECCEN. Subsequently, read back the written data from the NAND, the ECC module will calculate the ECC value for the read operation.
5) Read the new ECC result from the FEMC_NCTRL2/3 register. If the two ECC results are the same, no correction is needed. Otherwise, if an ECC error occurs, the software correction program will return information on whether the error can be corrected.

The principle of ECC error correction is as follows:

In ECC data, two bits are used to represent one bit of data. For example, with 512 bytes, there are a total of 512*8=4096 bits. To represent up to 4096 numbers, which is exactly 12 bits, but according to the table "Valid Bits of ECC Result" (Table 31 36), it can be seen that the ECC for 512 bytes uses 24 bits, and for 1024 bytes, it uses 26 bits. As a rule, an additional bit can represent twice the data. However, for ECC to represent twice the data, it requires two more bits, as seen in the pattern.

Taking the example of generating an ECC data for 512 bytes, according to the table "Valid Bits of ECC Result" (Table 31 36), the valid bits of ECC range from 0 to 23. When writing 512 bytes, an ECC data is generated. Then, when reading 512 bytes, another ECC data is generated. By XORing these two data sets, an ECC verification result is produced. Based on this result, it can be analyzed whether there are errors, where the errors are, and which bit is in error. Knowing which bit is in error enables the correction to be made.

Here is an example to illustrate the ECC error correction principle:

1. For ease of data analysis, write 512 bytes of 0x11 to the NAND Flash (other data can also be used as long as there is a pattern), and then read out the ECC data as 0.
2. Change a bit of one of the bytes, for example, change the 0th bit of the 0th byte (change the 0th byte to 0x10), and read out the ECC data as 0x55555555.
3. Change the 1st bit of the 0th byte (change the 0th byte to 0x13), and read out the ECC data as 0x55555556.
4. Change the 2nd bit of the 0th byte (change the 0th byte to 0x15), and read out the ECC data as 0x55555559.
5. Change the 3rd bit of the 0th byte (change the 0th byte to 0x19), and read out the ECC data as 0x5555555a.

In this example, we primarily write 0x11, then intentionally introduce an error by changing a bit, and read out the ECC. Since the data written is all 0x11, the ECC generated during writing is 0x0. XORing the read ECC with 0x0 gives the ECC verification result. The table below shows the ECC data corresponding to the test results mentioned above.

Table 31-34 Table of ECC Error Bits and ECC Check Values

Error Bit	Check Result（Lower 24 Bits）	Corresponding 24 Bits
Bit0	0x555555	010101010101010101010101
Bit1	0x555556	010101010101010101010110
Bit2	0x555559	010101010101010101011001
Bit3	0x55555a	010101010101010101011010

At this point, the pattern becomes clear. In the verification result, 2 bits are used to represent one bit of data, where 01 represents 0 and 10 represents 1. For example: the 24-bit data with an error in the 0th bit (010101010101010101010101) decodes to 12 bits as: 000000000000 (each 01 represents 0), which is 0x0, indicating an error in the 0th bit. Similarly, for the 1st bit error, the decoded 12 bits are 000000000001, which is 0x1, indicating an error in the 1st bit. For the 2nd bit error, the decoded 12 bits are 000000000010, which is 0x2, indicating an error in the 2nd bit. And for the 3rd bit error, the decoded 12 bits are 000000000011, which is 0x3, indicating an error in the 3rd bit. At this stage, the error correction principle becomes clear. Once the error is identified, the correction is straightforward by flipping the erroneous bit from 0 to 1 or from 1 to 0.

31.7 FEMC Register

FEMC Base Address: 0xA000 0000

31.7.1 FEMC Register Overview

Table 31-12 FEMC Register Overview

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
000h	FEMC_SNCTRL1	Reserved												NADVDIR	PGSIZE[2:0]			BURSTEN	WCFG	WAITDIR	WAITEN	WRAPEN	BURSTEN	MUXEN	WAITASYNC	EXTEN	ACCEN	WREN	MBEN	MDBW[1:0]		MTYPE[1:0]	
	Reset Value													0	0	0	0	0	0	0	1	0	0	1	0	0	1	1	1	0	1	1	0
004h	FEMC_SNTCFG1	Reserved		ACCMODE[1:0]		CLKDIV[3:0]				BUSRECOVERY[3:0]				DATAHLD[3:0]				DATABLD[7:0]								ADDHLD[3:0]				ADDBLD[3:0]			
	Reset Value			0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
008h	FEMC_SNWTCFG1	Reserved		ACCMODE[1:0]		Reserved								BUSRECOVERY[3:0]				DATABLD[7:0]								ADDHLD[3:0]				ADDBLD[3:0]			
	Reset Value			0	0									1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
00Ch	FEMC_SNCTRL2	Reserved												NADVDIR	PGSIZE[2:0]			BURSTEN	WCFG	WAITDIR	WAITEN	WRAPEN	BURSTEN	MUXEN	WAITASYNC	EXTEN	ACCEN	WREN	MBEN	MDBW[1:0]		MTYPE[1:0]	
	Reset Value													0	0	0	0	0	0	0	1	0	0	1	0	0	1	1	1	0	1	1	0
010h	FEMC_SNTCFG2	Reserved		ACCMODE[1:0]		CLKDIV[3:0]				BUSRECOVERY[3:0]				DATAHLD[3:0]				DATABLD[7:0]								ADDHLD[3:0]				ADDBLD[3:0]			
	Reset Value			0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
014h	FEMC_SNWTCFG2	Reserved		ACCMODE[1:0]		Reserved								BUSRECOVERY[3:0]				DATABLD[7:0]								ADDHLD[3:0]				ADDBLD[3:0]			
	Reset Value			0	0									1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
018h	FEMC_SNCTRL3	Reserved												NADVDIR	PGSIZE[2:0]			BURSTEN	WCFG	WAITDIR	WAITEN	WRAPEN	BURSTEN	MUXEN	WAITASYNC	EXTEN	ACCEN	WREN	MBEN	MDBW[1:0]		MTYPE[1:0]	
	Reset Value													0	0	0	0	0	0	0	1	0	0	1	0	0	1	1	1	0	1	1	0
01Ch	FEMC_SNTCFG3	Reserved		ACCMODE[1:0]		CLKDIV[3:0]				BUSRECOVERY[3:0]				DATAHLD[3:0]				DATABLD[7:0]								ADDHLD[3:0]				ADDBLD[3:0]			
	Reset Value			0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
020h	FEMC_SNWTCFG3	Reserved		ACCMODE[1:0]		Reserved								BUSRECOVERY[3:0]				DATABLD[7:0]								ADDHLD[3:0]				ADDBLD[3:0]			
	Reset Value			0	0									1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
024h	FEMC_SNCTRL4	Reserved												NADVDIR	PGSIZE[2:0]			BURSTEN	WCFG	WAITDIR	WAITEN	WRAPEN	BURSTEN	MUXEN	WAITASYNC	EXTEN	ACCEN	WREN	MBEN	MDBW[1:0]		MTYPE[1:0]	
	Reset Value													0	0	0	0	0	0	0	1	0	0	1	0	0	1	1	1	0	1	1	0
028h	FEMC_SNTCFG4	Reserved		ACCMODE[1:0]		CLKDIV[3:0]				BUSRECOVERY[3:0]				DATAHLD[3:0]				DATABLD[7:0]								ADDHLD[3:0]				ADDBLD[3:0]			
	Reset Value			0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
02Ch	FEMC_SNWTCFG4	Reserved		ACCMODE[1:0]		Reserved								BUSRECOVERY[3:0]				DATABLD[7:0]								ADDHLD[3:0]				ADDBLD[3:0]			
	Reset Value			0	0									1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
060h	FEMC_NCTRL2	Reserved															ECCPGS[2:0]			ECCEN	ARDLY[3:0]				CRDLY[3:0]				WAITEN	BUSWID[1:0]		Reserved	BANKEN
	Reset Value																0	0	0	0	0	0	0	0	0	0	0	0	0	0	1		0
064h	FEMC_NSTS2	Reserved																									FIFOEMPT	IHLIEN	IREEN	IFEEN	IHLF	IREF	IFEF
	Reset Value																										1	0	0	0	0	0	0
068h	FEMC_NCMEMTM2	WAIT[7:0]								SET[7:0]								HIZ[7:0]								HLD[7:0]							
	Reset Value	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0
06Ch	FEMC_NATTMEMTM1	WAIT[7:0]								SET[7:0]								HIZ[7:0]								HLD[7:0]							
	Reset Value	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0
074h	FEMC_ECC2	ECC[31:0]																															
	Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
080h	FEMC_NCTRL3	Reserved															ECCPGS[2:0]			ECCEN	ARDLY[3:0]				CRDLY[3:0]				WAITEN	BUSWID[1:0]		Reserved	BANKEN
	Reset Value																0	0	0	0	0	0	0	0	0	0	0	0	0	0	1		0
084h	FEMC_NSTS3	Reserved																									FIFOEMPT	IHLIEN	IREEN	IFEEN	IHLF	IREF	IFEF
	Reset Value																										1	0	0	0	0	0	0
088h	FEMC_NCMEMTM3	WAIT[7:0]								SET[7:0]								HIZ[7:0]								HLD[7:0]							
	Reset Value	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0
08Ch	FEMC_NATTMEMTM3	WAIT[7:0]								SET[7:0]								HIZ[7:0]								HLD[7:0]							
	Reset Value	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0
094h	FEMC_ECC3	ECC[31:0]																															
	Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

31.7.2 NOR Flash And PSRAM Controller Register

These external peripheral registers must be accessed in word (32-bit) operations.

FEMC register base address: 0xA000 0000

31.7.2.1 SRAM/NOR Flash Chip Select Control Register（FEMC_SNCTRL1/2/3/4）

Address Offset：0x000 + C * (x-1), (x=1,2,3,4)

Reset Value：0x0000 1276

This register contains control information for each bank and can be used for SRAM, ROM, asynchronous, or burst mode NOR flash memory.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
												NADVDIR	PGSIZE[2:0]		
rw	rw	rw										rw		rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BURST	WCFG	WAITDIR	WAITEN	WRAPEN	BURSTEN	MUXEN	WAIT	EXTEN	ACCEN	WREN	MBEN	MDBW[1:0]		MTYPE[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw		rw	

Bit Field	Name	Description
31:20	Reserved	Reserved, the reset value must be maintained
19	NADVDIR	NADV output pority control 0: NADV signal is active low 1: NADV signal is active high
18:16	PGSIZE[2:0]	CRAM page size. These are used for Cellular RAM 1.5 which does not allow burst access to cross the address boundaries between pages. When these bits are configured, the FEMC controller splits automatically the burst access when the memory page size is reached (refer to memory datasheet for page size). 000: reserved(1024byte) 001: 128 bytes 010: 256 bytes 011: 512 bytes 100: 1024 bytes Others: reserved.
15	BURSTWREN	Write burst enable. For Cellular RAM (PSRAM) memories, this bit enables the synchronous burst protocol during write operations. For Flash memory in burst transfer mode, this bit allows/disallows inserting wait state via the NWAIT signal. The synchronous burst transfer protocol enable bit is the FEMC_SNCTRLx.BURSTEN bit. 0: Write operations are always performed in asynchronous mode 1: Write operations are performed in synchronous mode.
14	WCFG	Wait timing configuration. The NWAIT signal indicates whether the data from the memory are valid or if a wait state must be inserted when accessing the Flash memory in burst mode. This configuration bit determines if NWAIT is asserted by the memory one clock cycle before the wait state or during the wait state: 0: NWAIT signal is active one data cycle before wait state (default after reset), 1: NWAIT signal is active during wait state (not used for PRAM).
13	WAITDIR	Wait signal polarity bit. Defines the polarity of the wait signal from memory. Valid only when accessing the memory in burst mode: 0: NWAIT active low (default after reset), 1: NWAIT active high.
12	WAITEN	Wait enable bit. This bit enables/disables wait-state insertion via the NWAIT signal when accessing the Flash

		memory in burst mode. 0: Disable the NWAIT signal, no detection of the NWAIT signal for wait state insertion after the set flash hold period. 1: Enable the NWAIT signal, wait state insertion based on the NWAIT signal after the set flash hold period(this is the default state after reset).
11	WRAPEN	Wrapped burst mode support. Defines whether the controller will or not split an AHB burst wrap access into two linear accesses. Valid only when accessing memories in burst mode 0: Direct wrapped burst is not enabled (this is the default state after reset), 1: Direct wrapped burst is enabled
10	BURSTEN	Burst enable bit. Allows for burst mode access to the flash memory; this bit is only valid in the synchronous burst mode of the flash memory. 0: Disable burst mode (this is the default state after reset). 1: Enable burst mode.
9	MUXEN	Address/Data Multiplexing Enable Bit: When this bit is set, the low 16 bits of the address and data will share the data bus. This bit is only valid for NOR and PSRM memory. 0: Address/data not multiplexed. 1: Address/data multiplexed on the data bus(this is the default state after reset).
8	WAITASYNC	Wait signal during asynchronous transfers This bit enables/disables the FEMC to use the wait signal even during an asynchronous protocol. 0: NWAIT signal is not taken into accountwhen running an asynchronous protocol (default after reset) 1: NWAIT signal is taken into account when running an asynchronous protocol
7	EXTEN	Extended mode enable. This bit allows FEMC to use the FEMC_SNWTCFGx register, enabling different timing for read and write operations. 0: Do not use the FEMC_SNWTCFGx register, this is the default state after reset. 1: FEMC uses the FEMC_SNWTCFGx register.
6	ACCEN	Flash access enable Enables NOR Flash memory access operations. 0: Corresponding NOR Flash memory access is disabled 1: Corresponding NOR Flash memory access is enabled
5	WREN	Write Enable Bit This bit indicates whether FEMC allows/disallows write operations to the memory. 0: Disables write operations to the memory by FEMC, resulting in an AHB error. 1: Allows write operations to the memory by FEMC, this is the default state after reset.
4	MBEN	Memory bank enable bit. Enables the memory bank. After reset Bank1 is enabled, all others are disabled. Accessing a disabled bank causes an ERROR on AHB bus. 0: Corresponding memory bank is disabled

		1: Corresponding memory bank is enabled
3:2	MDBW[1:0]	Memory data bus width. Defines the external memory device width, valid for all type of memories. 00: 8 bits, 01: 16 bits (this is the default state after reset.), Others: reserved
1:0	MTYPE[1:0]	Memory Type Defines the type of external memory 00: SRAM, ROM (default after reset for memory blocks 2...4) 01: PSRAM (Cellular RAM: CRAM) 10: NOR flash (default after reset for memory block 1) 11: Reserved

31.7.2.2 SRAM/NOR Flash Chip Select Timing Register（FEMC_SNTCFG1/2/3/4）

Address Offset: 0x004 + C * (x-1), (x=1,2,3,4)

Default Value: 0x0FFFFFFF

This register contains control information for each memory block and can be used for SRAM, ROM, and NOR flash memory. If the EXTEN bit is set in the FEMC_SNCTRLx register, there are two timing registers corresponding to read operations (this register) and write operations (FEMC_SNWTCFGx register).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		ACCMODE[1:0]		CLKDIV[3:0]				BUSRECOVERY[3:0]				DATAHLD[3:0]			
		rw		rw				rw				rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATABLD[7:0]								ADDHLD[3:0]				ADDBLD[3:0]			
rw								rw				rw			

Bit Field	Name	Description
31:30	Reserved	Reserved, the reset value must be maintained
29:28	ACCMODE[1:0]	Access mode Specifies the asynchronous access modes.These bits are taken into account only when the FEMC_SNCTRLx.EXTEN bit is 1. 00: access mode A 01: access mode B 10: access mode C 11: access mode D
27:24	CLKDIV[3:0]	Clock Divide Ratio (for CLK signal): Defines the period of the CLK clock output signal in terms of HCLK cycles. 0000: Reserved 0001: 1 CLK cycle = 2 HCLK cycles 0010: 1 CLK cycle = 3 HCLK cycles ... 1111: 1 CLK cycle = 16 HCLK cycles (default value after reset).

		This parameter does not have an effect when accessing asynchronous NOR flash, SRAM, or ROM. *Note: When the clock frequency division coefficient is odd, the output clock duty cycle is not 50%. Suggest configuring the clock frequency division coefficient as even.*
23:20	BUSRECOVERY[3:0]	Bus turnaround phase duration These bits are used to define the delay on the bus after a read operation (only applicable for NOR flash operations in bus multiplexing mode). After a read operation, the controller needs to send the address on the data bus for the next operation. This delay is to prevent bus conflicts. If the extended memory system does not include memory with bus multiplexing mode, or if the slowest memory can return the data bus to high-impedance state within 6 HCLK clock cycles, this parameter can be set to its minimum value. 0000：Bus turnaround phase duration=1 HCLK clock cycle …… 1111：Bus turnaround phase duration=16 HCLK clock cycle（this is the default state after reset）。
19:16	DATAHLD[3:0]	Data latency（for synchronous burst NOR Flash） NOR flash memory in synchronous burst mode requires a defined number of memory cycles to wait before reading the first data. This timing parameter is not expressed in terms of HCLK, but in terms of flash clock (CLK). This parameter does not apply when accessing asynchronous NOR flash, SRAM, or ROM. When operating with CRAM, this parameter must be set to 0. 0000: The hold time for the first data is 2 CLK clock cycles ...... 1111: The hold time for the first data is 17 CLK clock cycles (this is the default value after reset).
15:8	DATABLD[7:0]	Data-phase duration These bits define the hold time for data and apply to SRAM, ROM, and NOR flash operations in asynchronous bus multiplex mode. 0000 0000: Reserved 0000 0001: DATABLD hold time = 2 HCLK clock cycles 0000 0010: DATABLD hold time = 3 HCLK clock cycles ...... 1111 1111: DATABLD hold time = 256 HCLK clock cycles (default value after reset). For the data hold time for each memory type and access mode, refer to the corresponding chart. For example: Mode 1, read operation, DATABLD=1: Data hold time = DATABLD + 3 = 4 HCLK

		clock cycles.
7:4	ADDHLD[3:0]	Address-hold phase duration These bits define the hold time for the address and apply to SRAM, ROM, and NOR flash operations in asynchronous bus multiplex mode. 0000: Reserved 0001: ADDHLD hold time = 2 HCLK clock cycles ...... 1111: ADDHLD hold time = 16 HCLK clock cycles (default value after reset). Note: In synchronous operations, this parameter does not apply, and the address hold time is alway*s 1 memory clock cycle.*
3:0	ADDBLD[3:0]	Address setup phase duration These bits define the setup time for the address and apply to SRAM, ROM, and NOR flash operations in asynchronous bus multiplex mode. 0000: ADDBLD setup time = 1 HCLK clock cycle ...... 1111: ADDBLD setup time = 16 HCLK clock cycles (default value after reset). For the address setup time for each memory type and access mode, refer to the corresponding chart. For example: Mode 2, read operation, ADDBLD=1: Address setup time = ADDBLD + 1 = 2 HCLK clock cycles *Note: In synchronous operations, this parameter does not apply, and the address setup time is always 1 memory clock cycle.*

Note: Due to internal refresh, PSRAM (CRAM) has a variable hold delay, so such memory will output an NWAIT signal during the data hold period to extend the data hold time.

When using PSRAM (CRAM), the DATAHLD field should be set to 0 so that the FEMC can promptly exit its hold phase and begin sampling the NWAIT signal issued by the memory, then start read or write operations when the memory is ready.

This operating mode can also be used for the latest synchronous flash memory that can output an NWAIT signal, for more information please refer to the corresponding flash memory manual.

31.7.2.3 SRAM/NOR Flash Write Timing Register（FEMC_ SNWTCFG1/2/3/4）

Address Offset: 0x008 + c * (x-1), (x=1,2,3,4)

Reset Value: 0x0FFFFFFF

This register contains control information for each memory block and can be used for SRAM, ROM, and NOR flash memory. If the EXTEN bit is set in the FEMC_SNCTRLx register, this register corresponds to write operations.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		ACCMODE[1:0]		Reserved								BUSRECOVERY[3:0]			
		rw										rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

DATABLD[7:0]	ADDHLD[3:0]	ADDBLD[3:0]
rw	rw	rw

Bit Field	**Name**	**Description**
31:30	Reserved	Reserved, the reset value must be maintained
29:28	ACCMODE[1:0]	Access mode Define asynchronous access mode. These 2 bits only take effect when the EXTEN bit of the FEMC_BANK1_CRx register is set to 1. 00: access mode A 01: access mode B 10: access mode C 11: access mode D
27:20	Reserved	Reserved, the reset value must be maintained
19:16	BUSRECOVERY[3:0]	Bus turnaround phase duration These bits are used to define the delay on the bus after a read operation (only applicable to NOR flash operations in bus multiplex mode). After a read operation, the controller needs to send out the address on the data bus for the next operation, and this delay is to prevent bus conflicts. If the extended memory system does not include memory in bus multiplex mode, or if the slowest memory can restore the data bus to a high-impedance state within 6 HCLK clock cycles, this parameter can be set to its minimum value. 0000: Bus recovery time = 1 HCLK clock cycle ...... 1111: Bus recovery time = 16 HCLK clock cycles (default value after reset).
15:8	DATABLD[7:0]	Data-phase duration These bits define the data hold time and apply to SRAM, ROM, and NOR flash operations in asynchronous bus multiplex mode. 0000 0000: Reserved 0000 0001: DATABLD hold time = 2 HCLK clock cycles 0000 0010: DATABLD hold time = 3 HCLK clock cycles ...... 1111 1111: DATABLD hold time = 256 HCLK clock cycles (default value after reset). For the data hold time for each memory type and access mode, please refer to the corresponding chart. For example: Mode 1, read operation, DATABLD=1: Data hold time = DATABLD + 3 = 4 HCLK clock cycles. *Note: This parameter does not apply in synchronous access.*
7:4	ADDHLD[3:0]	Address-hold phase duration

Bit Field	Name	Description
		These bits define the hold time for the address and apply to SRAM, ROM, and NOR flash operations in asynchronous bus multiplex mode. 0000: Reserved 0001: ADDHLD hold time = 2 HCLK clock cycles ...... 1111: ADDHLD hold time = 16 HCLK clock cycles (default value after reset). *Note: In synchronous operations, this parameter does not apply, and the address hold time is always 1 memory clock cycle.*
3:0	ADDBLD[3:0]	Address setup phase duration These bits define the setup time for the address and apply to SRAM, ROM, and NOR flash operations in asynchronous bus multiplex mode. 0000: ADDBLD setup time = 1 HCLK clock cycle ...... 1111: ADDBLD setup time = 16 HCLK clock cycles (default value after reset). For the address setup time for each type of memory and access mode, please refer to the corresponding chart. For example: Mode 2, read operation, ADDBLD=1: Address setup time = ADDBLD + 1 = 2 HCLK clock cycles *Note: In synchronous operations, this parameter does not apply, and the address setup time is always 1 memory clock cycle.*

31.7.3 NAND Flash Controller Register

These peripheral registers must be accessed in word (32-bit) mode.

31.7.4 NAND Flash Controller Register(FEMC_NCTRL2/3）

Address Offset: 0x60 + 0x20 * (x-2), (x=2,3)

Reset Value: 0x0000 0004

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															ECCPGS
rw	rw	rw													rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ECCPGS[1:0]		ECCEN	ARDLY[3:0]				CRDLY[3:0]				WAITEN	BUSWID[1:0]		Reserved	BANKEN
rw		rw	rw				rw				rw	rw		rw	rw

Bit Field	Name	Description
31:20	Reserved	Reserved, the reset value must be maintained
16:14	ECCPGS[2:0]	ECC Page Size

Bit Field	Name	Description
		Defines the extended ECC page size 000: 256 bytes 001: 512 bytes 010: 1024 bytes 011: 2048 bytes 100: 4096 bytes 101: 8192 bytes
13	ECCEN	ECC computation logic enable bit 0: Disable and reset the ECC circuit (default value after reset) 1: Enable the ECC circuit.
12:9	ARDLY[3:0]	The delay from ALE to RE Set the time from ALE falling to RE falling in units of AHB clock cycles (HCLK). Time calculation: t_ar = (ARDLY + SET + 4) x THCLK, where THCLK represents the duration of an HCLK cycle. 0000: 1 HCLK cycle (default value) ...... 1111: 16 HCLK cycles *Note: Depending on the address space, SET is either CMEMTM SET or ATTMEMTM SET.*
8:5	CRDLY[3:0]	The delay from CLE to RE Set the time from CLE falling to RE falling in units of AHB clock cycles (HCLK). Time calculation: t_clr = (CRDLY + SET + 4) x THCLK, where THCLK represents the duration of an HCLK cycle. 0000: 1 HCLK cycle (default value) ...... 1111: 16 HCLK cycles *Note: Depending on the address space, SET is either CMEMTM SET or ATTMEMTM SET.*
4	WAITEN	Wait feature enable bit Enable the wait function of the NAND flash memory block 0: Disable the wait function (default value after reset) 1: Enable the wait function
3:2	BUSWID[1:0]	Databus width Define the width of the external NAND flash data bus. 00: 8-bit 01: 16-bit(default value after reset) 10: Reserved

Bit Field	Name	Description
		11: Reserved
1	Reserved	Reserved, the reset value must be maintained
0	BANKEN	NAND Flash memory bank enable bit Enable the memory block. Accessing a disabled memory block will result in an AHB bus error. 0: Disable the corresponding memory block (default value after reset) 1: Enable the corresponding memory block;

31.7.4.1 FIFO Status and Interrupt Registers(FEMC_NSTS2/3）

Address Offset: 0x64 + 0x20 * (x-2), (x=2,3)

Reset Value: 0x0000 0040

FEMC does not support interrupt functionality.

For ECC computation purposes, this register has an indicator bit reflecting the status of the FIFO. ECC calculation is performed simultaneously when data is written to memory, so the software must wait for the FIFO to empty before reading the correct ECC value.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									FIFO	IHLIEN	IREEN	IFEEN	IHLF	IREF	IFEF
									r	rw	rw	rw	rc_w0	rc_w0	rc_w0

Bit Field	Name	Description
31:7	Reserved	Reserved, the reset value must be maintained
6	FIFOEMPT	FIFO empty flag Read-only bit, indicating FIFO status 0: FIFO is not empty 1: FIFO is empty
5	IHLEN	Interrupt high-level detection enable 0: Disable interrupt high-level detection 1: Enable interrupt high-level detection
4	IREEN	Interrupt rising edge detection enable 0: Disable interrupt rising edge detection 1: Enable interrupt rising edge detection
3	IFEEN	Interrupt falling edge detection enable

		0: Disable interrupt falling edge detection 1: Enable interrupt falling edge detection
2	IHLF	Interrupt high-level status flag 0: No interrupt high-level status occured 1: Interrupt high-level status occured
1	IREF	Interrupt rising edge status flag 0: No interrupt rising edge occurred 1: Interrupt rising edge occurred
0	IFEF	Interrupt falling edge status flag 0: No interrupt falling edge occurred 1: Interrupt falling edge occurred

31.7.4.2 Common Memory Space Timing Register 2/3(FEMC_NCMEMTM2/3)

Address Offset: 0x68 + 0x20 * (x-2), (x=2,3)

Reset Value: 0xFCFC FCFC

Each NCMEMTMx (x=2,3) register contains timing parameters for operating NAND flash memory block x, These parameters are used for sending commands, addresses, and performing data read/write operations to NAND flash in the common memory space.

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">WAIT[7:0]</td><td colspan="8">SET[7:0]</td></tr>
<tr><td colspan="8">rw</td><td colspan="8">rw</td></tr>
</table>

<table>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="8">HIZ[7:0]</td><td colspan="8">HLD[7:0]</td></tr>
<tr><td colspan="8">rw</td><td colspan="8">rw</td></tr>
</table>

Bit Field	**Name**	**Description**
31:24	WAIT[7:0]	Common memory x wait time When performing read or write operations on NAND flash in common memory space x, this parameter defines the minimum time to hold the command (NWE, NOE low) in terms of HCLK (+1) clock cycles. When the time defined by this parameter ends, if the wait signal (NWAIT) is active (low), the command hold time will be extended. 0000 0000: Reserved 0000 0001: 2 HCLK cycles (plus the wait cycles introduced by the NWAIT signal going low) ...... 1111 1110: 255 HCLK cycles (plus the wait cycles introduced by the NWAIT signal going low), which is the default value after reset. 1111 1111: Reserved

Bit Field	Name	Description
23:16	SET[7:0]	Common memory x setup time When performing read or write operations on NAND flash in common memory space x, this parameter defines the time to setup the address signal before sending the command (NWE, NOE is low) in terms of HCLK (NAND flash operation +2) clock cycles. 0000 0000: NAND flash is 2 HCLK cycles ...... 1111 1110: NAND flash is 256 HCLK cycles, which is the default value after reset. 1111 1111: Reserved
15:8	HIZ[7:0]	Common memory x databus Hiz time When starting a write operation to NAND flash in common memory space x, the data bus needs to maintain a high-impedance state for a certain period of time. This parameter defines the duration of the data bus high-impedance state in terms of HCLK clock cycles (NAND type +1). This parameter is only applicable to write operations. 0000 0000: 1 HCLK cycle ...... 1111 1110: 255 HCLK cycles, which is the default value after reset. 1111 1111: Reserved
7:0	HLD[7:0]	Common memory x hold time When performing read or write operations on NAND flash in common memory space x, this parameter defines the time for which the address signal (or data signal for write operations) is held after sending the command (NWE, NOE high) in terms of HCLK clock cycles. 0000 0000: Reserved 0000 0001: 2 HCLK cycles ...... 1111 1110: 255 HCLK cycles (default value after reset). 1111 1111: Reserved

31.7.4.3 Attribute Memory Space Timing Register 2/3(FEMC_NATTMEMTM2/3)

Address Offset: 0x6C + 0x20 * (x-2), (x=2,3)

Reset Value: 0xFCFC FCFC

Each NATTMEMTMx (x=2,3) read/write register contains timing parameters for operating NAND flash memory block x. These parameters are used when the timing for the last address write operation to NAND flash differs from other operations (for managing ready/busy status, refer to section 31.6.5).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
WAIT[7:0]								SET[7:0]							

rw								rw							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HIZ[7:0]								HLD[7:0]							
rw								rw							

Bit Field	Name	Description
31:24	WAIT[7:0]	Common memory x wait time When performing read or write operations on NAND flash in common memory space x, this parameter defines the minimum time to hold the command (NWE, NOE low) in terms of HCLK (+2) clock cycles. When the time defined by this parameter ends, if the wait signal (NWAIT) is active (low), the command hold time will be extended. 0000 0000: Reserved 0000 0001: 2 HCLK cycles (plus the wait cycles introduced by the NWAIT signal going low) ...... 1111 1110: 255 HCLK cycles (plus the wait cycles introduced by the NWAIT signal going low), which is the default value after reset. 1111 1111: Reserved
23:16	SET[7:0]	Common memory x setup time When performing read or write operations on NAND flash in common memory space x, this parameter defines the time to setup the address signal before sending the command (NWE, NOE u\is low) in terms of HCLK (NAND flash operation +2) clock cycles. 0000 0000: 2 HCLK cycles ...... 1111 1110: 256 HCLK cycles, which is the default value after reset. 1111 1111: Reserved
15:8	HIZ[7:0]	Common memory x databus Hiz time When initiating a write operation to NAND flash in common memory space x, the data bus needs to maintain a high-impedance state for a certain duration of time. This parameter defines the duration of the data bus high-impedance state in terms of HCLK clock cycles (NAND type +1). This parameter is only applicable to write operations. 0000 0000: 1 HCLK cycle ...... 1111 1110: 255 HCLK cycles, which is the default value after reset. 1111 1111: Reserved
7:0	HLD[7:0]	Common memory x hold time When performing read or write operations on NAND flash in common memory space x, this parameter defines the time for which the address signal (or data signal for write operations) is held after sending the command (NWE, NOE is high) in terms of HCLK clock cycles.

Bit Field	Name	Description
		0000 0000: Reserved 0000 0001: 2 HCLK cycles …… 1111 1110: 255 HCLK cycles (default value after reset). 1111 1111: Reserved

31.7.4.4 ECC Result Registers2/3（FEMC_ECC2/3）

Address Offset: 0x74 + 0x20 * (x-2), (x=2,3)

Reset Value: 0x0000 0000

These two registers contain the current values of the error correction codes obtained by the ECC calculation module of the FEMC controller, with each NAND flash memory block having an ECC calculation module. When the CPU reads/writes data from/to NAND flash at the correct address (refer to 31.6.6), the ECC module automatically handles the written or read data. Depending on the setting of the ECCPGS field in FEMC_CTRLx, after reading the last byte of each page, the CPU must read the ECC value from the FEMC_ECCx register and compare it with the data recorded in the NAND flash spare area to determine if the page data is correct and, if possible, perform correction. After reading the value from the FEMC_ECCx register, the ECCEN bit should be set to '0' to clear its content. When calculating a new data page, set ECCEN to '1' again.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ECC[31:16]															
				rw								rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ECC[15:0]															
				rw								rw			

Bit Field	Name	Description
31:0	ECCx[31:0]	ECC Result The calculation result generated by the ECC calculation circuit. The table below shows the contents of these bits.

Table 31-36 ECC Result Valid Bit

ECCPGS[2:0]	Page Size（byte）	Valid Bit of ECC
000	256	ECC[21:0]
001	512	ECC[23:0]
010	1024	ECC[25:0]
011	2048	ECC[27:0]
100	4096	ECC[29:0]
101	8192	ECC[31:0]

32 SDIO Interface（SDIO）

32.1 Function

SDIO interface defines SD card, SD I/O card, Multimedia Card (MMC) host interface. It provides data transfer between AHB peripheral bus, SD memory card, SDIO card, and Multimedia Card (MMC) devices. Among them, the supported multimedia card system specifications are published by the MMCA technical committee and can be obtained on the website of the Multimedia Card Association (www.mmca.org), and supported SD memory cards and SD I/O card system specifications are available through the SD Card Association website (www.sdcard.org).

Main Features

- SD card: fully compatible with SD memory card specification version 2.0.
- SD I/O: Fully compatible with SD I/O card specification version 2.0, there are two different data bus modes: 1-bit (default) and 4-bit.
- MMC: Fully compatible with Multimedia Card System Specification Version 4.2 and earlier versions. There are three different data bus modes: 1-bit (default), 4-bit and 8-bit.
- Up to 50MHz data transfer rate in 8-bit data bus mode.
- Support interrupt and DMA request.
- Supports data and command output enable signals for controlling external bidirectional drivers.

Notes:

1) *SDIO has no SPI compatible communication mode.*

2) *In version 2.11 of the Multimedia Card System Specification, it is defined that the SD memory card protocol only supports the I/O part of the SD card or composite card in I/O mode, and does not support many required commands in the SD storage device, such as erasing etc. some commands don't work in SDI/O devices, so SDIO doesn't support these commands either. In addition, some commands are different in SD memory card and SD I/O card, and SDIO does not support these commands.*

SDIO supports only one SD/SDIO/MMC 4.2 card at a time, but can support multiple MMC version 4.1 or earlier cards.

32.2 SDIO Bus Topology

Communication on the SDIO bus is achieved by transferring commands and data. After power-on reset, the host must initialize the card through a special message-based bus protocol. Each message is a command/response structure, additionally, some messages have data tokens. Each part of the message is described in detail as follows:

- Command: The command is serially transmitted on the CMD line and is a token to initiate an operation, sent from the host to the card
- Response: A response is serially transmitted on the CMD line, sent from the card to the host in response to a previously received command.
- Data: Data is transmitted through the data line. Data can be transferred from the card to the host or from the host to the card. The number of data lines used for data transfer can be 1 (DAT0), 4 (DAT0-DAT3) or 8 (DAT0-DAT7).

The structure of commands, responses and data blocks is described in the card functional description chapter. A data transfer is a bus operation. A normal operation always consists of a command and response. Additionally, some operations have a data token. There are other operations that include their information directly in the command or response structure. In this case, the operation has no data token.

There are two types of data transfer commands: block and stream. Data transmitted on SD/SDIO and memory cards is transmitted in the form of data blocks; data transmitted on MMC is transmitted in the form of data blocks or data streams;

Data streams and data block transfers are defined as follows:

- Data flow: The command initiates a continuous data flow, and the data transmission is terminated only when a stop command appears on the CMD signal line. This mode minimizes command overhead (only MMC is supported).
- Data block: The command successfully sends a data block followed by a CRC check. Read and write operations allow single or multiple block transfers. As with continuous read, a multi-block transfer is terminated when a stop command appears on the CMD signal line.

The basic operations on the SDIO bus are command/response operations, and this type of bus transaction passes their information directly in a command or response structure. In addition, some operations have data tokens. Data transfer between the card and the device is done through blocks. Each transmission type is shown in the following figure:

Figure 32-1 SDIO "No Response" And "No Data" Operations

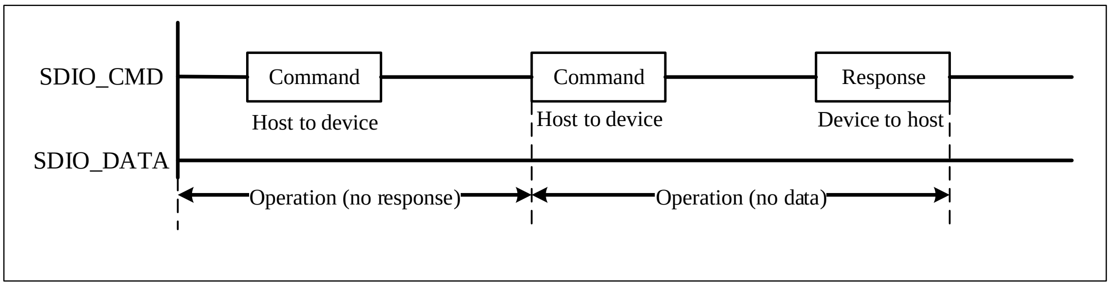


Figure 32-2 SDIO (Multi) Data Block Read Operation

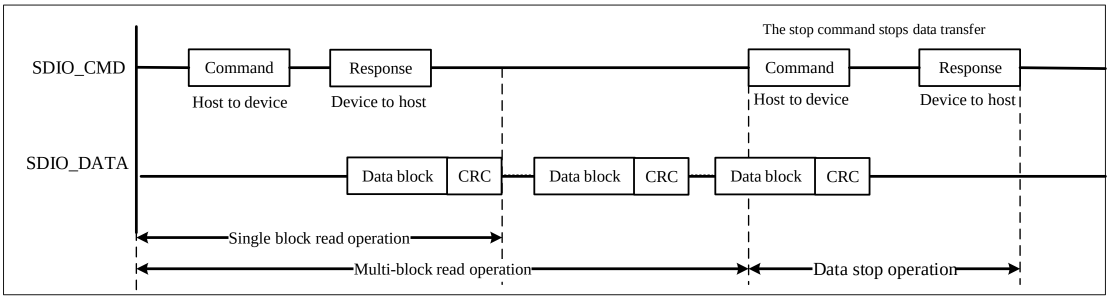
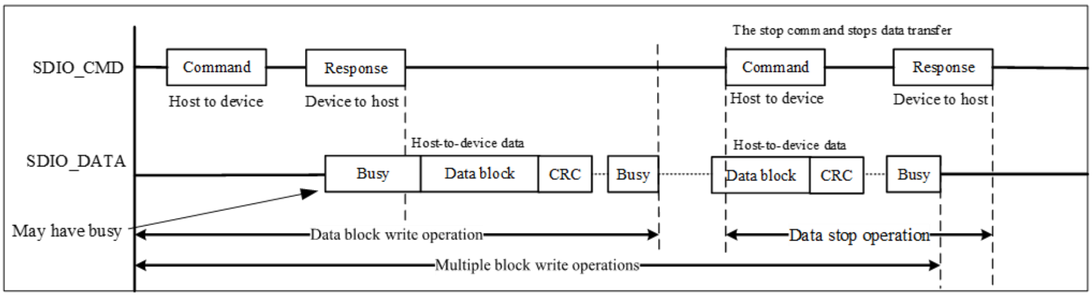


Figure 32-3 SDIO (Multi) Data Block Write Operation



Note: When there is a Busy signal, SDIO (SDIO_DAT0 is pulled low) will not send any data.

Figure 32-4 SDIO Continuous Read Operation

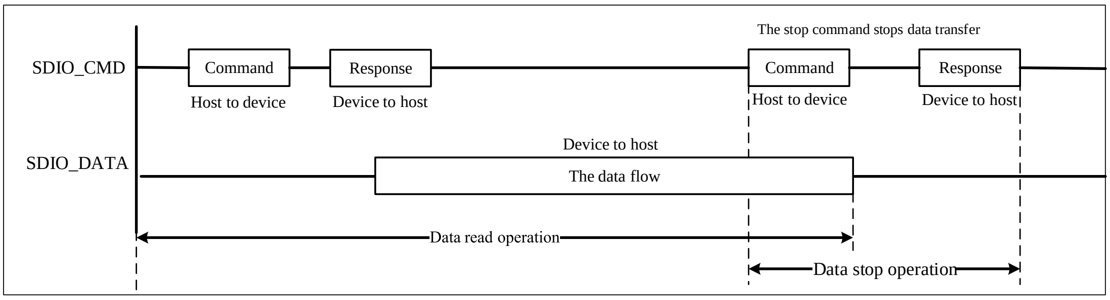
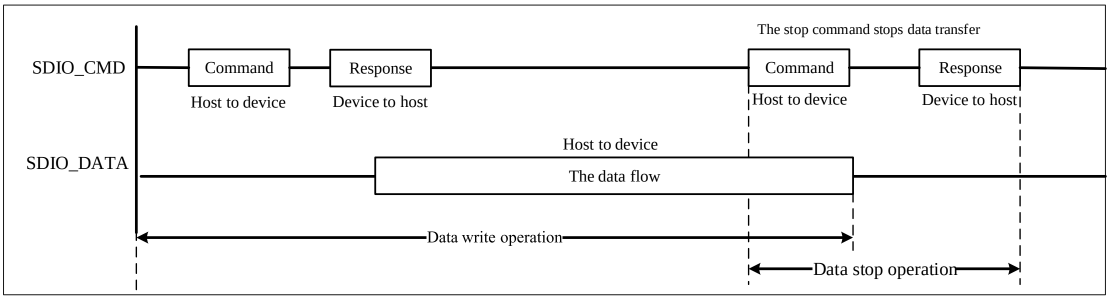


Figure 32-5 SDIO Continuous Write Operation

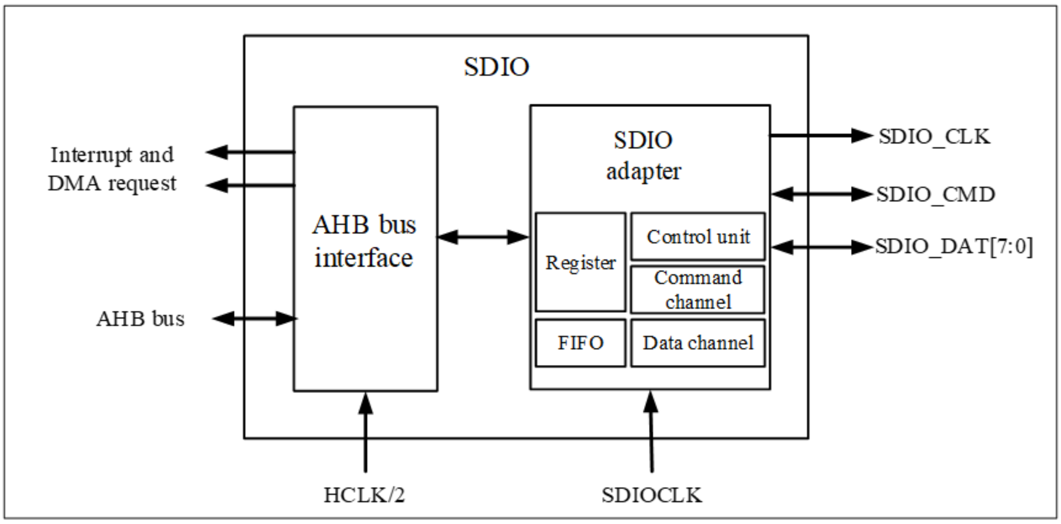


32.3 SDIO Function Description

Figure 32-6 is a block diagram of SDIO structure:

- SDIO adapter: It consists of control unit, command unit and data unit. The control unit generates a clock signal, and the command unit and data unit manage the transmission of commands and data respectively, thereby realizing the related functions of the MMC/SD/SD I/O card. .
- AHB bus interface: used to operate the registers in the SDIO adapter module, control the FIFO units for data transfer, and generate interrupt and DMA request signals.

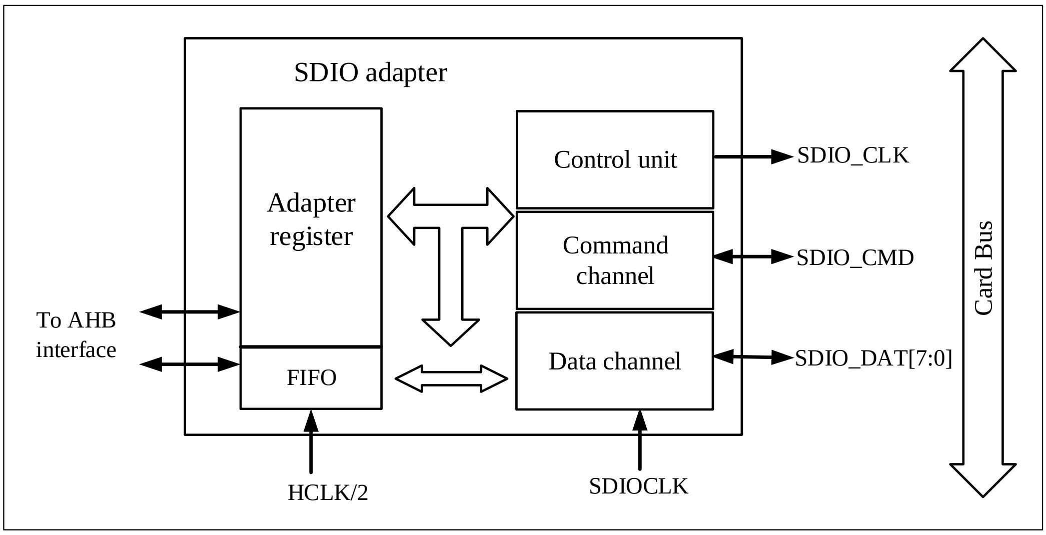
Figure 32-6 SDIO Block Diagram



32.3.1 SDIO Adapter

The following figure is a simplified block diagram of the SDIO adapter:

Figure 32-7 SDIO Adapter



The SDIO adapter is the host device of the Multimedia/ secure digital storage card bus, used to connect a set of multimedia cards or secure digital storage cards, utilizing two clock signals: the SDIO adapter clock (SDIOCLK = HCLK or HCLK/2) and the AHB bus clock (HCLK/2).

The SDIO adapter consists of five parts: control unit, adapter register module, command unit, data unit, and data FIFO. Please note that the adapter registers and FIFO use the AHB bus clock (HCLK/2), while the control unit, command channel, and data channel use the SDIO adapter clock (SDIOCLK).The signals output to the card bus are as follows:

Table 32-1 MMC/SD/SD I/O Card Bus Pin Definition

Pin	Direction	Description
SDIO_CLK	Output	MMC/SD/SDIO card clock, clock line from host to card
SDIO_CMD	Bidirectional	MMC/SD/SDIO card command, bidirectional command/response signal line
SDIO_DAT[7:0]	Bidirectional	MMC/SD/SDIO card data, bidirectional data bus

- SDIO_DAT[7:0]: The data signal line uses push-pull mode. By default, only SDIO_DAT0 is used for data transfer after power-on or reset. The SDIO adapter can configure a wider data bus for data transfer after initializing the host, using SDIO_DAT[3:0] or SDIO_DAT[7:0] (only for MMC V4.2). Note that the protocol of MMC version V3.31 and previous versions only supports 1-bit data line (only SDIO_DAT0 can be used). When an SD or SD I/O card is connected to the bus, SDIO_DAT0 or SDIO_DAT[3:0] can be used by the host to configure data transfers.
- SDIO_CMD: command/response signal lines, two operating modes:
  - Open-drain mode for initialization (only for MMC version V3.31 or earlier)
  - Push-pull mode for command transfer (SD/SD I/O cards and MMC V4.2 also use push-pull drive during initialization)
- SDIO_CLK: The clock provided to the card by the SDIO controller. One bit of command or data is sent directly on the command line (SDIO_CMD) and all data lines per clock cycle. The variation range of the clock frequency

of different cards is different, as follows:

- MMC V3.31 protocol, optional clock frequency between 0MHz and 20MHz;
- MMC V4.0/4.2 protocol, optional clock frequency between 0MHz and 48MHz;
- SD or SD I/O card, optional clock frequency between 0MHz and 25MHz.

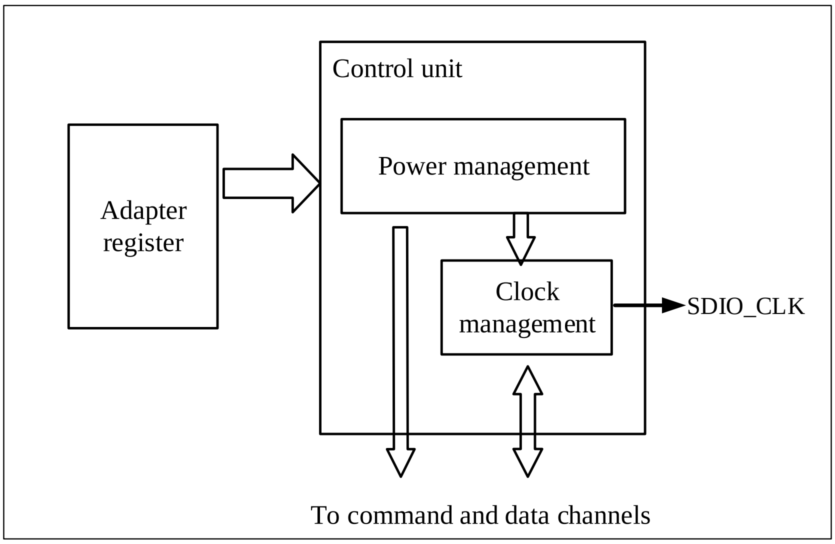
32.3.1.1 Adapter register block

The adapter register block contains all system registers. The adapter register module contains all system registers. This module also generates a signal to clear the static flags in the multimedia card, which is triggered when writing a '1' to the corresponding bit in the SDIO clear register.

32.3.1.2 Control unit

The control unit contains power management functions and clock management functions， which are userd for the memory card identification, initialization, clock control etc. The functional diagram is as follows:

Figure 32-8 Control Unit



Power management is controlled by the SDIO_PWRCTRL register to achieve power down and power up. There are three power phases: power off, power up, and power on.

The clock management subunit generates and controls the SDIO_CLK clock signal, which is then output to the card bus. The SDIO_CLK output supports clock division and clock bypass modes. When the CLKBYP bit in the SDIO_CLKCTRL register is 0, SDIO_CLK is derived from the SDIOCLK through division; when the CLKBYP bit in the SDIO_CLKCTRL register is 1, SDIO_CLK is directly sourced from SDIOCLK.

By configuring the PWRCFG bit in the SDIO_CLKCTRL register, power-saving mode can be enabled to switch off SDIO_CLK when the bus is idle.

Hardware clock control is enabled by setting the HWCLKEN bit in the SDIO_CLKCTRL register. This function is used to avoid FIFO underflow and overflow errors. The hardware controls the switch of SDIO_CLK according to whether the system bus is busy. When the FIFO cannot receive or transmit data, the host will switch off SDIO_CLK and freeze the SDIO state machine to avoid related errors. Only the state machine can be frozen, but the AHB interface is still working. Therefore, the FIFO can be accessed through the AHB bus.

SDIO_CLK clock is not output in the following three cases:

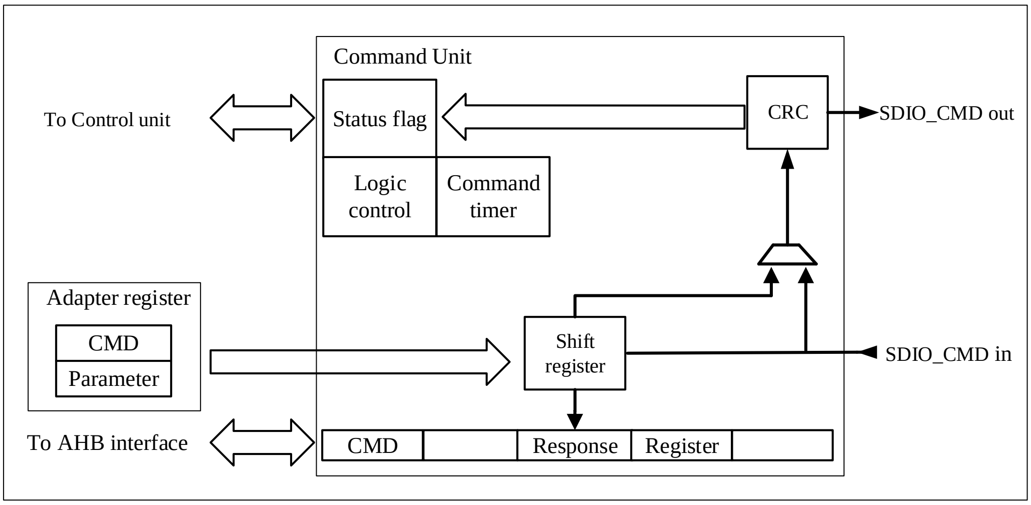
- After reset
- During power-off and power-on stages
- When power-saving mode is enabled and the card bus is idle (Within 8 clock cycles after the command channel and data channel subunits enter the idle state.)

32.3.1.3 Command unit

The command channel unit transmits commands to the card and receives responses from the card.

The data transfer flow of the command channel is controlled by the command state machine (CPSM). The command transfer begins after setting SDIO_CMDCTRL.CPSMEN bit to 1 and writing it once. First send a 48-bit command to the card through the SDIO_CMD line, one bit of data per SDIO_CLK. The 48-bit command contains 1 start bit, 1 transmit bit, 6-bit command index (defined by SDIO_CMDCTRL.CMDIDX), 32-bit parameters (defined by SDIO_CMDARG), 7-bit CRC, and 1-bit stop bit. Then receive the response from the card when SDIO_CMDCTRL.CMDIDX bits is not 0b00 or 0b10. The response is divided into a 48-bit short response and a 136-bit long response, and the responses are stored in the SDIO_RESPONSE1-SDIO_RESPONSE 4 registers. The command unit can also generate command status flags, which are defined in the SDIO_STS register.

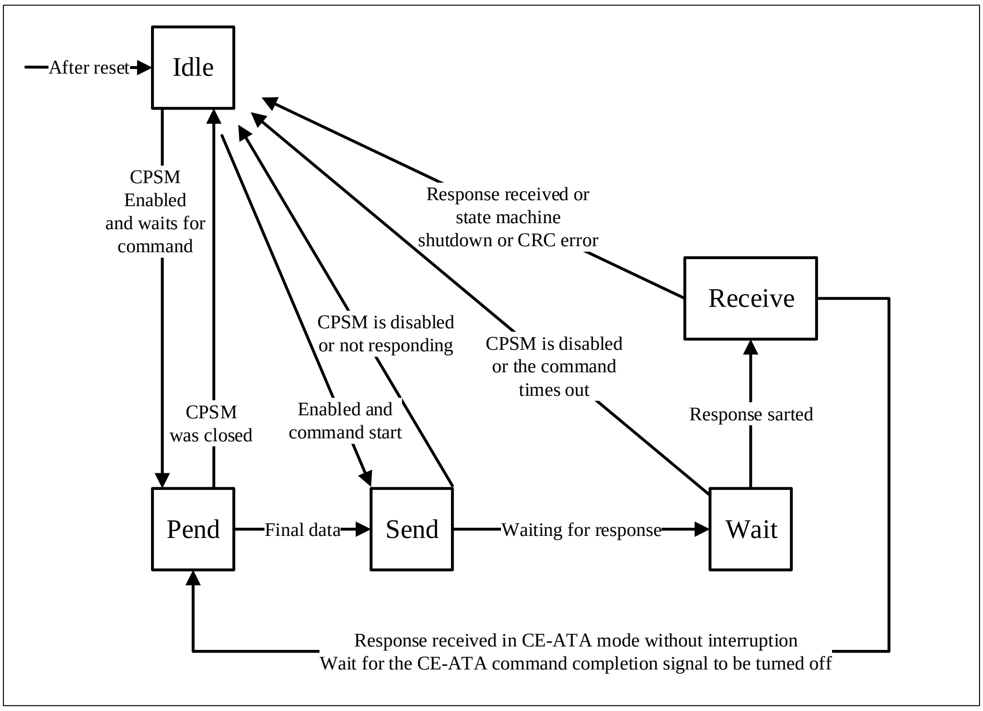
Figure 32-9 SDIO Adapter Command Unit



32.3.1.4 Command Path State Machine（CPSM）

When writing to the command register and setting the enable bit, the command transmission begins. Upon completion of the command transmission, the Command Path State Machine (CPSM) sets status flags and enters the idle state when no response is required (see the figure below). Upon receiving a response, the received CRC code is compared with the internally generated CRC code, and then the relevant status flags are set.

Figure 32-10 Command Path State Machine (CPSM)


After reset
Idle
CPSM Enabled and waits for command
Response received or state machine shutdown or CRC error
Receive
CPSM is disabled or not responding
CPSM is disabled or the command times out
CPSM was closed
Enabled and command start
Response sarted
Pend
Final data
Send
Waiting for response
Wait
Response received in CE-ATA mode without interruption
Wait for the CE-ATA command completion signal to be turned off

- Idle

This state is the idle state, which is entered after system reset in preparation for sending commands or when the command path state machine (CPSM) is disabled. When the command path state machine (CPSM) is enabled, enabling or disabling the waiting for data transfer to end (WDATEND) can transition to the Pending state.

Note: The command state machine remains idle for at least 8 SDIO_CLK cycles to meet NCC and NRC timing constraints. NCC is the minimum time interval between two host commands, and NRC is the minimum time interval between a host command and a card response.

- Pend

This state is a pending state, waiting for the end of data transfer. When the data transmission is completed, the command state machine enters the Send state from the Pend state; when the command path state machine (CPSM) is disabled, the CPSM enters the Idle state.

- Send

This state is the sending state, indicating that the command is being sent. If there is a response after the command is sent, the command state machine enters the Wait state, and if there is no response after the command is sent, the command state machine enters the Idle state.

- Wait

This state is a wait state, waiting for the response start bit. When entering the wait (Wait) state, the command timer starts to run; if a response is received, that is, the start bit is detected, the command path state machine (CPSM) enters the Receive state, If a timeout occurs before the Command Path State Machine (CPSM) enters the Receive state, the timeout flag is set, and the system transitions to the Idle state. *Note: The command timeout period is fixed at 64 SDIO_CLK clock cycles.*

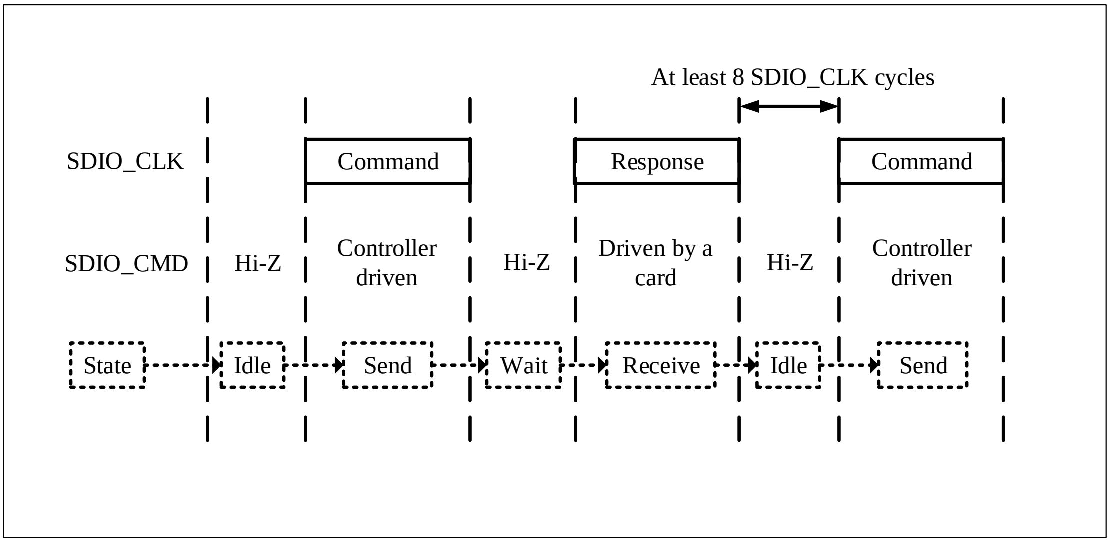
- Receive

This state is the Receive state, where the response is received and the CRC is checked. The CPSM will transition to the Idle state if it is disabled, a response is received, or if the command CRC check fails.

- Interrupt

If the interrupt bit is set in the command register, the timer is turned off, and the CPSM waits for an interrupt request from a card. If the suspend bit is set in the command register, the CPSM enters the Pending state and waits for the CmdPend signal from the data channel subunit. Upon detecting the CmdPend signal, the CPSM enters the Send state, triggering the data counter to send a stop command.

Figure 32-11 SDIO Command Transmission



- Command register

The 6-bit command index and command type sent to the card are stored in the command register; the command type determines whether and what type of response (48-bit or 136-bit). The status flags in the command channel are shown in the table below.

Table 32-2 Command Channel Status Flags

Flag	Description
SDIO_STS.CMDRESPRECV	CRC correct response
SDIO_STS.CCRCERR	CRC error response
SDIO_STS.CMDSEND	Command (command that does not require a response) has been sent
SDIO_STS.CMDTIMEOUT	Response timeout
SDIO_STS.CMDRUN	Command transfer in progress

- CRC generator

The CRC generator calculates the CRC checksum of all bits preceding the CRC code, including start bits, transmit bits, command index, and command parameters (or card status). For the long response format, the CRC checksum is calculated from the first 120 bits of the CID or CSD. Note that the start bit, transmission bits and 6 reserved bits in the long response format do not participate in the CRC calculation.

The CRC checksum is a 7-bit value: CRC[6:0] = remainder[(M(x) * x^7) / G(x)] where:

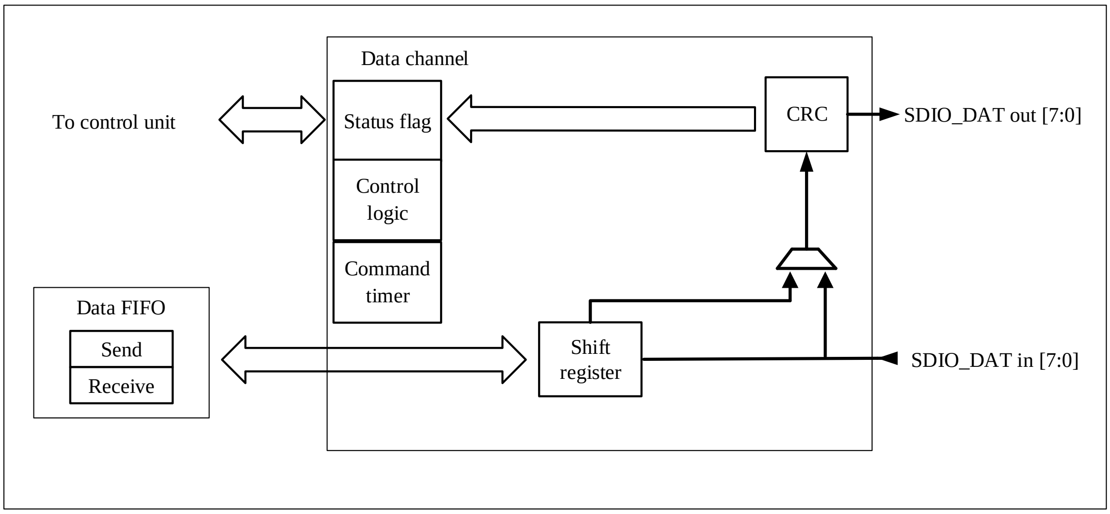
G(x) = $x^7 + x^3 + 1$

M(x) =（start bit）* x^{39} +…+ (last bit before CRC)* x^0, or M(x) =（start bit）* x^{119} +…+(last bit before CRC)* x^0

32.3.1.5 Data channel

The data between the host and the card is transmitted through the data channel. The diagram of the data channel is shown below.

Figure 32-12 Data Channel



The data bus width of the card can be configured in the clock control register (SDIO_CLKCTRL). When the data width is 4 bits (SDIO_ CLKCTRL.BUSMODE bit is 0b01), 4 bits of data will be transmitted on the four data signal lines SDIO_DAT[3:0] in each clock cycle.When the data width is 8 bits (SDIO_ CLKCTRL.BUSMODE bit is 0b10), 8-bit data will be transmitted on the eight data signal lines SDIO_DAT[7:0] per clock cycle.When the data width is 1 bit (SDIO_CLKCTRL.BUSMODE bit is 0b00) or the bus mode is not selected , only 1 bit of data is transmitted on SDIO_DAT0 per clock cycle.

The data transmission flow is controlled by the Data Channel State Machine (DPSM). Data transfer begins after writing to the SDIO_DATCTRL register and setting the SDIO_DATCTRL.DATEN bit to 1. When the SDIO_DATCTRL.DATDIR bit is 0, the data is from the controller to the card, when the SDIO_DATCTRL.DATDIR bit is 1, the data is from the card to the controller. Data units can also generate data status flags (defined in the SDIO_STS register).

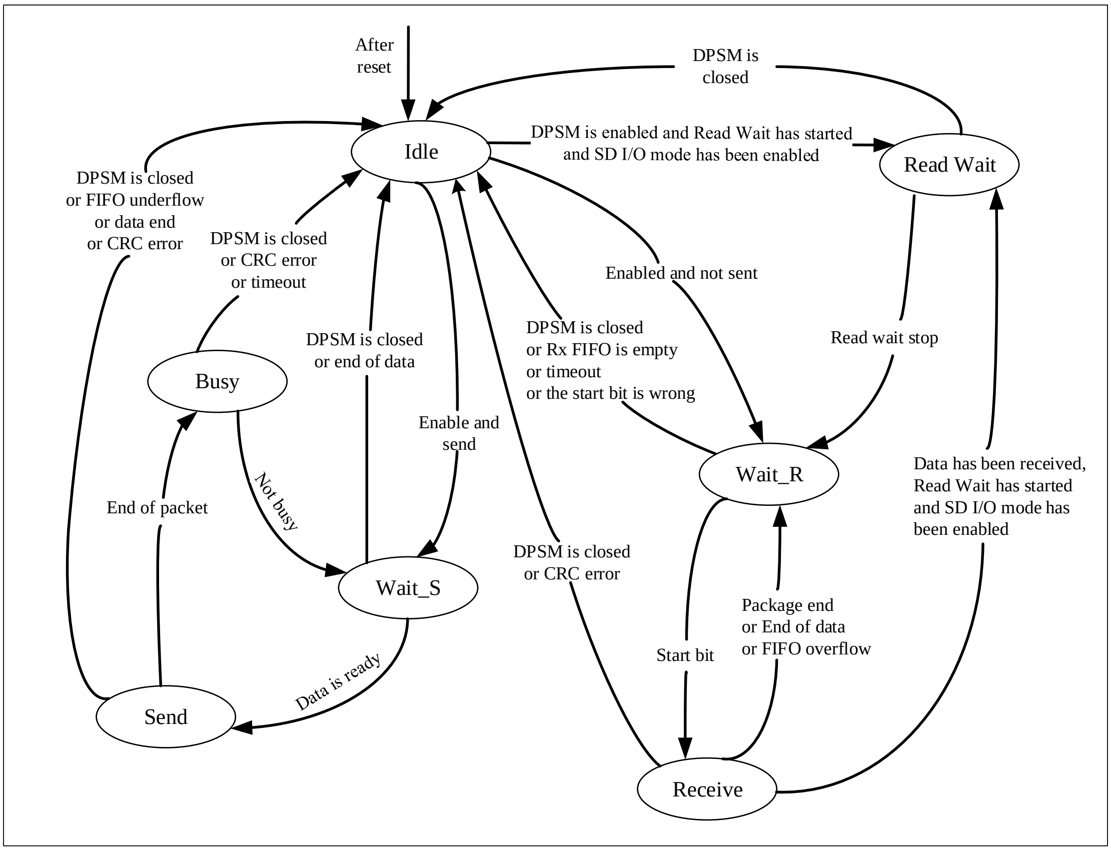
Depending on the direction of the transfer (transmit or receive), the Data Path State Machine (DPSM) will enter the Wait_S or Wait_R state:

- Transmit: The DPSM enters the Wait_S state. If there is data in the transmit FIFO, the DPSM enters the transmit state, and the data channel subunit begins sending data to the card.
- Receive: The DPSM enters the Wait_R state and waits for the start bit; upon receiving the start bit, the DPSM enters the receive state, and the data channel subunit starts receiving data from the card.

32.3.1.6 Data Path State Machine（DPSM）

The data path state machine (DPSM) operates at the SDIO_CLK frequency, and the card bus signal is synchronized with the rising edge of SDIO_CLK. DPSM has 6 states, as shown in the following figure:

Figure 32-13 Data Path State Machine (DPSM)


After reset
DPSM is closed
Idle
DPSM is enabled and Read Wait has started and SD I/O mode has been enabled
Read Wait
DPSM is closed or FIFO underflow or data end or CRC error
DPSM is closed or CRC error or timeout
Enabled and not sent
DPSM is closed or Rx FIFO is empty or timeout or the start bit is wrong
Read wait stop
Busy
DPSM is closed or end of data
Enable and send
Wait_R
Data has been received, Read Wait has started and SD I/O mode has been enabled
Not busy
End of packet
Wait_S
DPSM is closed or CRC error
Package end or End of data or FIFO overflow
Start bit
Data is ready
Send
Receive

- Idle:

  In this state, the data channel does not work, waiting to transmit and receive data, and the SDIO_DAT[7:0] output is in a high-impedance state. When the data control register is written and the enable bit is set, the DPSM loads a new value for the data counter, enters the Wait_S state when the data transfer direction is from the host to the card, and enters the Wait_R state when the data transfer direction is from the card to the host. The Read Wait state is entered when DPSM is enabled and a read wait has started and SD I/O mode is enabled.

- Wait for Reception (Wait_R):

  In this state, the DPSM waits for the start bit of the received data. If the data times out (counter equals 0, when the receive FIFO is empty) the DPSM enters the Idle state. If the data counter is not equal to 0, the DPSM waits for a start bit on SDIO_DAT; if the DPSM receives a start bit before the timeout, it enters the Receive state and loads the data block counter. If the DPSM times out before detecting a start bit, or if a start bit error occurs, the DPSM will go into the idle state and set the time-out status flag.

- Receive:

  In this state the DPSM receives the card's data and writes it to the data FIFO. Depending on the setting of the transfer mode bits in the data control register, the data transfer mode can be block transfer or stream transfer.

- In block mode, when the data block counter reaches 0, the DPSM waits to receive the CRC code, if the received code matches the internally generated CRC code, the DPSM enters the Wait_R state, otherwise the CRC failure state flag is set and the DPSM enters the idle state state.
- In streaming mode, when the data counter is not 0, the DPSM receives data; when the counter is 0, the remaining data in the shift register is written into the data FIFO, and the DPSM enters the Wait_R state. If a FIFO overflow error occurs, the DPSM sets the FIFO error flag and enters the idle state.

- Wait for Send (Wait_S):

  In this state, the DPSM waits for the data FIFO empty flag to be invalid or the end of the data transfer. If the data counter is 0, the DPSM enters the idle state; otherwise, the DPSM waits for the data FIFO empty flag to disappear before entering the sending state.

  Note: DPSM will remain in Wait_S state for at least 2 clock cycles to meet the timing requirements of NWR. NWR is the interval from receiving the response from the card to when the host starts data transmission.

- Send:

  In this state, the DPSM starts sending data to the card device. Depending on the setting of the transfer mode bits in the data control register, the data transfer mode can be block transfer or stream transfer:

  - In block mode, when the data block counter reaches 0, the DPSM sends the internally generated CRC code, followed by the end bit, and enters the busy state.
  - In streaming mode, when the enable bit is high and the data counter is not 0, the DPSM sends data to the card device, and then enters the idle state.
  - If a FIFO underflow error occurs, the DPSM sets the FIFO error flag and enters the idle state.

- Busy:

  In this state, the DPSM waits for the CRC status flag:

  - If the correct CRC status is not received, the DPSM enters the idle state and sets the CRC failure status flag.
  - If the correct CRC status is received, and the card is not busy (SDIO_DAT0 is not low), the DPSM enters the Wait_S state.
  - If the correct CRC status is not received, the DPSM enters the idle state and sets the CRC failure status flag.
  - If the data timeout DPSM sets the data timeout flag and enters the idle state.
  - When the DPSM is in Wait_R or busy state, the data timer is enabled and can generate a data timeout error:
  - When sending data, if the DPSM is in a busy state and exceeds the timeout interval set by the program, a timeout will occur.
  - When receiving data, if all data is not received and DPSM is in Wait_R state for more than the timeout interval set by the program, a timeout will occur

32.3.1.7 Data Transfer

Data can be transmitted from the host to the card, as well as in the reverse direction.

The data is transmitted over the data line. The data is stored in a FIFO with a depth of 32 words, each word being 32 bits wide.

Table 32-3 Data Token Format

Description	Start bit	Data	CRC16	End bit
Block Data	0	-	Yes	1
Stream Data	0	-	No	1

32.3.1.8 Data FIFO

The data FIFO unit has a data buffer for transmitting and receiving data. The FIFO contains a data buffer and transmit and receive circuits, where the buffer size is 32 bits wide per word, 32 words in total (the depth is 32 words.). The data FIFO operates in the AHB clock region (HCLK/2), and all signals connected to the SDIO clock region (SDIOCLK) are resynchronized.

Depending on the SDIO_STS.TXRUN and SDIO_STS.RXRUN flags, the FIFO can be disabled, the transmission can be enabled, or the reception can be enabled. SDIO_STS.TXRUN and SDIO_STS.RXRUN are set by the data channel subunit and are mutually exclusive:

- When SDIO_STS.TXRUN is valid, the transmit FIFO represents the transmit circuit and data buffer
- When SDIO_STS.RXRUN is valid, the receive FIFO represents the receive circuit and data buffer

Transmit FIFO: When the SDIO transmit function is enabled, data can be written to the transmit FIFO through the AHB interface. The transmit FIFO has 32 consecutive addresses. There is a data output register in the transmit FIFO that contains the data word pointed to by the read pointer. When the data channel subunit fills the shift register, it moves the read pointer to the next data and transfers the data out. If the transmit FIFO is not enabled, all status flags are invalid. When sending data, the data channel subunit sets SDIO_STS.TXRUN to be valid.

Table 32-4 Transmit FIFO Status Flags

Flag	Description
SDIO_STS.TFIFOF	This flag is set high when all 32 transmit FIFO words have valid data.
SDIO_STS.TFIFOE	This flag is set high when all 32 transmit FIFO words have no valid data.
SDIO_STS.TFIFOHE	This flag is set high when 8 or more transmit FIFO words are empty. This flag can be used as a DMA request.
SDIO_STS.TDATVALID	This flag is set high when the transmit FIFO contains valid data. The meaning of this flag is just the opposite of TFIFOE.
SDIO_STS.TXURERR	This flag is set high when an underflow error occurs. This flag is cleared when writing to the SDIO clear register.

Receive FIFO: When the data channel subunit receives a data word, it will write the data into the FIFO. After the write operation is completed, the write pointer will automatically increase by one; at the other end, there is a read pointer that always points to the current data in the FIFO. If the receive FIFO is disabled, all status flags are cleared and the read and write pointers are reset. The data channel subunit sets SDIO_STS.RXRUN when data is received. The following table lists the status flags of the receive FIFO. The receive FIFO can be accessed through 32 consecutive addresses.

Table 32-5 Receive FIFO Status Flags

Flag	Description
SDIO_STS.RFIFOF	This flag is set high when all 32 receive FIFO words have valid data.
SDIO_STS.RFIFOE	This flag is set high when all 32 receive FIFO words have no valid data.
SDIO_STS.RFIFOHF	This flag is set high when 8 or more receive FIFO words are empty. This flag can be used as a DMA request.
SDIO_STS.RDATVALID	This flag is set high when the receive FIFO contains valid data. The meaning of this flag is just the opposite of TFIFOE.
SDIO_STS.RXORERR	This flag is set high when an underflow error occurs. This flag is cleared when writing to the SDIO clear register.

32.3.2 SDIO AHB Interface

The AHB interface implements access to SDIO registers, data FIFOs, and generation of interrupts and DMA requests. It includes data channel, register decoder, and interrupt/DMA control logic.

32.3.2.1 SDIO interrupt

When at least one of the selected status flags is high, the interrupt control logic generates an interrupt request. The interrupt enable register allows the interrupt logic to generate the corresponding interrupt.

32.3.2.2 SDIO/DMA Interface

The DMA interface provides a way to quickly transfer data directly between the SDIO data FIFO and memory.

The following example details how to implement this method. The host controller uses CMD24 (WRITE_BLOCK) to transfer 512 bytes from the host to the MMC card, and the DMA controller is used to fill the SDIO FIFO with data from the memory.

1. Complete the card identification process
2. Increase SDIO_CLK clock frequency
3. Send CMD7 command to select card and configure bus width
4. The configuration process of DMA is as follows:
    a) Enable DMA controller and clear all interrupt flags
    b) Use the base address of the memory buffer to set the source address register of DMA channel, and use the address of the SDIO_FIFO register to configure the destination address register of DMA channel.
    c) Set the control register of DMA channel (the memory address pointer is incremented, the peripheral address pointer is fixed, and the data width of memory and peripherals is word width)
    d) Enable DMA channel
5. The process of writing a data block (CMD24) is as follows:
    a) Set the SDIO data length register, write the data size in bytes into the SDIO_DATLEN register, write the block size in bytes into the SDIO_DATCTRL register, and then the host sends data in each block size (BLKSIZE).

b) Write the address of the data to the SDIO parameter register (SDIO_CMDARG), which is the address of the card that needs to transmit data

c) Set the SDIO command control register (SDIO_CMDCTRL): SDIO_CMDCTRL.CMDIDX is set to 24 (WRITE_BLOCK), SDIO_CMDCTRL.CMDRESP[1:0] is set to 1 (SDIO card host waits for a short response), SDIO_CMDCTRL.CPSMEN is set to 1 (enable SDIO card host sends commands), other fields are their reset values.

d) Wait for SDIO_STS.CMDRESPRECV bit is set, then configure the SDIO data control register(SDIO_DATCTRL) as follows: set DATEN to 1 (enable data transfer from host to card); set DATDIR to 0 (transfer direction from controller to card), set TRANSMOD to 0 (block data transfer), set DMAEN to 1 (enable DMA),set BLKSIZE[3:0] to 9 (512 bytes). No need to set other fields.

e) Wait for the status register SDIO_STS.DATBLKEND flag to be set.

6. Query the enable status register of the DMA channel to confirm that no channel is still enabled.

32.4 Card Function Description

32.4.1 Confirmation of Operating Voltage Range

All cards can use any voltage within the specified range to communicate with the SDIO card host, the minimum and maximum voltage VDD values that can be supported are defined by the operating condition register (OCR) on the card. When communication between the host and the card is initiated, the host may not know the voltages supported by the card, and the card may not know whether the host can provide the voltages it supports. In order to verify the voltage, a series of special commands are required, which are defined in the relevant specifications.

The commands defined in the protocol specification include: CMD1 (SEND_OP_COND, for MMC), ACMD41 (SD_APP_OP_COND, for SD memory cards) and CMD5 (IO_SEND_OP_COND, for SD I/O cards). These commands provide a mechanism for the host to identify and reject cards that do not match the VDD range required by the host. This is because a card whose internal memory stores the Card Identification Number (CID) and Card Specific Data (CSD) can only transmit these information under the condition of data transfer VDD. When the SDIO card host module is inconsistent with the VDD range of the card, the card will not be able to complete the identification cycle and cannot send CSD data; therefore, when the VDD range does not match, the SDIO card host can use these special commands to identify and reject the card. The SDIO card host will generate the required VDD voltage when executing these commands. Cards that cannot transmit data within the specified voltage range are disconnected from the bus and become inactive.

If the card cannot operate at the supplied voltage, it does not return a response and remains in an idle state. It is mandatory to send CMD8 before ACMD41 command when initializing SD card. Receiving CMD8 is to let the card know that the host supports the physical layer 2.00 protocol and the card supports higher version functions. If the card can operate at the supplied voltage, the response will return the supply voltage and the check mode set in the command parameter.

32.4.2 Card Reset

The CMD0 command (GO_IDLE_STATE) is a software reset command, which sets the multimedia card (MMC) and

SD memory card into the idle state (Idle State). Regardless of the current card state. The reset command (CMD0) is only used for the memory part of the memory or combination card. The CMD52 command (IO_RW_DIRECT) resets the SD I/O card. Cards in the Inactive State are not affected by this command.

After the host is powered on, all cards are in the idle state (Idle State), including the cards that were previously in the inactive state (Inactive State). After power-on or after executing CMD0, the outputs of all cards are in a high-impedance state. The CMD lines of all cards are in input mode, waiting for the start bit of the next command, while all cards are initialized to a default relative card address (RCA=0x0001) and driven with a default clock frequency of 400kHz (lowest speed, maximum current drive capability).

32.4.3 Card Identification Mode

After the host resets, it enters the card identification mode to search for a new card on the bus. In card identification mode, the host resets all cards, verifies the operating voltage range, identifies the cards and asks each card for the relative card address (RCA). This operation is done separately on each card's own command signal line CMD. All data communication in card identification mode uses only the command line (CMD). During card identification, the card should operate at clock rate F_{OD} (400 kHz).

32.4.4 Card Identification Process

The identification process of different cards is different: for multimedia cards, the card identification process starts with the clock frequency F_{od}, and all SDIO_CMD outputs are driven open, allowing parallel connection of cards during this process.

The identification process of the multimedia card is as follows:

1. The bus is enabled
2. The SDIO card host broadcasts the CMD1 (SEND_OP_COND) command, receives the operating conditions, and obtains the "wire AND" of the contents of the operating condition registers of all cards
3. If the card is not compatible, it will be placed in an inactive state
4. The SDIO card host broadcasts a CMD2 command (ALL_SEND_CID) to all active cards. All active cards simultaneously send their CID numbers serially. Those cards that detect that the output CID bits do not match the data on the command line must stop sending and wait for the next recognition cycle. In the end, only one card can successfully transmit the complete CID to the SDIO card host and enter the identification state.
5. The SDIO card host sends a CMD3 command (SET_RELATIVE_ADDR) to the card. This new address is called the relative card address (RCA), which is shorter than the CID and is used to address the card. At this point, the card goes into a standby state and no longer responds to the new identification process, and at the same time its output drive changes from open circuit to push-pull mode.
6. The SDIO card host repeats steps 4 and 5 above until a timeout condition is received.

The SD card identification process starts with the clock frequency F_{od}, and all SDIO_CMD outputs are push-pull drivers instead of open-circuit drivers. The identification process is as follows:

1. The bus is enabled
2. The SDIO card host broadcasts the ACMD41 (SEND_APP_OP_COND) command to get the contents of the

operating condition registers of all cards

3. If the card is not compatible, it will be placed in an inactive state
4. The SDIO card host broadcasts CMD2 (ALL_SEND_CID) to all activated cards, and all activated cards return their unique card identification number (CID) and enter the identification state.
5. The SDIO card host sends a CMD3 (SET_RELATIVE_ADDR) command and an address to an activated card, this new address is called relative card address (RCA), which is shorter than CID and is used to address the card. At this point, the card goes into the standby state. The host of the SDIO card can send this command again to change the RCA, and the RCA of the card will be the last assignment.
6. The SDIO card host repeats steps 4 and 5 above for all activated cards until a timeout condition is received.

The SD I/O card identification process is as follows:

1. The bus is enabled
2. The SDIO card host sends the CMD5 (IO_SEND_OP_COND) command to get the contents of the card's operating condition register
3. If the card is not compatible, it will be placed in an inactive state
4. The SDIO card host sends a CMD3 (SET_RELATIVE_ADDR) command and an address to an activated card, this new address is called relative card address (RCA), which is shorter than CID and is used to address the card. At this point, the card goes into the standby state. The host of the SDIO card can send this command again to change the RCA, and the RCA of the card will be the last assignment.

32.4.5 Write Data Block

When executing write data block commands (CMD24-27), one or more data blocks are transferred from the host to the card. Each data block consists of a start bit (1 bit or 4 bits low level), data block, CRC, and end bit (1 bit or 4 bits high level). If a CRC verification error occurs, the card indicates transmission failure through the SDIO_DAT signal line. The transmitted data is discarded and not written, and all subsequent data blocks transmitted (in multi-block write mode) will be ignored.

If the host transmits partial data and the accumulated data length is not aligned with the data block, and block misalignment is not allowed (parameter WRITE_BLK_MISALIGN for CSD is not set), the card will detect block misalignment errors before the start of the first unaligned block (set the ADDRESS_ERROR error bit in the status register) and ignore subsequent data transfers at the same time. The write operation is also aborted when the host attempts to write to a write-protected area, in which case the card will set the WP_VIOLATION bit in the status register.

Setting the CID and CSD registers does not require setting the block length in advance, and the transmitted data is also protected by CRC. If part of the CSD or CID register is stored in ROM, then this unchangeable part must match the corresponding part of the receive buffer. If there is an inconsistency, the card will report an error without modifying the contents of any register.

Some cards may take a long or unpredictable time to complete the writing of a data block. After receiving a data block and completing the CRC check, the card will start the write operation. If the write buffer is full and cannot be re-issued with a new WRITE_BLOCK command When receiving new data, it will pull the SDIO_DAT signal line low. The host can use SEND_STATUS (CMD13) to query the status of the card at any time, and the card will return the current status. The status bit READY_FOR_DATA indicates whether the card can accept new data or whether a write operation is still in progress. The host can unselect the card (select another card) by sending the CMD7 command, and put the card in the disconnected state, which can release the SDIO_DAT signal line without interrupting the outstanding write operation; when a card is reselected, if The write operation is still in progress and the write buffer is still unavailable, it will again indicate the busy state by pulling the SDIO_DAT signal line low.

32.4.6 Read Data Block

The read data block is a block-based data transfer. The basic unit of data transfer is a data block. The size of the block is defined in CSD (READ_BL_LEN). If READ_BL_PARTIAL is set, smaller blocks of data can also be transferred, whose start and end addresses are fully contained within the 512-byte boundary, and READ_BL_LEN defines the size of the physical block.

CMD17 (READ_SINGLE_BLOCK) means to start reading a data block, and the card returns to the sending state after the transmission is over. CMD18 (READ_MULTIPLE_BLOCK) starts to read multiple consecutive data blocks. In order to ensure the integrity of data transmission, there is a CRC check code after each data block.

The block length is set by CMD16 and can be set to 512 bytes regardless of the setting of READ_BL_LEN.

The host can abort a multi-block read operation at any time, regardless of the type of operation. Send a stop transfer command (CMD12) to abort the operation. The stop command has a delay in execution due to serial command transmission. Data transfer is stopped after the end bit of the stop command.

When using CMD18 to read the last block of userland, the host should ignore possible OUT_OF_RANGE errors, even if the sequence is correct.

If the card detects an error (for example: out-of-bounds, address misplacement, or internal error) during a multi-block read operation (of either type), it stops the data transfer and remains in the data state; at this point the host must send a stop transfer command to abort operate. Read errors are reported in response to the stop transfer command. If the host sends the stop transmission command, the card has already transmitted the last data block in a certain number of multiple data block operations, because the card is no longer in the data state at this time, the host will get an illegal command response. If the cumulative length of the partial blocks transferred by the host is not block aligned and block misalignment is not allowed, the card will detect block misalignment at the beginning of the first unaligned block, set the ADDRESS_ERROR error flag in the status register, interrupt the transfer and wait for the data Status of the stop command.

32.4.7 Data Streaming Operation (Only For Multimedia Card)

Data stream operations include data stream write and data stream read. In streaming mode, data is transferred in bytes without CRC after each data block.

32.4.7.1 Data stream write

Data stream writing to CMD20 (WRITE_DAT_UNTIL_STOP) starts to transmit data from the host to the card, starting from the starting address and continuing to transmit until the host issues a stop command. If partial data block transfers are allowed (CSD parameter WRITE_BL_PARTIAL is set), data flow can be started and stopped at any address in the card's address space, otherwise data flow should only be started and stopped at data block boundaries.

Since the amount of data to be transmitted is not predetermined, the CRC check cannot be used. If the maximum memory address is reached when sending data, subsequent data transfers will be discarded even if the SDIO card host does not send a stop command.

If the host provides an out-of-range address as a parameter to CMD20, the card will reject the command, stay in the transmit state, and set ADDRESS_OUT_OF_RANGE to 1; it should be noted that the data stream write command is only applicable to 1-bit bus configuration (SDIO_DAT0 signal line). It is considered an illegal command if CMD20 is issued in other bus configuration.

The maximum clock frequency for streaming write operations is calculated by the formula given below:

$$Max\_Write\_Frequence = Min(TRAN\_SPEED, \frac{8 \times 2^{WRITE\_BL\_LEN} - 100 \times NSAC}{TAAC \times R2W\_FACTOR})$$

- Max_Write_Frequence: maximum write frequency
- TRAN_SPEED: Maximum bus clock frequency
- WRITE_BL_LEN: maximum write block length
- NSAC: Data read operation time 2 in CLK cycles
- TAAC: Data read operation time 1
- R2W_FACTOR: Write speed factor

All parameters are defined in CSD registers. If the host attempts to use a higher frequency, the card may not be able to process the data and stop programming while setting the error bit SDIO_STS.RXORERR in the status register and ignoring all subsequent data transfers, waiting (in the receive data state) for a stop command . If the host attempts to write a value in the write-protected area, the write operation will be aborted and the card will set the WP_VIOLATION bit to 1.

32.4.7.2 Data stream read

Data Streaming Data transfer is controlled by the READ_DAT_UNTIL_STOP (CMD11) command.

This command requires the card to read data from the specified address until the SDIO card host sends a STOP_TRANSMISSION (CMD12) command. Due to the delay of serial command transmission, the execution of the stop command will have a certain delay, so the data transfer will not stop until the end bit of the stop command. If the host provides an out-of-range address as a parameter to pass to CMD11, the card will reject the command and stay in the transfer state, the SDIO card host does not send a stop command, and the subsequently transferred data is also considered invalid data.

Another point to note is that the data stream read command only works in 1-bit bus mode (DAT0 signal line). If CMD11 is issued in other bus configurations, the command is considered illegal.

The maximum clock frequency for stream read operations can be calculated as:

$$Max\_Read\_Frequence = Min(TRAN\_SPEED, \frac{8 \times 2^{READ\_BL\_LEN} - 100 \times NSAC}{TAAC \times R2W\_FACTOR})$$

- Max_Read_Frequence: maximum read frequency
- TRAN_SPEED: maximum data transfer rate

- READ_BL_LEN: maximum read data block length
- NSAC: Data read operation time 2 in CLK cycles
- TAAC: Data read operation time 1
- R2W_FACTOR: Write speed factor

If the host tries to use a higher frequency, the card will not be able to process the data transfer, at this time the card sets the SDIO_STS.TXURERR error bit in the status register, aborts the data transfer and waits for a stop command in the data state.

32.4.8 Erase

Erase includes block erase and sector erase. The erasing unit of the multimedia card is the erasing group, the basic writing unit of the card is the writing data block, and the erasing group is calculated by the writing data block. The size of the erase group is a card specific parameter and is defined in the CSD.

The host can erase a contiguous range of erase groups. There are three steps involved in starting the erase operation. First, the host uses the ERASE_GROUP_START (CMD35) command to define the start address in the contiguous range, then uses the ERASE_GROUP_END (CMD36) command to define the end address of the contiguous range, and finally sends the erase command ERASE (CMD38) to start the erase operation. In the erase command, the address field is the erase group address in bytes. The card discards the portion that is not aligned with the size of the erase group and aligns the address boundary to the boundary of the erase group.

If an erase command (CMD35, CMD36, CMD38) is not received as described above, the card should set the ERASE_SEQ_ERROR bit in the card status register and restart the erase operation (waiting for the first step).

If a command other than SEND_STATUS and erase command is received, the card should set ERASE_RESET in the status register, reset the erase sequence and execute the new command.

If the erase range includes write-protected data blocks, the write-protected area will not be erased, only non-protected blocks can be erased, and the card should set the WP_ERASE_SKIP status bit in the status register.

If the host supplies an out-of-range address as an argument to CMD35 or CMD36, the card will reject the command, set the ADDRESS_OUT_OF_RANGE bit in the status register, and reset the entire erase sequence.

Throughout the entire erasure process, the card pulls the SDIO_DAT signal low.The actual erase time may be long, and the host can send a CMD7 command to deselect the card.

32.4.9 Wide Bus Selection And De-Selection

The default bus width is 1 bit after the card is powered on or after the GO_IDLE_STATE (CMD0) command. The bus width can be changed after the host has verified the functional pins on the bus and the card is initialized.

The wide bus (4-bit bus width) operation mode can be selected by the SET_BUS_WIDTH (ACMD6) command, it should be noted that the SET_BUS_WIDTH (ACMD6) command is only valid in the transfer state, which means that only after the card is selected using the SELECT/DESELECT_CARD (CMD7) command to change the bus width.

32.4.10 Protection Management

The host supports three card protection methods to protect data from being erased or rewritten:

1. Internal write protection of the card
2. Physical write protection switch
3. Card lock operation for password management

32.4.10.1 Write Protection Of Internal Cards

By permanently or temporarily setting the write-protect bit in the CSD, the user can permanently write-protect the entire card to prevent the card's data from being overwritten or erased. Some cards set the write protection of a group of sectors by setting the WP_GRP_ENABLE bit of the CSD, so that only part of the data can be selected to be protected. Write protection can be changed by program. The basic unit of write protection is the CSD parameter WP_GRP_SIZE sectors. Users can customize the size of the write protection area by configuring the value of WP_GRP_SIZE. The SET_WRITE_PROT command sets the write protection of the specified write protection group, and the CLR_WRITE_PROT command clears the write protection of the specified write protection group. The SEND_WRITE_PROT command requests the device to send the status of the write protection bit. This command is similar to the single data block read command. A data block of write protection bits, which represents 32 write protection groups starting from the specified address, followed by a 16-bit CRC code at the end. The address field of the write protect command is a group address in bytes, the card will truncate all addresses outside the group size.

32.4.10.2 Physical Write Protection Switch

There is a mechanical slide switch on the side of the card, which provides the user with setting whether to write-protect the card. When the sliding switch is placed in the open position of the small window, the card is in write-protected state, and when the sliding switch is placed in the closed position of the small window, the card is not write-protected, and the user can modify the content in the card. There is also a switch on the corresponding part of the slot of the card to indicate whether the card is in the write-protected state. It should be noted that this instruction is for the SDIO card host module, and the internal circuit of the card does not know the position of the write-protect switch.

32.4.10.3 Password protection

Password protection means that the host module can lock or unlock the card with a password. The password is stored in the 128-bit PWD register, and the length of the password is stored in the 8-bit register of PWDS_LEN. These registers are non-volatile, so their contents are not lost after a power loss. The locked card supports all basic commands, such as reset, initialization, and status query commands sent by the SDIO card host module. If the card has previously set a password (that is, the value of PWDS_LEN is not 0), the card will be automatically locked after each power-on.

The same as the CSD and CID register write commands, the lock/unlock command is only valid in the transmission state, which also means that the card must be selected before using the lock/unlock command, and there is no address parameter in this command.

The structure and bus operation type of the lock/unlock command of the card are the same as the single data block write command of the card. The data block transmitted by the command contains the information required by the command, such as password setting mode, PWD content and lock/unlock instructions. Before sending the lock/unlock command of the card, the SDIO card host module has defined the length of the command data block.

The structure of the command is shown in Table 32-6.

Table 32-6 Lock/Unlock Data Structure

Byte	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0	Reserved (Remains at 0)				ERASE	LOCK_UNLOCK	CLR_PWD	SET_PWD
1	PWDS_LEN							
2	Password Data (PWD)							
……								
PWDS_LEN-1								

- ERASE: Setting this bit to 1 will perform a forced erase, all other bits must be 0. In this case only the command byte is sent, all other bytes of the command will be ignored by the card.
- LOCK_UNLOCK: When this bit is 1, it means the card is locked, and when it is 0, it means it is unlocked. This bit can be set at the same time as SET_PWD, but not at the same time as CLR_PWD.
- CLR_PWD: Set this bit to 1 clear password data.
- SET_PWD: This bit is 1 to save the password data to the memory.
- PWDS_LEN: This parameter defines the length of the password, in bytes.
- PWD: password data, according to different commands, the password data is different. For example, in the case of setting a new password, it contains the new password, and when changing the password, it contains the old password and the new password set.

The following sections list the command sequences to set/clear password, lock/unlock, and force wipe.

32.4.10.4 Set Password

1. If the card has not been selected before, use the CMD7 (SELECT/DESELECT_CARD) command to select a card.
2. Use CMD16 (SET_BLOCKLEN) to define the length of the data block, the 8-bit card lock/unlock mode, the 8-bit PWDS_LEN (in bytes), and the number of bytes of the new password. When the password replacement is completed, the size of the data block in which the command is sent must take into account the lengths of both the old and new passwords.
3. On the data line, send the CMD42 (LOCK/UNLOCK) command with a suitable data block length and include a 16-bit CRC code. The data block contains the operation mode (SET_PWD=1), the password length (PWDS_LEN) and the password data itself (PWD). When the password replacement is completed, the password length value (PWDS_LEN) should be the sum of the lengths of the old and new passwords. The password data field (PWD) is preceded by the old password (in use) and followed by the new password.
4. When the old password sent is incorrect (size or content does not match the expected value), LOCK_UNLOCK_FAILED in the status register will be set and the old password will not be changed. If the

old password matches, the new password data and length are stored in PWD and PWDS_LEN respectively.

The password length field (PWDS_LEN) can indicate whether a password is currently set. If the field is zero, it means that the password is not used. Only when the field is not zero, the card will be automatically locked when powered on. If a password is set, and you want to lock the card immediately without power down, you can set the LOCK_UNLOCK bit or send an additional lock command.

32.4.10.5 Clear Password

1. If the card has not been selected before, use the CMD7 (SELECT/DESELECT_CARD) command to select a card.
2. Use CMD16 (SET_BLOCKLEN) to define the data block length, 8-bit card lock/unlock mode, 8-bit PWDS_LEN (in bytes), and the number of bytes of the currently used password.
3. On the data line, send the CMD42 (LOCK/UNLOCK) command with a suitable data block length and include a 16-bit CRC code. The data block contains the operating mode (CLR_PWD), the password length (PWDS_LEN) and the password data itself (PWD). If the passwords match, the contents of PWD will be cleared and PWDS_LEN will be set to 0. If the contents of PWD and PWDS_LEN do not match the transmitted password and its size, the LOCK_UNLOCK_FAILED error bit in the status register is set and the password is unchanged.

32.4.10.6 Card Lock

1. If the card has not been selected before, use the CMD7 (SELECT/DESELECT_CARD) command to select a card.
2. Use CMD16 (SET_BLOCKLEN) to define the data block length, 8-bit card lock/unlock mode, 8-bit PWDS_LEN indicates the number of bytes of the currently used password.
3. On the data line, send the CMD42 (LOCK/UNLOCK) command with a suitable data block length and include a 16-bit CRC code. The data block contains the operation mode (LOCK_UNLOCK=1), the password length (PWDS_LEN) and the password data itself (PWD).
4. If the PWD content is equal to the sent password, the card will be locked and the CARD_IS_LOCKED status bit in the status register will be set. If the sent password does not match the expected password (length or content), the LOCK_UNLOCK_FAILED error bit in the status register is set and the lock operation fails.

Setting the password and locking the card can be performed simultaneously in the same sequence of operations. In this case, the card host module first sets the password according to the above steps. It should be noted that the LOCK_UNLOCK bit should be set in the third step of sending the new password command.

Only the card that has previously set a password (PWDS_LEN is not 0) will be automatically locked when it is powered on and reset. Locking a card that is already locked or a card without a password will fail and set the LOCK_UNLOCK_FAILED error bit in the status register.

32.4.10.7 Card Unlock

1. If the card has not been selected before, use the CMD7 (SELECT/DESELECT_CARD) command to select a card.
2. Use CMD16 (SET_BLOCKLEN) to define the data block length, 8-bit card lock/unlock mode, 8-bit PWDS_LEN (in bytes), and the number of bytes of the currently used password.

3. On the data line, send the CMD42 (LOCK/UNLOCK) command with a suitable data block length and include a 16-bit CRC code. The data block contains the operation mode (LOCK_UNLOCK=0), the password length (PWDS_LEN) and the password data itself (PWD).

4. If the sent password does not match the expected password (length or content), set the LOCK_UNLOCK_FAILED error bit in the status register to 1, while the card remains locked. When the password is matched, the card lock is released, and the CARD_IS_LOCKED bit in the status register is cleared at the same time.

If the unlocked state is only valid during the current power supply process, as long as the PWD is not cleared, the card will still be automatically locked after the next power-on.

Attempting to unlock an already unlocked card will cause the operation to fail and set the LOCK_UNLOCK_FAILED error bit in the status register.

32.4.10.8 Force Erase

A forced erase operation can erase all data and passwords in the card. If the user forgets the password, the card can be made available again through a forced erase operation.

1. If the card has not been selected before, use the CMD7 (SELECT/DESELECT_CARD) command to select a card.

2. Use CMD16 (SET_BLOCKLEN) to define the data block length, 8-bit card lock/unlock mode, 8-bit PWDS_LEN indicates the number of bytes of the currently used password.

3. Send a CMD42 (LOCK/UNLOCK) command with a suitable data block length on the data line and include a 16-bit CRC code. The data block contains the operating mode (ERASE=1) all other bits are 0.

4. If and only if the ERASE bit in the data field is 1, all contents of the card will be erased, including the PWD and PWDS_LEN fields. The card is no longer locked after erasing. If any other bit is not 0, set the LOCK_UNLOCK_FAILED error bit in the status register, the data in the card remains unchanged, and the card remains locked.

NOTE: Attempting to perform an erase operation on an already unlocked card will cause the operation to fail and set the LOCK_UNLOCK_FAILED error bit in the status register.

32.4.11 Card Status Register

32.4.11.1 Card status register

Card status refers to the error and status information of the executed command, indicated in the response.

Generally, after receiving a command, the card will return the status information related to the command to the card host. These status information may be stored in the local status register. This status information is called the card's status field, and the response format R1 contains a 32-bit field called the card status.

Table 32-7 defines the different status messages.

Table 32-7 Card Status

Bits	Identifier	Type	Value	Description	Clear Condition
31	ADDRESS_OUT_OF_RANGE	EXR	‘0’=no error	The address parameter in the command is	C

Bits	Identifier	Type	Value	Description	Clear Condition
			‘1’=error	out of the allowed range for the card. A multi-block or stream read/write operation (even from a valid address) attempts to read or write beyond the capacity of the card.	
30	ADDRESS_MISALIGN		‘0’=no error ‘1’=error	The first data block defined by the address parameter in the command (contrasted with the current data block length) is not aligned with the card's physical block. A multi-block or stream read/write operation (even if starting at a legal address) attempts to read or write a block of data that is not aligned with the physical block.	C
29	BLOCK_LEN_ERROR		‘0’=no error ‘1’=error	The parameter of the SET_BLOCKLEN command exceeds the maximum allowable range of the card, or the previously defined data block length is illegal for the current command (for example: the host issues a write command, the current block length is less than the minimum length allowed by the card, and at the same time Partial data block writing is not allowed).	C
28	ERASE_SEQ_ERROR		‘0’=no error ‘1’=error	Erase commands were sent in the wrong order.	C
27	ERASE_PARAM	EX	‘0’=no error ‘1’=error	An illegal erase group was selected while erasing.	C
26	WP_VIOLATION	EX	‘0’=no error ‘1’=error	An attempt was made to program a write-protected block of data.	C
25	CARD_IS_LOCKED	SR	‘0’=card unlocked ‘1’=card locked	When this bit is set, it means the card has been locked.	A
24	LOCK_UNLOCK_FAILED	EX	‘0’=no error ‘1’=error	Wrong sequence of commands in lock/unlock or wrong password detected.	C
23	COM_CRC_ERROR	ER	‘0’=no error ‘1’=error	CRC check error in previous command.	B
22	ILLEGAL_COMMAND	ER	‘0’=no error ‘1’=error	The command is illegal for the current card state.	B
21	CARD_ECC_FAILED	EX	‘0’=success ‘1’=failure	The ECC check is implemented inside the card, but it fails to correct the data.	C
20	CC_ERROR	ER	‘0’=no error	(Not defined in the standard) An error	C

Bits	Identifier	Type	Value	Description	Clear Condition
			‘1’=error	occurred inside the card, independent of a command from the host.	
19	ERROR	EX	‘0’=no error ‘1’=error	An internal card error (eg: read or write error) related to the execution of the last host command (not defined in the standard) has occurred.	C
18	Reserved				
17	Reserved				
16	CID/CSD_OVERWRITE	EX	‘0’=no error ‘1’=error	Can be any of the following errors: The CID register has been written and cannot be overwritten; The read-only portion of the CSD does not match the contents of the card; An attempt was made to reverse the copy or permanent write protection, i.e. restore or release write protection.	C
15	WP_ERASE_SKIP	EX	‘0’= not protected ‘1’= protected	Encountering an existing write-protected data block, only part of the address space is erased.	C
14	CARD_ECC_DISABLED	SX	‘0’=enabled ‘1’=disabled	No internal ECC is used when executing the command.	A
13	ERASE_RESET		‘0’=clear ‘1’=set	The sequence into the erase process was aborted because a command outside the erase sequence was received (not a CMD35, CMD36, CMD38, or CMD13 command).	C
12:9	CURRENT_STATE	SR	‘0’= Idle ‘1’= Ready ‘2’= Ident ‘3’= Standby ‘4’= Send ‘5’= Data ‘6’= Receive ‘7’= Program ‘8’= Disconnect ‘9’= Busy test	The state of the state machine in the card when the command is received. If the execution of a command results in a change of state, this change will be reflected in the response of the next command. These four bits are interpreted as decimal numbers 0 to 15.	B

Bits	Identifier	Type	Value	Description	Clear Condition
			'10~15'= Reserved		
8	READY_FOR_DATA	SR	'0'= no ready '1'= ready	Corresponds to a buffer empty signal on the bus.	
7	SWITCH_ERROR	ER	'0'=no error '1'=error	The card did not switch to the desired mode as required by the SWITCH command.	B
6	Reserved				
5	APP_CMD	SR	'0'=enabled '1'=disabled	Card expects ACMD, or indicates that the command has been interpreted as an ACMD command.	C
4	Reserved for SD I/O Card				
3	AKE_SEQ_ERROR	ER	'0'=no error '1'=error	The order of validation is wrong.	C
2	Reserved for application specific commands				
1,0	Reserved for manufacturer test mode				

The abbreviations for the type and clear condition fields in the table are defined as follows:

Type:

- E: Error bit. Send an error condition to the host. These bits are cleared once a response (reporting an error) is issued.
- S: Status bit. These bits serve only as information fields and do not change in response to commands. These bits are persistent, they are set or cleared depending on the card state.
- R/X: Both R and X are detection bits, the difference is that R means that the card detects an anomaly in the command interpretation and verification phase (response mode), while X means that the card detects an anomaly in the command execution phase (execution mode).

SDIO card host can read these bits by sending a status command to query the status of the card.

Clear condition:

- A: According to the current state of the card
- B: Always relative to the previous command. Cleared when the correct command is received, this method has a delay of one command.
- C: Read clear

32.4.11.2 SD Status Register

The SD status contains not only status bits related to specific functions of the SD memory card, but also some status bits related to future applications. The length of the SD state is a 512-bit data block. After receiving the ACMD13 command (CMD55, then CMD13), the content of the SD status register is transferred to the SDIO card host. But it should be noted that ACMD13 commands can only be sent when the card is in the transmitting state (the card has been selected).

Table 32-8 defines the different SD status register information.

Table 32-8 SD Status

Bits	Identifier	Type	Value	Description	Clear Condition
511:510	DAT_BUS_WIDTH	SR	‘00’= 1 (Default) ‘01’= Reserved ‘10’= 4 Bit width ‘11’= Reserved	The current data bus width as defined by the SET_BUS_WIDTH command.	A
509	SECURED_MODE	SR	‘0’= Not in privacy mode ‘1’= In privacy mode	The card is in secure operation mode (see "SD Security Specification" for details).	A
508:496	Reserved				
495:480	SD_CARD_TYPE	SR	‘00xxh’= SD memory card in physical specification version 1.01~2.00 ('x' means any value). The defined cards are: '0000'= Regular SD RD/WR Card '0001'= SD ROM Card	The lower 8 bits of this field can define different variants of SD memory cards in the future (each bit can be used to define a different SD type). The upper 8 bits can be used to define SD cards that do not adhere to the current SD physical layer specification.	A
479:448	SIZE_OF_PROTECTED_AREA	SR	Protected area size (See below)	(See below)	A
447:440	SPEED_CLASS	SR	The speed type of the card (See below)	(See below)	A
439:432	PERFORMANCE_MOVE	SR	Transfer performance in units of 1MB/sec (See below)	(See below)	A
431:428	AU_SIZE	SR	Size of AU (see below)	(See below)	A
427:424	Reserved				
423:408	ERASE_SIZE	SR	Number of AUs that can be erased at one time	(See below)	A
407:402	ERASE_TIMEOUT	SR	Timeout value for the range specified by the ERASE_AU unit	(See below)	A

Bits	Identifier	Type	Value	Description	Clear Condition
401:400	ERASE_OFFSET	SR	Fixed offset value to increase when erasing	(See below)	A
399:312	Reserved				
311:0	Reserved for Manufacturer				

The abbreviations in the table for types and clearing condition fields are defined as follows:

Type:

- E: Error bit. Send an error condition to the host. These bits are cleared once a response (reporting an error) is issued.
- S: Status bit. These bits serve only as information fields and do not change in response to commands. These bits are persistent, they are set or cleared depending on the card state.
- R/X: R and X are both detection bits, the difference is that R indicates that the card detects an abnormality in the command interpretation and verification stage (response mode), while X indicates that the card detects an anomaly in the command execution stage (execution mode).

The SDIO card host queries the status of the card by sending status commands to read these bits.

Clear condition:

- A: According to the current state of the card
- B: Always relative to the previous command. Cleared when the correct command is received, this method has a delay of one command.
- C: Read clear

SIZE_OF_PROTECTED_AREA

This bit is set differently for standard-capacity and high-capacity cards.

The capacity of the protected area of a standard capacity card is calculated as follows:

Protected Area=SIZE_OF_PROTECTED_AREA * MULT * BLOCK_LEN

SIZE_OF_PROTECTED_AREA is in units of MULT * BLOCK_LEN.

The capacity of the protected area of the high-capacity card is calculated as follows:

Protected Area = SIZE_OF_PROTECTED_AREA

SIZE_OF_PROTECTED_AREA is in bytes.

SPEED_CLASS

These 8 bits indicate the type of speed and a value that can be calculated by calculating $P_W/2$ (P_W is the write performance).

Table 32-9 Speed Type Codes

SPEED_CLASS	Value definition
00h	Class 0

01h	Class 2
02h	Class 4
03h	Class 6
04h~FFh	Reserved

PERFORMANCE_MOVE

These 8 bits indicate the mobile performance (Pm) in units of 1MB/sec. If the card does not move data in RU (unit of record), Pm should be considered to be infinity. When this field is FFh, it means Pm is infinite.

Table 32-10 Mobility Performance Codes

PERFORMANCE_MOVE	Value definition
00h	Undefined
01h	1MB/sec
02h	2MB/sec
……	……
FEh	254MB/sec
FFh	Infinity

AU_SIZE

These 4 bits indicate the length of the AU, value is (16K bytes) × 2 $^{(\text{AU\_SIZE-1})}$.

Table 32-11 Au_Size Codes

AU_SIZE	Value definition
00h	Undefined
01h	16KB
02h	32KB
03h	64KB
04h	128KB
05h	256KB
06h	512KB
07h	1MB
08h	2MB
09h	4MB
Ah~Fh	Reserved

The maximum AU length is determined by the capacity of the card. The card can set any AU length between the RU length and the maximum AU length.

Table 32-12 Maximum Au Size

Capacity	16MB~64MB	128MB~256MB	512MB	1GB~32GB
Maximum AU Size	512KB	1MB	2MB	4MB

ERASE_SIZE

This 16-bit field represents NERASE. When NERASE AUs are erased, the timeout period is defined by ERASE_TIMEOUT. The host should determine the appropriate number of AUs to be erased in one operation so that the host can display the progress of the erase operation. If this field is 0, the timeout calculation for erasure is not supported.

Table 32-13 Erase_Size Codes

ERASE_SIZE	Value Definition
0000h	Erase timeout calculation is not supported.
0001h	1 AU
0002h	2 AU
0003h	3 AU
……	……
FFFFh	65535 AU

ERASE_TIMEOUT

These 6 bits represent TERASE. When multiple AUs indicated by ERASE_SIZE are erased, this value gives the erase timeout time from the offset. The range of ERASE_TIMEOUT can be defined up to 63 seconds. The manufacturer of the card can choose any combination of ERASE_SIZE and ERASE_TIMEOUT according to the specific implementation. Once ERASE_TIMEOUT is determined, then ERASE_SIZE is also determined. If the ERASE_SIZE field is set to 0, then ERASE_TIMEOUT should also be set is 0.

Table 32-14 Erase Timeout Code

ERASE_TIMEOUT	Value definition
00	Erase timeout calculation is not supported.
01	1 sec
02	2 sec
03	3 sec
……	……
63	63 sec

ERASE_OFFSET

These 2 bits give TOFFSET, which can select one of the four values shown in the table below. When ERASE_SIZE and ERASE_TIMEOUT are both 0, this value has no meaning.

Table 32-15 Erase Offset Codes

ERASE_OFFSET	Value definition
0	0 sec
1	1 sec
2	2 sec
3	3 sec

32.4.12 SD I/O mode

32.4.12.1 I/O interrupts

There is a pin (pin 8) with interrupt function on the SD interface, which enables the SD I/O card to interrupt the multi-media card/SD module. In 4-bit SD mode, this pin is SDIO_DAT1, through which the card requests an interrupt to the multi-media card. For each card or function within the card, the interrupt function is optional.

The interrupt of SD I/O is level-effective, that is, the interrupt signal line must maintain the active level (low) before it can be recognized and responded by the multimedia card/SD module, and remains inactive level (high) after the interrupt process ends. After the interrupt request has been serviced by the Multi MediaCard/SD module, the interrupt status bits can be cleared by writing the appropriate bits to the SD I/O card's internal registers through an I/O write operation.

The interrupt outputs of the SD I/O card are all active low, and the multimedia card/SD module provides pull-up resistors on all data lines (SDIO/D[3:0]). The multimedia card/SD module samples the 8th pin (SDIO_DAT/IRQ) and performs interrupt detection only in the interrupt stage, and ignores the value on the signal line at other times.

Both I/O operations and memory operations have interrupt stages, and the definition of the interrupt stage for single data block operations and multiple data block transfer operations is different.

32.4.12.2 I/O Suspend And Resume

In a multifunction SD I/O card, or in a card with both I/O and memory functions, multiple devices (I/O and memory) share the MMC/SD bus. In order to enable multiple devices to share the bus in the MMC/SD module, SD I/O cards and composite cards can selectively implement the concept of suspend/resume; in cards that support suspend/resume, the MMC/SD module can suspend a A data transfer operation of a function or memory to give up the bus to another function or memory with a higher priority, and resume the previously suspended transfer after the transfer of the function or memory with a higher priority is completed.

Whether to support suspend/resume operations is optional. Following are the steps to perform a suspend/resume operation on the MMC/SD bus:

1. Determine the current function of the data line (SDIO_DAT[3:0])
2. Request to suspend low-priority or slow operations
3. Wait for the pause operation to complete and confirm that the device has been paused
4. Start the transfer of the high-priority device
5. Wait for the high-priority device to complete the transfer
6. Resume from a suspend operation

32.4.12.3 I/O ReadWait

The Read Wait operation is optional and only works in 1-bit or 4-bit mode of the SD card. The read wait operation means that when a card is reading multiple registers (IO_RW_EXTENDED, CMD53), the MMC/SD module can ask it to temporarily stop data transmission, and at the same time allow the MMC/SD module to send commands to other functions in the SD I/O device. The MMC/SD module can judge whether a card supports the read waiting protocol by detecting the internal registers of the card. The read wait time is related to the interrupt phase.

32.5 Commands And Responses

32.5.1 Application Related Commands And General Commands

The SD card host module system is a standard interface, which is suitable for a variety of application types, while taking into account specific users and applications, so two types of general commands are defined in the standard: application-related commands (ACMD) and general commands (GEN_CMD) .

When the card receives the APP_CMD (CMD55) command, the card expects the next command to be an application related command. Application Dependent Commands (ACMD) and normal multimedia card commands have the same format structure, and they can also use the same CMD number. Because it appears after APP_CMD (CMD55), the card recognizes it as an ACMD command. If the APP_CMD(CMD55) command is not followed by a defined application-related command, it is recognized as a standard command; for example: if CMD13 (SD_STATUS(ACMD13) is defined in the application) is received immediately after APP_CMD(CMD55), it will Interpreted as SD_STATUS (ACMD13); but if the card receives CMD7 immediately after APP_CMD (CMD55), and the card does not define ACMD7, it will be interpreted as a standard CMD7 (SELECT/DESELECT_CARD) command.

If you want to use the manufacturer-defined ACMD, the SD card host needs to do the following:

1. Send APP_CMD (CMD55) command

2. The card returns a response to the multimedia/SD card module, indicating that the APP_CMD bit is set and waiting for the ACMD command.

3. Send the specified ACMD

4. The card returns a response to the multimedia/SD card module, the response indicates that the APP_CMD bit is set, and the received command has been correctly parsed according to the ACMD command; if the received command is not an ACMD command, the card will be processed according to the ordinary multimedia card command, At the same time, clear the APP_CMD bit of the status register.

If an illegal command is sent, error handling will be performed according to standard illegal multimedia card commands. The bus operation process of the GEN_CMD command is the same as the single data block read and write command (WRITE_BLOCK, CMD24 or READ_SINGLE_BLOCK, CMD17), at this time, the parameter of the command indicates the direction of data transmission instead of the address, and the data block has a user-defined format and meaning.

Before sending the GEN_CMD (CMD56) command, the state machine must be in the transmission state, that is, the card must be selected, and the length of the data block is defined by SET_BLOCKLEN (CMD16). The response to the GEN_CMD (CMD56) command is in R1b format.

32.5.2 Commands Of Multimedia Card/SD Card Module

Table 32-16 Write Commands For Block-Based Transfers

CMD Index	Type	Parameter	Response Format	Abbreviation	Description
CMD23	ac	[31:16]=0 [15:0]= Number of blocks	R1	SET_BLOCK_COUNT	Defines the number of blocks that need to be transferred in a subsequent multi-block read or write command.
CMD24	adtc	[31:0]= Data Address	R1	WRITE_BLOCK	Writes a block of the length selected by the SET_BLOCKLEN command.
CMD25	adtc	[31:0]= Data Address	R1	WRITE_MULTIPLE_BLOCK	Continuously write data blocks until a

CMD Index	Type	Parameter	Response Format	Abbreviation	Description
					STOP_TRANSMISSION command is received or the specified number of blocks is reached.
CMD26	adtc	[31:0]= Stuff bits	R1	PROGRAM_CID	Program the identification register of the card. This command can only be sent once per card. There are hardware mechanisms in the card to prevent multiple programming operations. Usually this order is reserved for the manufacturer.
CMD27	adtc	[31:0]= Stuff bits	R1	PROGRAM_CSD	Program the programmable bits in the CSD of the card.
CMD28	ac	[31:0]= Data Address	R1b	SET_WRITE_PROT	If the card is write-protected, this command sets the write-protect bit of the specified group. The write protection feature is set in the special data area of the card (WP_GRP_SIZE).
CMD29	ac	[31:0]= Data Address	R1b	CLR_WRITE_PROT	If the card is write-protected, this command clears the write-protect bit of the specified group.
CMD30	adtc	[31:0]= Write Protect Data Addresses	R1	SEND_WRITE_PROT	If the card is write-protected, this command requires the card to send the status of the write-protect bit.
CMD31	Reserved				

Table 32-17 Block-Based Write-Protect Commands

CMD Index	Type	Parameter	Response Format	Abbreviation	Description
CMD28	ac	[31:0]= Data Address	R1b	SET_WRITE_PROT	If the card is write-protected, this command sets the write-protect bit of the specified group. The write protection feature is set in the special data area of the card (WP_GRP_SIZE).
CMD29	ac	[31:0]= Data Address	R1b	CLR_WRITE_PROT	If the card is write-protected, this command clears the write-protect bit of the specified group.
CMD30	adtc	[31:0]= Write Protect Data Addresses	R1	SEND_WRITE_PROT	If the card is write-protected, this command requires the card to send the status of the write-protect bit.
CMD31	Reserved				

Table 32-18 Erase Command

CMD Index	Type	Parameter	Response Format	Abbreviation	Description
CMD32 …… CMD34	Reserve. For backward compatibility with older versions of the media card protocol, these command codes cannot be used.				
CMD35	ac	[31:0]= Data Address	R1	ERASE_GROUP_START	Within the selected erase range, set the address of the first erase group.
CMD36	ac	[31:0]= Data Address	R1	ERASE_GROUP_END	Sets the address of the last erase group within the selected contiguous erase range.
CMD37	Reserve. For backward compatibility with older versions of the media card protocol, these command codes cannot be used.				
CMD38	ac	31:0]= Stuff bits	R1	ERASE	Erase the previously selected block of data.

Table 32-19 I/O Mode Command

CMD Index	Type	Parameter	Response Format	Abbreviation	Description
CMD39	ac	[31:16] =RCA [15]= Register Write Flag [14:8]= Register Address [7:0]= Register Data	R4	FAST_IO	Used to write and read 8-bit (register) data fields. This command specifies a card and register and also provides written data if the write flag is set. The R4 response contains the data read from the specified register. This command accesses application-related registers not defined in the multimedia card standard.
CMD40	bcr	[31:0]= Data Address	R5	GO_IRQ_STATE	Put the system in interrupt mode.
CMD41	Reserved				

Table 32-20 Lock Command

CMD Index	Type	Parameter	Response Format	Abbreviation	Description
CMD42	adtc	[31:0]= Stuff bits	R1b	LOCK_UNLOCK	Set/clear password or lock/unlock card. The length of the data block is set by the SET_BLOCKLEN command.
CMD43 …… CMD54	Reserved				

Table 32-21 Application Related Commands

CMD Index	Type	Parameter	Response Format	Abbreviation	Description
CMD55	ac	[31:16]=RCA [15:0]= Stuff bits	R1	APP_CMD	Indicates that the next command to the card is an application-related command rather than a standard command.
CMD56	adtc	[31:1]= Stuff bits [0]=RD/WR			In general or application-related commands, either to transfer a block of data to the card, or to read a block of data from the card. The length of the data block is set by the SET_BLOCKLEN command.
CMD57 …… CMD59	Reserved				
CMD60 …… CMD63	Reserved for manufacturer.				

32.5.3 Command Type

ACMD and GEN_CMD contain four different types:

1. Broadcast Command (BC): Sent to all cards, no response.
2. Broadcast Command with Response (BCR): sent to all cards and received responses from all cards at the same time;
3. Command (AC) with addressing (point-to-point): sent to the addressed card, there is no data transmission on the SDIO_DAT signal line.
4. Data transfer command (AC) with addressing (point-to-point): sent to the addressed card, SDIO_DAT signal line for data transfer.

32.5.4 Command Format

The command format consists of command and response.

Command: A command is used to start an operation. The host sends a command with an address or a broadcast command to a specified card or all cards (the broadcast command is only applicable to MMC V3.31 or earlier versions). All commands have a fixed length of 48 bits and are transmitted serially on the CMD line. The following table gives the general command format on Multimedia Cards, SD Memory Cards and SDIO Cards

The command channel operates in half-duplex mode, so that commands and responses can be sent and received separately. If the CPSM is not in the transmit state, the SDIO_CMD output is in a high-impedance state, as shown in Figure 32-11. Data on SDIO_CMD is synchronized with the rising edge of SDIO_CLK.

Table 32-22 Command Format

Bit	Width	Value	Description

47	1	0	Start bit
46	1	1	Transmission bit
[45:40]	6	-	Command index
[39:8]	32	-	Parameter
[7:1]	7	-	CRC7
0	1	1	End bit

Response: The response is a response to the previously received command, which is sent to the host by a card with a specified address. For all cards of MMC V3.31 or earlier, the response is sent simultaneously; the response is serially transmitted on theCMD line.

SDIO supports two response types: 48-bit short response and 136-bit long response. Both types have CRC error detection:

Note: If the response does not contain a CRC (such as the response to CMD1), the device driver SHOULD ignore the CRC failure status.

Table 32-23 Short Response Format

Bit	Width	Value	Description
47	1	0	Start bit
46	1	0	Transmission bit
[45:40]	6	-	Command index
[39:8]	32	-	Parameter
[7:1]	7	-	CRC7 (or 1111111b)
0	1	1	End bit

Table 32-24 Long Response Format

Bit	Width	Value	Description
135	1	0	Start bit
134	1	0	Transmission bit
[133:128]	6	111111	Reserved
[127:1]	127	-	CID or CSD (including internal CRC7)
0	1	1	End bit

32.5.5 Response Format

All responses are sent over the CMD signal line. Response transfers always start with the MSB bit of the corresponding response string. The length of the response string depends on the response type.

Each response contains a start bit (always 0) followed by the direction bit of the transfer (card=0). The x in the table below represents a variable part. All responses are CRC protected except for the R3 response type. Each command codeword has an end bit (always 1).

There are 5 response types, and their formats are defined as follows:

R1 (normal response command)

The code length of the R1 response is 48 bits. Where bits 45:40 indicate the command index to respond to and its value is between 0 and 63. The state of the card is encoded by 32 bits.

Table 32-25 R1 Response

Bit	Width	Value	Description
47	1	0	Start bit
46	1	0	Transmission bit
[45:40]	6	X	Command index
[39:8]	32	X	Card status
[7:1]	7	X	CRC7
0	1	1	End bit

R1b

R1b has the same format as R1, the difference is that R1b can choose to send a busy signal on the data line. After receiving these commands, depending on the state before receiving the command, the card may become busy and the host should check the busy state in the response.

R2 (CID, CSD register)

The R2 code length is 136 bits. The responses of CMD2 and CMD10 are saved in the CID register and sent. The response of CMD9 is saved in the CSD register and sent. The card only responds and transmits bits [127...1] of CID and CSD, on the receiving side, these two register reserved bits [0] are replaced with the end bit of the response. The card informs the host that it is performing an erasure operation by pulling MCDAT low.The actual erase operation may take a long time, and the host can send a CMD7 command to deselect the card.

Table 32-26 R2 Response

Bit	Width	Value	Description
135	1	0	Start bit
134	1	0	Transmission bit
[133:128]	6	‘111111’	Command index
[127:1]	127	X	Card status
0	1	1	End bit

R3 (OCR register)

The R3 code length is 48 bits. The response of CMD1 is saved in the OCR register and sent. The definition of the level code is: the limited voltage window is low and the card is busy.

Table 32-27 R3 Response

Bit	Width	Value	Description
47	1	0	Start bit
46	1	0	Transmission bit
[45:40]	6	‘111111’	Reserved
[39:8]	32	X	OCR register
[7:1]	7	‘1111111’	Reserved
0	1	1	End bit

R4 (Fast I/O)

The R4 code is 48 bits long and is only applicable to MMC cards. The parameter field includes the RCA of the specified card, the address of the register that needs to be read or written, and its content.

Table 32-28 R4 Response

Bit		Width	Value	Description
47		1	0	Start bit
46		1	0	Transmission bit
[45:40]		6	'111111'	Reserved
[39:8]Argument field	[31:16]	16	x	RCA
	[15:8]	8	x	Register address
	[7:0]	8	x	Read register contents
[7:1]		7	'1111111'	CRC7
0		1		End bit

R4b

R4b is only available for SD I/O cards, and the code length is 48 bits. The SDIO card will return a unique SDIO response R4 after receiving the CMD5 command.

Table 32-29 R4b Response

Bit		Width	Value	Description
47		1	0	Start bit
46		1	0	Transmission bit
[45:40]		6	x	Reserved
[39:8]Argument field	39	1	x	Card is ready
	[38:36]	3	x	Number of I/O functions
	35	1	x	Present memory
	[34:32]	3	x	Stuff bits
	[31:8]	24	x	I/O ORC
[7:1]		7	x'	Reserved
0		1	1	End bit

When the SD I/O card receives the command CMD5, the I/O part of the card is enabled and can respond to all subsequent commands normally. The enabled state of the I/O card will remain until the next reset, power off, or CMD52 command for I/O reset. Note that the correct response for a memory-only SD card would be 1 for the current memory and 0 for the number of I/O functions. A memory-only SD card designed according to the SD Memory Card Specification Version 1.0 treats the detected CMD5 command as an illegal command and does not respond to it. The host that can handle the I/O card will send a CMD5 command, and if the card returns a response R4, the host will determine the configuration of the card based on the data in the R4 response.

R5 (Interrupt Request)

R5 only works with multimedia cards. The code length is 48 bits. The RCA field in the parameter is 0x0 when this response is generated by the host.

Table 32-30 R5 Response

Bit		Width	Value	Description
47		1	0	Start bit
46		1	0	Transmission bit
[45:40]		6	‘111111’	CMD40
[39:8]Argument field	[31:16]	16	x	RCA[31:16] of a successful card or host
	[15:0]	16	x	Undefined. Can be used as interrupt data.
[7:1]		7	x	CRC7
0		1	1	End bit

R6 (Interrupt Request)

R6 only works with SD I/O cards. The code length is 48 bits. Bits[45:40] represent the command index to the CMD3 response. The 16 most significant bits of the parameter field are used for the published RCA number. This is the normal response of the memory device to a CMD3 command.

Table 32-31 R6 Response

Bit		Width	Value	Description
47		1	0	Start bit
46		1	0	Transmission bit
[45:40]		6	‘101000’	CMD40
[39:8]Argument field	[31:16]	16	x	RCA[31:16] of a successful card or host
	[15:0]	16	x	Undefined. Can be used as interrupt data.
[7:1]		7	x	CRC7
0		1	1	End bit

When sending a CMD3 command to an I/O-only card, the card's status bits[23:8] will change, and bit 16 in the response is the value in the I/O-only SD card, bit 15 is COM_CRC_ERROR, Bit 14 is ILLEGAL_COMMAND, Bit 13 is ERROR, Bits[12:0] are reserved.

32.6 Hardware Flow Control

Using the hardware flow control function can avoid FIFO underflow (transmit mode) and overflow (receive mode) errors.

The operation process of hardware flow control is to stop SDIO_CLK and freeze the SDIO state machine. When the FIFO cannot send and receive data, the data transmission is suspended. It should be noted that only the state machine driven by SDIO_CLK is frozen, the AHB interface is still working. Even when flow control is in effect, the FIFO can still be read from or written to.

The SDIO_CLKCTRL.HWCLKEN bit must be set to '1' to enable hardware flow control. After reset, the hardware flow control function is automatically disabled.

32.7 SDIO Register

Devices communicate with the system through control registers. The width of the control registers is 32 bits wide, and these registers can be manipulated on the AHB bus. Note that these peripheral registers must be manipulated in word (32 bits).

32.7.1 SDIO Register Overview

SDIO base address: 0x40021400

Table 32-32 SDIO Register Overview

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
000h	SDIO_PWRCTRL	Reserved																														PWR CTRL	
	Reset Value																															0	0
004h	SDIO_CLKCTRL	Reserved														HWCLKEN	CLKEDGE	PWRCFG	CLKBYP	CLOCKEN	BUS MODE [1:0]		Reserved		DIV[8:0]								
	Reset Value															0	0	0	0	0	0	0			0	0	0	0	0	0	0	0	0
008h	SDIO_CMDARG	CMDARG[31:0]																															
	Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
00Ch	SDIO_CMDCTRL	Reserved																CMD RESP [1:0]		CMDIDX[5:0]						Reserved			SUSPEND	ENCMDF	WINTREQ	WDATEND	CPSMEN
	Reset Value																	0	0	0	0	0	0	0	0				0	0	0	0	0
010h	SDIO_CMDRESP	Reserved																										RESPCMDIDX[5:0]					
	Reset Value																											0	0	0	0	0	0
014h	SDIO_RESPONSE1	CARDSTS1[31:0]																															
	Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
018h	SDIO_RESPONSE2	CARDSTS2[31:0]																															
	Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
01Ch	SDIO_RESPONSE3	CARDSTS3[31:0]																															
	Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
020h	SDIO_RESPONSE4	CARDSTS4[31:0]																															
	Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
024h	SDIO_DTIMER	DATTIMEOUT[31:0]																															
	Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
028h	SDIO_DATLEN	Reserved							DATLEN[24:0]																								
	Reset Value								0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
02Ch	SDIO_DATCTRL	Reserved																			DMADIR	BLKSIZE[3:0]				SDIOEN	RWAITSTOP	RWAITMOD	RWAITEN	DMAEN	DATDIR	TRANSMOD	DATEN
	Reset Value																				0	0	0	0	0	0	0	0	0	0	0	0	0
030h	SDIO_DATCOUNT	Reserved							DATCOUNT[24:0]																								
	Reset Value								0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
034h	SDIO_STS	Reserved									SDIOINT	RDATVALID	RFIFOF	RFIFOHF	RFIFOE	TDATVALID	TFIFOF	TFIFOHE	TFIFOE	RXRUN	TXRUN	CMDRUN	DATBLKEND	SBERR	DATEND	CMDSEND	CMDRESPRECV	RXORERR	TXURERR	DATTIMEOUT	CMDTIMEOUT	DCRCERR	CCRCERR
	Reset Value										0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
038h	SDIO_INTCLR	Reserved									SDIOINTC	Reserved											DATBLKENDC	SBERRC	DATENDC	CMDSENDC	CMDRESPRECVC	RXORERRC	TXURERRC	DATTIMEOUTC	CMDTIMEOUTC	DCRCERRC	CCRCERRC
	Reset Value										0												0	0	0	0	0	0	0	0	0	0	0
03Ch	SDIO_INTEN	Reserved									SDIOINTEN	RDATVALIDEN	RFIFOFEN	RFIFOHFEN	RFIFOEEN	TDATVALIDEN	TFIFOFEN	TFIFOHEEN	TFIFOEEN	RXRUNEN	TXRUNEN	CMDRUNEN	DATBLKENDEN	SBERREN	DATENDEN	CMDSENDEN	CMDRESPRECVEN	RXORERREN	TXURERREN	DATTIMEOUTEN	CMDTIMEOUTEN	DCRCERREN	CCRCERREN
	Reset Value										0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
048h	SDIO_FIFOCOUNT	Reserved								FIFOCOUNT[23:0]																							
	Reset Value									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
080h	SDIO_DATFIFO	FIFIDAT[31:0]																															
	Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

32.7.2 SDIO Power Control Register (SDIO_PWRCTRL)

Address offset: 0x00

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved														PWRCTRL[1:0]	
							r							rw	

Bit Field	Name	Description
31:2	Reserved	Reserved, the reset value must be maintained.
1:0	PWRCTRL[1:0]	Power supply control bits. Defines the current functional state of the card clock: 00: The power is turned off and the clock of the card is stopped. 01: Reserved. 10: Reserved, power-on state. 11: Power-on state, the clock of the card is switched on.

Note: This register cannot be written within 7 HCLK clock cycles after writing data.

32.7.3 SDIO Clock Control Register (SDIO_CLKCTRL)

Address offset: 0x04

Reset value: 0x0000 0000

The SDIO_CLK output clock is controlled by the SDIO_CLKCTRL register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved														HWCLK	CLK
														rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PWRCFG	CLKBYP	CLOCK	BUSMODE		Reserved		DIV[8:0]								
rw	rw	rw	rw								rw				

Bit Field	Name	Description
31:18	Reserved	Reserved, the reset value must be maintained.
17	HWCLKEN	HW Flow Control enable. 0: Disable hardware flow control 1: Enable hardware flow control After hardware flow control is enabled, please refer to the definition of SDIO status register for the meaning of SDIO_STS.TFIFOE and SDIO_STS.RFIFOF interrupt signals.

Bit Field	Name	Description
16	CLKEDGE	SDIO_CLK dephasing selection bit. 0: SDIO_CLK is generated on the rising edge of the master clock SDIOCLK. 1: SDIO_CLK is generated on the falling edge of the master clock SDIOCLK.
15	PWRCFG	Power saving configuration bit. To save power, the SDIO_CLK clock output can be turned off by setting the PWRCFG bit when the bus is idle. 0: SDIO_CLK is always output. 1: SDIO_CLK is only output when there is bus activity.
14	CLKBYP	Clock divider bypass enable bit. 0: Bypass off: SDIOCLK is divided according to the DIV value before driving the SDIO_CLK output signal. 1: Bypass enabled: SDIOCLK directly drives the SDIO_CLK output signal.
13	CLOCKEN	Clock enable bit. 0: SDIO_CLK is disabled. 1: SDIO_CLK is enabled.
12:11	BUSMODE[1:0]	Wide bus mode enable bit. 00: Default bus mode, use SDIO_DAT0. 01: 4-bit bus mode, use SDIO_DAT[3:0]. 10: 8-bit bus mode, use SDIO_DAT[7:0].
10:9	Reserved	Reserved, the reset value must be maintained.
8:0	DIV[8:0]	Clock divide factor. This field defines the division factor between the input clock (SDIOCLK) and the output clock (SDIO_CLK): SDIO_CLK frequency = SDIOCLK/[DIV + 2]. Note: When the clock division factor is an odd number, the duty cycle of the output clock SDIO_CLK is not 50%. It is recommended to configure the clock division factor as an even number.

Notice:

(1) When SD/SDIO card or multimedia card is in identification mode, the frequency of SDIO_CLK must be lower than 400kHz.

(2) When all cards are assigned corresponding addresses, the clock frequency can be changed to the maximum frequency allowed by the card bus.

(3) This register cannot be written within 7 HCLK clock cycles after writing data. For SD I/O cards, SDIO_CLK can be stopped during the read wait period, when SDIO_CLKCTRL register does not control SDIO_CLK.

32.7.4 SDIO Command Argument Register（SDIO_CMDARG）

Address offset：0x08

Reset value：0x0000 0000

SDIO_CMDARG register contains the 32-bit command parameter, which will be sent to the card as part of the command.

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="16">CMDARG[31:16]</td></tr>
<tr><td colspan="16">rw</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="16">CMDARG[15:0]</td></tr>
<tr><td colspan="16">rw</td></tr>
</table>

Bit Field	Name	Description
31:0	CMDARG[31:0]	Command argument. Command parameters are part of the command sent to the card. If a command contains a parameter, this register must be loaded before writing the command to the command register.

32.7.5 SDIO Command Register（SDIO_CMDCTRL）

Address offset: 0x0C

Reset value: 0x0000 0000

SDIO_CMDCTRL register contains the command index and command type bits. The command index is sent to the card as part of the command. The Command Type bit controls the Command Channel State Machine (CPSM).

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="16">Reserved</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="2">CMDRESP[1:0]</td><td colspan="6">CMDIDX[5:0]</td><td colspan="3">Reserved</td><td>SUSPEND</td><td>ENCMDF</td><td>WINT</td><td>WDAT</td><td>CPSM</td></tr>
<tr><td colspan="2">rw</td><td colspan="6">rw</td><td colspan="3"></td><td>rw</td><td>rw</td><td>rw</td><td>rw</td><td>rw</td></tr>
</table>

Bit Field	Name	Description
31:16	Reserved	Reserved, the reset value must be maintained.
15:14	CMDRESP[1:0]	Wait for response bits. These 2 bits indicate whether the CPSM needs to wait for a response, and if it needs to wait for a response, indicates the response type. 00: No response, expect SDIO_STS.CMDSEND flag 01: Short response, expect SDIO_STS.CMDRESPRECV or SDIO_STS.CCRCERR flag 10: No response, expect SDIO_STS.CMDSEND flag

Bit Field	Name	Description
		11: long response, expect SDIO_STS.CMDRESPRECV or C SDIO_STS.CRCERR flag
13:8	CMDIDX[5:0]	Command index. The command index is sent to the card as part of the command.
7:5	Reserved	Reserved, the reset value must be maintained.
4	SUSPEND	SD I/O suspend command. If this bit is set, the command to be sent is a suspend command (for SDIO cards only).
3	ENCMDF	Enable CMD completion. If this bit is set, the command complete signal is enabled.
2	WINTREQ	CPSM waits for interrupt request. If this bit is set, the CPSM turns off the command timeout control and waits for an interrupt request.
1	WDATEND	CPSM Waits for ends of data transfer (CmdPend internal signal). If this bit is set, the CPSM waits for the end of the data transfer before starting to send a command.
0	CPSMEN	Command path state machine (CPSM) Enable bit. If this bit is set, CPSM is enabled.

Note:

(1) This register cannot be written within 7 HCLK clock cycles after writing data.

(2) The multimedia card can send 2 kinds of responses: 48-bit short response, or 136-bit long response. SD cards and SD I/O cards can only send short responses, the parameters can be changed according to the type of response, and the software will distinguish the type of response according to the command sent. CE-ATA devices only send short responses.

32.7.6 SDIO Command Response Register（SDIO_CMDRESP）

Address offset: 0x10

Reset value: 0x0000 0000

SDIO_CMDRESP register contains the command index in the last received command response. If the transmitted command response does not contain a command index (long response or OCR response), although it should contain 111111b (reserved field value in the response), the content of the RESPCMDIDX field is unknown.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										RESPCMDIDX[5:0]					
										r					

Bit Field	Name	Description
31:6	Reserved	Reserved, the reset value must be maintained.
5:0	RESPCMDIDX[5:0]	Response command index. Read-only bit that contains the command index in the last command response received.

32.7.7 SDIO Response 1.4 Register（SDIO_RESPONSEx）

Address offset: 0x14 + 4*(x-1), x = 1..4

Reset value: 0x0000 0000

SDIO_ RESPONSE1/2/3/4 registers contain the status of the card, i.e. the part of the response received.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CARDSTSx[31:16]															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CARDSTSx[15:0]															
r															

Bit Field	Name	Description
31:0	CARDSTSx[31:0]	Refer to table below.

Depending on the response status, the card's status length is either 32 bits or 127 bits.

Table 32-33 Response Type And SDIO_Responsex Register

Register	Short Response	Long Response
SDIO_ RESPONSE1	Card Status [31:0]	Card Status [127:96]
SDIO_ RESPONSE2	Unused	Card Status [95:64]
SDIO_ RESPONSE3	Unused	Card Status [63:32]
SDIO_ RESPONSE4	Unused	Card Status [31:1]

The highest bit of the card status is always received first, and the lowest bit of the SDIO_ RESPONSE4 register is always 0.

32.7.8 SDIO Data Timer Register（SDIO_DTIMER）

Address offset: 0x24

Reset value: 0x0000 0000

SDIO_DTIMER register contains the data timeout in card bus clock cycles.

A counter loads the value from the SDIO_DTIMER register and counts down when the data path state machine (DPSM) enters the Wait_R or busy state, and sets the timeout flag if the counter decrements to 0 while the DPSM is in these states.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATTIMEOUT[31:16]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATTIMEOUT[15:0]															
rw															

Bit Field	Name	Description
31:0	DATTIMEOUT[31:0]	Data timeout period. Data timeout in card bus clock cycles.

Note: Before writing to the data control register for data transfer, the data timer register and data length register must be written first.

32.7.9 SDIO Data Length Register（SDIO_DATLEN）

Address offset: 0x28

Reset value: 0x0000 0000

SDIO_DATLEN register contains the length of data bytes to be transferred. When data transfer starts, this value is loaded into the data counter.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved							DATLEN [24:16]								
							rw								

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATLEN[15:0]															
rw															

Bit Field	Name	Description
31:25	Reserved	Reserved, the reset value must be maintained.
24:0	DATLEN[24:0]	Data length value. Number of data bytes to transfer.

Note: For block data transfers, the value in the data length register must be a multiple of the data block length (see SDIO_DATCTRL). Before writing to the data control register for data transfer, the data timer register and data length register must be written first.

32.7.10 SDIO Data Control Register（SDIO_DATCTRL）

Address offset: 0x2C

Reset value: 0x0000 0000

SDIO_DATCTRL register controls the Data Path State Machine (DPSM).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			DMADIR	BLKSIZE[3:0]				SDIOEN	RWAIT	RWAIT	RWAIT	DMAEN	DATDIR	TRANS	DATEN
			rw	rw				rw	rw	rw	rw	rw	rw	rw	rw

Bit Field	Name	Description
31:13	Reserved	Reserved, the reset value must be maintained.
12	DMADIR	In DMA mode, it is configured as 1 when data output is transferred to an external device, and configured as 0 when external device data is input.

11:8	BLKSIZE[3:0]	Data block size.
		When block data transfer mode is selected, this field defines the data block length:
		0000: block length = 2^0 = 1 byte;
		0001: block length = 2^1 = 2 bytes;
		0010: block length = 2^2 = 4 bytes;
		0011: block length = 2^3 = 8 bytes;
		0100: (decimal 4) block length=2^4=16 bytes;
		0101: (decimal 5) block length=2^5=32 bytes;
		0110: (decimal 6) block length = 2^6 = 64 bytes;
		0111: block length = 2^7 = 128 bytes;
		1000: block length = 2^8 = 256 bytes;
		1001: block length = 2^9 = 512 bytes;
		1010: block length = 2^{10} = 1024 bytes;
		1011: block length = 2^{11} = 2048 bytes;
		1100: block length=2^{12}=4096 bytes;
		1101: block length=2^{13}=8192 bytes;
		1110: block length = 2^{14} = 16384 bytes;
		1111: reserved.

7	SDIOEN	SD I/O enable functions. If this bit is set, the DPSM performs SD I/O card specific operations.
6	RWAITSTOP	Read wait stop. 0: If RWAITEN is set, execute read wait; 1: Stop read wait if RWAITEN is set.

Bit Field	Name	Description
5	RWAITMOD	Read wait mode. 0: Stop SDIO_CLK control read wait; 1: Use SDIO_DAT2 to control read wait.
4	RWAITEN	Read wait start. Setting this bit starts a read wait operation.
3	DMAEN	DMA enable bit. 0: Disable DMA; 1: Enable DMA.
2	DATDIR	Data transfer direction selection. 0: Controller to card; 1: Card to the controller.
1	TRANSMOD	Data transfer mode selection. 0: block data transfer; 1: Streaming data transmission.
0	DATEN	Data transfer enabled bit. If this bit is set to 1, data transfer starts. Depending on the DATDIR direction bit, the DPSM enters the Wait_S or Wait_R state, and if the RWAITEN bit is set at the very beginning of the transfer, the DPSM enters the read wait state. The enable bit does not need to be cleared after a data transfer, but SDIO_DATCTRL must be changed to allow a new data transfer.

Note: This register cannot be written within 7 HCLK clock cycles after writing data.

32.7.11 SDIO Data Counter Register（SDIO_ DATCOUNT）

Address offset: 0x30

Reset value: 0x0000 0000

When the DPSM enters the Wait_R or Wait_S state from the idle state, the SDIO_DATCOUNT register loads the value from the data length register (see SDIO_DATLEN). During the data transfer, the value of this counter is decremented until it is reduced to 0, then the SDIO_STS.DPSM enters the idle state and sets the data state End mark DATEND.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved							DATCOUNT[24:16]								
							r								

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATCOUNT[15:0]															
r															

Bit Field	Name	Description
31:25	Reserved	Reserved, the reset value must be maintained.

Bit Field	Name	Description
24:0	DATCOUNT[24:0]	Data count value. When reading this register, it returns the number of data bytes to be transmitted. Writing this register has no effect.

Note: This register can only be read at the end of a data transfer.

32.7.12 SDIO Status Register（SDIO_STS）

Address offset: 0x34

Reset value: 0x0000 0000

SDIO_STS is a read-only register that contains two types of flags:

- Static flags (bits[23:22, 10:0]): These bits can be cleared by writing to the SDIO Interrupt Clear Register (see SDIO_ INTCLR).
- Dynamic flags (bits[21:11]): The state of these bits changes according to the part of the logic they correspond to (eg: FIFO full and empty flags go high or low with data writes to the FIFO).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved									SDIO INT	RDAT VALID	RFIFOF	RFIFO HF	RFIFOE	TDAT VALID	TFIFOF
									r	r	r	r	r	r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TFIFO HE	TFIFOE	RXRUN	TXRUN	CMDRUN	DATBLK END	SBERR	DAT END	CMD SEND	CMDRES PRECV	RXOR ERR	TXUR ERR	DAT TIMEOUT	CMD TIMEOUT	DCRC ERR	CCRC ERR
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

Bit Field	Name	Description
31:23	Reserved	Reserved, the reset value must be maintained.
22	SDIOINT	SDIO interrupt received.
21	RDATVALID	Data available in receive FIFO.
20	RFIFOF	Receive FIFO full. If hardware flow control is used, the RFIFOF signal becomes active when the FIFO is still 2 words full.
19	RFIFOHF	Receive FIFO half full: There are at least 8 words left in the FIFO.
18	RFIFOE	Receive FIFO empty.
17	TDATVALID	Data available in transmit FIFO.
16	TFIFOF	Transmit FIFO full.
15	TFIFOHE	Transmit FIFO half empty: At least 8 more words can be written to the FIFO.
14	TFIFOE	Transmit FIFO empty. If hardware flow control is used, the TFIFOE signal becomes active when the FIFO contains 2

Bit Field	Name	Description
		words.
13	RXRUN	Data receive in progress.
12	TXRUN	Data transmit in progress.
11	CMDRUN	Command transfer in progress.
10	DATBLKEND	Data block sent/received (CRC check passed).
9	SBERR	Start bit not detected on all data signals in wide bus mode
8	DATEND	Data end（data counter, SDIDCOUNT, is zero）
7	CMDSEND	Command sent (no response required).
6	CMDRESPRECV	Command response (CRC check passed).
5	RXORERR	Received FIFO overrun error
4	TXURERR	Transmit FIFO underrun error
3	DATTIMEOUT	Data timeout
2	CMDTIMEOUT	Command response timeout. The command timeout is a fixed value of 64 SDIO_CLK clock cycles.
1	DCRCERR	Data block sent/received (CRC check failed).
0	CCRCERR	Command response received (CRC check failed).

32.7.13 SDIO Interrupt Clear Register（SDIO_INTCLR）

Address offset: 0x38

Reset value: 0x0000 0000

SDIO_INTCLR is a write-only register, writing '1' in the corresponding register bit will clear the corresponding bit in the SDIO_STS status register.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved									SDIOINTC	Reserved					
									w						

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved					DATBLKENDC	SBERRC	DATENDC	CMDSENDC	CMDRESPRECVC	RXORERRC	TXURERRC	DATTIMEOUTC	CMDTIMEOUTC	DCRCERRC	CCRCERRC
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

Bit Field	Name	Description
31:23	Reserved	Reserved, the reset value must be maintained.
22	SDIOINTC	SDIOINT flag clear bit. Software sets this bit to clear the SDIO_STS.SDIOINT flag.

Bit Field	Name	Description
21:11	Reserved	Reserved, the reset value must be maintained.
10	DATBLKENDC	DATBLKEND flag clear bit. Software sets this bit to clear the SDIO_STS.DATBLKEND flag.
9	SBERRC	SBERR flag clear bit. Software sets this bit to clear the SDIO_STS.SBERR flag.
8	DATENDC	DATEND flag clear bit. Software sets this bit to clear the SDIO_STS.DATEND flag.
7	CMDSENDC	CMDSEND flag clear bit. Software sets this bit to clear the SDIO_STS.CMDSEND flag.
6	CMDRESPRECVC	CMDRESPRECV flag clear bit. Software sets this bit to clear the SDIO_STS.CMDRESPRECV flag.
5	RXORERRC	RXORERR flag clear bit. Software sets this bit to clear the SDIO_STS.RXORERR flag.
4	TXURERRC	TXURERR flag clear bit. Software sets this bit to clear the SDIO_STS.TXURERR flag.
3	DATTIMEOUTC	DATTIMEOUT flag clear bit. Software sets this bit to clear the SDIO_STS.DATTIMEOUT flag.
2	CMDTIMEOUTC	CMDTIMEOUT flag clear bit. Software sets this bit to clear the SDIO_STS.CMDTIMEOUT flag.
1	DCRCERRC	DCRCERR flag clear bit. Software sets this bit to clear the SDIO_STS.DCRCERR flag.
0	CCRCERRC	CCRCERR clear bit. Software sets this bit to clear the SDIO_STS.CCRCERR flag.

32.7.14 SDIO Interrupt Enable Register（SDIO_INTEN）

Address offset: 0x3C

Reset value: 0x0000 0000

In the corresponding bit '1', the SDIO_INTEN interrupt enable register determines which status bit generates the interrupt.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved									SDIOINTEN	RDATVALIDEN	RFIFOFEN	RFIFOHFEN	RFIFOEEN	TDATVALIDEN	TFIFOFEN
									rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TFIFOHEEN	TFIFOEEN	RXRUNEN	TXRUNEN	CMDRUNEN	DATBLKENDEN	SBERREN	DATENDEN	CMDSENDEN	CMDRESPRECVEN	RXORERREN	TXURERREN	DATTIMEOUTEN	CMDTIMEOUTEN	DCRCERREN	CCRCERREN
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit Field	Name	Description
31:23	Reserved	Reserved, the reset value must be maintained.

Bit Field	Name	Description
22	SDIOINTEN	SDIO mode interrupt received interrupt enable Setting/clearing this bit by software to enable/disable the SDIO mode interrupt received interrupt function. 1: SDIO mode interrupt has been received and no interrupt will be generated 0: SDIO mode interrupt has been received and an interrupt is generated
21	RDATVALIDEN	Data available in Rx FIFO interrupt enable. Setting/clearing this bit by software to enable/disable the data valid interrupt in the receive FIFO. 0: The data in the receiving FIFO is valid and no interrupt is generated 1: The data in the receiving FIFO is valid to generate an interrupt
20	RFIFOFEN	Rx FIFO full interrupt enable. Setting/clearing this bit by software enables/disables the receive FIFO full interrupt. 0: No interrupt will be generated when the receive FIFO is full 1: Receive FIFO full and generate an interrupt
19	RFIFOHFEN	Rx FIFO half full interrupt enable. Setting/clearing this bit by software enables/disables the receive FIFO half-full interrupt. 0: No interrupt is generated when the receive FIFO is half full 1: Interrupt generated when the receive FIFO is half full
18	RFIFOEEN	Rx FIFO empty interrupt enable. Setting/clearing this bit by software to enable/disable the data valid interrupt in the transmit FIFO. 0: The data in the transmit FIFO is valid and no interrupt is generated 1: The data in the transmit FIFO is valid to generate an interrupt
17	TDATVALIDEN	Data available in Tx FIFO interrupt enable. Setting/clearing this bit by software to enable/disable the data valid interrupt in the transmit FIFO. 0: The data in the transmit FIFO is valid and no interrupt is generated 1: The data in the transmit FIFO is valid to generate an interrupt
16	TFIFOFEN	Tx FIFO full interrupt enable. Setting/clearing this bit by software to enable/disable the transmit FIFO full interrupt. 0: No interrupt will be generated when the transmit FIFO is full 1: Transmit FIFO full generates an interrupt
15	TFIFOHEEN	Tx FIFO half empty interrupt enable. Setting/clearing this bit by software to enable/disable the transmit FIFO half-empty interrupt. 0: No interrupt is generated when the transmit FIFO is half-empty 1: Transmit FIFO half empty to generate interrupt
14	TFIFOEEN	Tx FIFO empty interrupt enable. Setting/clearing this bit by software to enable/disable the transmit FIFO empty interrupt. 0: No interrupt will be generated when the transmit FIFO is empty 1: Transmit FIFO empty and generate interrupt
13	RXRUNEN	Data receive acting interrupt enable. Setting/clearing this bit by software to enable/disable the receiving data interrupt.

Bit Field	Name	Description
		0: Data is being received without generating an interrupt 1: Interrupt when data is being received
12	TXRUNEN	Data transmit acting interrupt enable. Setting/clearing this bit by software to enable/disable the Transmitting Data interrupt. 0: Data is being sent without generating an interrupt 1: Interrupt when data is being sent
11	CMDRUNEN	Command acting interrupt enable. Setting/clearing this bit by software to enable/disable the command-in-transit interrupt. 0: Transmitting command without interrupt 1: Interrupt when command is being transmitted
10	DATBLKENDEN	Data block end interrupt enable. Setting/clearing this bit by software enables/disables the end of block transfer interrupt. 0: No interrupt is generated at the end of data block transfer 1: Interrupt generated at the end of data block transfer
9	SBERREN	Start bit error interrupt enable. Setting/clearing this bit by software to enable/disable start bit error interrupt. 0: Start bit error does not generate an interrupt 1: Start bit error generates an interrupt
8	DATENDEN	Data end interrupt enable. Set/clear this bit by software to enable/disable end of data transfer interrupt. 0: No interrupt will be generated at the end of data transfer 1: Interrupt generated at the end of data transfer
7	CMDSENDEN	Command sent interrupt enable. Setting/clearing this bit by software to enable/disable the command sent interrupt. 0: The command has been sent without generating an interrupt 1: The command has been sent to generate an interrupt
6	CMDRESPRECVEN	Command response received interrupt enable. Setting/clearing this bit by software to enable/disable the receive acknowledge interrupt. 0: No interrupt is generated when a response is received 1: Receive a response and generate an interrupt
5	RXORERREN	Rx FIFO overrun error interrupt enable. Setting/clearing this bit by software to enable/disable the receive FIFO overflow error interrupt. 0: Receive FIFO overflow error does not generate an interrupt 1: Receive FIFO overflow error generates interrupt
4	TXURERREN	Tx FIFO underrun error interrupt enable. Setting/clearing this bit by software to enable/disable the transmit FIFO underflow error interrupt. 0: Transmit FIFO underflow error does not generate an interrupt 1: Transmit FIFO underflow error generates an interrupt
3	DATTIMEOUTEN	Data timeout interrupt enable. Setting/clearing this bit by software to enable/disable the data timeout interrupt.

Bit Field	Name	Description
		0: Data timeout does not generate an interrupt 1: Data timeout generates an interrupt
2	CMDTIMEOUTEN	Command timeout interrupt enable. Setting/clearing this bit by software to enable/disable the command timeout interrupt. 0: Command timeout does not generate an interrupt 1: Command timeout generates an interrupt
1	DCRCERREN	Data CRC fail interrupt enable. Setting/clearing this bit by software to enable/disable the interrupt for block CRC detection failure. 0: Data block CRC detection failure does not generate an interrupt 1: Data block CRC detection failure generates an interrupt
0	CCRCERREN	Command CRC fail interrupt enable. Setting/clearing this bit by software to enable/disable the command CRC detection failure interrupt. 0: Command CRC detection failure does not generate an interrupt 1: Command CRC detection failure generates an interrupt

32.7.15 SDIO FIFO Counter Register（SDIO_ FIFOCOUNT）

Address offset: 0x48

Reset value: 0x0000 0000

The SDIO_FIFOCOUNT register contains the number of data words that have not been written to or read from the FIFO. When the data transfer enable bit SDIO_DATCTRL.DATEN is set and the DPSM is idle, the FIFO counter is loaded with the value from the data length register (see SDIO_DATLEN). If the data length is not word-aligned (a multiple of 4), the last remaining 1~3 bytes are treated as a word.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
								FIFOCOUNT[23:16]							
								r							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FIFOCOUNT[15:0]															
r															

Bit Field	Name	Description
31:24	Reserved	Reserved, the reset value must be maintained.
23:0	FIFOCOUNT[23:0]	The number of data words to be written to or read from the FIFO.

32.7.16 SDIO Data FIFO Register(SDIO_ DATFIFO)

Address offset: 0x80

Reset value: 0x0000 0000

The receive and transmit FIFO is a 32-bit wide read or write set of registers, it contains 32 registers on consecutive 32 addresses, the CPU can use the FIFO to read and write multiple operands.

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="16">FIFODAT[31:16]</td></tr>
<tr><td colspan="16">r</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="16">FIFODAT[15:0]</td></tr>
<tr><td colspan="16">r</td></tr>
</table>

Bit Field	Name	Description
31:0	FIFODAT[31:0]	Receive and transmit FIFO data. The FIFO data occupies 32 entries of 32-bit words at the address: (SDIO base address + 0x80) to (SDIO base address + 0xFC)

33 Universal Serial Bus Full-speed Device Interface（USB_FS_Device）

33.1 Introduction

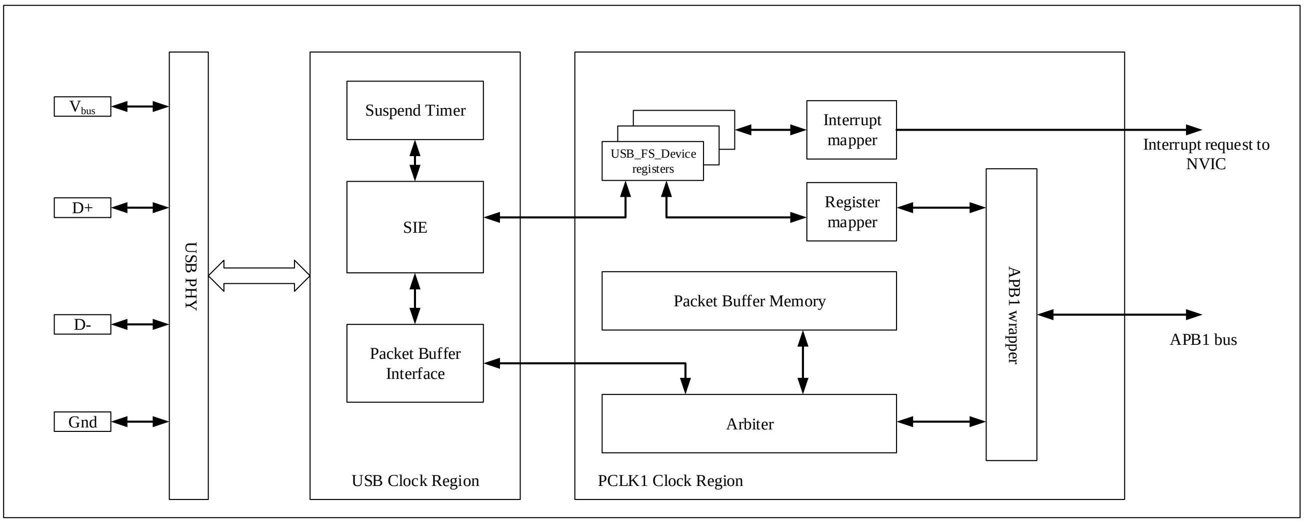
Universal serial bus full-speed device interface (USB_FS_Device) module is a peripheral that conforms to the USB2.0 full-speed protocol. It contains the USB PHY of the physical layer and does not require an additional PHY chip. USB_FS_Device supports four transfer types defined in USB2.0 protocol: control transfer, bulk transfer, interrupt transfer and isochronous transfer.

33.2 Main Features

- Comply with USB2.0 full-speed device specification
- Supports up to 8 configurable USB endpoints
- Each endpoint supports four transfer types in the USB2.0 protocol:
  - Control transfer
  - Bulk transfer
  - Interrupt transfer
  - Isochronous transfer
- Bulk endpoint/isochronous endpoint supports double buffering mechanism
- Cyclic redundancy check (CRC) generation/checking, non-return-to-zero inverted (NRZI) encoding/decoding and bit-stuffing
- Support USB suspend/resume operation
- Frame lock clock pulse generation

Figure 33-1 is a functional block diagram of a USB peripheral.

Figure 33-1 USB Device Block Diagram



33.3 Clock Configuration

The USB 2.0 protocol specification stipulates that the USB full-speed module uses a fixed 48MHz clock. In order to provide an accurate 48MHz clock to USB_FS_Device, a two-stage clock configuration is required, as follows:

- In the first stage, the 48MHz working clock is obtained by accurate frequency division of PLLCLK, so when using USB_FS_Device, it is necessary to ensure that the PLLCLK clock is 48MHz/96MHz/144MHz/192MHz/240MHz, otherwise USB_FS_Device cannot work normally;
- In the second stage, enable the USB peripheral clock mounted on the APB1 bus, that is, the APB1 bus clock. Its frequency does not have to be equal to 48MHz, but can be greater or less than 48MHz.

Note:

(1) The frequency of the APB1 bus clock must be greater than 8MHz, otherwise the data buffer may overflow/underflow.

(2) The USB registers and SRAM of the USB module can only be operated when the USB module clock is enabled.

33.4 Functional Description

Based on this module, data exchange can be realized between the microcontroller and the PC host through a USB connection. The data transfer between the microcontroller and the PC host is based on a 512-byte dedicated SRAM, which is the Packet Buffer Memory in Figure 33-1. USB peripherals can directly access this SRAM. The actual usage size of this dedicated SRAM is determined by the number of endpoints used and the endpoint packet buffer size of each endpoint. Each endpoint has a buffer description table entry, which describes the buffer address, size and the number of bytes that need to be transferred. For details, please refer to 33.4.2 Buffer Description Table. The SRAM is mapped to the APB1 peripheral memory area, its address is from 0x4000 6000 to 0x4000 63FF, the total capacity is 1KB, but only 512 bytes are used due to the bus width, and the buffer description table of each endpoint is also stored in this SRAM, so the maximum endpoint packet buffer that can be used by each endpoint is less than 512 bytes.

33.4.1 Access Packet Buffer Memory

As shown in Figure 33-1, the microcontroller communicates with the USB module through the APB1 bus, and the microcontroller accesses the Packet Buffer Memory through the APB1 wrapper. When the microcontroller and the USB module both access the Packet Buffer Memory, the Arbiter decides who can access, the arbitration logic is that half of the APB1 bus cycle is used for the microcontroller to access the Packet Buffer Memory, and the other half of the cycle is used for the USB module to access the Packet Buffer Memory, in this way, the access conflicts caused by the continuous access of the microcontroller to the Packet Buffer Memory can be avoided.

Note:

(1) APB1 bus and USB module access Packet Buffer Memory in different ways.

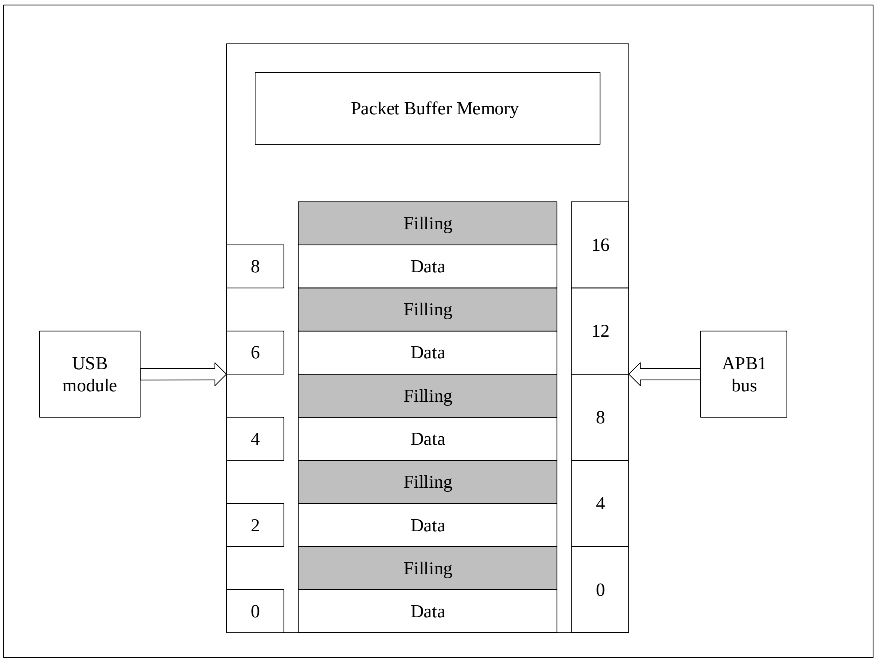
33.4.1.1 USB module access Packet Buffer Memory

The USB module accesses the Packet Buffer Memory in 16-bit mode, refer to Figure 33-2. When the USB module accesses the Packet Buffer Memory, first find the location of the buffer description table in the Packet Buffer Memory through the USB_BUFTAB register. The value of the USB_BUFTAB register indicates the starting address of the buffer description table, which must be within the memory range of the Packet Buffer Memory and be 8-byte aligned. If only endpoint 0 and endpoint 1 are used, the buffer description table only needs 16 bytes. If only endpoint 0 and endpoint 7 are used, the buffer description table needs 64 bytes. Although endpoint 1 to endpoint 6 are not used, but The description table of endpoint 7 starts from 56 bytes, so it will occupy 64 bytes of space.

33.4.1.2 User application access Packet Buffer Memory

The user application program on the microcontroller needs to access the Packet Buffer Memory from the APB1 bus according to 32-bit alignment and 16-bit read and write access, that is, the address of the operation data must be 32-bit aligned, and only 16-bit data can be read or written at a time, can't be 8-bit nor 32-bit. Figure 33-2 shows the way in which the user application program on the microcontroller and the USB module accesses the Packet Buffer Memory.

Figure 33-2 Accessing Packet Buffer Memory by the User Applications



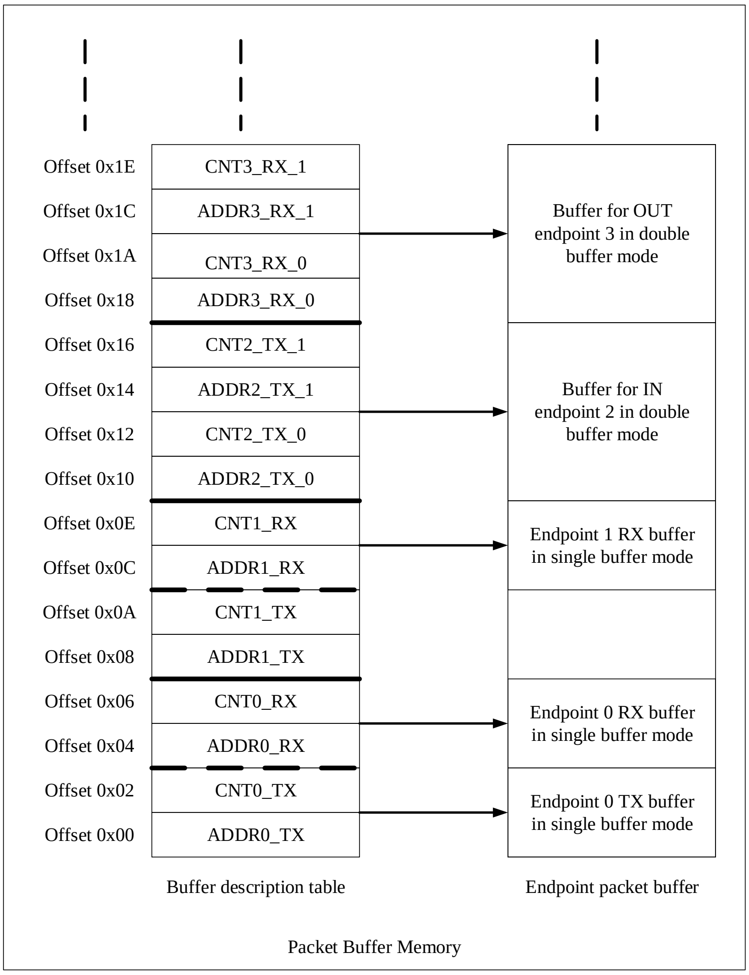
33.4.2 Buffer Description Table

The buffer description table defines the buffer address, size and the number of bytes to be transmitted for the endpoint used in the communication process. Each endpoint corresponds to two endpoint data packet buffers, one for sending and one for receiving. These endpoint packet buffers can be stored anywhere in the entire Packet Buffer Memory, and the buffer description table is also located in the Packet Buffer Memory, whose starting address is determined by the USB_BUFTAB register.

The buffer description table has a total of 8 entries, each entry corresponds to an endpoint register, each register has two directions of sending and receiving, and each direction requires two 16-bit word buffer description tables, so each table items consist of four 16-bit words, so the start address of the buffer description table must be 8-byte aligned. Endpoint packet buffers for unused endpoints or in the unused direction of a used endpoint may be used for other purposes. The relationship between the buffer description table and the endpoint packet buffer is shown in Figure 33-3 below.

Whether the endpoint is used for receiving or sending, the buffer description table starts with the first entry, which is the very bottom of the buffer description table. The USB module cannot access/modify the data of other endpoint packet buffers other than the currently allocated endpoint packet buffer area, For example: when the endpoint 0 packet receive buffer receives a data larger than the current endpoint 0 packet receive buffer from the PC host, the endpoint 0 only receives data up to the endpoint 0 packet receive buffer size, other redundant data is discarded and a buffer overflow exception occurs.

Figure 33-3 The Relationship between The Buffer Description Table and The Endpoint Packet Buffer



33.4.3 Double-buffered Endpoints

33.4.3.1 Double buffer endpoint function introduction

When a large amount of data needs to be transmitted between the PC host and the USB device, the use of bulk transmission allows the PC host to transmit data with maximum efficiency within one frame. However, when the transmission speed is too fast, the USB device will receive a new data packet when the USB device is processing the previous data transmission. In order to correctly complete the previous data transmission, the USB can only reply the NAK handshake signal to the PC host. Due to the retransmission mechanism of bulk transfer, the PC host will continue to retransmit the same data packet until the USB device can process the data packet and reply to the PC host with an ACK handshake signal, the PC host will stop retransmitting the data packet. Such retransmission will occupy a lot of bandwidth, thereby reducing the rate of bulk transfer. In order to solve this problem, a double buffering mechanism is introduced to improve the efficiency of bulk transfer, and flow control is implemented.

When the unidirectional endpoint uses the double buffer mechanism, both the receive buffer and the transmit buffer on the endpoint will be used, one of the buffers is used by the USB module, and the other buffer is used by the microcontroller, use the data toggle bit in the endpoint register to select which buffer is currently used, and introduce two flags for this: DATTOG and SW_BUF. DATTOG indicates the buffer currently being used by the USB module, and SW_BUF indicates the buffer currently being used by the application on the microcontroller. The definitions of DATTOG and SW_BUF are shown in Table 33-1 shown. A unidirectional endpoint using the double buffer mechanism only needs to use one USB_EPn register.

Table 33-1 DATTOG and SW_BUF Definitions

Buffer flag	Sending endpoint	Receiving endpoint
DATTOG	DATTOG_TX (Bit 6 of the USB_EPn register)	DATTOG_RX (Bit 14 of the USB_EPn register)
SW_BUF	Bit 14 of the USB_EPn register	Bit 6 of the USB_EPn register

As shown in Figure 33-3, when an endpoint uses the double buffer mechanism, all four buffer description table entries of the endpoint will be used. DATTOG and SW_BUF are responsible for flow control. When a transfer is complete, the USB hardware toggles the DATTOG bit; when the application on the microcontroller has finished processing the data, the software toggles SW_BUF bit. After the first transfer starts, in the subsequent transfer process, if the values of DATTOG and SW_BUF are equal, a buffer access conflict occurs between the USB module and the application, the transfer is paused, and a NAK handshake packet is sent to the host; when the values of DATTOG and SW_BUF are not equal, normal USB communication can be performed.

Table 33-2 How To Use Double Buffering

Endpoint type	DATTOG	SW_BUF	Buffer used by the USB module	Buffers used by the application
IN Endpoint	0	1	ADDRn_TX_0/CNTn_TX_0	ADDRn_TX_1/CNTn_TX_1
	1	0	ADDRn_TX_1/CNTn_TX_1	ADDRn_TX_0/CNTn_TX_0
	0	0	Endpoint is in NAK state	ADDRn_TX_0/CNTn_TX_0
	1	1	Endpoint is in NAK state	ADDRn_TX_1/CNTn_TX_1
OUT Endpoint	0	1	ADDRn_RX_0/CNTn_RX_0	ADDRn_RX_1/CNTn_RX_1
	1	0	ADDRn_RX_1/CNTn_RX_1	ADDRn_RX_0/CNTn_RX_0
	0	0	Endpoint is in NAK state	ADDRn_RX_0/CNTn_RX_0
	1	1	Endpoint is in NAK state	ADDRn_RX_1/CNTn_RX_1

Note:

1、*(1) The double-buffered bulk endpoint will only set the endpoint to the NAK state when there is a buffer access conflict, and will not set the endpoint to the NAK state after each correct transmission is completed.*

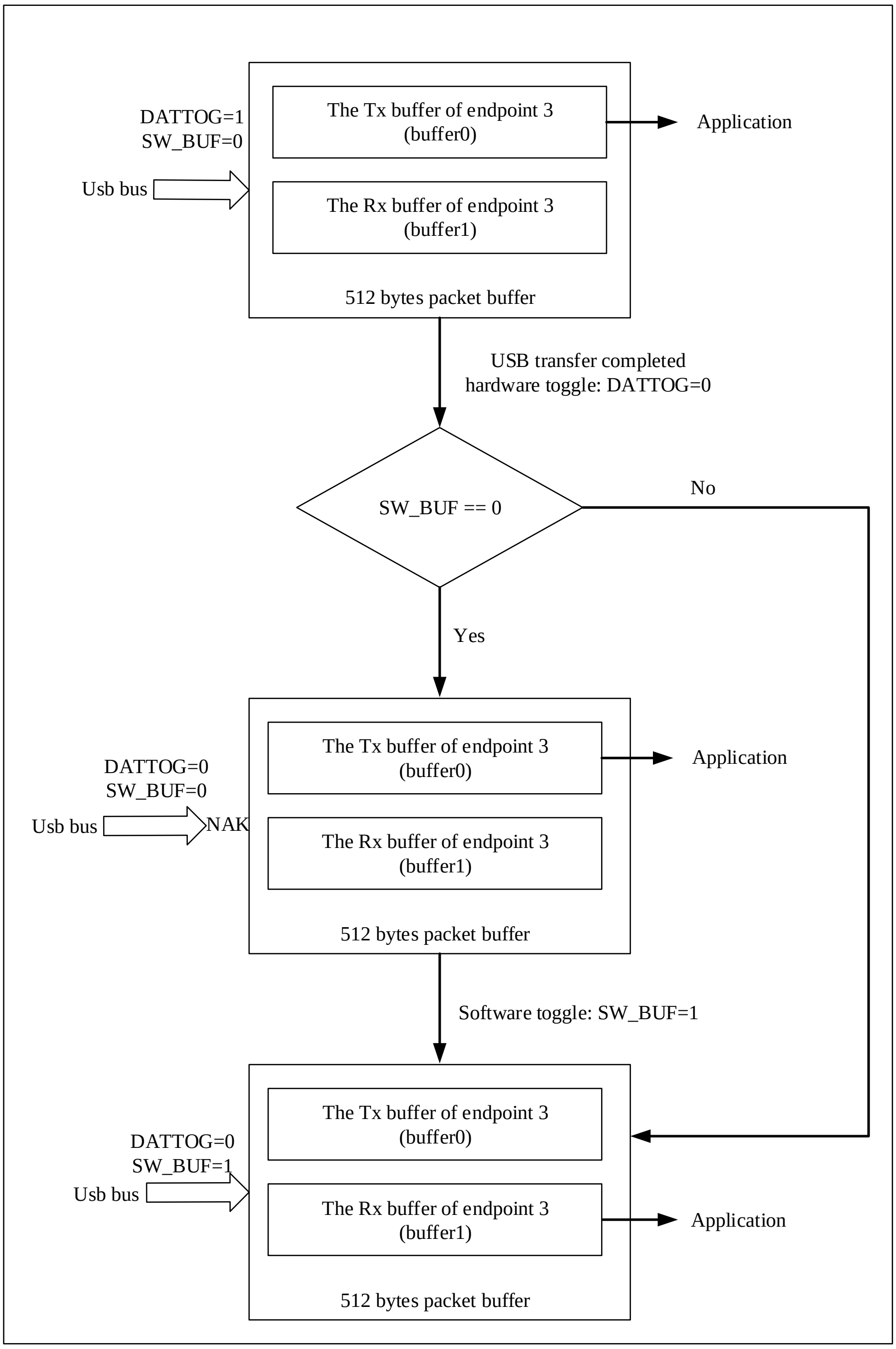
33.4.3.2 Double-buffered endpoint usage

If you want to use double-buffered bulk endpoints, you can set them up as follows:

- Set USB_EPn.EP_TYPE = 00, define the endpoint as a bulk endpoint
- Set USB_EPn.EP_KIND = 1, define endpoint as double buffer endpoint

As shown in Figure 33-3, when double-buffered bulk endpoint 3 performs data transmission in the OUT direction, assuming DATTOG = 1 and SW_BUF = 0, it means that the application can process the data in buffer0 corresponding to ADDR3_RX_0/CNT3_RX_0, after receiving the data from the USB bus, the USB module fills the data into the buffer1 corresponding to ADDR3_RX_1/CNT3_RX_1. When a transaction transfer on the USB bus is completed, the hardware will toggle DATTOG = 0. If the application has not finished processing the data in buffer0 corresponding to ADDR3_RX_0/CNT3_RX_0, the software will not toggle SW_BUF (SW_BUF = 0). If there is another OUT data packet transmission on the USB bus at this time, the USB device will automatically reply the NAK handshake signal to indicate flow control until the application finishes processing the data in buffer0 corresponding to ADDR3_RX_0/CNT3_RX_0, and the software toggle SW_BUF = 1. In this case, the DATTOG and SW_BUF values are different. If there is another OUT data packet transmission on the USB bus, the USB device can receive data normally, and fill the received data into buffer0 corresponding to ADDR3_RX_0/CNT3_RX_0, and the application can process the buffer1 corresponding to ADDR3_RX_1/CNT3_RX_1. As shown in Figure 33-4 below.

Figure 33-4 Double Buffered Bulk Endpoint Example


DATTOG=1
SW_BUF=0
Usb bus
The Tx buffer of endpoint 3 (buffer0)
The Rx buffer of endpoint 3 (buffer1)
512 bytes packet buffer
Application
USB transfer completed
hardware toggle: DATTOG=0
SW_BUF == 0
No
Yes
DATTOG=0
SW_BUF=0
Usb bus
NAK
The Tx buffer of endpoint 3 (buffer0)
The Rx buffer of endpoint 3 (buffer1)
512 bytes packet buffer
Application
Software toggle: SW_BUF=1
DATTOG=0
SW_BUF=1
Usb bus
The Tx buffer of endpoint 3 (buffer0)
The Rx buffer of endpoint 3 (buffer1)
512 bytes packet buffer
Application

33.4.4 USB Transfer

33.4.4.1 Overview of USB transfers

A USB transfer consists of multiple transactions, and a transaction consists of multiple packets.

A packet is the basic unit of USB transmission. All data must be packaged before being transmitted on the USB bus. The process of one time receiving or transmitting data on the USB is called a transaction, and there are three types of transactions: Setup transaction, Data IN transaction, and Data OUT transaction.

33.4.4.2 IN transaction

When the host wants to read the data of the USB device, the host sends a PID IN token packet to the USB device. After the USB device receives the IN token packet correctly, if the address matches a configured endpoint address, the USB module will access the corresponding USB_ADDRn_TX and USB_CNTn_TX registers according to the buffer description table entry of the endpoint, and store the values in these two registers to the internal 16-bit ADDR register and CNT register that cannot be accessed by the application. The ADDR register is used as a pointer to the endpoint's corresponding endpoint packet send buffer, and the CNT register is used to record the number of remaining untransferred bytes. The USB bus uses the low byte first method to read data from the endpoint data packet sending buffer. The data starts to read data from the endpoint data packet sending buffer pointed to by USB_ADDRn_TX, and the length is USB_CNTn_TX/2 words. If the data packet sent is an odd number of bytes, only the lower 8 bits of the last word are used.

After the USB device receives the PID IN token packet sent by the host, the USB processing flow for the IN transaction is as follows:

- If the device address information and endpoint information in this IN token packet are valid, and the status of the endpoint specified in the token packet is VALID, the USB device sends a PID DATA0 or DATA1 packet according to the USB_EPn.DATTOG_TX bit, send the prepared data to the host, when the last data byte is sent, the calculated data CRC will also be sent to the host. After the USB device receives the PID ACK handshake packet returned by the host. The hardware toggles the USB_EPn.DATTOG_TX bit, the hardware sets the endpoint's sending state to NAK state (USB_EPn.STS_TX = 10), and the hardware sets USB_EPn.CTRS_TX bit to generate a correct sending interrupt. The software responds to the CTRS_TX interrupt, identifies which endpoint the communication is on by checking the USB_STS.EP_ID bit, identifies the communication direction through USB_STS.DIR, clears the interrupt flag, and prepares the next data to be sent, and then the software sets the endpoint sending status to VALID status (USB_EPn.STS_TX = 11).
- If the endpoint specified in this IN token packet is invalid, the USB device does not send data packets, but sends PID NAK or STALL handshake packets according to USB_EPn.STS_TX.

33.4.4.3 OUT and SETUP transaction

When the host wants to transmit data or commands to the USB device, the host will send the PID OUT or SETUP token packet to the USB device. After the USB device receives the OUT or SETUP token correctly, if the address matches a configured endpoint address, the USB module will access the corresponding USB_ADDRn_RX and USB_CNTn_RX registers according to the buffer description table entry of the endpoint. Store the value of the USB_ADDRn_RX register into the internal ADDR register, and reset the internal CNT register at the same time. The ADDR register is used as a pointer to the endpoint data packet receiving buffer corresponding to the endpoint, and the CNT register is used to record the number of received data bytes, and initialize the internal 16-bit BUF_COUNT register that cannot be accessed by the application program with the BL_SIZE and NUM_BLK values in the USB_CNTn_RX register, which is used for buffer overflow detection. When the USB module receives data from the USB bus, the USB module organizes the received data in words (the first received is the low byte), and store it in the endpoint data packet receiving buffer pointed to by ADDR, at the same time, the CNT value is automatically incremented, and the BUF_COUNT value is automatically decremented.

After the USB device receives the PID OUT or SETUP token packet sent by the host, the USB processing flow for OUT or SETUP is as follows:

1、 If the device address information and endpoint information in the OUT or SETUP token packet are valid, and the status of the endpoint specified in the token packet is VALID, USB device moves data from the hardware buffer that cannot be accessed by the application to the endpoint data packet receiving buffer that can be accessed by the application. Then the USB device checks the received CRC. If the CRC is correct, the USB device replies to the host with a PID ACK handshake packet; If there is an error in the CRC or other error types (bit stuffing, frame error, etc.), the USB device will not reply to the host with an ACK handshake packet, and USB_STS.ERROR is set, at this time, the application does not need to do any processing, the USB device will automatically recover to be ready to receive the next transfer. If the received data size exceeds the data packet buffer size of the receiving endpoint, the USB device will stop receiving data, and the hardware will reply to the STALL handshake packet and set the buffer overflow error, but no interrupt will be generated. After the USB device replies the PID ACK handshake packet to the host, the USB device toggles the USB_EPn.DATTOG_RX bit by the hardware, the hardware sets the endpoint receiving state to NAK state (USB_EPn.STS_RX = 10), and the hardware sets USB_EPn.CTRS_RX to generate a correct receive interrupt. The software responds to the CTRS_RX interrupt, identifies the communication on which endpoint by checking the USB_STS.EP_ID bit, identifies the communication direction through USB_STS.DIR, clears the interrupt flag, processes the data received from the host, and after processing the received data, the software then sets the receiving state of the endpoint to the VALID state (USB_EPn.STS_RX = 11) to enable the next transmission.

2、 If the endpoint specified in this OUT or SETUP token packet is invalid, the USB device sends a PID NAK or STALL handshake packet according to USB_EPn.STS_RX.

Note:

(1) When the USB device receives data from the host, if the size of the received data exceeds the size of the data packet buffer of the receiving endpoint, the hardware will automatically stop writing, that is, the data in the data packet buffer of other endpoints will never be overwritten.

33.4.4.4 Control transfer

Control transfer consists of 3 stages, 1 Setup stage + 0/multiple Data stages in the same direction + 1 Status stage. SETUP transaction can only be completed by the control endpoint, and the process of SETUP transaction and OUT transaction is similar. When a Setup transaction is completed correctly, the hardware generates a USB_EPn.CTRS_RX interrupt. In the interrupt, the software first changes the Tx and Rx direction states of the USB device endpoint to NAK, and then checks the USB_EPn.SETUP bit to determine whether it is a SETUP transaction or an OUT transaction. And according to the corresponding fields in the SETUP token packet, it is judged whether there is a data stage in the future, and if there is a data stage, whether the data stage is IN transmission or OUT transmission. As shown in Figure 33-5, take control write transfer as an example. Before enabling subsequent data stages, determine whether the Data stage is the last Data stage:

- If it is not the last Data stage, that is, it is not the last data packet, before enabling the reception of OUT transactions, set the unused direction Tx status to STALL to prevent the host from prematurely ending the Data stage and entering the Status stage, the USB device can return a PID STALL handshake packet, and the Rx state of the direction to be used is set to VALID. When the first OUT transaction is completed correctly, the hardware generates the USB_EPn.CTRS_RX interrupt, and changes the Rx direction state of the USB device endpoint to NAK, the Tx direction state remains unchanged, the software judges whether the next OUT transaction to be enabled is the last Data stage in the interrupt. If it is not the last Data stage, before enabling the receiving OUT transaction, the software then sets the Rx direction status of the USB device endpoint to VALID, and the Tx direction status remains unchanged;

- If it is the last Data stage, before enabling the reception of the last OUT transaction, the software sets the Tx direction status that was not used in the previous Data stage to NAK, so that even if the host starts the Status stage immediately after the last Data stage, the USB device can still remain in the state of waiting for the end of the control transfer, and the Rx direction state is set to VALID, ready to receive the last packet of data;

After the last OUT transaction is completed correctly, the hardware generates the USB_EPn.CTRS_RX interrupt, and sets the Rx direction state of the USB device endpoint to NAK, and the TX direction state remains unchanged. When the software prepares the 0-length data packet that needs to be sent in the Status stage in the interrupt, the software changes the Tx direction status of the USB device endpoint to VALID.

Control read transfers are similar to control write transfers with the following differences:

- To control read transfer, after the last IN transaction in the Data stage is completed correctly, before enabling the Status stage in the OUT direction, in addition to setting the Rx direction status of the USB device endpoint to VALID, you also need to set STATUS_OUT (USB_EPn.EP_KIND) to 1, indicates that the next stage will be the Status stage in the OUT direction, and the subsequent OUT transaction must be a 0-length data packet, otherwise an error will be generated.

- After the Status stage is over, the software clears the STATUS_OUT (USB_EPn.EP_KIND) bit, the Rx direction status of the USB device endpoint is set to VALID, ready to receive a new command request, and the Tx direction status is set to NAK, indicating that before the next SETUP packet transmission is completed, the request for data transfer is not accepted.

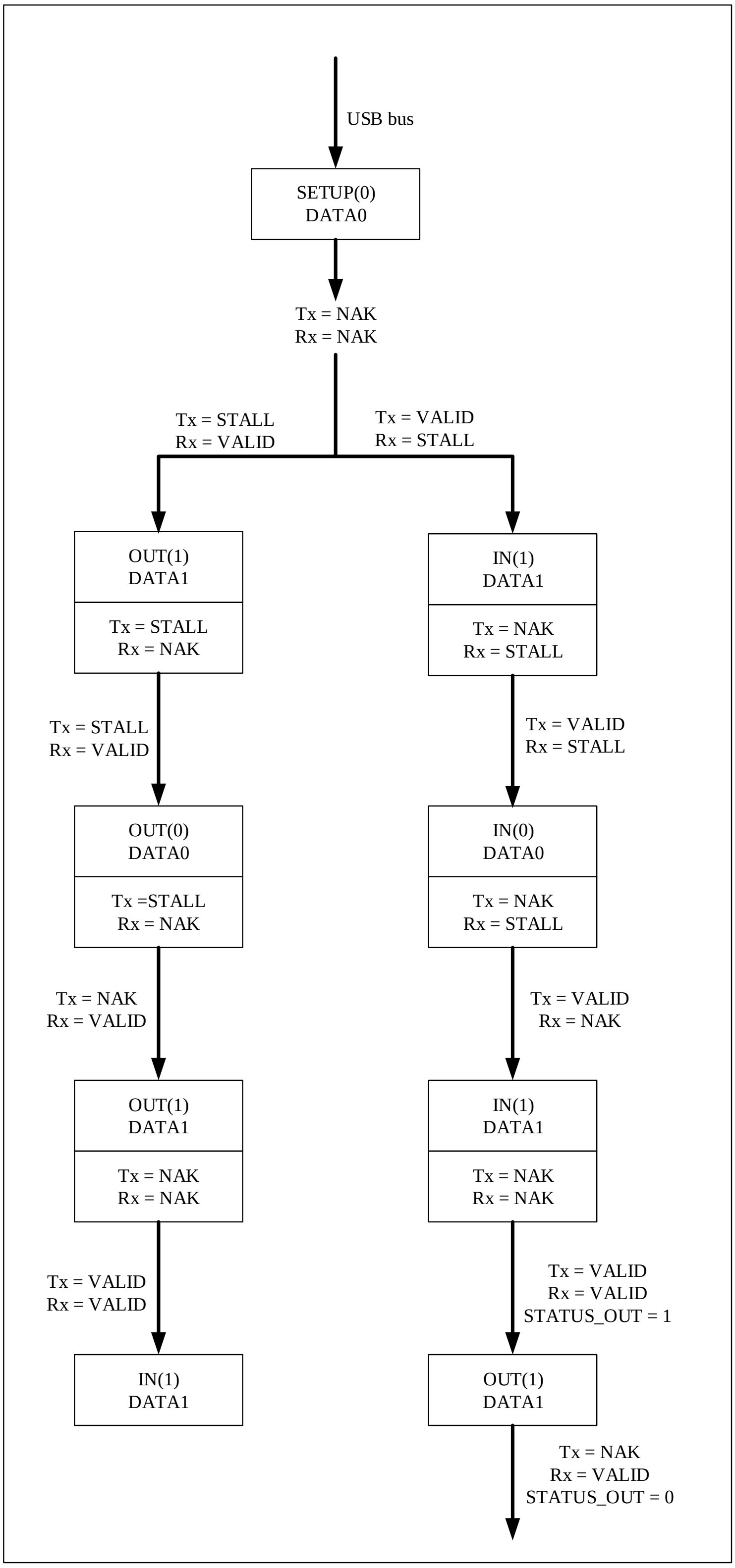
Note:

(1) Bidirectional endpoint 0 is used as the default control endpoint to handle control transfers.

(2) As defined in the USB2.0 specification, after the USB device receives the PID SETUP token packet, it cannot reply with the PID NAK or STALL handshake packet, but only with the PID ACK handshake packet. If the transmission of the SETUP packet fails, the next SETUP packet will be raised. If the Rx state of endpoint 0 is set to STALL or NAK, the USB module can still receive the SETUP token packet.

When USB_EP0.CTRS_RX = 1, the USB module receives the SETUP token packet again, the USB module will discard the SETUP token packet, and will not reply any handshake packet to the host, forcing the host to send the SETUP token packet again.

Figure 33-5 Control Transfer



33.4.4.5 Isochronous transfer

Transmissions that require a fixed and precise data rate are defined as isochronous transfer. If an endpoint is defined as a isochronous endpoint during enumeration, the USB host will allocate the required bandwidth for the endpoint in each frame of transmission, but in order to save bandwidth, isochronous transfer does not have a retransmission mechanism, that is, there is no handshake stage, there is no handshake packet after the data packet, so there is no need to use the data toggle mechanism, and the isochronous transfer only transmits the PID DATA0 data packet.

The isochronous endpoint uses a double buffer mechanism to reduce the processing pressure of the application. The buffer used by the USB module is identified by the DATTOG bit. In the same register, the USB_EPn.DATTOG_RX bit identifies the receiving isochronous endpoint, and the USB_EPn.DATTOG_TX bit identifies the sending isochronous endpoint. Compared with the bulk double buffering mechanism, the isochronous double buffering mechanism has no SW_BUF, because the buffer that the application can access is the one not indicated by DATTOG, so to achieve bidirectional isochronous transmission, two USB_EPn registers need to be used. The use of double-buffered isochronous endpoints is shown in Table 33-3.

Table 33-3 How to Use Isochronous Double Buffering

Endpoint type	DATTOG	Buffer used by the USB module	Buffers used by the application
IN Endpoint	0	ADDRn_TX_0/CNTn_TX_0	ADDRn_TX_1/CNTn_TX_1
	1	ADDRn_TX_1/CNTn_TX_1	ADDRn_TX_0/CNTn_TX_0
OUT Endpoint	0	ADDRn_RX_0/CNTn_RX_0	ADDRn_RX_1/CNTn_RX_1
	1	ADDRn_RX_1/CNTn_RX_1	ADDRn_RX_0/CNTn_RX_0

The application initializes the DATTOG bits based on the buffer to be used the first time. Each time the transfer is completed, USB_EPn.CTRS_RX or USB_EPn.CTRS_TX is set according to the direction in which the transmission is enabled, and a corresponding interrupt is generated. If a CRC error or buffer overflow error occurs, the USB_EPn.CTRS_RX or USB_EPn.CTRS_TX interrupt event can still be triggered, but if it is a CRC error, the hardware will set the USB_STS.ERROR bit, indicating that the data may be corrupted. At the same time, the hardware toggles the DATTOG bit, but the USB_EPn.STS_RX or USB_EPn.STS_TX bits are not affected.

Isochronous endpoint definition: set USB_EPn.EP_TYPE = 10. Since the isochronous endpoint has no handshake mechanism, the status of the isochronous endpoint can only be set to VALID or DISABLED, and it is illegal to set it to STALL or NAK.

Note:

(1) Compared with bulk double buffering, since isochronous *double buffering has no handshake mechanism,* isochronous *double buffering has no flow control mechanism.*

33.4.5 USB Events and Interrupts

Every USB behavior is initiated by the application and driven by USB interrupts or events. After a system reset, the application needs to wait for a series of USB interrupts and events.

33.4.5.1 Reset events

33.4.5.1.1 System reset and power-on reset

After a system reset or power-on reset occurs, the software first needs to enable the clock signal of the USB module, then clear the reset signal to access the registers of the USB module, and finally open the analog part connected to the USB transceiver. The software operation process is as follows:

- Enable the clock signal of the USB module

- Clear the USB_CTRL.PD bit
- Wait for the internal reference voltage to stabilize, because it takes a start-up time to turn on the internal voltage, during which the USB transceiver is in an indeterminate state
- Clear the USB_CTRL.FRST bit
- Clear the USB_STS register, remove pending interrupts, and enable other units

Note:

(1) Every time the USB module is enabled after system reset or power-on reset, the pull-up resistor on the DP signal line needs to be configured.

33.4.5.1.2 USB reset (reset interrupt)

When a USB reset occurs, the state of the USB module is the same as after a system reset: all endpoints are disabled for communication. The software needs to do the following:

- After the reset interrupt is generated, the software must enable the transmission of endpoint 0 within 10ms
- Set the USB_ADDR.EFUC bit
- Initialize the USB_EP0 register and its associated endpoint packet buffer

33.4.5.2 Suspend and resume events

33.4.5.2.1 Suspend events

When full-speed USB is communicating normally, the host will send a token packet with a PID of SOF every millisecond. If the USB module detects that 3 consecutive SOF packets are lost, that is, the USB bus is in an idle state within 3ms, the hardware sets the USB_STS.SUSPD bit, triggers a suspend interrupt, and the USB device enters the suspend state. The USB2.0 standard stipulates that in the suspend state, the average current consumption on the USB bus does not exceed 2.5mA, but self-powered devices do not need to strictly abide by this regulation.

Note: After the USB device enters the suspend state, it must still have the function of detecting the RESET signal.

33.4.5.2.2 Resume events

After the USB device enters the suspend state, to resume normal USB communication, the USB host can initiate a resume sequence or a reset sequence, or the USB device itself can trigger the resume sequence, but the resume sequence can only be ended by the USB host. If the reset sequence initiated by the USB host resumes the USB device, according to the regulations in the USB2.0 standard, it must be ensured that the resume process does not exceed 10ms.

Table 33-4 lists the USB_FN.RXDP_STS bit and the USB_FN.RXDM_STS bit to identify what triggers the resume event and the corresponding software action.

Table 33-4 Resume Event Detection

[USB_FN.RXDP_STS, USB_FN.RXDM_STS]	Wake-up event	Software operation
00	Root reset	None
01	Root resume	None
10	None (noise on bus)	Go back in Suspend mode
11	Not allowed (noise on bus)	Go back in Suspend mode

Note: The USB_CTRL.RESUM bit can only be set when USB_CTRL.FSUSPD = 1, i.e. the USB module is in *suspend state.*

Note: If you need to use Wakeup signal to wake up STOP0, reduce the main frequency to 144MHz or below before entering STOP0 mode.

33.4.5.3 USB interrupt

The USB controller has 3 interrupt lines, which are as follows:

- USB low priority interrupt (channel 21): can be triggered by all USB events;
- USB high-priority interrupt (channel 20): can only be triggered by correct transfer events for isochronous and double-buffered bulk transfers;
- USB resume interrupt (channel 42): triggered by a resume event from USB suspend mode.

33.4.6 Endpoint initialization

1. Initialize the USB_ADDRn_TX or USB_ADDRn_RX register, configure the endpoint Tx or Rx packet buffer start address;
2. According to the actual usage scenario of the endpoint, configure the USB_EPn.EP_TYPE bit and the USB_EPn.EP_KIND bit to set the endpoint type and buffer type;
3. Perform different operations based on the endpoint direction:
    - If it is a sending endpoint
        - Set the USB_EPn.STS_TX bit to enable the sending function of the endpoint
        - Configure the USB_CNTn_TX.CNTn_TX bit, set the endpoint data packet send buffer size
    - If it is a receiving endpoint
        - Set the USB_EPn.STS_RX bit to enable the receiving function of the endpoint

Configure the USB_CNTn_RX.BL_SIZE bit and the USB_CNTn_RX.NUM_BLK bit to set the endpoint packet receive buffer size

33.5 USB Registers

The peripheral registers can be accessed by half-words (16-bit) or words (32-bit).

USB base address: 0x4000 5C00

33.5.1 USB Register Overview

Table 33-5 USB Register Overview

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
000h	USB_EP0	Reserved																STS_TX[1:0]		DATTOG_TX	CTRS_TX	STS_RX[1:0]		DATTOG_RX	CTRS_RX	EP_TYPE[1:0]		EP_KIND	SETUP	EPADDR[3:0]			

	Reset Value		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
004h	USB_EP1	Reserved	STS_TX[1:0]		DATTOG_TX	CTRS_TX	STS_RX[1:0]		DATTOG_RX	CTRS_RX	EP_TYPE[1:0]		EP_KIND	SETUP	EPADDR[3:0]			
	Reset Value		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
008h	USB_EP2	Reserved	STS_TX[1:0]		DATTOG_TX	CTRS_TX	STS_RX[1:0]		DATTOG_RX	CTRS_RX	EP_TYPE[1:0]		EP_KIND	SETUP	EPADDR[3:0]			
	Reset Value		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
00Ch	USB_EP3	Reserved	STS_TX[1:0]		DATTOG_TX	CTRS_TX	STS_RX[1:0]		DATTOG_RX	CTRS_RX	EP_TYPE[1:0]		EP_KIND	SETUP	EPADDR[3:0]			
	Reset Value		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
010h	USB_EP4	Reserved	STS_TX[1:0]		DATTOG_TX	CTRS_TX	STS_RX[1:0]		DATTOG_RX	CTRS_RX	EP_TYPE[1:0]		EP_KIND	SETUP	EPADDR[3:0]			
	Reset Value		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
014h	USB_EP5	Reserved	STS_TX[1:0]		DATTOG_TX	CTRS_TX	STS_RX[1:0]		DATTOG_RX	CTRS_RX	EP_TYPE[1:0]		EP_KIND	SETUP	EPADDR[3:0]			
	Reset Value		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
018h	USB_EP6	Reserved	STS_TX[1:0]		DATTOG_TX	CTRS_TX	STS_RX[1:0]		DATTOG_RX	CTRS_RX	EP_TYPE[1:0]		EP_KIND	SETUP	EPADDR[3:0]			
	Reset Value		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
01Ch	USB_EP7	Reserved	STS_TX[1:0]		DATTOG_TX	CTRS_TX	STS_RX[1:0]		DATTOG_RX	CTRS_RX	EP_TYPE[1:0]		EP_KIND	SETUP	EPADDR[3:0]			
	Reset Value		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
040h	USB_CTRL	Reserved			PU	RESUM	FSUSPD	LP_MODE	PD	FRST	PMAOM	CTRSM	ERRORM	WKUPM	SUSPDM	RSTM	SOFM	ESOFM
	Reset Value				0	0	0	0	1	1	0	0	0	0	0	0	0	0
044h	USB_STS	Reserved				DIR	EP_ID[3:0]				PMAO	CTRS	ERROR	WKUP	SUSPD	RST	SOF	ESOF
	Reset Value					0	0	0	0	0	0	0	0	0	0	0	0	0
048h	USB_BUFTAB	Reserved	BUFTAB[12:0]													Reserved		
	Reset Value		0	0	0	0	0	0	0	0	0	0	0	0	0			
04Ch	USB_FN	Reserved	RXDP_STS	RXDM_STS	LSTSOF[1:0]		LOCK	FNUM[10:0]										
	Reset Value		0	0	0	0	0	X	X	X	X	X	X	X	X	X	X	X
050h	USB_ADDR	Reserved									EFUC	ADDR[6:0]						
	Reset Value										0	0	0	0	0	0	0	0

33.5.2 USB Endpoint N Register（USB_EPn）, n=[0..7]

Address offset: 0x00 ~ 0x1C

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STS_TX[1:0]		DATTOG_TX	CTRS_TX	STS_RX[1:0]		DATTOG_RX	CTRS_RX	EP_TYPE[1:0]		EP_KIND	SETUP	EPADDR[3:0]			
t		t	rc_w0	t		r	rc_w0	rw		rw	r	rw			

Bit Field	Name	Description
31：16	Reserved	Reserved, the reset value must be maintained.
15：14	STS_TX[1:0]	Send status This bit indicates the current state of the endpoint, Table 33-7 lists the available states of the endpoint. When a correct IN transaction completes, the hardware sets this bit to the NAK state. *Note:* *1、 Software can read and write this bit, but writing 0 is invalid, and writing 1 toggles this bit.* *2、 Double-buffered bulk endpoint, which controls the transmission status according to the buffer status used, refer to section 33.4.3.* *Isochronous endpoint, the hardware will not change the state of the endpoint after the transaction is successfully completed.*
13	DATTOG_TX	Send data PID toggle bit If the endpoint is not isochronous, this bit represents the toggle data bit (0 = DATA0, 1 = DATA1). Double-buffered endpoint, this bit is used to implement the flow control mechanism for double-buffered endpoints. Isochronous endpoint, this bit is used for double buffer exchange. *Note:* *1、 Software can read and write this bit, but writing 0 is invalid, and writing 1 toggles this bit.* *2、 Control endpoint, the hardware will set this bit after the USB module correctly receives the PID SETUP token packet.* *3、 In isochronous transfer, hardware toggles this bit just after the end of data packet transmission.*
12	CTRS_TX	Correct send flag This bit is set by hardware when an IN transaction on this endpoint completes successfully. If USB_CTRL.CTRSM = 1, the corresponding interrupt will be generated.

<table>
<tr><td></td><td></td><td><em>Note:</em>
<em>1. Software can read and write this bit, but only writing 0 is valid, and writing 1 is invalid.</em></td></tr>
<tr><td>11：10</td><td>STS_RX[1:0]</td><td>Receive status
This bit indicates the current state of the endpoint, Table 33-6 lists the available states of the endpoint. Hardware sets this bit to the NAK state when a correct OUT or SETUP transaction completes.
<em>Note:</em>
<em>1、 Software can read and write this bit, but writing 0 is invalid, and writing 1 toggles this bit.</em>
<em>2、 Double-buffered bulk endpoint, which controls the transmission status according to the buffer status used, refer to section 33.4.3.</em>
<em>3、 Isochronous endpoint, the hardware will not change the state of the endpoint after the transaction is successfully completed</em></td></tr>
<tr><td>9</td><td>DATTOG_RX</td><td>Receive data PID toggle bit
If the endpoint is not isochronous, this bit represents the toggle data bit (0 = DATA0, 1 = DATA1).
Double-buffered endpoint, this bit is used to implement the flow control mechanism for double-buffered endpoints.
Isochronous endpoint, this bit is used for double buffer exchange.
<em>Note:</em>
<em>1、 Software can read and write this bit, but writing 0 is invalid, and writing 1 toggles this bit.</em>
<em>2、 Control endpoint, the hardware clears this bit after the USB module correctly receives the PID SETUP token packet.</em>
<em>3、 In isochronous transfer, hardware toggles this bit just after the end of data packet reception.</em></td></tr>
<tr><td>8</td><td>CTRS_RX</td><td>Correct receive flag
This bit is set by hardware when an OUT or SETUP transaction on this endpoint completes successfully. If USB_CTRL.CTRSM = 1, the corresponding interrupt will be generated.
<em>Note:</em>
<em>1. Software can read and write this bit, but only writing 0 is valid, and writing 1 is invalid.</em></td></tr>
<tr><td>7：6</td><td>EP_TYPE[1:0]</td><td>Endpoint type
<table>
<tr><th>EP_TYPE[1:0]</th><th>Description</th></tr>
<tr><td>00</td><td>BULK: bulk endpoint</td></tr>
<tr><td>01</td><td>CONTROL: control endpoint</td></tr>
<tr><td>10</td><td>ISO: isochronous endpoint</td></tr>
<tr><td>11</td><td>INTERRUPT: interrupt endpoint</td></tr>
</table></td></tr>
<tr><td>5</td><td>EP_KIND</td><td>Endpoint special type
<table>
<tr><th colspan="2">EP_TYPE[1:0]</th><th>EP_KIND meaning</th></tr>
<tr><td>00</td><td>BULK</td><td>DBL_BUF: double buffered endpoint</td></tr>
</table></td></tr>
</table>

		01 CONTROL STATUS_OUT 10 ISO Undefined 11 INTERRUPT Undefined
4	SETUP	SETUP transfer completion flag This bit is set by hardware when the USB module correctly receives the PID SETUP token packet. *Note:* *1、 Software can only read this bit, not write this bit.* *2、 This bit USB_EPn.SETUP is only valid for control endpoints.*
3：0	EPADDR[3:0]	Endpoint address This bit indicates the destination endpoint of the communication and must be written before enabling the corresponding endpoint.

Note:

(1) When the USB module receives the USB bus reset signal, or USB_CTRL.FRST = 1, the USB module will be reset. Except for the CTRS_RX and CTRS_TX bits that remain unchanged to process the following USB transfer, all other bits are reset.

Table 33-6 Receive Status Code

STS_RX[1:0]	Description	Description
00	DISABLED: ignore all receive requests for this endpoint	DISABLED：Ignore all incoming requests for this endpoint
01	STALL: the status of the handshake packet is STALL	STALL：Handshake packet status is STALL
10	NAK: the status of the handshake packet is NAK	NAK：Handshake packet status is NAK
11	VALID: endpoints can be used to receive	VALID：Endpoint is available for reception

Table 33-7 Send Status Code

STS_TX[1:0]	Description	Description
00	DISABLED: ignore all send requests for this endpoint	DISABLED：Ignore all incoming requests for this endpoint
01	STALL: the status of the handshake packet is STALL	STALL：Handshake packet status is STALL
10	NAK: the status of the handshake packet is NAK	NAK：Handshake packet status is NAK
11	VALID: endpoints can be used to transmit	VALID：Endpoint is available for transmission

33.5.3 USB Control Register（USB_CTRL）

Address offset: 0x40

Reset value: 0x0000 0300

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

Reserved

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		PU	RESUM	FSUSPD	LP_ MODE	PD	FRST	PMAOM	CTRSM	ERRORM	WKUPM	SUSPDM	RSTM	SOFM	ESFOM
		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit Field	Name	Description
31:14	Reserved	Reserved, the reset value must be maintained.
13	PU	D+ 1.5k Pull-up resistor 0：disable 1.5k Pull-up resistor 1：enable 1.5k Pull-up resistor
12	RESUM	Resume request 0: No resume request 1: Send a resume request to the PC host *Note:* *If USB_CTRL.RESUM = 1 remains active for 1ms to 15ms, the PC host will implement a resume operation for the USB module.*
11	FSUSPD	Force suspend Software must set this bit when the USB_STS.SUSPD interrupt is triggered. 0: Suspend mode not entered 1: Enter suspend mode, but the clock and static power consumption of the USB analog transceiver are still present *Note:* *To enter the low power consumption mode (bus powered device), the software must first set USB_CTRL.FSUSPD, and then set USB_CTRL.LP_MODE.*
10	LP_MODE	Low power mode 0: No effect 1: Enter low power mode in suspend mode. Activity on the USB bus (wake event) resets this bit (software can also reset this bit) *Note:* *In low power mode, only the external pull-up resistor is used for power supply, and the system clock will also be stopped or reduced to a certain frequency to reduce power consumption.*
9	PD	Power-down mode 0: Exit power-down mode 1: Enter power-down mode *Note:* *When USB_CTRL.PD = 1, the USB module is completely shut down, disconnected from the host, and the USB module will not work.*
8	FRST	Force USB reset 0: No effect 1: Reset the USB module, if USB_CTRL.RSTM = 1, a reset interrupt will be generated

		Note: *When USB_CTRL.FRST = 1, the USB module will remain in reset state until software clears this bit.*
7	PMAOM	Packet buffer overflow/underflow interrupt enable 0: Disable packet buffer overflow/underflow interrupt 1: Enable packet buffer overflow/underflow interrupt, when USB_STS.PMAO = 1, an interrupt is generated.
6	CTRSM	Correct transfer interrupt enable 0: Disable correct transfer interrupt 1: Enable correct transfer interrupt, when USB_STS.CTRS = 1, an interrupt is generated.
5	ERRORM	Error interrupt enable 0: Disable error interrupt 1: Enable error interrupt, when USB_STS.ERROR = 1, an interrupt will be generated.
4	WKUPM	Wake-up interrupt enable 0: Disable wake-up interrupt 1: Enable wake-up interrupt, when USB_STS.WKUP = 1, an interrupt will be generated.
3	SUSPDM	Suspend mode interrupt enable 0: Disable suspend mode interrupt 1: Enable suspend mode interrupt, when USB_STS.SUSPD = 1, an interrupt will be generated.
2	RSTM	USB reset interrupt enable 0: Disable USB reset interrupt 1: Enable USB reset interrupt, when USB_STS.RST = 1, an interrupt will be generated.
1	SOFM	Start of frame interrupt enable 0: Disable start of frame interrupt 1: Enable start of frame interrupt, when USB_STS.SOF = 1, an interrupt will be generated.
0	ESOFM	Expected start of frame interrupt enable 0: Disable the expected start of frame interrupt 1: Enable the expected start of frame interrupt, when USB_STS.ESOF = 1, generate an interrupt.

33.5.4 USB Interrupt Status Register（USB_STS）

Address offset: 0x44

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			DIR	EP_ID[3:0]				PMAO	CTRS	ERROR	WKUP	SUSPD	RST	SOF	ESOF
			r	r				rc_w0	r	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0

Bit Field	Name	Description
31:13	Reserved	Reserved, the reset value must be maintained.
12	DIR	Transmission direction 0: IN packet transfer is completed, and USB_EPn.CTRS_TX is set by hardware 1: OUT packet transfer is complete, and USB_EPn.CTRS_RX is set by hardware *Note:* *1、 Software can only read this bit, not write this bit.* *2、 When USB_EPn.CTRS_TX and USB_EPn.CTRS_RX are set at the same time, it indicates that there are OUT group and IN group at the same time.*
11：8	EP_ID[3:0]	Endpoint number After the USB module completes the data transmission and generates an interrupt, it is written by the hardware according to the endpoint number of the interrupt request. *Note:* *1、 Software can only read this bit, not write this bit.* *2、 When multiple endpoint requests are interrupted at the same time, the hardware writes the endpoint number with the highest priority. Isochronous endpoints and double-buffered bulk endpoints have high priority, other endpoints have low priority (the lower the endpoint number, the higher the priority).*
7	PMAO	Packet buffer overflow/underflow interrupt flag This bit is set by hardware when the packet buffer cannot hold all the transmitted data. *Note:* *1、 Software can read and write this bit, but only writing 0 is valid, and writing 1 is invalid.* *2、 This interrupt will not be generated during isochronous transfer.*
6	CTRS	Correct transmission interrupt flag Set by hardware when the endpoint has completed a data transfer correctly. *Note:* *1、 Software can only read this bit, not write this bit.*
5	ERROR	Error interrupt flag Hardware sets this bit when the following errors occur: 1) No response, the host response timed out 2) CRC error, CRC check error in data or token packet 3) Bit stuffing error, bit stuffing error detected in PID, data or CRC 4) Frame format error, non-standard frame received *Note:* *1、 Software can read and write this bit, but only writing 0 is valid, writing 1 is invalid.*

4	WKUP	Wake-up interrupt flag In the suspend state, when the wake-up signal is detected, the hardware sets this bit, and the hardware resets the USB_CTRL.LP_MODE bit at the same time. *Note:* *1、 Software can read and write this bit, but only writing 0 is valid, writing 1 is invalid.*
3	SUSPD	Suspend mode interrupt flag This bit is set by hardware when there is no activity on the USB bus for more than 3ms, indicating a suspend request. *Note:* *1、 Software can read and write this bit, but only writing 0 is valid, and writing 1 is invalid.* *2、 In suspend mode, the USB hardware will not detect the suspend signal until the wake-up is over.* *3、 After the USB is reset, the hardware will immediately enable the detection of the suspend signal.*
2	RST	USB reset interrupt flag This bit is set by hardware when a USB reset signal is detected. *Note:* *1、 Software can read and write this bit, but only writing 0 is valid, and writing 1 is invalid.* *2、 When the USB reset interrupt is generated, the address and endpoint registers of the device will be reset, but the configuration registers will not be reset unless cleared by software.*
1	SOF	Start of frame interrupt flag This bit is set by hardware when a PID SOF token packet is detected on the USB bus. *Note:* *1、 Software can read and write this bit, but only writing 0 is valid, and writing 1 is invalid.*
0	ESOF	Expected start of frame interrupt flag This bit is set by hardware when the USB module does not receive the expected PID SOF token packet. *Note:* *1、 Software can read and write this bit, but only writing 0 is valid, and writing 1 is invalid.* *2、 When the USB module does not receive the PID SOF token packet for 3ms in a row, that is, 3 ESOF interrupts occur in a row, and a SUSPD interrupt will be generated.*

33.5.5 USB Packet Buffer Description Table Address Register（USB_BUFTAB）

Address offset: 0x48

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BUFTAB[12:0]													Reserved		
rw															

Bit Field	Name	Description
31:16	Reserved	Reserved, the reset value must be maintained.
15:3	BUFTAB[12:0]	Buffer table This bit holds the starting address of the buffer description table. The buffer description table is used to indicate the address and size of the endpoint packet buffer of each endpoint, aligned by 8 bytes (the lowest 3 bits are 000).
2:0	Reserved	Reserved, the reset value must be maintained.

33.5.6 USB Frame Number Register（USB_FN）

Address offset: 0x4C

Reset value: 0x0000 0XXX, X represents an undefined value

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RXDP_STS	RXDM_STS	LSTSOF[1:0]		LOCK	FNUM[10:0]										
r	r	r		r	r										

Bit Field	Name	Description
31:16	Reserved	Reserved, the reset value must be maintained.
15	RXDP_STS	D+ status Represents the state of the USB D+ line, and can detect the occurrence of a resume condition in the suspend state.
14	RXDM_STS	D- status Represents the state of the USB D- line, and can detect the occurrence of a resume condition in the suspend state.
13:12	LSTSOF[1:0]	Lost SOF flag The hardware increments this bit every time the USB_STS.ESOF event occurs, and once the PID SOF token packet is received, the hardware clears this bit.
11	LOCK	Lock USB This bit is set by hardware if at least 2 PID SOF token packets are detected continuously after the end of an USB reset condition or after the end of an USB resume sequence.

		Note: *1、 When USB_FN.LOCK = 1, the frame counter will stop counting before the USB module is reset or the USB bus is suspended.*
10:0	FNUM[10:0]	Number of frames Hardware increments this bit every time the USB module receives a PID SOF token packet.

33.5.7 USB Device Address Register（USB_ADDR）

Address offset：0x50

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								EFUC	ADD[6:0]						
								rw	rw						

Bit Field	Name	Description
31:8	Reserved	Reserved, the reset value must be maintained.
7	EFUC	USB module enable 0: The USB module stops working and does not respond to any USB communication 1: Enable USB module
6：0	ADDR[6:0]	USB device address This bit holds the address value assigned to the USB device by the USB host during enumeration. After a USB bus reset, this bit is reset to 0x00.

33.6 Buffer Description Table

The buffer description table is located in the packet buffer memory and is used to configure the address and size of the endpoint packet buffer shared by the USB module and the microcontroller core. Since the APB1 bus is addressed by 32 bits, the data packet buffer memory addresses use 32-bit aligned addresses, not the addresses used by the USB_BUFTAB register and the buffer description table.

33.6.1 Send Buffer Address Register N（USB_ADDRn_TX）

Address offset: [USB_BUFTAB] + n×16

USB local address：[USB_ BUFTAB] + n×8

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDRn_TX[14:0]															

rw rw

Bit Field	Name	Description
15：1	ADDRn_TX[14:0]	Send buffer address The starting address of the endpoint packet buffer of the endpoint that needs to send data when the next PID IN token packet is received
0	-	Since packet buffer memory addresses are word (32-bit) aligned, this bit must be 0

33.6.2 Send Data Byte Number Register N（USB_CNTn_TX）

Address offset：[USB_ BUFTAB] + n×16 + 4

USB local address: [USB_ BUFTAB] + n×8 + 2

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						CNTn_TX[9:0]									
						rw									

Bit Field	Name	Description
15:10	Reserved	Reserved, the reset value must be maintained.
9：0	CNTn_TX[9:0]	Number of bytes sent The number of data bytes to send on the next PID IN token packet

Note: As shown in Table 33-2 and Table 33-3, the double-buffered IN endpoint and the isochronous IN endpoint require two USB_CNTn_TX registers: USB_CNTn_TX_0 and USB_CNTn_TX_1.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved						CNTn_TX_1[9:0]									
						rw									

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						CNTn_TX_0[9:0]									
						rw									

33.6.3 Receive Buffer Address Register N（USB_ADDRn_RX）

Address offset：[USB_ BUFTAB] + n×16 + 8

USB local address：[USB_ BUFTAB] + n×8 + 4

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDRn_RX[14:0]															
rw															rw

Bit Field	Name	Description
15:1	ADDRn_RX[14:0]	Receive buffer address

		Endpoint packet buffer start address for the endpoint to hold data when the next PID SETUP or OUT token packet is received
0	-	Since packet buffer memory addresses are word (32-bit) aligned, this bit must be 0

33.6.4 Receive Data Byte Number Register N（USB_CNTn_RX）

Address offset：[USB_ BUFTAB] + n×16 + 12

USB local address: [USB_ BUFTAB] + n×8 + 6

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BLSIZE	NUM_BLK[4:0]					CNTn_RX[9:0]									
rw	rw					r									

Bit Field	Name	Description
15	BLSIZE	Memory block size 0: The memory block size is 2 bytes 1: The memory block size is 32 bytes
14:10	NUM_BLK[4:0]	Number of memory blocks Records the number of memory blocks allocated to the endpoint packet receive buffer and determines the size of the endpoint packet receive buffer that is ultimately used. For details, please refer to the following Table 33-8.
9:0	CNTn_RX[9:0]	Number of bytes received Written by the USB module to record the actual number of bytes of the latest PID SETUP or OUT token packet received by the endpoint.

Note: As shown in Table 33-2 and Table 33-3 double buffered OUT endpoints and isochronous OUT endpoints require two USB_CNTn_RX registers: USB_CNTn_RX_0 and USB_CNTn_RX_1.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BLSIZE	NUM_BLK_1[4:0]					CNTn_RX_1[9:0]									
rw	rw					r									

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BLSIZE	NUM_BLK_0[4:0]					CNTn_RX_0[9:0]									
rw	rw					r									

Table 33-8 Endpoint Packet Receive Buffer Size Definition

NUM_BLK[4:0]	BLSIZE = 0	BLSIZE = 1
00000	Not allowed	32 bytes
00001	2 bytes	64 bytes
00010	4 bytes	96 bytes
00011	6 bytes	128 bytes
…	…	…
01111	30 bytes	512 bytes

NUM_BLK[4:0]	BLSIZE = 0	BLSIZE = 1
10000	32 bytes	Reserved
10001	34 bytes	Reserved
10010	36 bytes	Reserved
…	…	…
11110	60 bytes	Reserved
11111	62 bytes	Reserved

Note: The size of the endpoint packet receive buffer is defined during the device enumeration process and is defined by the wMaxPacketSize field of the standard endpoint descriptor in the USB 2.0 protocol specification.

34 Universal Serial Bus High-speed Device Dual-role Interfaces （USB_HS_DualRole）

34.1 Introduction

Universal Serial Bus High-speed Device Dual-role Interfaces（USB HS Dual Role）is USBHS for short. The USBHS controller is designed to provide a standard interface for high-speed data transfer and connecting external devices. USBHS supports Host and Device. USBHS includes an internal high-speed PHY. The PHY can be configured as high-speed or full-speed, eliminating the need for an external PHY chip. USBHS can support all four transfer types defined by the USB 2.0 protocol (control transfer, bulk transfer, interrupt transfer, and isochronous transfer). In addition, there is a DMA inside the USBHS, which can act as an AHB bus master to accelerate data transfer between the USBHS and the system.

34.2 Main Features

- Supports USB2.0 high speed(480Mb/s) /full speed(12Mb/s)/low speed(1.5Mb/s) in host mode
- Supports USB2.0 high speed(480Mb/s) /full speed(12Mb/s) in device mode
- Supports Each endpoint supports four transfer types:
  - Control transfer
  - Bulk transfer
  - Interrupt transfer
  - Isochronous transfer
- Built-in high-speed PHY
- Supports the Keep-Alive in Low-Speed mode and SOFs in High/Full-Speed modes
- The SOF pulse can be output through the PAD.
- The SOF pulse is internally connected to the TIMx.
- support for A-B device identification (ID line)
- USBHS embeds an internal DMA with thresholding support and software selectable AHB burst type in DMA mode
- It has power-saving functions, such as system stop during USB suspend, turning off clock domains internal to the digital core, PHY and DFIFO power management
- It features 4 KB of dedicated RAM
- It is up to 16 channels in host mode: each channel is dynamically reconfigurable to allocate any type of USB transfer
- Built-in hardware scheduler holding in host mode:

- Up to 16 interrupt plus isochronous transfer requests in the periodic hardware queue
- Up to 16 control plus bulk transfer requests in the non-periodic hardware queue

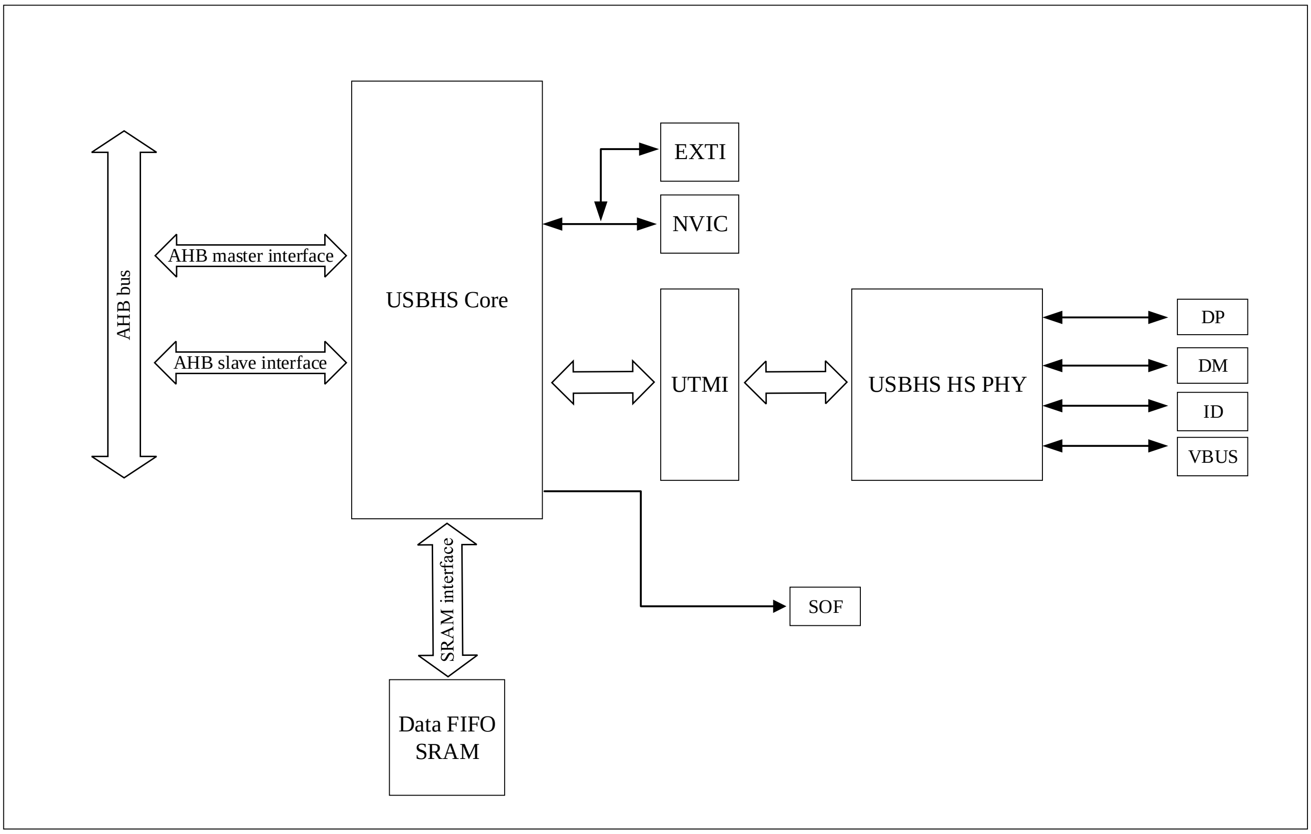
- There are Rx FIFO, a periodic Tx FIFO and a nonperiodic Tx FIFO in host mode
- Contain 1 bidirectional control endpoint0, and 8 IN/OUT endpoints configurable to support bulk, interrupt or isochronous transfers in device mode
- There are a shared Rx FIFO, a Tx-OUT FIFO and up to 9 dedicated Tx-IN FIFOs in device mode
- Support for the soft disconnect feature

Note:16MHz,19.2MHz,20MHz,24MHz,26MHz or 32MHz external crystal is required as the USBHS clock source.

34.3 Functional Description

34.3.1 USBHS System-Level Block Diagram

Figure 34-1 USBHS System-Level Block Diagram



34.3.2 USBHS Pins and Signals

Table 34-1 USBHS Input/Output Signal

Name	**Type**	Description
VBUS	**input**	power supply
DM	**input/output**	negative differential signal line
DP	**input/output**	positive differential signal line
ID	**input**	specifies whether the core is a host or device

34.3.3 Embedded High-speed PHY

The USBHS embeds high-speed PHY, supporting high-speed, full-speed, and low-speed without the need for an external PHY.Integrated pull-up resistor on the ID line for sampling, to configure the USBHS into Host mode or Device mode.The DP/DM controlled by the USBHS module have integrated pull-up and pull-down resistors, with the specific enablement depending on the current mode.

34.4 USBHS Dual-role Device

34.4.1 ID Line Detection

The host or device(the default) role is assumed depending on the ID input pin.

If the B-side of the USB cable is connected with a floating ID wire, the integrated pullup resistor detects a high ID level and the default device role is confirmed.

If the A-side of the USB cable is connected with a grounded ID, the USBHS issues an ID line status change interrupt (CIDSCHG bit in OTG_GINTSTS) for host software initialization, and automatically switches to the host role.

34.5 USB Device

This section introduces the features of USBHS in device mode. In the following scenarios, the USBHS is used as a device:

- Default state when device B is inserted into USB cable B End
- Set the forced device mode bit (USBHS_GCFG.FDMODE) in the USBHS configuration register to 1, thereby forcing the USBHS core to be only a USB device. In this case, even if the ID line is present on the USB connector, it will be ignored.

34.5.1 USB Device State

34.5.1.1 Powered state

In the powered state, the USBHS expects to receive some reset signaling from the host.
No other USB operation is possible. When a reset signaling is received the reset detected interrupt (USBHS_GINTSTS.USBRSTIF) is generated. When the reset signaling is complete, the enumeration done interrupt (USBHS_GINTSTS. ENUMDIF) is generated and the USBHS enters the Default state.

34.5.1.1.1 Default state

In the Default state the USBHS expects to receive a SET_ADDRESS command from the host. No other USB operation is possible. When a valid SET_ADDRESS command is decoded on the USB, the application writes the corresponding number into the device address field in the device configuration register (USBHS_DCFG.DEVADD[6:0]). The USBHS
then enters the address state and is ready to answer host transactions at the configured USB address.

34.5.1.1.2 Soft disconnect

The powered state can be exited by software with the soft disconnect feature. The DP pullup resistor is removed by setting the soft disconnect bit in the device control register (USBHS_DCTRL. SFTDIS), causing a device disconnect detection interrupt on the host side even though the USB cable was not really removed from the host port.

34.5.1.1.3 Suspended state

The USBHS device constantly monitors the USB activity. After counting 3 ms of USB idleness, the early suspend interrupt (USBHS_GINTSTS. ESUSPIF) is issued, and confirmed 3 ms later, if appropriate, by the suspend interrupt (USBHS_GINTSTS. USBSUSPIF). The device suspend bit is then automatically set in the device status register (USBHS_DSTS. SUSPF) and the USBHS enters the suspended state.
The suspended state may optionally be exited by the device itself. In this case the application sets the remote wakeup signaling bit in the device control register (USBHS_DCTRL. RMWKUP) and clears it after 1 to 15 ms.
When a resume signaling is detected from the host, the resume interrupt (USBHS_GINTSTS.WKUPIF) is generated and the device suspend bit is automatically cleared.

34.5.2 USB Device Endpoint

There are a total of 1 control endpoint 0, 8 IN endpoints, and 7 OUT endpoints for the device. Control endpoint 0 is bidirectional and handles control messages only. IN and OUT endpoints can be configured to support the isochronous, bulk or interrupt transfer type.

Each endpoint of the device has independent control registers, interrupt status registers, transfer size registers, and DMA address registers. The control registers and transfer size registers of endpoint 0 may slightly differ from those of other endpoints, please refer to the register description for specific details.

34.5.2.1 Endpoint control

The following endpoint controls are available to the application through the device endpoint-*x* IN/OUT control register (USBHS_DIEPxCTRL/USBHS_DOEPxCTRL):

- Endpoint enable/disable
- Endpoint activate in current configuration
- Program USB transfer type (isochronous, bulk, interrupt)
- Program supported packet size
- Program Tx FIFO number associated with the IN endpoint
- Program the expected or transmitted data0/data1 PID (bulk/interrupt only)
- Program the even/odd frame during which the transaction is received or transmitted (isochronous only)
- Optionally program the NAK bit to always negative-acknowledge the host regardless of the FIFO status
- Optionally program the STALL bit to always stall host tokens to that endpoint
- Optionally program the SNOOP mode for OUT endpoint not to check the CRC field of received data

34.5.2.2 Endpoint transfer

The device endpoint-*x* transfer size registers (USBHS_DINEPxTSIZ/USBHS_DOUTEPxTSIZ) allow the application to program the transfer size parameters and read the transfer status. Programming must be done before setting the endpoint enable bit in the endpoint control register. Once the endpoint is enabled, these fields are read-only as the USBHS core updates them with the current transfer status. The following transfer parameters can be programmed:

- Transfer size in bytes
- Number of packets that constitute the overall transfer size

34.5.2.3 Endpoint status/interrupt

The device endpoint-*x* interrupt registers (USBHS_DINEPxINT/USBHS_DOUTPEPxINT*)* indicate the status of an endpoint with respect to USB- and AHB-related events. The application must read these registers when the OUT endpoint interrupt bit or the IN endpoint interrupt bit in the core interrupt register (USBHS_GINTSTS.OUTEPIF or USBHS_GINTSTS. INEPIF) is set. Before the application can read these registers, it must first read the device all endpoints interrupt (USBHS_DAINT) register to get the exact endpoint number for the device endpoint-*x* interrupt register. The application must clear the appropriate bit in this register to clear the corresponding bits in the USBHS_DAEPINTSTS and USBHS_GINTSTS registers.

34.6 USB Host

This section gives the functional description of the USBHS in the USB host mode. The USBHS works as a USB host in the following circumstances:

- OTG A-device default state when the A-side of the USB cable is plugged in
- Set the forced host mode bit (USBHS_GCFG.FHMODE) in the USBHS configuration register to 0, thereby configuring the USBHS core in normal mode. In this case, the USB switches between host mode and device mode based on the ID line status. When the ID line is grounded, the USBHS is in host mode.

34.6.1 USB Host States

34.6.1.1 Host port power

On-chip 5 V VBUS generation is not supported. For this reason, a charge pump or, if 5 V are available on the application board, a basic power switch, must be added externally to drive the 5 V VBUS line. When the application decides to power on VBUS, it must also set the port power bit in the host port control and status register (USBHS_HPCS.PPWR).

34.6.1.2 Host detection of a peripheral connection

The USB device or B device will be detected immediately after connection. The OTG_HS core issues a host port interrupt triggered by the device connected bit in the host port control and status register (USBHS_HPCS. PCDET).

34.6.1.3 Host detection of peripheral a disconnection

The peripheral disconnection event triggers the disconnect detected interrupt (USBHS_GINTSTS. DISCIF).

34.6.1.4 Host enumeration

After detecting a device connection, if a new device is connected, the host must initiate the enumeration process by sending USB reset and configuration commands to the new device.

The application drives a USB reset signaling (single-ended zero) over the USB by keeping the port reset bit set in the host port control and status register (USBHS_HPCS. PRST) for a minimum of 10 ms and a maximum of 20 ms. The application takes care of the timing count and then of clearing the port reset bit.

Once the USB reset sequence has completed, the host port interrupt is triggered by the port enable/disable change bit (PENCHNG bit in OTG_HPRT). This informs the application that the speed of the enumerated peripheral can be read from the port speed field in the host port control and status register (PSPD bit in OTG_HPRT) and that the host is starting to drive SOFs (FS) or Keep alives (LS). The host is now ready to complete the peripheral enumeration by sending peripheral configuration commands.

34.6.1.5 Host suspend

The application decides to suspend the USB activity by setting the port suspend bit in the host port control and status register (USBHS_HPCS.PSUSP). The USBHS core stops sending SOFs and enters the suspended state.

The suspended state can be optionally exited on the remote device’s initiative (remote wakeup). In this case the remote wakeup interrupt (USBHS_GINTSTS.WKUPIF) is generated upon detection of a remote wakeup signaling, the port resume bit in the host port control and status register (USBHS_HPCS. PRST) self-sets, and resume signaling is automatically driven over the USB. The application must time the resume window and then clear the port resume bit to exit the suspended state and restart the SOF. If the suspended state is exited on the host initiative, the application must set the port resume bit to start resume signaling on the host port, time the resume window and finally clear the port resume bit.

34.6.2 Host Channels

The USBHS core instantiates 16 host channels. Each host channel supports an USB host transfer (USB pipe). The host is not able to support more than 16 transfer requests at the same time. If more than 16 transfer requests are pending from the application, the host controller driver (HCD) must re-allocate channels when they become available from previous duty, that is, after receiving the transfer completed and channel halted interrupts.

Each host channel can be configured to support in/out and any type of periodic/nonperiodic transaction. Each host channel makes us of proper control (USBHS_HCHxCTRL), transfer configuration (USBHS_HCHxSIZ) and status/interrupt (USBHS_HCHxINTSTS) registers with associated mask (USBHS_HCHxINTEN) registers.

34.6.2.1 Host channel control

The following host channel controls are available to the application through the host channel-*x* characteristics register (USBHS_HCHxCTRL):

- Channel enable/disable
- Program the HS/FS/LS speed of target USB peripheral
- Program the address of target USB peripheral
- Program the endpoint number of target USB peripheral
- Program the transfer IN/OUT direction
- Program the USB transfer type (control, bulk, interrupt, isochronous)
- Program the maximum packet size (MPS)
- Program the periodic transfer to be executed during odd/even frames

34.6.2.2 Host channel transfer

The host channel transfer size registers (USBHS_HCHxSIZ) allow the application to program the transfer size parameters, and read the transfer status. Programming must be done before setting the channel enable bit in the host channel characteristics register. Once the endpoint is enabled the packet count field is read-only as the USBHS core updates it according to the current transfer status.

The following transfer parameters can be programmed:

- transfer size in bytes
- number of packets making up the overall transfer size
- initial data PID

34.6.2.3 Host channel status/interrupt

The host channel-x interrupt register (USBHS_HCHxINTSTS) indicates the status of an endpoint with respect to USB- and AHB-related events. The application must read these register when the host channels interrupt bit in the core interrupt register (USBHS_GINTSTS. HCHIF) is set. Before the application can read these registers, it must first read the host all channels interrupt (USBHS_HACHINT) register to get the exact channel number for the host channel-x interrupt register. The application must clear the appropriate bit in this register to clear the corresponding bits in the USBHS_HACHINT and USBHS_GINTSTS registers. The mask bits for each interrupt source of each channel are also available in the USBHS_HCHxINTEN register.

The host core provides the following status checks and interrupt generation:

– Transfer completed interrupt, indicating that the data transfer is complete on both the application (AHB) and USB sides

- Channel has stopped due to transfer completed, USB transaction error or disable
- command from the application
- Associated transmit FIFO is half or completely empty (IN endpoints)
- ACK response received
- NAK response received
- STALL response received
- USB transaction error due to CRC failure, timeout, bit stuff error, false EOP

- Babble error
- frame overrun
- data toggle error

34.6.3 Host scheduler

The host core features a built-in hardware scheduler which is able to autonomously re-order and manage the USB transaction requests posted by the application. At the beginning of each frame the host executes the periodic (isochronous and interrupt) transactions first, followed by the nonperiodic (control and bulk) transactions to achieve the higher level of priority granted to the isochronous and interrupt transfer types by the USB specification.

The host processes the USB transactions through request queues (one for periodic and one for nonperiodic). Each request queue can hold up to 8 entries. Each entry represents a pending transaction request from the application, and holds the IN or OUT channel number along with other information to perform a transaction on the USB. The order in which the requests are written to the queue determines the sequence of the transactions on the USB interface.

At the beginning of each frame, the host processes the periodic request queue first, followed by the nonperiodic request queue. The host issues an incomplete periodic transfer interrupt (USBHS_GINTSTS.PTNCIF) if an isochronous or interrupt transaction scheduled for the
current frame is still pending at the end of the current frame. The USBHS core is fully responsible for the management of the periodic and nonperiodic request queues.The periodic transmit FIFO and queue status register (USBHS_HPTXFQSTS) and nonperiodic transmit FIFO and queue status register (USBHS_GNPTXFSTS) are read-only registers which can be used by the application to read the status of each request queue.
They contain:

- The number of free entries currently available in the periodic (nonperiodic) request queue (8 max)
- Free space currently available in the periodic (nonperiodic) Tx FIFO (out-transactions)
- IN/OUT token, host channel number and other status information.

As request queues can hold a maximum of 8 entries each, the application can push to schedule host transactions in advance with respect to the moment they physically reach the SB for a maximum of 8 pending periodic transactions plus 8 pending non-periodic transactions.
To post a transaction request to the host scheduler (queue) the application must check that there is at least 1 entry available in the periodic (nonperiodic) request queue by reading the USBHS_HPTXFQSTS.PTXRQSAVL[6:0] or USBHS_GNPTXFSTS. NPTXRQSAV[7:0].

34.7 SOF Trigger

The USBHS core provides means to monitor, track and configure SOF framing in the host and peripheral, as well as an SOF pulse output connectivity feature.

34.7.1 Host SOFs

In host mode the number of PHY clocks occurring between the generation of two consecutive SOF (HS/FS) or Keep-alive (LS) tokens is programmable in the host frame interval register (HFIR), thus providing application

control over the SOF framing period. An interrupt is generated at any start of frame (USBHS_GINTSTS.SOFIF). The current frame number and the time remaining until the next SOF are tracked in the host frame number register (USBHS_HFNUM).
A SOF pulse signal, is generated at any SOF starting token and with a width of 20 HCLK cycles. The SOF pulse is also internally connected to the input trigger of the timer, so that the input capture feature, the output compare feature and the timer can be triggered by the SOF pulse.

34.7.2 Device SOFs

In device mode, the start of frame interrupt is generated each time an SOF token is received on the USB (OTH_GINTSTS.SOFIF). The corresponding frame number can be read from the device status register (USBHS_DSTS.SOFFN[13:0]). A SOF pulse signal with a width of 12 system-clock cycles is also generated.The SOF pulse signal is also internally connected to the TIM input trigger, so that the input capture feature, the output compare feature and the timer can be triggered by the SOF pulse.

The end of periodic frame interrupt (USBHS_GINTSTS. EOPFIF) is used to notify the application when 80%, 85%, 90% or 95% of the time frame interval elapsed depending on the periodic frame interval field in the device configuration register (USBHS_DCFG. PFRITVL[1:0]). This feature can be used to determine if all of the isochronous traffic for that frame is complete.

34.7.3 Dynamic Update of The USBHS_HFRI Register

The USBHS core embeds a dynamic trimming capability of micro-SOF framing period in host mode allowing to synchronize an external device with the micro-SOF frames. When the USBHS_HFRI register is changed within a current micro-SOF frame, the SOF period correction is applied in the next frame

34.8 Power Options

The power consumption of the USBHS PHY is controlled by two bits in the General Module Configuration Register:
The power consumption of the USBHS PHY is controlled by two bits in the general core configuration register:

- Stop PHY clock（USBHS_PWRCTRL. PHYSTP）

  Setting the stop PHY clock bit in the clock gating control register to 1 will save power by reducing the dynamic power consumption caused by clock signal toggling in the module. Most of the transceiver is also disabled, and only the part in charge of detecting the asynchronous resume or remote wakeup event is kept alive.

- Gate HCLK （USBHS_PWRCTRL.GATEHCLK）

  When setting the Gate HCLK bit in the clock gating control register, most of the system clock domain internal to the USBHS core is switched off by clock gating. Only the register read and write interface is kept alive. The dynamic power consumption due to the USB clock switching activity is cut even if the system clock is kept running by the application for other purposes.

- USB system stop

  When the USBHS is in the USB suspended state, the application may decide to drastically reduce the overall power consumption by a complete shut down of all the clock sources in the system. USB System Stop is activated by first setting the Stop PHY clock bit and then configuring the system deep sleep mode in the power control system module (PWR).

  The USBHS core automatically reactivates both system and USB clocks by asynchronous detection of remote wakeup (as an host) or resume (as a device) signaling on the USB.

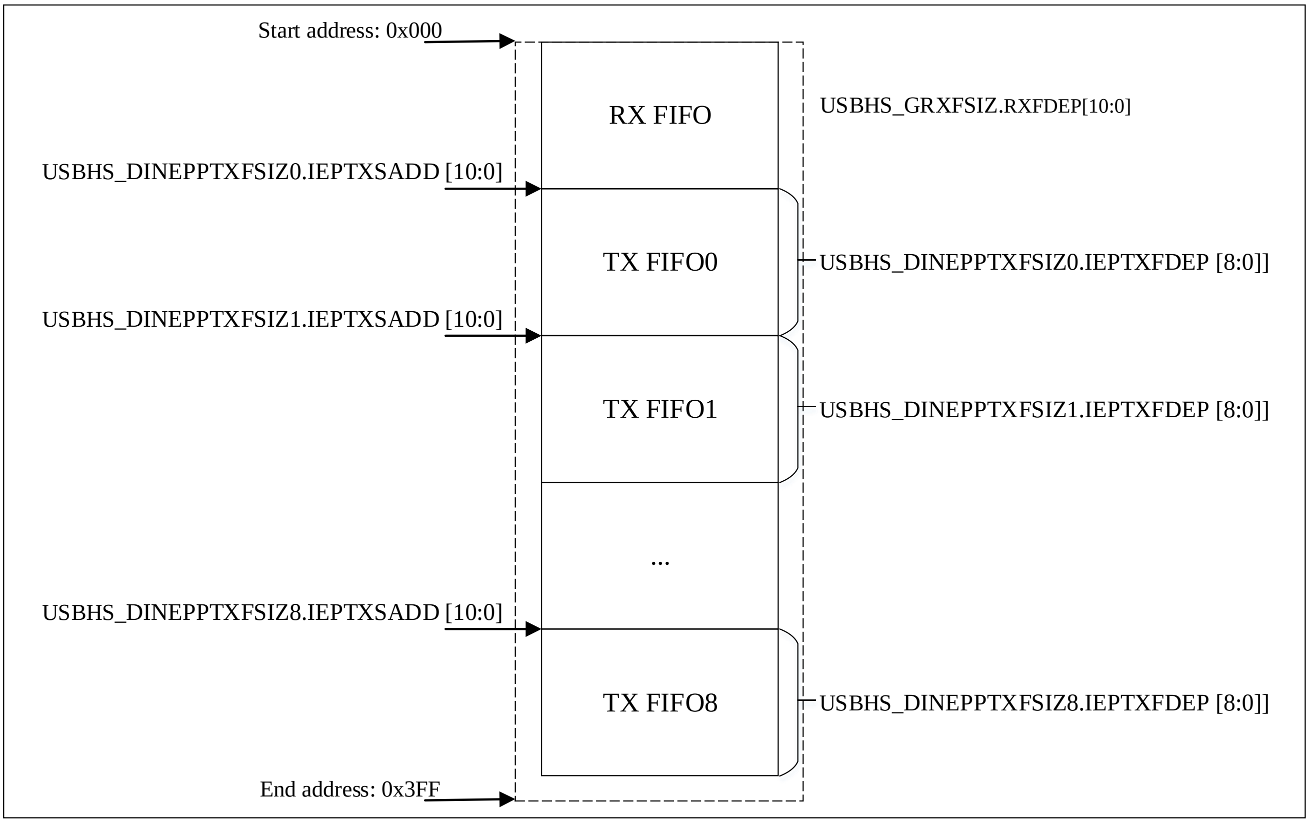
  To save dynamic power, the USB data FIFO is clocked only when accessed by the OTG_HS core.

34.9 USBHS Data FIFO

The USBHS system features 4 Kbytes of dedicated RAM with a sophisticated FIFO control mechanism. The packet FIFO controller module in the OTG_HS core organizes RAM space into Tx FIFOs into which the application pushes the data to be temporarily stored before the USB transmission, and into a single Rx FIFO where the data received from the USB are temporarily stored before retrieval (popped) by the application. The number of instructed FIFOs and how these are organized inside the RAM depends on the device’s role. In peripheral mode an additional Tx FIFO is instructed for each active IN endpoint. Any FIFO size is software configured to better meet the application requirements.

34.9.1 Device FIFO Architecture

Figure34-2 Device-mode FIFO Address Mapping and AHB FIFO Access Mapping



Start address: 0x000
RX FIFO
USBHS_GRXFSIZ.RXFDEP[10:0]
USBHS_DINEPPTXFSIZ0.IEPTXSADD [10:0]
TX FIFO0
USBHS_DINEPPTXFSIZ0.IEPTXFDEP [8:0]]
USBHS_DINEPPTXFSIZ1.IEPTXSADD [10:0]
TX FIFO1
USBHS_DINEPPTXFSIZ1.IEPTXFDEP [8:0]]
...
USBHS_DINEPPTXFSIZ8.IEPTXSADD [10:0]
TX FIFO8
USBHS_DINEPPTXFSIZ8.IEPTXFDEP [8:0]]
End address: 0x3FF

34.9.1.1 Device Rx FIFO

The USBHS peripheral uses a single receive FIFO that receives the data directed to all OUT endpoints. Received packets are stacked back-to-back until free space is available in the Rx FIFO. The status of the received packet (which contains the OUT endpoint destination number, the byte count, the data PID and the validity of the received data) is also stored by the core on top of the data payload. When no more space is available, host transactions are NACKed and an interrupt is received on the addressed endpoint. The size of the receive FIFO is configured in the receive FIFO size register (USBHS_GRXFSIZ).

The single receive FIFO architecture makes it more efficient for the USB peripheral to fill in the receive RAM buffer:

- All OUT endpoints share the same RAM buffer (shared FIFO)
- The OTG_HS core can fill in the receive FIFO up to the limit for any host sequence of OUT tokens

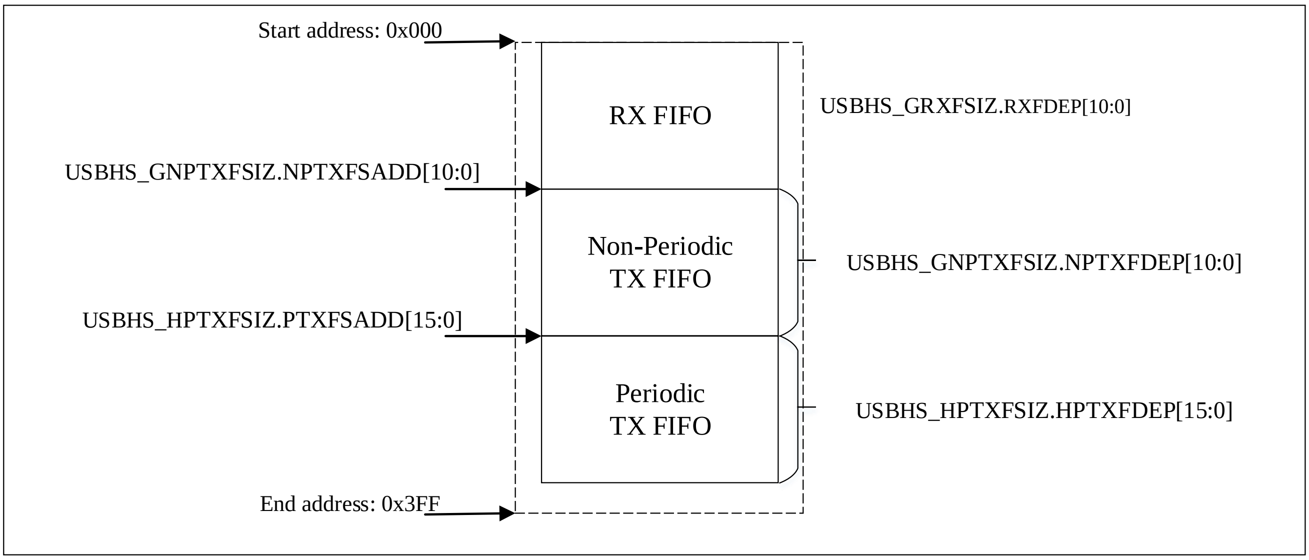
The application keeps receiving the Rx FIFO non-empty interrupt (USBHS_GINTSTS. RXFNEIF) as long as there is at least one packet available for download. It reads the packet information from the receive status read and pop register (USBHS_GRXSTSP) and finally pops data off the receive FIFO by reading from the endpoint-related pop address.

34.9.1.2 Device Tx FIFO

The core has a dedicated FIFO for each IN endpoint. The application configures FIFO sizes by writing the endpoint 0 transmit FIFO size register (USBHS_DIEP0TXFSIZ) for IN endpoint0 and the device IN endpoint transmit FIFOx registers (USBHS_DIEPxTXFSIZ) for IN endpoint-x.

34.9.2 Host FIFO architecture

Figure34-3 Host-mode FIFO Address Mapping and AHB FIFO Access Mapping


Start address: 0x000
RX FIFO
USBHS_GRXFSIZ.RXFDEP[10:0]
USBHS_GNPTXFSIZ.NPTXFSADD[10:0]
Non-Periodic
TX FIFO
USBHS_GNPTXFSIZ.NPTXFDEP[10:0]
USBHS_HPTXFSIZ.PTXFSADD[15:0]
Periodic
TX FIFO
USBHS_HPTXFSIZ.HPTXFDEP[15:0]
End address: 0x3FF

34.9.2.1 Host Rx FIFO

The host uses one receiver FIFO for all periodic and nonperiodic transactions. The FIFO is used as a receive buffer to hold the received data (payload of the received packet) from the USB until it is transferred to the system memory. Packets received from any remote IN endpoint are stacked back-to-back until free space is available. The status of each received packet with the host channel destination, byte count, data PID and validity of the received data are also stored into the FIFO. The size of the receive FIFO is configured in the receive FIFO size register (USBHS_GRXFSIZ).

The single receive FIFO architecture makes it highly efficient for the USB host to fill in the receive data buffer:

- All IN configured host channels share the same RAM buffer (shared FIFO)
- The USBHS core can fill in the receive FIFO up to the limit for any sequence of IN tokens driven by the host software The application receives the Rx FIFO not-empty interrupt as long as there is at least one packet available for download. It reads the packet information from the receive status read and pop register and finally pops the data off the receive FIFO.

34.9.2.2 Host Tx FIFO

The host uses one transmit FIFO for all non-periodic (control and bulk) OUT transactions and one transmit FIFO for all periodic (isochronous and interrupt) OUT transactions. FIFOs are used as transmit buffers to hold the data (payload of the transmit packet) to be transmitted over the USB. The size of the periodic (nonperiodic) Tx FIFO is configured in the host periodic (nonperiodic) transmit FIFO size (USBHS_HPTXFSIZ/USBHS_GNPTXFSIZ) register.

The two Tx FIFO implementation derives from the higher priority granted to the periodic type of traffic over the USB frame. At the beginning of each frame, the built-in host scheduler processes the periodic request queue first, followed by the nonperiodic request queue.

The two transmit FIFO architecture provides the USB host with separate optimization for periodic and nonperiodic transmit data buffer management:

- All host channels configured to support periodic (nonperiodic) transactions in the OUT direction share the same RAM buffer (shared FIFOs)
- The USBHS core can fill in the periodic (nonperiodic) transmit FIFO up to the limit for any sequence of OUT tokens driven by the host software

The USBHS core issues the periodic Tx FIFO empty interrupt (USBHS_GINTSTS.PTXFEI) as long as the periodic Tx FIFO is half or completely empty, depending on the value of the periodic Tx FIFO empty level bit in the AHB configuration register USBHS_GAHBCFG.PTXFETH). The application can push the transmission data in advance as long as free space is available in both the periodic Tx FIFO and the periodic request queue. The host periodic transmit FIFO and queue status register (USBHS_HPTXFQSTS) can be read to know how much space is available in both.

The USBHS core issues the non periodic Tx FIFO empty interrupt (USBHS_GINTSTS.NPTXFEIF) as long as the nonperiodic Tx FIFO is half or completely empty depending on the non periodic Tx FIFO empty level bit in the AHB configuration register (USBHS_GAHBCFG. NPTXFETH). The application can push the transmission data as long as free space is available in both the nonperiodic Tx FIFO and nonperiodic request queue. The host nonperiodic transmit FIFO and queue status register (USBHS_GNPTXFSTS) can be read to know how much space is available in both.

34.9.3 FIFO RAM Allocation

34.9.3.1 Device mode

Receive FIFO RAM allocation: the application should allocate RAM for SETUP packets:

- 10 locations must be reserved in the receive FIFO to receive SETUP packets on control endpoint. The core does not use these locations, which are reserved for SETUP packets, to write any other data.
- One location is to be allocated for Global OUT NAK.
- Status information is written to the FIFO along with each received packet. Therefore, a minimum space of (largest packet size / 4) + 1 must be allocated to receive packets. If multiple isochronous endpoints are enabled, then at least two (largest packet size / 4) + 1 spaces must be allocated to receive back-to-back packets. Typically, two (largest packet size / 4) + 1 spaces are recommended so that when the previous packet is being transferred to the CPU, the USB can receive the subsequent packet.
- Along with the last packet for each endpoint, transfer complete status information is also pushed to the FIFO. One location for each OUT endpoint is recommended.

Device RxFIFO = (5 * number of control endpoints + 8) + ((largest USB packet used / 4) + 1 for status information) + (2 * number of OUT endpoints) + 1 for Global NAK

Example: The MPS is 1,024 bytes for a periodic USB packet and 512 bytes for a nonperiodic USB packet. There are three OUT endpoints, three IN endpoints, one control endpoint, and three host channels.

Device RxFIFO = (5 * 1 + 8) + ((1,024 / 4) +1) + (2 * 4) + 1 = 279

Transmit FIFO RAM allocation: the minimum RAM space required for each IN endpoint. Transmit FIFO is the maximum packet size for that particular IN endpoint.

Note: More space allocated in the transmit IN endpoint FIFO results in better performance on the USB.

34.9.3.2 Host mode

Receive FIFO RAM allocation:

Status information is written to the FIFO along with each received packet. Therefore, a minimum space of (largest packet size / 4) + 1 must be allocated to receive packets. If multiple isochronous channels are enabled, then at least two (largest packet size / 4) + 1 spaces must be allocated to receive back-to-back packets. Typically, two (largest packet size / 4) + 1 spaces are recommended so that when the previous packet is being transferred to the CPU, the USB can receive the subsequent packet.

Along with the last packet in the host channel, transfer complete status information is also pushed to the FIFO. So one location must be allocated for this.

Host RxFIFO = (largest USB packet used / 4) + 1 for status information + 1 transfer complete

Example: Host RxFIFO = ((1,024 / 4) + 1) + 1 = 258

Transmit FIFO RAM allocation:

The minimum amount of RAM required for the host Non-periodic Transmit FIFO is the largest maximum packet size among all supported non-periodic OUT channels.

Typically, two largest packet sizes worth of space is recommended, so that when the current packet is under transfer to the USB, the CPU can get the next packet.

Non-Periodic TxFIFO = largest non-periodic USB packet used / 4

Example: Non-Periodic TxFIFO = (512 / 4) = 128

The minimum amount of RAM required for host periodic Transmit FIFO is the largest maximum packet size out of all the supported periodic OUT channels. If there is at least one isochronous OUT endpoint, then the space must be at least two times the maximum packet size of that channel.

Host Periodic TxFIFO = largest periodic USB packet used / 4

Example: Host Periodic TxFIFO = (1,024 / 4) = 256

Note: More space allocated in the Transmit Non-periodic FIFO results in better performance on the USB.

34.10 USBHS Programming Process

34.10.1 Module Initialization

The application must perform the core initialization sequence. If the cable is connected during power-up, the current mode of operation bit in the USBHS_GINTSTS (USBHS_GINTSTS. CMODE) reflects the mode. The USBHS controller enters host mode when an “A” plug is connected or device mode when a “B” plug is connected.

This section explains the initialization of the USBHS controller after power-on. The application must follow the initialization sequence irrespective of host or device mode operation. All core global registers are initialized according to the core’s configuration:

1. Program the following fields in the USBHS_GAHBCFG register:
   - Global interrupt mask bit GINTMSK GINTEN = 1
   - RX FIFO non-empty（USBHS_GINTSTS.RXFNEIF）
   - Periodic Tx FIFO empty level
2. Program the following fields in the USBHS_GCFG register:
   - USBHS timeout calibration field
   - USB turnaround time field
3. The software must enable the following bits in the USBHS_GINTEN register:
   - USBHS enable interrupt
4. The software can read the USBHS_GINTSTS.CMODE to determine whether the USBHS controller is operating in host or device mode.

34.10.2 Host initialization

To initialize the core as host, the application must perform the following steps:

1. Program the USBHS_GINTEN. HPIEN register to unmask.
2. Program the USBHS_HCFG register to select host mode (high speed, full speed, low speed).
3. Program the USBHS_HPCS. PPWR to 1. This drives VBUS on the USB.
4. Wait for the USBHS_HPCS. PCDET. This indicates that a device is connecting to the port.
5. Program the USBHS_HPCS.PRST to 1. This starts the reset process.
6. Wait at least 10 ms for the reset process to complete.
7. Program the USBHS_HPCS. PRST to 0.
8. Wait for the PENCHNG interrupt in the USBHS_HPCS .PENC register.
9. Read the PSPD bit in USBHS_HPRT to get the enumerated speed.
10. Program the USBHS_HFRI register with a value corresponding to the selected PHY clock 1
11. Program the SPSEL field in the USBHS_HCFG register following the speed of the device detected in step 9.
12. Program the USBHS_GRXFSIZ register to select the size of the receive FIFO.
13. Program the USBHS_GNPTXFSIZ register to select the size and the start address of the Non-periodic transmit FIFO for non-periodic transactions.
14. Program the USBHS_HPTXFSIZ register to select the size and start address of the periodic transmit FIFO for periodic transactions.

To communicate with devices, the system software must initialize and enable at least one channel.

34.10.3 Device initialization

The application must perform the following steps to initialize the core as a device on powerup or after a mode change from host to device.

1. Program the following fields in the USBHS_DCFG register:
   - Device speed
   - Non-zero-length status OUT handshake
2. Program the USBHS_ GINTEN register to unmask the following interrupts:
   - USB reset
   - Enumeration done
   - Early suspend
   - USB suspend
   - SOF
3. Wait for the USBRSTIF interrupt in USBHS_GINTSTS. It indicates that a reset has been detected on the USB that lasts for about 10 ms on receiving this interrupt.
4. Wait for the ENUMDIF interrupt in USBHS_GINTSTS. This interrupt indicates the end of reset on the USB. On receiving this interrupt, the application must read the USBHS_DSTS register to determine the enumeration speed and perform the steps.

At this point, the device is ready to accept SOF packets and perform control transfers on control endpoint 0.

34.11 USBHS Registers

These registers are available in both host mode and device mode, and do not need to be reprogrammed when switching between these two modes. Unless otherwise specified, the bit values in the register descriptions are represented in binary.

USB base address: 0x4004 0000

34.11.1 USBHS Register Overview

34.11.2 USBHS global control and status registers

34.11.2.1 USBHSglobal control and status registers（USBHS_GCTRLSTS）

Address offset：0x0000

Reset value: 0x000D 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved										CMODE	Reserved				IDSTS
										r					r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												VBVAL OVAL	VBVAL OVEN	Reserved	
												rw	rw		

Bit Field	Name	Description
31:22	Reserved	Reserved, must be kept at reset value
21	CMODE	Current mode of operation 0：Device mode 1：Host mode *Note: Indicates the current mode (host or device).*
20:17	Reserved	Reserved, must be kept at reset value
16	IDSTS	Connector ID status 0: The USBHS controller is in A-device mode 1: The USBHS controller is in B-device mode
15:4	Reserved	Reserved, must be kept at reset value
3	VBVALOVAL	VBUS valid override value. This bit is used to set override value for vbusvalid signal when VBVALOEN bit is set. 0: vbusvalid value is '0' when VBVALOEN = 1 1: vbusvalid value is '1' when VBVALOEN = 1 Note: Only accessible in host mode.
2	VBVALOVEN	VBUS valid override enable.

Bit Field	Name	Description
		This bit is used to enable/disable the software to override the vbusvalid signal using the VBVALOVAL bit. 0: Override is disabled and vbusvalid signal from the respective PHY selected is used internally by the core 1: Internally vbusvalid received from the PHY is overridden with VBVALOVAL bit value Note: Only accessible in host mode
1:0	Reserved	Reserved, must be kept at reset value

34.11.2.2 USBHS Global AHB Cnfiguration Register（USBHS_GAHBCFG）

Address offset：0x0008

Reset value : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved							PTXFETH	NPTXGETH	Reserved	DMAEN	BURSTTYP[3:0]				GINTEN
							rw	rw		rw	rw				rw

Bit Field	Name	Description
31:9	Reserved	Reserved, must be kept at reset value
8	PTXFETH	Periodic Tx FIFO empty level 0: PTXFEIF (in USBHS_GINTSTS) interrupt indicates that the Periodic Tx FIFO is half empty 1: PTXFEIF (in USBHS_GINTSTS) interrupt indicates that the Periodic Tx FIFO is completely empty *Note: Only accessible in host mode.*
7	NPTXFETH	Nonperiodic Tx FIFO empty level Host mode: 0: The NPTXFEIF (in USBHS_GINTSTS) interrupt indicates that the nonperiodic Tx FIFO is half empty 1: The NPTXFEIF (in USBHS_GINTSTS) interrupt indicates that the nonperiodic Tx FIFO is completely empty Device mode: 0:The TXFE (in USBHS_DINTPxINTSTS) interrupt indicates that the IN endpoint Tx FIFO is half empty 1: The TXFE (in USBHS_DINTPxINTSTS) interrupt indicates that the IN endpoint Tx FIFO is

Bit Field	Name	Description
		completely empty *Note: Accessible in host and device mode.*
6	Reserved	Reserved, must be kept at reset value
5	DMAEN	DMA enabled 0：DMA disabled 1：DMA enabled *Note: Accessible in host and device mode.*
4:1	BURSTTYP [3:0]	Burst bype 0000：single 0001：INCR 0011：INCR4 0101：INCR8 0111：INCR16 Others: Reserved *Note: Accessible in host and device mode.*
0	GINTEN	Global interrupt enable 0：disabled 1：enabled *Note: Accessible in host and device mode.*

34.11.2.3 USBHS Global Configuration Register（USBHS_GCFG）

Address offset：0x000C

Reset value：0x0000 1400

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved	FDMODE	FHMODE	Reserved												
	rw	rw													

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		TRDTIM[3:0]				Reserved			PHYSEL	Reserved		PHYIF	TOCAL[2:0]		
		rw							rw			rw	rw		

Bit Field	Name	Description
31	Reserved	Reserved, must be kept at reset value
30	FDMODE	Force device mode Writing a 1 to this bit, forces the core to device mode irrespective of the OTG_ID input pin. 0: Normal mode 1: Force device mode After setting the force bit, the application must wait at least 25 ms before the change takes effect.

Bit Field	Name	Description
		Note: Accessible in device and host modes.
29	FHMODE	Force host mode Writing a 1 to this bit, forces the core to host mode irrespective of the OTG_ID input pin. 0: Normal mode 1: Force host mode After setting the force bit, the application must wait at least 25 ms before the change takes effect. Note: Accessible in device and host modes.
28:14	Reserved	Reserved, must be kept at reset value
13:10	TRDTIM[3:0]	USB turnaround time These bits allows to set the turnaround time in PHY clocks. They must be configured according to or Table 34-2: TRDT values (HS), depending on the application AHB frequency. Higher TRDT values allow stretching the USB response time to IN tokens in order to compensate for longer AHB read access latency to the data FIFO. Note: Only accessible in device mode.
9:7	Reserved	Reserved, must be kept at reset value
6	PHYSEL	PHY select 0: USB 2.0 external ULPI high-speed PHY. 1: USB 1.1 full-speed serial transceiver. Note: Accessible in device and host modes.
5:4	Reserved	Reserved, must be kept at reset value
3	PHYIF	PHY type 0：8bit 1：16bit Note: Accessible in device and host modes.
2:0	TOCAL[2:0]	FS timeout calibration The additional delay introduced by the PHY includes the number of PHY clock cycles set in this field by the application, as well as the inter-packet gap between high-speed/full-speed data packets in the module. The impact of the delay introduced by different PHYs on the data line status varies. The application can use TOC[2:0] to increase this value (in terms of PHY clocks). In the USB standard, the timeout for high-speed operation is 736 to 816 bit times, and for full-speed operation, it is 16 to 18 bit times. The number of bit times added per PHY clock is as follows: High speed: 30MHz PHY clock = 16bit times 60MHz PHY clock = 8bit times Full speed: 30MHz PHY clock = 0.4bit times 60MHz PHY clock = 0.2bit times

Bit Field	Name	Description
		000：increase 0 PHY clock 001：increase 1 PHY clock … 111：increase 7 PHY clock Note: Accessible in device and host modes.

Table 34-2 TRDTIM Values（FS）

AHB Frequency Range (MHz)		TRDTIM Minimum Value
Min	Max	
14.2	15	0xF
15	16	0xE
16	17.2	0xD
17.2	18.5	0xC
18.5	20	0xB
20	21.8	0xA
21.8	24	0x9
24	27.5	0x8
27.5	32	0x7
32	-	0x6

Table 34-3 TRDTIM Values（HS）

AHB Frequency Range (MHz)		TRDTIM Minimum Value
Min	Max	
30	15	0x9

34.11.2.4 USBHS Global Reset Register（USBHS_GRSTCTRL）

Address offset：0x0010

Reset value：0x8000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AHBIDLE	DMAREQ	SRSTDNE	Reserved												
r	r	rc_w1													

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved					TXFNUM[4:0]					TXFFLSH	RXFFLSH	Reserved	HFCRST	PFSSRST	CSRST
							rw			rs	rs		rs	rs	rw

Bit Field	Name	Description
31	AHBIDLE	AHB idle state 0：AHB is busy

Bit Field	Name	Description
		1：AHB is idle Note: Accessible in device and host modes.
30	DMAREQ	DMA request 0：disable DMA 1：enable DMA Note: Accessible in device and host modes.
29	SRSTDNE	Core Soft Reset Done 0： No reset 1： Core Soft Reset is done Note: Accessible in device and host modes.
28:11	Reserved	Reserved, must be kept at reset value
10:6	TXFNUM[4:0]	TX FIFO Number This is the FIFO number that must be flushed using the TxFIFO Flush bit. This field must not be changed until the core clears the TxFIFO Flush bit. Host mode: 00000：Non-periodic TxFIFO flush 00001：Periodic TxFIFO flush 10000：Flush all the transmit FIFOs Device mode： 00000：Periodic TxFIFO flush 00001：Periodic TxFIF1 flush … 01111：Periodic TxFIF15 flush 10000：Flush all the transmit FIFOs
5	TXFFLSH	TX FIFO Flush The application uses this bit to refresh the TX FIFO data, with TXFNUM[4:0] determining the FIFO number to be refreshed. However, this operation cannot be executed while the module is processing communication transactions. Once the refresh is complete, the hardware automatically clears this bit. After setting this bit, the application needs to wait for it to be cleared, which takes eight clocks to reset (using the slower PHY_CLK or HCLK clock), and no other operations should be performed by the USBHS before this. Read-NAK valid interrupt ensures that the module has not performed a read operation on the FIFO. Write-AHBIDLE ensures that the module has not performed a write operation on the FIFO. It is recommended to refresh the TX FIFO when reconfiguring it, and to refresh the TX FIFO during device endpoint disablement. Note: Accessible in device and host modes.
4	RXFFLSH	RX FIFO Flush The application refreshes the RX FIFO data by setting this control bit. However, this operation cannot be executed while the module is processing communication

Bit Field	Name	Description
		transactions. Once the refresh is complete, the hardware automatically clears this bit. After setting this bit, the application needs to wait for it to be cleared, which takes eight clocks to reset (using the slower PHY_CLK or HCLK clock), and no other operations should be performed by the USBHS before this. Note: Accessible in device and host modes.
3	Reserved	Reserved, must be kept at reset value
2	HFCRST	Host Frame Counter Reset The application writes this bit to reset the (micro)Frame number counter inside the USBHS. When the (micro)Frame counter is reset, the subsequent SOF sent out by the core has a (micro)Frame number of 0. After the reset operation is completed, the hardware automatically clears this bit. The application needs to wait for this bit to be cleared, and no other operations should be performed by the USBHS before this. Note: Accessible in host modes.
1	PFSSRST	PIU FS Dedicated Controller Soft Reset All module state machines in FS Dedicated Controller of PIU are reset to the IDLE state. Used to reset the FS Dedicated controller in PIU in case of any PHY Errors like Loss of activity or Babble Error resulting in the PHY remaining in RX state for more than one frame boundary. Note: Accessible in device and host modes.
0	CSRST	USB Core Soft Reset Reset AHB and USB clock domain circuits, as well as most of the registers. The following registers or register bits will not be cleared: USBHS_GCFG USBHS_PWRCLKCTRL.PDMRST USBHS_PWRCLKCTRL.GATECLK USBHS_PWRCLKCTRL.PHYSTP Note: Accessible in device and host modes.

34.11.2.5 USBHS Global Interrupt Status Register（USBHS_GINTSTS）

Address offset：0x0014

Reset value：0x0400 0020

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
WKUPIF	Reserved	DISCIF	IDSTSCIF	Reserved	PTXFEIF	HCHIF	HPIF	RSTDIF	FET SUSPIF	PTNCIF ISOUTNCIF	ISOINCIF	OUTEPIF	INEPIF	Reserved	
rc_w1		rc_w1	rc_w1		r	r	r	rc_w1	rc_w1	rc_w1	rc_w1	r	r		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EOPFIF	ISOUT PDIF	ENUMDIF	USB RSTIF	USB SUSPIF	ESUSPIF	Reserved		GOUT NAKEIF	GINNP NAKEIF	NPTXF EIF	RXFNEIF	SOFIF	Reserved	MOD MISIF	CMODE
rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1			r	r	r	r	rc_w1		rc_w1	r

Bit Field	Name	Description
31	WKUPIF	Resume/Remote Wakeup Detected Interrupt flag： Detected the host sending a resume signal, this flag is set. Host mode： Detected the device sending a remote wakeup signal, this flag is set. Note: Accessible in device and host modes.
30	Reserved	Reserved, must be kept at reset value
29	DISCIF	Disconnect Detected Interrupt flag Asserted when a device disconnect is detected. Note: Accessible in host modes.
28	IDSTSCIF	Connector ID Status Change The core sets this bit when there is a change in connector ID status. Note: Accessible in device and host modes.
27	Reserved	Reserved, must be kept at reset value
26	PTXFEIF	Periodic TxFIFO Empty Interrupt flag When the periodic TX FIFO is half-empty or completely empty, and there is at least one entry to be written in the periodic request queue, this flag is set. The TX FIFO empty threshold is determined by USBHS_GAHBCFG.PTXFETH. Note: Accessible in host modes.
25	HCHIF	Host Channels Interrupt flag When an interrupt is suspended on any channel in host mode, this flag is set. Software needs to first read the USBHS_HACHINT register to obtain the corresponding channel number, and then read the corresponding USBHS_HCHxINTSTS register to obtain the specific flag that caused the interrupt on the corresponding channel. The corresponding channel interrupt flag can be cleared by the USBHS_HCHxINTSTS register, and after the interrupt flag of the corresponding channel is cleared, the flag will be automatically cleared. Note: Accessible in host modes.
24	HPIF	Host Port Interrupt flag When the USBHS detects a change in the port status in host mode, this flag is set. Software needs to first read the USBHS_HPCS register to obtain the interrupt source. The corresponding interrupt flag can be cleared by the USBHS_HPCS register, and after the flag causing the port interrupt is cleared, the flag will be automatically cleared. Note: Accessible in host modes.
23	RSTDIF	Reset detected Interrupt flag When the device is in suspend mode and a reset is detected, this flag is set. Note: Accessible in device modes.
22	FETSUSPIF	Data Fetch Suspended Interrupt flag This interrupt is only valid in DMA mode. This interrupt indicates that the module has stopped acquiring data for the IN endpoint due to unavailable space in the TX FIFO or request queue. This interrupt is used when the application

Bit Field	Name	Description
		endpoint does not match. For example, after detecting an endpoint mismatch, the application will perform the following actions: Sets a Global non-periodic IN NAK handshake ■ Disables IN endpoints ■ Flushes the FIFO ■ Determines the token sequence from the IN Token Sequence Learning Queue ■ Re-enables the endpoints ■ Clears the Global non-periodic IN NAK handshake If the Global non-periodic IN NAK is cleared, the core has not yet fetched data for the IN endpoint, and the IN token is received. The core generates an 'IN token received when FIFO empty' interrupt. It then sends the host a NAK response. To avoid this scenario, the application can check the GINTSTS.FetSusp interrupt, which ensures that the FIFO is full before clearing a Global NAK handshake. Alternatively, the application can mask the IN token received when FIFO empty interrupt when clearing a Global IN NAK handshake.Note: Accessible in device modes.
21	PTNCIF	Incomplete Periodic Transfer Interrupt flag In Host mode, the core sets this interrupt bit when there are incomplete periodic transactions still pending which are scheduled for the current microframe.
	ISOUTNCIF	Incomplete Isochronous OUT Transfer Interrupt flag The Device mode, the core sets this interrupt to indicate that there is at least one isochronous OUT endpoint on which the transfer is not completed in the current microframe. This interrupt is asserted along with the End of Periodic Frame Interrupt (EOPF) bit in this register.
20	ISOINCIF	Incomplete Isochronous IN Transfer Interrupt flag Incomplete Isochronous IN Transfer (incompISOIN) The core sets this interrupt to indicate that there is at least one isochronous IN endpoint on which the transfer is not completed in the current microframe. This interrupt is asserted along with the End of Periodic Frame Interrupt (EOPF) bit in this register Accessible in device modes.
19	OUTEPIF	OUT Endpoints Interrupt flag When an interrupt is suspended on any OUT endpoint, this flag is set. The corresponding OUT endpoint needs to be read first by reading the USBHS_DAEPINTSTS register, and then the corresponding USBHS_DOUTEPxINTSTS register should be read to obtain the flag that caused the interrupt. Once the flag of the corresponding endpoint that caused the interrupt is cleared, this interrupt flag will be automatically cleared.

Bit Field	Name	Description
		Accessible in device modes.
18	INEPIF	IN Endpoints Interrupt flag When an interrupt is suspended on any IN endpoint in device mode, this flag is set. Software should first read the USBHS_DAEPINT register to obtain the corresponding IN endpoint, and then read the corresponding USBHS_DINEPxINSTS register to obtain the flag that caused the interrupt. Once the flag of the corresponding endpoint that caused the interrupt is cleared, this interrupt flag will be automatically cleared. Accessible in device modes.
17:16	Reserved	Reserved, must be kept at reset value
15	EOPFIF	End of Periodic Frame Interrupt flag When the USB bus time within one frame reaches the value defined by USBHS_DCFG.PFRITVL, the USBHS sets this interrupt flag. Accessible in device modes.
14	ISOUTPDIF	Isochronous OUT Packet Dropped Interrupt flag The controller sets this bit when it fails to write an isochronous OUT packet into the RxFIFO because the RxFIFO does not have enough space to accommodate a maximum packet size packet for the isochronous OUT endpoint. Accessible in device modes.
13	ENUMDIF	Enumeration Done Interrupt flag The core sets this bit to indicate that speed enumeration is complete. The application can read USBHS_DSTS.ENUMSPD[1:0] to obtain the current device speed. Accessible in device modes.
12	USBRSTIF	USB Reset Interrupt flag The controller sets this bit to indicate that a reset is detected on the USB. Accessible in device modes.
11	USBSUSPIF	USB Suspend Interrupt flag The controller sets this bit to indicate that a suspend was detected on the USB. The controller enters the Suspended state when there is no activity on the linestate signal for an extended period of time. Accessible in device modes.
10	ESUSPIF	Early Suspend Interrupt flag The controller sets this bit to indicate that an Idle state has been detected on the USB for 3 ms. Accessible in device modes.
9:8	Reserved	Reserved, must be kept at reset value
7	GOUTNAKEIF	Global OUT NAK Effective flag When USBHS_DCTRL.SGONAK is set to 1, and after the global OUT NAK takes effect, this flag is set. Software can clear this flag by writing 1 to USBHS_DCTRT.CGONAK.

Bit Field	Name	Description
		Accessible in device modes.
6	GINNPNAKEIF	Global IN Non-periodic NAK Effective flag When USBHS_DCTRL.SGINAK is set to 1, and after the global IN NAK takes effect, this flag is set. Software can clear this flag by writing 1 to USBHS_DCTRT.CGINAK. This interrupt does not necessarily indicate that a NAK handshake signal has been sent on the USB. The STALL bit takes precedence over the NAK bit. Accessible in device modes.
5	NPTXFEIF	Non-periodic TxFIFO Empty Interrupt flag When the non-periodic TX FIFO is half-empty or completely empty, and there is at least one entry to be written in the non-periodic request queue, this flag is set. The TX FIFO empty threshold is determined by USBHS_GAHBCFG.NPTXFETH. Note: Accessible in device and host modes.
4	RXFNEIF	RxFIFO Non-Empty Interrupt flag Indicates that there is at least one packet pending to be read from the RxFIFO. Note: Accessible in device and host modes.
3	SOFIF	Start of (micro)Frame (SOF) Interrupt flag In Host mode, the core sets this bit to indicate that an SOF (FS), micro-SOF (HS), or Keep-Alive (LS) is transmitted on the USB. The application must write a 1 to this bit to clear the interrupt. In Device mode, the controller sets this bit to indicate that an SOF token has been received on the USB. The application can read the Device Status register to get the current (micro)Frame number. This interrupt is seen only when the core is operating at either HS or FS. This bit can be set only by the core and the application must write 1 to clear it. Note: Accessible in device and host modes.
2	Reserved	Reserved, must be kept at reset value
1	MODMISIF	Mode Mismatch Interrupt Flag The core sets this bit when the application is trying to access: ■ A Host mode register, when the controller is operating in Device mode ■ A Device mode register, when the controller is operating in Host mode The register access is completed on the AHB with an OKAY response, but is ignored by the controller internally and does not affect the operation of the controller. This bit can be set only by the core and the application must write 1 to clear it. Note: Accessible in device and host modes.

Bit Field	Name	Description
0	CMODE	Current Mode of Operation Indicates the current mode. ■ 0: Device mode ■ 1: Host mode Note: Accessible in device and host modes.

34.11.2.6 USBHS Global Interrupt enable Register（USBHS_GINTEN）

Address offset：0x0018

Reset value：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
WKUPIEN	Reserved	DISCIEN	IDSTS CIEN	Reserved	PTXF EIEN	HCHIEN	HPIEN	RSTDIEN	FETSUSP	PTNCIEN ISOUTNCIEN	ISOINCIEN	OUTEP IEN	INEPIEN	Reserved	
rw		rw	rw		rw	rw	rw	rw	rw	rw	rw	rw	rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EOPFIEN	ISOUT PDIEN	ENUM DIEN	USB RSTIEN	USB SUSPIEN	ESUSP IEN	Reserved		GOUT NAKEIEN	GINNP NAKEIEN	NPTXF EIEN	RXFNE IEN	SOFIEN	USBHS IEN	MOD MISIEN	Reserved
rw	rw	rw	rw	rw	rw			rw	rw	rw	rw	rw	rw	rw	

Bit Field	Name	Description
31	WKUPIEN	Resume/Remote Wakeup Detected Interrupt enable 0：disable Resume or Remote Wakeup Detected Interrupt 1：enable Resume or Remote Wakeup Detected Interrupt Note: Accessible in device and host modes.
30	Reserved	Reserved, must be kept at reset value
29	DISCIEN	Disconnect Detected Interrupt enable 0：disable 1：enable Note: Accessible in host modes.
28	IDSTSCIEN	Connector ID Status Change interrupt enable 0：disable 1：enable Note: Accessible in device and host modes.
27	Reserved	Reserved, must be kept at reset value
26	PTXFEIEN	Periodic TxFIFO Empty interrupt enable 0：disable 1：enable Note: Accessible in host modes.
25	HCHIEN	Host Channels Interrupt enable 0：disable 1：enable Note: Accessible in host modes.

Bit Field	Name	Description
24	HPIEN	Host Port Interrupt enable 0：disable 1：enable Note: Accessible in host modes.
23	RSTDIEN	Reset detected Interrupt enable 0：disable 1：enable Note: Accessible in device modes.
22	FETSUSPIEN	Data Fetch Suspended Interrupt enable 0：disable 1：enable Note: Accessible in device modes.
21	PTNCIEN	**Incomplete Periodic Transfer** Interrupt enable Host mode： 0：disable 1：enable
	ISOUTNCIEN	**Incomplete Isochronous OUT Transfer Interrupt enable** Device mode： 0：disable 1：enable
20	ISOINCIEN	Incomplete Isochronous IN Transfer enable 0：disable 1：enable Note: Accessible in device modes.
19	OUTEPIEN	OUT Endpoints Interrupt enable 0：disable 1：enable Note: Accessible in device modes.
18	INEPIEN	IN Endpoints Interrupt enable 0：disable 1：enable Note: Accessible in device modes.
17:16	Reserved	Reserved, must be kept at reset value
15	EOPFIEN	End of Periodic Frame Interrupt enable 0：disable 1：enable Note: Accessible in device modes.
14	ISOUTPDIEN	Isochronous OUT Packet Dropped Interrupt enable 0：disable 1：enable Note: Accessible in device modes.
13	ENUMDIEN	Enumeration Done Interrupt enable

Bit Field	Name	Description
		0：disable 1：enable Note: Accessible in device modes.
12	USBRSTIEN	USB reset Interrupt enable 0：disable 1：enable Note: Accessible in device modes.
11	USBSUSPIEN	USB suspend Interrupt enable 0：disable 1：enable Note: Accessible in device modes.
10	ESUSPIEN	Early suspend Interrupt enable 0：disable 1：enable Note: Accessible in device modes.
9:8	Reserved	Reserved, must be kept at reset value
7	GOUTNAKEIEN	Global OUT NAK Effective enable 0：disable 1：enable Note: Accessible in device modes.
6	GINNPNAKEIEN	Global Non-periodic IN NAK Effective enable 0：disable 1：enable Note: Accessible in device modes.
5	NPTXFEIEN	Non-periodic TxFIFO Empty Interrupt enable 0：disable 1：enable Note: Accessible in device and host modes.
4	RXFNEIEN	Receive FIFO Non-Empty Interrupt enable 0：disable 1：enable Note: Accessible in device and host modes.
3	SOFIEN	Start of (micro)Frame Interrupt enable 0：disable 1：enable Note: Accessible in device and host modes.
2	USBHSIEN	USBHS Interrupt enable 0：disable 1：enable Note: Accessible in device and host modes.
1	MODMISIEN	Mode Mismatch Interrupt enable 0：disable

Bit Field	Name	Description
		1：enable Note: Accessible in device and host modes.
0	Reserved	Reserved, must be kept at reset value

34.11.2.7 USBHS Global Receive Status Debug Read Register （USBHS_GRXSTS/USBHS_GRXSTSP）

Read Address offset：0x001C

Pop Address offset：0x0020

Reset value：0x0000 0000

Reading the USBHS_GRXSTS register will return the content at the top of the receive FIFO. Reading the USBHS_GRXSTSP register will additionally pop the top data from the RX FIFO.

It is necessary to distinguish the receive status content in Host mode and Device mode. When the receive FIFO is empty, the USBHS will ignore USBHS_GRXSTSP. After the RX FIFO not empty interrupt flag USBHS_GINTSTS.RXFNEIF is set, the application can only read the USBHS_GRXSTS register.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved							FNUM[3:0]				PKSTS[3:0]				DPID[1]
							r				r				r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DPID[0]	BCNT[10:0]											CHEPNUM[3:0]			
r	r											r			

Bit Field	Name	Description
31:25	Reserved	Reserved, must be kept at reset value
24:21	FNUM[3:0]	Frame Number This is the least significant 4 bits of the (micro)Frame number in which the packet is received on the USB. This field is supported only when isochronous OUT endpoints are supported. Note: Accessible in device modes.
20:17	PKTSTS[3:0]	Packet Status (PktSts) indicates the status of the received packet. In host mode, ■ 4'b0010: IN data packet received ■ 4'b0011: IN transfer completed (triggers an interrupt) ■ 4'b0101: Data toggle error (triggers an interrupt) ■ 4'b0111: Channel halted (triggers an interrupt) ■ Others: Reserved Reset:4'b0 In device mode, ■ 4'b0001: Global OUT NAK (triggers an interrupt)

Bit Field	Name	Description
		■ 4'b0010: OUT data packet received ■ 4'b0011: OUT transfer completed (triggers an interrupt) ■ 4'b0100: SETUP transaction completed (triggers an interrupt) ■ 4'b0110: SETUP data packet received ■ Others: Reserved Reset:4'h0
16:15	DPID[1:0]	Data PID In host mode, indicates the Data PID of the received packet. In device mode, indicates the Data PID of the received OUT data packet. ■ 2'b00: DATA0 ■ 2'b10: DATA1 ■ 2'b01: DATA2 ■ 2'b11: MDATA Reset: 2'h0
14:4	BCNT[10:0]	Bytes count In host mode, indicates the byte count of the received IN data packet. In device mode, indicates the byte count of the received data packet.
3:0	CHEPNUM[3:0]	**Channel /Endpoint number:** Mode: Host only Indicates the channel number to which the current received packet belongs. Mode: Device only Indicates the endpoint number to which the current received packet belongs.

34.11.2.8 USBHS Global Receive FIFO Size Register（USBHS_GRXFSIZ）

Address offset：0x0024

Reset value：0x0000 0400

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
rw	rw	rw													

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved					RXFDEP[10:0]										
					rw										

Bit Field	Name	Description
31:11	Reserved	Reserved, must be kept at reset value
10:0	RXFDEP[10:0]	RxFIFO Depth This value is in terms of 32-bit words. ■ Minimum value is 16 ■ Maximum value is 32,768

34.11.2.9 USBHS Global Non-periodic Transmit FIFO Size Register（USBHS_GNPTXFSIZ）/ Device IN endpoint 0 transmit FIFO size register（USBHS_DINEP0TXFSIZ）

Address offset：0x0028

Reset value：0x0400 0400

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved					NPTXFDEP[10:0]/ IEP0TXFDEP[10:0]										
					rw										

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved					NPTXFSADD[10:0]/ IEP0TXFSADD[10:0]										
					rw										

Host mode

Bit Field	Name	Description
31:27	Reserved	Reserved, must be kept at reset value
26:16	NPTXFDEP[10:0]	Host Non-periodic TxFIFO Depth This value is in terms of 32-bit words. Minimum value is 16, Maximum value is 32,768, Must not exceed the reset value
15:11	Reserved	Reserved, must be kept at reset value
10:0	NPTXFSADD[10:0]	Host non-periodic TX FIFO start address Non-periodic TX FIFO start address

Device mode

Bit Field	Name	Description
31:27	Reserved	Reserved, must be kept at reset value
26:16	IEP0TXFDEP[10:0]	IN endpoint 0 TX FIFO depth This value is in terms of 32-bit words. Minimum value is 16, Maximum value is 1024, Must not exceed the reset value
15:11	Reserved	Reserved, must be kept at reset value
10:0	IEP0TXFRSADD[10:0]	IN endpoint 0 TX RAM start address Endpoint 0 TX FIFO RAM start address

34.11.2.10 USBHS Global Non-periodic Transmit FIFO/Queue Status Register （USBHS_GNPTXFSTS）

Address offset：0x002C

Reset value：0x0000 0400

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved	NPTXRQTOP[6:0]							NPTXRQSAV[7:0]							
	r							r							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NPTXFSAV[15:0]															
r															

Bit Field	Name	Description
31	Reserved	Reserved, must be kept at reset value
30:24	NPTXRQTOP[6:0]	Top of the Non-periodic Transmit Request Queue Entry in the Non-periodic Tx Request Queue that is currently being processed by the MAC. ■ Bits [30:27]: Channel/endpoint number ■ Bits [26:25]: ■ 2'b00: IN/OUT token - 2'b01: Zero-length transmit packet (device IN/host OUT) - 2'b10: PING/CSPLIT token - 2'b11: Channel halt command ■ Bit [24]: Terminate (last Entry for selected channel/endpoint)
23:16	NPTXRQSAV[7:0]	Non-periodic Transmit Request Queue Space Available Indicates the amount of free space available in the Non-periodic Transmit Request Queue. This queue holds both IN and OUT requests in Host mode. Device mode has only IN requests. ■ 8'h0: Non-periodic Transmit Request Queue is full ■ 8'h1: 1 location available ■ 8'h2: 2 locations available ■ n: n locations available (0 <= n <= 8) ■ Others: Reserved Reset: Configurable
15:0	NPTXFSAV[15:0]	Non-periodic TxFIFO Space Avail (NPTxFSpcAvail) Indicates the amount of free space available in the Non-periodic TxFIFO. Values are in terms of 32-bit words. ■ 16'h0: Non-periodic TxFIFO is full ■ 16'h1: 1 word available ■ 16'h2: 2 words available

Bit Field	Name	Description
		■ 16'hn: n words available (where 0 <= n <= 512) ■ Others: Reserved

34.11.2.11 USBHS User ID Register（USBHS_CID）

Address offset：0x003C

Reset value：0x0000 3608

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CID[31:16]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CID[15:0]															
rw															

Bit Field	Name	Description
31:0	CID	User ID (UserID) Application-programmable ID field. Reset value is configurable. Value After Reset: 0x12345678

34.11.2.12 USBHS Global Power Down Register（USBHS_GPD）

Address offset：0x0058

Reset value：0x0000 0010

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved										IDDIG	LINSTS		STSCHNGINTEN	STSCHNGINTSTS	Reserved
										r	r		rw	rc_w1	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	CONNDETINTEN	CONNDETSTS	DISCONNDETINTEN	DISCONNDETSTS	RESTDETINTEN	RESTDETSTS	LINSTSCHNGINTEN	LINSTSCHNGSTS	Reserved						
	rw	rc_w1	rw	rc_w1	rw	rc_w1	rw	rc_w1							

Bit Field	Name	Description
31:22	Reserved	Reserved, must be kept at reset value
21	IDDIG	This bit indicates the status of the signal IDDIG. The application must read this bit after receiving GPWRDN.StsChngInt and decode based on the previous value stored by the application. Indicates the current mode. ■ 0: Host mode ■ 1: Device mode

Bit Field	Name	Description
20:19	LINSTS[0:1]	LineState. ■ 2'b00: DM = 0, DP = 0. ■ 2'b01: DM = 0, DP = 1. ■ 2'b10: DM = 1, DP = 0. ■ 2'b11: Not-defined.
18	STSCHNGINTEN	Mask for StsChng Interrupt ■ 0 (NOMASK): No Status Change Interrupt Mask ■ 1 (MASK): Mask for Status Change Interrupt
17	STSCHNGINTSTS	Status Change Interrupt (StsChngInt) ■ 0: No Status change ■ 1: Status change detected
16:15	Reserved	Reserved, must be kept at reset value
14	CONNDETINTEN	Mask for ConnectDet interrupt ■ 0 (NOMASK): No ConnectDet Interrupt Mask ■ 1 (MASK): Mask for ConnectDet Interrupt
13	CONNDETSTS	This field indicates that a new connect has been detected ■ 0: Connect not detected ■ 1: Connect detected
12	DISCONNDETINTEN	Mask For DisconnectDetect Interrupt ■ 0 (NOMASK): No DisconnectDetect Interrupt Mask ■ 1 (MASK): Mask for DisconnectDetect Interrupt
11	DISCONNDETSTS	This field generates an interrupt. After detecting disconnect during hibernation the application must not restore the core, but instead start the initialization process. ■ 0: Disconnect not detected ■ 1: Disconnect detected
10	RESTDETINTEN	Mask for ResetDetected interrupt ■ 0 (NOMASK): No ResetDetect Interrupt Mask ■ 1 (MASK): Mask for ResetDetect Interrupt
9	RESTDETSTS	This field indicates that Reset has been detected by the PMU module. This field generates an interrupt. ■ 0: Reset Not Detected ■ 1: Reset Detected
8	LINSTSCHNGINTEN	Mask for LineStateChange interrupt. ■ 0 (NOMASK): No LineStateChange Interrupt Mask ■ 1 (MASK): Mask for LineStateChange Interrupt
7	LINSTSCHNGSTS	This interrupt is asserted when there is a Linestate Change detected by the PMU. The application must read GPWRDN.Linestate to determine the current linestate on USB. ■ 0: No LineState change on USB ■ 1: LineState change on USB
6:0	Reserved	Reserved, must be kept at reset value

34.11.2.13 USBHS Host Periodic Transmit FIFO Size Register（USBHS_HPTXFSIZ）

Address offset：0x00100

Reset value：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
HPTXFDEP[15:0]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HPTXFSADD[15:0]															
rw															

Bit Field	Name	Description
31:16	HPTXFDEP[15:0]	Host Periodic TxFIFO Depth This value is in terms of 32-bit words. ■ Minimum value is 16 ■ Maximum value is 1024
15:0	HPTXFSADD[15:0]	Host Periodic TxFIFO Start Address

34.11.2.14 USBHS Device Periodic Transmit FIFO-x Size Register（USBHS_DINEPPTXFSIZx）(x=[1..8])

Address offset：0x00104 + (x-1)×4

Reset value：0x0100 0500

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved					INEPTXFDEP [10:0]										
					rw										

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved					INEPTXSADD [10:0]										
					rw										

Bit Field	Name	Description
31:27	Reserved	Reserved, must be kept at reset value
26:16	INEPTXFDEP [10:0]	Device Periodic TxFIFO Size (DPTxFSize) This value is in terms of 32-bit words. ■ Minimum value is 16 ■ Maximum value is 768
15:11	Reserved	Reserved, must be kept at reset value
10:0	INEPTXSADD [10:0]	Device Periodic TxFIFO RAM Start Address

34.11.3 USBHSHost control and status register

34.11.3.1 USBHS Host Configuration Register（USBHS_HCFG）

Address offset：0x0400

Reset value：0x0000 0200

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
rw	rw	rw													

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													SPSEL	Reserved	
													rw		

Bit Field	Name	Description
31:3	Reserved	Reserved, must be kept at reset value
2	SPSEL	The application uses this bit to control the core's enumeration speed. Using this bit, the application can make the core enumerate as a FS host, even If the connected device supports HS traffic. Do not make changes to this field after initial programming. ■ 0: HS/FS/LS, based on the maximum speed supported by the connected device ■ 1: FS/LS-only, even If the connected device can support HS
1:0	Reserved	Reserved, must be kept at reset value

34.11.3.2 USBHS Host Frame Interval Register（USBHS_HFRI）

Address offset：0x0404

Reset value：0x0000 EA60

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															DRLDEN
															rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FRI[15:0]															
rw															

Bit Field	Name	Description
31:17	Reserved	Reserved, must be kept at reset value
16	DRLDEN	Reload Control This bit allows dynamic reloading of the HFIR register during run time. ■ 0 : The HFIR cannot be reloaded dynamically ■ 1: the HFIR can be dynamically reloaded during runtime. This bit needs to be programmed during initial configuration and its value must not be changed during runtime.
15:0	FRI[15:0]	Frame Interval (FrInt) The value that the application programs to this field specifies the interval between two consecutive SOFs (FS) or micro- SOFs (HS) or Keep-Alive tokens (HS). This field contains the number of PHY clocks that constitute the required frame interval. The Default value set in this field is for FS operation when the PHY clock frequency is 60 MHz. The application can write a value to this register only after the Port Enable bit of the Host Port Control and Status register (HPRT.PrtEnaPort) has been Set. If no value is programmed, the core calculates the value based on the PHY clock specified in the FS/LS PHY Clock Select field of the Host Configuration register (HCFG.FSLSPclkSel). Do not change the value of this field after the initial configuration. ■ 125 us * (PHY clock frequency for HS) ■ 1 ms * (PHY clock frequency for FS/LS)

34.11.3.3 USBHS Host Frame Number/Frame Time Remaining Register（USBHS_HFNUM）

Address offset：0x0408

Reset value：0x0000 3FFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FRT[15:0]															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FRNUM[15:0]															
r															

Bit Field	Name	Description
31:16	FRT[15:0]	Frame Remaining Time It indicates the remaining time of the current frame (in PHY clock cycles). This field decrements by 1 for each PHY clock cycle. When the value reaches zero, this field reloads the value of the frame interval register (USBHS_HFRI.FRI[15:0]) and the

Bit Field	Name	Description
		module sends a new SOF on the USB.
15:0	FRNUM[15:0]	Frame Number This field increments when a new SOF is transmitted on the USB, and is reset to 0 when it reaches 16'h3FFF. ■ 0 (INACTIVE): No SOF is transmitted ■ 1 (ACTIVE): SOF is transmitted

34.11.3.4 USBHS Host Periodic Transmit FIFO/Queue Status Register（USBHS_HPTXFQSTS）

Address offset：0x0410

Reset value：0x0008 0400

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PTXRQTOP[8:0]									PTXRQSAVL[6:0]						
r									r						

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PTXFSAVL[15:0]															
r															

Bit Field	Name	Description
31:23	PTXRQTOP[8:0]	Top of the Periodic Transmit Request Queue This indicates the Entry in the Periodic Tx Request Queue that is currently being processed by the MAC. This register is used for debugging. ■ Bit [31]: Odd/Even (micro)Frame - 1'b0: send in even (micro)Frame - 1'b1: send in odd (micro)Frame ■ Bits [30:27]: Channel/endpoint number ■ Bits [26:25]: Type - 2'b00: IN/OUT - 2'b01: Zero-length packet - 2'b10: CSPLIT - 2'b11: Disable channel command ■ Bit [24]: Last Periodic Entry for the selected periodic channel/endpoint ■ Bit [23]: Terminate (last Entry for the selected channel/endpoint)
22:16	PTXRQSAVL[6:0]	Periodic Transmit Request Queue Space Available Indicates the number of free locations available to be written in the Periodic Transmit Request Queue. This queue holds both IN and OUT requests. ■ 7'h0: Periodic Transmit Request Queue is full

		■ 7'h1: 1 location available ■ 7'h2: 2 locations available ■ n: n locations available (0 <= n <= 16) ■ Others: Reserved
15:0	PTXFSAVL[15:0]	Periodic Transmit Data FIFO Space Available Indicates the number of free locations available to be written to in the Periodic TxFIFO. Values are in terms of 32-bit words ■ 16'h0 : Periodic TxFIFO is full ■ 16'h1 : 1 word available ■ 16'h2 : 2 words available ■ 16'hn : n words available (where0 ≤ n ≤USBHS_HPTXFSIZ.HPTXFDEP[15:0]) ■ Others : Reserved

34.11.3.5 USBHSHost All Channels Interrupt Register（USBHS_HACHINT）

Address offset：0x0414

Reset value：0x0000 0000

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="16">Reserved</td></tr>
</table>

<table>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="16">HACHINT[15:0]</td></tr>
<tr><td colspan="16">rw</td></tr>
</table>

Bit Field	**Name**	**Description**
31:16	Reserved	Reserved, must be kept at reset value
15:0	HACHINT[15:0]	Channel Interrupt for channel no. One bit for each channel: CH0 corresponds to bit 0, CH15 corresponds to bit 15.

34.11.3.6 USBHS Host All Channels Interrupt Mask Register（USBHS_HACHINTEN）

Address offset：0x0418

Reset value：0x0000 0000

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="16">Reserved</td></tr>
</table>

<table>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="16">HACHINTEN[15:0]</td></tr>
<tr><td colspan="16">rw</td></tr>
</table>

Bit Field	Name	Description
31:16	Reserved	Reserved, must be kept at reset value
15:0	HACHINTEN[15:0]	Channel Interrupt Mask (HAINTMsk) One bit per channel: Bit 0 for channel 0, bit 15 for channel 15 ■ 0x0 (UNMASK): Unmask Channel interrupt ■ 0x1 (MASK): Mask Channel interrupt

34.11.3.7 USBHS Host Port Control and Status Register（USBHS_HPCS）

Address offset：0x0440

Reset value：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved													PSPD[1:0]		PT CTRL[3]
rw	rw	rw											r		rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PTCTRL[2:0]			PPWR	PLSTS[1:0]		Reserved	PRST	PSUSP	PRES	POCC	POCA	PENC	PEN	PCDET	PCSTS
rw			rw	r			rw	rs	rw	rc_w1	r	rc_w1	rc_w1	rc_w1	r

Bit Field	Name	Description
31:19	Reserved	Reserved, must be kept at reset value
18:17	PSPD[1:0]	Port Speed (PrtSpd) Indicates the speed of the device attached to this port. ■ 2'b00: High speed ■ 2'b01: Full speed ■ 2'b10: Low speed ■ 2'b11: Reserved
16:13	PTCTRL[3:0]	Port Test Control The application writes a nonzero value to this field to put the port into a Test mode, and the corresponding pattern is signaled on the port. ■ 4'b0000: Test mode disabled ■ 4'b0001: Test_J mode ■ 4'b0010: Test_K mode ■ 4'b0011: Test_SE0_NAK mode ■ 4'b0100: Test_Packet mode ■ 4'b0101: Test_Force_Enable ■ Others: Reserved
12	PPWR	Port Power The application uses this field to control power to this port (write 1'b1 to set to 1'b1 and write 1'b0 to set to 1'b0), and the core can

Bit Field	Name	Description
		clear this bit on an over current condition. ■ 0: Power off ■ 1: Power on
11:10	PLSTS[1:0]	Port Line Status Indicates the current logic level USB data lines ■ Bit [10]: Logic level of D+ ■ Bit [11]: Logic level of D
9	Reserved	Reserved, must be kept at reset value
8	PRST	Port Reset When the application sets this bit, a reset sequence is started on this port. The application must time the reset period and clear this bit after the reset sequence is complete. ■ 0: Port not in reset ■ 1: Port in reset The application must leave this bit set for at least a minimum duration mentioned below to start a reset on the port. The application can leave it set for another 10 ms in addition to the required minimum duration, before clearing the bit, even though there is no maximum limit Set by the USB standard.This bit is cleared by the core even if there is no device connected to the Host.
7	PSUSP	Port Suspend The core only stops sending SOFs when this is Set. Once this control bit is set, it can only be cleared through the following operations. – set USBHS_HPCS.PRST – set USBHS_HPCS.PRES –Detected a remote wakeup signal –Detected a device disconnect ■ 0: Port not in Suspend mode ■ 1: Port in Suspend mode
6	PRES	Port Resume The application sets this bit to drive resume signaling on the port. The core continues to drive the resume signal until the application clears this bit. If the core detects a USB remote wakeup sequence, as indicated by the Port Resume/Remote Wakeup Detected Interrupt bit of the Core Interrupt register (USBHS_GINTSTS. WKUPIF), the core starts driving resume signaling without application intervention and clears this bit when it detects a disconnect condition. The read value of this bit indicates whether the core is currently driving resume signaling. ■ 0: No resume driven

Bit Field	Name	Description
		■ 1: Resume driven
5	POCC	Port Overcurrent Change The core sets this bit when the status of the Port Overcurrent Active bit (bit 4) in this register changes.This bit can be set only by the core and the application must write 1 to clear it
4	POCA	Port Overcurrent Active Indicates the overcurrent condition of the port. ■ 0: No overcurrent condition ■ 1: Overcurrent condition
3	PENC	Port Enable/Disable Change The core sets this bit when the status of the Port Enable bit [2] of this register changes.This bit can be set only by the core and the application must write 1 to clear it.
2	PEN	Port Enable A port is enabled only by the core after a reset sequence, and is disabled by an overcurrent condition, a disconnect condition, or by the application clearing this bit. The application cannot Set this bit by a register write. It can only clear it to disable the port by writing 1. This bit does not trigger any interrupt to the application. ■ 0: Port disabled ■ 1: Port enabled
1	PCDET	Port Connect Detected The core sets this bit when a device connection is detected to trigger an interrupt to the application using the Host Port Interrupt bit of the Core Interrupt register (GINTSTS.PrtInt).This bit can be set only by the core and the application must write 1 to clear it.The application must write a 1 to this bit to clear the interrupt.
0	PCSTS	Port Connect Status ■ 0: No device is attached to the port. ■ 1: A device is attached to the port.

34.11.3.8 USBHS Host Channel Characteristics Register（USBHS_HCHxCTRL）(x=[0..15])

Address offset：0x0500 + x×20

Reset value：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CHEN	CHDIS	ODDFRM	DEVADDR[6:0]							MCNT[1:0]		EPTYPE[1:0]		LSPDDEV	Reserved
rs	rs	rw	rw							rw		rw		rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EPDIR	EPNUM[3:0]				MPS[10:0]										
rw	rw				rw										

Bit Field	Name	Description
31	CHEN	Channel Enable This field is set to 1 by the application software and cleared by hardware. ■ 1'b0: Channel disabled ■ 1'b1: Channel enabled
30	CHDIS	Channel Disable The application sets this bit to stop transmitting/receiving data on a channel, even before the transfer for that channel is complete. The application must wait for the Channel Disabled interrupt before treating the channel as disabled. ■ 0x0 (INACTIVE): Transmit/Recieve normal ■ 0x1 (ACTIVE): Stop transmitting/receiving data on channel
29	ODDFRM	Odd Frame This field is set (reset) by the application to indicate that the OTG host must perform a transfer in an odd (micro)Frame. This field is applicable for only periodic (isochronous and interrupt) transactions. ■ 0: Even (micro)Frame ■ 1: Odd (micro)Frame
28:22	DEVADDR[6:0]	Device Address This field selects the specific device serving as the data source or sink.
21:20	MCNT[1:0]	Multi Count For periodic transfers, this Bit Field specifies the number of transactions the host must perform in each microframe. For non periodic transfers, this field is used only in DMA mode, and specifies the number packets to be fetched for this channel before the internal DMA engine changes arbitration. ■ 2'b00: Reserved This field yields undefined results. ■ 2'b01: 1 transaction ■ 2'b10: 2 transactions to be issued for this endpoint per microframe ■ 2'b11: 3 transactions to be issued for this endpoint per microframe
19:18	EPTYPE[1:0]	Endpoint Type Indicates the transfer type selected. ■ 2'b00: Control ■ 2'b01: Isochronous ■ 2'b10: Bulk ■ 2'b11: Interrupt
17	LSPDDEV	Low-Speed Device This field is Set by the application to indicate that this channel is communicating to a low-speed device.

Bit Field	Name	Description
16	Reserved	Reserved, must be kept at reset value
15	EPDIR	Endpoint Direction (EPDir) Indicates whether the transaction is IN or OUT. ■ 0: OUT ■ 1: IN
14:11	EPNUM[3:0]	Endpoint Number Indicates the endpoint number on the device serving as the data source or sink.
10:0	MPS[10:0]	Maximum Packet Size Indicates the maximum packet size of the associated endpoint.

34.11.3.9 USBHS Host Channel Split Control Register（USBHS_HCHxSCTRL）(x=[0..15])

Address offset：0x0504 + x×20

Reset value：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SPLEN	Reserved														COMP SPLF
rw															rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRANPOS[1:0]		HUBADDR[6:0]							PRTADD[6:0]						
rw		rw							rw						

Bit Field	Name	Description
31	SPLEN	Split Enable (SpltEna) The application sets this field to indicate that this channel is enabled to perform split transactions.
30:17	Reserved	Reserved, must be kept at reset value
16	COMPSPLF	Do Complete Split The application sets this field to request the OTG host to perform a complete split transaction.
15:14	TRANPOS[1:0]	Transaction Position This field is used to determine whether to send all, first, middle, or last payloads with each OUT transaction. ■ 2'b11: All. This is the entire data payload is of this transaction (which is less than or equal to 188 bytes). ■ 2'b10: Begin. This is the first data payload of this transaction (which is larger than 188 bytes). ■ 2'b00: Mid. This is the middle payload of this transaction (which is larger than 188 bytes). ■ 2'b01: End. This is the last payload of this transaction (which is larger than 188 bytes).

Bit Field	Name	Description
13:7	HUBADD[6:0]	Hub Address This field holds the device address of the transaction translator's hub.
6:0	PRTADD[6:0]	Port Address This field is the port number of the recipient transaction translator.

34.11.3.10 USBHS Host Channel Interrupt Register（USBHS_HCHxINTSTS）(x=[0..15])

Address offset：0x0508 + x×20

Reset value：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved					DTERRIF	FOVRIF	BBERRIF	TXERRIF	NYETIF	ACKIF	NAKIF	STALLIF	AHB ERRIF	CHHTDIF	TXCFIF
					rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1

Bit Field	Name	Description
31:11	Reserved	Reserved, must be kept at reset value
10	DTERRIF	Data Toggle Error ■ 0 (INACTIVE): No Data Toggle Error ■ 1 (ACTIVE): Data Toggle Error
9	FOVRIF	Frame Overrun : ■ 0 (INACTIVE): No Frame Overrun ■ 1 (ACTIVE): Frame Overrun
8	BBERRIF	Babble Error ■ 0 (INACTIVE): No Babble Error ■ 1 (ACTIVE): Babble Error
7	TXERRIF	Transaction Error Indicates one of the following errors occurred on the USB. ■ CRC check failure ■ Timeout ■ Bit stuff error ■ False EOP In Scatter/Gather DMA mode, the interrupt due to this bit is masked in the core.This bit can be set only by the core and the application must write 1 to clear it. ■ 0 (INACTIVE): No Transaction Error ■ 1 (ACTIVE): Transaction Error
6	NYETIF	NYET Response Received Interrupt (NYET)

Bit Field	Name	Description
		In Scatter/Gather DMA mode, the interrupt due to this bit is masked in the core.This bit can be set only by the core and the application must write 1 to clear it. ■ 0 (INACTIVE): No NYET Response Received Interrupt ■ 1 (ACTIVE): NYET Response Received Interrupt
5	ACKIF	ACK Response Received/Transmitted Interrupt (ACK) In Scatter/Gather DMA mode, the interrupt due to this bit is masked in the core.This bit can be set only by the core and the application must write 1 to clear it. ■ 0 (INACTIVE): No ACK Response Received or Transmitted Interrupt ■ 1 (ACTIVE): ACK Response Received or Transmitted Interrup
4	NAKIF	NAK Response Received Interrupt (NAK) In Scatter/Gather DMA mode, the interrupt due to this bit is masked in the core.This bit can be set only by the core and the application must write 1 to clear it. ■ 0 (INACTIVE): No NAK Response Received Interrupt ■ 1 (ACTIVE): NAK Response Received Interrupt
3	STALLIF	STALL Response Received Interrupt (STALL) In Scatter/Gather DMA mode, the interrupt due to this bit is masked in the core.This bit can be set only by the core and the application must write 1 to clear it. ■ 0 (INACTIVE): No Stall Response Received Interrupt ■ 1 (ACTIVE): Stall Response Received Interrupt
2	AHBERRIF	AHB Error This is generated only in Internal DMA mode when there is an AHB error during AHB read/write. The application can read the corresponding channel's DMA address register to get the error address. For details, see "AHB Error Handling" in the Programming Guide. ■ 0 (INACTIVE): No AHB error ■ 1 (ACTIVE): AHB error during AHB read/write
1	CHHTDIF	Channel Halted When DMA is not enabled: It indicates abnormal completion of the transfer, possibly due to any USB transaction error, application disable request, or completed transfer. When DMA is enabled: This indicates abnormal termination of the transfer due to any USB transaction error or in response to an application disable request. ■ 0 (INACTIVE): Channel not halted ■ 1 (ACTIVE): Channel Halted
0	TXCFIF	Transfer Completed ■ 0 (INACTIVE): Transfer in progress or No Active Transfer

Bit Field	Name	Description
		■ 1 (ACTIVE): Transfer completed normally without any errors

34.11.3.11 USBHS Host Channel Interrupt Mask Register（USBHS_HCHxINTEN）(x=[0..15])

Address offset：0x050C + x×20

Reset value：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved					DTERR IEN	FOVRIEN	BBERR IEN	TXERR IEN	NYETIEN	ACKIEN	NAKIEN	STALL IEN	AHBERR IEN	CHHLTD IEN	TXCIEN
					rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit Field	Name	Description
31:11	Reserved	Reserved, must be kept at reset value
10	DTERRIEN	Data Toggle Error Mask ■ 0 (MASK): Mask Data Toggle Error ■ 1 (NOMASK): No Data Toggle Error Mask
9	FOVRIEN	Frame Overrun Mask ■ 0 (MASK): Mask Overrun Mask ■ 1 (NOMASK): No Frame Overrun Mask
8	BBERRIEN	Babble Error Mask ■ 0 (MASK): Mask Babble Error ■ 1 (NOMASK): No Babble Error Mask
7	TXERRIEN	Transaction Error Mask ■ 0 (MASK): Mask Transaction Error ■ 1 (NOMASK): No Transaction Error Mask
6	NYETIEN	NYET Response Received Interrupt Mask ■ 0 (MASK): Mask NYET Response Received Interrupt ■ 1 (NOMASK): No NYET Response Received Interrupt Mask
5	ACKIEN	ACK Response Received/Transmitted Interrupt Mask ■ 0 (MASK): Mask ACK Response Received/Transmitted Interrupt ■ 1 (NOMASK): No ACK Response Received/Transmitted Interrupt Mask
4	NAKIEN	NAK Response Received Interrupt Mask ■ 0 (MASK): Mask NAK Response Received Interrupt ■ 1 (NOMASK): No NAK Response Received Interrupt Mask
3	STALLIEN	STALL Response Received Interrupt Mask ■ 0 (MASK): Mask STALL Response Received Interrupt ■ 1 (NOMASK): No STALL Response Received Interrupt Mask
2	AHBERRIEN	AHB Error Mask

Bit Field	Name	Description
		■ 0 (MASK): AHB Error Mask ■ 1 (NOMASK): No AHB Error Mask
1	CHHTDIEN	Channel Halted Mask ■ 0 (MASK): Channel Halted Mask ■ 1 (NOMASK): No Channel Halted Mask
0	TXCIEN	Transfer Completed Mask ■ 0 (MASK): Transfer Completed Mask ■ 1 (NOMASK): No Transfer Completed Mask

34.11.3.12 USBHS Host Channel Transfer Size Register（USBHS_HCHxTXSIZ）(x=[0..15])

Address offset：0x0510 + x×20

Reset value：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DPING	PID[1:0]		PKCNT[9:0]										TXSIZ[18:16]		
rw	rw		rw										rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TXSIZ[15:0]															
rw															

Bit Field	Name	Description
31	DPING	Token Request This bit is only valid for OUT transfers. If set by software, the USBHS will perform the PING protocol. When an OUT transaction receives a NAK or NYET handshake packet, the USBHS will automatically set this control bit. Do not set this bit for IN transfers.
30:29	PID[1:0]	PID Software should write this Bit Field before the start of the transfer. For OUT transfers, this Bit Field contains the data PID of the first transfer packet. For IN transfers, this Bit Field contains the data PID of the first received packet. After the transfer begins, the USBHS automatically changes and switches this Bit Field according to the USB protocol. 00： DATA0 01： DATA2 10： DATA1 11： MDATA (non-control)/SETUP (control)
28:19	PKCNT[9:0]	Packet Count This field is programmed by the application with the expected number of packets to be transmitted (OUT) or received (IN). The host decrements this count on every successful transmission

Bit Field	Name	Description
		or reception of an OUT/IN packet. Once this count reaches zero, the application is interrupted to indicate normal completion.
18:0	TXSIZ[18:0]	Transfer Size For an OUT, this field is the number of data bytes the host sends during the transfer. For an IN, this field is the buffer size that the application has Reserved for the transfer. The application is expected to program this field as an integer multiple of the maximum packet size for IN transactions (periodic and non-periodic).

34.11.3.13 USBHS Host Channel DMA Address Register（USBHS_HCHxDMADD）(x=[0..15])

Address offset：0x0514 + x×20

Reset value：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DMADD[31:0]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMADD[31:0]															
rw															

Bit Field	Name	Description
31:0	DMADD[31:0]	DMA Address This field stores the starting address of the memory used by the host to fetch data from a device endpoint or send data to a device endpoint. This register is incremented with each DMA transfer.

34.11.4 USBHS Device control and status register

34.11.4.1 USBHS Device Configuration Register（USBHS_DCFG）

Address offset：0x0800

Reset value：0x0802 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved						PSITVAL[1:0]		Reserved							
						rw									

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EERRIEN	XCVR DLY	Reserved	PFRITVL[1:0]		DEVADD[6:0]							Reserved	NZLSOUT HSK	DEVSPD[1:0]	
rw	rw	rw	rw					rw				rw	rw	rw	rw

Bit Field	Name	Description
31:26	Reserved	Reserved, must be kept at reset value
25:24	PSITVL[1:0]	Periodic Scheduling Interval PerSchIntvl must be programmed for Scatter/Gather DMA mode. This field specifies the amount of time the Internal DMA engine must allocate for fetching periodic IN endpoint data. Based on the number of periodic endpoints, this value must be specified as 25,50 or 75% of (micro)Frame. ■ When any periodic endpoints are active, the internal DMA engine allocates the specified amount of time in fetching periodic IN endpoint data . ■ When no periodic endpoints are active, Then the internal DMA engine services non-periodic endpoints, ignoring this field. ■ After the specified time within a (micro)Frame, the DMA switches to fetching for non-periodic endpoints. - 2'b00: 25% of (micro)Frame. - 2'b01: 50% of (micro)Frame. - 2'b10: 75% of (micro)Frame. - 2'b11: Reserved
23:16	Reserved	Reserved, must be kept at reset value
15	EERRAIEN	Erratic Error Interrupt Mask ■ 0 (NOMASK): Early suspend interrupt is generated on erratic error ■ 1 (MASK): Mask early suspend interrupt on erratic error
14	XCVRDLY	Transceiver delay Enable or disable the delay of PHY timing during device chirping. 0：Disable delay (use default timing) 1：Enable default timing delay
13	Reserved	Reserved, must be kept at reset value
12:11	PFRITVL[1:0]	Periodic Frame Interval Indicates the time within a (micro)Frame at which the application must be notified using the End Of Periodic Frame Interrupt. This can be used to determine If all the isochronous traffic for that (micro)Frame is complete. ■ 2'b00: 80% of the (micro)Frame interval ■ 2'b01: 85% of the (micro)Frame interval ■ 2'b10: 90% of the (micro)Frame interval ■ 2'b11: 95% of the (micro)Frame interval
10:4	DEVARR[6:0]	Device Address The application must program this field after every SetAddress control command.
3	Reserved	Reserved, must be kept at reset value

Bit Field	Name	Description
2	NZLSOUTHSK	Non-Zero-Length Status OUT Handshake The application can use this field to select the handshake the core sends on receiving a nonzero-length data packet during the OUT transaction of a control transfer's Status stage. ■ 1: Send a STALL handshake on a nonzero-length status OUT transaction and do not send the received OUT packet to the application. ■ 0: Send the received OUT packet to the application (zerolength or nonzero-length) and send a handshake based on the NAK and STALL bits for the endpoint in the Device Endpoint Control register.
1:0	DEVSPD[1:0]	Device Speed Indicates the speed at which the application requires the core to enumerate, or the maximum speed the application can support. However, the actual bus speed is determined only after the connect sequence is completed, and is based on the speed of the USB host to which the core is connected. Values: ■ 0x0 (USBHS20): High speed USB 2.0 PHY clock is 30 MHz or 60 MHz ■ 0x1 (USBFS20): Full speed USB 2.0 PHY clock is 30 MHz or 60 MHz ■ Others: Reserved

34.11.4.2 USBHSDevice Control Register（USBHS_DCTRL）

Address offset：0x0804

Reset value：0x0000 0002

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															NAKOBBLE
rw	rw	rw													rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				POPDNE	CGONAK	SGONAK	CGNPINAK	SGINAK	TSCTRL[2:0]			GONAKSTS	GINAKSTS	SFTDIS	RMWKUP
				rw	w	w	w	w		rw		r	r	rw	rw

Bit Field	Name	Description
31:17	Reserved	Reserved, must be kept at reset value
16	NAKOBBLE	NAK on Babble Error Set NAK automatically on babble (NakOnBble). The core sets NAK automatically for the endpoint on which babble is received.

Bit Field	Name	Description
15:12	Reserved	Reserved, must be kept at reset value
11	POPDNE	Power-On Programming Done The application uses this bit to indicate that register programming is completed after a wake-up from Power Down mode.
10	CGONAK	Clear Global OUT NAK A write to this field clears the Global OUT NAK.
9	SGONAK	Set Global OUT NAK Writing to this bit will set the global OUT NAK to 1. The application sends NAK handshake signals to all OUT endpoints. The application can set this bit to 1 only when it has ensured that the global OUT NAK valid bit in the module interrupt register (USBHS_GINTSTS.GOUTNAKEIF) has been cleared.
8	CGNPINAK	Clear Global Non-periodic IN NAK A write to this field clears the Global Non-periodic IN NAK.
7	SGINAK	Set Global Non-periodic IN NAK A write to this field sets the Global Non-periodic IN NAK.The application uses this bit to send a NAK handshake on all nonperiodic IN endpoints. The core can also Set this bit when a timeout condition is detected on a non-periodic endpoint in shared FIFO operation. The application can set this bit to 1 only when it has ensured that the global IN NAK valid bit in the module interrupt register (USBHS_GINTSTS.GINNPNAKEIF) has been cleared.
6:4	TSCTRL[2:]	Test Control ■ 3'b000: Test mode disabled ■ 3'b001: Test_J mode ■ 3'b010: Test_K mode ■ 3'b011: Test_SE0_NAK mode ■ 3'b100: Test_Packet mode ■ 3'b101: Test_Force_Enable ■ Others: Reserved
3	GONAKSTS	Global OUT NAK Status (GOUTNakSts) ■ 1'b0: A handshake is sent based on the FIFO Status and the NAK and STALL bit settings. ■ 1'b1: No data is written to the RxFIFO, irrespective of space availability. Sends a NAK handshake on all packets, except on SETUP transactions. All isochronous OUT packets are dropped.
2	GINAKSTS	Global Non-periodic IN NAK Status ■ 1'b0: A handshake is sent out based on the data availability in the transmit FIFO. ■ 1'b1: A NAK handshake is sent out on all non-periodic IN endpoints, irrespective of the data availability in the transmit FIFO.
1	SFTDIS	Soft Disconnect

Bit Field	Name	Description
		The application uses this bit to signal the controller to do a soft disconnect. As long as this bit is set, the host does not see that the device is connected, and the device does not receive signals on the USB. The core stays in the disconnected state until the application clears this bit. ■ 1'b0: Normal operation. When this bit is cleared after a soft disconnect, the core drives the phy_opmode_o signal on the UTMI+ to 2'b00, which generates a device connect event to the USB host. When the device is reconnected, the USB host restarts device enumeration. ■ 1'b1: The core drives the phy_opmode_o signal on the UTMI+ to 2'b01, which generates a device disconnect event to the USB host.
0	RMWKUP	Remote Wakeup Signaling (RmtWkUpSig) When the application sets this bit, the core initiates remote signaling to wake up the USB host. The application must Set this bit to instruct the core to exit the Suspend state. As specified in the USB 2.0 specification, the application must clear this bit 1-15 ms after setting it. ■ 0x0 (DISABLEDRMWKUP): Core does not send Remote Wakeup Signaling ■ 0x1 (ENABLERMWKUP): Core sends Remote Wakeup Signaling

34.11.4.3 USBHS Device Status Register（USBHS_DSTS）

Address offset：0x0808

Reset value：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								DEVLINSTS[1:0]		SOFFN[13:8]					
								r		r					

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOFFN[7:0]								Reserved				ERERRF	ENUMSPD[1:0]		SUSPF
r												r	r		r

Bit Field	Name	Description
31:24	Reserved	Reserved, must be kept at reset value
23:22	DEVLINSTS[1:0]	Device Line Status Indicates the current logic level USB data lines ■ DevLnSts[1]: Logic level of D+

Bit Field	Name	Description
		■ DevLnSts[0]: Logic level of DValue
21:8	SOFFN[13:0]	Frame or Microframe Number of the Received SOF The USBHS will update this field upon receipt of a SOF token.
7:4	Reserved	Reserved, must be kept at reset value
3	ERERRF	Erratic Error The module sets this bit to 1 to report any undefined error. Due to the undefined error, the USBHS controller will enter a suspended state, and an interrupt will be generated for the application with the early suspend bit (USBHS_GINTSTS.ESUSPIF) of the USBHS_GINTSTS register. If the early suspend interrupt is triggered by an undefined error, the application can only perform a soft disconnect to recover communication.
2:1	ENUMSPD[1:0]	Enumerated Speed (EnumSpd) Indicates the speed at which the controller has come up after speed detection through a connect or reset sequence. ■ 2'b00: High speed (PHY clock is running at 30 or 60 MHz) ■ 2'b01: Full speed (PHY clock is running at 30 or 60 MHz) ■ Others: Reserved
0	SUSPF	Suspend Status In Device mode, this bit is set as long as a Suspend condition is detected on the USB. The core enters the Suspend state when there is no activity on the phy_line_state_i signal for an extended period of time. The core comes out of the suspend under the following conditions : –There is activity on the USB data line. –The application writes to the remote wakeup bit (USBHS_DCTRL.RMWKUP) in the USBHS_DCTRL register.

34.11.4.4 USBHS Device IN Endpoint Common Interrupt Mask Register （USBHS_DINEPINTEN）

Address offset：0x0810

Reset value：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		NAKIEN	Reserved				TXFUD IEN	Reserved	INEPNAK EIEN	INTREP MSIEN	TXFERIN TKIEN	TOIEN	AHBERR IEN	EPDISIEN	TXCIEN
		rw					rw		rw	rw	rw	rw	rw	rw	rw

Bit Field	Name	Description
31:14	Reserved	Reserved, must be kept at reset value
13	NAKIEN	NAK interrupt Mask ■ 0 (MASK): Mask NAK Interrupt ■ 1 (NOMASK): No Mask NAK Interrupt
12:9	Reserved	Reserved, must be kept at reset value
8	TXFUDIEN	Fifo Underrun Mask Values: ■ 0 (MASK): Mask Fifo Underrun Interrupt ■ 1 (NOMASK): No Fifo Underrun Interrupt Mask
7	Reserved	Reserved, must be kept at reset value
6	INEPNAKEIEN	IN Endpoint NAK Effective Mask ■ 0 (MASK): Mask IN Endpoint NAK Effective Interrupt ■ 1 (NOMASK): No IN Endpoint NAK Effective Interrupt Mask
5	INTREPMISIEN	IN Token received with EP Mismatch Mask ■ 0 (MASK): Mask IN Token received with EP Mismatch Interrupt ■ 1 (NOMASK): No Mask IN Token received with EP Mismatch Interrupt
4	TXFERINTKIEN	IN Token Received When TxFIFO Empty Mask ■ 0 (MASK): Mask IN Token Received When TxFIFO Empty Interrupt ■ 1 (NOMASK): No IN Token Received When TxFIFO Empty Interrupt
3	TOIEN	Timeout Condition Mask (Non-isochronous endpoints) ■ 0 (MASK): Mask Timeout Condition Interrupt ■ 1 (NOMASK): No Timeout Condition Interrupt Mask
2	AHBERRIEN	AHB Error Mask (AHBErrMsk) ■ 0 (MASK): Mask AHB Error Interrupt ■ 1 (NOMASK): No AHB Error Interrupt Mask
1	EPDISIEN	Endpoint Disabled Interrupt Mask ■ 0 (MASK): Mask Endpoint Disabled Interrupt ■ 1 (NOMASK): No Endpoint Disabled Interrupt Mask
0	TXCIEN	Transfer Completed Interrupt Mask ■ 0 (MASK): Mask Transfer Completed Interrupt ■ 1 (NOMASK): No Transfer Completed Interrupt Mask

34.11.4.5 USBHS Device OUT Endpoint Common Interrupt Mask Register （USBHS_DOUTEPINTEN）

Address offset：0x0814

Reset value：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
rw	rw	rw													

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	NYETIEN	NAKIEN	BERRIEN	Reserved			OPERRIEN	Reserved	B2BSTUPIEN	Reserved	EPDISROTIEN	STUPDNEIEN	AHBERRIEN	EPDISIEN	TXCIEN
		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit Field	**Name**	**Description**
31:14	Reserved	Reserved, must be kept at reset value
14	NYETIEN	NYET interrupt Mask ■ 0 (MASK): Mask NYET Interrupt ■ 1 (NOMASK): No NYET Interrupt Mask
13	NAKIEN	NAK interrupt Mask ■ 0 (MASK): Mask NAK Interrupt ■ 1 (NOMASK): No NAK Interrupt Mask
12	BERRIEN	Babble Error interrupt Mask ■ 0 (MASK): Mask Babble Error Interrupt ■ 1 (NOMASK): No Babble Error Interrupt Mask
11:9	Reserved	Reserved, must be kept at reset value
8	OPERRIEN	OUT Packet Error Mask ■ 0 (MASK): Mask OUT Packet Error Interrupt ■ 1 (NOMASK): No OUT Packet Error Interrupt Mask
7	Reserved	Reserved, must be kept at reset value
6	B2BSTUPIEN	Back-to-Back SETUP Packets Received Mask Applies to control OUT endpoints only. ■ 0 (MASK): Mask Back-to-Back SETUP Packets Received Interrupt ■ 1 (NOMAS
5	Reserved	Reserved, must be kept at reset value
4	EPDISROTIEN	OUT Token Received when Endpoint Disabled Mask Applies to control OUT endpoints only. ■ 0 (MASK): Mask OUT Token Received when Endpoint Disabled Interrupt ■ 1 (NOMASK): No OUT Token Received when Endpoint Disabled Interrupt Mask
3	STUPDNEIEN	SETUP Phase Done Mask Applies to control endpoints only. ■ 0 (MASK): Mask SETUP Phase Done Interrupt ■ 1 (NOMASK): No SETUP Phase Done Interrupt Mask
2	AHBERRIEN	AHB Error ■ 0 (MASK): Mask AHB Error Interrupt ■ 1 (NOMASK): No AHB Error Interrupt Mask
1	EPDIEN	Endpoint Disabled Interrupt Mask ■ 0 (MASK): Mask Endpoint Disabled Interrupt ■ 1 (NOMASK): No Endpoint Disabled Interrupt Mask

Bit Field	Name	Description
0	TXCIEN	Transfer Completed Interrupt Mask ■ 0 (MASK): Mask Transfer Completed Interrupt ■ 1 (NOMASK): No Transfer Completed Interrupt Mask

34.11.4.6 USBHS Device All Endpoints Interrupt Register（USBHS_DAEPINTSTS）

Address offset：0x0818

Reset value：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OUTEP INT15	OUTEP INT14	OUTEP INT13	OUTEP INT12	OUTEP INT11	OUTEP INT10	OUTEP INT9	Reserved								OUTEP INT0
r	r	r	r	r	r	r									r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved							INEPINT8	INEPINT7	INEPINT6	INEPINT5	INEPINT4	INEPINT3	INEPINT2	INEPINT1	INEPINT0
							r	r	r	r	r	r	r	r	r

Bit Field	Name	Description
31:9	OUTEPINT[15:9]	Device OUT Endpoint Interrupt state Each bit represents an OUT endpoint: Bit 31 represents OUT endpoint 15, Bit 25 represents OUT endpoint 9.。
24:17	Reserved	Reserved, must be kept at reset value
16	OUTEPINT0	Device OUT endpoint interrupt 0 status
15:8	Reserved	Reserved, must be kept at reset value
7:0	INEPINT[8:0]	Device IN Endpoint Interrupt state Each bit represents an IN endpoint: Bit 0 represents IN endpoint 0, Bit 8 represents IN endpoint 8.

34.11.4.7 USBHS Device All Endpoints Interrupt Mask Register（USBHS_DAEPINTEN）

Address offset：0x081C

Reset value：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OUTEP IEN15	OUTEP IEN14	OUTEP IEN13	OUTEP IEN12	OUTEP IEN11	OUTEP IEN10	OUTEP IEN9	Reserved								OUTEP IEN0
rw	rw	rw	rw	rw	rw	rw									rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved							INEPIEN8	INEPIEN7	INEPIEN6	INEPIEN5	INEPIEN4	INEPIEN3	INEPIEN2	INEPIEN1	INEPIEN0
							rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit Field	Name	Description
31:25	OUTEPIEN[15:9]	Device OUT Endpoint Interrupt mask Each bit represents an OUT endpoint: Bit 25 represents OUT endpoint 9, Bit 31 represents OUT endpoint 15. The interrupt entry when an interrupt occurs is USB_HS_IRQn.
24:17	Reserved	Reserved, must be kept at reset value
16	OUTEPIEN0	Device OUT Endpoint 0 Interrupt mask OUT Endpoint 0 Interrupt mask. The interrupt entry when an interrupt occurs is USB_HS_IRQn.
15:8	Reserved	Reserved, must be kept at reset value
7:0	INEPIEN [8:0]	Device IN Endpoint Interrupt mask Translation: Each bit represents an IN endpoint: Bit0 represents IN endpoint 0, Bit8 represents IN endpoint 8. The interrupt entry for generating interrupts is USB_HS_IRQn.

34.11.4.8 USBHS Device Threshold Control Register（USBHS_DTHRCTRL）

Address offset：0x0830

Reset value：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				ARPEN	Reserved	RXTHRLEN[8:0]									RXTHREN
				rw						rw					rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved					TXTHRLEN[8:0]									ISOINEPTHREN	NISOINEPTHREN
									rw					rw	rw

Bit Field	Name	Description
31:28	Reserved	Reserved, must be kept at reset value
27	ARPEN	Arbiter Parking Enable This bit controls internal DMA arbiter parking for IN endpoints. If thresholding is enabled and this bit is set to one, then the arbiter parks on the IN endpoint for which there is a token received on the USB. This is done to avoid getting into underrun conditions. By default, arbiter parking is enabled. ■ 0 (DISABLED): Disable DMA arbiter parking ■ 1 (ENABLED): Enable DMA arbiter parking for IN endpoints
26	Reserved	Reserved, must be kept at reset value
25:17	RXTHRLEN[8:0]	Receive Threshold Length (RxThrLen) This field specifies Receive thresholding size in DWORDS. This field also specifies the amount of data received on the USB before

Bit Field	Name	Description
		the core can start transmitting on the AHB. The threshold length has to be at least eight DWORDS. The recommended value for ThrLen is to be the same as the programmed AHB Burst Length （ USBHS_GAHBCFG. BURSTTYP [3:0]）。
16	RXTHREN	Receive Threshold Enable When this bit is set, the core enables thresholding in the receive direction.
15:11	Reserved	Reserved, must be kept at reset value
10:2	TXTHRLEN[8:0]	Transmit Threshold Length This field specifies Transmit thresholding size in DWORDS. This also forms the MAC threshold and specifies the amount of data in bytes to be in the corresponding endpoint transmit FIFO, before the core can start transmit on the USB. The threshold length has to be at least eight DWORDS when the value of AHBThrRatio is 2'h00. In case the AHBThrRatio is non zero the application needs to ensure that the AHB Threshold value does not go below the recommended eight DWORD. This field controls both isochronous and nonisochronous IN endpoint thresholds. The recommended value for ThrLen is to be the same as the programmed AHB Burst Length （USBHS_GAHBCFG. BURSTTYP [3:0]）.
1	ISOINEPTHREN	ISO IN Endpoints Threshold Enable. (ISOThrEn) When this bit is Set, the core enables thresholding for isochronous IN endpoints. ■ 0 (DISABLED): No thresholding ■ 1 (ENABLED): Enables thresholding for isochronous IN endpoints
0	NISOINEPTHREN	Non-ISO IN Endpoints Threshold Enable. When this bit is Set, the core enables thresholding for Non Isochronous IN endpoints. ■ 0 (DISABLED): No thresholding ■ 1 (ENABLED): Enable thresholding for non-isochronous IN endpoints

34.11.4.9 USBHS Device IN Endpoint FIFO Empty Interrupt Mask Register（USBHS_DINEPFEINTEN）

Address offset：0x0834

Reset value：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INEPTXFEIEN[15:0]															

rw

Bit Field	Name	Description
31:16	Reserved	Reserved, must be kept at reset value
15:0	INEPTXFEIEN [15:0]	IN EP Tx FIFO Empty Interrupt Mask Bits (InEpTxfEmpMsk) These bits acts as mask bits for DIEPINTn.TxFEmp interrupt, one bit per IN Endpoint: Bit 0 for IN EP 0, bit 7 for IN EP 7 0：mask 1：nomask

34.11.4.10 USBHS Device Each Endpoints Interrupt Register（USBHS_DEEPINTSTS）

Address offset：0x0838

Reset value：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OUTEP INT15	OUTEP INT14	OUTEP INT13	OUTEP INT12	OUTEP INT11	OUTEP INT10	OUTEP INT9	Reserved								OUTEP INT0
r	r	r	r	r	r	r									r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved							INEPINT8	INEPINT7	INEPINT6	INEPINT5	INEPINT4	INEPINT3	INEPINT2	INEPINT1	INEPINT0
							r	r	r	r	r	r	r	r	r

Bit Field	Name	Description
31:9	OUTEPINT[15:9]	Each OUT Endpoint Interrupt Each bit represents an OUT endpoint: Bit31 represents OUT endpoint 15, Bit25 represents OUT endpoint 9.
24:17	Reserved	Reserved, must be kept at reset value
16	OUTEPINT0	Device OUT endpoint interrupt 0 status
15:8	Reserved	Reserved, must be kept at reset value
7:0	INEPINT[8:0]	Device IN endpoint interrupt status. Each bit represents an IN endpoint: Bit0 represents IN endpoint 0, Bit8 represents IN endpoint 8.

34.11.4.11 USBHS Device Each Endpoints Interrupt Enable Register（USBHS_DEEPINTEN）

Address offset：0x083C

Reset value：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OUTEP IEN15	OUTEP IEN14	OUTEP IEN13	OUTEP IEN12	OUTEP IEN11	OUTEP IEN10	OUTEP IEN9	Reserved								OUTEP IEN0

rw	rw	rw	rw	rw	rw	rw									rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved							INEPIEN8	INEPIEN7	INEPIEN6	INEPIEN5	INEPIEN4	INEPIEN3	INEPIEN2	INEPIEN1	INEPIEN0
							rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit Field	Name	Description
31:25	OUTEPIEN[15:9]	Device OUT endpoint interrupt enable. Each bit represents an OUT endpoint: Bit25 represents OUT endpoint 9, Bit31 represents OUT endpoint 15. The interrupt entry for generating interrupts is USB_HS_IRQn.
24:17	Reserved	Reserved, must be kept at reset value
16	OUTEPIEN0	Device OUT0 endpoint interrupt enable. OUT endpoint 0 enabled. The interrupt entry for generating interrupts is USB_HS_IRQn.
15:8	Reserved	Reserved, must be kept at reset value
7:0	INEPIEN [8:0]	Device IN endpoint interrupt enable. Each bit represents an IN endpoint: Bit0 represents IN endpoint 0, Bit8 represents IN endpoint 8. The interrupt entry for generating interrupts is USB_HS_IRQn.

34.11.4.12 USBHS Device Each IN Endpoint Interrupt Register（USBHS_DINEPxINTEN）(x=[0..8])

Address offset：0x0840 + x×4

Reset value：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		NAKIEN	Reserved				TXFUD IEN	Reserved	INEPNAK EIEN	INTREP MSIEN	TXFERIN TKIEN	TOIEN	AHBERR IEN	EPDISIEN	TXCIEN
		rw					rw		rw	rw	rw	rw	rw	rw	rw

Bit Field	Name	Description
31:14	Reserved	Reserved, must be kept at reset value
13	NAKIEN	NAK interrupt Mask ■ 0 (MASK): Mask NAK Interrupt ■ 1 (NOMASK): No NAK Interrupt Mask
12:9	Reserved	Reserved, must be kept at reset value

Bit Field	Name	Description
8	TXFUDIEN	Fifo Underrun Mask ■ 0 (MASK): Mask Fifo Underrun Interrupt ■ 1 (NOMASK): No Fifo Underrun Interrupt Mask
7	Reserved	Reserved, must be kept at reset value
6	INEPNAKEIEN	IN Endpoint NAK Effective Mask ■ 0(MASK): Mask IN Endpoint NAK Effective ■ 1 (NOMASK): No IN Endpoint NAK Effective Mask
5	INTREPMISIEN	IN Token received with EP Mismatch Mask ■ 0 (MASK): Mask IN Token received with EP Mismatch Interrupt ■ 1 (NOMASK): No Mask IN Token received with EP Mismatch Interrupt
4	TXFERINTKIEN	IN Token Received When TxFIFO Empty Mask ■ 0 (MASK): Mask IN Token Received When TxFIFO Empty Interrupt ■ 1 (NOMASK): No IN Token Received When TxFIFO Empty Interrupt
3	TOIEN	Timeout Condition Mask (Non-isochronous endpoints) ■ 0 (MASK): Mask Timeout Condition Interrupt ■ 1 (NOMASK): No Timeout Condition Interrupt Mask
2	AHBERRIEN	AHB Error Mask ■ 0 (MASK): Mask AHB Error ■ 1 (NOMASK): No AHB Error Mask
1	EPDISIEN	Endpoint Disabled Interrupt Mask ■ 0 (MASK): Mask Endpoint Disabled Interrupt ■ 1 (NOMASK): No Endpoint Disabled Interrupt Mask
0	TXCIEN	Transfer Completed Interrupt Mask ■ 0 (MASK): Mask Transfer Completed Interrupt ■ 1 (NOMASK): No Transfer Completed Interrupt Mask

34.11.4.13 USBHS Device Each OUT Endpoint Interrupt Register（USBHS_DOUTEPxINTEN）(x=[0,9..15])

Address offset：0x0880 + x×4

Reset value：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
rw	rw	rw													

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	NYETIEN	NAKIEN	BERRIEN	Reserved			OPERRIEN	Reserved	B2BSTUPIEN	Reserved	EPDISROTIEN	STUPDNEIEN	AHBERRIEN	EPDISIEN	TXCIEN
		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit Field	Name	Description
31:14	Reserved	Reserved, must be kept at reset value

Bit Field	Name	Description
14	NYETIEN	NYET interrupt Mask ■ 0 (MASK): Mask NYET Interrupt ■ 1 (NOMASK): No NYET Interrupt Mask
13	NAKIEN	NAK interrupt Mask ■ 0 (MASK): Mask NAK Response Received Interrupt ■ 1 (NOMASK): No NAK Response Received Interrupt Mask
12	BERRIEN	Babble Error interrupt Mask ■ 0 (MASK): Mask Babble Error Interrupt ■ 1 (NOMASK): No Babble Error Interrupt Mask
11:9	Reserved	Reserved, must be kept at reset value
8	OPERRIEN	OUT Packet Error Mask ■ 0 (MASK): Mask OUT Packet Error Interrupt ■ 1 (NOMASK): No OUT Packet Error Interrupt Mask
7	Reserved	Reserved, must be kept at reset value
6	B2BSTUPIEN	Back-to-Back SETUP Packets Received Mask Applies to control OUT endpoints only. ■ 0 (MASK): Mask Back-to-Back SETUP Packets Received ■ 1 (NOMASK): No Back-to-Back SETUP Packets Received Mask
5	Reserved	Reserved, must be kept at reset value
4	EPDISROTIEN	OUT Token Received when Endpoint Disabled Mask Applies to control OUT endpoints only. ■ 0 (MASK): Mask OUT Token Received when Endpoint Disabled Interrupt ■ 1 (NOMASK): No OUT Token Received when Endpoint Disabled Interrupt Mask
3	STUPDNEIEN	SETUP Phase Done Mask Applies to control endpoints only. ■ 0 (MASK): Mask SETUP Phase Done Interrupt ■ 1 (NOMASK): No SETUP Phase Done Interrupt Mask
2	AHBERRIEN	AHB Error ■ 0 (MASK): AHB Error Mask ■ 1 (NOMASK): No AHB Error Mask
1	EPDISIEN	Endpoint Disabled Interrupt Mask ■ 0 (MASK): Mask Endpoint Disabled Interrupt ■ 1 (NOMASK): No Endpoint Disabled Interrupt Mask
0	TXCIEN	Transfer Completed Interrupt Mask ■ 0 (MASK): Transfer Completed Mask ■ 1 (NOMASK): No Transfer Completed Mask

34.11.4.14 USBHS Device Control IN Endpoint 0 Control Register（USBHS_DINEP0CTRL）

Address offset：0x0900

Reset value：0x0000 8000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EPEN	EPDIS	Reserved		SNAK	CNAK	TXFNUM[3:0]				STALL	Reserved	EPTYPE[1:0]		NAKSTS	Reserved
rs	Rs			w	w	rw				rs		r		r	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EPACT	Reserved													MPLEN[1:0]	
r														rw	

Bit Field	Name	Description
31	EPEN	Endpoint Enable The application sets this bit to 1 to initiate data transmission on the endpoint. The module clears this bit before triggering any of the following interrupts on this endpoint: – SETUP completed –Endpoint Disabled –Transfer Completed ■ 0 (INACTIVE): No action ■ 1 (ACTIVE): Enable Endpoint
30	EPDIS	Endpoint Disable The application sets this bit to stop transmitting data on an endpoint, even before the transfer for that endpoint is complete. The application must wait for the Endpoint Disabled interrupt before treating the endpoint as disabled. The core clears this bit before setting the Endpoint Disabled Interrupt. The application must Set this bit only if Endpoint Enable is already set for this endpoint. ■ 0 (INACTIVE): No action ■ 1 (ACTIVE): Disabled Endpoint
29:28	Reserved	Reserved, must be kept at reset value
27	SNAK	Set NAK (SNAK) A write to this bit sets the NAK bit for the endpoint. Using this bit, the application can control the transmission of NAK handshakes on an endpoint. The core can also set this bit for an endpoint after a SETUP packet is received on that endpoint.
26	CNAK	Clear NAK A write to this bit clears the NAK bit for the endpoint.
25:22	TXFNUM[3:0]	TxFIFO Number Define the TX FIFO number for the IN endpoint.
21	STALL	STALL Handshake The application can only set this bit, and the core clears it, when a SETUP token is received for this endpoint. If a NAK bit, Global Nonperiodic IN NAK, or Global OUT NAK is set along with this bit, the STALL bit takes priority.

Bit Field	Name	Description
20	Reserved	Reserved, must be kept at reset value
19:18	EPTYPE[1:0]	Endpoint Type（For endpoint 0, this bit field is fixed at 00.） 00：control 01：Synchronization 10：Bulk 11：interruupt
17	NAKSTS	NAK Status Indicates the following: ■ 0: The core is transmitting non-NAK handshakes based on the FIFO status ■ 1: The core is transmitting NAK handshakes on this endpoint. When this bit is set, either by the application or core: - Stop receiving any data on the OUT endpoint - For asynchronous IN endpoints: Even if there is available data in the TX FIFO, the module will stop sending any data through the IN endpoint. - For synchronous IN endpoints: Even if there is available data in the TX FIFO, the module will send data packets with zero length. Regardless of how this bit is set, the module always responds to SETUP data packets with an ACK handshake.
16	Reserved	Reserved, must be kept at reset value
15	EPACT	USB Active Endpoint This bit is always SET to 1, indicating that control endpoint 0 is always active in all configurations and interfaces.
14:2	Reserved	Reserved, must be kept at reset value
1:0	MPLEN[1:0]	Maximum Packet Size Applies to IN and OUT endpoints. The application must program this field with the maximum packet size for the current logical endpoint. ■ 2'b00: 64 bytes ■ 2'b01: 32 bytes ■ 2'b10: 16 bytes ■ 2'b11: 8 bytes

34.11.4.15 USBHS Device Each IN Endpoint Interrupt Register（USBHS_DINEPxCTRL）(x=[1..8])

Address offset：0x0900 + x×20

Reset value：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EPEN	EPDIS	Reserved		SNAK	CNAK	TXFNUM[3:0]				STALL	Reserved	EPTYPE[1:0]		NAKSTS	Reserved

rs	rs			w	w		rw			rs		rw		r	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EPACT	Reserved					MPLEN[10:0]									
rw										rw					

Bit Field	Name	Description
31	EPEN	Endpoint Enable The application sets this bit to 1 to initiate data transmission on the endpoint. The module clears this bit before triggering any of the following interrupts on this endpoint: – SETUP completed –Endpoint Disabled –Transfer Completed ■ 0 (INACTIVE): No action ■ 1 (ACTIVE): Enable Endpoint
30	EPDIS	Endpoint Disable The application sets this bit to stop transmitting data on an endpoint, even before the transfer for that endpoint is complete. The application must wait for the Endpoint Disabled interrupt before treating the endpoint as disabled. The core clears this bit before setting the Endpoint Disabled Interrupt. The application must Set this bit only if Endpoint Enable is already set for this endpoint. ■ 0 (INACTIVE): No action ■ 1 (ACTIVE): Disabled Endpoint
29	SD1PID	Set DATA1 PID (applicable to interrupt and bulk IN endpoints) Software sets this bit to set the DPID bit of the register.
	SODDFRM	Set odd frame (applicable to synchronous IN endpoints) Software sets this bit to set the EOFRM bit of the register.
28	SD0PID	Set DATA0 PID (applicable to interrupt and bulk IN endpoints) Software sets this bit to clear the DPID bit of the register.
	SEVNFRM	Set even frame (applicable to synchronous IN endpoints) Software sets this bit to clear the EOFRM bit of the register.
27	SNAK	Set NAK. The application sets this bit to control the transmission of NAK handshake signal on the endpoint. After receiving a SETUP on the endpoint, the module will also set this bit to 1.
26	CNAK	Clear NAK. Writing to this bit will clear the NAK bit of the endpoint.
25:22	TXFNUM[3:0]	Tx FIFO number. Define the TX FIFO number for the IN endpoint.
21	STALL	STALL handshake

Bit Field	Name	Description
		When used for non-control, non-synchronous IN endpoints (access type rw): The application sets this bit to 1 to make the device respond with STALL to all tokens from the USB host. If the NAK bit, global IN NAK, or global OUT NAK are set simultaneously with this bit, the STALL bit takes priority. When used for control endpoints (access type rs): The application can only write to this bit, and after receiving a SETUP token, USBHS clears this bit. If the NAK bit, global IN NAK, or global OUT NAK are set simultaneously with this bit, the STALL bit takes priority. Regardless of how this bit is set, the module always responds to SETUP data packets with an ACK handshake.
20	Reserved	Reserved, must be kept at reset value
19:18	EPTYPE[1:0]	Endpoint Type（For endpoint 0, this bit field is fixed at 00.） 00：control 01：Synchronization 10：Bulk 11：interruupt
17	NAKSTS	NAK Status Indicates the following: ■ 0: The core is transmitting non-NAK handshakes based on the FIFO status ■ 1: The core is transmitting NAK handshakes on this endpoint. When this bit is set, either by the application or core: Stop receiving any data on the OUT endpoint For asynchronous IN endpoints: Even if there is available data in the TX FIFO, the module will stop sending any data through the IN endpoint. For synchronous IN endpoints: Even if there is available data in the TX FIFO, the module will send data packets with zero length. Regardless of how this bit is set, the module always responds to SETUP data packets with an ACK handshake.
16	EOFRM	Odd/Even frame (applicable to synchronous OUT endpoints) For synchronous transfers, software controls USBHS to send data packets to OUT transactions only in odd or even frames by using this bit. If the parity of the current frame number does not match this bit, USBHS does not store the data packet. 0：Send data only in even frames. 1：Send data only in odd frames.
	DPID	Endpoint data PID (applicable to interrupt or bulk endpoints) In endpoint or bulk transfers, there is a data PID flipping mechanism. Before the transfer starts, software sets this bit by setting SD0PID. According to the data PID flipping mechanism described in the USB protocol, USBHS maintains this bit during the transfer process.。 0：The data packet PID is DATA0. 1：The data packet PID is DATA1.

Bit Field	Name	Description
15	EPACT	USB active endpoint It indicates whether this endpoint is active in the current configuration and interface. After detecting a USB reset, the module clears this bit for all endpoints (except endpoint 0). Upon receiving SetConfiguration and SetInterface commands, the application must program the endpoint registers accordingly and set this bit to 1.
14:11	Reserved	Reserved, must be kept at reset value
10:0	MPLEN	Maximum packet length It defines the maximum packet length of the current logical endpoint data packets, this value is in bytes.

34.11.4.16 USBHS Device Each IN Endpoint Interrupt Register（USBHS_DINEPxINTSTS）(x=[0..8])

Address offset：0x0908 + x×20

Reset value：0x0000 0080

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	NYETIF	NAKIF	BBERRIF	PKDRPSTS	Reserved		TXFUDRIF	TXFEIF	INEPNAKEIF	INEPMISIF	TXFERINTIF	TOUTIF	AHBERRIF	EPDISIF	TXCIF
	rc_w1	rc_w1	rc_w1	rc_w1			rc_w1	r	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1

Bit Field	Name	Description
31:15	Reserved	Reserved, must be kept at reset value
14	NYETIF	NYET Interrupt The core generates this interrupt when a NYET response is transmitted for a non isochronous OUT endpoint. ■ 0 (INACTIVE): No interrupt ■ 1 (ACTIVE): NYET Interrupt
13	NAKIF	NAK Interrupt The core generates this interrupt when a NAK is transmitted or received by the device. <brIn case of isochronous IN endpoints the interrupt gets generated when a zero length packet is transmitted due to un-availability of data in the TXFifo. ■ 0 (INACTIVE): No interrupt ■ 1 (ACTIVE): NAK Interrupt
12	BBERRIF	NAK Interrupt The core generates this interrupt when babble is received for the

Bit Field	Name	Description
		endpoint.
11	PKDRPSTS	Packet Drop Status This bit indicates to the application that an ISOC OUT packet has been dropped. This bit does not have an associated mask bit and does not generate an interrupt.
10:9	Reserved	Reserved, must be kept at reset value
8	TXFUDRIF	Fifo Underrun Applies to IN endpoints only. The core generates this interrupt when it detects a transmit FIFO underrun condition in threshold mode for this endpoint. ■ 0 (INACTIVE): No Fifo Underrun interrupt ■ 1 (ACTIVE): Fifo Underrun interrupt
7	TXFEIF	Transmit FIFO Empty When the TX FIFO is half-empty or completely empty, this flag is set. The TX FIFO empty threshold is determined by USBHS_GAHBCFG.NPTXFETH.
6	INEPNAKEIF	IN Endpoint NAK Effective Applies to periodic IN endpoints only. This bit can be cleared when the application clears the IN endpoint NAK by writing to DIEPCTLn.CNAK. This interrupt indicates that the core has sampled the NAK bit Set (either by the application or by the core). The interrupt indicates that the IN endpoint NAK bit Set by the application has taken effect in the core. This interrupt does not guarantee that a NAK handshake is sent on the USB. A STALL bit takes priority over a NAK bit.
5	INEPMISIF	IN Token Received with EP Mismatch Applies to non-periodic IN endpoints only. Indicates that the data in the top of the non-periodic TxFIFO belongs to an endpoint other than the one for which the IN token was received. This interrupt is asserted on the endpoint for which the IN token was received.
4	TXFERINTIF	IN Token Received When TxFIFO is Empty Applies to non-periodic IN endpoints only. Indicates that an IN token was received when the associated TxFIFO (periodic/non-periodic) was empty. This interrupt is asserted on the endpoint for which the IN token was received. ■ 0 (INACTIVE): No IN Token Received when TxFIFO Empty interrupt ■ 1 (ACTIVE): IN Token Received when TxFIFO Empty Interrupt
3	TOUTIF	Timeout Condition ■ In shared TX FIFO mode, applies to non-isochronous IN

Bit Field	Name	Description
		endpoints only. ■ In dedicated FIFO mode, applies only to Control IN endpoints. ■ In Scatter/Gather DMA mode, the TimeOUT interrupt is not asserted. Indicates that the core has detected a timeout condition on the USB for the last IN token on this endpoint.
2	AHBERRIF	AHB Error ■ 0 (INACTIVE): No AHB Error Interrupt ■ 1 (ACTIVE): AHB Error interrupt
1	EPDISIF	Endpoint Disabled Interrupt Applies to IN and OUT endpoints. This bit indicates that the endpoint is disabled per the application's request.
0	TXCIF	Transfer Completed Interrupt ■ 0 (INACTIVE): No Transfer Complete Interrupt ■ 1 (ACTIVE): Transfer Completed Interrupt

34.11.4.17 USBHS Device IN Endpoint 0 Transfer Size Register（USBHS_DINEP0TXSIZ）

Address offset：0x0910

Reset value：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved											PKTCNT[1:0]		Reserved		
											rw				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									TLEN[6:0]						
									rw						

Bit Field	Name	Description
31:21	Reserved	Reserved, must be kept at reset value
20:19	PKTCNT[1:0]	Packet Count Indicates the total number of USB packets that constitute the Transfer Size amount of data for endpoint 0. ■ IN Endpoints : This field is decremented every time a packet (maximum size or short packet) is read from the TxFIFO. ■ OUT Endpoints: This field is decremented every time a packet (maximum size or short packet) is written to the RxFIFO.
18:7	Reserved	Reserved, must be kept at reset value
6:0	TLEN[6:0]	Transfer Size This field contains the transfer size in bytes for the current endpoint. The transfer size (XferSize) = Sum of buffer sizes across all

Bit Field	Name	Description
		descriptors in the list for the endpoint. In Buffer DMA, the core only interrupts the application after it has exhausted the transfer size amount of data. The transfer size can be set to the maximum packet size of the endpoint, to be interrupted at the end of each packet.

34.11.4.18 USBHS Device Each IN Endpoint Transfer Size Register（USBHS_DINEPxTXSIZ）(x=[1..8])

Address offset：0x0910 + x×20

Reset value：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved	MCNT[1:0]		PKTCNT[9:0]										TLEN[18:16]		
	rw		rw										rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TLEN[15:0]															
rw															

Bit Field	Name	Description
31:21	Reserved	Reserved, must be kept at reset value
30:29	MCNT[1:0]	Multi-packet count. This field describes the number of packets that need to be transmitted within one frame. 01: 1 packet 10: 2 packets 11: 3 packets
28:19	PKTCNT[9:0]	Packet count It indicates the number of data packets included in one data transfer of endpoint x. This field will decrement each time a data packet (maximum size packet or short packet) is read from the TX FIFO.
18:0	TLEN[18:0]	Transfer size It indicates the amount of data included in one data transfer of endpoint x, in bytes. The module interrupts the application only after the application has transmitted this data. The transfer size can be set to the maximum data packet size of the endpoint, interrupting at the end of each data packet. The module decrements this field each time a data packet from external memory is written to the TX FIFO.

34.11.4.19 USBHS Device Each IN Endpoint DMA Address Register（USBHS_DINEPxDMADD）(x=[0..8])

Address offset：0x0914 + x×20

Reset value：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DMADD[31:16]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMADD[15:0]															
rw															

Bit Field	Name	Description
31:0	DMADD	DMAAddr This field holds the start address of the external memory for storing or fetching endpoint data. Note: For control endpoints, this field stores control OUT data packets as well as SETUP transaction data packets. When more than three SETUP packets are received back-to-back, the SETUP data packet in the memory is overwritten.

34.11.4.20 USBHS Device Each IN Endpoint Transmit FIFO Status Register（USBHS_DINEPxTXFSTS）(x=[0..8])

Address offset：0x0918 + x×20

Reset value：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INEPTXFSPCAVL[15:0]															
r															

Bit Field	Name	Description
31:16	Reserved	Reserved, must be kept at reset value
15:0	TXFSPCAVL[15:0]	IN Endpoint TxFIFO Space Avail Indicates the amount of free space available in the Endpoint TxFIFO.

		Values are in terms of 32-bit words. ■ 16'h0: Endpoint TxFIFO is full ■ 16'h1: 1 word available ■ 16'h2: 2 words available ■ 16'hn: n words available

34.11.4.21 USBHS Device Control OUT Endpoint 0 Control Register（USBHS_DOUTEP0CTRL）

Address offset：0x0B00

Reset value：0x0000 8000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EPEN	EPDIS	Reserved		SNAK	CNAK	Reserved				STALL	Reserved	EPTYPE[1:0]		NAKSTS	Reserved
rs	r			w	w	rw				rs		r		r	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EPACT	Reserved													MPLEN[1:0]	
r														r	

Bit Field	Name	Description
31	EPEN	Endpoint Enable The application sets this bit to 1 to initiate data transmission on the endpoint. The module clears this bit before triggering any of the following interrupts on this endpoint: - SETUP Phase Done - Endpoint Disabled - Transfer Complete ■ 0 (INACTIVE): No action ■ 1 (ACTIVE): Enable Endpoint
30	EPDIS	Endpoint Disable The application cannot disable control OUT endpoint 0.
29:28	Reserved	Reserved, must be kept at reset value
27	SNAK	Set NAK A write to this bit sets the NAK bit for the endpoint. Using this bit, the application can control the transmission of NAK handshakes on an endpoint. The core can also set bit on a Transfer Completed interrupt, or after a SETUP is received on the endpoint.
26	CNAK	Clear NAK A write to this bit clears the NAK bit for the endpoint.
25:22	Reserved	Reserved, must be kept at reset value
21	STALL	STALL Handshake The application can only set this bit, and the core clears it, when a SETUP token is received for this endpoint. If a NAK bit or Global

Bit Field	Name	Description
		OUT NAK is Set along with this bit, the STALL bit takes priority. Irrespective of this bit's setting, the core always responds to SETUP data packets with an ACK handshake.
20	Reserved	Reserved, must be kept at reset value
19:18	EPTYPE[1:0]	Endpoint Type (EPType)（Hardcoded to 2'b00 for control.） 00：control 01：Synchronization 10：Bulk 11：interruupt
17	NAKSTS	NAK Status Indicates the following: ■ 1'b0: The core is transmitting non-NAK handshakes based on the FIFO status. ■ 1'b1: The core is transmitting NAK handshakes on this endpoint. When either the application or the core sets this bit, the core stops receiving data, even If there is space in the RxFIFO to accommodate the incoming packet. Irrespective of this bit's setting, the core always responds to SETUP data packets with an ACK handshake.
16	Reserved	Reserved, must be kept at reset value
15	EPACT	USB Active Endpoint (USBActEP) This bit is always set to 1, indicating that a control endpoint 0 is always active in all configurations and interfaces.
14:2	Reserved	Reserved, must be kept at reset value
1:0	MPLEN[1:0]	Maximum Packet Size The maximum packet size for control OUT endpoint 0 is the same as what is programmed in control IN Endpoint 0. ■ 2'b00: 64 bytes ■ 2'b01: 32 bytes ■ 2'b10: 16 bytes ■ 2'b11: 8 bytes

34.11.4.22 USBHS Device Each OUT Endpoint Control Register（USBHS_DOUTEPxCTRL）(x=[9..15])

Address offset：0x0B00 + x×20

Reset value：0x0000 8000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EPEN	EPDIS	SD1PID/ SODDFRM	SD0PID/ SEVNFRM	SNAK	CNAK	Reserved				STALL	Reserved	EPTYPE[1:0]		NAKSTS	EPDPID/ EPEOFRM
rs	rs	w	w	w	w					rs		rw		r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

EPACT	Reserved	MPLEN[10:0]
rw		rw

Bit Field	Name	Description
31	EPEN	Endpoint Enable The application sets this bit to 1 to initiate data transmission on the endpoint. The module clears this bit before triggering any of the following interrupts on this endpoint: - SETUP Phase Done - Endpoint Disabled - Transfer Complete ■ 0 (INACTIVE): No action ■ 1 (ACTIVE): Enable Endpoint
30	EPDIS	Endpoint Disable The application cannot disable control OUT endpoint x.
29	SD1PID	Set DATA1 PID Applicable to interrupt/bulk IN and OUT endpoints, writing to this bit will set the endpoint data PID (DPID) field in this register to DATA1.
	SODDFRM	Set odd frame Applicable only to synchronous IN and OUT endpoints. Software sets this bit to set the EPEOFRM bit of this register, setting it to an odd frame.
28	SD0PID	Set DATA0 PID Applicable to interrupt/bulk IN and OUT endpoints, writing to this bit will set the endpoint data PID (DPID) field in this register to DATA0.
	SEVNFRM	Set even frame Applicable only to synchronous IN and OUT endpoints. Software sets this bit to clear the EPEOF bit of this register, setting it to an even frame.
27	SNAK	Set NAK. The application sets this bit to control the transmission of NAK handshake signal on the endpoint. The module will also set this bit to 1 after receiving a SETUP on the endpoint.
26	CNAK	Clear NAK (CNAK) A write to this bit clears the NAK bit for the endpoint.
25:22	Reserved	Reserved, must be kept at reset value
21	STALL	STALL Handshake The application can only set this bit, and the core clears it, when a SETUP token is received for this endpoint. If a NAK bit or Global OUT NAK is Set along with this bit, the STALL bit takes priority. Irrespective of this bit's setting, the core always responds to SETUP data packets with an ACK handshake
20	Reserved	Reserved, must be kept at reset value
19:18	EPTYPE[1:0]	Endpoint Type (EPType)（Hardcoded to 2'b00 for control.）

Bit Field	Name	Description
		00：control 01：Synchronization 10：Bulk 11：interruupt
17	NAKSTS	NAK Status Indicates the following: ■ 1'b0: The core is transmitting non-NAK handshakes based on the FIFO status. ■ 1'b1: The core is transmitting NAK handshakes on this endpoint. When either the application or the core sets this bit, the core stops receiving data, even If there is space in the RxFIFO to accommodate the incoming packet. Irrespective of this bit's setting, the core always responds to SETUP data packets with an ACK handshake.
16	EPDPID	Endpoint data packet PID Applicable to interrupt/bulk IN and OUT endpoints, indicates the data packet PID currently received/transmitted by the endpoint. The application needs to program bit 29 and bit 28 of this register to set the PID of the first received/transmitted data packet.。 0：DATA0 1：DATA1
	EPEOFRM	Endpoint odd/even frame Applicable to synchronous IN and OUT endpoints, indicates the frame number of the data packet currently received/transmitted by the endpoint. The application needs to program bit 29 and bit 28 of this register to set the frame number of the first received/transmitted data packet. 0：even frame 1：Odd frame
15	EPACT	USB active endpoint It indicates whether this endpoint is active in the current configuration and interface. Upon detecting a USB reset, the module clears this bit for all endpoints (except endpoint 0). After receiving SetConfiguration and SetInterface commands, the application must programmatically configure the endpoint registers accordingly and set this bit to 1.
14:11	Reserved	Reserved, must be kept at reset value
10:0	MPLEN	Maximum packet length It defines the maximum packet length of the current logical endpoint data packets, this value is in bytes.

34.11.4.23 USBHS Device Each OUT Endpoint Interrupt Register（USBHS_DOUTEPxINTSTS）(x=[0,9..15])

Address offset：0x0B08 + x×20

Reset value：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STUPPRXIF	NYETIF	NAKIF	BBERRIF	PKDRPSTS	Reserved		OUTPCKERRIF	Reserved	B2BSTUPRIF	STSPRXIF	OUTTRXEPDISIF	STUPPDNEIF	AHBERRIF	EPDISIF	TXCIF
rc_w1	rc_w1	rc_w1	rc_w1	rc_w1			rc_w1		rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1

Bit Field	Name	Description
31:16	Reserved	Reserved, must be kept at reset value
15	STUPPRXIF	Setup Packet Received Applicable for Control OUT Endpoints in only in the Buffer DMA Mode Set by the controller, this bit indicates that this buffer holds 8 bytes of setup data. There is only one Setup packet per buffer. On receiving a Setup packet, the controller closes the buffer and disables the corresponding endpoint. The application has to reenable the endpoint to receive any OUT data for the Control Transfer and reprogram the buffer start address. Note: Because of the above behavior, the controller can receive any number of back to back setup packets and one buffer for every setup packet is used. ■ 0: No Setup packet received ■ 1: Setup packet received
14	NYETIF	NYET Interrupt The core generates this interrupt when a NYET response is transmitted for a non isochronous OUT endpoint. ■ 0 (INACTIVE): No NYET interrupt ■ 1 (ACTIVE): NYET Interrupt
13	NAKIF	NAK Interrupt The core generates this interrupt when a NAK is transmitted or received by the device. In case of isochronous IN endpoints the interrupt gets generated when a zero length packet is transmitted due to un-availability of data in the TXFifo. ■ 0 (INACTIVE): No NAK interrupt ■ 1 (ACTIVE): NAK Interrupt
12	BBERRIF	NAK Interrupt The core generates this interrupt when babble is received for the

Bit Field	Name	Description
		endpoint.
11	PKDRPSTS	Packet Drop Status This bit indicates to the application that an ISOC OUT packet has been dropped. This bit does not have an associated mask bit and does not generate an interrupt
10:9	Reserved	Reserved, must be kept at reset value
8	OUTPCKERRIF	OUT Packet Error Applies to OUT endpoints Only This interrupt is valid only when thresholding is enabled. This interrupt is asserted when the core detects an overflow or a CRC error for non-Isochronous OUT packet.
7	Reserved	Reserved, must be kept at reset value
6	B2BSTUPRIF	Back-to-Back SETUP Packets Received Applies to Control OUT endpoints only. This bit indicates that the core has received more than three backto-back SETUP packets for this particular endpoint. For information about handling this interrupt,
5	STSPRXIF	Status Phase Received for Control Write This interrupt is valid only for Control OUT endpoints. This interrupt is generated only after the core has transferred all the data that the host has sent during the data phase of a control write transfer, to the system memory buffer. The interrupt indicates to the application that the host has switched from data phase to the status phase of a Control Write transfer. The application can use this interrupt to ACK or STALL the Status phase, after it has decoded the data phase.
4	OUTTRXEPDISIF	OUT Token Received When Endpoint Disabled Applies only to control OUT endpoints. Indicates that an OUT token was received when the endpoint was not yet enabled. This interrupt is asserted on the endpoint for which the OUT token was received.
3	STUPPDNEIF	SETUP Phase Done Applies to control OUT endpoints only. Indicates that the SETUP phase for the control endpoint is complete and no more back-to-back SETUP packets were received for the current control transfer. On this interrupt, the application can decode the received SETUP data packet.
2	AHBERRIF	AHB Error ■ 0 (INACTIVE): No AHB Error Interrupt ■ 1 (ACTIVE): AHB Error interrupt
1	EPDISIF	Applies to IN and OUT endpoints. This bit indicates that the endpoint is disabled per the application's request.

Bit Field	Name	Description
		■ 0 (INACTIVE): No Endpoint Disabled Interrupt ■ 1 (ACTIVE): Endpoint Disabled Interrupt
0	TXCIF	Transfer Completed Interrupt Applies to IN and OUT endpoints. Values: ■ 0 (INACTIVE): No Transfer Complete Interrupt ■ 1 (ACTIVE): Transfer Complete Interrupt

34.11.4.24 USBHS Device OUT Endpoint 0 Transfer Size Register（USBHS_DOUTEP0TXSIZ）

Address offset：0x0B10

Reset value：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved	STUPPCNT[1:0]		Reserved									PKTCNT	Reserved		
	rw														

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									TLEN[6:0]						
									rw						

Bit Field	Name	Description
31	Reserved	Reserved, must be kept at reset value
30:29	STUPPCNT[1:0]	SETUP Packet Count This field specifies the number of back-to-back SETUP data packets the endpoint can receive. ■ 2'b01: 1 packet ■ 2'b10: 2 packets ■ 2'b11: 3 packets
28:20	Reserved	Reserved, must be kept at reset value
19	PKTCNT[1:0]	Packet Count (PktCnt) This field is decremented to zero after a packet is written into the RxFIFO.
18:7	Reserved	Reserved, must be kept at reset value
6:0	TLEN[6:0]	Transfer Size Indicates the transfer size in bytes for endpoint 0. The core interrupts the application only after it has exhausted the transfer size amount of data. The transfer size can be Set to the maximum packet size of the endpoint, to be interrupted at the end of each packet. The core decrements this field every time a packet is read from the RxFIFO and written to the external memory.

34.11.4.25 USBHS Device Each OUT Endpoint Transfer Size Registe （USBHS_DOUTEPxTXSIZ）(x=[9..15])

Address offset：0x0B10 + x×20

Reset value：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved	RXDPID[1:0]/ STUPPCNT[1:0]		PKTCNT[9:0]										TLEN[18:16]		
	rw		rw										rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TLEN[15:0]															
rw															

Bit Field	Name	Description
31:21	Reserved	Reserved, must be kept at reset value
30:29	RXDPID[1:0]	Received data PID Applicable to synchronous OUT endpoints. This is the PID of the last data packet received by this endpoint. 00：DATA0 01：DATA2 10：DATA1 11：MDATA
	STUPPCNT[1:0]	SETUP Packet Count This field specifies the number of back-to-back SETUP data packets the endpoint can receive. ■ 2'b01: 1 packet ■ 2'b10: 2 packets ■ 2'b11: 3 packets
28:19	PKTCNT[9:0]	Packet count It indicates the number of data packets included in one data transfer of endpoint x. This field will decrement each time a data packet (maximum size packet or short packet) is written to the RX FIFO.
18:0	TLEN[18:0]	Transfer size It indicates the amount of data included in one data transfer of endpoint x, in bytes. The module interrupts the application only after the application has transmitted this data. The transfer size can be set to the maximum data packet size of the endpoint, interrupting at the end of each data packet. This field decrements each time data is read from the RX FIFO and written to external memory.

34.11.4.26 USBHS Device OUT Endpoint DMA Address Register（USBHS_DOUTEPxDMADD）(x=[0,9..15])

Address offset：0x0B14 + x×20

Reset value：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DMADD[31:16]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMADD[15:0]															
rw															

Bit Field	Name	Description
31:0	DMADD	DMA address This field defines the DMA start address for the endpoint. DMA uses this address to fetch packet data for IN endpoints or write packet data for OUT endpoints.

34.11.5 USBHS Power control register

34.11.5.1 USBHS Power and Clock Gating Control Register（USBHS_PWRCTRL）

Address offset：0x0E00

Reset value：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								DSLEEP	PHY SLEEP	Reserved		PDMRST	Reserved	GATE HCLK	PHYSTP
								r	r			rw	rw	rw	rw

Bit Field	Name	Description
31:8	Reserved	Reserved, must be kept at reset value
7	DSLEEP	Deep Sleep Indicates that the PHY is in deep sleep when in L1 state.
6	PHYSLEEP	PHY In Sleep Indicates that the PHY is in Sleep State.
5:4	Reserved	Reserved, must be kept at reset value

Bit Field	Name	Description
3	PDMRST	Reset Power-Down Modules This bit is valid only in Partial Power-Down mode. ■ 0 (ON): Power is turned on ■ 1 (OFF): Power is turned off
2	Reserved	Reserved, must be kept at reset value
1	GATEHCLK	Gate Hclk When USBHS is in suspend or session invalid state, the application can set this bit to 1 to stop providing clock to modules other than the AHB bus interface, host interface, and wakeup logic. When USBHS resumes communication or a new session starts, the application clears this bit.
0	PHYSTP	Stop Pclk_PHY When USBHS is in suspend state, session invalid state, or the device is disconnected, the application can set this bit to 1 to stop the PHY clock. When USB resumes or a new session starts, the application clears this bit.

34.11.5.2 USBHS Power and Clock Gating Control Register1（USBHS_PWRCTRL1）

Address offset：0x0E04

Reset value：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								DSLEEP	PHY SLEEP	Reserved		PDMRST	Reserved	GATE HCLK	PHYSTP
								rw	rw			rw	rw	rw	rw

Bit Field	Name	Description
31:4	Reserved	Reserved, must be kept at reset value
3	RAMCLKEN	RAM Clock Gating Enable The application programs RAMGateEn to enable RAM Clock Gating in the ACG feature. ■ 0: Disable RAM Clock Gating (Default) ■ 1: Enable RAM Clock Gating
2:1	CNT	Count to Gate Clock CntGateClk indicates to the Controller how many PHY Clock cycles and AHB Clock cycles of 'IDLE' (no activity) the Controller waits for before Gating the respective PHY and AHB clocks internal to the Controller. ■ 2b00: 64 clocks (Default).

Bit Field	Name	Description
		■ 2b01: 128 clocks
0	GATEN	Enable Active Clock Gating The application programs GateEn to enable Active Clock Gating feature for the PHY and AHB clocks. ■ 0: Disable Active Clock Gating (Default) ■ 1: Enable Active Clock Gating

35 Ethernet (ETH)

Note: Only N32H488, N32H487 series chips support the Ethernet module.

35.1 Introduction

Supports an Ethernet peripheral module. It includes a 10/100Mbps Ethernet MAC. ETH utilizes dedicated DMA to improve the performance of data transmission and reception, and supports standard interfaces for MII, RMII. ETH supports the following standards:

- IEEE 802.3-2015
- IEEE 1588-2008 for precision networked clock synchronization
- AMBA 2.0 for AHB master, AHB slave, and APB slave ports
- RMII specification version 1.2 from RMII consortium

35.2 Main Features

35.2.1 MAC Features

MAC Tx and Rx Features

- Separate transmission, reception, and control interfaces to the application
- 10/100Mbps data transfer rates with the following PHY interfaces:
  - MII interface to directly communicate with a remote MAC
  - RMII interface to communicate with an external Fast Ethernet PHY
- Half-duplex operation:
  - CSMA/CD Protocol support
  - Flow control using backpressure support
- 32-bit data transfer interface on the application side
- Full-duplex flow control operations (IEEE 802.3x Pause packets and Priority flow control)
- Support network statistics with RMON or MIB Counters (RFC2819/RFC2665)
- Support Ethernet packet timestamp as described in IEEE 1588-2002 and IEEE 1588-2008 (64-bit timestamps given in the Tx or Rx status of PTP packet). Both one-step and two-step timestamp is supported in TX direction
- Flexibility to control the Pulse-Per-Second (PPS) output signal
- Optional MDIO (Clause 22 and Clause 45) master interface for PHY device configuration and management

MAC Tx Features

- Preamble and start of packet data (SFD) insertion

- Separate 32-bit status for each packet transmitted from the application
- Automatic CRC and pad generation controllable on a per-packet basis
- Programmable packet length to support Standard or Jumbo Ethernet packets with up to 16 KB of size
- Programmable Inter Packet Gap (40–96 bit times in steps of 8)
- IEEE 802.3x Flow Control automatic transmission of zero-quanta Pause packet when flow control input transitions from assertion to de-assertion (in full-duplex mode)

MAC Rx Features

- Automatic Pad and CRC Stripping options
- Preamble and SFD deletion
- Programmable watchdog timeout limit
- Flexible address filtering modes:
  - four 48-bit perfect (DA) address filters with masks for each byte
  - four 48-bit perfect (SA) address filters with masks for each byte
  - 64 bit Hash filter (optional) for multicast and unicast (DA) addresses
- Support to pass all multi-cast addressed packets
- Promiscuous mode to pass all packets without any filtering for network monitoring
- Pass all incoming packets (as per filter) with a status report
- Additional packet filtering:
  - VLAN tag-based: Perfect match and Hash-based (optional) filtering. Filtering based on either outer or inner VLAN tag is possible
- IEEE 802.1Q VLAN tag detection and option to delete the VLAN tags in received packets
- Detect remote wake-up packets and AMD magic packets
- Forwarding of received Pause packets to the application (in full-duplex mode)

35.2.2 Transaction Layer (MTL) Features

MTL Tx and Rx Common Features

- 32-bit Transaction Layer block (bridges the application and the MAC)
- Optimization for packet-oriented transfers with packets delimiters
- Single-port RAM based synchronous FIFO controllers
- Programmable burst length, up to half the size of the MTL Rx queue or Tx queue size, to support burst data transfer in the EQOS-MTL configuration
- Programmable threshold capability for each queue (default of 64 bytes)

MTL Tx Features

- 2KB transmit FIFO with programmable threshold capability
- One queues on the Transmit path with a common memory for all Tx queues
- Store-and-Forward mechanism or threshold mode (cut-through) for transmission to the MAC
- Automatic retransmission of collision packets in half-duplex mode
- Discard packets on late collision, excessive collisions, excessive deferral, and under-run conditions with appropriate status
- Calculate and insert IPv4 header checksum and TCP, UDP, or ICMP checksum
- Statistics by generating pulses for packets dropped (because of underflow) in the Tx FIFO
- Packet-level control for:
  - Timestamp control
  - CRC and pad control

MTL Rx Features

- 2KB receive FIFO with programmable threshold capability
- One queues on the Receive path with a common memory for all Rx queues
- Insertion of Rx Status vectors into the Rx queue after the EOP transfer (in Threshold mode) and before SOP (in Store-and-Forward mode) in EQOS-MTL configuration
- Programmable Rx queue threshold (default fixed at 64 bytes) in Threshold (or cut-through) mode
- Option to filter all error packets on reception and not forward them to the application in the store-and-forward mode
- Forward the undersized good packets
- Statistics by generating pulses for packets dropped (because of overflow) in the Rx FIFO
- Automatic generation of Pause packet control or backpressure signal to the MAC based on the Rx Queue fill level

35.2.3 DMA Block Features

- 32-bit data transfers
- Separate DMA for transmit and receive paths
- Separate DMA channel in the Transmit path for each queue in MTL
- Byte-aligned addressing for data buffer support
- Dual-buffer (ring) descriptor support
- Descriptor architecture to allow large blocks of data transfer with minimum CPU intervention (each descriptor can transfer up to 32 KB of data)
- Comprehensive status reporting for normal operation and transfers with errors

- Individual programmable burst length for Tx DMA and Rx DMA engines for optimal host bus utilization
- Programmable interrupt options for different operational conditions
- Per-packet Transmit or Receive Complete Interrupt control
- Round-robin or fixed-priority arbitration between the Receive and Transmit engines
- Start and Stop modes
- Separate ports for host CSR access and host data interface

35.2.4 Bus Features

AHB Master Interface Features

- 32-bit, Interfaces with the application through AHB
- Little-endian modes
- Software-selected type of AHB burst (fixed burst, indefinite burst, or mix of both)

AHB Slave Interface Features

- AHB slave interface (32-bit) for CSR access
- Little-endian modes
- All AHB burst types

35.2.5 Monitoring, Testing, and Debugging Features

- Internal loopback from Tx to Rx on the MII for debugging
- DMA states (Tx and Rx) as status bits
- Debug status register that gives status of FSMs in Transmit and Receive data paths and FIFO fill-levels
- Application Abort status bits
- MMC (RMON) module
- Current Tx or Rx Buffer pointer as status registers
- Current Tx or Rx Descriptor pointer as status registers
- Tx or Rx Queues memory accessible through Slave port for debug

35.3 Block Diagram

ETH is mainly composed of the following 4 functional modules:

- **Control and Status Register Module (CSR)**

  The CSR space consists of all internal control registers and status registers of the Ethernet peripheral, accessed through the AHB 32-bit interface.

- **DMA**

Containing one physical channel for reception and one physical channel for transmission. Used to control data transfers between the MAC and system memory via the AHB 32-bit master interface.

- **MAC Transaction Layer (MTL)**

  Used to control the data flow between the application and the MAC.

- **Media Access Control (MAC)**

  Used to implement the Ethernet protocol.

Additionally, it includes a protocol adaptation module to support the RMII PHY Media Independent Interface. The Ethernet peripheral functional block diagram is shown in Figure 35-1.

35.3.1 DMA

The DMA has independent Transmit (Tx) and Receive (Rx) engines, and a CSR space. The Tx engine transfers data from the system memory to the device port (MTL), whereas the Rx engine transfers data from the device port to the system memory.

The DMA engine uses descriptors to efficiently move data from source to destination with minimal application CPU intervention. The DMA is designed for packet-oriented data transfers such as packets in Ethernet. The DMA controller can be programmed to interrupt the application CPU for situations such as Packet Transmit and Receive Transfer completion, and other normal or error conditions.

35.3.2 MTL

The MAC Transaction Layer (MTL) provides the FIFO memory Interface to buffer and regulate the packets between the application system memory and the MAC. It also enables the data to be transferred between the application clock and MAC clock domains. The MTL layer has two data paths: Transmit path and Receive Path. The data path for both directions can be 32-bit.

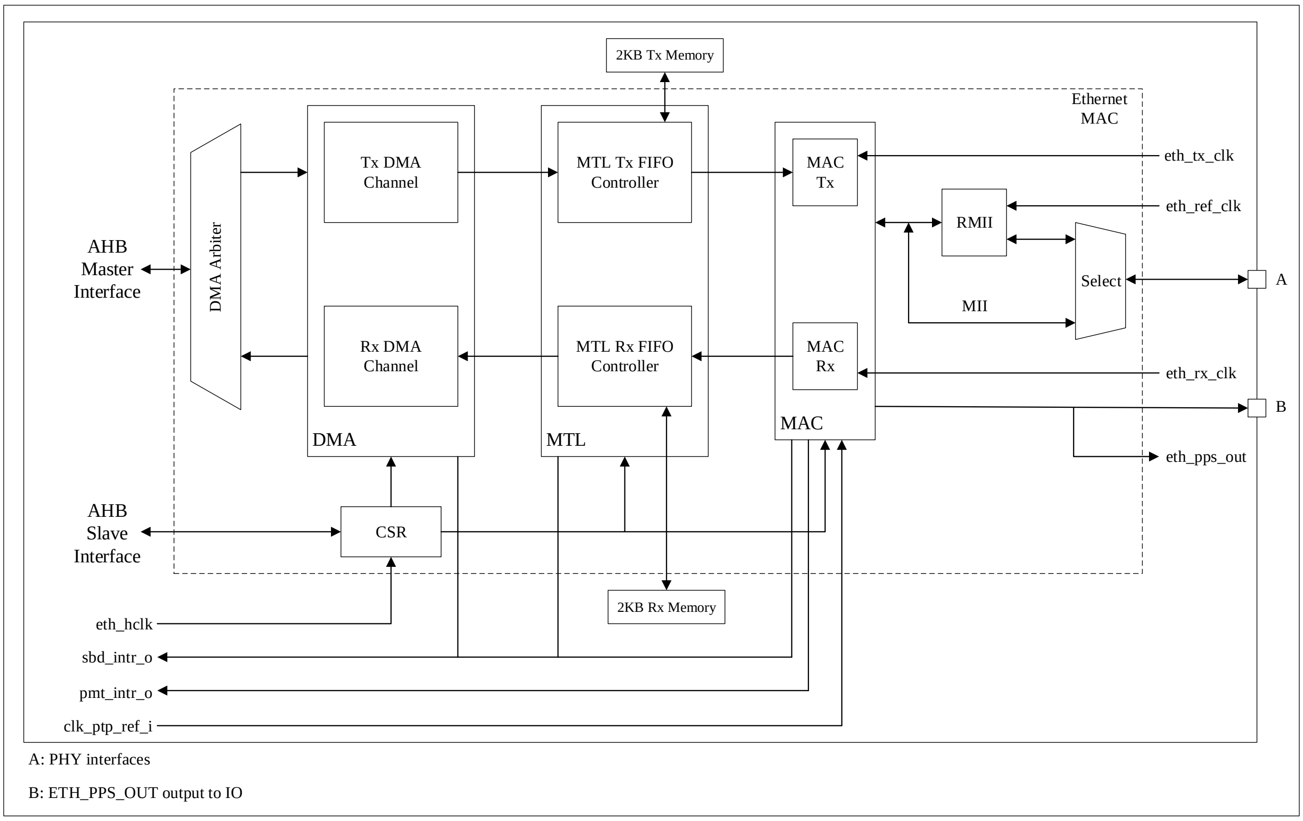
The MTL communicates with the application through Application Transmit Interface (ATI) on the Transmit path and Application Receive Interface (ARI) on the Receive path. The MTL also provides the MAC Control Interface (MCI) as a control path.

35.3.3 MAC

The MAC is responsible for processing Ethernet protocol data packets. In transmit mode, the MAC receives data from the Tx FIFO of the MTL module and then transmits it to the PHY interface. In receive mode, the MAC receives data from the PHY interface and then transfers it to the Rx FIFO of the MTL module.

The MAC supports many interfaces towards the PHY chip. The PHY interface can be selected only once after reset. The MAC communicates with the application side with the MAC Transmit Interface (MTI), MAC Receive Interface (MRI), and MAC Control Interface (MCI).

Figure 35-1 Ethernet Architecture Diagram



35.4 Pins and Signals

This chapter lists the Ethernet input/output signals connected to the GPIO, and also lists and explains the Ethernet internal signals mentioned in Figure 35-1 and the following text. Specific descriptions are shown in Table 35-1 and Table 35-2.

Table 35-1 Ethernet Peripheral Pins

Pin Name	Function Name	Description
PB7, PB8, PE2	ETH_MII_TX_D3	Transmit Data Line 3, for MII use only
PC1	ETH_MDC	Management Data Clock.
PC2	ETH_MII_TX_D2	Transmit Data Line 2, for MII use only
PC3	ETH_MII_TX_CLK	Transmit clock Line, 10Mbps: 2.5MHz, 100Mbps: 25MHz, for MII use only
PA0	ETH_MII_CRS	Carrier Sense, for MII use only
PA1	ETH_MII_RX_CLK/ETH_RMII_REF_CLK	Reference clock, 50MHz, for RMII use only
PA2	ETH_MDIO	Management Data line
PA3	ETH_MII_COL	Collision Detect, for MII use only
PA7, PD8	ETH_MII_RX_DV/ETH_RMII_CRS_DV	MII: Receive Data Valid Signal, RMII: Carrier Sense and Receive Data Valid Signal
PC4, PD9	ETH_MII_RX_D0/ETH_RMII_RX_D0	Receive Data line 0, Shared by MII and RMII
PC5, PD10	ETH_MII_RX_D1/ETH_RMII_RX_D1	Receive Data line 1, Shared by MII and RMII

Pin Name	Function Name	Description
PB0, PD11	ETH_MII_RX_D2	Receive Data line 2, for MII use only
PB1, PD12	ETH_MII_RX_D3	Receive Data line 3, for MII use only
PB10	ETH_MII_RX_ER	Receive error, for MII use only
PB11, PG11	ETH_MII_TX_EN/ETH_RMII_TX_EN	Transmit enable, Shared by MII and RMII
PB12, PG13	ETH_MII_TX_D0/ETH_RMII_TX_D0	Transmit Data Line 0, Shared by MII and RMII
PB13, PG14	ETH_MII_TX_D1/ETH_RMII_TX_D1	Transmit Data Line 1, Shared by MII and RMII
PG8, PB5, PB6	ETH_PPS_OUT	Pulse Per Second

Table 35-2 Ethernet Internal Signals

Name	I/O Type	Description
eth_hclk	Digital Input	AHB clock
sbd_intr_o	Digital Output	Eth global interrupt signal, which is the output signal of logical AND operation on DMA, MTL, and MAC related interrupt signals.
sbd_perch_tx_intr_o[0]	Digital Output	DMA Transmit Channel 0 Interrupt Signal
sbd_perch_rx_intr_o[0]	Digital Output	DMA Receive Channel 0 Interrupt Signal
pmt_intr_o	Digital Output	Interrupt signal when a magic packet or remote wakeup data packet is received, this signal is connected to EXTI Line27
clk_ptp_ref_i	Digital Input	PTP reference clock input signal, the clock signal source is HCLK
eth_tx_clk	Digital Input	MII Transmit clock
eth_ref_clk	Digital Input	RMII Reference clock
eth_rx_clk	Digital Input	MII receive clock
eth_pps_out	Digital Output	Pulse Per Second，The signal output to other peripheral modules such as CAN, the second pulse signal output to GPIO is in uppercase: ETH_PPS_OUT
phy_intr_i	Digital Input	External PHY interrupt input signal, supports connection to any GPIO, after the input front end of EXTI Line11 is multiplexed, it is output to the Ethernet module. The MAC detects the rising edge of this signal and triggers the PHY interrupt

35.5 Function Description

35.5.1 DMA

The DMA and the application communicate through the following two data structures:

- Control and Status registers (CSR)
- Descriptor lists and data buffers

The DMA supports 1 Tx and 1 Rx Descriptor lists (or DMA channels). The base address of each list is written to the respective Tx Descriptor List Address register and Rx Descriptor List Address register. The descriptor list is forward linked and the next descriptor is always considered at a fixed offset to the current one. The offset is controlled by the DSL field of DMA Ch0 Control register. The number of descriptors in the list is programmed in the respective Tx (or Rx) Descriptor Ring Length register. After processing the last descriptor in the list, DMA automatically jumps back to the descriptor in the List Address register to create a descriptor ring.

The descriptor lists reside in the physical memory address space of the application. Each descriptor can point to a maximum of two buffers in the system memory. This enables two buffers to be used, physically addressed, rather than contiguous buffers in memory.

A data buffer resides in the application physical memory space and consists of an entire packet or part of a packet but cannot exceed a single packet. Buffers contain only data. Buffer status is maintained in the descriptor. Data chaining refers to packets that span multiple data buffers. However, a single descriptor cannot span multiple packets. The DMA skips to the data buffer of next packet when EOP is detected.

The ETH supports the ring structure for the DMA descriptor. For more information on the descriptor, see 35.5.12 that describes the descriptor structure and how the DMA accesses the descriptors.

35.5.1.1 DMA Application Bus Burst Access

The DMA engines attempts to transfer data in a burst of maximum size as programmed in the PBL fields of Transmit Control and Receive Control registers of the respective DMA. The Rx and Tx descriptors are always accessed in the maximum possible (limited by PBL or 16 * 8/bus width) burst length for 16 bytes to be read. The burst transfers initiated by the DMA can be split into multiple burst transfers as per the Application Interface protocol (AHB or AXI) requirements and the settings of DMA SysBus Mode register.

The Tx DMA initiates a data transfer only when sufficient space is available in the MTL Tx Queue to accommodate either of the following:

- Bytes corresponding to the configured burst (PBL * bus_width/8)
- Remaining bytes in Tx Buffer without EOP
- Number of bytes till EOP

The Rx DMA initiates a data transfer in the following conditions:

- Sufficient data is available in MTL Rx Queue to accommodate the configured burst
- EOP (when it is less than the configured burst length) is detected in the Rx Queue

The DMA indicates the start address and the number of transfers required to the AHB or AXI master interface. When the AHB or AXI Interface is configured for fixed-length burst, it transfers the data by using the best combination of INCR4, INCR8, or INCR16 and SINGLE transactions. If EOP is reached before the fixed-burst ends on the AHB or AXI interface, dummy transfers are performed in-order to complete the fixed-burst. Otherwise [Bit 0 of DMA SysBus Mode register is reset], the DMA transfers the data using INCR (undefined length) and SINGLE transactions.

When the AHB or AXI interface is configured for address-aligned beats, both DMA engines ensure that the first burst transfer initiated by the AHB or AXI is less than or equal to the size of the configured PBL. Therefore, all subsequent beats start at an address that is aligned to the configured PBL. The DMA can only align the address for beats up to size 16 (for PBL > 16) for AHB interface because it does not support more than INCR16.

35.5.1.2 DMA Application Data Buffer Alignment

The Tx and Rx data buffers do not have any restrictions on start address alignment. For example, in systems with 32-bit memory, the start address for buffers can be aligned to any of the four bytes. However, the DMA always initiates write transfers with address aligned to the bus width and dummy data (old data) in the invalid byte lanes. This typically happens during the transfer of the beginning or end of an Ethernet packet.

The software driver should discard the dummy bytes based on the start address of the buffer and size of the packet.

Table 35-3 Application Data Buffer Alignment Examples

Access	Description
Buffer Read	If the Tx buffer address is 32'h00000FF2 (for 32-bit data bus), and 15 bytes is to be transferred, the DMA reads five full words from address 32'h00000FF0. But when transferring data to the MTL Tx queue, the extra bytes (the first two bytes) are dropped or ignored. Similarly, the last 3 bytes of the last transfer are also ignored. The DMA always ensures that it transfers a full 32-bit data to the MTL Tx queue, unless it is the end of packet.
Buffer Write	If the Rx buffer address is 32'h0000FF2 (for 64-bit data bus) and 16 bytes of a received packet is to be transferred, the DMA writes 3 full words from address 32'h00000FF0. However, the first 2 bytes of the first transfer and the last 6 bytes of the third transfer have dummy data. The DMA considers the offset address only if it is the first Rx buffer of the packet. The DMA ignores the offset address and performs full word writes for the middle and the last Rx buffer of the packet.

35.5.1.3 DMA Buffer Size Calculations

The DMA does not update the size fields in the Tx and Rx descriptors. The DMA updates only the status fields (RDES and TDES) of the descriptors. The driver has to perform the size calculations.

The Tx DMA transfers the exact number of bytes (indicated by buffer size field of TDES2) towards the MAC.

If a descriptor is marked as first (FD bit of TDES3 is set), the DMA marks the first transfer from the buffer as SOP. If a descriptor is marked as last (LD bit of TDES3), the DMA marks the last transfer from that data buffer as EOP to the MTL.

The Rx DMA transfers data to a buffer until the buffer is full or the end of packet is received from the MTL.

When the FD bit of a descriptor is set, the amount of valid data in a buffer is accurately indicated by the buffer size field (programmed in DMA Channel Receive Control register) minus the data buffer pointer offset. The offset is zero when the data buffer pointer is aligned to the data bus width. If a descriptor is marked as last, the buffer may not be full (as indicated by the buffer size in Bits [14:1] of Receive Control register). To compute the amount of valid data in this final buffer, the driver must read the packet length (PL bits of RDES3 [14:0]) and subtract the sum of the buffer sizes of the preceding buffers in this packet. The Rx DMA always transfers the start of next packet with a new descriptor.

Note：Even when the start address of a Rx buffer is not aligned to the data width of system bus, the system should allocate a Rx buffer of a size aligned to the system bus width. For example, if the system allocates a Rx buffer of 1,024 bytes (1 KB) starting from address 0x1000, the software can program the buffer start address in the Rx descriptor to have a 0x1002 offset. The Rx DMA writes the packet to this buffer with dummy data in the first two locations (0x1000 and 0x1001). The actual packet is written from location 0x1002. Therefore, the actual useful space in this buffer is 1,022 bytes, even though the buffer size is programmed as 1,024 bytes, because of the start address offset.

35.5.1.4 DMA Arbiter

The arbiter inside the DMA module performs the arbitration between the Tx and Rx channel accesses to the AHB master interface. The following two types of arbitrations are supported:

- Round-Robin Arbitration

When Bit 1 of the DMA Mode register is reset and both Tx and Rx DMAs are simultaneously requesting for access, the arbiter allocates the data bus in ratio set by Bits[14:12] of DMA Mode register.

- Fixed-Priority Arbitration

  When Bit 1 of the DMA Mode register is set, the Rx DMA always gets priority over the Tx DMA for data access by default. When Bit 11 of DMA Mode register is also set, the Tx DMA gets priority over the Rx DMA

35.5.1.5 Transmit DMA Operation: Default (Non-OSP) Mode

The Tx DMA engine in default mode proceeds as follows:

1 The application sets up the Transmit descriptor (TDES0~TDES3) and sets the Own bit (TDES3[31]) after setting up the corresponding data buffer(s) with Ethernet Packet data.

2 The application advances the Descriptor Tail pointer offset value of the Transmit Channel.

3 While in the Run state, the DMA runs an Arbitration cycle to select the next Tx DMA channel from which the packets requiring transmission should be processed

4 The DMA fetches the descriptor from the application memory.

5 If the DMA detects one of the following conditions, the transmission from that channel is suspended and Bit 2 and Bit 16 of Status Register of corresponding DMA channel are set and the Tx Engine proceeds to step 11:

  - The descriptor is flagged as owned by the application (TDES3 [31] = 1'b0)
  - The Descriptor Tail pointer is equal to the Current Descriptor pointer in Ring Descriptor list Mode
  - An error condition occurs

6 If the acquired descriptor is flagged as owned by the DMA (TDES3 [31] = 1’b1), the DMA decodes the Transmit Data Buffer address from the acquired descriptor.

7 The DMA fetches the Transmit data from the system memory and transfers the data to the MTL for transmission.

8 If an Ethernet packet is stored over data buffers in multiple descriptors, the DMA closes the intermediate descriptor and fetches the next descriptor. Steps 3 through 7 are repeated until the end of Ethernet-packet data is transferred to the MTL.

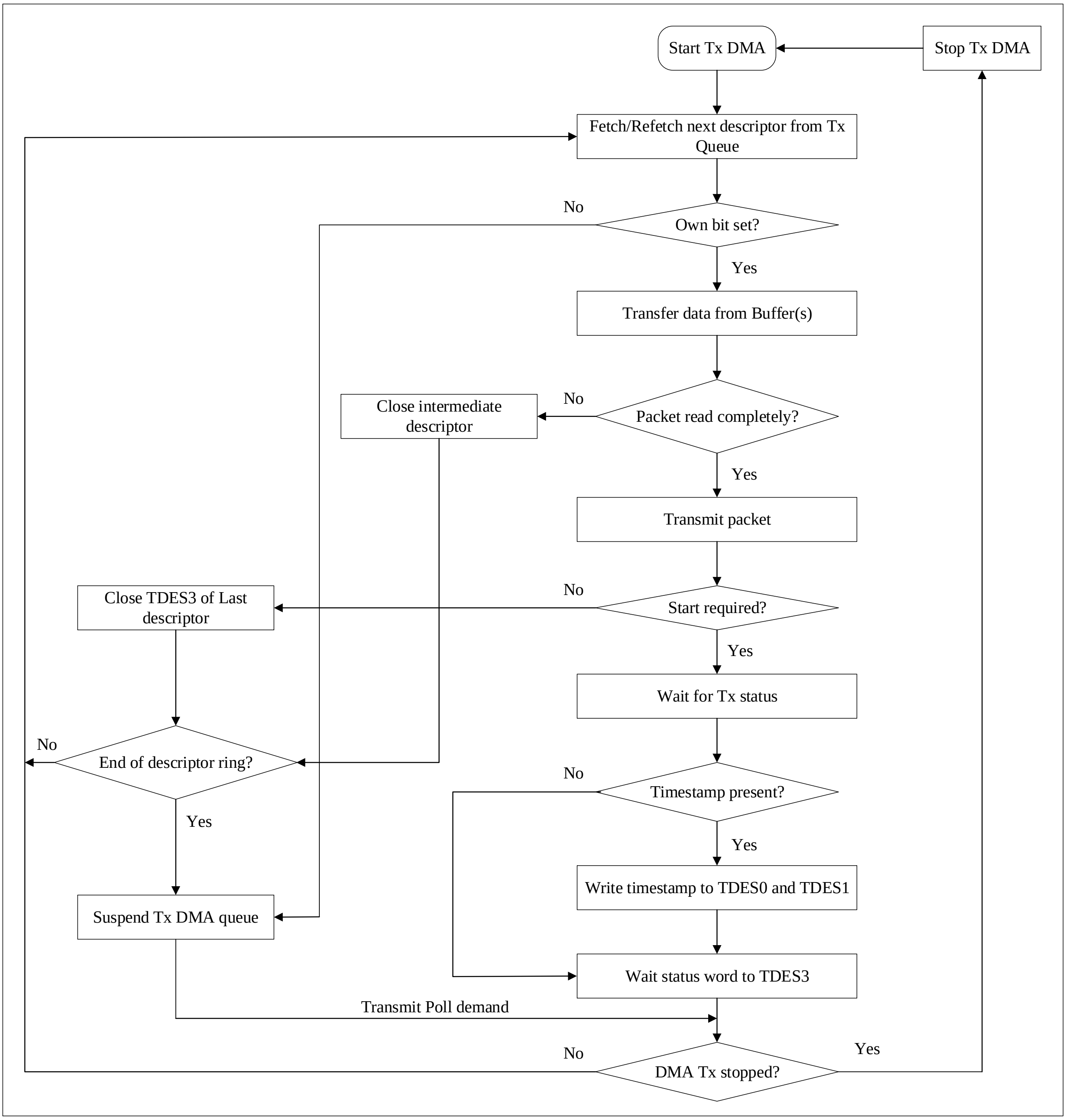
9 When packet transmission is complete, if IEEE 1588 Timestamp feature was enabled for the packet (as indicated in the Tx status), the timestamp value obtained from MTL is written to the Tx descriptor (TDES0 and TDES1) that contains the EOP buffer. The status information is written to this Tx descriptor (TDES3). The application now owns this descriptor because the Own bit is cleared during this step. If timestamp feature is not enabled for this packet, the DMA does not alter the contents of TDES0 and TDES1.

10 Bit 0 of Status register of corresponding channel is set after completing transmission of a packet that has Interrupt on Completion (TDES2[31]) set in its Last Descriptor. The DMA engine returns to step 3.

11 In the Suspend state, the DMA tries to acquire the descriptor again (and thereby return to step 3). A poll demand command is triggered by writing any value to the DMA CH0 TxDesc Tail Pointer when it receives a Transmit Poll demand and the Underflow Interrupt Status bit is cleared. If the application stopped the DMA by clearing Bit 0 of Transmit Control register of corresponding DMA channel, the DMA enters the Stop state

The Tx DMA transmission flow in default mode is shown in Figure 35-2.

Figure 35-2 Transmit DMA Operation in Default Mode



35.5.1.6 DMA Transmit Operation: OSP Mode

In the Run state, if Bit 4 is set in the Transmit Control Register of corresponding DMA channel, the Transmit process can simultaneously acquire two packets without closing the Status descriptor of the first packet. As the Transmit process finishes transferring the first packet, it immediately polls the Transmit Descriptor list for the second packet. If the second packet is valid, the Transmit process transfers this packet before writing the status information of the first packet.

In OSP mode, the Run state Tx DMA operates in the following sequence:

1 The DMA operates as described in step 1 to step 7 of the Tx DMA (default mode).

2 The DMA fetches the next descriptor without closing the last descriptor of previous packet.

3 If the DMA owns the acquired descriptor, the DMA decodes the transmit buffer address in this descriptor. If the DMA does not own the descriptor, the DMA goes into Suspend mode and skips to step 7.

4 The DMA fetches the Transmit packet from the system memory and transfers the packet to the MTL until the EOP data is transferred, closing the intermediate descriptors if this packet is split across multiple descriptors.

5 The DMA waits for the packet transmission status and timestamp of previous packet. When the status is available, the DMA writes the timestamp to TDES0 and TDES1 if such timestamp was captured (as indicated by a status bit). The DMA writes the status, with a cleared Own bit, to the corresponding TDES1, thus closing the descriptor.

6 If Timestamp feature is not enabled for the previous packet, the DMA does not alter the contents of TDES2 and TDES3.

7 The Transmit interrupt is set (if enabled). The DMA fetches the next descriptor and proceeds to step 3 (when Status is normal). If the previous transmission status shows an underflow error, the DMA goes into Suspend mode (step 7).

8 In Suspend mode, if a pending status and timestamp are received from the MTL, the DMA does the following:

- Writes the timestamp (if enabled for the current packet) to TDES0 and TDES1
- Writes the status to the corresponding TDES1
- Sets relevant interrupts and returns to Suspend mode

9 If no status is pending and the application stopped the DMA by clearing Bit 0 of Transmit Control Register of corresponding DMA channel, the DMA enters the Stop state.

10 The DMA can exit Suspend mode and enter the Run state (goes to step 1 or step 2 depending on pending status) only after receiving a Transmit Poll demand in Transmit Descriptor Tail Pointer register of corresponding channel.

Note:

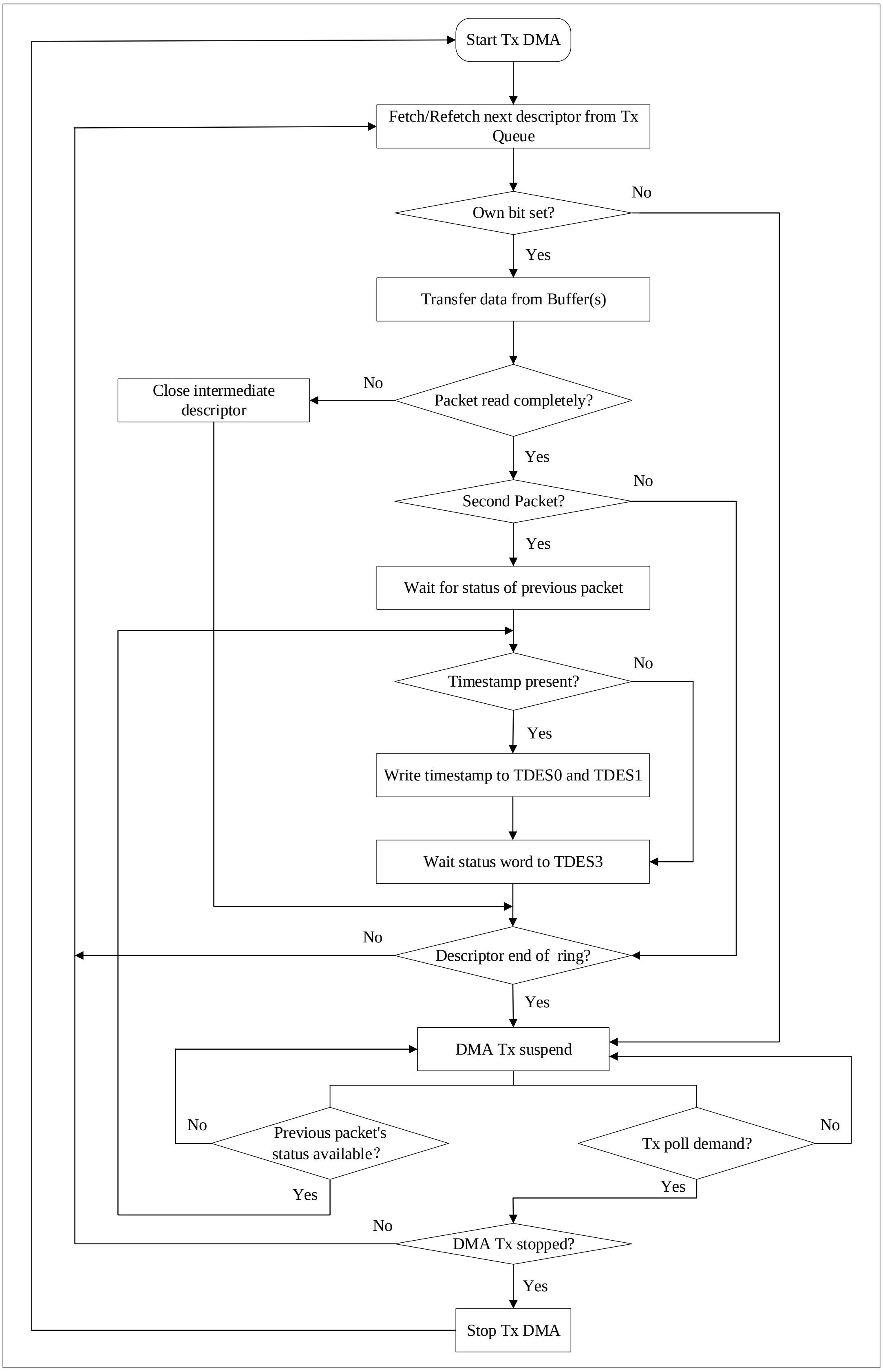
1) The DMA fetches the next descriptor before closing the current descriptor. Therefore, the descriptor ring length must be more than two. Synopsys recommends having a minimum descriptor length of four.

2) In the OSP mode, all except the last descriptor is closed immediately. The last descriptor is closed after the packet is transmitted on the line.

Therefore, to minimize the complexity, the transmission status and only the required control bits are updated in the pending last descriptor of the previous packet.

The basic flow is described in Figure 35-3.

Figure 35-3 Transmit DMA Operation in OSP Mode


Start Tx DMA
Fetch/Refetch next descriptor from Tx Queue
Own bit set?
No
Yes
Transfer data from Buffer(s)
Close intermediate descriptor
No
Packet read completely?
Yes
Second Packet?
No
Yes
Wait for status of previous packet
Timestamp present?
No
Yes
Write timestamp to TDES0 and TDES1
Wait status word to TDES3
No
Descriptor end of ring?
Yes
DMA Tx suspend
No
Previous packet's status available?
Yes
Tx poll demand?
No
Yes
No
DMA Tx stopped?
Yes
Stop Tx DMA

35.5.1.7 Transmit Packet Processing

The Tx DMA expects that the data buffers contain complete Ethernet packets, excluding preamble, pad bytes, and FCS fields. The DA, SA, and Type/Length fields contain valid data. If the Tx Descriptor indicates that the MAC must disable CRC or PAD insertion, the buffer must have complete Ethernet packets (excluding preamble), including the CRC bytes.

Packets can be data-chained and can span several buffers. Packets must be delimited by the First Descriptor (TDES3[29]) and the Last Descriptor (TDES3[28]). As transmission starts, the First Descriptor must have TDES3[29] set. When this occurs, the packet data is transferred from the application buffer to the MTL Tx Queue. Concurrently, if the current packet has the Last Descriptor (TDES3[28]) clear, the Tx Process attempts to acquire the Next Descriptor. The Tx Process expects this descriptor to have TDES3[29] clear. If TDES3[28] is clear, it indicates an intermediary buffer. If TDES3[28] is set, it indicates the last buffer of the packet.

After the last buffer of the packet has been transmitted, the DMA writes back the final status information to the Transmit Descriptor 3 (TDES3) word of the descriptor that has the Last Descriptor Bit set in Transmit Descriptor 3 (TDES3[28]). At this time, if Interrupt on Completion (TDES2[31]) is set, Bit 0 of Status Register of corresponding DMA channel is set, the Next Descriptor is fetched, and the process repeats. The actual packet transmission begins after either of the following:

- The MTL Tx Queue has reached a programmable Transmit threshold (Bits[6:4] of Transmit Operation Mode register of corresponding MTL Transmit Queue)
- A full packet is contained in the FIFO

You can also use the store-and-forward mode (Bit 1 of MTL Transmit Operation Mode Register of a queue). In this mode, descriptors are released (Own bit TDES0[31] clears) when the DMA finishes transferring the packet.

Note: To ensure proper transmission of a packet and the next packet, you must specify a non-zero buffer size for the Transmit descriptor that has the Last Descriptor (TDES3[28]) set.

35.5.1.8 Transmit Polling Suspended

Transmit polling can be suspended by any of the following conditions:

- The DMA detects a descriptor owned by the application (TDES3[31]=0).

  To resume, the driver must give descriptor ownership to the DMA and then issue a Poll Demand command by writing the Tail Pointer register. If the DMA goes into SUSPEND state because of this condition, Bit 15 and Bit 2 of Status Register of corresponding DMA channel are set.

- A packet transmission is aborted when a Transmit error is detected because of underflow.

  The appropriate Transmit Descriptor 3 (TDES3) bit is set. When this condition occurs, the following bits are set and the information is written to Transmit Descriptor 0, causing the suspension:

  - Bit 14 of Status Register of corresponding DMA channel
  - Transmit Underflow bit of corresponding queue in MTL Interrupt Status register

- The DMA detects that the Tail Pointer is equal to the Current descriptor closed by the it.

  To resume, the software driver must modify the Tail Pointer register.

In all conditions, the position in the Transmit List is retained. The retained position is that of the descriptor following the Last Descriptor closed by the DMA. The driver must explicitly issue a Transmit Poll Demand command after rectifying the suspension cause.

35.5.1.9 DMA Receive Operation

In the Receive path, the DMA reads a packet from the MTL receive queue and writes it to the packet data buffers of the corresponding DMA channel. Figure 35-4 shows the reception sequence for Rx DMA engine. The following list describes this sequence:

1 The application sets up the Rx descriptors (RDES0~RDES3) and the Own bit (RDES3[31]). The application must set the correct value in the Receive Descriptor Tail Pointer register of corresponding DMA channel.

2 When Bit 0 of Receive Control register of corresponding DMA channel is set, the DMA enters the Run state. The DMA looks for free descriptors based on the Rx Current Descriptor and Descriptor Tail Pointer register values. If there are no free descriptors, the DMA Channel enters the suspend state and goes to step 11.

3 The DMA fetches the next available descriptor in the ring and decodes the receive data buffer address from acquired descriptors.

4 If IEEE 1588 timestamping is enabled and the timestamp is available for the previous packet, the DMA writes the timestamp (if available) to the RDES0 and RDES1 of current descriptor and sets the CTXT field (RDES3[30]).

5 The DMA processes the incoming packets and places these in the data buffers of acquired descriptor.

6 If the current packet transfer is not complete, the DMA closes the current descriptor as intermediate and goes to step 10.

7 The DMA takes the status of the Receive frame from the MTL and writes the status word to current descriptor with the Own bit cleared and the Last Descriptor bit set.

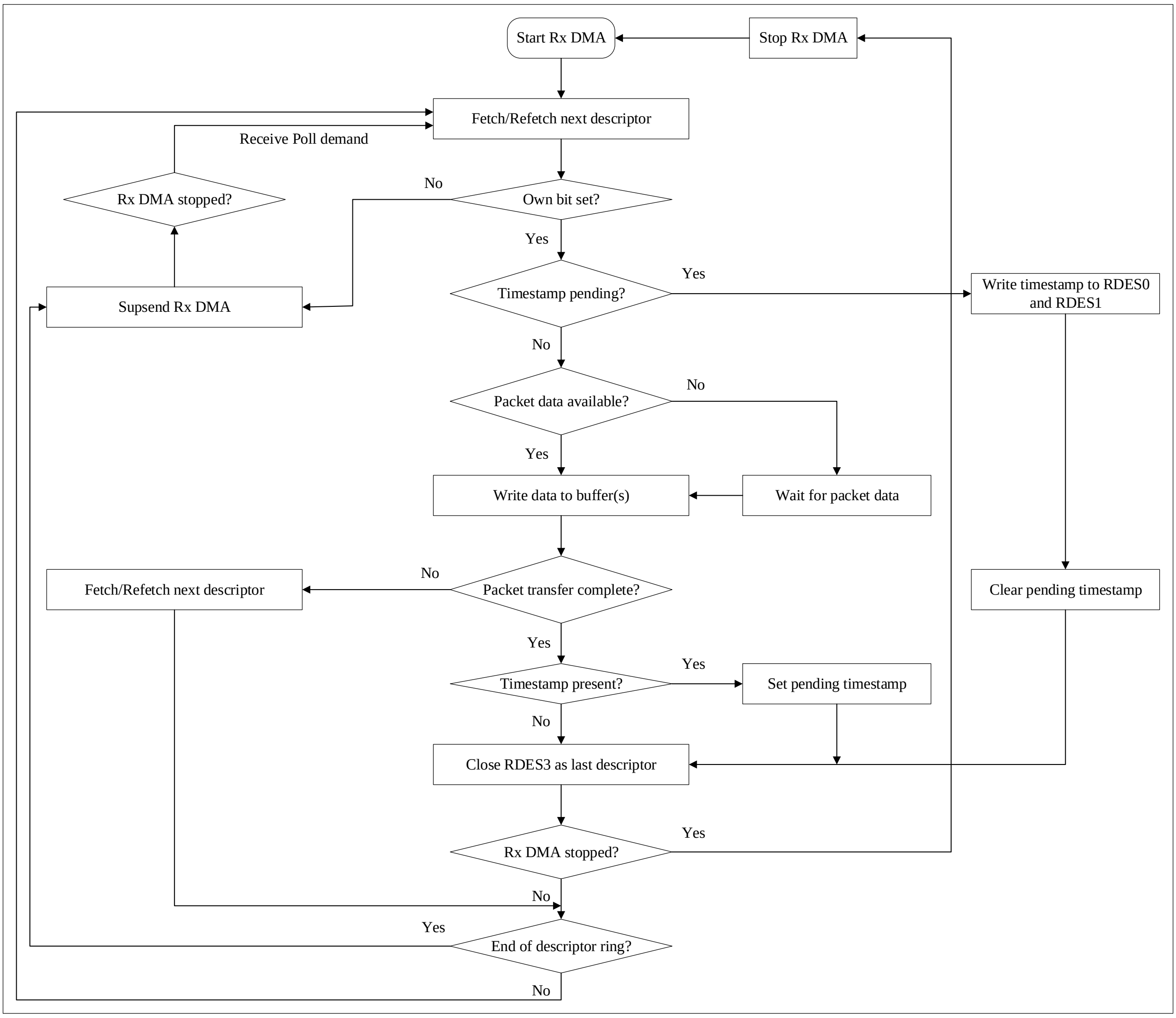
8 The DMA writes the Frame Length to RDES3 and VLAN Tag to RDES0. The DMA also writes the MAC control frame opcode, OAM control frame code, and extended status information (if available) to RDES1 of the last descriptor.

9 If IEEE 1588 Timestamp feature is enabled, the DMA stores the timestamp (if available). The DMA writes the context descriptor after the last descriptor for the current packet (in the next available descriptor).

10 If more descriptors are available in the Rx DMA Descriptor Ring, go to step 3; otherwise, go to the Suspend state (step 11).

11 The Receive DMA exits the Suspend state when a Receive Poll demand is given and the application advances the Receive Tail Pointer register of a channel.

Figure 35-4 Receive DMA Operation



35.5.1.10 Receive Descriptor Acquisition

Descriptor acquisition is attempted if any of the following conditions is satisfied:

- Bit 0 of Receive Control Register of corresponding DMA channel is set immediately after being placed in the Run state.
- The Descriptor Tail pointer register value is ahead of the Current Descriptor acquired by the Rx DMA.
- The controller has completed packet reception, but the current Receive Descriptor is not yet closed.
- A Receive poll demand is issued (update of the Tail Pointer register).

35.5.1.11 Receive Packet Processing

The sequence for processing a Receive packet is as follows:

1 The MAC transfers the received packets to the MTL memory only if the packet passes the address filter.

If the packet fails the address filtering, it is dropped in the MAC block (unless Bit 31 of MAC Packet Filter register is set).

2 If packet size is greater than or equal to configurable threshold bytes set for Rx Queue of MTL, or when the complete packet is written to the queue in the store-and-forward mode, the MTL block requests the DMA block to begin transferring the packet data to the Receive Buffer pointed by the current descriptor.

Packets smaller than 64 bytes, because of collision or premature termination, are removed from the MTL Receive Queue.

3 When the DMA application Interface (AHB/AXI or MDC) becomes ready, it transfers the data and sets the following:

- If the packet fits in a single descriptor, the DMA sets both Last Descriptor (RDES3[28]) and First Descriptor (RDES3[29]).
- If the packets fits into more than one descriptor, the DMA sets the First Descriptor (RDES3[29]) to delimit the packet.

4 The DMA releases the descriptors by resetting the Own (RDES3[31]) bit to 1'b0, either because the Receive buffer filled up or the last segment of the packet is transferred to the Receive buffer. The received packets status is updated in the last descriptor.

5 If Interrupt Enabled on Completion (RDES3[30]) bit is set in any of the Descriptors between the First and Last Descriptor of the Packet and Bit 6 of Interrupt Enable register of corresponding DMA channel is set, the DMA sets Bit 6 of Status register of corresponding DMA channel.

The same process repeats unless the DMA encounters a descriptor flagged as being owned by the application or when there are no more descriptors in the ring. When the DMA finds a descriptor owned by the application and if Bit 7 of Interrupt Enable register of corresponding DMA channel is set, the Receive Process sets Bit 7 of Status register of corresponding DMA channel and then enters the Suspend state. The position in the receive list is retained.

35.5.1.12 Error Response to DMA

For any data transfer initiated by a DMA channel, if the slave replies with an error response, the DMA stops all operations and updates the error bits and the Fatal Bus Error bit in the Status register of corresponding DMA channel. The application can either perform a reset to ETH or re-initialize the DMA descriptor list and start again. The rest of the DMA channels are not affected by such errors.

35.5.2 MTL (MAC Transaction Layer)

Transmit Path: The application module drives all transactions related to the transmit path through ATI. Internal DMA handles all transactions of the transmit path through ATI. The application or internal DMA pushes Ethernet data packets read from the application or system memory into the Tx queue (FIFO). Then, when the queue threshold is reached (threshold mode) or when there is a complete data packet in the queue (store and forward mode), the data packet is popped out and transferred to the MAC. Upon completion of the transmission (EOP), the send status is obtained from the MAC and returned to the application or internal DMA. The default size of the transmit queue is 2Kbytes. The fill level of the queue indicates to the application or internal DMA (using PBL and watermark) so that it can fetch data from the application or system memory with the desired burst start. The application or internal DMA indicates the Start of Packet (SOP) and End of Packet (EOP) as data packet delimiters through the ATI interface.

Receive Path：The MTL Rx module receives the packets from the MAC and pushes them into the Rx Queue. The status (fill level) of the queue is indicated to the application or DMA when it crosses the configured Receive threshold (RTC bits[1:0] of MTL RxQ0 Operation Mode register of corresponding MTL queue), or the complete packet is received. The MTL also indicates the fill level of the queue so that the DMA can initiate preconfigured burst transfers towards the AHB.

35.5.2.1 Transmit Operation

The following two modes of operation trigger reading of the data towards the MAC:

- Threshold mode:

  In Threshold (or cut-through) mode, as soon as the number of bytes in the Queue cross the configured threshold level (or when the end of packet is written before the threshold is crossed), the data is ready to be popped out and forwarded to the MAC. The threshold level is configured by using the TTC bits of MTL TxQ0 Operation Mode register corresponding to an MTL queue.

- Store-and-forward mode:

  In store-and-forward mode, the MTL pops out the packet towards the MAC only when one or more of the following conditions are true:

  - A complete packet is stored in the Queue
  - The Tx FIFO becomes almost full
  - The ATI watermark becomes low

  The watermark becomes low when the requested Queue does not have space to accommodate the requested burst length on the ATI. Therefore, the MTL when operating in the store-and-forward mode allows packet transmission even if the packet length is bigger than the Tx Queue size.

The application can flush complete content of the Tx Queue by setting Bit 0 (FTQ) of Transmit Operation Mode register corresponding to an MTL queue. This bit is self-clearing and initializes the Queue pointers to the default state. If the FTQ bit is set during a packet transfer from the MTL to the MAC, the MTL stops further transfer because the queue is considered to be empty. Therefore, an underflow event occurs at the MAC transmitter.

35.5.2.2 Single-Packet Transmit Operation

The following flow is valid when the core is operating in non-OSP mode, that is, when bit 4 (OSP) of the DMA CH0 TX Control register of the DMA channel is set to 0.

During a Transmit operation, the MTL block is a slave for the DMA controller. The general sequence of events for a Transmit operation is as follows:

1 If the system has data to be transferred, the DMA controller (if enabled) fetches the data from the application through the AHB, AXI, or Native DMA master interface and starts forwarding it to the MTL. The DMA descriptor memory contains the information related to packet control which is used to drive the ATI Control Word. The DMA data buffer memory contains the packet data.

2 The MTL pushes the data received from the DMA into the corresponding queue. This process continues until the EOP is transferred.

3 When the threshold level is crossed or a full packet of data is received into the queue, the MTL reads the packet

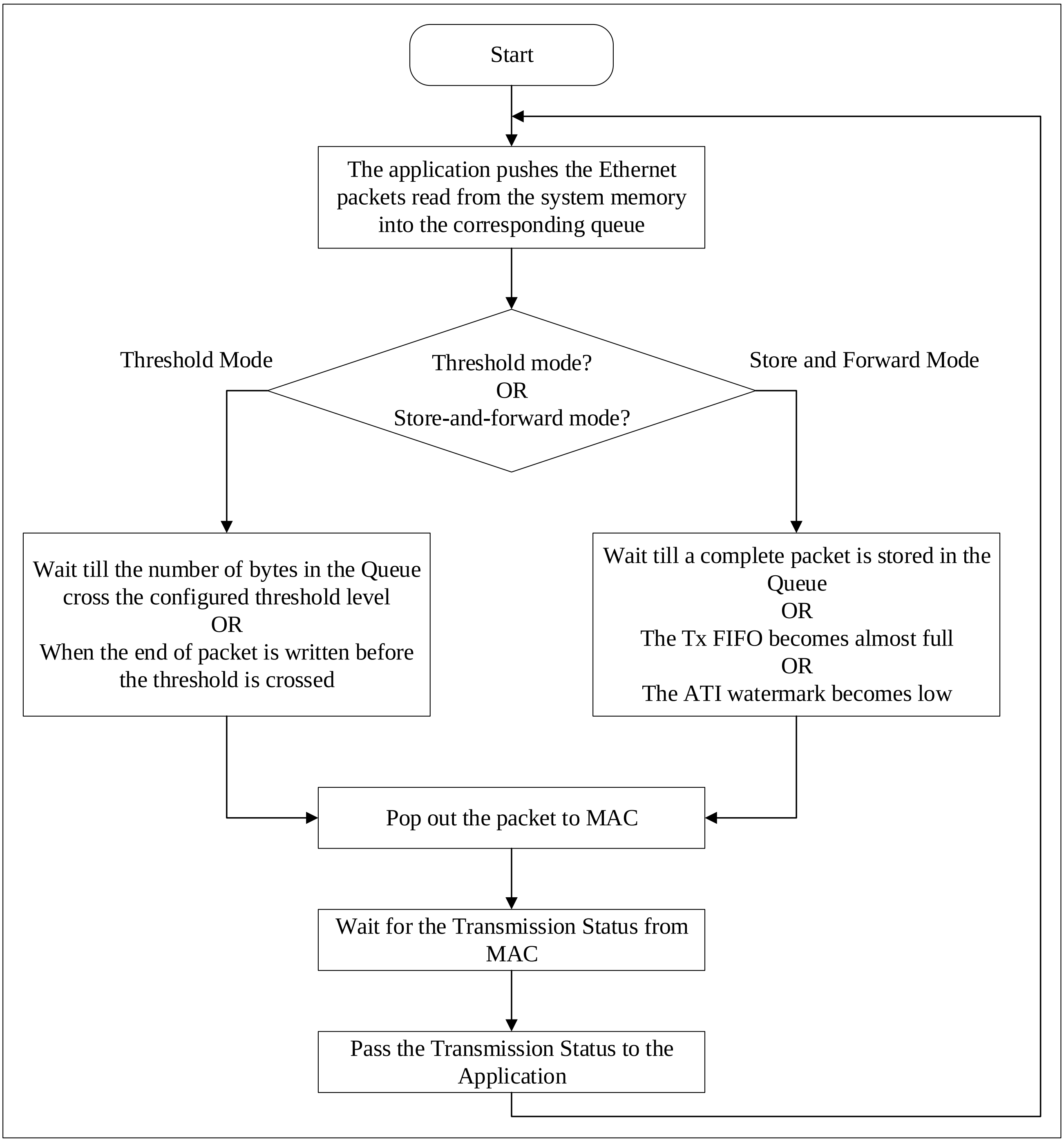
data and drives it to the MAC.

4 The queue controller continues to transfer data from the queue until a complete packet is transferred to the MAC.

5 When the packet transfer is complete, the MTL receives the status from the MAC and notifies the DMA controller.

The single data packet transmission process of MTL is as follows:

Figure 35-5 MTL Single Packet Transmit Flow


Start
The application pushes the Ethernet packets read from the system memory into the corresponding queue
Threshold Mode
Threshold mode? OR Store-and-forward mode?
Store and Forward Mode
Wait till the number of bytes in the Queue cross the configured threshold level OR When the end of packet is written before the threshold is crossed
Wait till a complete packet is stored in the Queue OR The Tx FIFO becomes almost full OR The ATI watermark becomes low
Pop out the packet to MAC
Wait for the Transmission Status from MAC
Pass the Transmission Status to the Application

Note：When the Disable Transmit Status in MTL option is selected or Bit 1 (DTXSTS) of MTL Operation Mode *register is set, the MTL does not provide any status to the DMA.*

35.5.2.3 Two Packets in the Buffer

The following flow is valid when the core is operating in OSP mode, that is, when bit 4 (OSP) of the DMA CH0 TX Control register of the DMA channel is set to 1.

However after fetching a packet DMA instead of waiting for the status, it continues to fetch another packet if available in the system memory. So, MTL can receive the second packet while it is processing the first packet. This flow improves the performance because the DMA can process two packets back to back before waiting for the status of the first packet.

35.5.2.4 Retransmission During Collision

While a packet is being transferred from the MTL to the MAC, a collision event can occur on the MAC line interface in half-duplex mode. The MAC indicates a Retry attempt to the MTL by giving the status even before the EOP is transferred from the MTL. The MTL then enables the retransmission by popping out the packet again from the queue.

After more than 96 bytes are read out towards the MAC, the Queue Controller frees up that space and makes it available to the application or the DMA to push in more data.

This means that the retransmission is not possible after this threshold is crossed or when the MAC indicates a late-collision event.

When a packet transmission is aborted because of underflow and a collision event immediately follows (initiating a retry), retry has higher priority than abort.

35.5.2.5 Transmit Status Word

At the end of transfer of the Ethernet packet to the MAC and after the MAC completes the transmission of the packet, the MTL will provide a valid transmit status. The detailed description of the transmit status is the same as the bit [17:0] in the normal descriptor write back format of TDES3, see 35.5.12 section for details. If IEEE 1588 timestamp is enabled, MTL will also return the 64-bit timestamp of the packet and the transmission status of ATI.

Bit1 (DTXSTS) of the MTL operation mode register can be programmed to disable the dependency of the application or DMA to read the status word from the status FIFO.

35.5.2.6 Receive Operation

During a Receive operation, the MTL is a slave for the MAC. The general sequence of events is as follows:

1 When the MAC receives a packet, it indicates the availability of receive data.

2 The MAC indicates the SOP and EOP delimiters.

3 The MTL accepts the data and pushes it into corresponding Rx queue.

4 After the EOP is transferred, the MAC drives the status word which is also pushed into the corresponding Rx queue by the MTL.

Note: In Threshold (cut-through) mode, the status words are stored after the packet EOP. In store-and-forward mode, the location for the maximum status words are reserved before writing the SOP and the status is written to reserved locations after writing the EOP.

5 If IEEE 1588 timestamp feature is enabled and the 64-bit timestamp is available along with the packet status, it

is pushed into the Rx queue as a part of the status word. Therefore, in 32-bit data bus mode, two additional locations are taken per packet to store the timestamp in the Rx queue.

6 The MTL takes the data out of the queue and sends it to the DMA depending on the mode:

- Threshold Mode:

  In the (default) Threshold mode, the MTL reads the data and indicates its availability to the application or DMA when one of the following occurs:

  - Data bytes equal to the threshold amount are written to the Rx queue (RTC bits[1:0] of MTL RxQ0 Operation Mode register of corresponding MTL queue)
  - A full packet of data is received into the queue

- Store-and-Forward Mode:

  In the store-and-forward mode (when Bit 5 of MTL RxQ0 Operation Mode register of a queue is set to 1), the initial Rx queue locations are reserved for the status words before writing the SOP. A packet is read out only after it is completely written into the Rx queue. In this mode, all error packets are dropped (if configured through Bit 4 of MTL RxQ0 Operation Mode register of a queue) such that only valid packets are read and forwarded to the application.

35.5.2.7 Multi-Packet Receive Operation

In Threshold mode, the packet status is available immediately after the packet data. In store-and-forward mode, the packet data is available after the packet status. The MTL is capable of storing any number of packets into the queue as long as it is not full.

If the MAC receives a packet when the corresponding Rx queue is full, the MTL ignores that packet. In addition, the MTL increments the overflow counter in the MTL RxQ0 Missed Packet Overflow Count register of corresponding queue.

35.5.2.8 Error Handling in Receive Operation

If the MTL Rx queue is full before it receives the EOP data from the MAC, the following happens:

1 An overflow is declared

2 The whole packet (including the status word) is dropped

3 The overflow counter in the DMA (Overflow Counter register of corresponding MTL queue) is incremented.

This is true even if Bit 4 (FEP) of MTL RxQ0 Operation Mode register of corresponding MTL queue is set. If the start address of such a packet has already been transferred to the Read Controller, the rest of the packet is dropped and a dummy EOP is written to the queue along with the status word with overflow status. The status indicates a partial packet because of overflow. In such packets, the Packet Length field is invalid. If the MTL Receive Queue is configured to operate in the store-and-forward mode and the length of the received packet is more than the queue size, overflow occurs and all such packets are dropped.

The MTL Rx Control logic can filter error and undersized packets, if enabled by using the FEP and FUP bits of MTL RxQ0 Operation Mode register of corresponding MTL queue. If the start address of such a packet has already been transferred to the Rx Queue Read Controller, that packet is not filtered. The start address of the packet is transferred to the Read Controller after the packet crosses the receive threshold set by bits[1:0] of MTL RxQ0 Operation Mode register of corresponding MTL queue.

If the MTL Receive Queue is configured to operate in the store-and-forward mode, all error packets can be filtered and dropped. The MTL then stops transferring data to the application (DMA). It internally reads the rest of the packet and drops it. The MTL then starts the transfer of next packet (if available).

35.5.2.9 Receive Status Word Format

As described in the multiple data packet reception operation section, in threshold mode, the Rx data packet status word is sent to the application after the data packet data, while in store and forward mode, the Rx data packet status word is sent to the application before the data packet data. The detailed description of the receive status is the same as the normal RDES3 descriptor write-back format, please refer to the descriptor section for specific information.

35.5.3 MAC Function

35.5.3.1 MAC Transmission

The MAC transmission process is as follows:

1 Transmission is initiated when the MTL application pushes in data with the SOP signal asserted.

2 When the SOP signal is detected, the MAC accepts the data and begins transmitting to the MII. The time required to transmit the packet data to the MII after the application initiates the transmission depends on delay factors such as IPG delay, time to transmit preamble or SFD, and any back-off delays for half-duplex mode.

3 After the EOP is transferred to the MAC, the MAC does one of the following:

- The MAC completes the normal transmission and gives the transmission status to the MTL.
- If a normal collision (in half-duplex mode) occurs during transmission, the MAC gives the Transmit Status with retry bit set to the MTL. The MAC gives the Retry request till one of the following is true:
    - Packet is successfully transmitted
    - Maximum retry requests expire

      When maximum retry requests expire, the MAC aborts the packet transmission with Excessive Collision Transmit Status. The MAC accepts and drops all further data until the next SOP is received. The MTL block should retransmit the same packet from SOP on observing a Retry request (in the Status) from the MAC.
- If any one of the following occurs, the MAC aborts the packet transmission:
    - No carrier (half-duplex mode)
    - Loss of carrier (half-duplex mode)
    - Excessive deferral (half-duplex mode)
    - Late collisions (half-duplex mode)
    - Jabber

  The MAC accepts and drops all further data until the next SOP is received.

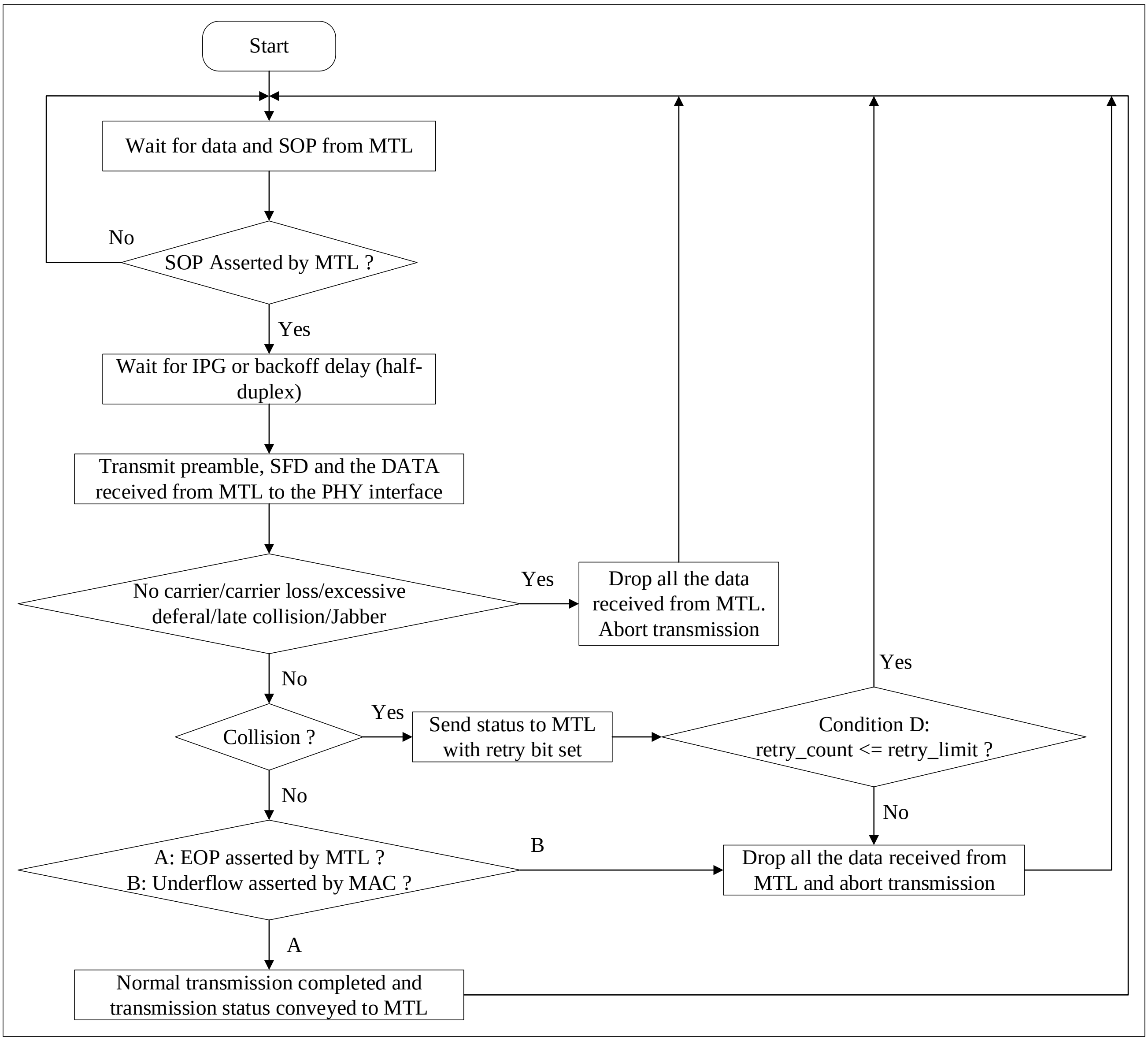
4 The MAC issues an underflow status if the MTL is not able to provide the data continuously during the

transmission. The MAC accepts and drops all further data until the next SOP is received.

5 During the normal transfer of a packet from MTL, if the MAC receives a SOP without getting an EOP for the previous packet, it ignores the SOP and considers the new packet as continuation of the previous packet.

The flowchart of MAC transmission is as follows:

Figure 35-6 Overview of MAC Transmission Process Flow


Start
Wait for data and SOP from MTL
No
SOP Asserted by MTL ?
Yes
Wait for IPG or backoff delay (half-duplex)
Transmit preamble, SFD and the DATA received from MTL to the PHY interface
No carrier/carrier loss/excessive deferal/late collision/Jabber
Yes
Drop all the data received from MTL. Abort transmission
No
Collision ?
Yes
Send status to MTL with retry bit set
Condition D: retry_count <= retry_limit ?
Yes
No
A: EOP asserted by MTL ? B: Underflow asserted by MAC ?
B
Drop all the data received from MTL and abort transmission
A
Normal transmission completed and transmission status conveyed to MTL

35.5.3.2 Transmit Bus Interface Module

The MAC Transmit Bus Interface (TBU) accepts data in 32-bit wide bus and runs on the Tx clock of MII interface.

The Transmit Bus Interface (TBU) module connects the Transmit path of the MAC with an external packet through a FIFO interface.

The TBU module performs the following functions:

- Outputs the (32-bit) Transmit status to the application at the end of normal transmission or collision

- Outputs the Transmit snapshot register value to the timestamp signal and asserts the status valid signal
- Performs the Endian conversion of data bus by swapping the byte lanes and corresponding byte enables
- Converts the input data into an 8-bit bus towards the Transmit Packet Controller

35.5.3.3 Transmit Packet Controller Module

The Transmit Packet Controller (TPC) module consists of eight registers to hold the data and the last data control received from the TBU.

The register provides a buffer between the application and the Transmit Protocol Engine (TPE) to regulate data flow.

When the number of bytes received from the application are less than 60 (DA+SA+LT+DATA), the state machine that interfaces with TBU automatically appends zeros to the packet being transmitted. This is done to make the data length exactly 46 bytes to meet the minimum data field requirement of IEEE 802.3.

The cyclic redundancy check (CRC) for the Frame Check Sequence (FCS) field is calculated before transmission to the TPE module. This value is computed by the CTX module. The TPC module receives the computed CRC and appends it to the data being transmitted to the TPE module. When the MAC is programmed to not append the CRC value to the end of Ethernet packets, the TPC module ignores the computed CRC and transmits only the data received from the TBU module to the TPE module. An exception to this rule is that when the MAC is programmed to append pads for packets (DA+SA+LT+DATA) less than 60 bytes sent by the TBU module, the TPC module appends the CRC at the end of padded packet.

35.5.3.4 Transmit Protocol Engine Module

The Transmit Protocol Engine (TPE) module consists of a Transmit State Machine that controls the operation of Ethernet packet transmission.

The Transmit State Machine of this module contains the following features to meet the IEEE 802.3/802.3z specification:

- Generates preamble and SFD
- Generates jam pattern in the half-duplex mode after normal collision
- Supports jabber timeout
- Supports flow control for the half-duplex mode (backpressure)
- Generates Transmit packet status
- Contains timestamp snapshot logic for IEEE 1588 support

When the TPC module requests the TPE module for a new packet transmission, the Transmit State Machine sends out the preamble and SFD, followed by the data received. The preamble is defined as 7 bytes of 8’b10101010 pattern and the SFD is defined as 1 byte of 8b’10101011 pattern.

The collision window is defined as 1 slot time (512-bit times for 10/100 Mbps Ethernet). The JAM pattern generation is applicable only to half-duplex mode, not to full duplex mode. In full-duplex mode, the Transmit State Machine ignores the collision signal from the PHY.

In MII mode, if a collision occurs any time from the beginning of the packet to the end of the CRC field, the Transmit State Machine sends a 32-bit jam pattern of 32’h55555555 on MII to inform all other stations that a collision has occurred. If the collision is seen during the preamble transmission phase, the Transmit state machine completes the transmission of preamble and SFD, and then sends the JAM pattern.

If the collision occurs after the collision window and before the end of the FCS field (or the end of Burst if the Packet Burst mode is enabled), the Transmit State Machine sends a 32-bit jam pattern and sets the late collision bit in the Transmit packet status.

Note: On the MII interface, the asynchronous collision signal is checked by the transmitter after being double synchronized with the transmit clock domain. This additional delay delays the recognition of collision or late collision events. Even if the COL signal of the MII interface is high before the end of transmission, it will not be recognized as a collision after the complete data packet is sent. Similarly, the COL signal at the MII input end may be recognized as a late collision due to the delay of the synchronizer even at the last byte of the normal collision window.

After enabling packet burst mode, the transmit state machine will send burst data packets without releasing the carrier of the PHY (as long as the TPC module provides the packets). To achieve this, the state machine inserts carrier extension of a minimum IPG period (96 bit times) between packets. The transmit state machine will continue to burst data packets as long as the TPC module can provide additional packets and does not exceed the burst limit of 8192 bytes per burst.

If the packets transmitted by the TPC module exceed 2048 (default) bytes, the TPE module will maintain a jabber timer to stop the transmission of Ethernet packets. When jumbo packets are enabled, the timeout period will be increased to 10240 bytes.

In half-duplex mode, the transmit state machine uses a delay mechanism for flow control (backpressure). When the application requests to stop receiving packets, the transmit state machine will send a (8'h55) 32-byte JAM pattern whenever it detects a received packet, provided that the send flow control is enabled. This will cause a collision, and the remote station will back off. The application requests flow control by setting the BPA bit of the send queue flow control register. If the application requests to send packets, the packets will be scheduled and sent even if backpressure is active. If backpressure is activated for a long time (more than 16 collision events in a row), the remote station will abort the transmission due to excessive collisions.

If the IEEE 1588 timestamp for transmitted packets is enabled, the module takes a snapshot of the system time when the SFD enters the MII bus for transmission. The system time source is internally generated based on the selected configuration.

Note: The Collision input is an asynchronous signal and should be asserted for at least 32-bit times (4 clocks for 1Gbps, and 8 clocks for slower speeds).

35.5.3.5 Transmit Module

The Transmit (STX) module is responsible for scheduling the packet transmission on MII. It provides an enable signal to the TPE module after satisfying the IPG and back-off delays. The STX module performs the following functions:

- Maintains the inter-packet gap between two transmitted packets

  The STX module maintains an idle period of the configured inter-packet gap (IPG bits of MAC Configuration register) between any two transmitted packets. If packets from the TPC arrive at the TPE module sooner than the configured IPG time, the TPE module waits for the enable signal from the STX module before starting the transmission on MII. The STX module starts its IPG counter as soon as the carrier signal of MII goes inactive. At the end of programmed IPG value, the module issues an enable signal to the TPE module in the full-duplex mode.

In the half-duplex mode and when IPG is configured for 96-bit times, the STX module follows the rule of deference specified in the IEEE 802.3, Section 4.2.3.2.1. The module resets its IPG counter if a carrier is detected during the first two-thirds (64-bit times for all IPG values) of the IPG interval. If the carrier is detected during the final one-third of the IPG interval, the STX module continues the IPG count and enables the transmitter after the IPG interval.

- Implements the Truncated Binary Exponential Back-off algorithm

The STX module implements the Truncated Binary Exponential Back-off algorithm when it operates in the half duplex mode.

35.5.3.6 Transmit CRC Generator Module

The MAC Transmit CRC Generator (CTX) module interfaces with the TFC module to generate CRC for the FCS field of the Ethernet packet.

The TPC module sends the packet data and any necessary padding to the CTX module through an 8-bit interface.

The CTX module calculates the 32-bit CRC for the FCS field of the Ethernet packet. The encoding is defined by the following generating polynomial:

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

The CTX module gets the byte data of Ethernet packet from the TPC module (DA + SA + LT + DATA + PAD) qualified with a Data Valid signal. The TPC also indicates to the CTX when to reset the previously calculated CRC and to start the new CRC calculation for the coming packet. The TPC module issues the start command before sending the new packet data for calculation. The calculated CRC is valid on the next clock after the data is received.

In MII mode, this signal is valid every alternate clock.

35.5.3.7 MAC Transmit Flow Control Module

The Transmit Flow Control (FTX) module generates and transmits the Pause packets to the TFC module based on the flow control triggers in full-duplex mode. The TFC module receives the Pause packet from the FTX module, appends the calculated CRC, and sends the packet to the TPE module.

Flow control in full-duplex mode

In full-duplex mode, the Ethernet peripheral uses IEEE 802.3x Pause frames for flow control. The meanings of the fields in the Pause frames are as shown in the table below:

Table 35-4 Pause Packet Fields

Bit Field	Description
DA	Contains the special multicast address
SA	Contains the MAC address 0
Type	0x8808
MAC Control Operation Code	Contains 0001 for IEEE 802.3x Pause Control packets
PT	Contains Pause time specified in the PT field

When the FCB bit is set, the MAC generates and transmits a single Pause packet. If the FCB bit is set again after the Pause packet transmission is complete, the MAC sends another Pause packet irrespective of whether the pause time is complete or not. To extend the pause or terminate the pause prior to the time specified in the previously-transmitted Pause packet, the application should program the Pause Time register with appropriate value and then again set the FCB bit.

If the DZPQ bit in the MAC transmit flow control register is set to 0, the MAC will transmit a Pause frame with a pause time of zero to indicate to the remote end that the receive buffer is ready to accept new packets.

Flow Control in Half-Duplex Mode

In half-duplex mode, the MAC uses the deferral mechanism for the flow control (backpressure). When the application requests to stop receiving packets, the MAC sends a JAM pattern of 32 bytes when it senses a packet reception, provided the transmit flow control is enabled. This results in a collision and the remote station backs off. If the application requests a packet to be transmitted, it is scheduled and transmitted even when the backpressure is activated. If the backpressure is kept activated for a long time (and more than 16 consecutive collision events occur), the remote stations abort the transmission because of excessive collisions.

Based on the settings of the following bits, the Tx channel flow control of Queue 0 is as shown in Table 35-5:

- TFE bit in the MAC transmit flow control register
- DM bit in the MAC configuration register

Table 35-5 Tx MAC Flow Control

TFE	DM	Description
0	x	The MAC transmitter does not perform the flow control or backpressure operation.
1	0	The MAC transmitter performs back-pressure when Bit 0 of MAC Tx Flow Ctrl register is set is 1.
1	1	The MAC transmitter sends the Pause packet when Bit 0 of MAC Tx Flow Ctrl register is set is 1.

35.5.3.8 MAC Reception

A receive operation is initiated when the MAC detects an SFD on MII. The MAC strips the preamble and SFD before proceeding to process the packet. The header fields are checked for filtering and the FCS field used to verify the CRC for the packet. The received packet is stored in a shallow buffer until the address filtering is performed. The packet is dropped in the MAC if it fails the address filter. The following are the functional blocks in the Receive path of the MAC.

35.5.3.9 Receive Protocol Engine Module

The Receive Protocol Engine (RPE) consists of the Receive State Machine which strips the preamble, SFD, and carrier extension of the received Ethernet packet.

The receive process in RPE is as follows:

1. When the RX_DV signal of MII becomes active, the Receive State Machine of RPE starts looking for the SFD field (0xD nibble in MII mode).

   The state machine drops received packets until it detects SFD.

2. When SFD is detected, the state machine begins sending the data of Ethernet packet to the RPC module, beginning with the first byte following the SFD (destination address).

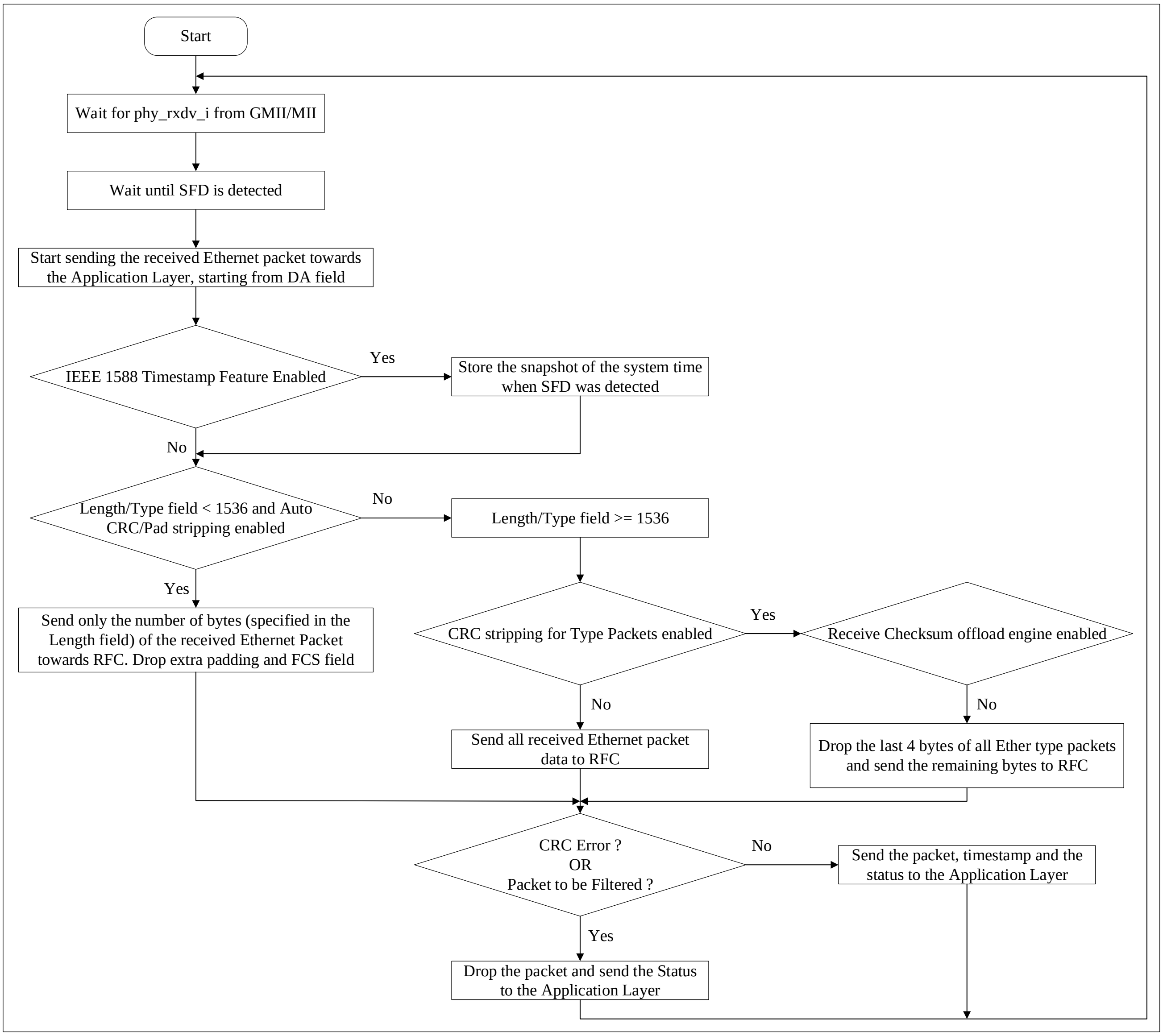
3. If IEEE 1588 Timestamp feature is enabled, the RPE takes a snapshot of the system time at which SFD of any packet is detected on MII. If this packet is not dropped during MAC filtering, the time-stamp is passed to the application. In MII mode, the RPE converts the received nibble data into bytes and forwards the valid packet data to the RFC module.
4. The Receive State Machine of the RPE module decodes the Length/Type field of the receiving Ethernet packet. If the Length/Type field is less than 1,536 and if the MAC is programmed for the Auto CRC/Pad Strip-ping (Bit 20 of the MAC Configuration register), the state machine sends the data of the packet up to the count specified in the Length/Type field and starts dropping bytes (including the FCS field). The state machine of the RPE module decodes the Length/Type field and checks for the Length interpretation.

   If the Length/Type field is greater than or equal to 1,536, the RPE module sends all received Ethernet packet data to the RFC module if you have not enabled the CRC stripping for Type packet in Bit 21 of the MAC Configuration register. However, if you have enabled the CRC stripping for Type packets and not enabled the Receive Checksum Offload Engine, the MAC strips and drops the last 4 bytes of all packets of Ether type before forwarding the packets to the application.
5. By default, the MAC is programmed for watchdog timer to be enabled, that is, packets larger than 2,048 (10,240 if Jumbo Packet is enabled) bytes, (DA + SA + LT + DATA + PAD + FCS) are cut off at the RPE module. In addition, you can use a programmable watchdog timer (Bit 16 of MAC Watchdog Timeout register) to override the fixed timeout of 2,048 or 10,240 bytes. You can disable the watchdog timer by programming Bit 19 of MAC Configuration register. However, even if the watchdog timer is disabled, a packet greater than 32 KB size is cut off and a watchdog timeout status is given.

At the end of every received packet, the RPE module generates received packet status and sends it to the RPC module. Control, missed packet, and filter fail status are added to the receive status in the RPC module.

The flowchart of MAC reception is as follows:

Figure 35-7 MAC Receive Flow Transmission


Start
Wait for phy_rxdv_i from GMII/MII
Wait until SFD is detected
Start sending the received Ethernet packet towards the Application Layer, starting from DA field
IEEE 1588 Timestamp Feature Enabled
Yes
Store the snapshot of the system time when SFD was detected
No
Length/Type field < 1536 and Auto CRC/Pad stripping enabled
No
Length/Type field >= 1536
Yes
Send only the number of bytes (specified in the Length field) of the received Ethernet Packet towards RFC. Drop extra padding and FCS field
CRC stripping for Type Packets enabled
Yes
Receive Checksum offload engine enabled
No
Send all received Ethernet packet data to RFC
No
Drop the last 4 bytes of all Ether type packets and send the remaining bytes to RFC
CRC Error ? OR Packet to be Filtered ?
No
Send the packet, timestamp and the status to the Application Layer
Yes
Drop the packet and send the Status to the Application Layer

35.5.3.10 Receive CRC Module

The MAC Receive CRC (CRX) interfaces with the RPE module to check any CRC error in the packet being received.

The Receive CRC (CRX) interfaces with the RPE module to check any CRC error in the packet being received. This module calculates the 32-bit CRC for received packet that includes the Destination address field through the FCS field.

The encoding is defined by the following generating polynomial:

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

The module gets the data from the RPE module (DA+SA+LT+DATA+PAD+FCS). The RPE module also sends a control signal that indicates the validity of the data. Irrespective of the Auto Pad or CRC strip, the CRX module receives the entire packet to compute the CRC check for received packet.

35.5.3.11 Receive Packet Controller Module

The MAC Receive Packet Controller (RPC) receives the Ethernet packet data and status from the RPE module.

The Receive Packet Controller (RPC) receives the Ethernet packet data and status from the RPE module.

The RPC module consists of a FIFO of parameterized depth (default set to 4 deep and 33-bits wide) and two state machines for writing and reading the FIFO. The FIFO holds the received Ethernet packet data and byte enables, along with a control bit to indicate the last data. The state machines manage the FIFO and provide a packet buffering for the Ethernet packet being received from the RPE module. The following are main functions of the RPC module:

- Converting Data path — converts 8-bit data to 32-bit data to the RBU module
- Packet filtering
- Attaching the calculated IP Checksum input from IPC
- Updating the Receive Status and forwarding it to RBU

If RA bit of the MAC Packet Filter register is set, the RPC module initiates the data transfer to the RBU module when 4 bytes of Ethernet data are received from the RPE module. At the end of the data transfer, the RPC module sends out the received packet status that includes the packet filter bits (SA Filter fail and DA Filter fail) and status from the RPC module. These bits are generated based on the filter-fail signals from the AFM module. This status bit indicates to the application whether the received packet has passed the filter controls (both address filter and Packet Filter controls from CSR). The RPC module does not drop any packet on its own in this mode.

If the RA bit is reset, the RPC module performs packet filtering based on the destination or source address.

If the application does not want to receive any bad packets such as runt, CRC error packets, the application still needs to perform another level of filtering. The RPC module waits to receive the first 14 bytes of received data (type field) from the RPE module. Until then, the module does not initiate any transfers to the RBU module. After receiving the destination or source address bytes, the RPC checks the filter-fail signal from the AFM module for an address match. On detecting filter-fail from AFB, the packet is dropped at the RPC module and not transferred to the application.

On a delayed filter response from the AFM (this can only occur if you change the AFM logic), the RPC module waits until the FIFO is full, and then proceeds with the packet transfer to the RBU module. However, the RPC module still takes the delayed response from the AFM module and if it is a (DA or SA) filter failure, it drops the rest of the packet and sends the Rx Status Word (with zero packet-length, CRC Error and Runt Error bits set) immediately indicating the filter fail. If there is no response from the AFM until EOP is transmitted, the filter fail status in the Rx Status Word is updated accordingly. When the optional PMT module is present and configured for power-down mode, this block drops all received packets and does not forward the packets to the application.

35.5.3.12 Receive Flow Control Module

The Receive Flow Controller (FRX) detects the Pause packet being received and pauses the packet transmission for the delay specified within the received Pause packet. The FRX module is enabled only in the full-duplex mode.

Table 35-6 describes the flow control in the Rx path based on the setting of the following bits:

- RFE bit of MAC Rx Flow Ctrl register
- DM bit of MAC Configuration register

Table 35-6 Rx MAC Flow Control

RFE	DM	Description
0	x	The MAC receiver does not detect the received Pause packets.
1	0	The MAC receiver does not detect the received Pause packets but recognizes such packets as Control packets.
1	1	The MAC receiver detects or processes the Pause packets and responds to such packets by stopping the MAC transmitter.

The following list describes the Rx flow control:

1. The MAC checks the destination address of the received Pause packet for either of the following:
   - Multicast destination address: The DA matches the unique multicast address specified for the control packet (48'h0180C2000001).
   - Unicast destination address: The DA matches the content of the MAC Address Register 0 and the UP bit of MAC Rx Flow Ctrl register is set.

     If the UP bit is set and the MAC processes Pause packets with unicast destination address in addi-tion to the unique multicast address.
2. The MAC decodes the following fields of the received packet:
   - Type field: This field is checked for 16'h8808.
   - Opcode field: This field is checked for 16'h0001 (Pause packet) or 16'h0101 (PFC packet).
   - Priority Enable Vector field: This field is valid only for PFC packets. It is captured to determine the MTL Tx queue corresponding to the received priority.
3. If the byte count of the status indicates 64 bytes and there is no CRC error, the MAC transmitter does one of the following:
   - For 802.3x Pause packets, the MAC pauses the transmission of any data packet for the duration of the decoded Pause Time value multiplied by the slot time (64 byte times).
4. The MAC transfers the received control packet to the application based on the setting of the PCF field in MAC Packet Filter register.

35.5.3.13 Receive Bus Interface Unit Module

The Receive Bus Interface Unit (RBU) converts the 32-bit data received from the RPC module into a 32-bit, 64-bit, or 128-bit FIFO protocol on the Application side. The RBU module interfaces with the application through the MAC receive interface (MRI). This block also performs the endian conversion if the EQOS-CORE is configured for big-endian mode.

If IEEE 1588 Timestamp feature is enabled, the RBU module also outputs the timestamp captured from the received packet, along with the status.

35.5.3.14 Address Filtering Module

The Address Filtering (AFM) module performs the destination and source address checking function on all received packets and reports the address filtering status to the RPC module.

The address checking is done based on different parameters (Packet Filter register) chosen by the application. These parameters are inputs to the AFM module as control signals, and the AFM module reports the status of the address filtering based on the combination of these inputs. The AFM module does not filter the receive packets but reports the status of the address filtering (whether to drop the packet or not) to the RFC module. The AFM module also reports address filter status and whether the received packet is a multicast packet or a broadcast packet.

The AFM module probes the 8-bit receive data path between the RPE module and the RFC module and checks the destination and source address field of each incoming packet. In MII mode, the module takes 14/26 clocks (from the start of packet) to compare the destination or source address of the receiving packet. The AFM module gets the physical (MAC) address of the station and the Multicast Hash table from CSR module for address checking. The CSR module provides the Packet Filter register parameters to AFM.

For other filtering-related functionalities, please refer to the Packet Filtering section.

35.5.4 PHY Interfaces

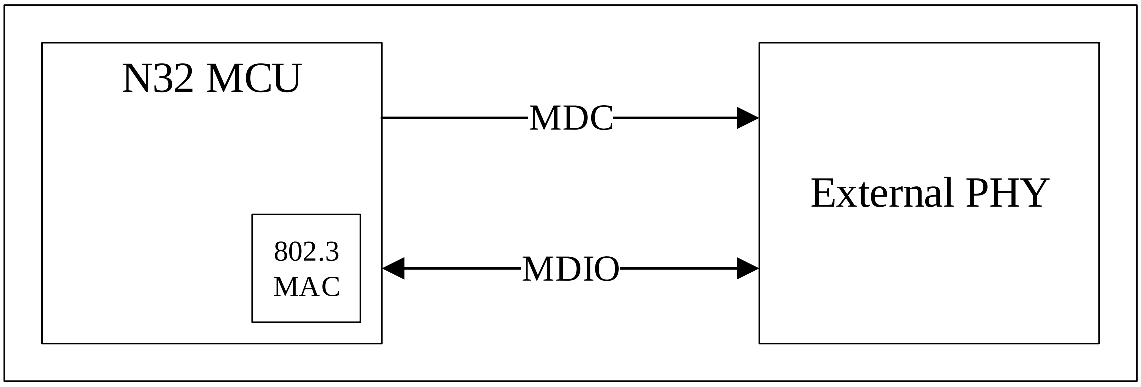
The Ethernet peripheral supports multiple PHY interfaces, including MII and RMII. This chapter describes the SMA (Station Management Agent) interface used for PHY control (management) and the different PHY interfaces.

35.5.4.1 Station Management Agent (SMA)

The application can access the PHY registers through the Station Management Agent (SMA). SMA is a two-wire Station Management interface (MIM).

The SMA module supports access to up to 32 PHYs. The application can address any one of the 32 registers in any one of the 32 PHYs. However, only one register in one PHY can be addressed at a time. The application sends control data to the PHY and receives status information from the PHY through the SMA module.

Figure 35-8 SMA Interfaces



For MIM accesses, the maximum operating frequency of the MDC is 2.5 MHz, as specified in the IEEE 802.3. In the ETH core, the MDC clock is derived from the application clock or CSR clock, using a divider-counter. The divide factor depends on the clock range setting (CR field) in the MAC MDIO Address register.

The MDIO frame structure is as follows:

Table 35-7 MDIO Clause 45 Frame Structure

Field	Description
IDLE	The MDIO line is in tri-state; there is no clock on MDC.
PREAMBLE	32 continuous bits of value 1
START	Start of packet is 2’b00
OPCODE	■ 2’b00 ■ 2’b01 ■ 2’b10 ■ 2’b11
PHY ADDR	5-bit address select for one of 32 PHYs
DEV ADDR	5-bit address select for one of 32 devices
TA	Turnaround ■ 2’bZ0: Read and post-read increment address ■ 2’b10: Write and address MDIO accesses Where Z is the tri-state level
DATA/ADDRESS	16-bit value: For an address cycle (OPCODE = 2'b00), this frame contains the address of the register to be accessed on the next cycle. For the data cycle of a write frame, this field contains the data to be written to the register. For read or post-read increment address frames, this field contains the contents of the register read from the PHY. ■ In address and data write cycles, the ETH drives the MDIO line during the transfer of these 16 bits. ■ In read and post-read increment address cycles, the PHY drives the MDIO line during the transfer of these 16 bits.

The frame structure for Clause 22 frames is also supported. The C45E bit in the MAC MDIO Address register can be programmed to enable Clause 22 or Clause 45 mode of operation. Table 35-8 shows the Clause 22 frame format.

Table 35-8 MDIO Clause 22 Frame Structure

Field	Description
IDLE	The MDIO line is in tri-state; there is no clock on MDC.
PREAMBLE	32 continuous bits of value 1
START	Start of packet is 2’b01
OPCODE	■ 2’b01 for Write ■ 2’b10 for Read
PHY ADDR	5-bit address select for one of 32 PHYs
DEV ADDR	5-bit address to select the register within each MMD
TA	Turnaround ■ 2’bZ0: Read and post-read increment address ■ 2’b10: Write and address MDIO accesses Where Z is the tri-state level
DATA/ADDRESS	Any 16-bit value: ■ In a write operation, the ETH drives MDIO. ■ In read operation, the PHY drives MDIO.

MII Management Write Operation

After the Station Management Agent receives the PHY address and the write data from the MAC CSR module, the SMA starts a Write operation to the PHY registers.

The write operation process is as follows:

1. Set bits [3:2] of the MAC MDIO Address Register to 2'b01 and bit 0 of the device.
2. The MAC CSR module transfers the PHY address, the register address within the PHY, and the data to be written (MAC MDIO Data Register) to the SMA to initiate the write operation to the PHY register.
3. The SMA module initiates the write operation on the GMII management interface using the management data packet format specified in the GMII specification (according to IEEE 802.3-2002 Section 22.2.4.5).
4. When the SMA module starts a write operation on the MDIO, the write data packet is transmitted over the MDIO lines. The MAC drives the MDIO lines throughout the duration of the packet. The "Busy" bit is set to 1 until the write operation is completed. During this time (while the "Busy" bit is high), the CSR ignores write operations to the MAC MDIO Address Register or MAC MDIO Data Register.
5. After the write operation is completed, the SMA will notify the CSR.
6. CSR resets busy bit.

MII Management Read Operation

After the Station Management Agent receives the PHY address and the register address in the PHY from the

MAC CSR module, the SMA initiates a Read operation to the PHY registers.

The flow of a Management read operation is as follows:

1 Set bits [3:2] to 2’b11 and Bit 0 in the MAC MDIO Address register

2 The MAC CSR module transfers the PHY address and the register address in PHY to the SMA to initiate a Read operation in the PHY registers

3 The SMA module starts a Read operation on the GMII Management Interface using the Management

4 Packet Format specified in the GMII specifications (as per IEEE 802.3-2002, Section 22.2.4.5).

5 When the SMA module starts a Read operation on the MDIO, the CSR ignores the Write operations to the MAC MDIO Address or MAC MDIO Data register during this period (the busy bit is high) and the transaction is completed without any error on the MCI interface.

6 When the Read operation is complete, the SMA indicates this to the CSR.

7 The CSR resets the busy bit and updates the MAC MDIO Data register with the data read from the PHY.

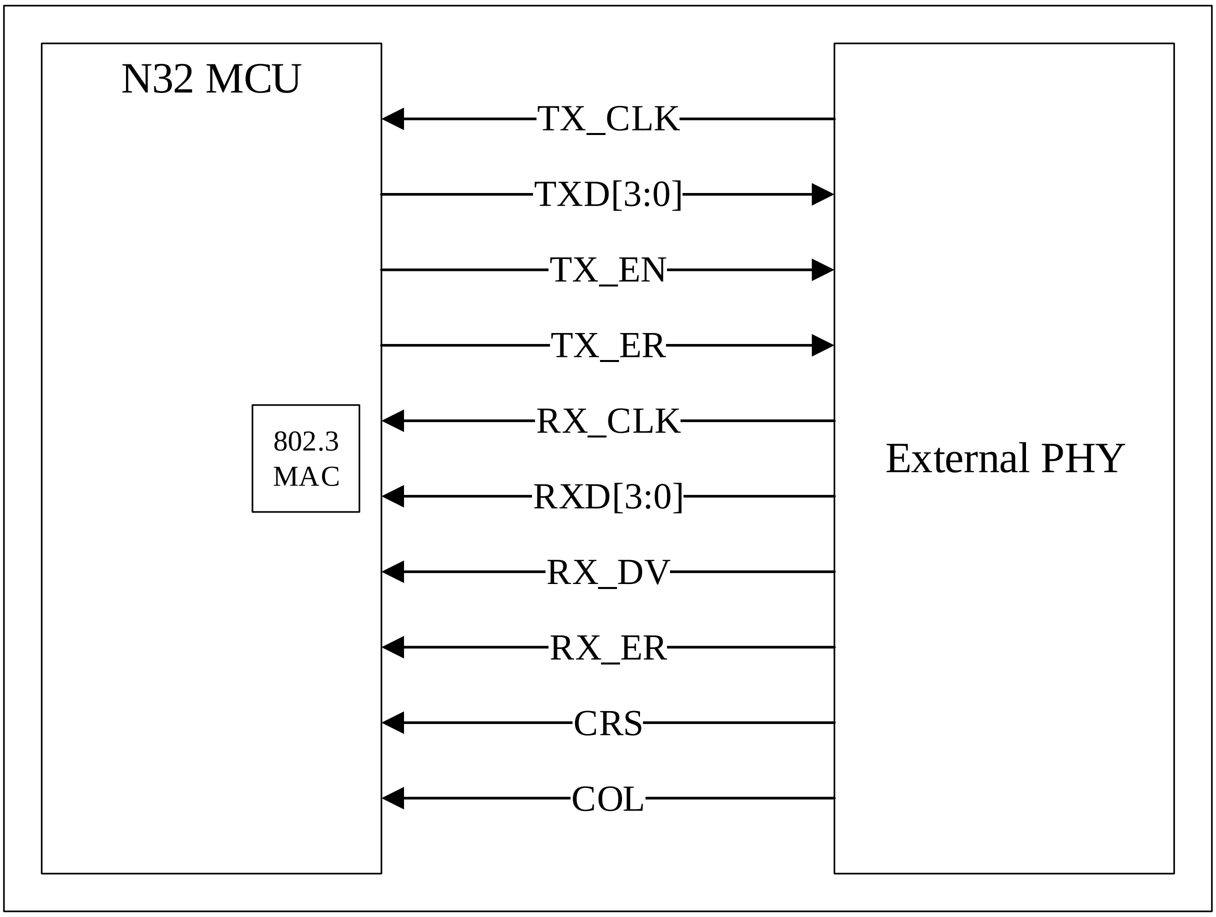
For more information about the communication from the application to the PHYs, see the “Reconciliation Sublayer” and “Media Independent Interface Specifications” sections of the IEEE 802.3z, 1000BASE Ethernet.

35.5.4.2 Media Independent Interface (MII)

The MII defines the interconnection between the MAC sublayer and PHY for data transmission at 10Mb/s and 100Mb/s.

The signals of MII are as shown in the following diagram:

Figure 35-9 MII Interface



- TX_CLK: Continuous clock signal. Provides timing reference for Tx data transmission. The nominal frequency is 2.5MHz for 10Mbit/s and 25MHz for 100Mbit/s.
- TXD[3:0]: Transmit data. TXD is a bundle of 4 data signals synchronously driven by the MAC sublayer, which are valid only when the TX_EN signal is active (valid data). TXD[0] is the least significant bit, and TXD[3] is the most significant bit. The transmission data has no impact on the PHY when TX_EN is inactive.
- TX_EN: Transmit enable signal. Indicates that the MAC is currently sending half-bytes to the MII. It must be synchronized with the first half-byte of the preamble (TX_CLK) and remain active until all bytes to be transmitted are sent to the MII.
- TX_ER (optional): Only required for Energy-Efficient Ethernet (EEE). Transmit errors are indicated by the inverted CRC. The remote station can detect transmit errors through incorrect CRC detection.
- RX_CLK: Continuous clock signal. Provides timing reference for Rx data transmission. The nominal frequency is 2.5MHz for 10Mbit/s and 25MHz for 100Mbit/s.
- RXD[3:0]: Receive data. RXD is a bundle of 4 data signals synchronously driven by the PHY, which are valid only when the RX_DV signal is active (valid data). RXD[0] is the least significant bit, and RXD[3] is the most significant bit. Specific RXD[3:0] values are used for transmitting specific information from the PHY when RX_EN is inactive and RX_ER is active.
- RX_DV: Receive data valid. Indicates that the PHY has recovered and decoded half-bytes for MII reception. It must be synchronized with the first recovered half-byte of the frame (RX_CLK) and remain active until the last

recovered half-byte. It must be set to inactive before the first clock cycle after the last half-byte. To correctly receive the frame, RX_DV must cover the entire frame, starting no later than the SFD field.

- RX_ER: Receive error. Must remain active for one or more clock cycles (RX_CLK) to indicate an error detected in the frame at some point. The error condition must be limited by the RX_DV status.
- CRS: Carrier sense. PHY sets this signal to active when the medium is not idle during transmission or reception. It must be set to inactive when both transmission and reception media are idle. The PHY must ensure that the CS signal remains active during the duration of the collision condition. This signal does not require synchronization with the transmit and receive clock transitions. In full-duplex mode, the status of this signal is irrelevant to the MAC sublayer.
- COL: Collision detection. PHY sets this signal to active when a collision is detected on the medium and maintains it during the collision condition. This signal does not require synchronization with the transmit and receive clock transitions. In full-duplex mode, the status of this signal is irrelevant to the MAC sublayer.

Note：The MII follows the IEEE 802.3-2015 standard. For more information on MII, please refer to the relevant documentation of IEEE 802.3-2015.

35.5.4.3 Reduced Media Independent Interface (RMII)

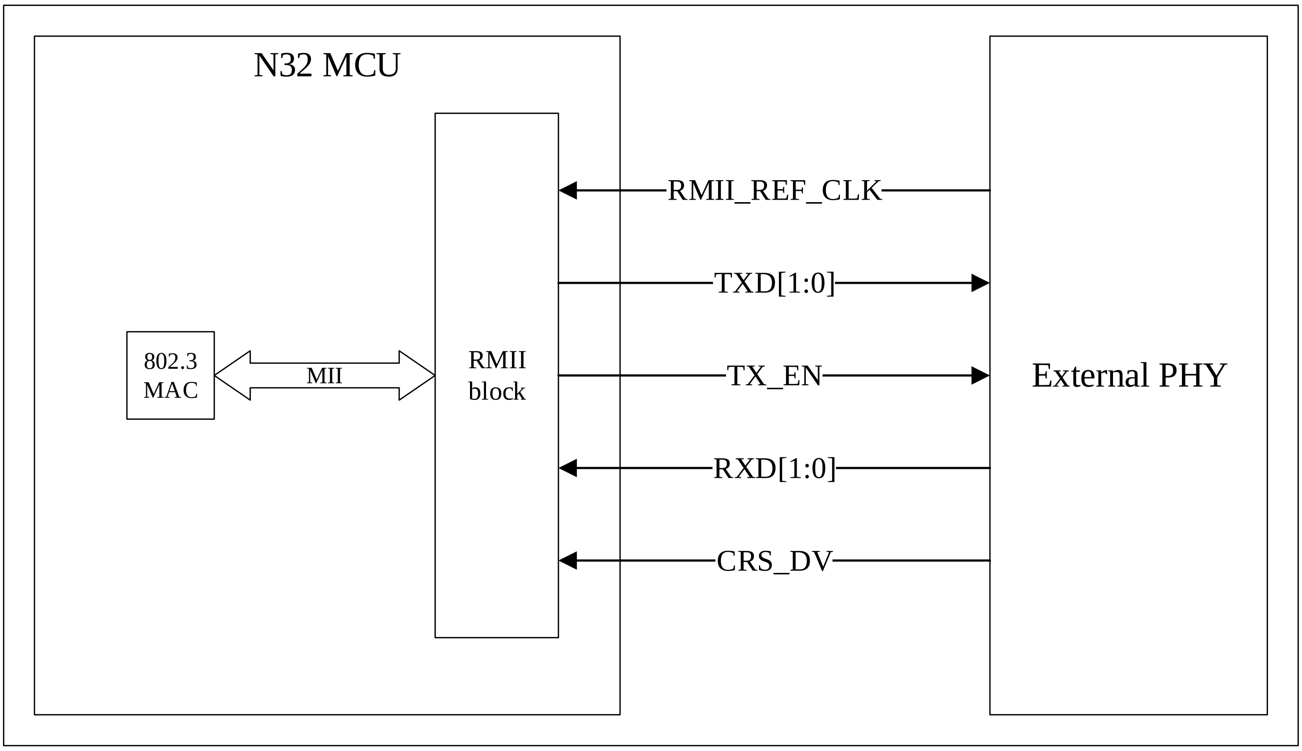
The RMII specification reduces the number of pins between the Ethernet PHY and the switch ASIC (only in 10/100Mbps mode). According to IEEE 802.3u, the MII includes 16 pins for data and control. In devices containing multiple MAC or PHY interfaces, such as switches, the cost increases significantly as the number of ports increases. The RMII specification addresses this issue by reducing the number of pins per port to 7, reducing the pin count by 62.5%.

The RMII module is instantiated between the MAC and PHY, facilitating the conversion of the MAC's MII to RMII. The RMII module has the following characteristics:

- Supports 10Mbps and 100Mbps operating rates. Does not support 1000Mbps operating rate.
- Provides independent 2-bit wide transmit and receive paths through two external clock reference sources.

The RMII module is located before the Ethernet peripheral core to convert MII signals to RMII signals. The signals of RMII are as shown in the following diagram:

Figure 35-10 RMII Interface

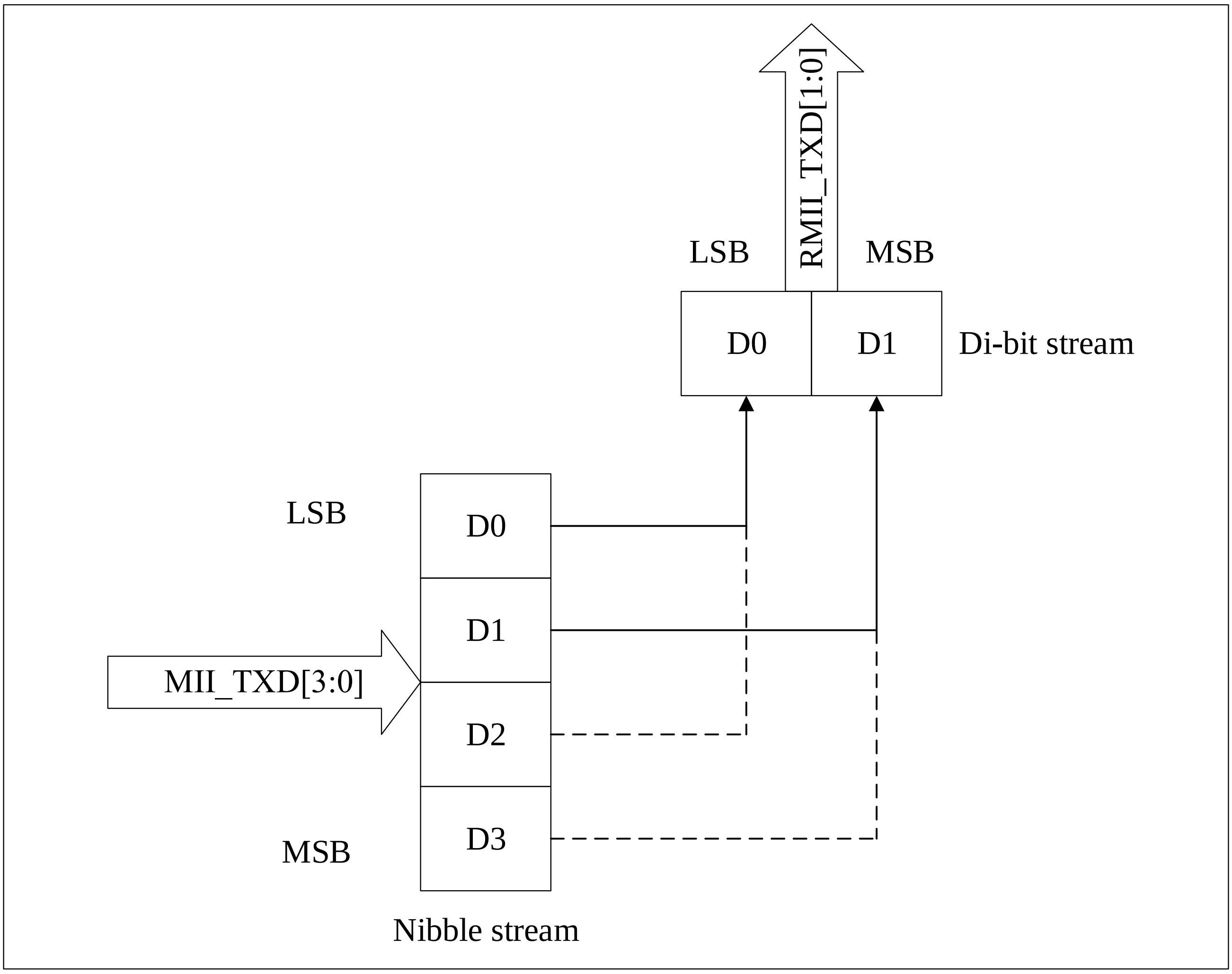


- RMII_REF_CLK: Continuous clock signal, 50MHz reference clock input.
- TXD[1:0]: Transmit data.
- TX_EN: Transmit data enable. When high, indicates valid data is being transmitted on TXD[1:0].
- RXD[1:0]: Receive data.
- CRS_DV: Carrier Sense (CRS) and Receive Data Valid (RX_DV) are multiplexed in alternate clock cycles. In 10Mbit/s mode, they alternate every 10 clock cycles.

Transmit bit sequence

Each half-byte of the MII interface must be sent on the RMII interface, transmitting two bits at a time, in the order shown in Figure 35-11. The low bits (D1 and D0) are sent first, followed by the high bits (D2 and D3).

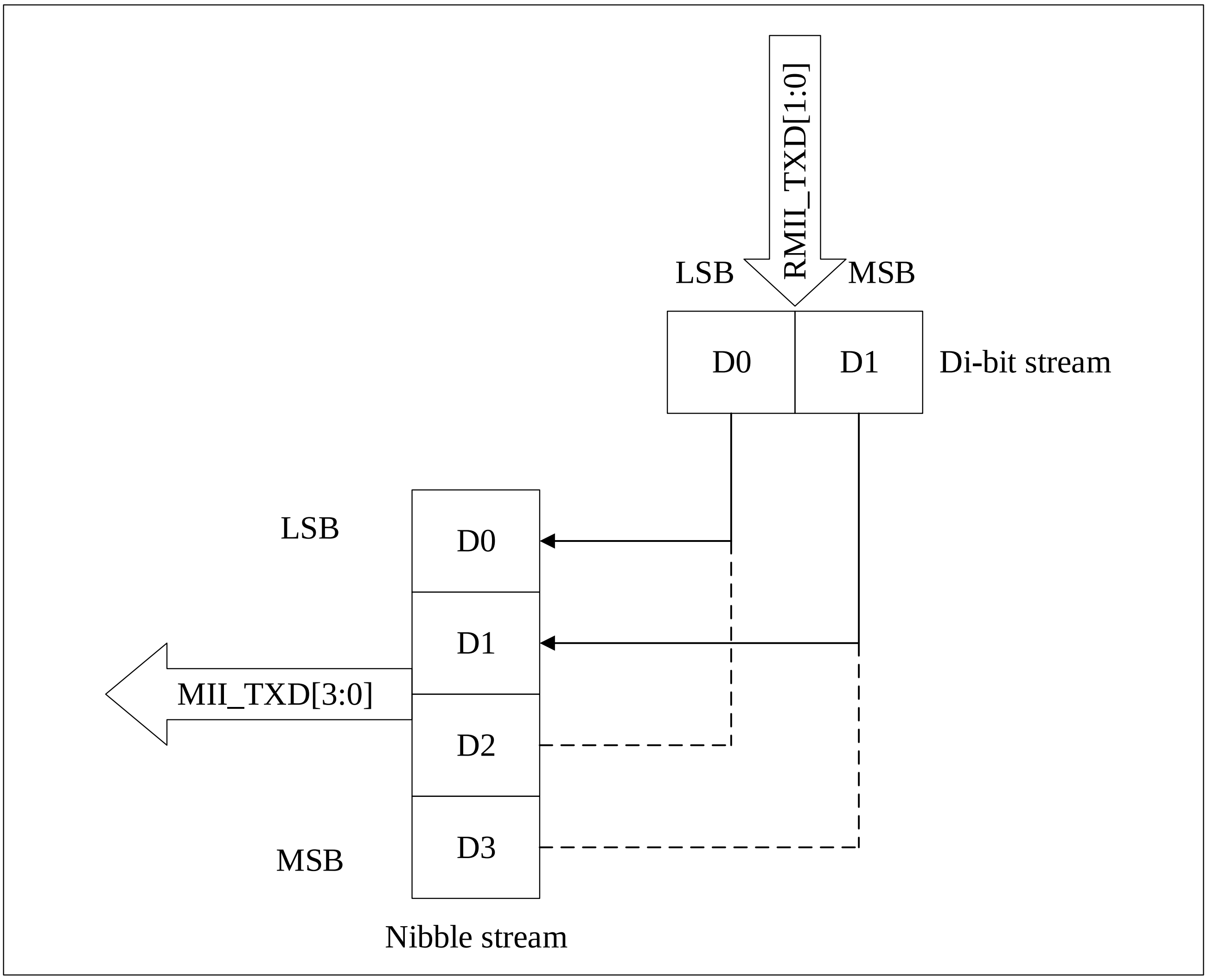
Figure 35-11 RMII Transmit bit sequence



Receive bit sequence

Each half-byte is transmitted in the order shown in Figure 35-12, with the two bits received from the RMII interface transferred to the MII interface. The low bits (D1 and D0) are received first, followed by the high bits (D2 and D3).

Figure 35-12 RMII Receive bit sequence



Note：RMII follows the RMII Consortium Specification version 1.2. For more information on RMII, please refer to the documentation related to the RMII Consortium Specification version 1.2.

35.5.5 Packet Filtering

The MAC supports the following types of Rx data packet filtering:

MAC Source Address or Destination Address Filtering: The Address Filtering Module (AFM) examines the source address and destination address fields of each incoming data packet.

VLAN Filtering: The MAC supports VLAN filtering based on VLAN tags and VLAN hashing.

These two filtering types can be cascaded. When all filters are active, filtering follows the order of address filtering first, then VLAN filtering. If any layer of the filter is not enabled, it is bypassed, and the subsequent filters are applied. Packets that do not pass through any filter are discarded. However, the discarded packets can be forwarded to the host based on register control. For example, when the RA bit in the MAC data packet filter register is set to 1, all discarded packets are forwarded to the host, but their packet status will indicate a specific filter failure. If RA is 0, the VTFE bit in the MAC data packet filter register controls whether packets that do not pass through VLAN filtering are discarded or forwarded to the host.

35.5.5.1 MAC Source Address or Destination Address Filtering

The MAC's address filtering module checks the source address (SA) and destination address (DA) fields of each incoming data packet.

Unicast Destination Address Filtering

The MAC supports unicast perfect filtering for 4 MAC addresses. If perfect filtering is selected (resetting the HUC bit in the MAC data packet filter register), the MAC compares all 48 bits of the received unicast address with the programmed MAC addresses to find a match. By default, MacAddr0 is always enabled. MacAddr1 to MacAddr3 are selected through individual enable bits. For MacAddr1 to MacAddr3, masking byte control bits in the register can be set to mask each byte when comparing with the respective received DA bytes, enabling the group address filtering for DA. In hash filtering mode (when the HUC bit is set to 1), the MAC performs incomplete filtering for unicast addresses using a 64-bit hash table. In hash filtering mode, the MAC uses the high 6 bits of the received destination address's CRC to index the contents of the hash table. A value of 00000 selects the 0th bit of the selected register, and a value of 11111 selects the 63rd bit of the hash table register. If the corresponding bit (represented by the 6-bit CRC) is set to 1, the unicast data packet is considered to have passed the hash filter; otherwise, it is considered to have not passed the hash filter.

Multicast destination address filtering

To program the MAC to pass all multicast packets, the PAM bit (bit4) in the MAC packet filter register needs to be set. If the PAM bit is reset, the MAC will filter multicast addresses based on the HMC bit in the MAC packet filter register. The multicast address is compared with the programmed MAC destination address registers (1~3). Group address filtering is also supported. In hash filtering mode, the MAC performs incomplete filtering using a 64-bit hash table. The MAC uses the high 6 bits of the received multicast address CRC to index the hash table content. When the value is 00000, it selects the 0th bit of the selected register, and when the value is 11111, it selects the 63rd bit of the hash table register. If the corresponding bit is set to 1, the multicast packet is considered to have passed the hash filter. Otherwise, the packet is considered to have not passed the hash filter.

Hash or Perfect Address Filtering

To configure the DA filter to pass a packet when its DA matches either the Hash filter or the Perfect filter, set the HPF bit and the corresponding HUC or HMC bits in MAC Packet Filter register. This is applicable to both unicast and multicast packets. If the HPF bit is reset, only one of the filters (Hash or Perfect) is applied to received packet.

Broadcast Address Filtering

The MAC does not filter any broadcast packets by default. To program the MAC to reject all broadcast packets, set the DBF bit in MAC Packet Filter register.

Unicast Source Address Filtering

The MAC can perform perfect filtering based on the source address field of received packets. By default, the MAC compares the SA field with the values programmed in the SA registers. You can configure the MAC Address registers (1~3) to use SA instead of DA for comparison by setting Bit 30 of corresponding register.

The MAC also supports group filtering with SA. You can filter a group of addresses by masking one or more bytes of the address. The MAC drops the packets that fail the SA filter if the SAF bit is set in MAC Packet Filter register. Otherwise, the result of the SA filter is given as a status bit in the Receive Status word. When the SAF bit is set, the

SA filter and DA filter result is AND'ed to decide whether the packet needs to be forwarded. This means that the packet is dropped if either filter fails. The packet is forwarded to the application only if the packet passes both filters in-order.

Inverse Filtering

For DA and SA filtering, you can invert the filter-match result at the final output by setting the DAIF and SAIF bits of MAC Packet Filter register. The DAIF bit is applicable for both Unicast and Multicast DA packets. The result of the unicast or multicast destination address filter is inverted in this mode. Similarly, when the SAIF bit is set, the result of unicast SA filter is reversed.

Note: When the RA bit of MAC Packet Filter register is set, all packets are forwarded to the system along with the correct result of the address filtering in the Rx Status.

Table 35-9 and Table 35-10 summarize the DA and SA filtering based on the type of packets received.

Table 35-9 DA Filtering

Packet Type	PM	HPF	HUC	DAIF	HMC	PAM	DBP	DA Filter Operation
Broadcast	1	x	x	x	x	x	x	Pass
	0	x	x	x	x	x	0	Pass
	0	x	x	x	x	x	1	Fail
Unicast	1	x	x	x	x	x	x	Pass all packets
	0	x	0	0	x	x	x	Pass on Perfect/Group filter match
	0	x	0	1	x	x	x	Fail on Perfect/Group filter match 0 0
	0	0	1	0	x	x	x	Pass on Hash filter match
	0	0	1	1	x	x	x	Fail on Hash filter match
	0	1	1	0	x	x	x	Pass on Hash or Perfect/Group filter match
	0	1	1	1	x	x	x	Fail on Hash or Perfect/Group filter match
Multicast	1	x	x	x	x	x	x	Pass all packets
	x	x	x	x	x	1	x	Pass all packets
	0	x	x	0	0	0	x	Pass on Perfect/Group filter match and drop Pause packets if PCF = 0x
	0	0	x	0	1	0	x	Pass on Hash filter match and drop Pause packets if PCF = 0x
	0	1	x	0	1	0	x	Pass on Hash or Perfect/Group filter match and drop Pause packets if PCF = 0x
	0	x	x	1	0	0	x	Fail on Perfect/Group filter match and drop Pause packets if PCF = 0x
	0	0	x	1	1	0	x	Fail on Hash filter match and drop Pause packets if PCF = 0x
	0	1	x	1	1	0	x	Fail on Hash or Perfect/Group filter match and drop Pause packets if PCF = 0x

Table 35-10 SA Filtering

Packet Type	PM	SAIF	SAF	SA Filter Operation
Unicast	1	x	x	Pass all packets
	0	0	0	Pass status on Perfect or Group filter match but do not drop packets that fail
	0	1	0	Fail status on Perfect or Group filter match but do not drop packet
	0	0	1	Pass on Perfect or Group filter match and drop packets that fail
	0	1	1	Fail on Perfect or Group filter match and drop packets that fail

Note: x represents any value.

35.5.5.2 VLAN Filtering

VLAN Tag Perfect Filtering

In VLAN tag perfect filtering, the MAC compares the VLAN tag of received packet and provides the VLAN packet status to the application. Based on the programmed mode, the MAC compares the lower 12 bits or all 16 bits of received VLAN tag to determine the perfect match.

If VLAN tag perfect filtering is enabled, the MAC forwards the VLAN-tagged packets along with VLAN tag match status and drops the VLAN packets that do not match. You can also enable the inverse matching for VLAN packets by setting the VTIM bit of MAC VLAN Tag register. In addition, you can enable processing of S-VLAN tagged packets along with the default C-VLAN tagged packets by setting the ESVL bit of MAC VLAN Tag register. The VLAN packet status bit (Bit 10 of RDES0) indicates the VLAN tag match status for the matched packets.

Note: The source or destination address (if enabled) has precedence over the VLAN tag filters. This means that a packet that fails the source or destination address filter is dropped irrespective of the VLAN tag filter results. By default, the VLAN tag-based perfect filter is available in all configurations.

VLAN Tag Hash Filtering

The 16-bit VLAN Hash Table is used for group address filtering based on the VLAN tag. The VLAN Tag Hash Filtering feature can be enabled using the VTHM (VLAN Tag Hash Table Match Enable) bit of the MAC VLAN Tag register.

The MAC provides VLAN tag hash filtering with a 16-bit Hash table.

The MAC performs the VLAN hash matching based on the VTHM of the MAC VLAN Tag register. If the VTHM bit is set, the most significant four bits of CRC-32 of VLAN tag are used to index the content of the MAC VLAN Hash Table register. A value of 1 in the MAC VLAN Hash Table register, corresponding to the index, indicates that the VLAN tag of the packet matched and the packet should be forwarded. A value of 0 indicates that VLAN-tagged packet should be dropped.

Note: The 16 or 12 bits of VLAN Tag are considered for CRC-32 computation based on ETV bit in MAC_VLAN_TAG register. When ETV bit is reset, most significant four bits of CRC-32 of VLAN Tag are inverted and used to index the content of MAC VLAN Hash Table register. When ETV bit is set, most significant four bits of CRC-32 of VLAN Tag are directly used to index the content of MAC VLAN Hash Table register.

The MAC also supports the inverse matching for VLAN packets. In the inverse matching mode, when the VLAN tag of a packet matches the perfect or hash filter, the packet should be dropped. If the VLAN perfect and VLAN hash match are enabled, a packet is considered as matched if either the VLAN hash or the VLAN perfect filter matches.

When inverse match is set, a packet is forwarded only when both perfect and hash filters indicate mismatch.

Table 35-11 shows the different possibilities for VLAN matching and the final VLAN match status. When the RA bit of MAC Packet Filter register is set, all packets are received and the VLAN match status is indicated in the VF bit of RDES2 Normal Descriptor (Write-Back Format). When the RA bit is not set and the VTFE bit is set in MAC Packet Filter register, the packet is dropped if the final VLAN match status is Fail. In Table 35-11, value X means that this column can have any value.

When VLAN VID is programmed to 0 in the VL field of MAC VLAN Tag register, all VLAN-tagged packets are considered as perfect matched but the status of the VLAN hash match depends on the VTHM and VTIM bits in MAC VLAN Tag register.

Table 35-11 VLAN Match Status

VID	VLAN Perfect Filter Match Result	VTHM	VLAN Hash Filter Match Result	VTIM	Final VLAN Match Status
VID = 0	Pass	0	x	x	Pass
	Pass	1	x	0	Pass
	Pass	1	Fail	1	Pass
	Pass	1	Pass	1	Fail
VID != 0	Pass	x	x	0	Pass
	Fail	0	x	0	Fail
	Fail	1	Fail	0	Fail
	Fail	1	Pass	0	Pass
	Fail	0	x	1	Pass
	Pass	x	x	1	Fail
	Fail	1	Pass	1	Fail
	Fail	1	Fail	1	Pass

Note: x represents any value.

35.5.6 IEEE 1588 Timestamp Support

The IEEE 1588 defines a Precision Time Protocol (PTP) which enables precise synchronization of time in measurement and control systems. This protocol enables heterogeneous systems that include clocks of varying inherent precision, resolution, and stability to synchronize. The protocol supports system-wide synchronization accuracy in the sub-microsecond range with minimal network and local clock computing resources.

The ETH supports the IEEE 1588-2002 (version 1) and IEEE 1588-2008 (version 2). The IEEE 1588-2002 supports PTP transported over UDP/IP and IEEE 1588-2008 supports PTP transported over Ethernet. The ETH provides programmable support for both standards.

The controller supports the following features:

- Provides an option to take snapshot of all packets or only PTP type packets
- Provides an option to take snapshot of only event messages
- Provides an option to take the snapshot based on the clock type: ordinary, boundary, end-to-end transparent, and peer-to-peer transparent

- Provides an option to select the node to be a master or slave for ordinary and boundary clock
- Identifies the PTP message type, version, and PTP payload in packets sent directly over Ethernet and sends the status
- Provides an option to measure sub-second time in digital or binary format

35.5.6.1 Clock Types

Ordinary Clock

The ordinary clock has a single PTP state and a single physical port. In a domain, an ordinary clock supports a single copy of the protocol.

The ordinary clock in a domain supports a single copy of the protocol. The ordinary clock has a single PTP state and a single physical port. In typical industrial automation applications, an ordinary clock is associated with an application device such as a sensor or an actuator. In telecom applications, the ordinary clock can be associated with a timing demarcation device.

The ordinary clock can be a grandmaster or a slave clock. It supports the following features:

- Sends and receives PTP messages. The timestamp snapshot can be controlled as described in MAC Timestamp Control register.
- Maintains the data sets such as timestamp values.

Table 35-12 shows the messages for which you can take the timestamp snapshot on the receive side for master and slave nodes.

Table 35-12 Ordinary Clock: PTP Messages for Snapshot

Master	slave
Delay_Req	SYNC

For an ordinary clock, you can take the snapshot of either of the following PTP message types: version 1 or version 2. You cannot take the snapshots for both PTP message types. You can take the snapshot by setting the TSVER2ENA bit and selecting the snapshot mode in MAC Timestamp Control register.

Boundary Clock

The boundary clock typically has several physical ports communicating with the network. The messages related to synchronization, master-slave hierarchy, and signaling terminate in the protocol engine of the boundary clock and such messages are not forwarded. The PTP message type status given by the MAC helps you to identify the type of message and take appropriate action.

The boundary clock is similar to the ordinary clock except for the following features:

- The clock data sets are common to all ports of the boundary clock.
- The local clock is common to all ports of the boundary clock.

End-to-End Transparent Clock

The end-to-end transparent clock supports the end-to-end delay measurement mechanism between the slave clocks and the master clock. The end-to-end transparent clock forwards all messages like normal bridge, router, or repeater. The residence time of a PTP packet is the time taken by the PTP packet from the Ingress port to the Egress port.

The residence time of a SYNC packet inside the end-to-end transparent clock is updated in the correction field of the associated Follow_Up PTP packet before it is transmitted. Similarly, the residence time of a Delay_Req packet, inside the end-to-end transparent clock, is updated in the correction field of the associated Delay_Resp PTP packet before it is transmitted. Therefore, the snapshot must be taken at both Ingress and Egress ports only for the messages mentioned in Table 35-13. You can take the snapshot by setting the SNAPTYPSEL bits to 10 in the MAC Timestamp Control register.

Table 35-13 End to End Transparent Clock: PTP Messages for Snapshot

PTP Messages
SYNC
Delay_Req

Peer-to-Peer Transparent Clock

In the peer-to-peer transparent clock, the computation of the link delay is based on an exchange of Pdelay_Req, Pdelay_Resp, and Pdelay_Resp_Follow_Up messages with the link peer. The peer-to-peer transparent clock differs from the end-to-end transparent clock in the way it corrects and handles the PTP timing messages. In all other aspects, it is identical to the end-to-end transparent clock. In the peer-to-peer transparent clock, the computation of the link delay is based on an exchange of Pdelay_Req, Pdelay_Resp, and Pdelay_Resp_Follow_Up messages with the link peer. The residence time of the Pdelay_Req and the associated Pdelay_Resp packets is added and inserted into the correction field of the associated Pdely_Resp_Followup packet. Therefore, support for taking snapshot for the event messages related to Pdelay is added as shown in Table 35-14.

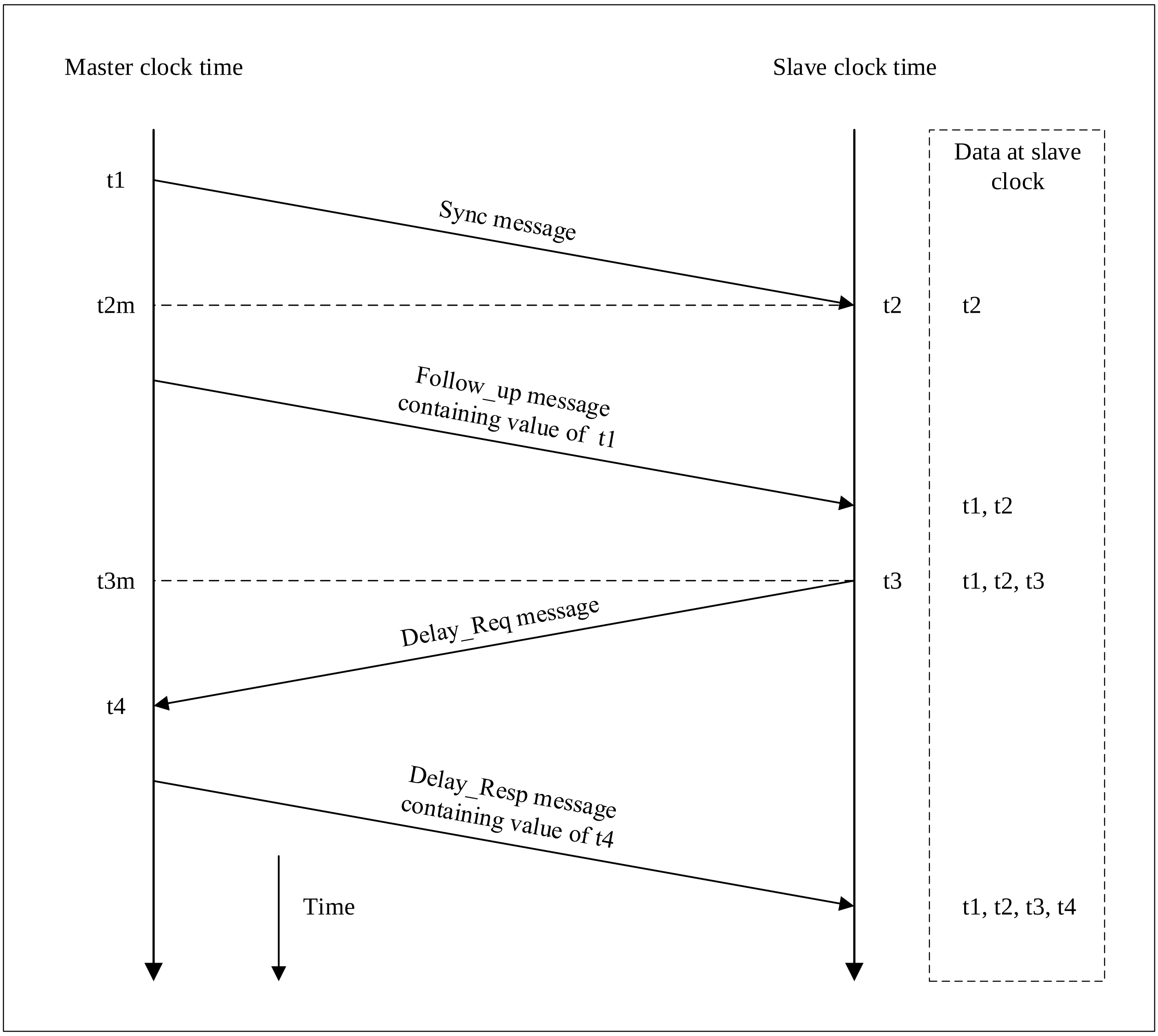
Table 35-14 Peer-to-Peer Transparent Clock: PTP Messages for Snapshot

PTP Messages
SYNC
Pdelay_Req
Pdelay_Resp

35.5.6.2 Delay Request-Response Mechanism

The system or network is classified into the master and slave nodes for distributing the timing and clock information. Figure 35-13 shows the process that PTP uses for synchronizing a slave node to a master node by exchanging PTP messages.

Figure 35-13 Networked Time Synchronization



As shown in Figure 35-13, PTP uses the following process:

1 The master broadcasts the PTP Sync messages to all its nodes. The Sync message contains the reference time information of the master. This message leaves the system of the master at t1. This time must be captured for Ethernet ports at MII.

2 The slave receives the Sync message and also captures the exact time, t2, using its timing reference.

3 The master sends a Follow_up message to the slave, which contains t1 information for later use.

4 The slave sends a Delay_Req message to the master and notes the exact time, t3, at which this packet leaves the MII interface.

5 The master receives the message, capturing the exact time t4, at which the message enters its system.

6 The master sends the t4 information to the slave in the Delay_Resp message.

7 The slave uses the four values of t1, t2, t3, and t4 to synchronize its local timing reference to the timing reference

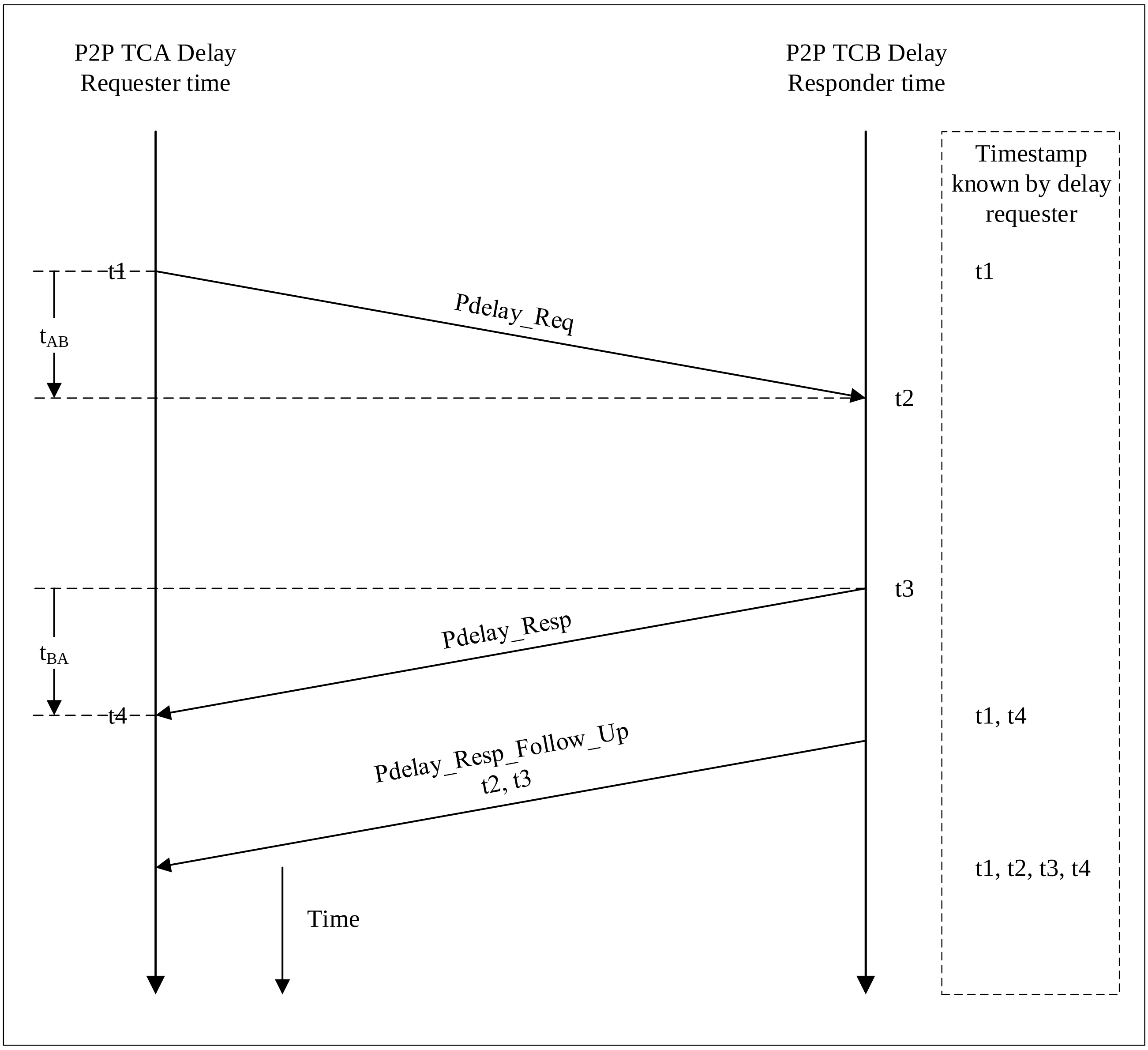
of the master.

Most of the PTP implementation is done in the software above the Ethernet/UDP layer. However, the hardware support is required to capture the exact time when specific PTP packets enter or leave the Ethernet port at the MII interface. This timing information must be captured and returned to the software for proper implementation of PTP with high accuracy.

35.5.6.3 Peer-to-Peer PTP Transparent Clock (P2P TC) Message Support

The IEEE 1588-2008 supports peer-to-peer PTP (Pdelay) message in addition to the SYNC, Delay Request, Follow-up, and Delay Response messages. Figure 35-14 shows the method to calculate the propagation delay in clocks supporting peer-to-peer path correction.

Figure 35-14 Propagation Delay Calculation in Clocks Supporting Peer-to-Peer Path Correction



As shown in Figure 35-14, the propagation delay is calculated in the following way:

1 Port 1 issues a Pdelay_Req message and generates a timestamp (t1) for the Pdelay_Req message.

2 Port 2 receives the Pdelay_Req message and generates a timestamp (t2) for this message.

3 Port 2 returns a Pdelay_Resp message and generates a timestamp (t3) for this message.

To minimize errors because of any frequency offset between the two ports, Port 2 returns the Pdelay_Resp message as quickly as possible after the receipt of the Pdelay_Req message. Port 2 returns any one of the following:

- Difference between the timestamps t2 and t3 in the Pdelay_Resp message
- Difference between the timestamps t2 and t3 in the Pdelay_Resp_Follow_Up message
- Timestamps t2 and t3 in the Pdelay_Resp and Pdelay_Resp_Follow_Up messages, respectively

4 Port 1 generates a timestamp (t4) on receiving the Pdelay_Resp message.

5 Port 1 uses all four timestamps to compute the mean link delay.

35.5.6.4 Timestamp Correction

According to the IEEE 1588 specification, a timestamp must be captured when the PTP message timestamp point (leading edge of the first bit of the octet immediately following the Start Frame Delimiter octet) crosses the boundary between the node and the network. As the MAC takes the timestamp at an internal point far from the actual boundary of the node and network, this captured timestamp is corrected/updated for the ingress/egress path latency (including the delay in the PHY layers). Further correction is done for the inaccuracies/errors introduced due to the clock (MII Tx, Rx clock) being different at the capture point as compared to the PTP clock (clk_ptp_ref_i) that is used to generate the time. The resultant CDC (Clock Domain Crossing) circuits add error depending on the clock period of the MII and PTP clocks.

Ingress Correction

Compared to the time at which that packet's SFD bit is received at the port's boundary. Therefore, the captured timestamp must be reduced by the ingress latency and the errors in CDC sampling. This correction value must be determined/calculated by the software and written into the MAC Timestamp Ingress Corr_* registers.

The correction value consists of the following 3 components:

1 External latency in the PHY layer between boundary point and the input of the core.

If the PHY is compliant to the IEEE 802.3 Clause 45 MMD registers, it has a register indicating the maximum and minimum ingress latency. The software can read these registers and determine the average ingress latency in the PHY. Alternatively (if the PHY does not support these registers), the ingress latency must be determined from its datasheet or timing characteristics.

2 Internal latency from the input of the core to the internal capture point

The internal ingress latency can be read from the MAC Ingress Timestamp Latency register. This is a read-only register and gives the latency in scaled Nanoseconds format defined in IEEE 1588 Clause 5.3.2. The latency differs based on the active PHY interface (RGMII, RMII, so on) and the operating speed. Therefore, the software must read this register after any speed change in the MAC to determine the current internal latency.

3 CDC Synchronization

The CDC synchronization error is almost equal to 2 times the clock-period of the PTP clock (clk_ptp_ref_i).

The values determined from these 3 components should be added by the software and must be written into the TSIC & TSICSNS fields of the MAC Timestamp Ingress Corr_* registers.

Note: The value written to the register must be negative (two's complement), because it has to be subtracted from the captured timestamp. The MAC Receiver adds the value in this register to the captured timestamp and then gives the resultant value as the timestamp of the received packet.

Timestamp is in decimal format with a granularity of 1ns. So the Bit31 of TSIC must be set to 1 (for negative value) and bits [30:0] must be written with "10^9 - total ingress correction value [nanosecond part]" represented in binary. For example, if the required correction value is -5 ns, then the value is 0xBB9A_C9FB.

When TSCTRLSSR bit in MAC Timestamp Control register is reset, the nanoseconds field of the captured timestamp is in binary format with a granularity of ~0.466ns. Therefore, bits [30:0] must be written with "2^{31} - total ingress correction value" represented in binary with bit [31] = 1.

Egress Correction

In the Transmit side the timestamp captured at the internal snapshot point is earlier (advanced in time) as compared to the time at which that packet's SFD bit is output at the port's boundary. Therefore, the captured timestamp must be compensated by the egress latency and the errors in CDC sampling. This correction value must be determined or calculated by the software and written into the MAC Timestamp Egress Corr_* registers.

The correction value consists of the following 3 components:

1 External latency in the PHY layer between the output of the core and the boundary of the port and the network

If the PHY is compliant to the IEEE 802.3 Clause 45 MMD registers, it has a register indicating the maximum and minimum egress latency. The software can read these registers and determine the average egress latency in the PHY. Alternatively (if the PHY does not support these registers), the egress latency must be determined from its datasheet or timing characteristics.

2 Internal latency from the internal capture point and the output of the core

This internal egress latency can be read from the MAC Egress Timestamp Latency register. This is a read-only register and gives the latency in scaled Nanoseconds format defined in IEEE 1588 Clause 5.3.2. The latency differs based on the active PHY interface (RMII) and the operating speed. Therefore, the software must read this register after any speed change in the MAC, to determine the current internal latency.

3 CDC synchronization error

The CDC synchronization error value differs depending on the One-step timestamp mode. When One-step timestamp is enabled, the value = (1 * period of clk_ptp_ref_i + 4 * period of tx_clk). Otherwise (Two-step timestamp mode), the value = - (2 * period of clk_ptp_ref_i).

35.5.6.5 PTP Processing and Control

Table 35-15 shows the common message header for the PTP messages. This format is taken from the IEEE 1588-2008.

Table 35-15 Message Format Defined in IEEE 1588-2008

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Octet	Offset
transportSpecific				messageType				1	0

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Octet	Offset
Reserved				versionPTP				1	1
messageLength								2	2
domainNumber								1	4
Reserved								1	5
flagField								2	6
correctionField								8	8
Reserved								4	16
sourcePortIdentity								10	20
sequenceId								2	30
controlField[(1)]								1	32
logMessageInterval								1	33

1. controlField is used in version 1. In version 2, message Type field is used for detecting different message types.

There are some fields in the Ethernet payload that you can use to detect the PTP packet type and control the snapshot to be taken. These fields are different for the following PTP packets:

PTP Packets Over IPv4

Table 35-16 provides information about the fields that are matched to control snapshot for the PTP packets sent over UDP over IPv4 for IEEE 1588 version 1 and 2. The octet positions for the tagged packets are offset by 4. This is based on the IEEE 1588-2008, Annex D and the message format defined in Table 35-15.

Table 35-16 IPv4-UDP PTP Packet Fields Required for Control and Status

Field Matched	Octet Position	Matched Value	Description
MAC Packet Type	12, 13	0x0800	IPv4 datagram
IP version and Header Length	14	0x45	IP version is IPv4
Layer 4 Protocol	23	0x11	UDP
IP Multicast Address (IEEE 1588 version 1)	30, 31, 32, 33	0xE0, 0x00, 0x01, 0x81 (/0x82/0x83/0x84)	Multicast IPv4 addresses allowed: ■ 224.0.1.129 ■ 224.0.1.130 ■ 224.0.1.131 ■ 224.0.1.132
IP Multicast Address (IEEE 1588 version 2)	30, 31, 32, 33	0xE0, 0x00, 0x01, 0x81 0xE0, 0x00, 0x00, 0x6B	• PTP Primarymulticast address: 224.0.1.129 • PTP Pdelaymulticast address: 224.0.0.107
UDP Destination Port	36, 37	0x013F, 0x0140	• 0x013F: PTP event messages [(1)] • 0x0140: PTP general messages
PTP Control Field (IEEE 1588 version 1)	74	0x00, 0x01, 0x02, 0x03, 0x04	■ 0x00: SYNC ■ 0x01: Delay_Req

Field Matched	Octet Position	Matched Value	Description
			■ 0x02: Follow_Up ■ 0x03: Delay_Resp ■ 0x04: Management
PTP Message Type Field (IEEE 1588 version 2)	42 (nibble)	0x0, 0x1, 0x2, 0x3, 0x8, 0x9, 0xB, 0xC, 0xD	■ 0x0: SYNC ■ 0x1: Delay_Req ■ 0x2: Pdelay_Req ■ 0x3: Pdelay_Resp ■ 0x8: Follow_Up ■ 0x9: Delay_Resp ■ 0xA: Pdelay_Resp_Follow_Up ■ 0xB: Announce ■ 0xC: Signaling ■ 0xD: Management
PTP Version	43 (nibble)	0x1/0x2	■ 0x1: PTP version 1 ■ 0x2: PTP version 2

1. PTP event messages are SYNC, Delay_Req (IEEE 1588 version 1 and 2) or Pdelay_Req, Pdelay_Resp (IEEE 1588 version 2 only).

PTP Frames Over IPv6

Table 35-17 provides information about the fields that are matched to control the snapshots for the PTP packets sent over UDP over IPv6 for IEEE 1588 version 1 and 2. The octet positions for the tagged packets are offset by 4. This is based on the IEEE 1588-2008, Annex D and the message format defined in Table 35-15.

Table 35-17 IPv6-UDP PTP Packet Fields Required for Control and Status

Field Matched	Octet Position	Matched Value	Description
MAC Packet Type	12, 13	0x86DD	IP datagram
IP Version	14	0x6	IP version is IPv6
Layer 4 Protocol	20[(1)]	0x11	UDP
PTP Multicast Address	38~53	FF0x:0:0:0:0:0:0:181 (Hex) FF02:0:0:0:0:0:0:6B (Hex)	• PTP Primary multicast address：FF0x:0:0:0:0:0:0:0:181 (Hex) • PTP Pdelay multicast address: FF02:0:0:0:0:0:0:0:6B (Hex)
UDP Destination Port	56, 57a	0x013F, 0x0140	• 0x013F: PTP event message[(1)] • 0x0140: PTP general messages
PTP Control Field (IEEE 1588 version 1)	94a	0x00, 0x01, 0x02, 0x03, 0x04	■ 0x00: SYNC ■ 0x01: Delay_Req

Field Matched	Octet Position	Matched Value	Description
			■ 0x02: Follow_Up ■ 0x03: Delay_Resp ■ 0x04: Management (version1)
PTP Message Type Field (IEEE 1588 version 2)	62a (nibble)	0x0, 0x1, 0x2, 0x3, 0x8, 0x9, 0xB, 0xC, 0xD	■ 0x0: SYNC ■ 0x1: Delay_Req ■ 0x2: Pdelay_Req ■ 0x3: Pdelay_Resp ■ 0x8: Follow_Up ■ 0x9: Delay_Resp ■ 0xA: Pdelay_Resp_Follow_Up ■ 0xB: Announce ■ 0xC: Signaling ■ 0xD: Management
PTP Version	63 (nibble)	0x1/0x2	■ 0x1: PTP version 1 ■ 0x2: PTP version 2

1. The Extension Header is not defined for PTP packets

PTP Packets Over Ethernet

Table 35-18 provides information about the fields that are matched to control the snapshots for the PTP packets sent over Ethernet for IEEE 1588 version 1 and 2. The octet positions for the tagged packets are offset by 4. This is based on the IEEE 1588-2008, Annex D and the message format.

Table 35-18 Ethernet PTP Packet Fields Required for Control and Status

Field Matched	Octet Position	Matched Value	Description
MAC Destination Multicast Address [1]	0~5	01-1B-19-00-00-00 01-80-C2-00-00-0E	All PTP messages can use any of the following multicast addresses [2]: ■ 01-1B-19-00-00-00 ■ 01-80-C2-00-00-0E[3]
MAC Packet Type	12, 13	0x88F7	PTP Ethernet packet
PTP Control Field (IEEE 1588 version 1)	46	0x00, 0x01, 0x02, 0x03, 0x04	■ 0x00: SYNC ■ 0x01: Delay_Req ■ 0x02: Follow_Up ■ 0x03: Delay_Resp ■ 0x04: Management
PTP Message Type Field (IEEE 1588 version 2)	14 (nibble)	0x0, 0x1, 0x2, 0x3, 0x8, 0x9, 0xB, 0xC, 0xD	■ 0x0: SYNC ■ 0x1: Delay_Req ■ 0x2: Pdelay_Req ■ 0x3: Pdelay_Resp ■ 0x8: Follow_Up

Field Matched	Octet Position	Matched Value	Description
			■ 0x9: Delay_Resp ■ 0xA: Pdelay_Resp_Follow_Up ■ 0xB: Announce ■ 0xC: Signaling ■ 0xD: Management
PTP Version	15 (nibble)	0x1/0x2	■ 0x1: PTP version 1 ■ 0x2: PTP version 2

1. The unicast address match of destination addresses (DA), programmed in MAC address 0 to 31, is used if the TSENMACADDR bit of MAC Timestamp Control register is set.

2. IEEE 1588-2008, Annex F.

3. The MAC does not consider the PTP version 1 messages with Peer delay multicast address (01-80-C2-00-00-0E) as valid PTP messages.

35.5.6.6 Transmit Path Functions

The MAC captures a timestamp when the Start Packet Delimiter (SFD) of a packet is sent on the MII interface. The packets, for which the timestamps has to be captured can be controlled on per-packet basis.

Each Transmit packet can be marked to indicate whether a timestamp should be captured for it.

MAC does not process incoming packets to identify PTP packets. It is necessary to specify the packets to capture timestamps. The control bits in the sending Description field can be used to specify the packets. MAC will return the timestamp to the corresponding software in the sending Description field, automatically attaching the timestamp to the specific PTP packet.

The 64-bit timestamp information is written into the TDES0 and TDES1 fields. The TDES0 field stores the 32 least significant bits of the timestamp.

35.5.6.7 Receive Path Functions

The MAC can be programmed to capture the timestamp of all packets received on the MII interface or to process packets to identify the valid PTP messages. Use the following options of the MAC Timestamp Control register to control the snapshot of the time to be sent to the application:

- Enable snapshot for all packets
- Enable snapshot for IEEE 1588 version 1 or version 2 timestamp
- Enable snapshot for PTP packets transmitted directly over Ethernet or UDP-IP-Ethernet
- Enable timestamp snapshot for the received packet for IPv4 or IPv6
- Enable timestamp snapshot only for EVENT messages (SYNC, DELAY_REQ, PDELAY_REQ, or PDELAY-_RESP)
- Enable the node to be a master or slave and select the snapshot type

Note: The ETH also supports the PTP messages over VLAN packets.

Table 35-19 Timestamp Snapshot Dependency on Register Bits

SNAPTYPSEL	TSMSTRENA	TSEVNTENA	PTP message
00	x	0	SYNC, Follow_Up, Delay_Req, Delay_Resp
00	0	1	SYNC
00	1	1	Delay_Req
01	x	0	SYNC, Follow_Up, Delay_Req, Delay_Resp, Pdelay_Req, Pdelay_Resp, Pdelay_Resp_Follow_Up
01	0	1	SYNC, Pdelay_Req, Pdelay_Resp
01	1	1	Delay_Req, Pdelay_Req, Pdelay_Resp
10	x	x	SYNC, Delay_Req
11	x	x	Pdelay_Req, Pdelay_Resp

DMA will return the timestamp to the corresponding software within the receiving Description field. The extended status containing timestamp information status and IPC status is written into the normal Description field RDES1, and the timestamp snapshot is written into the RDES0 and RDES1 fields of the context Description field. The RDES0 field stores the 32 least significant bits of the timestamp.

35.5.6.8 IEEE 1588 System Time Source

To obtain a timestamp snapshot, the MAC requires a reference time in the 64-bit format defined in IEEE 1588-2002 (or the 80-bit format defined in IEEE 1588-2008). The Ethernet peripheral only receives a reference clock input and uses it to internally generate the reference time (also known as system time) and capture timestamps. The timestamp includes the following fields:

- UInteger48 secondsField

  The secondsfield is the integer portion of the timestamp in units of seconds. It is 48-bits wide.

  For example, 2.000000001 seconds are represented as seconds Field = 0x0000_0000_0002.

- UInteger32 nanoSecondsField

  The nanoseconds field is the fractional portion of the timestamp in units of nanoseconds. For example, 2.000000001 seconds are represented as nanoSeconds = 0x0000_0001.

  The nanoseconds field supports the following two modes:

  - Digital rollover mode: In this mode, the maximum value in the nanoseconds field is 0x3B9A_C9FF, that is, (10e9-1) nanoseconds.
  - Binary rollover mode: In this mode, the nanoseconds field rolls over and increments the seconds field after value 0x7FFF_FFFF. Accuracy is ~0.466 ns per bit.

35.5.6.9 System Time Register Module

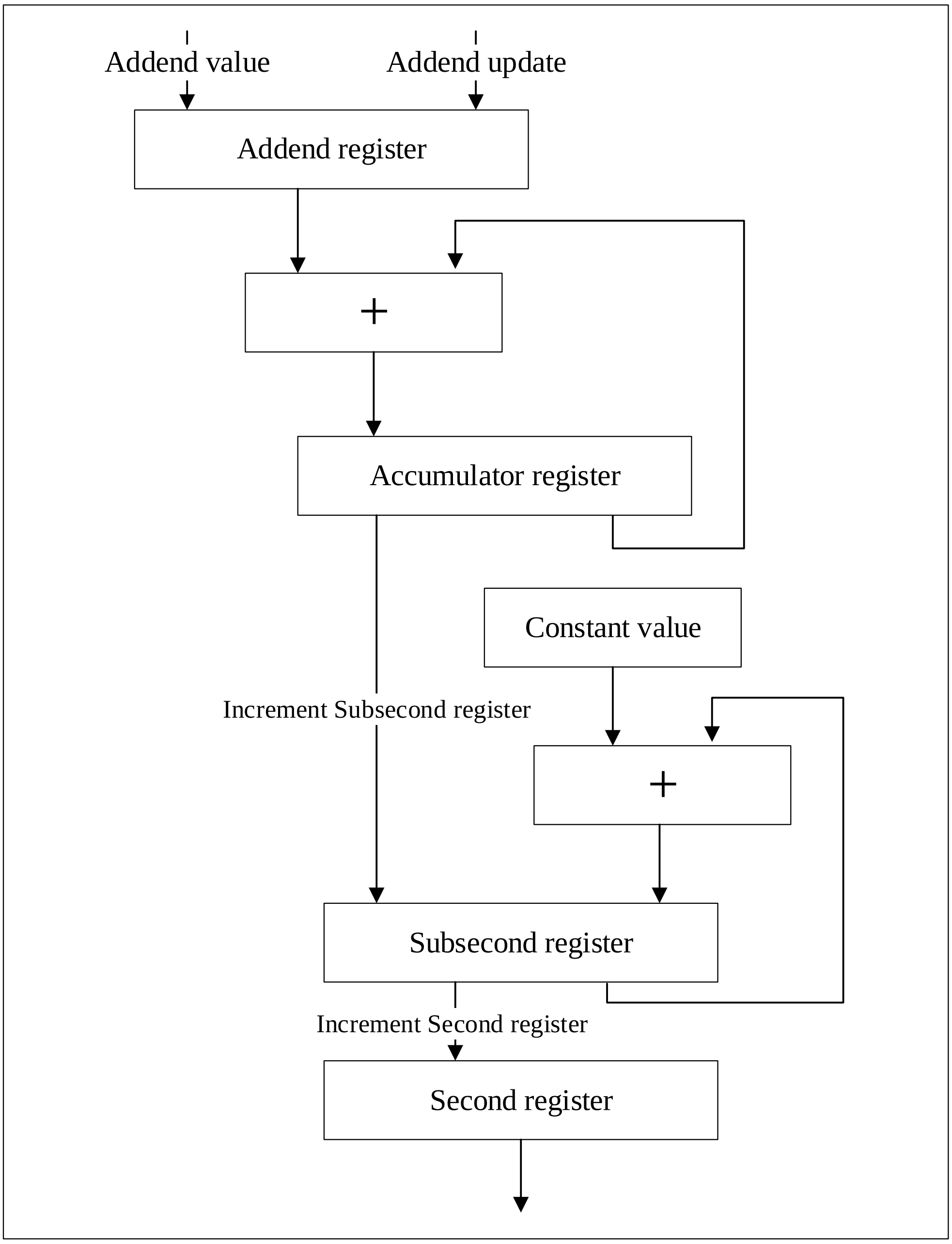
The 80-bit time is maintained in this module and updated using the input reference clock (clk_ptp_ref_i). This time is the source for taking snapshots (timestamps) of Ethernet packets being transmitted or received at the GMII interface.

The system time counter can be initialized or corrected using the coarse correction method. In this method, the initial value or the offset value is written to the Timestamp Update register. For initialization, the system time counter is written with the value in the Timestamp Update register. For system time correction, the offset value is added to or subtracted from the system time.

In the fine correction method, the frequency offset and/or frequency drift of a slave clock (clk_ptp_ref_i) with respect to the master clock (as defined in IEEE 1588-2002) is corrected over a period of time instead of in one clock, as in coarse correction. This helps maintain linear time and does not introduce drastic changes (or a large jitter) in the reference time between PTP Sync message intervals. In this method, an accumulator sums up the contents of the Addend register, as shown in Figure 35-15. The arithmetic carry that the accumulator generates is used as a pulse to increment the system time counter. The accumulator and the addend are 32-bit registers. The accumulator acts as a high-precision frequency multiplier or divider.

Note: You must connect a PTP clock with a frequency higher than the frequency required for the specified accuracy.

Figure 35-15 System Time Update Using Fine Method



The System Time Update logic requires a 50MHz clock frequency to achieve 20ns accuracy. The frequency division is the ratio of the reference clock frequency to the required clock frequency. For example, if the reference clock (clk_ptp_ref_i) is 66 MHz, this ratio is calculated as 66MHz/50MHz = 1.32. Therefore, the default addend value to be set in the register is 2^{32}/1.32, 0xC1F07C1F.

If the reference clock drifts lower, for example, to 65MHz, the ratio is 65/50, or 1.3 and the value to set in the addend register is 2^{32}/1.30, or 0xC4EC4EC4. If the clock drifts higher, for example, to 67MHz, the addend register must be set to 0xBF0B7672. When the clock drift is nil, the default addend value of 0xC1F07C1F (2^{32}/1.32) must be programmed.

In Figure 35-15， the constant value used to accumulate the sub-second register is decimal 43, which achieves an accuracy of 20 ns in the system time (in other words, it is incremented in 20 ns steps). When External Time Update is enabled, the optional System Time module is not available.

Initially, the slave clock is set with FreqCompensationValue0 in the Addend register. This value is as follows:

$$\text{FreqCompensationValue0} = 2^{32}/\text{FreqDivisionRatio}$$

If MasterToSlaveDelay is initially assumed to be the same for consecutive Sync messages, the algorithm given in this section must be applied. After a few Sync cycles, frequency lock occurs. The slave clock can then determine a precise MasterToSlaveDelay value and re-synchronize with the master using the new value.

The algorithm is as follows:

- At time MasterSyncTime_n the master sends the slave clock a Sync message. The slave receives this message when its local clock is SlaveClockTime_n and computes MasterClockTime_n as $\text{MasterClockTime}_n = \text{MasterSyncTime}_n + \text{MasterToSlaveDelay}_n$
- The master clock count for current Sync cycle, $\text{MasterClockCount}_n$ is $\text{MasterClockCount}_n = \text{MasterClockTime}_n – \text{MasterClockTime}_n – 1$ (assuming that MasterToSlaveDelay is the same for Sync cycles n and n – 1)
- The slave clock count for current Sync cycle, SlaveClockCount_n is $\text{SlaveClockCount}_n = \text{SlaveClockTime}_n – \text{SlaveClockTime}_n – 1$
- The difference between master and slave clock counts for current Sync cycle, ClockDiffCount_n is $\text{ClockDiffCount}_n = \text{MasterClockTime}_n – \text{SlaveClockTime}_n$
- The frequency-scaling factor for slave clock, FreqScaleFactor_n is $\text{FreqScaleFactor}_n = (\text{MasterClockCount}_n + \text{ClockDiffCount}_n) / \text{SlaveClockCount}_n$
- The frequency compensation value for Addend register, $\text{FreqCompensationValue}_n$ is $\text{FreqCompensationValue}_n = \text{FreqScaleFactor}_n * \text{FreqCompensationValue}_n – 1$

In theory, this algorithm achieves lock in one Sync cycle. However, it may take several cycles, because of changing network propagation delays and operating conditions. This algorithm is self-correcting. If the slave clock is initially set to an incorrect value from the master, the algorithm corrects it at the cost of more Sync cycles.

35.5.6.10 Flexible Pulse-Per-Second Output (PPS)

The Ethernet peripheral is in "Fixed Second Pulse Output" mode (referred to as fixed mode), indicating a 1-second interval. All PPS outputs are controlled by the programmed value in the PPSCTRL field. In fixed mode, independent control of individual PPS outputs is not supported. The MAC PPS target time seconds/nanoseconds registers are only used for generating target time arrival interrupts; they are not used for generating PPS outputs. The PPS output frequency can be changed by setting the PPSCTRL field in the MAC PPS control register.

35.5.6.11 One-Step Timestamp

The ETH supports the one-step timestamp feature. When the One step timestamp feature is enabled, the MAC identifies the offset in the packet and inserts the timestamp received from the application at that offset.

MAC Transmit PTP Mode

Depending upon the type of message and its mode, the MAC updates the following fields of Transmit PTP packets:

- correctionField in the PTP header of messages
- originTimestamp in SYNC, Delay_Req, and Pdelay_Req messages

Table 35-20 shows how the PTP mode is selected based on the settings of SNAPTYPSEL, TSMSTRENA, and TSEVNTENA bits of the MAC Timestamp Control register and the fields that are updated for the incoming PTP packets based on the message type in that mode, during the one-step timestamp operation.

Table 35-20 MAC Transmit PTP Mode and One-Step Timestamp Operation (1)

SNAPTYPSEL	TSMSTRENA	TSEVNTENA	Mode	TTSE(2)	OSTC(3)	TTS(4)	Messages Processed on Tx
x	x	x	N/A	1	x	x	Timestamp is captured and returned to application
x	x	x	N/A	x	0	x	OST operation is not performed (PTP packet is not modified)
2'b00	x	0	End-to-end transparent	0	1	Ingress TS	Sync (correction field for residence time and Ingress Asym cor)
							Delay_Req (correction field for residence time and Egress Asym Cor)
2'b00	0	1	Ordinary or Boundary Slave	1	1	x	Delay_Req (originTimestamp field) Delay_Req (correction field for Egress Asym cor)
2'b00	1	1	Ordinary or Boundary Master	0	1	x	Sync (originTimestamp field) Sync (correction field for sub-nanosecond

SNAPTYPSEL	TSMSTRENA	TSEVNTENA	Mode	TTSE(2)	OSTC(3)	TTS(4)	Messages Processed on Tx
							cor)
2’b01	x	0	End-to-End Transparent with support for peer delay mechanism	0	1	Ingress TS	Sync (correction field for residence time and Ingress Asym cor)
						Ingress TS	Pdelay_Req (correction field for residence time and Egress Asym Cor)
						Ingress TS	Pdelay_Resp (correction field for residence time and Ingress Asym Cor)
2’b01	0	1	Ordinary or Boundary Slave with support for peer delay mechanism or Peer to Peer Transparent	0	1	Ingress TS	Sync (correction field for residence time and Ingress Asym cor) (applicable only for Peer to Peer transparent clock operation)
				1	1	x	Delay_Req (originTimestamp field) Delay_Req (correction field for Egress Asym cor)
				1	1	x	Pdelay_Req (originTimestamp field) Pdelay_Req (correction field for Egress Asym Cor)
				0	1	Ingress TS for Pdelay_Req	Pdelay_Resp (correction field for turnaround time and Ingress Asym Cor)
2’b01	1	1	Ordinary or Boundary Master with support for peer delay mechanism	0	1	x	Sync (originTimestamp field) Sync (correction field for sub-nanosecond cor)
				1	1	x	Pdelay_Req (originTimestamp

SNAPTYPSEL	TSMSTRENA	TSEVNTENA	Mode	TTSE(2)	OSTC(3)	TTS(4)	Messages Processed on Tx
							field) Pdelay_Req (correction field for Egress Asym Cor)
				0	1	Ingress TS for Pdelay_Req	Pdelay_Resp (correction field for turnaround time and Ingress Asym Cor)
2’b10	x	x	End-to-End Transparent	0	1	Ingress TS	Sync (correction field for residence time and Ingress Asym cor)
						Ingress TS	Delay_Req (correction field for residence time and Egress Asym Cor)
2’b11	x	x	Peer-to-Peer Transparent	0	1	Ingress TS	Sync (correction field for residence time and Ingress Asym cor)
				1	1	x	Pdelay_Req (originTimestamp field) Pdelay_Req (correction field for Egress Asym Cor)
				0	1	Ingress TS for Pdelay_Req	Pdelay_Resp (correction field for turnaround time and Ingress Asym Cor)

1. The per packet control values provided here are the recommended settings used by devices in typical PTP operation, for the programmed mode.

2. TTSE represents TTSE bit of transmit descriptor. The TTSE function is independent of the OST function and the programmed operation mode for OST. The MAC captures and returns the timestamp when the TTSE bit is set.

3. OSTC represents OSTC bit of transmit context descriptor.

4. TTS represents the timestamp value provided in the TTSH, TTSL fields of transmit normal descriptor (write back format).

Note: Residence time/turnaround time is calculated as the difference between the captured timestamp (egress timestamp) and the ingress timestamp. When sub-nanosecond feature is enabled, residence time calculation includes sub-nanosecond accuracy. Clocks supporting peer delay mechanism do not use delay request or response, but it is included in OST for flexibility.

Enabling One-Step Timestamp

One-step timestamp of packets can be enabled by setting bit27 (OSTC) in the send upper and lower descriptor TDES3. To update the correction fields in some PTP packets, you must timestamp the exits in the TSSL and TSSH fields.

35.5.7 Transmit Checksum Offload Engine

In the transmit path MAC calculates the checksum and inserts it in the Tx packet. This feature helps in reducing the load on the software and can improve the overall throughput of the system.

The checksum offload engine module supports two types of checksum calculation and insertion. The checksum engine can be controlled for each packet by setting the CIC bits (TDES3 bits [17:16]).

Note: The checksum for TCP, UDP, or ICMP is calculated over a complete packet, and then inserted into its corresponding header field. Because of this requirement, when this function is enabled, the Tx FIFO automatically operates in the store-and-forward mode even if the ETH is configured for Threshold (cut-through) mode.

Hint: See IETF specifications RFC 791, RFC 793, RFC 768, RFC 792, RFC 2460, and RFC 4443 for IPv4, TCP, UDP, ICMP, IPv6, and ICMPv6 packet header specifications, respectively.

35.5.7.1 IP Header Checksum Engine

In IPv4 datagrams, the integrity of the header fields is indicated by the 16-bit Header Checksum field (the eleventh and twelfth bytes of the IPv4 datagram). The COE detects an IPv4 datagram when the Type field of Ethernet packet has the value 0x0800 and the Version field of IP datagram has the value 0x4. The checksum field of the input packet is ignored during calculation and replaced with the calculated value.

Note: IPv6 headers do not have a checksum field. Therefore, the COE does not modify the IPv6 header fields.

The result of the IP header checksum calculation is indicated by the IP header error status bit in the transmit status (bit 0 of the TDES3 field in the normal Description format). This status bit is set if there is inconsistency between the Ethernet type field and the version field of the IP header, or if the data size of the Ethernet packet is insufficient (as indicated by the IP header length field). In other words, this bit is set when there is an IP header error in the following cases:

- For IPv4 datagrams:
  - The received Ethernet type is 0x0800, but the Version field of IP header is not equal to 0x4.
  - The IPv4 Header Length field indicates a value less than 0x5 (20 bytes).
  - The total packet length is less than the value given in the IPv4 Header Length field.
- For IPv6 datagrams:
  - The Ethernet type is 0x86dd but the IP header Version field is not equal to 0x6.
  - The packet ends before the IPv6 header (40 bytes) or extension header (as given in the corresponding Header Length field in an extension header) is completely received.

35.5.7.2 TCP/UDP/ICMP Checksum Engine

The TCP/UDP/ICMP Checksum Engine processes the IPv4 or IPv6 header (including extension headers) and determines whether the encapsulated payload is TCP, UDP, or ICMP. The checksum is calculated for the TCP, UDP, or ICMP payload and inserted into its corresponding field in the header. The Tx COE can work in the following two modes:

- The TCP, UDP, or ICMPv6 pseudo-header is not included in the checksum calculation and is assumed to be present in the Checksum field of the input packet. This engine includes the Checksum field in the checksum calculation, and then replaces the Checksum field with the final calculated checksum.
- The engine ignores the Checksum field, includes the TCP, UDP, or ICMPv6 pseudo-header data into the checksum calculation, and overwrites the checksum field with the final calculated value.

Note: For ICMP-over-IPv4 packets, the Checksum field in the ICMP packet must always be 16'h0000 in both modes, because pseudo-headers are not defined for such packets. If it does not equal 16'h0000, an incorrect checksum may be inserted into the packet.

The result of this operation is indicated by the payload checksum error status bit in the transmit status vector (bit 12 of the TDES1 field in the normal Description read format). When a data packet is forwarded to the MAC transmit engine in store-and-forward mode without the packet end-of-packet (EOP) being written into the FIFO, or when the packet ends before the number of bytes indicated in the payload length field of the IP header is received, the engine sets the payload checksum error status bit. If the packet length exceeds the indicated payload length, the COE (Checksum Offload Engine) will ignore the excess as padding bytes and will not report an error. When the engine detects the first type of error, it will not modify the TCP, UDP, or ICMP headers. For the second type of error, it will still insert the calculated checksum into the corresponding header field.

Table 35-21 describes the functions supported by Transmit Checksum Offload engine based on the packet type. When the MAC does not insert the checksum, it is indicated as “No” in the table.

Table 35-21 Transmit Checksum Offload Engine Functions for Different Packet Types

Packet Type	Hardware IP Header Checksum Insertion	Hardware TCP/UDP Checksum Insertion
Non-IPv4 or IPv6 packet	No	No
IPv4 with TCP, UDP, or ICMP	Yes	Yes
IPv4 packet has IP options (IP header is longer than 20 bytes)	Yes	Yes
Packet is an IPv4 fragment	Yes	No
IPv6 packet with the following next header fields in main or extension headers: ■ Hop-by-hop options (in IPv6 main header) ■ Hop-by-hop options (in IPv6 extension header) ■ Destinations options ■ Routing (with segment left 0) ■ Routing (with segment left > 0) ■ TCP, UDP, or ICMP ■ Authentication ■ Any other next header field in main or extension headers	■ Not Applicable ■ Not Applicable ■ Not Applicable ■ Not Applicable ■ Not Applicable ■ Not Applicable ■ Not Applicable ■ Not Applicable	■ Yes ■ No ■ Yes ■ No ■ No ■ Yes ■ No ■ No
IPv4 packet has TCP header with Options fields	Yes	Yes
IPv4 Tunnels: ■ IPv4 packet in an IPv4 tunnel ■ IPv6 packet in an IPv4 tunnel	■ Yes (IPv4 tunnel header) ■ Yes (IPv4 tunnel header)	■ No ■ No
IPv6 Tunnels: ■ IPv4 packet in an IPv6 tunnel	■ Not Applicable	■ No

Packet Type	Hardware IP Header Checksum Insertion	Hardware TCP/UDP Checksum Insertion
■ IPv6 packet in an IPv6 tunnel	■ Not Applicable	■ No
IPv4 packet has 802.3ac tag (with C-VLAN Tag or S-VLAN Tag when enabled).	Yes	Yes
IPv6 packet has 802.3ac tag (with C-VLAN Tag or S-VLAN Tag when enabled).	Not Applicable	Yes
IPv4 frames with security features (such as encapsulated security payload)	Yes	No
IPv6 frames with security features (such as encapsulated security payload)	Not Applicable	No

35.5.8 Receive Checksum Offload Engine

The ETH provides the Checksum Offload Engine that is used to detect any error in an IPv4 or IPv6 packet in the receive path. The MAC verifies the checksum field of the received packet with the internally calculated checksum and provides the status.

The Receive Checksum Offload engine is used to detect errors in IP packets by calculating the header checksum and further matching it with the received header checksum. This engine also identifies a TCP, UDP, or ICMP payload in received IP packets and calculates the checksum of such payloads appropriately.

Here, both IPv4 and IPv6 packet in the received Ethernet packets are detected and processed for data integrity. The MAC receiver identifies IPv4 or IPv6 packets by checking for value 0x0800 or 0x86DD, respectively, in the Type field of the received Ethernet packet. This identification is applicable to single VLAN-tagged packets. It is also applicable to double VLAN-tagged packets when the Enable Double VLAN Processing option is selected and the EDVLP bit of the MAC VLAN Tag register is set.

The Rx COE calculates the IPv4 header checksums and checks that they match the received IPv4 header checksums. The result of this operation (pass or fail) is given to the RFC module for insertion into the receive status word. The IP Header Error bit is set for any mismatch between the indicated payload type (Ethernet Type field) and the IP header version, or when the received packet does not have enough bytes, as indicated by the Length field of the IPv4 header (or when fewer than 20 bytes are available in an IPv4 or IPv6 header).

Packets with TCP/IP errors (header or payload) are dropped in MTL when DISTCPEF bit of the MTL RxQ Operation Mode register is reset and FEP bit is set.

This engine also identifies a TCP, UDP, or ICMP payload in the received IP datagrams (IPv4 or IPv6) and calculates the checksum of such payloads properly, as defined in the TCP, UDP, or ICMP specifications.

This engine includes the TCP, UDP, or ICMPv6 pseudo-header bytes for checksum calculation and checks whether the received checksum field matches the calculated value. The result of this operation is given as a Payload Checksum Error bit in the receive status word. This status bit is also set if the length of the TCP, UDP, or ICMP payload does not match the expected payload length given in the IP header.

Table 35-22 describes the functions supported by the Rx COE based on the packet type. When the payload of an IP packet is not processed (indicated as "No" in the table), the information (whether the checksum engine is bypassed or not) is given in the receive status.

Note: The MAC does not append any payload checksum bytes to the received Ethernet packets.

Table 35-22 Receive Checksum Offload Engine Functions for Different Packet Types

Data packet type	hardware IP header checksum verification	hardware TCP/UDP checksum verification
Non-IPv4 or IPv6	No	No
IPv4 with TCP, UDP, or ICMP	IPv4 with TCP, UDP, or ICMP	IPv4 with TCP, UDP, or ICMP
IPv4 header's protocol field contains a protocol other than TCP, UDP, or ICMP	IPv4 header's protocol field contains a protocol other than TCP, UDP, or ICMP	IPv4 header's protocol field contains a protocol other than TCP, UDP, or ICMP
IPv4 packet has IP options (IP header is longer than 20 bytes)	IPv4 packet has IP options (IP header is longer than 20 bytes)	IPv4 packet has IP options (IP header is longer than 20 bytes)
Packet is an IPv4 fragment	Yes	No
IPv6 packet with the following next header fields in main or extension headers: ■ Hop-by-hop options (in IPv6 main header) ■ Hop-by-hop options (in IPv6 extension header) ■ Destinations options ■ Routing (with segment left 0) ■ Routing (with segment left > 0) ■ TCP, UDP, or ICMP ■ Any other next header field in main or extension headers	■ Not Applicable ■ Not Applicable ■ Not Applicable ■ Not Applicable ■ Not Applicable ■ Not Applicable ■ Not Applicable	■ Yes ■ No ■ Yes ■ Yes ■ No ■ Yes ■ No
IPv4 packet has TCP header with Options fields	Yes	Yes
IPv4 Tunnels: ■ IPv4 packet in an IPv4 tunnel ■ IPv6 packet in an IPv4 tunnel	■ Yes (IPv4 tunnel header) ■ Yes (IPv4 tunnel header)	■ No ■ No
IPv6 Tunnels: ■ IPv4 packet in an IPv6 tunnel ■ IPv6 packet in an IPv6 tunnel	■ Not Applicable ■ Not Applicable	■ No ■ No
IPv4 packet has 802.3ac tag (with C-VLAN Tag or S-VLAN Tag when enabled)	Yes	Yes
IPv6 packet has 802.3ac tag (with C-VLAN Tag or S-VLAN Tag when enabled)	Not Applicable	Yes
IPv4 frames with security features (such as encapsulated security payload)	Yes	No
IPv6 frames with security features (such as encapsulated security payload)	Not Applicable	No

35.5.9 Low-Power Management (PMT)

The ETH supports the following techniques to save power.

- Magic Packet
- Remote Wakeup

The Magic Packet and Remove Wakeup techniques are used to save power in the host system when it is idle (Sleep mode) and has to be woken up only at the reception of specific packets from the Ethernet network. In the Sleep mode, the power to the host logic along with majority of the ETH (except the MAC receiver logic), can be shut down. On receiving the specific packets from the network, the MAC provides the trigger to restore the power to the host system and come back to normal state.

When low-power mode is enabled in the PMT module, the MAC will discard all received data packets and will not forward any packets to the MTL Rx queue or application.

35.5.9.1 Description of Magic Packet Mode

In the Magic Packet based power saving is a mode, the reception of valid magic packet by MAC receiver triggers an exit from low-power mode. The MAC enters power saving mode when PWRDWN bit of MAC PMT Control Status register is programmed to 1. Exit from the magic packet based low-power mode is enabled by setting the MGKPKTEN bit of MAC PMT Control Status register to 1.

The magic packet contains a unique pattern at any offset after the Destination address, Source address, and Length/Type fields. In addition to the unique pattern matching, the MAC receiver also checks for the following, to detect the received packet as a valid magic packet:

- The packet must be addressed to it (Destination Address of the received packet should perfect match the MAC Address0 High and MAC Address0 Low registers) or with multicast/broadcast address
- The packet must not have length error, FCS error, dribble bit error, MII error, and collision
- The packet must not be runt (length including Ethernet header and FCS is at least 64 bytes)

Note: The Magic Packet functionality is based on the Advanced Micro Device (AMD) Magic Packet technology white paper. Regardless of the programmed values in the WD bit of the MAC configuration register and the PWE bit of the MAC watchdog timeout register, the watchdog timeout limit for Magic Packet is always 2048 bytes.

Magic Packet Data Format

The content of the unique pattern in magic packet is described as:

- Bytes of all-ones (48'hFF_FF_FF_FF_FF_FF) called the synchronization stream. There can be more than six bytes of 8'hFF, but last 6 are considered.
- The synchronization stream is immediately followed by 16 repetitions of Destination address field of the packet (MAC Address (MAC Address0 High and MAC Address0 Low registers) or multicast/broadcast address).
- No break or interruption between synchronization stream and first repetition of Destination address field or within its 16 repetitions.

If the MAC address of a node is 48'h00_11_22_33_44_55, the MAC scans for the following data sequence:

(Destination Address) (Source Address) (Length/Type) …………… FF FF FF FF FF FF

00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55

00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55

00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55

00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55

……………………………………………………………………………..CRC

35.5.9.2 Description of Remote Wakeup Packet Mode

In the Remote Wakeup Magic Packet based power saving mode, the reception of expected remote wakeup packet by MAC receiver triggers the exit from low-power mode. The MAC enters power saving mode when PWRDWN bit of MAC PMT Control Status register is programmed to 1. Exit from the remote wakeup based low-power mode is enabled by programming RWKPKTEN bit of MAC PMT Control Status register to 1.

The MAC implements a filter lookup table (programmed through MAC RWK Packet Filter register) in which CRC, offset, and byte mask of the pattern embedded in remote wakeup packet and the filter operation commands are programmed.

The pattern embedded in the remote wakeup packet is located at any offset after the Destination address and Source address fields. In addition to the CRC match for the pattern, the MAC receiver also checks the following, to detect the received packet as a valid remote wakeup packet:

- The packet must be addressed to it (Destination Address of the received packet should perfect match the MAC Address0 High and MAC Address0 Low registers) or with multicast/broadcast address.
- The packet must not have length error, FCS error, dribble bit error, MII error, and collision.
- The packet must not be runt (length including Ethernet header and FCS is at least 64 bytes).

When a valid remote wakeup packet is received, the MAC receiver sets the RWKPRCVD bit in MAC PMT Control Status register and triggers the interrupt on output port. The PMTIS bit in MAC Interrupt Status register is set when power-gating is not enabled in low-power mode. An interrupt is triggered to the application when interrupt is enabled (PMTIE bit in MAC Interrupt Enable register is set) and CSR clock is not gated off in low-power mode.

Remote Wake-Up Packet Filters

When Remote Wakeup based power saving mode is selected, it allows selection of 4, 8 or 16 Remote Wakeup Filters. The Remote Wakeup Filters structure is shown in Table 35-23.

Table 35-23 Remote Wake-Up Packet Filter Register

<table>
<tr><th>Register</th><th>31</th><th>30</th><th>29</th><th>28</th><th>27</th><th>26</th><th>25</th><th>24</th><th>23</th><th>22</th><th>21</th><th>20</th><th>19</th><th>18</th><th>17</th><th>16</th><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td>Wkuppktfilter_reg0</td><td colspan="32">Filter 0bytes Mask</td></tr>
<tr><td>Wkuppktfilter_reg 1</td><td colspan="32">Filter 1bytes Mask</td></tr>
<tr><td>Wkuppktfilter_reg 2</td><td colspan="32">Filter 2bytes Mask</td></tr>
<tr><td>Wkuppktfilter_reg 3</td><td colspan="32">Filter 3bytes Mask</td></tr>
<tr><td>Wkuppktfilter_reg 4</td><td colspan="4">RSVD</td><td colspan="4">Filter 3 command</td><td colspan="4">RSVD</td><td colspan="4">Filter 2 command</td><td colspan="4">RSVD</td><td colspan="4">Filter 1 command</td><td colspan="4">RSVD</td><td colspan="4">Filter 0 command</td></tr>
<tr><td>Wkuppktfilter_reg 5</td><td colspan="8">Filter 3 offset</td><td colspan="8">Filter 2 offset</td><td colspan="8">Filter 1 offset</td><td colspan="8">Filter 0 offset</td></tr>
<tr><td>Wkuppktfilter_reg 6</td><td colspan="16">Filter 1 CRC-16</td><td colspan="16">Filter 0 CRC-16</td></tr>
<tr><td>Wkuppktfilter_reg 7</td><td colspan="16">Filter 3 CRC-16</td><td colspan="16">Filter 2 CRC-16</td></tr>
</table>

The fields of Remote Wakeup Filter are as follows:

Filter i Byte Mask

The filter i byte mask register defines the bytes of the packet that are examined by filter i (0, 1, 2, 3) to determine whether or not a packet is a wake-up packet.

- The MSB (31st bit) must be zero.
- Bit j [30:0] is the byte mask.
- If Bit j (byte number) of the byte mask is set, the CRC block processes the Filter i Offset + j of the incoming packet; otherwise Filter i Offset + j is ignored

Filter i Command

- Bit 3 specifies the address type, defining the destination address type of the pattern. When the bit is set, the pattern applies to only multicast packets; when the bit is reset, the pattern applies only to unicast packet
- Bit 2 (Inverse Mode), when set, reverses the logic of the CRC16 hash function signal, to reject a packet with matching CRC-16 value. Bit 2, along with Bit 1, allows a MAC to reject a subset of remote wake-up packets by creating filter logic such as "Pattern 1 AND NOT Pattern 2".
- Bit 1 (And_Previous) implements the Boolean logic. When set, the result of the current entry is logically ANDed with the result of the previous filter. This AND logic allows a filter pattern longer than 32 bytes by splitting the mask among two, three, or four filters. This depends on the number of filters that have the And_Previous bit set. The details are as follows:
  - The And_Previous bit setting is applicable within a set of 4 filters.
  - Setting of And_Previous bit of filter that is not enabled has no effect, that is setting And_Previous bit of lowest number filter in the set of 4 filters has no effect. For example, setting of And_Previous bit of Filter 0 has no effect.
  - If And_Previous bit is set for filter to form AND chained filter, the AND chain breaks at the point any filter is not enabled. For example: If Filter 2 And_Previous bit is set (bit 1 in Filter 2 command is set) but Filter 1 is not enabled (bit 0 in Filter 1 command is reset), then only Filter 2 result is considered. If Filter 2 And_Previous bit is set (bit 1 in Filter 2 command is set), Filter 3 And_Previous bit is set (bit 1 in Filter 3 command is set), but Filter 1 is not enabled (bit 0 in Filter 1 command is reset), then only Filter 2 result ANDed with Filter 3 result is considered. If Filter 2 And_Previous bit is set (bit 1 in Filter 2 command is set), Filter 3 And_Previous bit is set (bit 1 in Filter 3 command is set), but Filter 2 is not enabled (bit 0 in Filter 2 command is reset), then since setting of Filter 2 And_Previous bit has no effect only Filter 1 result ORed with Filter 3 result is considered.
  - If filters chained by And_Previous bit setting have complementary programming, then a frame may never pass the AND chained filter. For example, if Filter 2 And_Previous bit is set (bit 1 in Filter 2 command is set), Filter 1 Address_Type bit is set (bit 3 in Filter 1 command is set) indicating multicast detection and Filter 2 Address_Type bit is reset (bit 3 in Filter 2 command is reset) indicating unicast detection or vice versa, a remote wakeup frame does not pass the AND chained filter as a remote wakeup frame cannot be of both unicast and multicast address type.
- Bit 0 is the enable for filter i. If Bit 0 is not set, filter i is disabled.

Filter i Offset

The filter i offset register defines the offset (within the packet) from which the filter i examines the packets.

- This 8-bit pattern-offset is the offset for the filter i first byte to be examined.

- The minimum allowed offset is 12, which refers to the 13th byte of the packet.
- The offset value 0 refers to the first byte of the packet.

Filter i CRC-16

The filter i CRC-16 register contains the CRC-16 value calculated from the pattern and the byte mask programmed in the Remote Wakeup filter register.

- The 16-bit CRC calculation uses the following polynomial:

$$G(x) = x^{16} + x^{15} + x^{2} + 1$$

- Each mask, used in the hash function calculation, is compared with a 16-bit value associated with that mask. Each filter has the following:
  - 32-bit Mask: Each bit in this mask corresponds to one byte in the detected packet. If the bit is 1, the corresponding byte is taken into the CRC-16 calculation.
  - 8-bit Offset Pointer: Specifies the byte to start the CRC-16 computation. The pointer and the mask are used together to locate the bytes to be used in the CRC-16 calculations.

The Remote Wakeup Filter registers are implemented as 8 indirect access registers (wkuppktfilter_reg#i) based on whether 4 Remote Wakeup Filters are selected in the configuration and accessed by application through MAC RWK Filter register. When the Remote Wakeup Filters are to be programmed, the entire set of wkuppktfilter_reg registers must be written. The wkuppktfilter_reg register is programmed by sequentially writing the eight, sixteen or thirty-two register values in MAC RWK Filter register for wkuppktfilter_reg0, wkuppktfilter_reg1, ..., wkuppktfilter_reg7 respectively. The wkuppktfilter_reg register is read in a similar way. The MAC updates the wkuppktfilter_reg register current pointer value in RWKPTR field of MAC PMT Control Status register.

Note: When MAC RWK Filter register is written, the content is transferred from CSR clock domain to PHY receive clock domain after the write operation, there should not be any further write to the MAC RWK Filter register until the first write is updated in PHY receive clock domain. Otherwise, the second write operation does not get updated to the PHY receive clock domain. Therefore, the delay between two writes to the MAC RWK Filter register should be at least 4 cycles of the PHY receive clock.

35.5.10 MAC Management Counters

The ETH supports storing the statistics about the received and transmitted packets in registers that are accessible through the application.

The counters in the MAC Management Counters (MMC) module can be viewed as an extension of the register address space of the CSR module. The MMC module maintains a set of registers for gathering statistics on the received and transmitted packets. The register set includes a control register for controlling the behavior of the registers, two status registers containing interrupts generated (receive and transmit), and Interrupt Enable registers (receive and transmit). These registers are accessible from the Application through the MAC Control Interface (MCI). Each register is 32-bits wide. The write data is qualified with the corresponding mci_be_i signals. Therefore, non-32-bit accesses are allowed as long as the address is word-aligned. The MMCs are accessed using transactions, in the same way the CSR address space is accessed.

The MMC counters are free running. There is no separate enable for the counters to start. If a particular MMC counter is present in the RTL, it starts counting when corresponding packet is received or transmitted. The Receive MMC counters are updated for packets that are passed by the Address Filter (AFM) block. The statistics of packets, dropped by the AFM module, are not updated unless they are runt packets of less than 6 bytes (DA bytes are not received fully). To get statistics of all packets, set Bit 0 in the MAC Packet Filter register.

The following definitions define the terminology used in MMC register descriptions.

- Transmitted packets are considered "good" if transmitted successfully. In other words, a transmitted packet is good if the packets transmission is not aborted because of any of the following errors:
  - Jabber Timeout
  - No Carrier or Loss of Carrier
  - Late Collision
  - Packet Underflow
  - Excessive Deferral
  - Excessive Collision
- Received packets are considered "good" if none of the following errors exists:
  - CRC error
  - Runt packet (shorter than 64 bytes)
  - Alignment error (in 10/100 Mbps only)
  - Length error (non-Type packet only)
  - Out of Range (non-Type packet only, longer than 1518 bytes)
  - MII_RXER Input error
- The maximum transmit frame size depends on the frame type, as follows:
  - Untagged frame max size = 1,518
  - VLAN Frame max size = 1,522
  - Jumbo Frame max size = 9,018
  - Jumbo VLAN Frame max size = 9,022
- The maximum receive packet size depends on the packet type and control bits (JE, S2KP, GPSLCE and EDVLP), as shown in the Table 35-24.

Table 35-24 Size of the Maximum Receive Packet

JE	S2KP	GPSLCE	Untagged Frame max size (bytes)	Single VLAN Frame max size (bytes)	Double VLAN Frame max size (bytes)
1	x	x	9018	9022	9026
0	1	x	2000	2000	2000
0	0	1	GPSL	GPSL + 4	GPSL + 8
0	0	0	1518	1522	1526
1	x	x	9018	9022	9022
0	0	1	GPSL	GPSL + 4	GPSL + 4
0	0	0	1518	1522	1522

35.5.11 Loopback Mode

The MAC supports Loopback of transmitted packets to its receiver. The following are some guidelines for using the loopback mode:

- Enable loopback only with the full-duplex mode. In half-duplex mode, the carrier sense signal (CRS) or collision (COL) signal inputs get sampled which may result into issues such as packet dropping.
- If the loopback mode is enabled without connecting a PHY chip, you should externally generate the Tx and Rx clocks and provide these clocks to the MAC.
- Do not loop back big packets. Big packets may get corrupted in the loopback FIFO.

The MAC does not process ARP or PMT packets that are looped back. To enable this feature, programming of the LM bit in the MAC configuration register is required.

35.5.12 Description

The DMA in the Ethernet subsystem transfers data based on a linked list of descriptors. The application creates the descriptors in the system memory. The ETH supports the following two types of descriptors:

- Normal Descriptor: Normal descriptors are used for packet data and to provide control information applicable to the packets to be transmitted.
- Context Descriptor: Context descriptors are used to provide control information applicable to the packet to be transmitted.

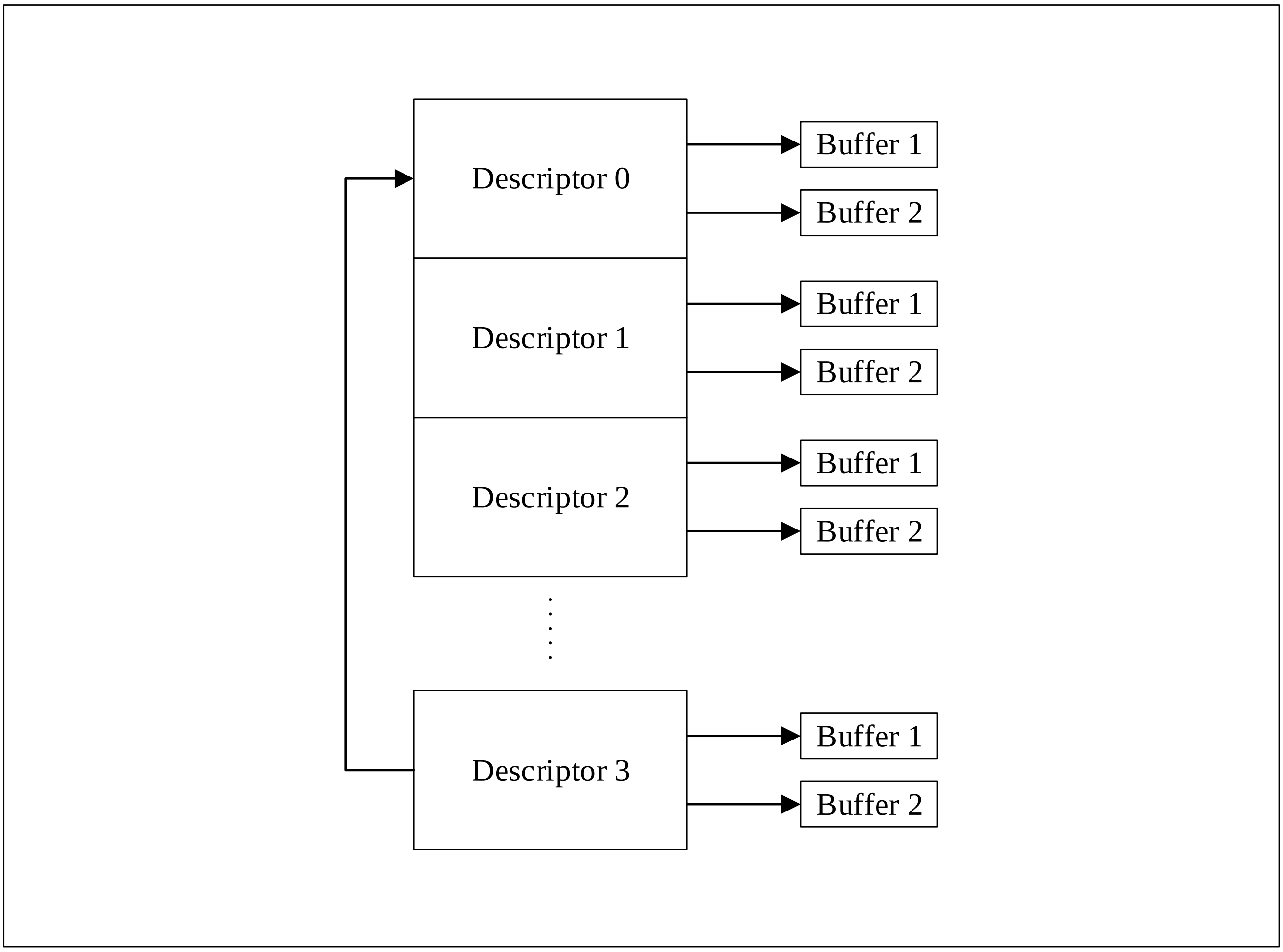
Each normal descriptor contains two buffers and two address pointers. These buffers enable the adapter port to be compatible with various types of memory management schemes.

Note: There is no limit for the number of descriptors that can be used for a single packet.

35.5.12.1 Description Structure

The ETH supports the ring structure for DMA descriptor as shown in Figure 35-16:

Figure 35-16 Description Ring Structure



In Ring structure, descriptors are separated by the Word, number programmed in the DSL field of the DMA CH0 Control register. The application needs to program the total ring length, that is, the total number of descriptors in ring span in the following registers of a DMA channel:

- ETH DMA CH0 TxDesc Ring Length register (ETH_DMACH0TXDRLEN)
- ETH DMA CH0 Rx Control register 2 (ETH_DMACH0RXCTRL2)

The Descriptor Tail Pointer Register contains the pointer to the descriptor address (N). The base address and the current descriptor pointer decide the address of the current descriptor that the DMA can process.

The descriptors up to one location less than the one indicated by the descriptor tail pointer (N – 1) are owned by the DMA. The DMA continues to process the descriptors until the following condition occurs:

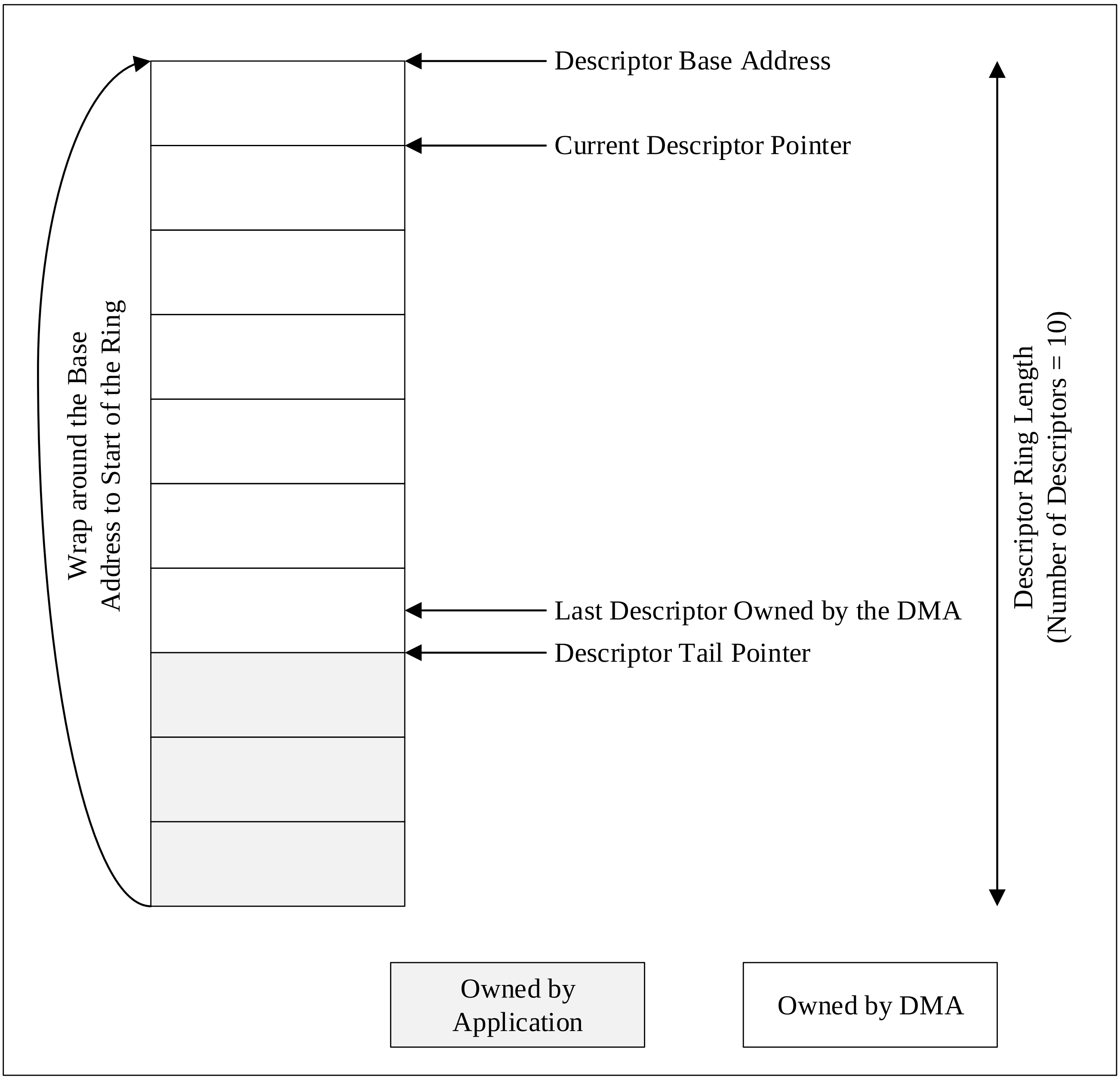
Current Descriptor Pointer == Descriptor Tail Pointer;

The DMA goes into the Suspend mode when this condition occurs. The application must perform a write to the Descriptor Tail pointer register and update the tail pointer so that the following condition is true:

Current Descriptor Pointer < Descriptor Tail Pointer;

The DMA automatically wraps around the base address when the end of ring is reached, as shown in Figure 35-17.

Figure 35-17 DMA Descriptor Ring


Descriptor Base Address
Current Descriptor Pointer
Last Descriptor Owned by the DMA
Descriptor Tail Pointer
Wrap around the Base Address to Start of the Ring
Descriptor Ring Length (Number of Descriptors = 10)
Owned by Application
Owned by DMA

For descriptors owned by the application, the OWN bit of DES3 is reset to 0. For descriptors owned by the DMA, the OWN bit is set to 1. If the application has only one descriptor in the beginning, the application sets the last descriptor address (tail pointer) to Descriptor Base Address + 1. The DMA processes the first descriptor and then waits for the application to advance the tail pointer.

35.5.12.2 Transmit Descriptor

The DMA in ETH requires at least one descriptor for a transmit packet. In addition to two buffers, two byte-count buffers, and two address pointers, the transmit descriptor has control fields which can be used to control the MAC operation on per-transmit packet basis. The Transmit Normal descriptor has two formats: Read format and Write-Back format.

Transmit Normal Descriptor (Read Format)

Table 35-25 shows the Read Format for a Transmit normal descriptor. Table 35-26 through Table 35-29 describe the read format for the Transmit Normal Descriptors: TDES0, TDES1, TDES2, and TDES3.

Table 35-25 Transmit Descriptor Read Format

Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TDES0	Buffer1address[31:0]																															
TDES1	Buffer2 address [31:0]																															
TDES2	IOC	TTSE	Buffer2 length														Reserved		Buffer1 length													
TDES3	OWN	control													CIC		-	Frame Length														

Table 35-26 TDES0 Normal Descriptor (Read Format)

Bit Field	Name	Description
31:0	BUF1AP	Buffer 1 Address Pointer or TSO Header Address Pointer These bits indicate the physical address of Buffer 1. These bits indicate the TSO Header Address pointer when the following bits are set: ■ FD bit of TDES3

Table 35-27 TDES1 Normal Descriptor (Read Format)

Bit Field	Name	Description
31:0	BUF2AP	Buffer 2 or Buffer 1 Address Pointer This bit indicates the physical address of Buffer 2 when a descriptor ring structure is used. There is no limitation for the buffer address alignment. In 40- or 48-bit addressing mode, these bits indicate the most-significant 8- or 16- bits of the Buffer 1 Address Pointer.

Table 35-28 TDES2 Normal Descriptor (Read Format)

Bit Field	Name	Description
31	IOC	Interrupt on Completion. This bit controls the setting of TI and ETI status bits in the DMA CH0 Status register. When ETIC = 1 and TDES2 [LD] = 0, this bit sets the ETI bit. When TDES3 [LD] = 1, this bit sets the TI status bit.
30	TTSE	Transmit Timestamp Enable or External TSO Memory Write Enable. This bit enables the IEEE1588 time stamping for Transmit packet referenced by the descriptor, if TSE bit is not set.
29:16	B2L	Buffer 2 Length. The driver sets this field. When set, this field indicates Buffer 2 length.

Bit Field	Name	Description
15:14	Reserved	Reserved.
13:0	B1L	Buffer 1 Length For Header length only bits [9:0] are taken. The size 13:0 is applicable only when interpreting buffer 1 length.

Table 35-29 TDES3 Normal Descriptor (Read Format)

Bit Field	Name	Description
31	OWN	Own Bit. When this bit is set, it indicates that the DMA owns the descriptor. When this bit is reset, it indicates that the application owns the descriptor. The DMA clears this bit after it completes the transfer of data given in the associated buffer(s).
30	CTXT	Context Type. This bit should be set to 1'b0 for normal descriptor.
29	FD	First Descriptor. When this bit is set, it indicates that the buffer contains the first segment of a packet.
28	LD	Last Descriptor. When this bit is set, it indicates that the buffer contains the last segment of the packet. When this bit is set, the B1L or B2L field should have a non-zero value.
27:26	CPC	CRC Pad Control. This field controls the CRC and Pad Insertion for Tx packet. This field is valid only when the first descriptor bit (TDES3 [29]) is set. The following list describes the values of Bits[27:26]: ■ 2'b00: CRC and Pad Insertion The MAC appends the cyclic redundancy check (CRC) at the end of the transmitted packet of length greater than or equal to 60 bytes. The MAC automatically appends padding and CRC to a packet with length less than 60 bytes. ■ 2'b01: CRC Insertion (Disable Pad Insertion) The MAC appends the CRC at the end of the transmitted packet but it does not append padding. The application should ensure that the padding bytes are present in the packet being transferred from the Transmit Buffer, that is, the packet being transferred from the Transmit Buffer is of length greater than or equal to 60 bytes. ■ 2'b10: Disable CRC Insertion The MAC does not append the CRC at the end of the transmitted packet. The application should ensure that the padding and CRC bytes are present in the packet being transferred from the Transmit Buffer. ■ 2'b11: CRC Replacement The MAC replaces the last four bytes of the transmitted packet with recalculated CRC bytes. The application should ensure that the padding and CRC bytes are present in the packet being transferred from the Transmit Buffer.

Bit Field	Name	Description
		This field is valid only for the first descriptor.
25:18	Reserved	Reserved
17:16	CIC	Checksum Insertion Control or TCP Payload Length. These bits control the checksum calculation and insertion. The following list describes the bit encoding: ■ 2'b00: Checksum Insertion Disabled. ■ 2'b01: Only IP header checksum calculation and insertion are enabled. ■ 2'b10: IP header checksum and payload checksum calculation and insertion are enabled, but pseudo-header checksum is not calculated in hardware. ■ 2'b11: IP Header checksum and payload checksum calculation and insertion are enabled, and pseudo-header checksum is calculated in hardware. This field is valid only for the first descriptor.
15	Reserved	Reserved.
14:0	FL	Frame Length or TCP Payload Length. This field is equal to the length of the packet to be transmitted in bytes. This field is equal to the total length of the packet to be transmitted: Ethernet Header Length + TCP/IP Header Length – Preamble Length – SFD Length + Ethernet Payload Length.

Transmit Normal Descriptor (Write-Back Format)

The write-back format of the Transmit Descriptor includes timestamp low, timestamp high, OWN, and Status bits.

The write-back format is applicable only for the last descriptor of the corresponding packet. The LD bit (TDES3 [28]) is set in the descriptor where the DMA writes back the status and timestamp information for the corresponding Transmit packet.

Table 35-30 displays the write-back format for transmitting normal Description descriptors. Table 35-31 to Table 35-34 describe the write-back formats for transmitting normal Description descriptors: TDES0, TDES1, TDES2, and TDES3.

Table 35-30 Transmit Descriptor Write-Back Format

Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TDES0	Timestamp Low																															
TDES1	Timestamp High																															
TDES2	Reserved																															
TDES3	OWN	Status[30:0]																														

Table 35-31 TDES0 Normal Descriptor (Write-Back Format)

Bit Field	Name	Description
31:0	TTSL	Transmit Packet Timestamp Low. The DMA updates this field with least significant 32 bits of the timestamp captured for the corresponding Transmit packet. The DMA writes the timestamp only if TTSE bit of TDES2 is set in the first descriptor of the packet. This field has the timestamp only if the Last Segment bit (LS) in the descriptor is set and the Timestamp status (TTSS) bit is set.

Table 35-32 TDES1 Normal Descriptor (Write-Back Format)

Bit Field	Name	Description
31:0	TTSH	Transmit Packet Timestamp High. The DMA updates this field with the most significant 32 bits of the timestamp captured for corresponding transmit packet. The DMA writes the timestamp only if the TTSE bit of TDES2 is set in the first descriptor of the packet. This field has the timestamp only if the Last Segment bit (LS) in the descriptor is set and Timestamp status (TTSS) bit is set.

Table 35-33 TDES2 Normal Descriptor (Write-Back Format)

Bit Field	Name	Description
31:0	Reserved	Reserved.

Table 35-34 TDES3 Normal Descriptor (Write-Back Format)

Bit Field	Name	Description
31	OWN	Own Bit. When this bit is set, it indicates that the ETH DMA owns the descriptor. The DMA clears this bit when it completes the packet transmission. After the write-back is complete, this bit is set to 'b0.
30	CTXT	Context Type. This bit should be set to 'b0 for Normal descriptor.
29	FD	First Descriptor. This bit indicates that the buffer contains the first segment of a packet.
28	LD	Last Descriptor. This bit is set 'b1 for last descriptor of a packet. The DMA writes the status fields only in the last descriptor of the packet.
27:24	Reserved	Reserved.
23	DE	Descriptor Error. When this bit is set, it indicates that the descriptor content is incorrect. The DMA sets this bit during write-back while closing the descriptor.

Bit Field	Name	Description
		Descriptor Errors can be: ■ Incorrect sequence from the context descriptor. For example, a location after the first descriptor for a packet. ■ All 1s ■ CTXT is set to 1 along with LD or FD bits set to 1. *Note 1: When Descriptor Error occurs due to All 1s or CTXT, LD, and FD bits set to 1, the Transmit DMA closes the transmit descriptor with DE and LD bits set to 1. When IOC bit in TDES2 of corresponding first descriptor is set to 1, Transmit DMA sets the TI bit in the DMA CH0 Status register* *Note 2: Based on CTXT, LD, and FD bits of the transmit descriptor, the subsequent descriptor might be considered as the First Descriptor (even if FD bit is not set) and partial packet is sent.*
22:18	Reserved	Reserved.
17	TTSS	Tx Timestamp Status. This status bit indicates that a timestamp has been captured for the corresponding transmit packet. When this bit is set, TDES0 and TDES1 have timestamp values that were captured for the Transmit packet. This field is valid only when the Last Segment control bit (TDES3 [28]) in a descriptor is set. This bit is valid only when IEEE1588 timestamp feature is enabled; otherwise, it is reserved.
16	Reserved	Reserved.
15	ES	Error Summary. This bit indicates the logical OR of the following bits: TDES3[0]: IP Header Error TDES3[14]: Jabber Timeout TDES3[13]: Packet Flush TDES3[12]: Payload Checksum Error TDES3[11]: Loss of Carrier TDES3[10]: No Carrier TDES3[9]: Late Collision TDES3[8]: Excessive Collision TDES3[3]: Excessive Deferral TDES3[2]: Underflow Error This bit is also set when EUE (bit 16) is set.
14	JT	Jabber Timeout. This bit indicates that the MAC transmitter has experienced a jabber time-out. This bit is set only when the JD bit of the MAC Configuration register is not set.
13	PF	Packet Flushed. This bit indicates that the DMA or MTL flushed the packet because of a software flush command given by the CPU.
12	PCE	Payload Checksum Error. This bit indicates that the Checksum Offload engine had a failure and did not insert any checksum into the encapsulated TCP, UDP, or ICMP payload. This failure can be either

Bit Field	Name	Description
		because of insufficient bytes, as indicated by the Payload Length field of the IP Header or the MTL starting to forward the packet to the MAC transmitter in Store-and-Forward mode without the checksum having been calculated yet. This second error condition only occurs when the Transmit FIFO depth is less than the length of the Ethernet packet being transmitted to avoid deadlock, the MTL starts forwarding the packet when the FIFO is full, even in the store-and-forward mode. This error can also occur when Bus Error is detected during packet transfer.
11	LOC	Loss of Carrier. This bit indicates that Loss of Carrier occurred during packet transmission (that is, the CRS signal was inactive for one or more transmit clock periods during packet transmission). This is valid only for the packets transmitted without collision and when the MAC operates in the half-duplex mode.
10	NC	No Carrier. This bit indicates that the carrier sense signal form the PHY was not asserted during transmission.
9	LC	Late Collision. This bit indicates that packet transmission was aborted because a collision occurred after the collision window (64 byte times including Preamble in MII mode and 512 byte times including Preamble and Carrier Extension in GMII mode). This bit is not valid if Underflow Error is set.
8	EC	Excessive Collision. This bit indicates that the transmission was aborted after 16 successive collisions while attempting to transmit the current packet. If the DR bit is set in the MAC Configuration register, this bit is set after first collision and the transmission of the packet is aborted.
7:4	CC	Collision Count. This 4-bit counter value indicates the number of collisions occurred before the packet was transmitted. The count is not valid when the EC bit is set.
3	ED	Excessive Deferral. This bit indicates that the transmission ended because of excessive deferral of over 24,288 bit times if DC bit is set in the MAC Configuration register. When TBS is enabled in full duplex mode and this bit is set, it indicates that the frame has been dropped after the expiry time has reached.
2	UF	Underflow Error. This bit indicates that the MAC aborted the packet because the data arrived late from the system memory. The underflow error can occur because of either of the following conditions: ■ The DMA encountered an empty Transmit Buffer while transmitting the packet ■ The application filled the MTL Tx FIFO slower than the MAC transmit rate The transmission process enters the suspended state and sets the underflow bit corresponding to a queue in the MTL Interrupt Status register.
1	DB	Deferred Bit. This bit indicates that the MAC deferred before transmitting because of presence of

Bit Field	Name	Description
		carrier. This bit is valid only in the half-duplex mode.
0	IHE	IP Header Error. When IP Header Error is set, this bit indicates that the Checksum Offload engine detected an IP header error. This bit is valid only when Tx Checksum Offload is enabled. Otherwise, it is reserved. If COE detects an IP header error, it still inserts an IPv4 header checksum if the Ethernet Type field indicates an IPv4 payload.

Transmit Context Descriptor

The context descriptor is used to provide the timestamps for one-step timestamp correction.

The Transmit Context descriptor can be provided any time before a packet descriptor. The context is valid for the current packet and subsequent packets. The context descriptor is used to provide the timestamps for one-step timestamp correction and VLAN Tag ID for VLAN insertion feature. Write back is done on a context descriptor only to reset the OWN bit.

Table 35-35 displays the format for transmitting context Description descriptors. Table 35-36 to Table 35-39 describe the formats for transmitting context Description descriptors: TDES0, TDES1, TDES2, and TDES3.

Table 35-35 Transmit Context Descriptor Format

Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TDES0	Timestamp Low																															
TDES1	Timestamp High																															
TDES2	Rsvd																															
TDES3	OWN	Rsvd					Rsvd		CDE	Rsvd							VT															

Table 35-36 TDES0 Context Descriptor

Bit Field	Name	Description
31:0	TTSL	Transmit Packet Timestamp Low. For one-step correction, the driver can provide the lower 32 bits of timestamp in this descriptor word. The DMA uses this value as the low word for doing one-step timestamp correction. This field is valid only if the OSTC and TCMSSV bits of TDES3 context descriptor are set.

Table 35-37 TDES1 Context Descriptor

Bit Field	Name	Description
31:0	TTSH	Transmit Packet Timestamp High. For one-step correction, the driver can provide the upper 32 bits of timestamp in this descriptor. The DMA uses this value as the high word for doing one-step timestamp

Bit Field	Name	Description
		correction. This field is valid only if the OSTC and TCMSSV bits of TDES3 context descriptor are set.

Table 35-38 TDES2 Context Descriptor

Bit Field	Name	Description
31:0	Reserved	Reserved.

Table 35-39 TDES3 Context Descriptor

Bit Field	Name	Description
31	OWN	Own Bit. When this bit is set, it indicates that the ETH DMA owns the descriptor. When this bit is reset, it indicates that the application owns the descriptor. The DMA clears this bit immediately after the read.
30	CTXT	Context Type. This bit should be set to 1'b1 for Context descriptor.
29:28	Reserved	Reserved.
27	OSTC	One-Step Timestamp Correction Enable. When this bit is set, the DMA performs a one-step timestamp correction with reference to the timestamp values provided in TDES0 and TDES1.
26	TCMSSV	One-Step Timestamp Correction Input or MSS Valid. When this bit and the OSTC bit are set, it indicates that the Timestamp Correction input provided in TDES0 and TDES1 is valid. When the OSTC bit is reset and this bit and the TSE bit of TDES3 are set in subsequent normal descriptor, it indicates that the MSS input in TDES2 is valid.
25:24	Reserved	Reserved.
23	CDE	Context Descriptor Error. When this bit is set, it indicates that the descriptor content is incorrect. The DMA sets this bit during write-back while closing the context descriptor. Context Descriptor Errors can be: ■ Incorrect sequence from the context descriptor. For example, a location before the first descriptor for a packet. ■ All 1s. ■ CD, LD, and FD bits set to 1. *Note 1: When the Context Descriptor Error occurs due to All 1s or CTXT, LD, and FD bits set to 1, the Transmit DMA closes the transmit descriptor with DE and LD bits set to 1. When IOC bit in TDES2 of corresponding first descriptor is set to 1, Transmit DMA sets the TI bit in the DMA CH0 Status register* *Note 2: Based on CTXT, LD, and FD bits of the transmit descriptor, the subsequent descriptor might be considered as the First Descriptor (even if FD bit is not set) and partial packet is sent.*

Bit Field	Name	Description
		Note3: This error is categorized as an abnormal event; recovery is only by issuing a software reset (DMA stopping-reconfiguring-restarting recovery mechanism is not supported)
22:16	Reserved	Reserved.
15:0	VT	VLAN Tag. In the N32H487 series, this field is used only as a VLAN tag.

35.5.12.3 Receive Descriptor

The DMA in ETH attempts to read a descriptor only if the Tail Pointer is different from the Base Pointer or current pointer. It is recommended to have a descriptor ring with a length that can accommodate at least two complete packets received by the MAC. Otherwise, the performance of the DMA is impacted greatly because of the unavailability of the descriptors. In such situations, the Rx FIFO in MTL becomes full and starts dropping packets.

All RX descriptors are prepared by the software and given to the DMA as "Normal" Descriptors with the content as shown in Receive Normal Descriptor (Read Format). The DMA reads this descriptor and after transferring a received packet (or part of) to the buffers indicated by the descriptor, the Rx DMA closes the descriptor with the corresponding packet status. The format of this status is given in the "Receive Normal Descriptor (Write-Back Format)".

For some packets, the normal descriptor bits are not enough to write the complete status. For such packets, the RX DMA writes the extended status to the next descriptor (without processing or using the Buffers/ Pointers embedded in that descriptor). The format and content of the descriptor write back is described in "Receive Context Descriptor".

Receive Normal Descriptor (Read Format)

The read format for a Receive Normal descriptor is made up of a header or Buffer 1 address, reserved field, payload or Buffer 2 or Next Descriptor address, a 30-bit reserved filed, OWN bit, and an interrupt bit.

Table 35-40 shows the read format for receiving normal descriptors. Table 35-41 to Table 35-44 describe read formats for receiving normal descriptors: RDES0, RDES1, RDES2, and RDES3.

Table 35-40 Receive Normal Descriptor (Read Format)

Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RDES0	Buffer 1 Address																															
RDES1	Reserved																															
RDES2	Buffer 2 Address																															
RDES3	OWN	IOC	Reserved																													

Note: In the Receive Descriptor (Read Format), if the Buffer Address field is all 0s, ETH does not transfer data to that buffer and skips to the next buffer or next descriptor.

Table 35-41 RDES0 Normal Descriptor (Read Format)

Bit Field	Name	Description
31:0	BUF1AP	Buffer 1 Address Pointer.

Bit Field	Name	Description
		These bits indicate the physical address. The application can program a byte-aligned address for this buffer which means that the LS bits of this field can be non-zero. However, while transferring the start of packet, the DMA performs a Write operation with RDES0 [1:0] as zero. However, the packet data is shifted as per actual offset as given by buffer address pointer. If the address pointer points to a buffer where the middle or last part of the packet is stored, the DMA ignores the offset address and writes to the full location as indicated by the data-width.

Table 35-42 RDES1 Normal Descriptor (Read Format)

Bit Field	Name	Description
31:0	Reserved	Reserved.

Table 35-43 RDES2 Normal Descriptor (Read Format)

Bit Field	Name	Description
31:0	BUF2AP	Buffer 2 Address Pointer. These bits indicate the physical address of Buffer 2. When the SPH bit of the DMA CH0 Control register is set, the buffer address pointer must be bus width-aligned, that is, RDES2[1:0] = 0. LSBs are ignored internally. When the SPH bit of the DMA CH0 Control register is reset, there is no limitations on the RDES2 value. However, the Rx DMA uses the LS Bits of the pointer address only while transferring the start bytes of a packet. If the BUF2AP is giving the address of a buffer in which the middle or last part of a packet is stored, the DMA ignores BUF2AP [1:0] and writes to the complete location.

Table 35-44 RDES3 Normal Descriptor (Read Format)

Bit Field	Name	Description
31	OWN	Own Bit. When this bit is set, it indicates that the ETH DMA owns the descriptor. When this bit is reset, it indicates that the application owns the descriptor. The DMA clears this bit when either of the following conditions is true: ■ The DMA completes the packet reception ■ The buffers associated with the descriptor are full
30	IOC	Interrupt Enabled on Completion. When this bit is set, an interrupt is issued to the application when the DMA closes this descriptor.
29:26	Reserved	Reserved
25	BUF2V	Buffer 2 Address Valid. When this bit is set, it indicates to the DMA that the buffer 2 address specified in

Bit Field	Name	Description
		RDES2 is valid. The application must set this bit so that the DMA can use the address, to which the Buffer 2 address in RDES2 is pointing, to write received packet data.
24	BUF1V	Buffer 1 Address Valid. When set, this indicates to the DMA that the buffer 1 address specified in RDES1 is valid. The application must set this value if the address pointed to by Buffer 1 address in RDES1 can be used by the DMA to write received packet data.
23:0	Reserved	Reserved.

Receive Normal Descriptor (Write-Back Format)

Table 35-45 displays the write-back format for receiving normal Description descriptors. Table 35-46 to Table 35-49 describe the write-back formats for receiving normal Description descriptors: RDES0, RDES1, RDES2, and RDES3.

Table 35-45 Receive Normal Descriptor (Write-Back Format)

<table>
<tr><th>Register</th><th>31</th><th>30</th><th>29</th><th>28</th><th>27</th><th>26</th><th>25</th><th>24</th><th>23</th><th>22</th><th>21</th><th>20</th><th>19</th><th>18</th><th>17</th><th>16</th><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td>RDES0</td><td colspan="16">RSVD</td><td colspan="16">OVT</td></tr>
<tr><td>RDES1</td><td colspan="16">OAM code, or MAC Control</td><td colspan="16">Extended Status</td></tr>
<tr><td>RDES2</td><td colspan="17">MAC Filter</td><td colspan="15">RSVD</td></tr>
<tr><td>RDES3</td><td>OWN</td><td>CTXC</td><td>FD</td><td>LD</td><td colspan="13">Status</td><td colspan="15">Packet Length</td></tr>
</table>

Table 35-46 RDES0 Normal Descriptor (Write-Back Format)

Bit Field	Name	Description
31:16	Reserved	Reserved.
15:0	OVT	Outer VLAN Tag. This field contains the Outer VLAN tag of the received packet if the RS0V bit of RDES3 is set.

Table 35-47 RDES1 Normal Descriptor (Write-Back Format)

Bit Field	Name	Description
31:16	OPC	OAM Sub-Type Code, or MAC Control Packet opcode. **OAM Sub-Type Code** If bits [18:16] of RDES3 are set to 3'b111, this field contains the OAM sub-type and code fields. **MAC Control Packet opcode** The bits [15:8] of RDES3 contains the subtype and bits [7:0] contains the code.

Bit Field	Name	Description
15	TD	Timestamp Dropped. This bit indicates that the timestamp was captured for this packet but it got dropped in the MTL Rx FIFO because of overflow.
14	TSA	Timestamp Available. When Timestamp is present, this bit indicates that the timestamp value is available in a context descriptor word 2 (RDES2) and word 1(RDES1). This is valid only when the Last Descriptor bit (RDES3 [28]) is set. The context descriptor is written in the next descriptor just after the last normal descriptor for a packet.
13	PV	PTP Version. This bit indicates that the received PTP message has the IEEE 1588 version 2 format. When this bit is reset, it indicates the IEEE 1588 version 1 format.
12	PFT	PTP Packet Type. This bit indicates that the PTP message is sent directly over Ethernet.
11:8	PMT	PTP Message Type. These bits are encoded to give the type of the message received: 0000: No PTP message received 0001: SYNC (all clock types) 0010: Follow_Up (all clock types) 0011: Delay_Req (all clock types) 0100: Delay_Resp (all clock types) 0101: Pdelay_Req (in peer-to-peer transparent clock) 0110: Pdelay_Resp (in peer-to-peer transparent clock) 0111: Pdelay_Resp_Follow_Up (in peer-to-peer transparent clock) 1000: Announce 1001: Management 1010: Signaling 1011~1110: Reserved 1111: PTP packet with Reserved message type
7	IPCE	IP Payload Error. When this bit is set, it indicates either of the following: ■ The 16-bit IP payload checksum (that is, the TCP, UDP, or ICMP checksum) calculated by the MAC does not match the corresponding checksum field in the received segment. ■ The TCP, UDP, or ICMP segment length does not match the payload length value in the IP Header field. ■ The TCP, UDP, or ICMP segment length is less than minimum allowed segment length for TCP, UDP, or ICMP. Bit 15 (ES) of RDES3 is not set when this bit is set.
6	IPCB	IP Checksum Bypassed. This bit indicates that the checksum offload engine is bypassed.
5	IPV6	IPv6 header Present. This bit indicates that an IPV6 header is detected.

Bit Field	Name	Description
4	IPV4	IPV4 Header Present. This bit indicates that an IPV4 header is detected.
3	IPHE	IP Header Error. When this bit is set, it indicates either of the following: ■ The 16-bit IPv4 header checksum calculated by the MAC does not match the received checksum bytes. ■ The IP datagram version is not consistent with the Ethernet Type value. ■ Ethernet packet does not have the expected number of IP header bytes. This bit is valid when either Bit 5 or Bit 4 is set.
2:0	PT	Payload Type. These bits indicate the type of payload encapsulated in the IP datagram processed by the Receive Checksum Offload Engine (COE): ■ 3'b000: Unknown type or IP/AV payload not processed ■ 3'b001: UDP ■ 3'b010: TCP ■ 3'b011: ICMP ■ 3'b110: AV Tagged Data Packet ■ 3'b111: AV Tagged Control Packet ■ 3'b101: AV Untagged Control Packet ■ 3'b100: IGMP if IPV4 Header Present bit is set else DCB (LLDP) Control Packet If the COE does not process the payload of an IP datagram because there is an IP header error or fragmented IP, it sets these bits to 3'b000.

Table 35-48 RDES2 Normal Descriptor (Write-Back Format)

Bit Field	Name	Description
31:27	Reserved	Reserved.
26:19	MADRM	MAC Address Match or Hash Value. When the HF bit is reset, this field contains the MAC address register number that matched the Destination address of the received packet. This field is valid only if the DAF bit is reset. When the HF bit is set, this field contains the hash value computed by the MAC. A packet passes the hash filter when the bit corresponding to the hash value is set in the hash filter register.
18	HF	Hash Filter Status. When this bit is set, it indicates that the packet passed the MAC address hash filter. Bits [26:19] indicate the hash value.
17	DAF	Destination Address Filter Fail. When this bit is set, it indicates that the packet failed the DA Filter in the MAC.
16	SAF	SA Address Filter Fail. When this bit is set, it indicates that the packet failed the SA Filter in the MAC.
15	VFS	VLAN Filter Status.

Bit Field	Name	Description
		When this bit is set, this bit indicates that the VLAN Tag of the received packed passed the VLAN filter.
14:0	Reserved	Reserved.

Table 35-49 RDES3 Normal Descriptor (Write-Back Format)

Bit Field	Name	Description
31	OWN	Own Bit. When this bit is set, it indicates that the ETH DMA owns the descriptor. When this bit is reset, it indicates that the application owns the descriptor. The DMA clears this bit when either of the following conditions is true: ■ The DMA completes the packet reception ■ The buffers associated with the descriptor are full
30	CTXT	Receive Context Descriptor. When this bit is set, it indicates that the current descriptor is a context type descriptor. The DMA writes 1'b0 to this bit for normal receive descriptor. When CTXT and FD bits are used together, {CTXT, FD} ■ 2’b00: Intermediate Descriptor ■ 2’b01: First Descriptor ■ 2’b10: Reserved ■ 2’b11: Descriptor Error (due to all 1s) *Note: When Descriptor Error occurs, the Receive DMA closes the receive descriptor indicating Descriptor Error. This receive descriptor is skipped and the buffer addresses are not used to write the packet data. Receive DMA sets the CDE field of the DMA CH0 Status register but does not the RI field even when IOC field is set, as this is not marked as last receive descriptor for the packet. The subsequent valid receive descriptor is used to write the packet data.*
29	FD	First Descriptor. When this bit is set, it indicates that this descriptor contains the first buffer of the packet. If the size of the first buffer is 0, the second buffer contains the beginning of the packet. If the size of the second buffer is also 0, the next descriptor contains the beginning of the packet.
28	LD	Last Descriptor. When this bit is set, it indicates that the buffers to which this descriptor is pointing are the last buffers of the packet.
27	RS2V	Receive Status RDES2 Valid. When this bit is set, it indicates that the status in RDES2 is valid and it is written by the DMA. This bit is valid only when the LD bit of RDES3 is set.
26	RS1V	Receive Status RDES1 Valid. When this bit is set, it indicates that the status in RDES1 is valid and it is written by the DMA. This bit is valid only when the LD bit of RDES3 is set.
25	RS0V	Receive Status RDES0 Valid.

Bit Field	Name	Description
		When this bit is set, it indicates that the status in RDES0 is valid and it is written by the DMA. This bit is valid only when the LD bit of RDES3 is set.
24	CE	CRC Error. When this bit is set, it indicates that a Cyclic Redundancy Check (CRC) Error occurred on the received packet. This field is valid only when the LD bit of RDES3 is set.
23	GP	Giant Packet. When this bit is set, it indicates that the packet length exceeds the specified maximum Ethernet size of 1518, 1522, or 2000 bytes (9018 or 9022 bytes if jumbo packet enable is set). *Note: Giant packet indicates only the packet length. It does not cause any packet truncation.*
22	RWT	Receive Watchdog Timeout. When this bit is set, it indicates that the Receive Watchdog Timer has expired while receiving the current packet. The current packet is truncated after watchdog timeout.
21	OE	Overflow Error. When this bit is set, it indicates that the received packet is damaged because of buffer overflow in Rx FIFO. Note: This bit is set only when the DMA transfers a partial packet to the application. This happens only when the Rx FIFO is operating in the threshold mode. In the store-and-forward mode, all partial packets are dropped completely in Rx FIFO.
20	RE	Receive Error. When this bit is set, it indicates that the RX_ER signal is asserted while the RX_DV signal is asserted during packet reception.
19	DE	Dribble Bit Error. When this bit is set, it indicates that the received packet has a non-integer multiple of bytes (odd nibbles). This bit is valid only in the MII Mode.
18:16	LT	Length/Type Field. This field indicates if the packet received is a length packet or a type packet. The encoding of the 3 bits is as follows: ■ 3'b000: The packet is a length packet ■ 3'b001: The packet is a type packet. ■ 3'b011: The packet is a ARP Request packet type ■ 3'b100: The packet is a type packet with VLAN Tag ■ 3'b101: The packet is a type packet with Double VLAN Tag ■ 3'b110: The packet is a MAC Control packet type ■ 3'b111: The packet is a OAM packet type ■ 3'b010: Reserved
15	ES	Error Summary. When this bit is set, it indicates the logical OR of the following bits: RDES3[24]: CRC Error RDES3[19]: Dribble Error RDES3[20]: Receive Error

Bit Field	Name	Description
		RDES3[22]: Watchdog Timeout RDES3[21]: Overflow Error RDES3[23]: Giant Packet RDES2[17]: Destination Address Filter Fail, when Flexible RX Parser is enabled RDES2[16]: SA Address Filter Fail, when Flexible RX Parser is enabled This field is valid only when the LD bit of RDES3 is set
14:0	PL	Packet Length. These bits indicate the byte length of the received packet that was transferred to system memory (including CRC). This field is valid when the LD bit of RDES3 is set and Overflow Error bits are reset. The packet length also includes the two bytes appended to the Ethernet packet when IP checksum calculation is enabled and the received packet is not a MAC control packet. This field is valid when the LD bit of RDES3 is set. When the Last Descriptor and Error Summary bits are not set, this field indicates the accumulated number of bytes that have been transferred for the current packet.

Receive Context Descriptor

This descriptor is read-only for the application. Only the DMA can write to this descriptor. The context descriptor provides information about the extended status related to the last received packet. The Bit 30 of RDES3 indicates the context type descriptor.

Table 35-50 displays the format for receiving context Description descriptors. Table 35-51 to Table 35-54 describe the formats for receiving context Description descriptors: RDES0, RDES1, RDES2, and RDES3.

Table 35-50 Receive Context Descriptor Format

Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RDES0	Timestamp Low																															
RDES1	Timestamp High																															
RDES2	Reserved																															
RDES3	OWN	CTXT	Reserved																													

Table 35-51 RDES0 Context Descriptor

Bit Field	Name	Description
31:0	RTSL	Receive Packet Timestamp Low. The DMA updates this field with least significant 32 bits of the timestamp captured for corresponding Receive packet. When this field and the RTSH field of RDES1 show all-ones value, the timestamp must be considered as corrupt.

Table 35-52 RDES1 Context Descriptor

Bit Field	Name	Description
31:0	RTSH	Receive Packet Timestamp High. The DMA updates this field with most significant 32 bits of the timestamp captured for corresponding receive packet. When this field and the RTSL field of RDES0 show all-ones value, the timestamp must be considered as corrupt

Table 35-53 RDES2 Context Descriptor

Bit Field	Name	Description
31:0	Reserved	Reserved.

Table 35-54 RDES3 Context Descriptor

Bit Field	Name	Description
31	OWN	Own Bit. When this bit is set, it indicates that the DMA owns the descriptor. When this bit is reset, it indicates that the application owns the descriptor. The DMA clears this bit when either of the following conditions is true: ■ The DMA completes the packet reception ■ The buffers associated with the descriptor are full
30	CTXT	Receive Context Descriptor. When this bit is set, it indicates that the current descriptor is a context descriptor. The DMA writes 1'b1 to this bit for context descriptor. DMA writes 2’b11 to indicate a descriptor error due to all 1s. When CTXT and DE bits are used together, {CTXT, DE} 2’b00: Reserved 2’b01: Reserved 2’b10: Context Descriptor 2’b11: Descriptor Error *Note: When Descriptor Error occurs, the Receive DMA closes the receive descriptor indicating Descriptor Error. This receive descriptor is skipped and the buffer addresses are not used to write the packet data Receive DMA sets the CDE bit in DMA CH0 Status register but does not set the RI field even when IOC is set, as this is not marked as last receive descriptor for the packet. The subsequent valid receive descriptor is used to write the packet data.*
29	DE	Descriptor Error. See the CTXT bit description for details of using the DE bit along with CTXT bit.
28:0	Reserved	Reserved.

35.5.13 Ethernet Interrupt

Interrupts can be generated as a result of various events in the ETH controller. These events are captured in status registers, and interrupt enables are provided for each source of an interrupt such that the interrupt signal is asserted for an event only when the corresponding interrupt enable is set.

The interrupt status and corresponding enable registers are organized in a hierarchical manner so that it is easier for software to traverse and identify the source of an interrupt event quickly. When sbd_intr_o is asserted, the DMA Interrupt Status register is the first level that indicates the major blocks for the interrupt event source. This register is read-only, and it contains bits corresponding to each DMA channel (TX and RX pair), the MTL, and the MAC. The software application must then read one (or more) of the following registers corresponding to the bits that are set:

- ETH MAC Interrupt Status register (ETH_MACINTSTS)
- ETH MTL Interrupt Status register (ETH_MTLINTSTS)
- ETH DMA CH0 Status register (ETH_DMACH0STS)

35.5.13.1 Interrupts from DMA

Interrupts are not queued. If the same interrupt event occurs again before the driver responds to the previous one, no additional interrupts are generated. For example, Receive Interrupt bit [6] of DMA CH0 Status register indicates that one or more packets were transferred to the application buffer.

The driver must scan all descriptors, from the last recorded position to the first one, owned by the DMA to determine how many packets are received.

An interrupt is generated only once for multiple events. The driver must scan the DMA Interrupt Status register for the cause of the interrupt and clear the source in the respective Status register. The sbd_intr_o is cleared only when all the bits of DMA Interrupt Status register are cleared.

Periodic Scheduling of Transmit and Receive Interrupt

It is not preferable to generate interrupts for every packet transferred by DMA (RI and TI) for system throughput performance reasons. The ETH gives the flexibility to schedule the interrupt at regular intervals using two methods:

- Set Interrupt on Completion bit in Transmit descriptor (TDES2 [31]) once for every “required” number of packets to be transmitted.
- Similarly, set the IOC (see RDES3 [30]) bit only at some specific intervals of Receive descriptors. This way, whenever a received packet transfer to system memory is complete and any of the descriptors used for that packet transfer has the IOC bit set, only then the RI event is generated.

In addition, an interrupt timer (DMA CH0 Rx Interrupt Watchdog Timer register) is given for flexible control and periodic scheduling of Receive Interrupt. When this interrupt timer is programmed with a nonzero value, it gets activated as soon as the Rx DMA completes a transfer of a received packet to system memory without asserting the Receive Interrupt because the corresponding interrupt of completion IOC bit (RDES3 [30]) is not set. When this timer runs out as per the programmed value, RI bit is set and the interrupt is asserted if the corresponding RIE is enabled in DMA CH0 Interrupt Enable register. The timer is stopped and cleared before it expires, if the RI is set for a packet transfer whose descriptor’s IOC was set. The timer is reactivated automatically after the next packet transfer is complete without the RI event being generated.

Channel Transfer Complete Interrupt

The Transmit Transfer complete interrupt (TI) and Receive Transfer complete interrupt (RI) is reflected in DMA CH0 Status register. The TI bit is set whenever the Tx DMA channel closes the transmit descriptor in which the IOC

(Interrupt On Completion - TDES2 [31]) bit is set. Similarly, the RI bit is set whenever the Rx DMA channel closes the receive descriptor with LD bit set and in any of the descriptors used for transferring that packet, IOC (Interrupt Enable on completion - RDES3 [30]) bit is set.

The common sbd_intr_o output signal is asserted for the Transfer complete interrupts only when the corresponding interrupts are enabled in DMA CH0 Interrupt Enable register.

Also supports the following per Channel Transfer Complete interrupt signals.

- sbd_perch_tx_intr_o[0] (Transmit Channel 0 Interrupts)
- sbd_perch_rx_intr_o[0] (Receive Channel 0 Interrupts)

The behavior of RI/TI/sbd_perch_tx_intr_o/sbd_perch_rx_intr_o will change based on the setting of the INTM field in the DMA mode register. Table 35-55 explains the behavior of the transmit completion interrupt.

Table 35-55 Transfer Complete Interrupt Behavior

Interrupt Mode	Behavior of the sbd_perch_tx_intr_o and sbd_perch_rx_intr_o	Behavior of TI/RI and sbd_intr_o
INTM = 0	A pulse is asserted on these output signals when corresponding TX/RX transfer complete event (for which IOC descriptor bits are enabled) is detected, irrespective of the corresponding interrupt status.	The TI/RI status signals are set whenever the “Transfer complete” event is detected. The bits get cleared whenever the software driver writes‘1’ to these bits. The sbd_intr_o is asserted when ever the corresponding interrupts are also enabled in the DMA CH0 Interrupt Enable register. The NIS status bit is asserted for RI/TI events.
INTM = 1	These signals reflect the value of corresponding TI/RI bits in the DMA CH0 Status register when the corresponding interrupt enable is set. Therefore, they are level signals and are cleared by the application by writing 1’b1 to the RI/TI status bits. This signal is not asserted when the corresponding interrupt enable bit is not set.	The sbd_intr_o signal and NIS status bit are not asserted for RI/TI events.
INTM = 2	In this mode, RI/TI interrupts are queued. These signals reflect the value of corresponding TI/RI bits in the DMA CH0 Status register when the corresponding interrupt enable is set. They are level signals and cleared by software by writing 1’b1 to the RI/TI status bits. However, it is set again if another TI/RI event(s) is detected before the TI/RI bits are cleared for the previous event.	The RI/TI status bits are set whenever the Transfer Complete event is detected and gets reset whenever software driver clears those bits by writing 1. However, if another Transfer Complete event is detected before it is cleared (serviced) by the software, then ETH automatically set these status bits again. However, the sbd_intr_o signal is not generated based on TI/RI. The NIS status bit is not asserted for RI/TI events.

35.5.13.2 Interrupts from MTL

MTL interrupt events are combined with events in the DMA to generate interrupt signals. The MTL interrupt status register reports the queue number responsible for the event. The MTL queue interrupt control status register should be read to obtain an explanation of the event. MTL interrupts are disabled by default. When the corresponding bit in the MTL queue interrupt control status register is set to 1, each event can trigger an interrupt. MTL interrupt signals are driven by one of the following events:

- Receive Queue Overflow
- Transmit Queue Underflow

35.5.13.3 Interrupts from the MAC

Various events from the MAC receiver, MAC transmitter, or other modules/functions (such as RMON counters) can generate interrupts from the MAC. MAC interrupt events are combined with events in the DMA to generate interrupt signals. MAC interrupts are level-sensitive, meaning the interrupt remains asserted (high) until cleared by the application or software. The MAC interrupt status register describes the events that may cause MAC interrupts. MAC interrupts are disabled by default. When the corresponding bit in the MAC interrupt status register is set to 1, each event can trigger an interrupt. The interrupt register bits only indicate the block reporting the event. To clear the interrupt, the corresponding status register and other registers must be read. MAC interrupt signals are driven by one of the following events:

- receive interrupt
- transmit interrupt
- timestamp interrupt status
- MMC interrupt status
- MMC transmit interrupt
- MMC receive interrupt
- PMT interrupt status
- PHY interrupt

Note: The sideband signal pmt_intr_o is generated by the PMT interrupt. This signal is used for wake-up event detection at the EXTI level.

Note: By default, the MAC interrupt status bits are cleared when the register that contains the source of the interrupt is read. If RCWE bit in the MAC CSR SW Ctrl register is programmed to 1, the MAC interrupt status bits are cleared when the bit that contains the source of the interrupt is explicitly written to 1.

35.5.14 Programming guide

This chapter provides the instructions for initializing the DMA or MAC registers in the proper sequence.

Note: When any register content is being transferred to a different clock domain after a write operation, there should not be any further writes to the same location until the first write is updated. Otherwise, the second write operation does not get updated to the destination clock domain. Therefore, the delay between two writes to the same register location should be at least four cycles of the destination clock (PHY Receive clock, PHY Transmit clock, or PTP clock). This limitation is removed when you select Enable support for back-to-back register writes during RTL configuration.

35.5.14.1 Initializing DMA

Complete the following steps to initialize the DMA:

1 Provide a software reset. This resets all of the MAC internal registers and logic (bit-0 of DMA Mode register).

2 Wait for the completion of the reset process (poll bit 0 of the DMA Mode register, which is only cleared after the reset operation is completed).

3 Program the following fields to initialize the DMA SysBus Mode register:

a) AAL

b) Fixed burst or undefined burst

c) Burst mode values in case of AHB bus interface.

4 Create a descriptor list for transmit and receive. In addition, ensure that the descriptors are owned by DMA (set bit 31 of descriptor TDES3/RDES3).

5 Program the Transmit and Receive Ring length registers (DMA CH0 TxDesc Ring Length register and DMA CH0 RxDesc Ring Length register). The ring length programmed must be at least 4.

Note: The descriptor address from the start to the end of the ring must not cross the 4GB boundary.

6 Initialize the base addresses of the transmit and receive Descriptor rings using the base address of the transmit and receive Descriptors (DMA Channel 0 transmit Descriptor list address register and DMA Channel 0 receive Descriptor list address register). Additionally, program the transmit and receive tail pointer registers (DMA Channel 0 transmit Descriptor tail pointer register and DMA Channel 0 receive Descriptor tail pointer register) to inform the DMA of the available Descriptors.

7 Program the parameter settings for the following registers, such as the maximum burst length for DMA initiation (PBL), Descriptor skip length, OSP for Tx DMA, RBSZ for Rx DMA, etc.:

- ETH DMA CH0 Control register (ETH_DMACH0CTRL)
- ETH DMA CH0 Tx Control register (ETH_DMACH0TXCTRL)
- ETH DMA CH0 Rx Control register (ETH_DMACH0RXCTRL)

8 Enable interrupts by programming the ETH DMA Channel 0 interrupt enable register (ETH_DMACH0INTEN).

9 Start the receive and transmit DMA by setting the SR (bit 0) in the DMA Channel 0 receive control register and the ST (bit 0) in the DMA Channel 0 transmit control register.

35.5.14.2 Initializing MTL

The Transaction Layer (MTL) registers must be initialized to establish the transmit and receive operating modes and commands.

Complete the following steps to initialize the MTL registers:

1. Program the following fields to initialize the operation mode in the MTL transmit queue operation mode register.

   a) Transmit Store and Forward (TSF) or Transmit Threshold Control (TTC) (when using threshold mode)

2. Program the following fields to initialize the operation mode in the MTL receive queue operation mode register:

   a) Receive Store and Forward (RSF) or Receive Threshold Control (RTC) (when using threshold mode)

   b) Enable forwarding of error packets and undersized good packets (FEP and FUP)

35.5.14.3 Initializing MAC

The following MAC Initialization operations can be performed after DMA initialization. If the MAC initialization is completed before the DMA is configured, enable the MAC receiver (last step in the following sequence) only after the DMA is active. Otherwise, received frames fill the Rx FIFO and overflow.

1. Provide the MAC address registers: MAC Address0 High register and MAC Address0 Low register. And the other 3 additional MAC addresses are properly programmed.
2. Program the following fields to set the appropriate filters for the incoming frames in the MAC Packet Filter register:
   - Receive All
   - Promiscuous mode
   - Hash or Perfect Filter
   - Unicast, multicast, broadcast, and control frames filter settings
3. Program the following fields for proper flow control in the MAC Q0 Tx Flow Ctrl register:
   - Pause time and other Pause frame control bits
   - Transmit Flow control bits
   - Flow Control Busy
4. Program the MAC interrupt enable register as needed (if applicable) to suit the user's configuration.
5. Program the corresponding fields in the MAC configuration register. For example, the packet gap during transmission and disabling jabber.
6. Set the 0th and 1st bits in the MAC configuration register to start the MAC transmitter and receiver.

To support Jumbo Transmit/Receive packets, follow these steps:

In the MAC configuration register:

a) Set the JE field to 1

b) Set the JD and WD fields to 0

c) Avoid reporting jumbo frame errors

d) Set the GPSLCE field to 1

e) Set the GPSL field in the MAC Extended Configuration Register to a value greater than 9026.

To support Transmit/Receive packets, up to 16K, follow these steps:

In the MAC configuration register:

a) Set the JD and WD fields to 1

b) Avoid reporting jumbo frame errors

c) Set the GPSLCE field to 1

d) Set the GPSL field in the MAC Extended Configuration Register to 16383

35.5.14.4 Performing Normal Receive and Transmit Operation

During normal operation of the ETH, normal and transmit interrupts are read, descriptors polled, the DMA is suspended (if it does not own descriptors), and values of current host transmitter or receiver descriptor pointers are read for debugging.

For normal operation, complete the following steps:

1. For normal transmit and receive interrupts, read the interrupt status. Then, poll the descriptors, reading the status of the descriptor owned by the Host (either transmit or receive).
2. Set appropriate values for the descriptors, ensuring that transmit and receive descriptors are owned by the DMA to resume the transmission and reception of data.
3. If there are no Descriptors for DMA (or no available Descriptors), the DMA will enter a paused state. To resume transmission or reception, release the Descriptors and write the Descriptor tail pointer to the Tx/Rx Descriptor tail pointer register.
4. During the debugging process, you can read the current host transmitter or receiver Descriptor address pointer values (DMA Channel 0 current application transmit Descriptor register and ETH DMA Channel 0 current application receive Descriptor register).
5. During the debugging process, you can read the values of the current host transmit buffer address pointer and receive buffer address pointer (DMA Channel 0 current application transmit buffer register and ETH DMA Channel 0 current application receive buffer register).

35.5.14.5 Stopping and Starting Transmission

To pause transmission for a period of time, follow these steps:

1 Disable the transmit DMA (if applicable) by clearing the 0th bit (ST) in the DMA Channel 0 transmit control register.

2 Wait for the previous frame transmission to complete. You can check this by reading the corresponding bits in the MTL transmit queue debug register (TRCSTS not equal to 01 and TXQSTS = 0).

3 Disable the MAC transmitter and MAC receiver by clearing the RE and TE bits in the MAC configuration register.

4 Disable the receive DMA (if applicable) after ensuring that the data in the Rx FIFO has been transferred to the system memory (by checking the corresponding bits in the MTL receive queue debug register, PRXQ = 0 and RXQSTS = 00).

5 Ensure that the Tx and Rx queues are empty (TXQSTS is 0 in the MTL transmit queue debug register, and RXQSTS is 0 in the MTL receive queue debug register).

6 To restart operations, first start the DMA, then enable the MAC transmitter and receiver.

Note: Do not change the configuration (such as duplex mode, speed, port, or loop back) when the MAC is actively transmitting or receiving. These parameters are changed by software only when the MAC transmitter and receiver are not active. Similarly, do not change the DMA-related configuration when Transmit and Receive DMA are active.

35.5.14.6 Programming Guidelines for Switching to New Descriptor List in Rx DMA

Switching to a new descriptor list is different in the Rx DMA compared to the Tx DMA. Switching to a new descriptor list is permitted when the Rx DMA is in SUSPEND state, as clarified by the following points:

- Generally, Rx DMA prepares the descriptors in advance.
- If the Rx DMA goes to SUSPEND due to descriptors not being available, a major failure occurs (software is not able to free the filled-up descriptors/buffers). If this issue is not rectified immediately, frames are lost because of an Rx FIFO Overflow. Therefore, the software is allowed to create a new descriptor list and program the Rx DMA to start using it immediately, without going into STOP state.

35.5.14.7 Programming Guidelines for Switching the AHB Clock Frequency

To dynamically change the AHB clock frequency without applying a software reset or hard reset, follow these steps:

1 Disable the transmit DMA (if applicable) and wait for the previous frame transmission to complete. Once the frame transmission is complete, the Tx FIFO becomes empty, and the Tx DMA enters the STOP state. The status of the Tx FIFO is indicated in the MTL transmit queue debug register, and the DMA status is indicated in the DMA debug status register.

2 Clear the corresponding bits in the MAC configuration register to disable the MAC transmitter and MAC receiver.

3 Disable the receive DMA (if applicable) after ensuring that the data in the Rx FIFO has been transferred to the system memory. The Rx FIFO empty status is provided in the MTL receive queue debug register.

4 Ensure that the application does not perform any register read/write operations.

5 Change the AHB clock frequency.

6 Enable the MAC transmitter or MAC receiver along with the transmit or receive DMA. These steps ensure that there is no valid data in the Tx FIFO or Rx FIFO during the clock frequency switch, preventing data corruption.

35.5.14.8 PHY Interface Link State Transition

When the link is disconnected, the transmit and receive clocks remain active

When the link is disconnected but the transmit and receive clocks are running, follow these steps:

1 Disable the transmit DMA (if applicable) by clearing the 0th bit (ST) in the DMA Channel 0 transmit control register.

2 Disable the MAC receiver by clearing the 2nd bit (RE) in the MAC configuration register.

3 Wait for the previous frame transmission to complete. You can check this by reading the corresponding bits in the MTL transmit queue debug register (TRCSTS is not 01).

4 Disable the MAC transmitter by clearing the 1st bit (TE) in the MAC configuration register.

5 Ensure that the Tx and Rx queues are empty (TXQSTS is 0 in the MTL transmit queue debug register, and RXQSTS is 0 in the MTL receive queue debug register).

6 Upon link establishment, read the PHY registers to understand the latest configuration and program the MAC registers accordingly.

7 If a restart is required, start the Tx DMA first, then enable the MAC transmitter and receiver. There is no need to disable the Rx DMA. Since the receiver is disabled, no data can be fetched from the Rx FIFO.

When the link is disconnected, the transmit and receive clocks are in a stopped state

When the link is disconnected and the transmit and receive clocks are stopped, follow these steps:

1 Disable the MAC transmitter and receiver by clearing the RE and TE bits in the MAC configuration register. Since the clocks are not present, this action will not take immediate effect.

2 Wait for the link to be restored and the clocks to resume.

3 While the transmit/receive clocks are stopped, wait for any partial frames (if any) to complete transmission. This can be checked by reading the MAC debug registers (should all be zeros). Some old packets may still remain in the Tx FIFO when the MAC transmitter is stopped.

4 Read the PHY registers to understand the current operating mode and program the MAC registers accordingly.

5 Restart the MAC transmitter and receiver by setting the RE and TE bits.

35.5.14.9 Programming Guidelines for IEEE 1588 Timestamp

To enable the timestamp feature, you can set the 0th bit of the MAC timestamp control register. However, after setting this bit, the timestamp counter must be initialized. During the initialization process, follow these steps:

1 Clear the 12th bit of the MAC interrupt enable register to mask the timestamp trigger interrupt.

2 Set the 0th bit of the MAC timestamp control register to enable the timestamp.

3 Program the MAC sub-second increment register based on the PTP clock frequency.

4 If using the fine calibration method, program the MAC timestamp addend register and set the 5th bit of the MAC timestamp control register.

5 Poll the MAC timestamp control register until the 5th bit is cleared.

6 Program the 1st bit of the MAC timestamp control register to select the fine update method (if needed).

7 Program the MAC system time seconds update register and MAC system time nanoseconds update register with the appropriate time values.

8 Set the 2nd bit in the MAC timestamp control register. The timestamp counter will start running immediately after initialization with the value written to the timestamp update register. If enabling a one-step timestamp:

   a) To enable a one-step timestamp, program the 27th bit of the TDES3 context Descriptor for a one-step timestamp.

9 Enable the MAC receiver and transmitter to obtain the correct timestamp.

Note: If timestamp operation is disabled by clearing Bit 0 of MAC Timestamp Control register, repeat all these steps to restart the timestamp operation.

35.5.14.10 IEEE 1588 Timestamp - Coarse Calibration of System Time

To synchronize or update the system time in a single process (coarse calibration method), follow these steps:

1 Set the offset (positive or negative) in the timestamp update registers (MAC system time seconds update register and MAC system time nanoseconds update register).

2 Set the 3rd bit of the MAC timestamp control register (TSUPDT).

After clearing the TSUPDT bit, the value in the timestamp update registers will be added to or subtracted from the system time.

35.5.14.11 IEEE 1588 Timestamp - Fine Calibration of System Time

To synchronize or update the system time to reduce system time jitter (fine calibration method), follow these steps:

1 Calculate the rate at which you want the system time increment to slow down or speed up, following the algorithm described in the system time register module section.

2 Update the MAC timestamp addend register with the new value and set the 5th bit of the MAC timestamp control register.

3 Wait for the new value in the addend register to take effect. You can enable the timestamp trigger interrupt once the system time reaches the target value.

4 Set the desired target time in the MAC PPS target time seconds register and MAC PPS target time nanoseconds register.

5 Enable the timestamp interrupt by setting the 12th bit in the MAC interrupt enable register.

6 When the trigger generates an interrupt, read the MAC interrupt status register.

7 Reprogram the MAC timestamp addend register with the old value and set the 5th bit again.

35.5.14.12 Programming Guidelines for VLAN Filtering on Receive

To set up VLAN filtering during reception, follow these steps:

1 Program the following bits of the MAC VLAN Tag register to select the filtering method:

- ETV: Enable 12-bit VLAN tag comparison or 16-bit VLAN tag comparison
- VTHM: Enable VLAN tag hash table matching
- ERSVLM: Enable matching for received S-VLAN or C-VLAN (to enable S-VLAN processing, set ESVL)
- DOVLTC: Ignore VLAN type in tag matching
- VTIM: Enable VLAN tag inverse matching instead of normal VLAN tag matching

2 Program the VL field of the MAC VLAN Tag register for 12-bit or 16-bit VLAN tag pairs.

3 If VLAN tag hash filtering is enabled, program the MAC VLAN Hash Table register. When the ETV bit is reset, the high 4 bits of the calculated CRC-32 for the VLAN tag will be inverted and used to index the contents of the MAC VLAN Hash Table register. When the ETV bit is set to 1, the high 4 bits of the calculated CRC-32 for the VLAN tag will be used to index the contents of the MAC VLAN Hash Table register. For example, when the ETV bit is set, the hash value 4b'1000 selects the 8th entry of the VLAN hash table. When the ETV bit is reset, the hash value 4'b'1000 selects the 7th entry of the VLAN hash table.

35.6 ETH Register

ETH consists of the following registers:

- MAC registers, which include MMC-related registers (refer to section 35.6.1)
- MTL registers (refer to section 35.6.2)
- DMA registers (refer to section 35.6.3)

35.6.1 ETH MAC Registers

35.6.1.1 ETH MAC Configuration register (ETH_MACCFG)

Address offset: 0x0000

Reset value: 0x0000 8000

The MAC Configuration Register establishes the operating mode of the MAC.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				CSO	IPG			GPSLCE	S2KP	CST	ACS	WD	Reserved	JD	JE
				rw	rw			rw	rw	rw	rw	rw		rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PS	FES	DM	LM	ECRSFD	DO	DCRS	DR	Reserved	BL		DC	PRELEN		TE	RE
r	rw	rw	rw	rw	rw	rw	rw		rw		rw	rw		rw	rw

Bit Field	Name	Description
31:28	Reserved	Reserved, must be kept at reset value.
27	CSO	Checksum Offload. When set, this bit enables the IPv4 header checksum checking and IPv4 or IPv6 TCP, UDP, or ICMP payload checksum checking. When this bit is reset, the COE function in the receiver is disabled. The Layer 3 and Layer 4 Packet Filter and Enable Split Header features automatically selects the IPC Full Checksum Offload Engine on the Receive side. When any of these features are enabled, you must set the IPC bit. 0: Disable IP header/payload checksum validation feature 1: Enable IP header/payload checksum validation feature
26:24	IPG[2:0]	Inter-Packet Gap. These bits control the minimum gap between packets during transmission. This minimum packet gap range is effective in full-duplex mode. In half-duplex mode, the packet gap can only be configured as 64 bits (IPG = 100). When sending in JAM mode due to the activation of backpressure, the MAC does not consider the minimum IPG. The above function (IPG less than 96 bits) is only valid when the EIPGEN bit in ETH_MACEXTCFG register is reset. When EIPGEN is set to 1, the minimum IPG will be controlled as per the description provided in the EIPG field of the

Bit Field	Name	Description
		ETH_MACEXTCFG register (greater than 96 bits). 000: 96 bit times 001: 88 bit times 010: 80 bit times ... 111: 40 bit times
23	GPSLCE	Giant Packet Size Limit Control Enable. When this bit is set to 1, the MAC considers the value in the GPSL field of the ETH_MACEXTCFG register to declare received packets as giant packets. This field must be programmed to a value greater than 1518 bytes. Otherwise, the MAC will consider 1518 bytes as the giant packet limit. When this bit is reset, if the size of the received packet is greater than 1518 bytes (1522 bytes for tagged packets), the MAC will consider it as a giant packet. The watchdog timeout limit, giant packet enable, and 2K Packet Enable have higher precedence over this bit. That is the MAC considers a received packet as Giant packet when its size is greater than 9,018 bytes (9,022 bytes for tagged packet) with Jumbo Packet Enabled and greater than 2,000 bytes with 2K Packet Enabled. The watchdog timeout, if enabled, will terminate the received packet upon reaching the watchdog limit. Therefore, to obtain the giant packet status, the programmed giant packet limit should be less than the watchdog limit. 0: Disable giant packet size limit control function 1: Enable giant packet size limit control function
22	S2KP	IEEE 802.3as Support for 2K Packets. When this bit is set, the MAC considers all packets with up to 2,000 bytes length as normal packets. When the JE bit is not set, the MAC considers all received packets of size more than 2K bytes as Giant packets. When this bit is reset and the JE bit is not set, the MAC considers all received packets of size more than 1,518 bytes (1,522 bytes for tagged) as giant packets. For detailed information on how the setting of this bit and the JE bit affects the giant packet status, refer to Table 35-56. 0: Disable support for up to 2K packets 1: Enable support for up to 2K packets
21	CST	CRC Stripping for Type Packets. When this bit is set to 1, the last four bytes (FCS) of all EtherType packets (with a type field greater than 1536) will be stripped and dropped before forwarding the packets to the application. 0: Disable CRC stripping for Type packets 1: Enable CRC stripping for Type packets
20	ACS	Automatic Pad or CRC Stripping. When this bit is set, the MAC strips the Pad or FCS field on the incoming packets only if the value of the length field is less than 1,536 bytes. All received packets with length field greater than or equal to 1,536 bytes are passed to the application without stripping the Pad or FCS field. When this bit is reset, the MAC passes all incoming packets to the application, without any modification.

Bit Field	Name	Description
		0: Disable Automatic Pad or CRC Stripping 1: Enable Automatic Pad or CRC Stripping
19	WD	Disable Watchdog. When this bit is set to 1, the MAC disables the watchdog timer on the receiver. The MAC can receive packets of up to 16383 bytes in length. When this bit is reset, the MAC does not allow the reception of packets exceeding 2048 bytes (or 10240 bytes if JE is set high). The MAC will discard all bytes received after 2048 bytes. 0: Enable Watchdog function 1: Disable Watchdog function
18	Reserved	Reserved, must be kept at reset value.
17	JD	Disable Jabber. When this bit is set to 1, the MAC disables the jabber timer on the transmitter. The MAC can transmit packets of up to 16383 bytes in length. When this bit is reset, if the application sends data exceeding 2048 bytes during transmission (or 10240 bytes if JE is set high), the MAC will not transmit the remaining bytes of that packet. 0: Enable Jabber function 1: Disable Jabber function
16	JE	Jumbo Packet Enable. When this bit is set to 1, the MAC allows jumbo packets of 9018 bytes (9022 bytes for packets with VLAN tags) and does not report giant packet errors in the Rx packet status. 0: Disable Jumbo Packet function 1: Enable Jumbo Packet function
15	PS	Port Select. This bit selects the Ethernet rate. Together with the 14th bit, it chooses the exact line rate. In configurations where only 10/100Mbps (always 1) or only 1000Mbps (always 0) are used, this bit is read-only and has the appropriate value. 0: Used for 1000Mbps operation 1: Used for 10Mbps or 100Mbps operation *Note: This bit is read-only and always reads as 1.*
14	FES	Speed. This bit selects the speed mode. 0: When the PS bit is 1, the speed is 10Mbps; when the PS bit is 0, the speed is 1000bps 1: When the PS bit is 1, the speed is 100Mbps *Note: This bit is shared with the PS bit only in ETH1.*
13	DM	Duplex Mode. When this bit is set to 1, the MAC operates in full-duplex mode, allowing it to send and receive simultaneously. 0: Half-duplex mode 1: Full-duplex mode
12	LM	Loopback Mode.

Bit Field	Name	Description
		When this bit is set to 1, the MAC operates in loopback mode under MII. Loopback operation requires MII Rx clock input to function properly. This is because the Tx clock does not loop back internally. 0: Disable loopback function 1: Enable loopback function
11	ECRSFD	Enable Carrier Sense Before Transmission in Full-Duplex Mode. When this bit is set to 1, the MAC transmitter checks the CRS signal before sending data packets in full-duplex mode. The MAC will only start transmission when the CRS signal is low. When this bit is reset, the MAC transmitter will ignore the status of the CRS signal. 0: Disable carrier sense before transmission in full-duplex mode 1: Enable carrier sense before transmission in full-duplex mode
10	DO	Disable Receive Own. When this bit is set to 1, the MAC will disable receiving packets in half-duplex mode when TX_EN is active. When this bit is reset, the MAC will receive all packets provided by the PHY. This bit does not apply to full-duplex mode. 0: Enable Receive Own function 1: Disable Receive Own function
9	DCRS	Disable Carrier Sense During Transmission. When this bit is set to 1, the MAC transmitter will ignore the MII CRS signal during packet transmission in half-duplex mode. This prevents errors from being generated due to carrier loss or no carrier during transmission. When this bit is reset, the MAC transmitter will generate errors due to carrier sense during transmission. The MAC may even abort the transmission. 0: Enable carrier sense during transmission 1: Disable carrier sense during transmission
8	DR	Disable Retry. When this bit is set to 1, the MAC will attempt to send the packet only once. In the event of a collision on the MII interface, the MAC will ignore the current packet transmission, report "packet abort" and "excessive collision error" in the Tx packet status. When this bit is reset, the MAC will retry based on the setting of the BL field. This bit applies only to half-duplex mode. 0: Enable retry function 1: Disable retry function
7	Reserved	Reserved, must be kept at reset value.
6:5	BL[1:0]	Back-Off Limit. The back-off limit determines the random integer delay in slot times (4096 bit times for 1000Mbps; 512 bit times for 10/100Mbps) that the MAC waits before rescheduling a transmission attempt after a collision during retries. Here, n = number of retransmission attempts. The random integer r ranges from 0 ≤ r < 2^k. This bit applies only to half-duplex mode.

Bit Field	Name	Description
		00: k = min(n, 10) 01: k = min(n, 8) 10: k = min(n, 4) 11: k = min(n, 1)
4	DC	Deferral Check. When this bit is set to 1, the MAC enables the deferral check function. When the Tx state machine delays exceed 24288 bit times in 10/100Mbps mode, the MAC will issue a packet abort status and set the excessive deferral error bit in the Tx packet status. The delay begins when the transmitter is ready to transmit but is blocked due to the presence of a carrier sense signal (CRS) on the MII. The delay time is non-cumulative. For example, if the transmitter delays for 10000 bit times due to a valid CRS signal and then the CRS signal becomes invalid, causing a collision when the transmitter attempts to transmit. Following the collision, the transmitter needs to back off, and after completing the back-off, it will defer again. In this scenario, the deferral timer will reset to 0 and restart. When this bit is reset, the deferral check function is disabled, and the MAC will defer until the CRS signal becomes invalid. This bit applies only to half-duplex mode. 0: Disable deferral check function 1: Enable deferral check function
3:2	PRELEN[1:0]	Preamble Length for Transmit Packets. These bits control the number of preamble bytes added to the beginning of each Tx packet. The preamble is reduced only when the MAC operates in full-duplex mode. 00: 7-byte preamble 01: 5-byte preamble 10: 3-byte preamble 11: Reserved
1	TE	Transmitter Enable. When this bit is set to 1, it enables the MAC transmit state machine for transmission on the MII interface. When this bit is reset, the MAC transmit state machine is disabled after completing the transmission of the current data packet. The transmit state machine will no longer transmit any data packets. 0: Disable the transmit state machine 1: Enable the transmit state machine
0	RE	Receiver Enable. When this bit is set to 1, it enables the MAC receive state machine to receive data packets from the MII interface. When this bit is reset, the MAC receive state machine is disabled after completing the reception of the current data packet. The receive state machine will no longer receive any data packets from the MII interface. 0: Disables the receive state machine 1: Enables the receive state machine

Table 35-56 Gigantic Packet Status Based on S2KP and JE Bits

Length/Type Field	Received Packet Length	S2KP	JE	Giant Packet Status
Untagged data packet	＞1518	0	0	1
	＞2000	1	0	1
	＞9018	x	1	1
VLAN tagged data packet	＞1522	0	0	1
	＞2000	1	0	1
	＞9022	x	1	1

35.6.1.2 ETH MAC Extended Configuration register (ETH_MACEXTCFG)

Address offset: 0x0004

Reset value: 0x0000 0000

The MAC Extended Configuration Register establishes the operating mode of the MAC.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		EIPG					EIPGEN	Reserved					USP	SPEN	DCRCC
		rw					rw						rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		GPSL													
		rw													

Bit Field	Name	Description
31:30	Reserved	Reserved, must be kept at reset value.
29:25	EIPG[4:0]	Extended Inter-Packet Gap (EIPG). When the EIPGEN bit is set to 1, the value in this field is valid. This field (as the most significant bit) along with the IPG field in ETH_MACCFG together provide the minimum IPG (greater than 96 bit times, in steps of 8 bit times): {EIPG, IPG} 0x00: 104 bit times 0x01: 112 bit times 0x02: 120 bit times ... 0xFF: 2144 bit times
24	EIPGEN	Extended Inter-Packet Gap Enable. When this bit is set to 1, the MAC interprets the EIPG field and the IPG field in ETH_MACCFG together as the minimum IPG (greater than 96 bit times, in steps of 8 bit times). When this bit is reset, the MAC ignores the EIPG field and interprets the IPG field in ETH_MACCFG as the minimum IPG (less than or equal to 96 bit times, in steps of 8 bit times). 0: Disable the extended inter-packet gap feature

Bit Field	Name	Description
		1: Enable the extended inter-packet gap feature
23:19	Reserved	Reserved, must be kept at reset value.
18	USP	Unicast Slow Protocol Packet Detect. When this bit is set to 1, the MAC will detect slow protocol data packets with unicast addresses specified by the MAC Address 0 High Register and MAC Address 0 Low Register. The MAC will also detect slow protocol data packets with the slow protocol multicast address (01-80-C2-00-00-02). When this bit is reset, the MAC will only detect slow protocol data packets with the slow protocol multicast address specified in Section 5 of IEEE 802.3-2015. 0: Disable unicast slow protocol packet detection feature 1: Enable unicast slow protocol packet detection feature
17	SPEN	Slow Protocol Detection Enable. When this bit is set to 1, the MAC will process slow protocol data packets (EtherType: 0x8809) and provide Rx status. The MAC will discard slow protocol data packets with invalid subtypes. When this bit is reset, the MAC will forward all error-free slow protocol data packets to the application. The MAC will treat these packets as normal type packets. 0: Disable slow protocol detection feature 1: Enable slow protocol detection feature
16	DCRCC	Disable CRC Checking for Received Packets. When this bit is set to 1, the MAC receiver will not check the CRC field in received packets. When this bit is reset, the MAC receiver will always check the CRC field in received packets. 0: Enable CRC checking feature 1: Disable CRC checking feature
15:14	Reserved	Reserved, must be kept at reset value.
13:0	GPSL	Disable CRC Checking for Received Packets. When this bit is set to 1, the MAC receiver will not check the CRC field in received packets. When this bit is reset, the MAC receiver will always check the CRC field in received packets. 0: Enable CRC checking feature 1: Disable CRC checking feature

35.6.1.3 ETH MAC Packet Filter register (ETH_MACPFLT)

Address offset: 0x0008

Reset value: 0x0000 0000

The MAC Packet Filter register contains the filter controls for receiving packets. Some of the controls from this register go to the address check block of the MAC which performs the first level of address filtering. The second level of filtering is performed on the incoming packet based on other controls such as Pass Bad Packets and Pass Control Packets.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

RA	Reserved	VTFE
rw		rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved					HPF	SAF	SAIF	PCP		DBP	PAM	DAIF	HMC	HUC	PM
					rw	rw	rw	rw		rw	rw	rw	rw	rw	rw

Bit Field	Name	Description
31	RA	Receive All. When this bit is set to 1, the MAC receiver module will forward all received packets to the application, regardless of whether they pass through the address filters. The results of SA or DA filtering are updated in the corresponding bits of the Rx status word (pass or fail). When this bit is reset, the receiver module will only forward packets that pass through the SA or DA address filters to the application. 0: Disable the receive all packets feature 1: Enable the receive all packets feature
30:17	Reserved	Reserved, must be kept at reset value.
16	VTFE	VLAN Tag Filter Enable. When this bit is set to 1, the MAC will discard VLAN-tagged packets that do not match the VLAN tag filter. When this bit is reset, the MAC will forward all packets regardless of the VLAN tag matching status. 0: Disable VLAN tag filter 1: Enable VLAN tag filter
15:11	Reserved	Reserved, must be kept at reset value.
10	HPF	Hash or Perfect Filter. When this bit is set to 1, if a packet matches the perfect filter or hash filter (set by the HMC or HUC bits), the packet will pass through the address filter. When this bit is reset and either the HUC or HMC bit is set to 1, the packet will only pass through the filter if it matches the hash filter. 0: Disable hash or perfect filter 1: Enable hash or perfect filter
9	SAF	Source Address Filter Enable. When this bit is set to 1, the MAC will compare the SA field of received packets with the programmed values in the enabled SA registers. If the comparison result is not the same, the MAC will discard the packet. When this bit is reset, the MAC will forward the received packets to the application and update the SAF bit of the Rx status based on the SA address comparison result. 0: Disable the source address filter 1: Enable the source address filter *Note: According to the IEEE specification, the 47th bit of the SA is a reserved bit. However, in this IP, the MAC will compare all 48 bits. Software drivers should take this into consideration when programming MAC address registers for SA.*

Bit Field	Name	Description
8	SAIF	SA Inverse Filtering. When this bit is set to 1, the address checking module performs SA address comparison in inverse filtering mode. If the SA of the packet matches the programmed value in the SA registers, it will be marked as SA address filtering failure. When this bit is reset, if the SA of the packet does not match the programmed value in the SA registers, it will be marked as SA address filtering failure. 0: Disable SA inverse filtering 1: Enable SA inverse filtering
7:6	PCP[1:0]	Pass Control Packets. These bits control the forwarding of all control packets (including unicast and multicast pause packets). 00: MAC filters out all control packets, preventing them from reaching the application. 01: MAC forwards all control packets except pause packets to the application, even if these packets do not pass through the address filter. 10: MAC forwards all control packets to the application, even if these packets do not pass through the address filter. 11: MAC forwards control packets that pass through the address filter.
5	DBP	Disable Broadcast Packets. When this bit is set to 1, the address filtering module will block all incoming broadcast packets. Additionally, it will override all other filtering settings. When this bit is reset, the address filtering module will pass all received broadcast packets. 0: Enable the reception of all broadcast packets 1: Disable the reception of all broadcast packets
4	PAM	Pass All Multicast. When this bit is set to 1, it indicates that all received packets with a multicast destination address (where the first bit of the destination address field is "1") should be passed. When this bit is reset, the filtering of multicast packets depends on the HMC bit. 0: Disable the function of passing all multicast packets 1: Enable the function of passing all multicast packets
3	DAIF	DA Inverse Filtering. When this bit is set to 1, the address checking module performs DA address comparison in inverse filtering mode for both unicast and multicast packets. When this bit is reset, normal packet filtering is performed. 0: Disable DA inverse filtering feature 1: Enable DA inverse filtering feature
2	HMC	Hash Multicast. When this bit is set to 1, the MAC performs destination address filtering on received multicast packets based on a hash table. When this bit is reset, the MAC performs perfect destination address filtering on multicast packets, meaning it compares the DA field with the programmed values in the DA registers. 0: Disable hash multicast filtering feature

Bit Field	Name	Description
		1: Enable hash multicast filtering feature
1	HUC	Hash Unicast. When this bit is set to 1, the MAC performs destination address filtering on received unicast packets based on a hash table. When this bit is reset, the MAC performs perfect destination address filtering on unicast packets, meaning it compares the DA field with the programmed values in the DA registers. 0: Disable hash unicast filtering feature 1: Enable hash unicast filtering feature
0	PM	Promiscuous Mode. When this bit is set to 1, the address filtering module will pass through all incoming packets regardless of their destination or source addresses. When PM is set, the SA or DA filter fail status bit of the Rx status word will always be cleared. 0: Disable Promiscuous Mode 1: Enable Promiscuous Mode

35.6.1.4 ETH MAC Watchdog Timeout register (ETH_MACWDGTO)

Address offset: 0x000C

Reset value: 0x0000 0000

The Watchdog Timeout register controls the watchdog timeout for received packets.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved							PWE	Reserved				WTO			
							rw					rw			

Bit Field	Name	Description
31:9	Reserved	Reserved, must be kept at reset value.
8	PWE	Programmable Watchdog Enable. When this bit is set to 1 and the WD bit of the ETH_MACCFG register is reset, the WTO field is used as the watchdog timeout for received data packets. When this bit is reset, the watchdog timeout for received data packets is controlled by setting the WD and JE bits of the ETH_MACCFG register. 0: Disable Programmable Watchdog feature 1: Enable Programmable Watchdog feature
7:4	Reserved	Reserved, must be kept at reset value.
3:0	WTO	Watchdog Timeout. When the PWE bit is set to 1 and the WD bit of the ETH_MACCFG register is reset, this field is used as the watchdog timeout for received data packets. If the length of a

Bit Field	Name	Description
		received data packet exceeds the value in this field, the packet will be terminated and declared as an error packet. 0x0: 2Kbytes 0x1: 3Kbytes 0x2: 4Kbytes 0x3: 5Kbytes 0x4: 6Kbytes 0x5: 7Kbytes ... 0xD: 15Kbytes 0xE: 16383 bytes 0xF: Reserved *Note: When setting the PWE bit, the value in this field should be greater than 1522. Otherwise, packets that are valid according to the IEEE 802.3 standard will be declared as error packets and discarded.*

35.6.1.5 ETH MAC Hash Table register 0 (ETH_MACHASHTR0)

Address offset: 0x0010

Reset value: 0x0000 0000

The Hash Table Register 0 contains the first 32 bits of the hash table.

The Hash table is used for group address filtering. For hash filtering, the content of the destination address in the incoming packet is passed through the CRC logic and the upper six bits of the CRC register are used to index the content of the Hash table.

The hash value of the destination address is calculated in the following way:

1 Calculate the 32-bit CRC for the DA (See IEEE 802.3, Section 3.2.8 for the steps to calculate CRC32).

2 Perform bitwise reversal for the value obtained in Step 1.

3 Take the upper 6 (or 7 or 8) bits from the value obtained in Step 2.

If the corresponding bit value of the register is 1'b1, the packet is accepted. Otherwise, it is rejected. If the PM bit is set in MAC Packet Filter, all multicast packets are accepted regardless of the multicast hash values.

If the Hash Table register is configured to be double-synchronized to the MII clock domain, the synchronization is triggered only when Bits [31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the Hash Table Register X registers are written.

If double-synchronization is enabled, consecutive writes to this register should be performed after at least four clock cycles in the destination clock domain.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
HT31T0[31:16]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HT31T0[15:0]															
rw															

Bit Field	Name	Description
31:0	HT31T0	MAC Hash Table First 32 Bits. This field contains the first 32 Bits [31:0] of the Hash table.

35.6.1.6 ETH MAC Hash Table register 1 (ETH_MACHASHTR1)

Address offset: 0x0014

Reset value: 0x0000 0000

The Hash Table Register 1 contains the second 32 bits of the hash table.

The Hash table is used for group address filtering. For hash filtering, the content of the destination address in the incoming packet is passed through the CRC logic and the upper six bits of the CRC register are used to index the content of the Hash table.

The hash value of the destination address is calculated in the following way:

1 Calculate the 32-bit CRC for the DA (See IEEE 802.3, Section 3.2.8 for the steps to calculate CRC32).

2 Perform bitwise reversal for the value obtained in Step 1.

3 Take the upper 6 (or 7 or 8) bits from the value obtained in Step 2.

If the corresponding bit value of the register is 1'b1, the packet is accepted. Otherwise, it is rejected. If the PM bit is set in MAC Packet Filter, all multicast packets are accepted regardless of the multicast hash values.

If the Hash Table register is configured to be double-synchronized to the MII clock domain, the synchronization is triggered only when Bits [31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the Hash Table Register X registers are written.

If double-synchronization is enabled, consecutive writes to this register should be performed after at least four clock cycles in the destination clock domain.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
HT63T32[31:16]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HT63T32[15:0]															
rw															

Bit Field	Name	Description
31:0	HT63T32	MAC Hash Table Second 32 Bits.

Bit Field	Name	Description
		This field contains the second 32 Bits [63:32] of the Hash table.

35.6.1.7 ETH VLAN Tag register (ETH_MACVLANTAG)

Address offset: 0x0050

Reset value: 0x0000 0000

The VLAN tag register is used to identify IEEE 802.1Q VLAN packets.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved						VTHM	EVLRXS	Reserved	EVLS		DOVLTC	ERSVLM	ESVL	VTIM	ETV
						rw	rw		rw		rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VL[15:0]															
rw															

Bit Field	Name	Description
31:26	Reserved	Reserved, must be kept at reset value.
25	VTHM	VLAN Tag Hash Table Match Enable. When this bit is set to 1, the four most significant bits of the CRC of the VLAN tag are used to index the contents of the ETH_MACVHASHT register. A value of 1 in the VLAN hash table register corresponding to the index indicates a match between the packet and the VLAN hash table. When the ETV bit is set to 1, a comparison is made using the CRC of the 12-bit VLAN Identifier (VID). When the ETV bit is reset, the comparison is made using the CRC of the 16-bit VLAN tag. When this bit is reset, the VLAN hash matching operation is not performed. 0: Disable VLAN Tag Hash Table Match feature 1: Enable VLAN Tag Hash Table Match feature
24	EVLRXS	Enable VLAN Tag in Rx status. When this bit is set to 1, the MAC will provide the external VLAN tag in the Rx status. When this bit is reset, the MAC will not provide the external VLAN tag in the Rx status. 0: Disable VLAN tag in Rx status 1: Enable VLAN tag in Rx status
23	Reserved	Reserved, must be kept at reset value.
22:21	EVLS	Enable VLAN Tag Stripping on Receive. This field indicates the stripping operation performed on external VLAN tags in received packets. 00: No stripping 01: Strip if passed through VLAN filter 10: Strip if not passed through VLAN filter 11: Always perform stripping operation

Bit Field	Name	Description
20	DOVLTC	Disable VLAN Type Check. When this bit is set to 1, the MAC does not check whether the VLAN tag is an S-VLAN type or C-VLAN type. When this bit is reset, the MAC will only filter or match VLAN tags if the VLAN tag type is similar to that specified by the ERSVLM bit. If the VLAN type of the received packet does not match the VLAN type programmed in the VLAN filter, it will bypass the VLAN filter. 0: Enable VLAN Type Check feature 1: Disable VLAN Type Check feature
19	ERSVLM	Enable Receive S-VLAN Match. When this bit is set to 1, the MAC receiver enables filtering or matching for S-VLAN (type = 0x88A8) packets. When this bit is reset, the MAC receiver enables filtering or matching for C-VLAN (type = 0x8100) packets. 0: Disable Receive S-VLAN Match feature 1: Enable Receive S-VLAN Match feature
18	ESVL	Enable S-VLAN. When this bit is set to 1, the MAC transmitter and receiver will treat S-VLAN packets (type = 0x88A8) as valid packets with VLAN tags. 0: Disable S-VLAN 1: Enable S-VLAN
17	VTIM	VLAN Tag Inverse Match Enable. When this bit is set to 1, VLAN tag inverse matching is enabled. Packets without a matching VLAN tag are marked as a match. When this bit is reset, VLAN tag perfect matching is enabled. Packets with a matching VLAN tag are marked as a match. 0: Disable VLAN Tag Inverse Match feature 1: Enable VLAN Tag Inverse Match feature
16	ETV	Enable 12-Bit VLAN Tag Comparison. When this bit is set to 1, comparison and filtering are done using the 12-bit VLAN Identifier instead of the full 16-bit VLAN tag. The bits [11:0] of the VLAN tag are compared with the corresponding field in the received VLAN-tagged data packet. Similarly, when enabled, only the 12 bits of the VLAN tag in the received packet are used for hash-based VLAN filtering. When this bit is reset, all 16 bits of the 15th and 16th bytes of the received VLAN data packet are used for comparison and VLAN hash filtering. 0: Disable 12-Bit VLAN Tag Comparison feature 1: Enable 12-Bit VLAN Tag Comparison feature
15:0	VL[15:0]	VLAN Tag Identifier for Receive Packets. This field contains the 802.1Q VLAN tag used to identify VLAN packets. The VLAN tag identifier is compared with the 15th and 16th bytes of the received data packet to identify VLAN packets. The following describes the individual bits of this field: bits[15:13]: User Priority bit12: Canonical Format Indicator (CFI) or Drop Eligible Indicator (DEI) bits[11:0]: VLAN Identifier (VID) field of the VLAN tag When ETV is set to 1, only the VID is used for comparison. If all bits in this field ([11:0] when ETV

Bit Field	Name	Description
		is set to 1) are zero, the MAC will not check the 15th and 16th bytes used for VLAN tag comparison. Instead, all packets with a type field value of 0x8100 or 0x88a8 will be declared as VLAN packets.

35.6.1.8 ETH VLAN Hash Table register (ETH_MACVHASHT)

Address offset: 0x0058

Reset value: 0x0000 0000

When the ERSVLM bit of the ETH_MACVLANTAG register is set to 1, the 16-bit VLAN hash table register is used for group address filtering based on VLAN tags. For hash filtering, the content of the 16-bit VLAN tag or 12-bit VLAN ID (depending on the ETV bit of the ETH_MACVLANTAG register) in the incoming data packet is passed through CRC logic. The high four bits of the calculated CRC are used to index the content of the VLAN hash table. For example, the hash value 4b'1000 is used to select bit 8 of the VLAN hash table.

The hash value for the destination address is calculated as follows:

1 Calculate a 32-bit CRC for the VLAN tag or ID (refer to section 3.2.8 of IEEE 802.3 for CRC32 calculation steps).

2 Perform a bit-wise reversal on the value obtained in step 1.

3 Take the high 4 bits from the value obtained in step 2.

If the VLAN hash table register is configured to be double synchronized with the MII clock domain, synchronization is triggered only when writing to bit [15:8] (little-endian mode) or bit [7:0] (big-endian mode) of the register. If double synchronization is enabled, consecutive writes to this register should occur at least four clock cycles after the target clock domain.

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="16">Reserved</td></tr>
</table>

<table>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="16">VLHT[15:0]</td></tr>
<tr><td colspan="16">rw</td></tr>
</table>

Bit Field	Name	Description
31:16	Reserved	Reserved, must be kept at reset value.
15:0	VLHT	VLAN Hash Table. This field contains the 16-bit VLAN Hash Table.

35.6.1.9 ETH MAC Tx Flow Control register (ETH_MACTXFLWCTRL)

Address offset: 0x0070

Reset value: 0x0000 0000

The Flow Control Register controls the MAC's flow control module to generate and receive control (pause command) packets. When writing to the register and setting the "busy" bit to 1, it triggers the flow control module to generate a pause packet. The fields of the control packet are selected according to the specifications of 802.3x, and the pause time value in this register is used to control the pause time field of the data packet. The "busy" bit remains set before the control packet is transmitted onto the cable. The application must ensure that the "busy" bit is cleared before writing data to the register.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PT[15:0]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								DZPQ	PLT[2:0]			Reserved		TFE	FCB_BPA
								rw	rw					rw	rw

Bit Field	Name	Description
31:16	PT	Pause Time. This field holds the value to be used for the pause time field in the Tx control packets. If the pause time bit is configured to be double synchronized with the MII clock domain, consecutive write operations to this register can only be performed at least four clock cycles after the target clock domain.
15:8	Reserved	Reserved, must be kept at reset value.
7	DZPQ	Disable Zero-Quanta Pause. When this bit is set to 1, it disables the automatic generation of zero-quanta pause packets when the flow control signal from the FIFO layer becomes active. Resetting this bit enables the normal operation of automatic generation of zero-quanta pause packets. 0: Enable the functionality for generating zero-quanta pause packets. 1: Disable the functionality for generating zero-quanta pause packets.
6:4	PLT	Pause Low Threshold. This field configures the threshold of the pause timer, at which the input flow control signal is checked to automatically resend pause packets. This threshold should always be less than the pause time configured in bits [31:16]. For example, if PT = 100H (256 time slots) and PLT = 001, after the first pause data packet is sent, the second pause data packet will automatically be sent when the flow control signal becomes active after 228 (256 - 28) time slots. The following thresholds are provided for different values: 000: Pause time minus 4 time slots (PT - 4 time slots) 001: Pause time minus 28 time slots (PT - 28 time slots) 010: Pause time minus 36 time slots (PT - 36 time slots) 011: Pause time minus 144 time slots (PT - 144 time slots) 100: Pause time minus 256 time slots (PT - 256 time slots) 101: Pause time minus 512 time slots (PT - 512 time slots) 110-111: Reserved

Bit Field	Name	Description
		Time slots are defined as the time required for the MII interface to transmit 512 bits (64 bytes). This (approximate) calculation is based on the packet size (64, 1518, 2000, 9018, 16384, or 32768) + 2 pause packet sizes + the time slot of the Inter Packet Gap (IPG).
3:2	Reserved	Reserved, must be kept at reset value.
1	TFE	Transmit Flow Control Enable In full-duplex mode, when this bit is set to 1, the MAC enables flow control operations to send pause packets. Resetting this bit disables flow control operations in the MAC, and the MAC will not send any pause packets. In half-duplex mode, when this bit is set to 1, the MAC enables backpressure operations. Resetting this bit disables the backpressure functionality. 0: Disable transmit flow control function 1: Enable transmit flow control function
0	FCB_BPA	Flow Control Busy or Backpressure Activate. This bit initiates the sending of pause packets in full-duplex mode, and activates backpressure functionality in half-duplex mode when TFE is set to 1. In full-duplex mode, this bit should read as 0 before writing data to this register. To initiate the sending of pause packets, the application must set this bit to 1. During the transmission of control packets, this bit remains set to 1 to indicate that packet transmission is in progress. When the pause packet transmission is completed, the MAC resets this bit to 0. No write operations should be performed on this register until this bit is cleared. In half-duplex mode, if this bit is set to 1 (and TFE is also set to 1), the MAC activates backpressure. During backpressure operation, when the MAC receives a new data packet, the transmitter will start sending a JAM pattern that causes a collision. This control register bit is logically ORed with the flow control input signal for backpressure functionality. When the MAC is configured in full-duplex mode, BPA will be automatically disabled. 0: Disable Flow Control Busy or Backpressure Activate function 1: Enable Flow Control Busy or Backpressure Activate function *Note: This bit has access restrictions, writing 1 is effective, automatically cleared, writing 0 has no effect.*

35.6.1.10 ETH MAC Rx Flow Control register (ETH_MACRXFLWCTRL)

Address offset: 0x0090

Reset value: 0x0000 0000

The Receive Flow Control register controls the pausing of MAC Transmit based on the received Pause packet.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved														UP	RFE

rw rw

Bit Field	Name	Description
31:2	Reserved	Reserved, must be kept at reset value.
1	UP	Unicast Pause Packet Detect. When a pause packet has the unique multicast address specified in IEEE 802.3, it will be processed. When this bit is set to 1, the MAC can also detect pause packets with a unicast address matching the station's MAC address. This unicast address should be the same as the address specified in the MAC Address 0 High Register and MAC Address 0 Low Register. When this bit is reset, the MAC only detects pause packets with a unique multicast address. 0: Disable Unicast Pause Packet Detect function 1: Enable Unicast Pause Packet Detect function *Note: If the multicast address is different from the unique multicast address, the MAC will not process the pause packets. This also applies to PFC packets received when Priority Flow Control (PFC) is enabled. The unique multicast address (0x01_80_C2_00_00_01) is as specified in IEEE 802.1 Qbb-2011.*
0	RFE	Receive Flow Control Enable. When this bit is set to 1 and the MAC operates in full-duplex mode, the MAC decodes incoming pause packets and prevents transmission for the specified (pause) time. When this bit is reset or the MAC operates in half-duplex mode, the decoding of pause packets is disabled. When PFC is enabled, flow control is enabled for PFC packets. The MAC decodes received PFC packets and disables the transmission queue with a matching priority within the received pause time. 0: Disable Receive Flow Control function 1: Enable Receive Flow Control function

35.6.1.11 ETH MAC Interrupt Status register (ETH_MACINTSTS)

Address offset: 0x00B0

Reset value: 0x0000 0000

The Interrupt Status register contains the status of interrupts.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved													MDIOIS	Reserved	
													r	r	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	RXSTSIS	TXSTSIS	TSIS	Reserved	MMCTXIS	MMCRXIS	MMCIS	Reserved			PMTIS	PHYIS	Reserved		
	r	r	r		r	r	r				r	r			

Bit Field	Name	Description
31:19	Reserved	Reserved, must be kept at reset value.
18	MDIOIS	MDIO Interrupt Status. This bit indicates the interrupt event after an MDIO operation is completed. To reset this bit, the application must read this bit or write a 1 to this bit when the RCWE bit in the MAC CSR software control register is set. 0: MDIO Interrupt Status not active 1: MDIO Interrupt Status active *Note: This bit has access restrictions. Reading (or writing 1 when the RCWE bit in the MAC CSR software control register is set) clears it. It is automatically set to 1 during internal events.*
17:15	Reserved	Reserved, must be kept at reset value.
14	RXSTSIS	Receive Status Interrupt. This bit indicates the status of received data packets. It is set when the RWT bit in the MAC Rx Tx status register is set. To clear this bit, read the corresponding interrupt source bit in the MAC Rx Tx status register (or write "1" to the corresponding interrupt source bit when the RCWE bit in the MAC CSR software control register is set). 0: Receive interrupt status not active 1: Receive interrupt status active
13	TXSTSIS	Transmit Status Interrupt. This bit indicates the status of transmitted data packets. It is set when any of the following bits in the MAC Rx Tx status register are set: ■ Excessive Collisions (EXCOL) ■ Late Collisions (LCOL) ■ Excessive Deferral (EXDEF) ■ Carrier Loss (LCARR) ■ No Carrier (NCARR) ■ Jabber Timeout (TJT) To clear this bit, read the corresponding interrupt source bit in the MAC Rx Tx status register (or write "1" to the corresponding interrupt source bit when the RCWE bit in the MAC CSR software control register is set). 0: Transmit interrupt status not active 1: Transmit interrupt status active
12	TSIS	Timestamp Interrupt Status. If the timestamp function is enabled, this bit is set when any of the following conditions are true: ■ The system time value equals or exceeds the value specified in the target time high register and target time low register. ■ There is an overflow in the second register. ■ A target time error occurs, meaning the programmed target time has passed. If the discard transmit status is enabled in the MTL, this bit is set to 1 when the captured transmit timestamp is updated in the MAC Tx timestamp status nanosecond register and MAC Tx timestamp status second register.

Bit Field	Name	Description
		To clear this bit, read the corresponding interrupt source bit in the MAC timestamp status register (or write "1" to the corresponding interrupt source bit when the RCWE bit in the MAC CSR software control register is set). 0: Timestamp interrupt status not active 1: Timestamp interrupt status active *Note: To generate a timestamp interrupt (if enabled) when the first condition is true, ensure that the third condition is false and set bit 4 (TSTRIG) in the MAC timestamp control register to 1.*
11	Reserved	Reserved, must be kept at reset value.
10	MMCTXIS	MMC Transmit Interrupt Status. This bit is set to 1 when an interrupt is generated in the MMC transmit interrupt register. After clearing all the bits in the interrupt register, this bit will be cleared as well. 0: MMC Transmit interrupt status not active 1: MMC Transmit interrupt status active
9	MMCRXIS	MMC Receive Interrupt Status. This bit is set to 1 when an interrupt is generated in the MMC receive interrupt register. Upon clearing all bits in the interrupt register, this bit will be cleared as well. 0: MMC Receive interrupt status not active 1: MMC Receive interrupt status active
8	MMCIS	MMC Interrupt Status. This bit is set to 1 when bit11, bit10, or bit9 is set to high in the MMC interrupt register. When all of these bits are low, this bit will be cleared. 0: MMC interrupt status not active 1: MMC interrupt status active
7:5	Reserved	Reserved, must be kept at reset value.
4	PMTIS	PMT Interrupt Status. This bit is set to 1 when a magic packet or Wake-on-LAN packet (remote wake-up packet) is received. When a read operation is performed on the MAC PMT control status register to clear the corresponding interrupt source bit (or when writing "1" to the corresponding interrupt source bit in the MAC PMT control status register when the RCWE bit in the MAC CSR software control register is set), this bit is cleared. 0: PMT interrupt status not active 1: PMT interrupt status active
3	PHYIS	PHY Interrupt. This bit is set to 1 when the rising edge of the phy_intr_i input signal is detected. When reading this register (or writing "1" to this bit when the RCWE bit in the MAC CSR software control register is set), the bit is cleared. 0: No PHY interrupt detected 1: PHY interrupt detected
2:0	Reserved	Reserved, must be kept at reset value.

35.6.1.12 ETH MAC Interrupt Enable register (ETH_MACINTEN)

Address offset: 0x00B4

Reset value: 0x0000 0000

The Interrupt Enable register contains the masks for generating the interrupts.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved													MDIOIE	Reserved	
													rw	rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	RXSTSIE	TXSTSIE	TSIE	Reserved							PMTIE	PHYIE	Reserved		
	rw	rw	rw								rw	rw			

Bit Field	Name	Description
31:19	Reserved	Reserved, must be kept at reset value.
18	MDIOIE	MDIO Interrupt Enable. When this bit is set to 1, the interrupt signal is enabled in the MDIOIS bit of the MAC interrupt status register. 0: Disable MDIO interrupt 1: Enable MDIO interrupt
17:15	Reserved	Reserved, must be kept at reset value.
14	RXSTSIE	Receive Status Interrupt Enable. When this bit is set to 1, the interrupt signal is enabled due to the RXSTSIS bit in the MAC interrupt status register. 0: Disable Receive Status interrupt 1: Enable Receive Status interrupt
13	TXSTSIE	Transmit Status Interrupt Enable. When this bit is set to 1, the interrupt signal is enabled due to the TXSTSIS bit in the MAC interrupt status register. 0: Disable Transmit Status interrupt 1: Enable Transmit Status interrupt
12	TSIE	Timestamp Interrupt Enable. When this bit is set to 1, the interrupt signal is enabled due to the TSIS bit in the MAC interrupt status register. 0: Disable Timestamp interrupt 1: Enable Timestamp interrupt
11:5	Reserved	Reserved, must be kept at reset value.
4	PMTIE	PMT Interrupt Enable. When this bit is set to 1, the interrupt signal is enabled due to the PMTIS bit in the MAC interrupt status register. 0: Disable PMT interrupt

Bit Field	Name	Description
		1: Enable PMT interrupt
3	PHYIE	PHY Interrupt Enable. When this bit is set to 1, the interrupt signal is enabled due to the PHYIS bit in the MAC interrupt status register. 0: Disable PHY interrupt 1: Enable PHY interrupt
2:0	Reserved	Reserved, must be kept at reset value.

35.6.1.13 ETH MAC Rx Tx Status register (ETH_MACRXTXSTS)

Address offset: 0x00B8

Reset value: 0x0000 0000

The Receive Transmit Status register contains the Receive and Transmit Error status.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved							RWT	Reserved		EXCOL	LCOL	EXDEF	LCARR	NCARR	TJT
							r			r	r	r	r	r	r

Bit Field	Name	Description
31:9	Reserved	Reserved, must be kept at reset value.
8	RWT	Receive Watchdog Timeout. This bit is set to 1 when a data packet greater than 2048 bytes (or 10240 bytes if jumbo frame mode is enabled) is received and the WD bit in the MAC configuration register is reset. It is also set to 1 when a data packet greater than 16383 bytes is received and the WD bit in the MAC configuration register is set. 0: No Receive Watchdog Timeout 1: Receive Watchdog Timeout occurred *Note: This bit has access restrictions. Reading (or writing 1 when the RCWE bit in the MAC CSR software control register is set) clears it. It is automatically set to 1 during internal events.*
7:6	Reserved	Reserved, must be kept at reset value.
5	EXCOL	Excessive Collisions. When the DTXSTS bit in the MTL operation mode register is set to 1, this bit indicates that the transmission of the current data packet was aborted due to encountering 16 consecutive collisions during the attempt to send. If the DR bit in the MAC configuration register is set to 1, this bit is set to 1 after the first collision, and the data packet transmission process is aborted. 0: No collisions

Bit Field	Name	Description
		1: Excessive Collisions detected *Note: This bit has access restrictions. Reading (or writing 1 when the RCWE bit in the MAC CSR software control register is set) clears it. It is automatically set to 1 during internal events.*
4	LCOL	Late Collision When the DTXSTS bit in the MTL operation mode register is set to 1, this bit indicates that the data packet transmission was interrupted due to a collision window (64 bytes, including the preamble in MII mode). This bit is invalid if an underflow error occurs. 0: No collisions 1: Late Collision detected *Note: This bit has access restrictions. Reading (or writing 1 when the RCWE bit in the MAC CSR software control register is set) clears it. It is automatically set to 1 during internal events.*
3	EXDEF	Excessive Deferral. When the DTXSTS bit in the MTL operation mode register is set to 1 and the DC bit in the MAC configuration register is set to 1, this bit indicates that the transmission was terminated due to a delay exceeding 24288 bit times. 0: No Excessive Deferral 1: Excessive Deferral *Note: This bit has access restrictions. Reading (or writing 1 when the RCWE bit in the MAC CSR software control register is set) clears it. It is automatically set to 1 during internal events.*
2	LCARR	Loss of Carrier. When the DTXSTS bit in the MTL operation mode register is set to 1, this bit indicates that a loss of carrier occurred during the data packet transmission. This means that the phy_crs_i signal was inactive for one or more transmission clock cycles during the data packet transmission. This bit is only valid for collision-free data packet transmissions. 0: Carrier present 1: Loss of Carrier *Note: This bit has access restrictions. Reading (or writing 1 when the RCWE bit in the MAC CSR software control register is set) clears it. It is automatically set to 1 during internal events.*
1	NCARR	No Carrier. When the DTXSTS bit in the MTL operation mode register is set to 1, this bit indicates the absence of a carrier signal from the PHY at the end of the preamble transmission. 0: Carrier present 1: No Carrier *Note: This bit has access restrictions. Reading (or writing 1 when the RCWE bit in the MAC CSR software control register is set) clears it. It is automatically set to 1 during internal events.*
0	TJT	Transmit Jabber Timeout. When the size of a data packet exceeds 2048 bytes (or 10240 bytes if jumbo frame

Bit Field	Name	Description
		mode is enabled) and the JD bit in the MAC configuration register is reset, this bit indicates that the transmit Jabber timer has expired. It is also set to 1 when the size of a data packet exceeds 16383 bytes and the JD bit in the MAC configuration register is set. 0: No Transmit Jabber Timeout 1: Transmit Jabber Timeout occurred *Note: This bit has access restrictions. Reading (or writing 1 when the RCWE bit in the MAC CSR software control register is set) clears it. It is automatically set to 1 during internal events.*

35.6.1.14 ETH PMT Control Status register (ETH_MACPMTCTRLSTS)

Address offset: 0x00C0

Reset value: 0x0000 0000

The PMT Control Status Register contains control and status bits related to remote wake-up packets and magic packets.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RWKFILTRST	Reserved		RWKPTR[4:0]					Reserved							
rw					r										

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved					RWKPFE	GLBLUCAST	Reserved		RWKPRCVD	MGKPRCVD	Reserved		RWKPKTEN	MGKPKTEN	PWRDWN
					rw	rw			r	r			rw	rw	rw

Bit Field	Name	Description
31	RWKFILTRST	Remote Wakeup Packet Filter Register Pointer Reset. When this bit is set to 1, the pointer of the Remote Wakeup Packet Filter Register will be reset to 3'b000. It will automatically clear after one clock cycle. 0: Remote wakeup packet filter register pointer not reset 1: Remote wakeup packet filter register pointer reset
30:29	Reserved	Reserved, must be kept at reset value.
28:24	RWKPTR	Remote Wakeup FIFO Pointer. This field indicates the current value (0 to 7) of the pointer in the Remote Wakeup Packet Filter Register. If the value of this pointer is 7, writing to this register will transfer the contents of the Remote Wakeup Packet Filter Register to the clk_rx_i domain.
23:11	Reserved	Reserved, must be kept at reset value.
10	RWKPFE	Remote Wakeup Packet Forwarding Enable. When this bit is set along with RWKPKTEN, the MAC receiver will discard all received frames until the expected wakeup frame is received. After this, all subsequent frames, including the received wakeup frame, will be forwarded to the application. This

Bit Field	Name	Description
		bit is automatically cleared upon receiving the wakeup packet. The application can also clear this bit before receiving the expected wakeup frame. In this case, the MAC will revert to its default behavior of forwarding received packets to the application. This bit should only be set to 1 when RWKPKTEN is set to high and PWRDWN is set to low. The setting of this bit is invalid when PWRDWN is set to high. 0: Disable Remote Wakeup Packet Forwarding function 1: Enable Remote Wakeup Packet Forwarding function *Note: If both magic packet and wakeup frame are enabled while setting this bit, and a magic packet is received before the wakeup frame, this bit will be automatically cleared upon receiving the magic packet. The received magic packet will be discarded, and all frames received after the magic packet will be forwarded to the application. Note: This bit has access restrictions, writing 1 is effective, automatically cleared, writing 0 has no effect.*
9	GLBLUCAST	Global Unicast. When this bit is set to 1, any unicast data packet that passes through the MAC Address Recognition Filter (DAF) will be detected as a remote wakeup packet. 0: Disable Global Unicast feature 1: Enable Global Unicast feature
8:7	Reserved	Reserved, must be kept at reset value.
6	RWKPRCVD	Remote wakeup Packet Received. When this bit is set to 1, it indicates that a power management event was generated due to the reception of a remote wakeup packet. This bit is cleared when a read operation is performed on this register. 0: No remote wakeup packet received 1: Remote wakeup packet received *Note: This bit has access restrictions. Reading (or writing 1 when the RCWE bit in the MAC CSR software control register is set) clears it. It is automatically set to 1 during internal events.*
5	MGKPRCVD	Magic Packet Received When this bit is set to 1, it indicates that a power management event was generated due to the reception of a magic packet. This bit is cleared when a read operation is performed on this register. 0: No magic packet received 1: Magic packet received *Note: This bit has access restrictions. Reading (or writing 1 when the RCWE bit in the MAC CSR software control register is set) clears it. It is automatically set to 1 during internal events.*
4:3	Reserved	Reserved, must be kept at reset value.
2	RWKPKTEN	Remote Wakeup Packet Enable When this bit is set to 1, a power management event will be generated when the MAC receives a remote wakeup packet. 0: Disable Remote Wakeup Packet function

Bit Field	Name	Description
		1: Enable Remote Wakeup Packet function
1	MGKPKTEN	Magic Packet Enable. When this bit is set to 1, a power management event will be generated when the MAC receives a magic packet. 0: Disable Magic Packet function 1: Enable Magic Packet function
0	PWRDWN	Power Down. When this bit is set to 1, the MAC receiver will discard all received packets until the expected magic packet or remote wakeup packet is received. Subsequently, this bit will be automatically cleared, and the power down mode will be disabled. Software can clear this bit before receiving the expected magic packet or remote wakeup packet. After clearing this bit, packets received by the MAC will be forwarded to the application. This bit should only be set to 1 when the Magic Packet Enable, Global Unicast, or Remote Wakeup Packet Enable bits are set to high. 0: Disable Power Down mode 1: Enable Power Down mode *Note: The CSR clock can be turned off in Power Down mode. However, when the CSR clock is off, no read or write operations can be performed on this register, so the software cannot clear this bit.*

35.6.1.15 ETH MAC RWK Filter register (ETH_MACRWUPFLT)

Address offset: 0x00C4

Reset value: 0x0000 0000

The Remote Wakeup Filter Register is implemented in the form of 8 indirect access registers (wkuppktfilter_reg#i) and is accessed by the application through the MAC Remote Wakeup Filter Register. When programming the Remote Wakeup Filter, the entire set of wkuppktfilter_reg registers must be written. Programming of the wkuppktfilter_reg register is accomplished by sequentially writing the values of the 8 registers wkuppktfilter_reg0, wkuppktfilter_reg1, ..., wkuppktfilter_reg7 in the MAC Remote Wakeup Filter Register. Reading from the wkuppktfilter_reg registers is done in a similar manner. The MAC updates the current pointer value of the wkuppktfilter_reg register in the RWKPTR field of the MAC PMT Control Status Register.

For a description of the fields in the Remote Wakeup Filter Register, refer to section 35.5.9.2.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
WKUPFRMFTR[31:16]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WKUPFRMFTR[15:0]															
rw															

Bit Field	Name	Description
31:0	WKUPFRMFTR	Remote Wakeup Filter (RWK Packet Filter) This field contains various controls for the RWK packet filter.

35.6.1.16 ETH MAC Version register (ETH_MACVER)

Address offset: 0x0110

Reset value: 0x0000 0852

The version register identifies the version of the ETH. This register contains two bytes: one that Synopsys uses to identify the IP release number, and the other that you set while configuring the IP.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USERVER[7:0]								SNPSVER[7:0]							
r								r							

Bit Field	Name	Description
31:16	Reserved	Reserved, must be kept at reset value.
15:8	USERVER[7:0]	User-defined Version.
7:0	SNPSVER[7:0]	Synopsys-defined Version.

35.6.1.17 ETH MAC Debug register (ETH_MACDBG)

Address offset: 0x0114

Reset value: 0x0000 0000

The Debug register provides the debug status of various MAC blocks.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved													TFCSTS		TPESTS
													r		r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													RFCFCSTS		RPESTS
													r		r

Bit Field	Name	Description
31:19	Reserved	Reserved, must be kept at reset value.
18:17	TFCSTS	MAC Transmit Packet Controller Status. This field indicates the status of the MAC Transmit Packet Controller module:

Bit Field	Name	Description
		00: Idle state 01: Waiting for one of the following: completion of the previous packet, Inter-Packet Gap (IPG), or back-off period 10: Generating and sending a pause control packet (in full-duplex mode) 11: Transmitting the input data packet to be sent
16	TPESTS	MAC MII Transmit Protocol Engine Status. When this bit is set to 1, it indicates that the MAC MII transmit protocol engine is actively sending data and is not in an idle state. 0: MAC MII Transmit Protocol Engine status not detected 1: MAC MII Transmit Protocol Engine status detected
15:3	Reserved	Reserved, must be kept at reset value.
2:1	RFCFCSTS	MAC Receive Packet Controller FIFO Status. When this field is set to 1, it indicates the active status of the read and write controllers of the various small FIFOs in the MAC Receive Packet Controller module.
0	RPESTS	MAC MII Receive Protocol Engine Status. When this bit is set to 1, it indicates that the MAC MII receive protocol engine is actively receiving data and is not in an idle state. 0: MAC MII Receive Protocol Engine status not detected 1: MAC MII Receive Protocol Engine status detected

35.6.1.18 ETH MAC HW Feature register 0 (ETH_MACHWF0)

Address offset: 0x011C

Reset value: 0x020D 51F5

This register indicates the first set of optional Ethernet peripheral functions. The software driver can use this register to dynamically enable or disable the programs associated with the optional modules.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved	ACTPHYSEL			SAVLANINS	TSSTSSEL		MACADR64SEL	MACADR32SEL	ADDMACADRSEL					Reserved	RXCOESEL
	r			r	r		r	r	r						r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	TXCOESEL	EEESEL	TSSEL	Reserved		ARPOFFSEL	MMCSEL	MGKSEL	RWKSEL	SMASEL	VLHASH	Reserved	HDSEL	GMIISEL	MIISEL
	r	r	r			r	r	r	r	r	r		r	r	r

Bit Field	Name	Description
31	Reserved	Reserved, must be kept at reset value.
30:28	ACTPHYSEL	Active PHY Selected. When there are multiple PHY interfaces in the configuration, this field indicates the sampled value of phy_intf_sel_i during reset support. 0x0: MII 0x4: RMII

Bit Field	Name	Description
		Other values: Reserved
27	SAVLANINS	Source Address or VLAN Insertion Enabled. When the "Enable SA and VLAN Insertion on Tx" option is selected, this bit is set to 1. 0: The "Enable SA and VLAN Insertion" option is not selected 1: The "Enable SA and VLAN Insertion" option is selected
26:25	TSSTSSEL	Timestamp System Time Source. This field indicates the source of the timestamp system time: 0x0: Internal 0x1: External 0x2: Internal and External
24	MACADR64SEL	MAC Addresses 64-127 Selected. When the "Enable Additional MAC Addresses Register 64-127" option is selected, this bit is set to 1. 0: The "Additional MAC Addresses Register 64-127" option is not selected 1: The "Additional MAC Addresses Register 64-127" option is selected
23	MACADR32SEL	MAC Addresses 32-63 Selected. When the "Enable Additional MAC Addresses Register 32-63" option is selected, this bit is set to 1. 0: The "Additional MAC Addresses Register 32-63" option is not selected 1: The "Additional MAC Addresses Register 32-63" option is selected
22:18	ADDMACADRSEL	MAC Addresses 1-31 Selected. This field indicates the additional MAC addresses. 0x1: Enable Additional MAC Address 1 0x2: Enable Additional MAC Address 2 0x3: Enable Additional MAC Address 3 ... 0x1F: Enable Additional MAC Address 31
17	Reserved	Reserved, must be kept at reset value.
16	RXCOESEL	Receive Checksum Offload Enabled. When the "Enable Receive TCP/IP Checksum Offload" option is selected, this bit is set to 1. 0: The "Receive Checksum Offload" option is not selected 1: The "Receive Checksum Offload" option is selected
15	Reserved	Reserved, must be kept at reset value.
14	TXCOESEL	Transmit Checksum Offload Enabled. When the "Enable Transmit TCP/IP Checksum Offload" option is selected, this bit is set to 1. 0: The "Transmit Checksum Offload" option is not selected 1: The "Transmit Checksum Offload" option is selected
13	EEESEL	Energy Efficient Ethernet Enabled. When the "Enable Energy Efficient Ethernet" option is selected, this bit is set to 1. 0: The "Enable Energy Efficient Ethernet" option is not selected

Bit Field	Name	Description
		1: The "Enable Energy Efficient Ethernet" option is selected
12	TSSEL	IEEE 1588-2008 Timestamp Enabled. When the "Enable IEEE 1588 Timestamp" option is selected, this bit is set to 1. 0: The "Enable IEEE 1588 Timestamp" option is not selected 1: The "Enable IEEE 1588 Timestamp" option is selected
11:10	Reserved	Reserved, must be kept at reset value.
9	ARPOFFSEL	ARP Offload Enabled. When the "Enable IPv4 ARP Offload" option is selected, this bit is set to 1. 0: The "Enable IPv4 ARP Offload" option is not selected 1: The "Enable IPv4 ARP Offload" option is selected
8	MMCSEL	RMON Module Enabled. When the "Enable MAC Management Counters (MMC)" option is selected, this bit is set to 1. 0: The "Enable MAC Management Counters (MMC)" option is not selected 1: The "Enable MAC Management Counters (MMC)" option is selected
7	MGKSEL	PMT Magic Packet Enabled. When the "Enable Magic Packet Detection" option is selected, this bit is set to 1. 0: The "Enable Magic Packet Detection" option is not selected 1: The "Enable Magic Packet Detection" option is selected
6	RWKSEL	PMT Remote Wake-up Packet Enabled. When the "Enable Remote Wake-up Packet Detection" option is selected, this bit is set to 1. 0: The "Enable Remote Wake-up Packet Detection" option is not selected 1: The "Enable Remote Wake-up Packet Detection" option is selected
5	SMASEL	SMA (MDIO) Interface. When the "Enable Station Management (MDIO Interface)" option is selected, this bit is set to 1. 0: SMA (MDIO) Interface not selected 1: SMA (MDIO) Interface selected
4	VLHASH	VLAN Hash Filter Selected. This bit is set to 1 when the Enable VLAN Hash Table Based Filtering option is selected. 0: VLAN Hash Filter not selected 1: VLAN Hash Filter selected
3	Reserved	Reserved, must be kept at reset value.
2	HDSEL	Half-duplex Support. This bit is set to 1 when the half-duplex mode is selected. 0: Half-duplex not supported 1: Half-duplex supported
1	GMIISEL	1000 Mbps Support. This bit is set to 1 when 1000 Mbps is selected as the Mode of Operation. 0: 1000 Mbps not supported

Bit Field	Name	Description
		1: 1000 Mbps supported
0	MIISEL	10/100 Mbps Support. This bit is set to 1 when 10/100 Mbps is selected as the Mode of Operation. 0: 10/100 Mbps not supported 1: 10/100 Mbps supported

35.6.1.19 ETH MAC HW Feature register 1 (ETH_MACHWF1)

Address offset: 0x0120

Reset value: 0x0100 0124

This register indicates the second group of optional Ethernet peripheral functions. The software driver can use this register to dynamically enable or disable programs related to the optional module.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved	L3L4FNUM				Reserved	HASHTBLSZ		POUOST	Reserved	RAVSEL	AVSEL	DBGMEMA	TSOEN	SPHEN	DCBEN
	r					r		r		r	r	r	r	r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR64		ADVTHWORD	PTOEN	OSTEN	TXFIFOSIZE					SPRAM	RXFIFOSIZE				
r		r	r	r	r					r	r				

Bit Field	Name	Description
31	Reserved	Reserved, must be kept at reset value.
30:27	L3L4FNUM	Total number of L3 or L4 Filters. This field indicates the total number of L3 or L4 filters: 0000: No L3 or L4 filters 0001: 1 L3 or L4 filter 0010: 2 L3 or L4 filters ... 1000: 8 L3 or L4 filters
26	Reserved	Reserved, must be kept at reset value.
25:24	HASHTBLSZ	Hash Table Size. This field indicates the size of the hash table: 00: No hash table 01: 64 10: 128 11: 256
23	POUOST	One Step for PTP over UDP/IP Feature Enable. This bit is set to 1 when the Enable One step timestamp for PTP over UDP/IP feature is selected. 0: "One Step for PTP over UDP/IP" feature not selected 1: "One Step for PTP over UDP/IP" feature selected

Bit Field	Name	Description
22	Reserved	Reserved, must be kept at reset value.
21	RAVSEL	Rx Side Only AV Feature Enabled. This bit is set to 1 when the Enable Audio Video Bridging option on Rx Side Only is selected. 0: "Rx Side Audio/Video Bridging (AV)" feature not selected 1: "Rx Side Audio/Video Bridging (AV)" feature selected
20	AVSEL	AV Feature Enabled. This bit is set to 1 when the Enable Audio Video Bridging option is selected. 0: "Audio/Video Bridging (AV)" feature not selected 1: "Audio/Video Bridging (AV)" feature selected
19	DBGMEMA	DMA Debug Registers Enabled. This bit is set to 1 when the Debug Mode Enable option is selected. 0: "DMA Debug Registers" option not selected 1: "DMA Debug Registers" option selected
18	TSOEN	TCP Segmentation Offload Enabled. This bit is set to 1 when the Enable TCP Segmentation Offloading for TCP/IP Packets option is selected. 0: "TCP Segmentation Offload" feature is not selected 1: "TCP Segmentation Offload" feature is selected
17	SPHEN	Split Header Feature Enabled. This bit is set to 1 when the Enable Split Header Structure option is selected. 0: "Split Header" feature not selected 1: "Split Header" feature selected
16	DCBEN	DCB Feature Enabled. This bit is set to 1 when the Enable Data Center Bridging option is selected. 0: "Data Center Bridging" feature not selected 1: "Data Center Bridging" feature selected
15:14	ADDR64	Address Width. This field indicates the configured address width: 0x0: 32 0x1: 40 0x2: 48 0x3: Reserved
13	ADVTHWORD	IEEE 1588 High Word Register Enabled. This bit is set to 1 when the Add IEEE 1588 Higher Word Register option is selected. 0: "IEEE 1588 High Word Register" option not selected 1: "IEEE 1588 High Word Register" option selected
12	PTOEN	PTP Offload Enabled. This bit is set to 1 when the Enable PTP Timestamp Offload Feature is selected. 0: "PTP Offload" feature not selected 1: "PTP Offload" feature selected
11	OSTEN	One-Step Timestamp Enabled.

Bit Field	Name	Description
		This bit is set to 1 when the Enable One-Step Timestamp Feature is selected. 0: "One-Step Timestamp" feature not selected 1: "One-Step Timestamp" feature selected
10:6	TXFIFOSIZE	MTL Transmit FIFO Size. This field contains the configuration value of the MTL Tx FIFO (in bytes), represented as the logarithm to the base 2 minus 7, i.e., Log2(TXFIFO_SIZE) - 7: 0x0: 128 bytes 0x1: 256 bytes 0x2: 512 bytes 0x3: 1024 bytes 0x4: 2048 bytes 0x5: 4096 bytes 0x6: 8192 bytes 0x7: 16384 bytes 0x8: 32KB 0x9: 64KB 0xA: 128KB 0xB: Reserved
5	SPRAM	Single Port RAM Enabled. This bit is set to 1 when the Use single port RAM Feature is selected. 0: "Single Port RAM" feature not selected 1: "Single Port RAM" feature selected
4:0	RXFIFOSIZE	MTL Receive FIFO Size. This field contains the configuration value of the MTL Rx FIFO (in bytes), represented as the logarithm to the base 2 minus 7, i.e., Log2(RXFIFO_SIZE) - 7: 0x0: 128 bytes 0x1: 256 bytes 0x2: 512 bytes 0x3: 1024 bytes 0x4: 2048 bytes 0x5: 4096 bytes 0x6: 8192 bytes 0x7: 16384 bytes 0x8: 32KB 0x9: 64KB 0xA: 128KB 0xB: 256KB 0xC: Reserved

35.6.1.20 ETH MAC HW Feature register 2 (ETH_MACHWF2)

Address offset: 0x0124

Reset value: 0x0000 0000

This register indicates the third group of optional Ethernet peripheral functions. The software driver can use this register to dynamically enable or disable programs related to the optional module.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved	AUXSNAPNUM			Reserved	PPSOUTNUM			TDCSZ		TXCHCNT				RDCSZ	
	r				r			r		r				r	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RXCHCNT				Reserved		TXQCNT				Reserved		RXQCNT			
r						r						r			

Bit Field	Name	Description
31	Reserved	Reserved, must be kept at reset value.
30:28	AUXSNAPNUM	Number of Auxiliary Snapshot Inputs. This field indicates the number of auxiliary snapshot inputs: 0x0: No auxiliary inputs 0x1: 1 auxiliary input 0x2: 2 auxiliary inputs 0x3: 3 auxiliary inputs 0x4: 4 auxiliary inputs 0x5: Reserved
27	Reserved	Reserved, must be kept at reset value.
26:24	PPSOUTNUM	Number of PPS Outputs. This field indicates the number of PPS outputs: 0x0: No PPS outputs 0x1: 1 PPS output 0x2: 2 PPS outputs 0x3: 3 PPS outputs 0x4: 4 PPS outputs 0x5: Reserved
23:22	TDCSZ	Tx DMA Descriptor Cache Size in terms of 16 bytes descriptors . 0x0: Cache not configured 0x1: 4 0x2: 8 0x3: 16
21:18	TXCHCNT	Number of DMA Transmit Channels. This field indicates the number of DMA transmit channels: 0x0: 1 DMA transmit channel 0x1: 2 DMA transmit channels … 0x7: 8 DMA transmit channels
17:16	RDCSZ	Rx DMA Descriptor Cache Size in terms of 16 bytes descriptors.

Bit Field	Name	Description
		0x0: Cache not configured 0x1: 4 0x2: 8 0x3: 16
15:12	RXCHCNT	Number of DMA Receive Channels. This field indicates the number of DMA receive channels: 0x0: 1 DMA receive channel 0x1: 2 DMA receive channels ... 0x7: 8 DMA receive channels
11:10	Reserved	Reserved, must be kept at reset value.
9:6	TXQCNT	Number of MTL Transmit Queues. This field indicates the number of MTL transmit queues: 0x0: 1 MTL transmit queue 0x1: 2 MTL transmit queues ... 0x7: 8 MTL transmit queues
5:4	Reserved	Reserved, must be kept at reset value.
3:0	RXQCNT	Number of MTL Receive Queues. This field indicates the number of MTL receive queues: 0x0: 1 MTL receive queue 0x1: 2 MTL receive queues ... 0x7: 8 MTL receive queues

35.6.1.21 ETH MAC HW Feature register 3 (ETH_MACHWF3)

Address offset: 0x0128

Reset value: 0x0000 0000

This register indicates the fourth group of optional Ethernet peripheral functions. The software driver can use this register to dynamically enable or disable programs related to the optional module.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		ASP		TBSSEL	FPESEL	Reserved				ESTWID		ESTDEP			ESTSEL
		r		r	r					r		r			r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	FRPES		FRPBS		FRPSEL	PDUPSEL	Reserved			DVLAN	CBTISEL	Reserved	NRVF		
	r		r		r	r				r	r		r		

Bit Field	Name	Description
31:30	Reserved	Reserved, must be kept at reset value.

Bit Field	Name	Description
29:28	ASP	Automotive Safety Package. The following are the codes for different safety features: 0x0: No safety features selected 0x1: Only "ECC protection for external memory" feature selected 0x2: All automotive safety features selected, but "Enable external interface parity check port" feature not selected 0x3: All automotive safety features selected, including "Enable external interface parity check port" feature
27	TBSSEL	Time Based Scheduling Enable. This bit is set to 1 when the Time Based Scheduling feature is selected. 0: "Time Based Scheduling" feature not selected 1: "Time Based Scheduling" feature selected
26	FPESEL	Frame Preemption Enable. This bit is set to 1 when the Enable Frame preemption feature is selected. 0: "Frame Preemption" feature not selected 1: "Frame Preemption" feature selected
25:22	Reserved	Reserved, must be kept at reset value.
21:20	ESTWID	Width of the Time Interval field in the Gate Control List. This field indicates the width of the configured time interval field. 0x0: Width not configured 0x1: 4 0x2: 8 0x3: 16
19:17	ESTDEP	Depth of the Gate Control List. This field indicates the depth of the gate control list, represented as Log2(DWC_EQOS_EST_DEP) - 5. 0x0: Depth not configured 0x1: 64 0x2: 128 0x3: 256 0x4: 512 0x5: 1024 0x6: Reserved
16	ESTSEL	Enhancements to Scheduled Traffic Enable. This bit is set to 1 when the Enable Enhancements to Scheduling Traffic feature is selected. 0: "Enhancements to Scheduled Traffic" feature not selected 1: "Enhancements to Scheduled Traffic" feature selected
15	Reserved	Reserved, must be kept at reset value.
14:13	FRPES	Flexible Receive Parser Table Entries size. This field indicates the maximum number of parser entries supported by the flexible receive parser.

Bit Field	Name	Description
		0x0: 64 0x1: 128 0x2: 256 0x3: Reserved
12:11	FRPBS	Flexible Receive Parser Buffer size. This field indicates the maximum number of bytes of packet data that can be parsed by the flexible receive parser. 0x0: 64 bytes 0x1: 128 bytes 0x2: 256 bytes 0x3: Reserved
10	FRPSEL	Flexible Receive Parser Selected. This bit is set to 1 when the Enable Flexible Programmable Receive Parser option is selected. 0: "Flexible Receive Parser" feature not selected 1: "Flexible Receive Parser" feature selected
9	PDUPSEL	Broadcast/Multicast Packet Duplication. This bit is set to 1 when the Broadcast/Multicast Packet Duplication feature is selected. 0: "Broadcast/Multicast Packet Duplication" feature not selected 1: "Broadcast/Multicast Packet Duplication" feature selected
8:6	Reserved	Reserved, must be kept at reset value.
5	DVLAN	Double VLAN Tag Processing Selected. This bit is set to 1 when the Enable Double VLAN Processing Feature is selected. 0: "Double VLAN Tag Processing" option not selected 1: "Double VLAN Tag Processing" option selected
4	CBTISEL	Queue/Channel based VLAN tag insertion on Tx Enabled. This bit is set to 1 when the Enable Queue/Channel based VLAN tag insertion on Tx Feature is selected. 0: "Queue/Channel based VLAN tag insertion on Tx" feature not selected 1: "Queue/Channel based VLAN tag insertion on Tx" feature selected
3	Reserved	Reserved, must be kept at reset value.
2:0	NRVF	Number of Extended VLAN Tag Filters Enabled. This field indicates the Number of Extended VLAN Tag Filters selected. 0: No Extended Rx VLAN Filters 1: 4 Extended Rx VLAN Filters 2: 8 Extended Rx VLAN Filters 3: 16 Extended Rx VLAN Filters 4: 24 Extended Rx VLAN Filters 5: 32 Extended Rx VLAN Filters 6: Reserved

35.6.1.22 ETH MAC MDIO Address register (ETH_MACMDIOADDR)

Address offset: 0x0200

Reset value: 0x0000 0000

The MDIO Address register controls the management cycles to external PHY through a management interface.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				PSE	BTB	PA					RDA				
				rw	rw	rw					rw				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	NTC			CR				Reserved			SKAP	MOC1	MOC0	C45E	MB
	rw			rw							rw	rw	rw	rw	rw

Bit Field	Name	Description
31:28	Reserved	Reserved, must be kept at reset value.
27	PSE	Preamble Suppression Enable. When this bit is set, the SMA suppresses the 32-bit preamble and transmits MDIO frames with only 1 preamble bit. When this bit is 0, the MDIO frame always has 32 bits of preamble as defined in the IEEE specificationsn. 0: Disable Preamble Suppression feature 1: Enable Preamble Suppression feature
26	BTB	Back to Back transactions. When this bit is set and the NTC has value greater than 0, then the MAC informs the completion of a read or write command at the end of frame transfer (before the trailing clocks are transmitted). The software can thus initiate the next command which is executed immediately irrespective of the number trailing clocks generated for the previous frame. When this bit is reset, then the read/write command completion (GB is cleared) only after the trailing clocks are generated. In this mode, it is ensured that the NTC is always generated after each frame. This bit must not be set when NTC=0. 0: Disable Back to Back transactions feature 1: Enable Back to Back transactions feature
25:21	PA	Physical Layer Address. This field indicates which Clause 22 PHY devices (out of 32 devices) the MAC is accessing. This field indicates which Clause 45 capable PHYs (out of 32 PHYs) the MAC is accessing.
20:16	RDA	Register/Device Address. These bits select the PHY register in selected Clause 22 PHY device. These bits select the Device (MMD) in selected Clause 45 capable PHY.
15	Reserved	Reserved, must be kept at reset value.
14:12	NTC	Number of Training Clocks.

Bit Field	Name	Description
		This field controls the number of trailing clock cycles generated on gmii_mdc_o (MDC) after the end of transmission of MDIO frame. The valid values can be from 0 to 7. Programming the value to 3'h3 indicates that there are additional three clock cycles on the MDC line after the end of MDIO frame transfer.
11:8	CR	CSR Clock Range. The CSR Clock Range determines the frequency of the MDC clock based on the CSR clock frequency used in the design: 0000: CSR clock = 60100MHz; MDC clock = CSR clock/42 0001: CSR clock = 100150MHz; MDC clock = CSR clock/62 0010: CSR clock = 2035MHz; MDC clock = CSR clock/16 0011: CSR clock = 3560MHz; MDC clock = CSR clock/26 0100: CSR clock = 150250MHz; MDC clock = CSR clock/102 0101: CSR clock = 250300MHz; MDC clock = CSR clock/124 0110: CSR clock = 300500MHz; MDC clock = CSR clock/204 0111: CSR clock = 500800MHz; MDC clock = CSR clock/324 The recommended CSR clock frequency ranges for each value (when bit11 = 0) ensure that the MDC clock falls roughly between 1.0MHz to 2.5MHz frequency range. When bit11 is set to 1, it allows achieving MDC clock frequencies higher than the 2.5MHz limit specified in IEEE 802.3 and programming for lower clock dividers. For example, when the CSR clock frequency is 100MHz and these bits are programmed as 1010, the resulting MDC clock is 12.5MHz, which is above the range specified in IEEE 802.3. Only set the following values if the interface chip supports a faster MDC clock: 1000: CSR clock/4 1001: CSR clock/6 1010: CSR clock/8 1011: CSR clock/10 1100: CSR clock/12 1101: CSR clock/14 1110: CSR clock/16 1111: CSR clock/18
7:5	Reserved	Reserved, must be kept at reset value.
4	SKAP	Skip Address Packet. When this bit is set, the SMA does not send the address packets before read, write, or post-read increment address packets. This bit is valid only when C45E is set. 0: Disable Skip Address Packet feature 1: Enable Skip Address Packet feature
3	MOC1	MII Operation Command 1. This bit indicates the high bit of the operation command for the PHY. 00: Reserved 01: Write operation 10: Post-read operation on an incremental address for a PHY compliant with Clause 45 11: Read operation When enabling a PHY compliant with Clause 22, only read and

Bit Field	Name	Description
		write commands are valid.
2	MOC0	MII Operation Command 0. This bit indicates the low bit of the operation command for the PHY. 00: Reserved 01: Write operation 10: Post-read operation on an incremental address for a PHY compliant with Clause 45 11: Read operation
1	C45E	Clause 45 PHY Enable. When this bit is set, Clause 45 capable PHY is connected to MDIO. When this bit is reset, Clause 22 capable PHY is connected to MDIO.
0	MB	MII Busy The application sets this bit to indicate that the SMA has initiated a read or write access to the MDIO slave. After the completion of the MDIO frame transmission, the MAC clears this bit. Therefore, as long as this bit is set, the software must not write to or change any fields in the MDIO Address Register and MDIO Data Register. For a write transfer, the application must first write 16-bit data to the MD field of the MDIO Data Register before setting this bit. When C45E is set, data should also be written to the RA field of the MDIO Data Register before starting a read transfer. After a read transfer is completed (MB=0), the data read from the PHY register is valid in the MD field of the MDIO Data Register. *Note: Even if the addressed PHY is not present, there is no change in the functionality of this bit. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect.*

35.6.1.23 ETH MAC MDIO Data register (ETH_MACMDIODATA)

Address offset: 0x0204

Reset value: 0x0000 0000

The MDIO Data Register stores the data written to the PHY register at the address specified by the MDIO Address Register. It also stores the data read from the PHY register located at the address specified by the MDIO Address Register.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RA															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MD															
rw															

Bit Field	Name	Description
31:16	RA	Register Address.

Bit Field	Name	Description
		This field is only valid when C45E is set to 1. It contains the register address in the PHY to be used with the MDIO frame.
15:0	MD	MII Data. This field contains the 16-bit data value read from the PHY after a management read operation or the 16-bit data value to be written to the PHY before a management write operation.

35.6.1.24 ETH CSR SW Control register (ETH_MACCSRSWCTRL)

Address offset: 0x0230

Reset value: 0x0000 0000

This register contains software programmable controls for changing the CSR access response and status bits clearing.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved							SEEN	Reserved							RCWE
							rw								rw

Bit Field	Name	Description
31:9	Reserved	Reserved, must be kept at reset value.
8	SEEN	Slave Error Response Enable. When this bit is set, the MAC responds with Slave Error for accesses to reserved registers in CSR space. When this bit is reset, the MAC responds with Okay response to any register accessed from CSR space. 0: Disable Slave Error Response feature 1: Enable Slave Error Response feature
7:1	Reserved	Reserved, must be kept at reset value.
0	RCWE	Register Clear on Write 1 Enable. When this bit is set, the access mode of some register fields changes to Clear on Write 1, the application needs to set that respective bit to 1 to clear it. When this bit is reset, the access mode of these register fields remain as Clear on Read. 0: Disable Register Clear on Write 1 feature 1: Enable Register Clear on Write 1 feature

35.6.1.25 ETH MAC Address0 High register (ETH_MACADDR0H)

Address offset: 0x0300

Reset value: 0x8000 FFFF

The MAC Address 0 High Register stores the high 16 bits of the first 6 bytes MAC address of the station. The first DA bytes received on the MII interface correspond to the LS bytes (bits [7:0]) of the MAC Address Low Register. For example, if 0x112233445566 (0x11 in the first column and row) is received as the destination address on the MII, the MAC Address 0 Register [47:0] will be compared with 0x665544332211.

If the MAC address registers are configured to be double synchronously with the MII clock domain, synchronization will only be triggered when writing to bits [31:24] (little-endian mode) or bits [7:0] (big-endian mode) of the MAC Address 0 Low Register. To achieve correct synchronized updates, continuous writes to this address low register should occur at least four clock cycles after the target clock domain.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AE	Reserved														
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDRHI															
rw															

Bit Field	Name	Description
31	AE	Address Enable This bit is always set to 1.
30:16	Reserved	Reserved, must be kept at reset value
15:0	ADDRHI	MAC Address 0 [47:32] This field contains the high 16 bits [47:32] of the first 6 bytes MAC address. The MAC uses this field to filter received packets and to insert the MAC address in transmitted flow control (pause) packets.

35.6.1.26 ETH MAC Address0 Low register (ETH_MACADDR0L)

Address offset: 0x0304

Reset value: 0xFFFF FFFF

The MAC Address0 Low register holds the lower 32 bits of the 6-byte first MAC address

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADDRLO[31:16]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDRLO[15:0]															
rw															

Bit Field	Name	Description
31:0	ADDRLO	MAC Address0[31:0]

Bit Field	Name	Description
		This field contains the lower 32 bits of the first 6-byte MAC address. The MAC uses this field for filtering the received packets and inserting the MAC address in the Transmit Flow Control (Pause) Packets.

35.6.1.27 ETH MAC Address1 High register (ETH_MACADDR1H)

Address offset: 0x0308

Reset value: 0x0000 FFFF

MAC Address 1 High Register stores the high 16 bits of the second 6 bytes MAC address of the station.

If the MAC address registers are configured to be double synchronously with the MII clock domain, synchronization will only be triggered when writing to bits [31:24] (little-endian mode) or bits [7:0] (big-endian mode) of the MAC Address 1 Low Register. To achieve correct synchronized updates, continuous writes to this address low register should occur at least four clock cycles after the target clock domain.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AE	SA	MBC						Reserved							
rw	rw	rw													

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDRHI															
rw															

Bit Field	Name	Description
31	AE	Address Enable. When this bit is set, the address filter module uses the second MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0: Address is ignored 1: Address is enabled
30	SA	Source Address. When this bit is set, the MAC Address1[47:0] is used to compare with the SA fields of the received packet. When this bit is reset, the MAC Address1[47:0] is used to compare with the DA fields of the received packet 0: Compare with the destination address 1: Compare with the source address
29:24	MBC	Mask Byte Control. These bits are used to control the masking of each MAC address byte for comparison. When set to a high level, the MAC will not compare the corresponding bytes of the received DA or SA with the content of MAC Address 1 register. Each bit controls the byte masking as follows: bit29: bits[15:8] of MAC Address 1 High Register

Bit Field	Name	Description
		bit28: bits[7:0] of MAC Address 1 High Register bit27: bits[31:24] of MAC Address 1 Low Register ... bit24: bits[7:0] of MAC Address 1 Low Register By masking one or more bytes of the address, a group of addresses can be filtered (referred to as group address filtering).
23:16	Reserved	Reserved, must be kept at reset value.
15:0	ADDRHI	MAC Address 1 [47:32]. This field contains the high 16 bits [47:32] of the second 6 bytes MAC address.

35.6.1.28 ETH MAC Address0 Low register (ETH_MACADDR1L)

Address offset: 0x030C

Reset value: 0xFFFF FFFF

The MAC Address0 Low register holds the lower 32 bits of the 6-byte first MAC address

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADDRLO[31:16]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDRLO[15:0]															
rw															

Bit Field	Name	Description
31:0	ADDRLO	MAC Address0[31:0]. This field contains the lower 32 bits of the first 6-byte MAC address. The MAC uses this field for filtering the received packets and inserting the MAC address in the Transmit Flow Control (Pause) Packets.

35.6.1.29 ETH MAC Address2 High register (ETH_MACADDR2H)

Address offset: 0x0310

Reset value: 0x0000 FFFF

MAC Address 2 High Register stores the high 16 bits of the third 6 bytes MAC address of the station.

If the MAC address registers are configured to be double synchronously with the MII clock domain, synchronization will only be triggered when writing to bits [31:24] (little-endian mode) or bits [7:0] (big-endian mode) of the MAC Address 2 Low Register. To achieve correct synchronized updates, continuous writes to this address low register should occur at least four clock cycles after the target clock domain.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AE	SA	MBC						Reserved							
rw	rw	rw													

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDRHI															
rw															

Bit Field	Name	Description
31	AE	Address Enable. When this bit is set, the address filter module uses the third MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0: Address is ignored 1: Address is enabled
30	SA	Source Address. When this bit is set, the MAC Address1[47:0] is used to compare with the SA fields of the received packet. When this bit is reset, the MAC Address1[47:0] is used to compare with the DA fields of the received packet. 0: Compare with the destination address 1: Compare with the source address
29:24	MBC	Mask Byte Control. These bits are used to control the masking of each MAC address byte for comparison. When set to a high level, the MAC will not compare the corresponding bytes of the received DA or SA with the content of the MAC Address 2 register. Each bit controls the byte masking as follows: bit29: bits[15:8] of MAC Address 2 High Register bit28: bits[7:0] of MAC Address 2 High Register bit27: bits[31:24] of MAC Address 2 Low Register ... bit24: bits[7:0] of MAC Address 2 Low Register By masking one or more bytes of the address, a group of addresses can be filtered (referred to as group address filtering).
23:16	Reserved	Reserved, must be kept at reset value.
15:0	ADDRHI	MAC Address 2 [47:32]. This field contains the high 16 bits [47:32] of the third 6 bytes MAC address.

35.6.1.30 ETH MAC Address2 Low register (ETH_MACADDR2L)

Address offset: 0x0314

Reset value: 0xFFFF FFFF

MAC Address 2 Low Register stores the low 32 bits of the third 6 bytes MAC address of the station.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADDRLO[31:16]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDRLO[15:0]															
rw															

Bit Field	Name	Description
31:0	ADDRLO	MAC Address 2 [31:0]. This field contains the low 32 bits of the third 6 bytes MAC address. The MAC uses this field to filter received data packets. The content of this field is undefined before it is loaded by the application after the initialization process.

35.6.1.31 ETH MAC Address3 High register (ETH_MACADDR3H)

Address offset: 0x0318

Reset value: 0x0000 FFFF

MAC Address 3 High Register stores the high 16 bits of the fourth 6 bytes MAC address of the station.

If the MAC address registers are configured to be double synchronously with the MII clock domain, synchronization will only be triggered when writing to bits [31:24] (little-endian mode) or bits [7:0] (big-endian mode) of the MAC Address 3 Low Register. To achieve correct synchronized updates, continuous writes to this address low register should occur at least four clock cycles after the target clock domain.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AE	SA	MBC						Reserved							
rw	rw	rw													

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDRHI															
rw															

Bit Field	Name	Description
31	AE	Address Enable. When this bit is set, the address filter module uses the fourth MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0: Address is ignored 1: Address is enabled
30	SA	Source Address. When this bit is set, the MAC Address1[47:0] is used to compare with the SA fields of the received packet. When this bit is reset, the MAC Address1[47:0] is used to compare with the DA fields of the received packet. 0: Compare with the destination address 1: Compare with the source address

Bit Field	Name	Description
29:24	MBC	Mask Byte Control. These bits are used to control the masking of each MAC address byte for comparison. When set to a high level, the MAC will not compare the corresponding bytes of the received DA or SA with the content of the MAC Address 3 register. Each bit controls the byte masking as follows: bit29: bits[15:8] of MAC Address 3 High Register bit28: bits[7:0] of MAC Address 3 High Register bit27: bits[31:24] of MAC Address 3 Low Register ... bit24: bits[7:0] of MAC Address 3 Low Register By masking one or more bytes of the address, a group of addresses can be filtered (referred to as group address filtering).
23:16	Reserved	Reserved, must be kept at reset value.
15:0	ADDRHI	MAC Address 3 [47:32]. This field contains the high 16 bits [47:32] of the fourth 6 bytes MAC address.

35.6.1.32 ETH MAC Address3 Low register (ETH_MACADDR3L)

Address offset: 0x031C

Reset value: 0xFFFF FFFF

MAC Address 3 Low Register stores the low 32 bits of the fourth 6 bytes MAC address of the station.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADDRLO[31:16]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDRLO[15:0]															
rw															

Bit Field	Name	Description
31:0	ADDRLO	MAC Address 3 [31:0]. This field contains the low 32 bits of the fourth 6 bytes MAC address. The MAC uses this field to filter received data packets. The content of this field is undefined before being loaded by the application after the initialization process.

35.6.1.33 ETH MMC Control register (ETH_MMCCTRL)

Address offset: 0x0700

Reset value: 0x0000 0000

This register establishes the operating mode of MMC.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

Reserved

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved							UCDBC	Reserved		CNTPRST LVL	CNTPRST	CNTFREE Z	RSTONR D	CNTSTOP RO	CNTRST
							rw			rw	rw	rw	rw	rw	rw

Bit Field	Name	Description
31:9	Reserved	Reserved, must be kept at reset value.
8	UCDBC	Update MMC Counters for Dropped Broadcast Packets. When set, the MAC updates all related MMC Counters for Broadcast packets that are dropped because of the setting of the DBF bit of MAC Packet Filter register. When reset, the MMC Counters are not updated for dropped Broadcast packets. 0: Disable updating MMC counters for dropped broadcast packets 1: Enable updating MMC counters for dropped broadcast packets *Note: The CNTRST bit has a higher priority than the CNTPRST bit. Therefore, when the software tries to set both bits in the same write cycle, all counters are cleared and the CNTPRST bit is not set.*
7:6	Reserved	Reserved, must be kept at reset value.
5	CNTPRSTLVL	Full-Half Preset. When this bit is low and the CNTPRST bit is set to 1, all MMC counters are preset to values close to half. All eight-bit byte counters are preset to 0x7FFF_F800 (half 2KB), and all packet counters are preset to 0x7FFF_FFF0 (half 16). When this bit is high and the CNTPRST bit is set to 1, all MMC counters are preset to values close to full. All eight-bit byte counters are preset to 0xFFFF_F800 (full 2KB), and all packet counters are preset to 0xFFFF_FFF0 (full 16). For 16-bit counters, the corresponding half preset values for the byte and packet counters are 0x7800 and 0x7FF0, respectively. Similarly, the almost full preset values for the 16-bit counters are 0xF800 and 0xFFF0. 0: Disable Full-Half Preset feature 1: Enable Full-Half Preset feature
4	CNTPRST	Counter Preset. When this bit is set, all counters are initialized or preset to almost full or almost half according to the CNTPRSTLVL bit. This bit is cleared automatically after 1 clock cycle. This bit, along with the CNTPRSTLVL bit, is useful for debugging and testing the assertion of interrupts because of MMC counter becoming half-full or full. 0: Disable counter preset function 1: Enable counter preset function
3	CNTFREEZ	MMC Counter Freeze. When this bit is set to 1, all MMC counters will be frozen at their current values. The counters will not be updated due to any transmitted or received data packets until this bit is reset to 0. If the RSTONRD bit is set when reading any MMC counter, the counter will also be cleared in this mode.

Bit Field	Name	Description
		0: Disable MMC counter freeze function 1: Enable MMC counter freeze function
2	RSTONRD	Reset on Read. When this bit is set to 1, the MMC counter will be reset to zero after being read (self-clearing reset). After reading the least significant bytes channel (bits [7:0]), the counter will be cleared. 0: Disable Reset on Read function 1: Enable Reset on Read function
1	CNTSTOPRO	Counter Stop Rollover. When this bit is set to 1, the counter will not rollover to zero after reaching its maximum value. 0: Disable Counter Stop Rollover function 1: Enable Counter Stop Rollover function
0	CNTRST	Counters Reset. When this bit is set to 1, all counters will be reset. The bit will automatically clear to zero after one clock cycle. 0: Counters do not reset 1: All counters reset

35.6.1.34 ETH MMC Rx Interrupt register (ETH_MMCRXINT)

Address offset: 0x0704

Reset value: 0x0000 0000

The MMC Receive Interrupt register maintains the interrupts that are generated when the following occur:

- Receive statistic counters reach half of their maximum values (0x8000_0000 for 32 bit counter and 0x8000 for 16 bit counter).
- Receive statistic counters cross their maximum values (0xFFFF_FFFF for 32 bit counter and 0xFFFF for 16 bit counter).

When the Counter Stop Rollover is set, interrupts are set but the counter remains at all-ones. The MMC Receive Interrupt register is a 32 bit register. An interrupt bit is cleared when the respective MMC counter that caused the interrupt is read. The least significant byte lane (bits [7:0]) of the respective counter must be read to clear the interrupt bit.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved														RXUCGPIS	Reserved
														r	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									RXALGNERPIS	RXCRCERPIS	RXMCGPIS	RXBCGPIS	Reserved		
									r	r	r	r			

Bit Field	Name	Description
31:18	Reserved	Reserved, must be kept at reset value.
17	RXUCGPIS	Receive Unicast Good Packet Counter Interrupt Status. When the receive unicast good packet counter reaches half of its maximum value or the maximum value, this bit is set to 1. 0: MMC receive unicast good packet counter interrupt status not detected 1: MMC receive unicast good packet counter interrupt status detected *Note: This bit has access restrictions. Read-clear. Automatically set to 1 on internal events.*
16:7	Reserved	Reserved, must be kept at reset value.
6	RXALGNERPIS	MMC Receive Alignment Error Packet Counter Interrupt Status. When the receive alignment error packet counter reaches half of its maximum value or the maximum value, this bit is set to 1. 0: MMC receive alignment error packet counter interrupt status not detected 1: MMC receive alignment error packet counter interrupt status detected *Note: This bit has access restrictions. Read-clear. Automatically set to 1 on internal events.*
5	RXCRCERPIS	MMC Receive CRC Error Packet Counter Interrupt Status. When the receive CRC error packet counter reaches half of its maximum value or the maximum value, this bit is set to 1. 0: MMC receive CRC error packet counter interrupt status not detected 1: MMC receive CRC error packet counter interrupt status detected *Note: This bit has access restrictions. Read-clear. Automatically set to 1 on internal events.*
4	RXMCGPIS	Receive Multicast Good Packet Counter Interrupt Status. When the receive multicast good packet counter reaches half of its maximum value or the maximum value, this bit is set to 1. 0: MMC receive multicast good packet counter interrupt status not detected 1: MMC receive multicast good packet counter interrupt status detected *Note: This bit has access restrictions. Read-clear. Automatically set to 1 on internal events.*
3	RXBCGPIS	Receive Broadcast Good Packet Counter Interrupt Status. When the receive broadcast good packet counter reaches half of its maximum value or the maximum value, this bit is set to 1. 0: MMC receive broadcast good packet counter interrupt status not detected 1: MMC receive broadcast good packet counter interrupt status detected *Note: This bit has access restrictions. Read-clear. Automatically set to 1 on internal events.*
2:0	Reserved	Reserved, must be kept at reset value.

35.6.1.35 ETH MMC Tx Interrupt register (ETH_MMCTXINT)

Address offset: 0x0708

Reset value: 0x0000 0000

This register maintains the interrupts generated from all Transmit statistics counters.

The MMC Transmit Interrupt register maintains the interrupts generated when transmit statistic counters reach half their maximum values (0x8000_0000 for 32 bit counter and 0x8000 for 16 bit counter), and when they cross their maximum values (0xFFFF_FFFF for 32-bit counter and 0xFFFF for 16-bit counter). When Counter Stop Rollover is set, the interrupts are set but the counter remains at all-ones. The MMC Transmit Interrupt register is a 32 bit register. An interrupt bit is cleared when the respec-tive MMC counter that caused the interrupt is read.

The least significant byte lane (bits [7:0]) of the respective counter must be read to clear the interrupt bit.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved										TXGPKTIS	Reserved				
				r	r					r					

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TXMCOLGPIS	TXSCOLGPIS	Reserved													
r	r														

Bit Field	Name	Description
31:22	Reserved	Reserved, must be kept at reset value.
21	TXGPKTIS	MMC Transmit Good Packet Counter Interrupt Status. When the transmit good packet counter reaches half of its maximum value or the maximum value, this bit is set to 1. 0: MMC transmit good packet counter interrupt status not detected 1: MMC transmit good packet counter interrupt status detected *Note: This bit has access restrictions. Read-clear. Automatically set to 1 on internal events.*
20:16	Reserved	Reserved, must be kept at reset value.
15	TXMCOLGPIS	MMC Transmit Multiple Collision Good Packet Counter Interrupt Status. When the transmit multiple collision good packet counter reaches half of its maximum value or the maximum value, this bit is set to 1. 0: MMC transmit multiple collision good packet counter interrupt status not detected 1: MMC transmit multiple collision good packet counter interrupt status detected *Note: This bit has access restrictions. Read-clear. Automatically set to 1 on internal events.*
14	TXSCOLGPIS	MMC Transmit Single Collision Good Packet Counter Interrupt Status. When the transmit single collision good packet counter reaches half of its maximum value or the maximum value, this bit is set to 1. 0: MMC transmit single collision good packet counter interrupt status not detected 1: MMC transmit single collision good packet counter interrupt status detected *Note: This bit has access restrictions. Read-clear. Automatically set to 1 on internal events.*
13:0	Reserved	Reserved, must be kept at reset value.

35.6.1.36 ETH MMC Rx Interrupt Mask register (ETH_MMCRXINTMSK)

Address offset: 0x070C

Reset value: 0x0000 0000

This register maintains the masks for interrupts generated from all Receive statistics counters.

The MMC Receive Interrupt Mask register maintains the masks for the interrupts generated when receive statistic counters reach half of their maximum value or the maximum values.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved														RXUCGPIM	Reserved
				rw	rw									rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									RXALGNERPIM	RXCRCERPIM	RXMCGPIM	RXBCGPIM	Reserved		
									rw	rw	rw	rw			

Bit Field	Name	Description
31:18	Reserved	Reserved, must be kept at reset value.
17	RXUCGPIM	Receive Unicast Good Packet Counter Interrupt Mask. Setting this bit will mask the interrupt when the receive unicast good packet counter reaches half of its maximum value or the maximum value. 0: Disable MMC receive unicast good packet counter interrupt mask 1: Enable MMC receive unicast good packet counter interrupt mask
16:7	Reserved	Reserved, must be kept at reset value.
6	RXALGNERPIM	MMC Receive Alignment Error Packet Counter Interrupt Mask. Setting this bit will mask the interrupt when the receive alignment error packet counter reaches half of its maximum value or the maximum value. 0: Disable MMC receive alignment error packet counter interrupt mask 1: Enable MMC receive alignment error packet counter interrupt mask
5	RXCRCERPIM	MMC Receive CRC Error Packet Counter Interrupt Mask. Setting this bit will mask the interrupt when the receive CRC error packet counter reaches half of its maximum value or the maximum value. 0: Disable MMC receive CRC error packet counter interrupt mask 1: Enable MMC receive CRC error packet counter interrupt mask
4	RXMCGPIM	Receive Multicast Good Packet Counter Interrupt Mask. Setting this bit will mask the interrupt when the receive multicast good packet counter reaches half of its maximum value or the maximum value. 0: Disable MMC receive multicast good packet counter interrupt mask 1: Enable MMC receive multicast good packet counter interrupt mask
3	RXBCGPIM	Receive Broadcast Good Packet Counter Interrupt Mask. Setting this bit will mask the interrupt when the receive broadcast good packet counter reaches half of its maximum value or the maximum value.

Bit Field	Name	Description
		0: Disable MMC receive broadcast good packet counter interrupt mask 1: Enable MMC receive broadcast good packet counter interrupt mask
2:0	Reserved	Reserved, must be kept at reset value.

35.6.1.37 ETH MMC Tx Interrupt Mask register (ETH_MMCTXINTMSK)

Address offset: 0x0710

Reset value: 0x0000 0000

This register maintains the masks for interrupts generated from all Transmit statistics counters.

The MMC Transmit Interrupt Mask register maintains the masks for the interrupts generated when the transmit statistic counters reach half of their maximum value or the maximum values.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved										TXGPKTIM	Reserved				
				rw	rw					rw					

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TXMCOLGPIM	TXSCOLGPIM	Reserved													
rw	rw														

Bit Field	Name	Description
31:22	Reserved	Reserved, must be kept at reset value.
21	TXGPKTIM	MMC Transmit Good Packet Counter Interrupt Mask. Setting this bit will mask the interrupt when the transmit good packet counter reaches half of its maximum value or the maximum value. 0: Disable MMC transmit good packet counter interrupt mask 1: Enable MMC transmit good packet counter interrupt mask
20:16	Reserved	Reserved, must be kept at reset value.
15	TXMCOLGPIM	MMC Transmit Multiple Collision Good Packet Counter Interrupt Mask. Setting this bit will mask the interrupt when the transmit multiple collision good packet counter reaches half of its maximum value or the maximum value. 0: Disable MMC transmit multiple collision good packet counter interrupt mask 1: Enable MMC transmit multiple collision good packet counter interrupt mask
14	TXSCOLGPIM	MMC Transmit Single Collision Good Packet Counter Interrupt Mask. Setting this bit will mask the interrupt when the transmit single collision good packet counter reaches half of its maximum value or the maximum value. 0: Disable MMC transmit single collision good packet counter interrupt mask 1: Enable MMC transmit single collision good packet counter interrupt mask
13:0	Reserved	Reserved, must be kept at reset value.

35.6.1.38 ETH Tx Single Collision Good Packets register (ETH_MMCTXSCGP)

Address offset: 0x074C

Reset value: 0x0000 0000

This register provides the number of successfully transmitted packets by the Ethernet peripheral after a single collision occurs in half-duplex mode.

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="16">TXSNGLCOLG[31:16]</td></tr>
<tr><td colspan="16">r</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="16">TXSNGLCOLG[15:0]</td></tr>
<tr><td colspan="16">r</td></tr>
</table>

Bit Field	Name	Description
31:0	TXSNGLCOLG	Tx Single Collision Good Packets. This field indicates the number of packets successfully transmitted after a single collision occurs in half-duplex mode.

35.6.1.39 ETH Tx Multiple Collision Good Packets register (ETH_MMCTXMCGP)

Address offset: 0x0750

Reset value: 0x0000 0000

This register provides the number of successfully transmitted packets by the Ethernet peripheral after multiple collisions occur in half-duplex mode.

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="16">TXMULTCOLG[31:16]</td></tr>
<tr><td colspan="16">r</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="16">TXMULTCOLG[15:0]</td></tr>
<tr><td colspan="16">r</td></tr>
</table>

Bit Field	Name	Description
31:0	TXMULTCOLG	Tx Multiple Collision Good Packets. This field indicates the number of packets successfully transmitted after multiple collisions occur in half-duplex mode.

35.6.1.40 ETH Tx Packet Count Good register (ETH_MMCTXPCG)

Address offset: 0x0768

Reset value: 0x0000 0000

This register provides the number of good packets transmitted by the Ethernet peripheral.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TXPKTG[31:16]															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TXPKTG[15:0]															
r															

Bit Field	Name	Description
31:0	TXPKTG	Tx Packet Count Good. This field indicates the number of good packets that have been transmitted.

35.6.1.41 ETH Rx Broadcast Packets Good register (ETH_MMCRXBPG)

Address offset: 0x078C

Reset value: 0x0000 0000

This register provides the number of good broadcast packets received by the Ethernet peripheral.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RXBCASTG[31:16]															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RXBCASTG[15:0]															
r															

Bit Field	Name	Description
31:0	RXBCASTG	Rx Broadcast Packets Good. This field indicates the number of good broadcast packets received.

35.6.1.42 ETH Rx Multicast Packets Good register (ETH_MMCRXMPG)

Address offset: 0x0790

Reset value: 0x0000 0000

This register provides the number of good multicast packets received

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RXMCASTG[31:16]															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RXMCASTG[15:0]															

r

Bit Field	Name	Description
31:0	RXMCASTG	Rx Multicast Packets Good. This field indicates the number of good multicast packets received.

35.6.1.43 ETH Rx CRC Error Packets register (ETH_MMCRXCRCEP)

Address offset: 0x0794

Reset value: 0x0000 0000

This register provides the number of packets with CRC errors received by the Ethernet peripheral.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RXCRCERR[31:16]															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RXCRCERR[15:0]															
r															

Bit Field	Name	Description
31:0	RXCRCERR	Rx CRC Error Packets. This field indicates the number of packets received with CRC errors.

35.6.1.44 ETH Rx Alignment Error Packets register (ETH_MMCRXAEP)

Address offset: 0x0798

Reset value: 0x0000 0000

This register provides the number of packets with alignment (dribble) errors received by the Ethernet peripheral. This counter register is only valid in 10/100 mode.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RXALGNERR[31:16]															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RXALGNERR[15:0]															
r															

Bit Field	Name	Description
31:0	RXALGNERR	Rx Alignment Error Packets.

Bit Field	Name	Description
		This field indicates the number of packets received with alignment (dribble) errors.

35.6.1.45 ETH Rx Unicast Packets Good register (ETH_MMCRXUPG)

Address offset: 0x07C4

Reset value: 0x0000 0000

This register provides the number of good unicast packets received by the Ethernet peripheral.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RXUCASTG[31:16]															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RXUCASTG[15:0]															
r															

Bit Field	Name	Description
31:0	RXUCASTG	Rx Unicast Packets Good. This field indicates the number of good unicast packets received.

35.6.1.46 ETH MAC Timestamp Control register (ETH_MACTSCTRL)

Address offset: 0x0B00

Reset value: 0x0000 2000

This register controls the operation of the system time generator and the processing of PTP packets for the receiver timestamps.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved			AV8021ASMEN	Reserved			TXTSSTSM	Reserved					TSENMACADDR	SNAPTYPSEL	
			rw				rw						rw	rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSMSTRENA	TSEVNTENA	TSIPV4ENA	TSIPV6ENA	TSIPENA	TSVER2ENA	TSCTRLSSR	TSENALL	Reserved		TSADDREG	TSTRIG	TSUPDT	TSINIT	TSCFUPDT	TSENA
rw	rw	rw	rw	rw	rw	rw	rw			rw	rw	rw	rw	rw	rw

Bit Field	Name	Description
31:29	Reserved	Reserved, must be kept at reset value.
28	AV8021ASMEN	AV 802.1AS Mode Enable. When this bit is set to 1, the MAC only processes untagged Ethernet PTP packets to provide PTP status and capture timestamp snapshots, i.e., IEEE 802.1AS operation mode.

Bit Field	Name	Description
		0: Disable AV 802.1AS Mode 1: Enable AV 802.1AS Mode
27:25	Reserved	Reserved, must be kept at reset value.
24	TXTSSTSM	Transmit Timestamp Status Mode. When this bit is set to 1, the MAC will overwrite the previous transmit timestamp status even if the software has not read it. The MAC indicates this by setting the TXTSSMIS bit in the MAC Transmit Timestamp Status Nanoseconds register. When this bit is reset, if the software has not read the timestamp status of the previous packet, the MAC will ignore the timestamp status of the current packet. The MAC indicates this by setting the TXTSSMIS bit in the MAC Transmit Timestamp Status Nanoseconds register. 0: Disable Transmit Timestamp Status Mode 1: Enable Transmit Timestamp Status Mode
23:19	Reserved	Reserved, must be kept at reset value.
18	TSENMACADDR	Enable MAC Address for PTP Packet Filtering. When this bit is set to 1, PTP packets will be filtered using the DA MAC address (matching any MAC address register) when PTP is sent directly over Ethernet. When this bit is set to 1, for PTP packets received when PTP is sent directly over Ethernet, if the DA in the received PTP packet contains a special multicast or unicast address that matches the address programmed in the MAC address register, it will be processed accordingly. For normal timestamp operations, MAC address registers 0 to 3 will be treated as unicast destination address matches. 0: Disable the function of using MAC address for PTP packet filtering 1: Enable the function of using MAC address for PTP packet filtering
17:16	SNAPTYPSEL	Select PTP packets for Taking Snapshots. These bits, along with bit 15 and bit 14, determine the set of PTP packet types for which snapshots need to be taken. The timestamp snapshot depends on the encoding provided in the register bit number table.
15	TSMSTRENA	Enable Snapshot for Messages Relevant to Master. When this bit is set to 1, snapshots will only be taken for messages relevant to the master node. Otherwise, snapshots will be taken for messages relevant to the slave node. 0: Disable snapshot function for messages relevant to the master node 1: Enable snapshot function for messages relevant to the master node
14	TSEVNTENA	Enable Timestamp Snapshot for Event Messages. When this bit is set to 1, timestamp snapshots will only be taken for event messages (SYNC, Delay_Req, Pdelay_Req, or Pdelay_Resp). When this bit is reset, snapshots will be taken for all messages except Announce, Management, and Signaling. For more information on timestamp snapshots, please refer to Table 35-19. 0: Disable timestamp snapshot function for event messages 1: Enable timestamp snapshot function for event messages
13	TSIPV4ENA	Enable Processing of PTP Packets Sent over IPv4-UDP.

Bit Field	Name	Description
		When this bit is set to 1, the MAC receiver processes PTP packets encapsulated in IPv4-UDP packets. When this bit is reset, the MAC will ignore PTP transmitted via IPv4-UDP packets. This bit is set to 1 by default. 0: Disable processing of PTP packets sent over IPv4-UDP 1: Enable processing of PTP packets sent over IPv4-UDP
12	TSIPV6ENA	Enable Processing of PTP Packets Sent over IPv6-UDP. When this bit is set to 1, the MAC receiver processes PTP packets encapsulated in IPv6-UDP packets. When this bit is reset, the MAC will ignore PTP transmitted via IPv6-UDP packets. 0: Disable processing of PTP packets sent over IPv6-UDP 1: Enable processing of PTP packets sent over IPv6-UDP
11	TSIPENA	Enable Processing of PTP over Ethernet Packets. When this bit is set to 1, the MAC receiver processes PTP packets directly encapsulated in Ethernet packets. When this bit is reset, the MAC will ignore PTP packets encapsulated in Ethernet packets. 0: Disable processing of PTP over Ethernet packets 1: Enable processing of PTP over Ethernet packets
10	TSVER2ENA	Enable PTP Packet Processing for Version 2 Format. When this bit is set to 1, PTP packets will be processed using the IEEE 1588 version 2 format. When this bit is reset, PTP packets will be processed using the IEEE 1588 version 1 format. Refer to section 35.5.6.5 for an explanation of the IEEE 1588 format. 0: Disable processing of PTP packets in version 2 format 1: Enable processing of PTP packets in version 2 format
9	TSCTRLSSR	Timestamp Digital or Binary Rollover Control. When this bit is set to 1, the timestamp low register will roll over after the value of 0x3B9A_C9FF (i.e., 1 nanosecond precision) and increment the timestamp (high) seconds. When this bit is reset, the rollover value for the sub-second register is 0x7FFF_FFFF. The sub-second increment must be programmed correctly based on the PTP reference clock frequency and the value of this bit. 0: Disable Timestamp Digital or Binary Rollover Control 1: Enable Timestamp Digital or Binary Rollover Control
8	TSENALL	Enable Timestamp for All Packets. When this bit is set to 1, timestamp snapshots will be enabled for all packets received by the MAC. 0: Disable Timestamp for All Packets 1: Enable Timestamp for All Packets
7:6	Reserved	Reserved, must be kept at reset value.
5	TSADDREG	Update Addend Register. When this bit is set to 1, the contents of the timestamp addend register will be updated to the PTP block for precise calibration. This bit will be cleared at the end of the update. This bit should be zero before being set to 1. 0: Addend register not updated

Bit Field	Name	Description
		1: Addend register updated *Note: This bit has access restrictions, writing 1 is effective, automatically cleared, writing 0 has no effect.*
4	TSTRIG	Enable Timestamp Interrupt Trigger. When this bit is set to 1, a timestamp interrupt will be generated when the system time exceeds the value written in the target time register. This bit will be reset after the timestamp interrupt trigger. 0: Disable Timestamp Interrupt Trigger 1: Enable Timestamp Interrupt Trigger *Note: This bit has access restrictions, writing 1 is effective, automatically cleared, writing 0 has no effect.*
3	TSUPDT	Update Timestamp. When this bit is set to 1, the system time will be updated (incremented or decremented) by the values specified in the MAC system time seconds update register and MAC system time nanoseconds update register. This bit should be zero before the update. After the hardware update is completed, this bit will be reset. 0: Timestamp not updated 1: Timestamp updated *Note: This bit has access restrictions, writing 1 is effective, automatically cleared, writing 0 has no effect.*
2	TSINIT	Initialize Timestamp. When this bit is set to 1, the system time will be initialized (overwritten) with the values specified in the MAC system time seconds update register and MAC system time nanoseconds update register. This bit should be zero before the initialization. After the initialization is completed, this bit will be reset. 0: Timestamp not initialized 1: Timestamp initialized *Note: This bit has access restrictions, writing 1 is effective, automatically cleared, writing 0 has no effect.*
1	TSCFUPDT	Fine or Coarse Timestamp Update. 0: Update system timestamp using coarse method 1: Update system timestamp using fine method
0	TSENA	Enable Timestamp. When this bit is set to 1, timestamps will be added to sent and received packets. When this bit is reset, timestamps will not be added to sent and received packets, and the timestamp generator will also be paused. After enabling this mode, the timestamp (system time) needs to be initialized. On the receive side, the MAC only processes 1588 packets when this bit is set to 1. 0: Disable timestamp function 1: Enable timestamp function

35.6.1.47 ETH MAC Sub Second Increment register (ETH_MACSUBSINC)

Address offset: 0x0B04

Reset value: 0x0000 0000

This register specifies the value added to the internal system time register for each clk_ptp_ref_i clock cycle.

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">Reserved</td><td colspan="8">SSINC[7:0]</td></tr>
<tr><td colspan="8"></td><td colspan="8">rw</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="16">Reserved</td></tr>
</table>

Bit Field	Name	Description
31:24	Reserved	Reserved, must be kept at reset value.
23:16	SSINC	Sub-second Increment Value. The value programmed in this field will be accumulated with the content of the sub-second register for each clock cycle of PTP. For example, when the PTP clock is 50MHz (period is 20ns), and the precision of the system time nanoseconds register is 1ns (TSCTRLSSR set to 1 in the MAC Timestamp Control register), it should be programmed as 20 (0x14). When TSCTRLSSR is cleared, the resolution of the nanoseconds register is ~0.465ns. In this case, the programmed value should be 43 (0x2B), derived from 20ns/0.465.
15:0	Reserved	Reserved, must be kept at reset value.

35.6.1.48 ETH MAC System Time Seconds register (ETH_MACSYSTS)

Address offset: 0x0B08

Reset value: 0x0000 0000

The System Time Seconds register and System Time Nanoseconds register indicate the current value of the system time maintained by the MAC. While the system time is continuously updated, there is some delay in the actual time due to clock domain transfer latency (from clk_ptp_ref_i to the CSR clock).

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="16">TSS[31:16]</td></tr>
<tr><td colspan="16">r</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="16">TSS[15:0]</td></tr>
<tr><td colspan="16">r</td></tr>
</table>

Bit Field	Name	Description
31:0	TSS	Timestamp Second. The value in this field indicates the current value in seconds of the System

Bit Field	Name	Description
		Time maintained by the MAC.

35.6.1.49 ETH MAC System Time Nanoseconds register (ETH_MACSYSTNS)

Address offset: 0x0B0C

Reset value: 0x0000 0000

The system time nanoseconds register and system time seconds register indicate the current value of the system time maintained by the MAC.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TSS[31:16]															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSS[15:0]															
r															

Bit Field	Name	Description
31:0	TSS	Timestamp Sub Seconds. The value in this field has the sub-second representation of time, with an accuracy of 0.46 ns. When Bit 9 is set in MAC Timestamp Control register, each bit represents 1 ns. The max value is 0x3B9A_C9FF after which it rolls-over to zero.

35.6.1.50 ETH MAC System Time Seconds Update register (ETH_MACSYSTSUP)

Address offset: 0x0B10

Reset value: 0x0000 0000

The system time seconds update register, along with the system time nanoseconds update register, is used for initializing or updating the system time maintained by the MAC. Before setting the second bit (TSINIT) or third bit (TSUPDT) in the MAC Timestamp Control register, both of these registers (system time seconds update register and system time nanoseconds update register) must be written to.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TSS[31:16]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSS[15:0]															
rw															

Bit Field	Name	Description
31:0	TSS	Timestamp Seconds.

Bit Field	Name	Description
		The value in this field is the seconds part of the update. When ADDSUB is reset, this field must be programmed with the seconds part of the update value. When ADDSUB is set, this field must be programmed with the complement of the seconds part of the update value. For example, to subtract 2.000000001 seconds from the system time, the TSS field in the MAC Timestamp Seconds Update register must be 0xFFFF_FFFE (that is, 2^32 - 2).

35.6.1.51 ETH MAC System Time Nanoseconds Update register (ETH_MACSYSTNSUP)

Address offset: 0x0B14

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADDSUB	TSSS[16:30]														
rw	rw														

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSSS[15:0]															
rw															

Bit Field	Name	Description
31	ADDSUB	Add or Subtract Time. When this bit is set to 1, the time value will be subtracted from the content of the update register. When this bit is reset, the time value will be added to the content of the update register. 0: Add Time 1: Subtract Time
30:0	TSSS	Timestamp Sub-seconds. The value in this field represents the sub-second part of the update value. ADDSUB is 1: The field must be programmed with the two's complement of the sub-second part of the update value as described. ADDSUB is 0: The field must be programmed with the sub-second part of the update value, with the precision based on the TSCTRLSSR bit in the MAC Timestamp Control register. For TSCTRLSSR bit field as 1: The programmed value must be 10^9 - <sub-second value>; each bit represents 1ns, and the programmed value should not exceed 0x3B9A_C9FF. For TSCTRLSSR bit field as 0: The programmed value must be 2^31 - <sub-second value>: each bit represents a precision of 0.46ns. For example, to subtract 2.000000001 seconds from the system time, the TSSS field in the MAC System Time Nanoseconds Update Register must be 0x7FFF_FFFF (i.e.,

Bit Field	Name	Description
		2^31 - 1) when the TSCTRLSSR bit is 0 in the MAC Timestamp Control Register, and must be 0x3B9A_C9FF (i.e., 10^9 - 1) when the TSCTRLSSR bit is 1 in the MAC Timestamp Control Register.

35.6.1.52 ETH MAC Timestamp Addend register (ETH_MACTSADD)

Address offset: 0x0B18

Reset value: 0x0000 0000

Only when the system time is configured in fine update mode (by setting the TSCFUPDT bit in the MAC Timestamp Control register), will the value of this register be used. The content of this register is added to a 32-bit accumulator every clock cycle (clk_ptp_ref_i), and when the accumulator overflows, the system time will be updated.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TSAR[31:16]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSAR[15:0]															
rw															

Bit Field	Name	Description
31:0	TSAR	Timestamp Addend Register. This field indicates the 32-bit time value to be added to the Accumulator register to achieve time synchronization.

35.6.1.53 ETH MAC Timestamp Status register (ETH_MACTSSTS)

Address offset: 0x0B20

Reset value: 0x0000 0000

When the application reads this register, all bits except for bits [27:25] will be cleared to zero.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
				r			r						r		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TXTSSIS	Reserved											TSTRGTERR0	Reserved	TSTARGT0	TSSOVF
r												r		r	r

Bit Field	Name	Description
31:16	Reserved	Reserved, must be kept at reset value.

Bit Field	Name	Description
15	TXTSSIS	Tx Timestamp Status Interrupt Status. If the discard transmit status is enabled in the MTL, this bit will be set to 1 when the captured transmit timestamp is updated in the MAC Transmit Timestamp Status Nanoseconds register and MAC Transmit Timestamp Status Seconds register. Clear this bit when reading the MAC Transmit Timestamp Status Seconds register (or when writing to the MAC Transmit Timestamp Status Seconds register by setting the RCWE bit in the MAC CSR Software Control Register). 0: Tx Timestamp Status Interrupt status not detected 1: Tx Timestamp Status Interrupt status detected
14:4	Reserved	Reserved, must be kept at reset value.
3	TSTRGTERR0	Timestamp Target Time Error. This bit is set to 1 after the latest target time programmed in the MAC PPS0 Target Time Seconds register and MAC PPS0 Target Time Nanoseconds register has elapsed. When the application reads this bit, it is cleared. 0: Timestamp Target Time Error status not detected 1: Timestamp Target Time Error status detected *Note: This bit has access restrictions. Read (or write 1 when the RCWE bit in the MAC CSR Software Control Register is set) to clear. Automatically set to 1 on internal events.*
2	Reserved	Reserved, must be kept at reset value.
1	TSTARGT0	Timestamp Target Time Reached. When this bit is set to 1, it indicates that the system time value is greater than or equal to the value specified in the MAC PPS0 Target Time Seconds register and MAC PPS0 Target Time Nanoseconds register. 0: Timestamp Target Time Reached status not detected 1: Timestamp Target Time Reached status detected *Note: This bit has access restrictions. Read (or write 1 when the RCWE bit in the MAC CSR Software Control Register is set) to clear. Automatically set to 1 on internal events.*
0	TSSOVF	Timestamp Seconds Overflow. When this bit is set to 1, it indicates that the seconds value of the timestamp has exceeded 0xFFFF_FFFF (when supporting version 2 format). 0: Timestamp Seconds Overflow status not detected 1: Timestamp Seconds Overflow status detected *Note: This bit has access restrictions. Read (or write 1 when the RCWE bit in the MAC CSR Software Control Register is set) to clear. Automatically set to 1 on internal events.*

35.6.1.54 ETH MAC Tx Timestamp Status Nanoseconds register (ETH_MACTXTSSTSNS)

Address offset: 0x0B30

Reset value: 0x0000 0000

This register contains the nanosecond part of the timestamp of the transmitted packet captured when the transmit status is disabled.

The MAC Transmit Timestamp Status Nanoseconds Register, together with the MAC Transmit Timestamp Status Seconds Register, provides a 64-bit timestamp captured when the MAC successfully transmits a PTP packet. When reading the last byte of the MAC Transmit Timestamp Status Nanoseconds Register, it is assumed that the application has read this value. In little-endian mode, this means reading bits [31:24]; in big-endian mode, it means reading bits [7:0].

If the application does not read these registers and captures a timestamp of another packet, the current timestamp will be lost (overwritten) or the new timestamp will be lost (discarded), depending on the setting of the TXTSSTSM bit in the MAC Timestamp Control Register. When the MAC transmitter captures a timestamp, the status bit TXTSSIS (bit [15]) in the MAC Timestamp Status Register will be set.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TXTSSMIS	TXTSSLO[16:30]														
rw	r														

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TXTSSLO[15:0]															
r															

Bit Field	Name	Description
31	TXTSSMIS	Transmit Timestamp Status Missed. When this bit is set to 1, it indicates one of the following: – If the TXTSSTSM bit in the MAC Timestamp Control Register is 0, the timestamp of the current packet is ignored. – If the TXTSSTSM bit in the MAC Timestamp Control Register is 1, the timestamp of the previous packet is overwritten by the timestamp of the current packet. 0: Transmit Timestamp Status Missed status not detected 1: Transmit Timestamp Status Missed status detected *Note: This bit has access restrictions. Read-clear. Automatically set to 1 on internal events.*
30:0	TXTSSLO	Transmit Timestamp Status Low. This field contains the 31 bits of the Nanoseconds field of the Transmit packet's captured timestamp

35.6.1.55 ETH MAC Tx Timestamp Status Seconds register (ETH_MACTXTSSTSS)

Address offset: 0x0B34

Reset value: 0x0000 0000

The register contains the higher 32 bits of the timestamp (in seconds) captured when a PTP packet is transmitted.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

TXTSSHI[31:16]															
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TXTSSHI[15:0]															
r															

Bit Field	Name	Description
31:0	TXTSSHI	Transmit Timestamp Status High. This field contains the lower 32 bits of the Seconds field of Transmit packet's captured timestamp.

35.6.1.56 ETH MAC Timestamp Ingress Corr Nanosecond register (ETH_MACTSIGCNS)

Address offset: 0x0B58

Reset value: 0x0000 0000

This register contains the correction value in nanoseconds to be used with the captured timestamp value in the ingress path.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TSIC[31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSIC[15:0]															
rw															

Bit Field	Name	Description
31:0	TSIC	Timestamp Ingress Correction. This field contains the ingress path correction value as defined by the Ingress Correction expression.

35.6.1.57 ETH MAC Timestamp Egress Corr Nanosecond register (ETH_MACTSEGCNS)

Address offset: 0x0B5C

Reset value: 0x0000 0000

This register contains the correction value in nanoseconds to be used with the captured timestamp value in the egress path.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TSEC[31:16]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSEC[15:0]															
rw															

Bit Field	Name	Description
31:0	TSEC	Timestamp Egress Correction. This field contains the nanoseconds part of the egress path correction value as defined by the Egress Correction expression.

35.6.1.58 ETH MAC Timestamp Ingress Latency register (ETH_MACTSIGLAT)

Address offset: 0x0B68

Reset value: 0x0000 0000

This register holds the Ingress MAC latency.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				ITLNS											
				r											

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ITLSNS								Reserved							
r															

Bit Field	Name	Description
31:28	Reserved	Reserved, must be kept at reset value.
27:16	ITLNS	Ingress Timestamp Latency, in nanoseconds. This register stores the average delay (in nanoseconds) between the MAC input port (TXD) and the actual point of ingress timestamp capture (MII). The calculation method for the ingress correction value is as described in section 35.5.6.4.
15:8	ITLSNS	Ingress Timestamp Latency, in sub-nanoseconds. This register stores the average delay (in sub-nanoseconds) between the MAC input port (TXD) and the actual point of ingress timestamp capture (MII). The calculation method for the ingress correction value is as described in section 35.5.6.4.
7:0	Reserved	Reserved, must be kept at reset value.

35.6.1.59 ETH MAC Timestamp Egress Latency register (ETH_MACTSEGLAT)

Address offset: 0x0B6C

Reset value: 0x0000 0000

This register holds the Egress MAC latency.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

Reserved	ETLNS
	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETLSNS								Reserved							
r															

Bit Field	Name	Description
31:28	Reserved	Reserved, must be kept at reset value.
27:16	ETLNS	Egress Timestamp Latency, in nanoseconds. This register stores the average delay (in nanoseconds) between the MAC output port (RXD) and the actual point of egress timestamp capture (MII). The calculation method for the egress correction value is as described in section 35.5.6.4.
15:8	ETLSNS	Egress Timestamp Latency, in sub-nanoseconds. This register stores the average delay (in sub-nanoseconds) between the MAC output port (RXD) and the actual point of egress timestamp capture (MII). The calculation method for the egress correction value is as described in section 35.5.6.4.
7:0	Reserved	Reserved, must be kept at reset value.

35.6.1.60 ETH MAC PPS Control register (ETH_MACPPSCTRL)

Address offset: 0x0B70

Reset value: 0x0000 0000

PPS Control register (PPS).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												PPSCTRL			
												rw			

Bit Field	Name	Description
31:4	Reserved	Reserved, must be kept at reset value.
3:0	PPSCTRL	**PPSCTRL:** PPS Output Frequency Control. This field controls the frequency of the PPS output signal (ptp_pps_o). The default value for PPSCTRL is 0000, resulting in PPS output of one pulse per second (width equal to clk_ptp_i). For other values of PPSCTRL, the PPS output will become a clock generated at the following frequencies: 0001: Binary flip for 2Hz, Digital flip for 1Hz

Bit Field	Name	Description
		0010: Binary flip for 4Hz, Digital flip for 2Hz 0011: Binary flip for 8Hz, Digital flip for 4Hz 0100: Binary flip for 16Hz, Digital flip for 8Hz ... 1111: Binary flip for 32.768KHz, Digital flip for 16.384KHz *Note: In binary flip mode, the duty cycle of the PPS output (ptp_pps_o) is 50% for these frequencies. In digital flip mode, the PPS output frequency is the average value. The actual clock frequency is not the same, synchronized once per second.* *For example:* – *When PPSCTRL = 0001, the low level period of PPS (1Hz) is 537ms, and the high level period is 463ms.* – *When PPSCTRL = 0010, PPS (2Hz) is a sequence: a clock with a 50% duty cycle and a period of 537ms; the second clock has a period of 463ms (low level 268ms, high level 195ms).* – *When PPSCTRL = 0011, PPS (4Hz) is a sequence: three clocks with a 50% duty cycle and a period of 268ms; the fourth clock has a period of 195ms (low level 134ms, high level 61ms).* *This occurs due to non-linear switching of the bit in the digital flip mode of the MAC system time nanoseconds register.*

35.6.1.61 ETH MAC PPS Target Time Seconds register (ETH_MACPPSTTS)

Address offset: 0x0B80

Reset value: 0x0000 0000

The PPS Target Time Seconds register, along with PPS Target Time Nanoseconds register, is used to schedule an interrupt event [bit 1 of MAC Timestamp Status register] when the system time exceeds the value programmed in these registers.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TSTRH0[31:16]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSTRH0[15:0]															
rw															

Bit Field	Name	Description
31:0	TSTRH0	PPS Target Time Seconds Register This field stores the time in seconds. When the timestamp value matches or exceeds the two target timestamp registers, the MAC will start or stop the PPS signal output, and generate an interrupt based on the target time mode chosen for the respective PPS output in the MAC PPS Control Register (if interrupts are enabled).

35.6.1.62 ETH MAC PPS Target Time Nanoseconds register (ETH_MACPPSTTNS)

Address offset: 0x0B84

Reset value: 0x0000 0000

PPS Target Time Nanoseconds register.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved	TTSL0[30:16]														
	rw														

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TTSL0[15:0]															
rw															

Bit Field	**Name**	**Description**
31	Reserved	Reserved, must be kept at reset value.
30:0	TTSL0	Target Time Low for PPS Register. This register stores the time in nanoseconds (signed). When the value of the timestamp matches the values in the two target timestamp registers, the MAC will start or stop the PPS signal output. When the TSCTRLSSR bit in the MAC Timestamp Control Register is reset, this value should be the time in ns ÷ 0.465. There may be an error of up to one sub-nanosecond increment in the actual start or stop time of the PPS signal output. When the TSCTRLSSR bit in the MAC Timestamp Control Register is set to 1, this value should not exceed 0x3B9A_C9FF. There may be an error of up to one sub-nanosecond increment in the actual start or stop time of the PPS signal output.

35.6.2 ETH MTL Registers

35.6.2.1 ETH MTL Operation Mode register (ETH_MTLOPMOD)

Address offset: 0x0C00

Reset value: 0x0000 0000

The Operation Mode register establishes the Transmit and Receive operating modes and commands.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						CNTCLR	CNTPRST	Reserved						DTXSTS	Reserved
						rw	rw							rw	

Bit Field	Name	Description
31:10	Reserved	Reserved, must be kept at reset value.
9	CNTCLR	Counters Reset. When this bit is set to 1, all counters will be reset. The bit will automatically clear to zero after one clock cycle. If this bit and the CNTPRST bit are set to 1 at the same time, CNTPRST takes priority. 0: Counters do not reset 1: All counters reset *Note: This bit has access restrictions, writing 1 is effective, automatically cleared, writing 0 has no effect.*
8	CNTPRST	Counter Preset. When this bit is set: – The MTL send queue underflow register is initialized/preset to 0x7F0. – The MTL receive queue lost and overflow packet count registers for lost and overflow packets are initialized/preset to 0x7F0. 0: Disable counter preset function 1: Enable counter preset function *Note: This bit has access restrictions, writing 1 is effective, it will automatically clear, writing 0 has no effect.*
7:2	Reserved	Reserved, must be kept at reset value.
1	DTXSTS	Drop Transmit Status. When this bit is set, the Tx packet status received from the MAC is discarded in the MTL. When this bit is reset, the Tx packet status received from the MAC will be forwarded to the application. 0: Disable the function of dropping transmit status

Bit Field	Name	Description
		1: Enable the function of dropping transmit status
0	Reserved	Reserved, must be kept at reset value.

35.6.2.2 ETH MTL Interrupt Status register (ETH_MTLINTSTS)

Address offset: 0x0C20

Reset value: 0x0000 0000

The software driver (application) reads this register during interrupt service routine or polling to determine the interrupt status of MTL queues and the MAC.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															Q0IS
															rw

Bit Field	Name	Description
31:1	Reserved	Reserved, must be kept at reset value.
0	Q0IS	Queue Interrupt Status. This bit indicates an interrupt from the queue. To reset this bit, the application must read the MTL queue interrupt control status register to determine the exact reason for the interrupt and clear its source. 0: No queue interrupt status detected 1: Queue interrupt status detected

35.6.2.3 ETH MTL TxQ0 Operation Mode register (ETH_MTLTXQOPMOD)

Address offset: 0x0D00

Reset value: 0x0000 0000

The Queue 0 Transmit Operation Mode register establishes the Transmit queue operating modes and commands

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									TTC			Reserved		TSF	FTQ
									rw					rw	rw

Bit Field	Name	Description
31:7	Reserved	Reserved, must be kept at reset value.
6:4	TTC	Transmit Threshold Control. This Bit Field controls the threshold level of the MTL Tx queue. When the size of a packet in the MTL Tx queue exceeds the threshold, transmission is initiated. Additionally, complete packets with a length smaller than the threshold will also be transmitted. These bits can only be used after the TSF bit is reset. 000 (M_32BYTES): 32 001 (M_64BYTES): 64 010 (M_96BYTES): 96 011 (M_128BYTES): 128 100 (M_192BYTES): 192 101 (M_256BYTES): 256 110 (M_384BYTES): 384 111 (M_512BYTES): 512
3:2	Reserved	Reserved, must be kept at reset value.
1	TSF	Transmit Store and Forward. When this bit is set, the transmission starts when a full packet resides in the MTL Tx queue. When this bit is set, the TTC values specified in Bit s[6:4] of this register are ignored. This bit should be changed only when the transmission is stopped. 0: Disable Tx store and forward function 1: Enable Tx store and forward function
0	FTQ	Flush Transmit Queue. When this bit is set, the Tx queue controller logic is reset to its default values. Therefore, all data in the Tx queue will be lost or flushed. This bit is internally reset when the flushing operation is complete. Until this bit is reset, you should not write to the MTL transmit queue operating mode register. The data which is already accepted by the MAC transmitter is not flushed. It is scheduled for transmission and results in underflow and runt packet transmission. 0: Disable the function of flushing the transmit queue 1: Enable the function of flushing the transmit queue *Note: The flush operation is considered complete only when the Tx queue is empty, and the application has acknowledged the pause on Tx status for all transmitted packets. To complete this flush operation, the PHY Tx clock should be in a valid state.* *Note: This bit has access restrictions, writing 1 is effective, it will automatically clear, writing 0 has no effect.*

35.6.2.4 ETH MTL TxQ0 Underflow register (ETH_MTLTXQUDF)

Address offset: 0x0D04

Reset value: 0x0000 0000

The Queue 0 Underflow Counter register contains the counter for packets aborted because of Transmit queue underflow and packets missed because of Receive queue packet flush

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				UFCNTOVF	UFFRMCNT										
				r	r										

Bit Field	Name	Description
31:12	Reserved	Reserved, must be kept at reset value.
11	UFCNTOVF	Overflow Bit for Underflow Packet Counter. Whenever the Tx queue underflow packet counter field overflows (i.e., exceeds the maximum count), this bit is set to 1. In this case, the overflow packet counter will be reset to zero, indicating a flip has occurred. 0: No overflow detected for the underflow packet counter field 1: Overflow detected for the underflow packet counter field *Note: This bit has access restrictions. Read-clear. Automatically set to 1 during internal events.*
10:0	UFFRMCNT	Underflow Packet Counter. This field indicates the number of packets that the controller has aborted due to underflow in the Tx queue. Each time the MAC aborts the transmission of a packet due to underflow, this counter is incremented. When reading this register while data is valid on the data line, the counter is cleared. *Note: This Bit Field has access restrictions. Read-clear. Automatically set to 1 during internal events.*

35.6.2.5 ETH MTL TxQ0 Debug register (ETH_MTLTXQDBG)

Address offset: 0x0D08

Reset value: 0x0000 0000

The Queue 0 Transmit Debug register gives the debug status of various blocks related to the Transmit queue.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved									STXSTSF			Reserved	PTXQ		
									r				r		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										TXSTSFSTS	TXQSTS	TWCSTS	TRCSTS		TXQPAUSED
										r	r	r	r		r

Bit Field	Name	Description
31:23	Reserved	Reserved, must be kept at reset value.
22:20	STXSTSF	Number of Status Words in Tx Status FIFO of Queue. This field indicates the current number of status in the Tx Status FIFO of this queue. When the DTXSTS bit of MTL Operation Mode register is set to 1, this field does not reflect the number of status words in Tx Status FIFO.
19	Reserved	Reserved, must be kept at reset value.
18:16	PTXQ	Number of Packets in the Transmit Queue. This field indicates the current number of data packets in the Tx queue. When the DTXSTS bit in the MTL operating mode register is set to 1, this field does not reflect the number of packets in the transmit queue.
15:6	Reserved	Reserved, must be kept at reset value.
5	TXSTSFSTS	MTL Tx Status FIFO Full Status. When high, this bit indicates that the MTL Tx Status FIFO is full. Therefore, the MTL cannot accept any more packets for transmission. 0: No MTL Tx status FIFO full status detected 1: MTL Tx status FIFO full status detected
4	TXQSTS	MTL Tx Queue Not Empty Status. When this bit is high, it indicates that the MTL Tx Queue is not empty and some data is left for transmission. 0: No MTL Tx queue not empty status detected 1: MTL Tx queue not empty status detected
3	TWCSTS	MTL Tx Queue Write Controller Status. When high, this bit indicates that the MTL Tx Queue Write Controller is active, and it is transferring the data to the Tx Queue. 0: No MTL Tx Queue Write Controller status detected 1: MTL Tx Queue Write Controller status detected
2:1	TRCSTS	MTL Tx Queue Read Controller Status. This field indicates the status of the Tx queue read controller: 00 (IDLE): Idle state 01 (READ): Read state (transferring data to the MAC transmitter) 10 (WAIT): Waiting for pause on Tx status from the MAC transmitter 11 (FLUSH): Flushing the Tx queue due to a packet abort request from the MAC
0	TXQPAUSED	Transmit Queue in Pause. When this bit is high and Rx flow control is enabled, it indicates that the Tx queue is in a paused state for the following reasons (only in full-duplex mode): − When PFC is enabled, receiving a PFC data packet assigned to the Tx queue − When PFC is disabled, receiving an 802.3x pause data packet 0: Transmit queue not in pause state detected 1: Transmit queue in pause state detected

35.6.2.6 ETH MTL Q0 Interrupt Control Status register (ETH_MTLQINTCTRLSTS)

Address offset: 0x0D2C

Reset value: 0x0000 0000

This register contains the interrupt enable and status bits for the queue 0 interrupts.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved							RXOIE	Reserved							RXOVFIS
							rw								rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved							TXUIE	Reserved							TXUNFIS
							rw								rw

Bit Field	Name	Description
31:25	Reserved	Reserved, must be kept at reset value.
24	RXOIE	Receive Queue Overflow Interrupt Enable. 0: Disable receive queue overflow interrupt 1: Enable receive queue overflow interrupt
23:17	Reserved	Reserved, must be kept at reset value.
16	RXOVFIS	Receive Queue Overflow Interrupt Status. This bit indicates an overflow in the receive queue during packet reception. If some packets have been transferred to the application, the overflow status in RDES3 [21] is set to 1. Writing a 1 to this bit by the application will clear it. 0: No receive queue overflow interrupt status detected 1: Receive queue overflow interrupt status detected *Note: This bit has access restrictions. Automatically set to 1 during internal events. Writing 1 clears it. Writing 0 has no effect.*
15:9	Reserved	Reserved, must be kept at reset value.
8	TXUIE	Transmit Queue Underflow Interrupt Enable. 0: Disable transmit queue underflow interrupt 1: Enable transmit queue underflow interrupt
7:1	Reserved	Reserved, must be kept at reset value.
0	TXUNFIS	Transmit Queue Underflow Interrupt Status. This bit indicates an underflow in the transmit queue during packet transmission. The transmission is paused, and the underflow error is set (TDES3 [2] bit is set). Writing a 1 to this bit by the application will clear it. 0: No transmit queue underflow interrupt status detected 1: Transmit queue underflow interrupt status detected *Note: This bit has access restrictions. Automatically set to 1 during internal events. Writing 1 clears it. Writing 0 has no effect.*

35.6.2.7 ETH MTL RxQ0 Operation Mode register (ETH_MTLRXQOPMOD)

Address offset: 0x0D30

Reset value: 0x0000 0000

The Queue 0 Receive Operation Mode register establishes the Receive queue operating modes and command.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									DISTCPEF	RSF	FEP	FUP	Reserved	RTC	
									rw	rw	rw	rw		rw	

Bit Field	Name	Description
31:7	Reserved	Reserved, must be kept at reset value.
6	DISTCPEF	Disable Dropping of TCP/IP Checksum Error Packets. When this bit is set, the MAC does not drop the packets which only have the errors detected by the Receive Checksum Offload engine. Such packets have errors only in the encapsulated payload. There are no errors (including FCS error) in the Ethernet packet received by the MAC. When this bit is reset, all error packets are dropped if the FEP bit is reset. 0: Enable the function of dropping TCP/IP checksum error packets 1: Disable the function of dropping TCP/IP checksum error packets
5	RSF	Receive Queue Store and Forward. When this bit is set, the ETH reads a packet from the Rx queue only after the complete packet has been written to it, ignoring the RTC field of this register. When this bit is reset, the Rx queue operates in the Threshold (cut-through) mode, subject to the threshold specified by the RTC field of this register. 0: Disable the function of Receive Queue Store and Forward 1: Enable the function of Receive Queue Store and Forward
4	FEP	Forward Error Packets. When this bit is reset, the Rx queue will discard packets with error statuses (CRC error, receive error, watchdog timeout, or overflow). However, if the start bytes of a packet (write) pointer have been transferred to the read controller end (in threshold mode), the packet will not be discarded. When this bit is set, all packets except the runt error packets are forwarded to the application or DMA. If the RSF bit is set and the Rx queue overflows when a partial packet is written, the packet is dropped irrespective of the setting of this bit. However, if the RSF bit is reset and the Rx queue overflows when a partial packet is written, a partial packet might be forwarded to the application or DMA. 0: Disable the function of forwarding error packets 1: Enable the function of forwarding error packets
3	FUP	Forward Undersized Good Packets. When this bit is set, the Rx queue forwards the undersized good packets (packets with no error and length less than 64 bytes), including pad-bytes and CRC. When this bit is

Bit Field	Name	Description
		reset, the Rx queue drops all packets of less than 64 bytes, unless a packet is already transferred because of the lower value of Rx Threshold, for example, RTC = 01. 0: Disable the function of forwarding undersized good packets 1: Enable the function of forwarding undersized good packets
2	Reserved	Reserved, must be kept at reset value.
1:0	RTC	Receive Queue Threshold Control. This field controls the threshold level of the MTL Rx queue in bytes. When the size of a packet in the MTL Rx queue exceeds the threshold, the received packet will be transferred to the application or DMA. Additionally, complete packets with a length smaller than the threshold will also be automatically transferred. This field is only effective when the RSF bit is zero. When the RSF bit is set to 1, this field will be ignored. 00 (M_64BYTE): 64 01 (M_32BYTE): 32 10 (M_96BYTE): 96 11 (M_128BYTE): 128

35.6.2.8 ETH MTL RxQ0 Missed Packet Overflow Count register (ETH_MTLRXQMPOFCNT)

Address offset: 0x0D34

Reset value: 0x0000 0000

The Queue 0 Missed Packet and Overflow Counter register contains the counter for packets missed because of Receive queue packet flush and packets discarded because of Receive queue overflow.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				MISCNTOVF	MISPKTCNT										
				r	r										

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				OVFCNTOVF	OVFPKTCNT										
				r	r										

Bit Field	Name	Description
31:28	Reserved	Reserved, must be kept at reset value.
27	MISCNTOVF	Missed Packet Counter Overflow Bit. When set, this bit indicates that the Rx Queue Missed Packet Counter crossed the maximum limit. 0: No missed packet counter overflow detected 1: Missed packet counter overflow detected *Note: This bit has access restrictions. Read-clear. Automatically set to 1 during internal events.*
26:16	MISPKTCNT	Missed Packet Counter.

Bit Field	Name	Description
		This field indicates the number of packets that the Ethernet peripheral has missed due to a packet flush request by the application for that queue. This counter is reset when a read operation is performed on this register. The counter is incremented when the DMA discards packets due to unavailable buffers. *Note: This Bit Field has access restrictions. Read-clear. Automatically set to 1 during internal events.*
15:12	Reserved	Reserved, must be kept at reset value.
11	OVFCNTOVF	Overflow Counter Overflow Bit. When set, this bit indicates that the Rx Queue Overflow Packet Counter field crossed the maximum limit. 0: No overflow counter overflow detected 1: Overflow counter overflow detected *Note: This bit has access restrictions. Read-clear. Automatically set to 1 during internal events.*
10:0	OVFPKTCNT	Overflow Packet Counter. This field indicates the number of packets that the Ethernet peripheral has discarded due to an overflow in the receive queue. Each time the Ethernet peripheral discards an incoming packet due to overflow, the value of this counter is incremented. This counter is reset when a read operation is performed on this register. *Note: This Bit Field has access restrictions. Read-clear. Automatically set to 1 during internal events.*

35.6.2.9 ETH MTL RxQ0 Debug register (ETH_MTLRXQDBG)

Address offset: 0x0D38

Reset value: 0x0000 0000

The Queue 0 Receive Debug register gives the debug status of various blocks related to the Receive queue.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		PRXQ													
		r													

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										RXQSTS		Reserved	RRCSTS		RWCSTS
										r			r		r

Bit Field	Name	Description
31:30	Reserved	Reserved, must be kept at reset value.
29:16	PRXQ	Number of Packets in Receive Queue. This field indicates the current number of packets in the Rx Queue. The theoretical maximum value for this field is 256KB/16B = 16K Packets, that is, Max_Queue_Size/Min_Packet_Size.

Bit Field	Name	Description
15:6	Reserved	Reserved, must be kept at reset value.
5:4	RXQSTS	MTL Rx Queue Fill-Level Status. This field indicates the status of the fill level of the Rx queue: 00 (EMPTY): Rx queue is empty 01 (BLW_THR): Rx queue fill level is below the flow control deactivate threshold 10 (ABV_THR): Rx queue fill level is above the flow control activate threshold 11 (FULL): Rx queue is full
3	Reserved	Reserved, must be kept at reset value.
2:1	RRCSTS	MTL Rx Queue Read Controller State. This field indicates the state of the Rx queue read controller: 00 (IDLE): Idle state 01 (READ_DATA): Reading packet data 10 (READ_STS): Reading packet status (or timestamp) 11 (FLUSH): Flushing packet data and status
0	RWCSTS	MTL Rx Queue Write Controller Active Status When high, this bit indicates that the MTL Rx queue Write controller is active, and it is transferring a received packet to the Rx Queue. 0: No MTL Rx Queue Write Controller active status detected 1: MTL Rx Queue Write Controller active status detected

35.6.3 ETH DMA Registers

35.6.3.1 ETH DMA Mode register (ETH_DMAMODE)

Address offset: 0x1000

Reset value: 0x0000 0000

The Bus Mode register establishes the bus operating modes for the DMA.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved														INTM	
														rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	PR			TXPR	Reserved									DA	SWR
	rw			rw										rw	rw

Bit Field	Name	Description
31:18	Reserved	Reserved, must be kept at reset value.
17:16	INTM	Interrupt Mode This field defines the interrupt mode of the Ethernet peripheral. The behavior of the following output signals will change based on the settings: ■ sbd_perch_tx_intr_o (Transmit Channel Interrupt) ■ sbd_perch_rx_intr_o (Receive Channel Interrupt) ■ sbd_intr_o (General Interrupt) This field will also alter the behavior of the RI/TI bits in the DMA Channel 0 Status Register. 00: sbd_perch_xxx is a pulse signal used to enable the IOC bit in the Description word for each Tx/Rx data packet transfer completion event (regardless of whether the corresponding interrupt is enabled). When the corresponding interrupt is enabled, sbd_intr_o will also be set to active, and sbd_intr_o will only be cleared when the software clears the corresponding RI/TI status bit. 01: sbd_perch_xxx is a level signal that is set to active when the corresponding interrupt is enabled, upon the occurrence of TX/RX data packet transfer completion events, and is set to inactive when the software clears the corresponding RI/TI status bit. sbd_intr_o will not be set to active during these TX/RX data packet transfer completion events. 10: sbd_perch_xxx is a level signal that is set to active when the corresponding interrupt is enabled, upon the occurrence of TX/RX data packet transfer completion events, and is set to inactive when the software clears the corresponding RI/TI status bit. However, if the same event occurs again before being cleared, the signal will be set to active once more. sbd_intr_o will not be set to active during these TX/RX data packet transfer completion events.

Bit Field	Name	Description
		11: Reserved For more details, refer to Table 35-55 Transmit Complete Interrupt Behavior.
15	Reserved	Reserved, must be kept at reset value.
14:12	PR	Priority Ratio. This Bit Field controls the priority ratio for weighted round-robin arbitration between Rx DMA and Tx DMA. These bits are only effective when the DA bit is reset. The priority ratio is either Rx or Tx, depending on whether the TXPR bit is reset or set. 000 (R_1_1): Priority ratio is 1:1 001 (R_2_1): Priority ratio is 2:1 010 (R_3_1): Priority ratio is 3:1 011 (R_4_1): Priority ratio is 4:1 100 (R_5_1): Priority ratio is 5:1 101 (R_6_1): Priority ratio is 6:1 110 (R_7_1): Priority ratio is 7:1 111 (R_8_1): Priority ratio is 8:1
11	TXPR	Transmit Priority. When set, this bit indicates that the Tx DMA has higher priority than the Rx DMA during arbitration for the system-side bus.
10:2	Reserved	Reserved, must be kept at reset value.
1	DA	DMA Tx or Rx Arbitration Scheme. This bit specifies the arbitration scheme between the transmit and receive paths for all channels: 0: Weighted round-robin arbitration between Rx and Tx paths, depending on the priority specified in bits [14:12], with the priority weight specified in the TXPR bit. 1: Fixed priority. When the TXPR bit is set to 1, the priority of the Tx path is higher than the Rx path. Otherwise, the priority of the Rx path is higher than the Tx path.
0	SWR	Software Reset. When this bit is set, the MAC and the DMA controller reset the logic and all internal registers of the DMA, MTL, and MAC. This bit is automatically cleared after the reset operation is complete in all ETH clock domains. Before reprogramming any ETH register, a value of zero should be read in this bit. This bit must be read at least 4 CSR clock cycles after it is written to 1. *Note: The reset operation is complete only when all resets in all active clock domains are de-asserted. Therefore, it is essential that all PHY inputs clocks (applicable for the selected PHY interface) are present for software reset completion. The time to complete the software reset operation depends on the frequency of the slowest active clock.* 0: Disable software reset 1: Enable software reset

35.6.3.2 ETH DMA SysBus Mode register (ETH_DMASBMODE)

Address offset: 0x1004

Reset value: 0x0000 0000

The System Bus mode register controls the behavior of the AHB or AXI master. It mainly controls burst splitting and number of outstanding requests.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RB	MB	Reserved	AAL	Reserved											FB
rw	rw		rw												rw

Bit Field	Name	Description
31:16	Reserved	Reserved, must be kept at reset value.
15	RB	Rebuild INCRx Burst. When this bit is set high and the AHB master gets SPLIT, RETRY, or Early Burst Termination (EBT) response, the AHB master interface rebuilds the pending beats of any initiated burst transfer with INCRx and SINGLE transfers. By default, the AHB master interface rebuilds the pending beats of an EBT with an unspecified (INCR) burst. 0: Disable Rebuild INCRx Burst functionality 1: Enable Rebuild INCRx Burst functionality
14	MB	Mixed Burst. When this bit is high and the FB bit is low, the AHB master performs undefined bursts transfers (INCR) for burst length of 16 or more. For burst length of 16 or less, the AHB master performs fixed burst transfers (INCRx and SINGLE). 0: Disable Mixed Burst functionality 1: Enable Mixed Burst functionality
13	Reserved	Reserved, must be kept at reset value.
12	AAL	Address-Aligned Beats. When this bit is set, the master performs address-aligned burst transfers on Read and Write channels. When this bit is set to 0, the master performs burst transfers on Read and Write channels without aligning to address boundaries. 0: Disable Address-Aligned Beats functionality 1: Enable Address-Aligned Beats functionality
11:1	Reserved	Reserved, must be kept at reset value.
0	FB	Fixed Burst Length. When this bit is set, the AHB master device will initiate burst transfers of a specific length (INCRx or SINGLE). When this bit is reset, the AHB master device will initiate transfers of non-specific length (INCR) or SINGLE transfers. 0: Disable Fixed Burst Length functionality 1: Enable Fixed Burst Length functionality

35.6.3.3 ETH DMA Interrupt Status register (ETH_DMAINTSTS)

Address offset: 0x1008

Reset value: 0x0000 0000

The application reads this Interrupt Status register during interrupt service routine or polling to determine the interrupt status of DMA channels, MTL queues, and the MAC.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved														MACIS	MTLIS
														r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															DC0IS
															r

Bit Field	Name	Description
31:18	Reserved	Reserved, must be kept at reset value.
17	MACIS	MAC Interrupt Status. This bit indicates an interrupt event in the MAC. To reset this bit to 1'b0, software must read the corresponding register in the MAC to determine the exact cause of the interrupt and clear its source. 0: No MAC interrupt status detected 1: MAC interrupt status detected
16	MTLIS	MTL Interrupt Status. This bit indicates an interrupt event in the MTL. To reset this bit to 1'b0, software must read the corresponding register in the MTL to determine the exact cause of the interrupt and clear its source. 0: No MTL interrupt status detected 1: MTL interrupt status detected
15:1	Reserved	Reserved, must be kept at reset value.
0	DC0IS	DMA Channel 0 Interrupt Status. This bit indicates an interrupt event in DMA Channel 0. To reset this bit to 1'b0, software must read the corresponding register in DMA Channel 0 to determine the exact cause of the interrupt and clear its source. 0: No DMA Channel 0 interrupt status detected 1: DMA Channel 0 interrupt status detected

35.6.3.4 ETH DMA Debug Status register (ETH_DMADBGSTS)

Address offset: 0x100C

Reset value: 0x0000 0000

The Debug Status 0 register gives the Receive and Transmit process status for DMA Channel 0 for debugging purpose.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPS0				RPS0				Reserved							AHBMS
r				r											r

Bit Field	Name	Description
31:16	Reserved	Reserved, must be kept at reset value.
15:12	TPS0	DMA Channel 0 Transmit Process State. This field indicates the Tx DMA FSM state of Channel 0. The MSB of this field always returns 0. This field does not generate interrupts. 0000 (STOP): Stopped (issued a reset or stop transmit command) 0001 (RUN_FTTD): Running (fetching Tx transmit Description) 0010 (RUN_WS): Running (waiting for status) 0011 (RUN_RDS): Running (reading data from system memory buffer and placing it in the Tx buffer (Tx FIFO) queue) 0100 (TSTMP_WS): Timestamp write state 0101 (RSVD): Reserved 0110 (SUSPND): Suspended (Tx Description not available or Tx buffer underflow) 0111 (RUN_CTD): Running (closing Tx Description)
11:8	RPS0	DMA Channel 0 Receive Process State. This field indicates the Rx DMA FSM state of Channel 0. The MSB of this field always returns 0. This field does not generate interrupts. 0000 (STOP): Stopped (issued a reset or stop receive command) 0001 (RUN_FRTD): Running (fetching Rx receive Description) 0010 (RSVD): Reserved 0011 (RUN_WRP): Running (waiting for Rx data packet) 0100 (SUSPND): Suspended (Rx Description not available) 0101 (RUN_CRD): Running (closing Rx Description) 0110 (TSTMP): Timestamp write state 0111 (RUN_TRP): Running (transferring received data packet data from Rx buffer to system memory)
7:1	Reserved	Reserved, must be kept at reset value.
0	AHBMS	AHB Master Status. When high, this bit indicates that the AHB master FSMs are in the non-idle state. 0: No AHB Master status detected 1: AHB Master status detected

35.6.3.5 ETH DMA CH0 Control register (ETH_DMACH0CTRL)

Address offset: 0x1100

Reset value: 0x0000 0000

This register specifies the length to skip between two descriptors, as well as functions such as 8xPBL mode.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved											DSL			Reserved	PBLx8
											rw				rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															

Bit Field	Name	Description
31:21	Reserved	Reserved, must be kept at reset value.
20:18	DSL	Descriptor Skip Length. This bit specifies the Word number (32-bit bus) to skip between two unchained descriptors. The address skipping starts from the end of the current descriptor to the start of the next descriptor. When the DSL value is equal to zero, the DMA takes the descriptor table as contiguous.
17	Reserved	Reserved, must be kept at reset value.
16	PBLx8	8xPBL mode. When this bit is set, the PBL value programmed in Bits[21:16] in DMA CH0 Tx Control and Bits[21:16] in DMA CH0 Rx Control is multiplied by eight times. Therefore, the DMA transfers the data in 8, 16, 32, 64, 128, and 256 beats depending on the PBL value.
15:0	Reserved	Reserved, must be kept at reset value.

35.6.3.6 ETH DMA CH0 Tx Control register (ETH_DMACH0TXCTRL)

Address offset: 0x1104

Reset value: 0x0000 0000

This register controls the Tx features such as PBL.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved									ETIC	TxPBL					
									rw	rw					

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved											OSF	Reserved			ST
											rw				rw

Bit Field	Name	Description
31:23	Reserved	Reserved, must be kept at reset value.

Bit Field	Name	Description
22	ETIC	Early Transmit Interrupt Control. When this bit is set, Early Transmit Interrupt (ETI) status is set after completion of transfer of data from buffers of a transmit descriptor in which IOC bit (TDES2[31]) is set. When this bit is reset, ETI is set only after a complete packet is transferred to the MTL TX FIFO memory.
21:16	TxPBL	Transmit Programmable Burst Length. This Bit Field indicates the maximum number of beats to be transferred during a single DMA data transfer process. This value represents the maximum number used in a single block read or block write operation. Each time the DMA initiates a burst transfer on the application bus, it always attempts to burst according to the PBL specified. The following values can be programmed for PBL: 1, 2, 4, 8, 16, or 32. Any other value will result in undefined behavior. To transfer more than 32 beats, follow these steps: 1 Set the PBLx8 mode in the DMA Channel 0 Control Register. 2 Set the TxPBL.
15:5	Reserved	Reserved, must be kept at reset value.
4	OSF	Operate on Second Packet. When this bit is set, it instructs the DMA to process the second packet of the Transmit data even before the status for the first packet is obtained. 0: Disable the function of operating on the second packet 1: Enable the function of operating on the second packet
3:1	Reserved	Reserved, must be kept at reset value.
0	ST	Start or Stop Transmission Command. When this bit is set, transmission is placed in the Running state. The DMA checks the Transmit list at the current position for a packet to be transmitted. The DMA tries to acquire descriptor from either of the following positions: ■ The current position in the list: determined by the transmit list base address set in the DMA Channel 0 Transmit list address register. ■ The last stopped position. If the DMA does not possess the current Description descriptor, the transmission process will enter a pause state, and the TBU bit in the DMA Channel 0 Status register will be set. The start transmission command is only effective when the transmission is stopped. If this command is issued before setting the transmit list base address register for DMA Channel 0, the DMA behavior will be unpredictable. When this bit is reset, the transmission process will enter a "stop" state after completing the transmission of the current data packet. The position of the next Description descriptor in the transmit list will be saved and become the current position when transmission resumes. To change the list address, a new value must be programmed into the DMA Channel 0 Transmit list address register when this bit is reset. The new value will be adopted when this bit is set again. The stop transmission command is only effective when the current data packet has been sent or the transmission process is in a pause state. 0: Stop transmission command

Bit Field	Name	Description
		1: Start transmission command

35.6.3.7 ETH DMA CH0 Rx Control register (ETH_DMACH0RXCTRL)

Address offset: 0x1108

Reset value: 0x0000 0000

This register controls the Rx features such as PBL, buffer size, and extended status.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RPF	Reserved								ERIC	RxPBL					
rw									rw	rw					

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	RBSZH												RBSZL		SR
	rw												r		rw

Bit Field	Name	Description
31	RPF	Rx Packet Flush. When this bit is set to 1, the Ethernet core will automatically flushes the packet from the Rx Queues destined to this DMA Rx Channel, when it is stopped. When this bit remains set and the DMA is re-started by the software driver, the packets residing in the Rx Queues that were received when this Rx DMA was stopped, are flushed out. The packets that are received by the MAC after the Rx DMA is re-started are routed to the Rx DMA. The flushing is done on the Read side of the Rx Queue. When this bit is reset, the Ethernet core will not flush the data packets in the Rx queue destined for this Rx DMA channel when the Rx DMA channel is in the STOP state. This may result in head blocking in the corresponding Rx queue. 0: Disable Rx Packet Flush functionality 1: Enable Rx Packet Flush functionality
30:23	Reserved	Reserved, must be kept at reset value.
22	ERIC	Early Receive Interrupt Control. When this bit is set, Early Receive Interrupt (ERI) status is set after the completion of every burst transfer of data from the Rx DMA to the buffer. When this bit is reset, ERI is set only after a complete buffer is filled up by the Rx DMA.
21:16	RxPBL	Receive Programmable Burst Length. This Bit Field indicates the maximum number of beats to be transferred during a single DMA data transfer process. This value represents the maximum number used in a single block read or block write operation. Each time the DMA initiates a burst transfer on the application bus, it always attempts to burst according to the PBL specified. The following values can be programmed for PBL: 1, 2, 4, 8, 16, or 32. Any other value will result in undefined behavior. To transfer more than 32 beats, follow these steps: 1 Set the PBLx8 mode in the DMA Channel 0 Control Register.

Bit Field	Name	Description
		2 Set the RxPBL.
15	Reserved	Reserved, must be kept at reset value.
14:3	RBSZH	Receive Buffer Size High. The RBSZ [13:0] is divided into two fields: RBSZH and RBSZL. The RBSZ [13:0] field represents the size of the Rx buffer in bytes. The maximum buffer size is limited to 16Kbytes. This buffer size is applicable for the payload buffer when split headers are enabled. *Note: The buffer size must be a multiple of 4, depending on the data bus width (32 bits). Even if the value of the buffer address pointer is not aligned with the data bus width, it must be so. Therefore, the RBSZL is read-only, and its value is considered as all zeros. Thus, RBSZH represents the size of the buffer in terms of bytes (aligned with the bus width).*
2:1	RBSZL	Receive Buffer Size Low. The RBSZ [13:0] is divided into two fields: RBSZH and RBSZL. RBSZL is the low field, and this field is read-only (RO).
0	SR	Start or Stop Receive. When this bit is set, the DMA tries to acquire the descriptor from the Receive list and processes the incoming packets. The DMA tries to acquire descriptor from either of the following positions: ■ The current position in the list: determined by the address set in the DMA Channel 0 Receive Description list address register. ■ The last stopped Rx process position. If the DMA does not possess the current Description descriptor, it will pause the reception, and set the RBU bit in the DMA Channel 0 Status register to 1. The start receive command is only effective when the reception is stopped. If this command is issued before setting the DMA Channel 0 Receive Description list address register, the DMA behavior will be unpredictable. When this bit is reset, the Rx DMA operation will stop after completing the current data packet transmission. The position of the next Description descriptor in the receive list will be saved and become the current position when the Rx process is restarted. The stop receive command is only effective when the Rx process is running (waiting for Rx data packets) or in a pause state. 0: Stop receive 1: Start receive

35.6.3.8 ETH DMA CH0 TxDesc List Address register (ETH_DMACH0TXDLA)

Address offset: 0x1114

Reset value: 0x0000 0000

This register points the DMA to the start of Transmit descriptor list. The descriptor lists reside in the physical memory space of the application and must be Word-aligned (for 32-bit data bus). The DMA internally converts it to bus width aligned address by making the corresponding LSB to low.

You can write to this register only when the Tx DMA has stopped, that is, the ST bit is set to zero in DMA CH0 Tx

Control register. When stopped, this register can be written with a new descriptor list address. When you set the ST bit to 1, the DMA takes the newly-programmed descriptor base address. If this register is not changed when the ST bit is set to 0, the DMA takes the descriptor address where it was stopped earlier.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TDESLA[31:16]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TDESLA[15:2]														TDESLA[1:0]	
rw														ro	

Bit Field	Name	Description
31:0	TDESLA	Start of Transmit List. This field contains the base address of the first descriptor in the Transmit descriptor list. The DMA ignores the LSB bits (1:0) for 32-bit bus width and internally takes these bits as all-zero. Therefore, these LSB bits are read-only (RO).

35.6.3.9 ETH DMA CH0 RxDesc List Address register (ETH_DMACH0RXDLA)

Address offset: 0x111C

Reset value: 0x0000 0000

This register points the DMA to the start of Receive descriptor list.

This register points to the start of the Receive Descriptor List. The descriptor lists reside in the physical memory space of the application and must be Word-aligned (for 32-bit data bus). The DMA internally converts it to bus width aligned address by making the corresponding LS bits low. Writing to this register is permitted only when reception is stopped. When stopped, this register must be written to before the receive Start command is given.

You can write to this register only when Rx DMA has stopped, that is, SR bit is set to zero in DMA CH0 Rx Control register. When stopped, this register can be written with a new descriptor list address. When you set the SR bit to 1, the DMA takes the newly programmed descriptor base address.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RDESLA[31:16]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RDESLA[15:2]														RDESLA[1:0]	
rw														ro	

Bit Field	Name	Description
31:0	RDESLA	Start of Receive List. This field contains the base address of the first descriptor in the Rx Descriptor list. The DMA ignores the LSB bits (1:0) for 32-bit bus width and internally takes these bits as all-zero. Therefore, these LSB bits are read-only (RO).

35.6.3.10 ETH DMA CH0 TxDesc Tail Pointer register (ETH_DMACH0TXDTP)

Address offset: 0x1120

Reset value: 0x0000 0000

This pointer points to an offset from the base and indicates the location of the last valid descriptor.

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="16">TDTP[31:16]</td></tr>
<tr><td colspan="16">rw</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="14">TDTP[15:2]</td><td colspan="2">TDTP[1:0]</td></tr>
<tr><td colspan="14">rw</td><td colspan="2">ro</td></tr>
</table>

Bit Field	Name	Description
31:0	TDTP	Transmit Descriptor Tail Pointer. This field contains the tail pointer of the Tx Descriptor ring. Software writes to the tail pointer to add more Description descriptors to the Tx channel. The hardware attempts to transmit all data packets referenced by the Description descriptors between the head pointer and the tail pointer registers. *Note: The LSB bits (1:0) are read-only (RO).*

35.6.3.11 ETH DMA CH0 RxDesc Tail Pointer register (ETH_DMACH0RXDTP)

Address offset: 0x1128

Reset value: 0x0000 0000

This pointer points to an offset from the base and indicates the location of the last valid descriptor

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="16">RDTP[31:16]</td></tr>
<tr><td colspan="16">rw</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="14">RDTP[15:2]</td><td colspan="2">RDTP[1:0]</td></tr>
<tr><td colspan="14">rw</td><td colspan="2">ro</td></tr>
</table>

Bit Field	Name	Description
31:0	RDTP	Receive Descriptor Tail Pointer. This field contains the tail pointer of the Rx Descriptor ring. Software writes to the tail pointer to add more Description descriptors to the Rx channel. The hardware attempts to process all data packets referenced by the Description descriptors between the head pointer and the tail pointer registers. *Note: The LSB bits (1:0) are read-only (RO).*

35.6.3.12 ETH DMA CH0 TxDesc Ring Length register (ETH_DMACH0TXDRLEN)

Address offset: 0x112C

Reset value: 0x0000 0000

The Tx Descriptor Ring Length register contains the length of the Transmit descriptor ring.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						TDRL									
						rw									

Bit Field	Name	Description
31:10	Reserved	Reserved, must be kept at reset value.
9:0	TDRL	Transmit Descriptor Ring Length. This field sets the maximum number of Tx descriptors in the circular descriptor ring. The maximum number of descriptors is limited to 1K descriptors. Synopsys recommends a minimum ring descriptor length of 4. For example, You can program any value up to 0x3FF in this field. This field is 10 bits wide, if you program 0x3FF, you can have 1024 descriptors. If you want to have 10 descriptors, program it to a value of 0x9.

35.6.3.13 ETH DMA CH0 Rx Control register 2 (ETH_DMACH0RXCTRL2)

Address offset: 0x1130

Reset value: 0x0000 0000

This register controls the Rx features such as Rx Descriptor Ring Length and Alternate Rx Buffer Size.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								ARBS							
								rw							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						RDRL									

rw

Bit Field	Name	Description
31:24	Reserved	Reserved, must be kept at reset value.
23:16	ARBS	Alternate Receive Buffer Size. Indicates size in bytes for Buffer 1 when ARBS is programmed to a non-zero value (when split header feature is not enabled). When split header feature is enabled, ARBS indicates the buffer size for header data. The maximum alternate buffer is limited to 1020-bytes depending on the data bus widths (32-bit bus widths). When ARBS = 0, Rx Buffer1 and Rx Buffer2 sizes are based on RBSZ field of DMA CH0 Rx Control register.
15:10	Reserved	Reserved, must be kept at reset value.
9:0	RDRL	Receive Descriptor Ring Length. This register sets the maximum number of Rx descriptors in the circular descriptor ring. The maximum number of descriptors is limited to 1K descriptors. For example, You can program any value up to 0x3FF in this field. This field is 10 bits wide, if you program 0x3FF, you can have 1024 descriptors. If you want to have 10 descriptors, program it to a value of 0x9.

35.6.3.14 ETH DMA CH0 Interrupt Enable register (ETH_DMACH0INTEN)

Address offset: 0x1134

Reset value: 0x0000 0000

This register enables the interrupts reported by the Status register.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NIE	AIE	CDEE	FBEE	ERIE	ETIE	RWTE	RSE	RBUE	RIE	Reserved			TBUE	TXSE	TIE
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw				rw	rw	rw

Bit Field	Name	Description
31:16	Reserved	Reserved, must be kept at reset value.
15	NIE	Normal Interrupt Summary Enable. When this bit is set, the normal interrupt summary is enabled. This bit enables the following interrupts in the DMA Channel 0 Status register: bit[0]: Transmit interrupt bit[2]: Transmit buffer unavailable bit[6]: Receive interrupt bit[11]: Early receive interrupt

Bit Field	Name	Description
		When this bit is reset, the normal interrupt summary will be disabled. 0: Disable normal interrupt summary 1: Enable normal interrupt summary
14	AIE	Abnormal Interrupt Summary Enable. When this bit is set to 1, it enables the abnormal interrupt summary. This bit can enable the following interrupts in the DMA Channel 0 status register: bit[1]: Send process stopped bit[7]: Receive buffer unavailable bit[8]: Receive process stopped bit[9]: Receive watchdog timeout bit[10]: Early send interrupt bit[12]: Fatal bus error bit[13]: Context descriptor error When this bit is reset, it disables the abnormal interrupt summary. 0: Disable abnormal interrupt summary 1: Enable abnormal interrupt summary
13	CDEE	Context Descriptor Error Enable. When this bit is set to 1 along with the AIE bit, the context descriptor error interrupt is enabled. When this bit is reset, the context descriptor error interrupt is disabled. 0: Disable context descriptor error interrupt 1: Enable context descriptor error interrupt
12	FBEE	Fatal Bus Error Enable. When this bit is set to 1 along with the AIE bit, the fatal bus error interrupt is enabled. When this bit is reset, the fatal bus error interrupt is disabled. 0: Disable fatal bus error interrupt 1: Enable fatal bus error interrupt
11	ERIE	Early Receive Interrupt Enable. When this bit is set to 1 along with the NIE bit, the early receive interrupt is enabled. When this bit is reset, the early receive interrupt is disabled. 0: Disable early receive interrupt 1: Enable early receive interrupt
10	ETIE	Early Transmit Interrupt Enable. When this bit is set to 1 along with the AIE bit, the early transmit interrupt is enabled. When this bit is reset, the early transmit interrupt is disabled. 0: Disable early transmit interrupt 1: Enable early transmit interrupt
9	RWTE	Receive Watchdog Timeout Enable. When this bit is set to 1 along with the AIE bit, the receive watchdog timeout interrupt is enabled. When this bit is reset, the receive watchdog timeout interrupt is disabled. 0: Disable receive watchdog timeout interrupt 1: Enable receive watchdog timeout interrupt
8	RSE	Receive Stopped Enable.

Bit Field	Name	Description
		When this bit is set to 1 along with the AIE bit, the receive stopped interrupt is enabled. When this bit is reset, the receive stopped interrupt is disabled. 0: Disable receive stopped interrupt 1: Enable receive stopped interrupt
7	RBUE	Receive Buffer Unavailable Enable. When this bit is set to 1 along with the AIE bit, the receive buffer unavailable interrupt is enabled. When this bit is reset, the receive buffer unavailable interrupt is disabled. 0: Disable receive buffer unavailable interrupt 1: Enable receive buffer unavailable interrupt
6	RIE	Receive Interrupt Enable. When this bit is set to 1 along with the NIE bit, the receive interrupt is enabled. When this bit is reset, the receive interrupt is disabled. 0: Disable receive interrupt 1: Enable receive interrupt
5:3	Reserved	Reserved, must be kept at reset value.
2	TBUE	Transmit Buffer Unavailable Enable. When this bit is set to 1 along with the NIE bit, the transmit buffer unavailable interrupt is enabled. When this bit is reset, the transmit buffer unavailable interrupt is disabled. 0: Disable transmit buffer unavailable interrupt 1: Enable transmit buffer unavailable interrupt
1	TXSE	Transmit Stopped Enable. When this bit is set to 1 along with the AIE bit, the transmit stopped interrupt is enabled. When this bit is reset, the transmit stopped interrupt is disabled. 0: Disable transmit stopped interrupt 1: Enable transmit stopped interrupt
0	TIE	Transmit Interrupt Enable. When this bit is set to 1 along with the NIE bit, the transmit interrupt is enabled. When this bit is reset, the transmit interrupt is disabled. 0: Disable transmit interrupt 1: Enable transmit interrupt

35.6.3.15 ETH DMA CH0 Rx Interrupt Watchdog Timer register (ETH_DMACH0RXINTWT)

Address offset: 0x1138

Reset value: 0x0000 0000

The Receive Interrupt Watchdog Timer register indicates the watchdog timeout for Receive Interrupt (RI) from the DMA. When this register is written with a non-zero value, it enables the watchdog timer for the RI bit of the DMA CH0 Status register.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved														RWTU	
														rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								RWT							
								rw							

Bit Field	Name	Description
31:18	Reserved	Reserved, must be kept at reset value.
17:16	RWTU	Receive Interrupt Watchdog Timer Count Units. This field indicates the number of system clock cycles corresponding to one unit in the RWT field. 00: 256 01: 512 10: 1024 11: 2048 For example, when RWT = 2 and RWTU = 1, the watchdog timer is set to a period of 2 x 512 = 1024 system clock cycles.
15:8	Reserved	Reserved, must be kept at reset value.
7:0	RWT	Receive Interrupt Watchdog Timer Count. This field represents the number of system clock cycles that the watchdog timer is set for (multiplied by the factor indicated in the RWTU field). When the Rx DMA completes the transfer of a data packet for which the RI bit is not set to 1 in the DMA Channel 0 status register, the watchdog timer will trigger at the programmed value because the interrupt enable bit in the corresponding descriptor, RDES3[30], has been set to 1. When the watchdog timer count expires, the RI bit is set, and the timer stops. When the RI bit is set to a high level, the watchdog timer is reset because any received data packet will automatically set the RI based on the interrupt enable bit RDES3[30].

35.6.3.16 ETH DMA CH0 Current App TxDesc register (ETH_DMACH0CATXD)

Address offset: 0x1144

Reset value: 0x0000 0000

This register points to the current Transmit descriptor read by the DMA.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CURTDESAPTR[31:16]															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CURTDESAPTR[15:0]															
r															

Bit Field	Name	Description
31:0	CURTDESAPTR	Application Transmit Descriptor Address Pointer. The DMA updates this pointer during Tx operation. This pointer is cleared on reset.

35.6.3.17 ETH DMA CH0 Current App RxDesc register (ETH_DMACH0CARXD)

Address offset: 0x114C

Reset value: 0x0000 0000

This register points to the current Receive descriptor read by the DMA.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CURRDESAPTR[31:16]															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CURRDESAPTR[15:0]															
r															

Bit Field	Name	Description
31:0	CURRDESAPTR	Application Receive Descriptor Address Pointer. The DMA updates this pointer during Rx operation. This pointer is cleared on reset.

35.6.3.18 ETH DMA CH0 Current App TxBuffer register (ETH_DMACH0CATXB)

Address offset: 0x1154

Reset value: 0x0000 0000

This register points to the current Tx buffer address read by the DMA.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CURTBUFAPTR[31:16]															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CURTBUFAPTR[15:0]															
r															

Bit Field	Name	Description
31:0	CURTBUFAPTR	Application Transmit Buffer Address Pointer. The DMA updates this pointer during Tx operation. This pointer is cleared on reset.

35.6.3.19 ETH DMA CH0 Current App RxBuffer register (ETH_DMACH0CARXB)

Address offset: 0x115C

Reset value: 0x0000 0000

This register points to the current Rx buffer address read by the DMA.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CURRBUFAPTR[31:16]															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CURRBUFAPTR[15:0]															
r															

Bit Field	Name	Description
31:0	CURRBUFAPTR	Application Receive Buffer Address Pointer. The DMA updates this pointer during Rx operation. This pointer is cleared on reset.

35.6.3.20 ETH DMA CH0 Status register (ETH_DMACH0STS)

Address offset: 0x1160

Reset value: 0x0000 0000

The software driver (application) reads the Status register during interrupt service routine or polling to determine the status of the DMA.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved										REB			TEB		
											r			r	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NIS	AIS	CDE	FBE	ERI	ETI	RWT	RPS	RBU	RI	Reserved			TBU	TPS	TI
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw				rw	rw	rw

Bit Field	Name	Description
31:22	Reserved	Reserved, must be kept at reset value.
21:19	REB	Rx DMA Error Bits. This field indicates the type of error that caused the bus error, such as an error response on the AHB interface. ■ bit[21]: 1: Error occurred during Rx DMA data transfer, 0: No error occurred during Rx DMA data transfer ■ bit[20]: 1: Error occurred while accessing the descriptor, 0: Error occurred while accessing the data buffer ■ bit[19]: 1: Error occurred during read transfer, 0: Error occurred during write transfer This field is only valid when the FBE bit is set to 1. It does not generate interrupts.

Bit Field	Name	Description
18:16	TEB	Tx DMA Error Bits. This field indicates the type of error that caused the bus error, such as an error response on the AHB interface. ■ bit[18]: 1: Error occurred during Tx DMA data transfer, 0: No error occurred during Tx DMA data transfer ■ bit[17]: 1: Error occurred while accessing the descriptor, 0: Error occurred while accessing the data buffer ■ bit[16]: 1: Error occurred during read transfer, 0: Error occurred during write transfer This field is only valid when the FBE bit is set to 1. It does not generate interrupts.
15	NIS	Normal Interrupt Summary. When the corresponding interrupt bits in the DMA Channel 0 interrupt enable register are enabled, the value of the Normal Interrupt Summary bit is the logical OR of the following bits: ■ bit[0]: Transmit Interrupt ■ bit[2]: Transmit Buffer Unavailable ■ bit[6]: Receive Interrupt ■ bit[11]: Early Receive Interrupt Only unmasked bits (interrupts enabled in the DMA Channel 0 interrupt enable register) will affect the Normal Interrupt Summary bit. Each time a bit that causes NIS to be set to 1 is cleared, that bit must also be cleared (write a 1 to that bit). 0: Normal Interrupt Summary status not detected 1: Normal Interrupt Summary status detected *Note: This bit has access restrictions, automatically set to 1 for internal events, writing a 1 clears it, writing a 0 has no effect.*
14	AIS	Abnormal Interrupt Summary. When the corresponding interrupt bits in the DMA Channel 0 interrupt enable register are enabled, the value of the Abnormal Interrupt Summary bit is the logical OR of the following bits: ■ bit[1]: Send Process Stopped ■ bit[7]: Receive Buffer Unavailable ■ bit[8]: Receive Process Stopped ■ bit[9]: Receive Watchdog Timeout ■ bit[10]: Early Transmit Interrupt ■ bit[12]: Fatal Bus Error ■ bit[13]: Context Descriptor Error Only unmasked bits (interrupts enabled in the DMA Channel 0 interrupt enable register) will affect the Abnormal Interrupt Summary bit. Each time a bit that causes the NIS to be set to 1 is cleared, that bit must also be cleared (write a 1 to that bit). 0: Abnormal Interrupt Summary status not detected 1: Abnormal Interrupt Summary status detected

Bit Field	Name	Description
		Note: This bit has access restrictions, automatically set to 1 for internal events, writing a 1 clears it, writing a 0 has no effect.
13	CDE	Context Descriptor Error. This bit indicates that the DMA Tx/Rx engine has received a Context Descriptor Error, which signifies an invalid context (intermediate Descriptor) appearing in the middle of a data packet flow, or in the case of transmission, it indicates that all Descriptors are invalid. On the receive side, it indicates that the buffer address of the Descriptors read by the DMA is all ones, which is often considered invalid in many cases. 0: Context Descriptor Error status not detected 1: Context Descriptor Error status detected *Note: This bit has access restrictions, automatically set to 1 for internal events, writing a 1 clears it, writing a 0 has no effect.*
12	FBE	Fatal Bus Error. This bit indicates that a bus error has occurred (as described in the TEB/REB fields). When this bit is set to 1, the corresponding DMA channel engine will disable all bus accesses. 0: Fatal Bus Error status not detected 1: Fatal Bus Error status detected *Note: This bit has access restrictions, automatically set to 1 for internal events, writing a 1 clears it, writing a 0 has no effect.*
11	ERI	Early Receive Interrupt. When this bit is set to 1, it indicates that the Rx DMA has completed the transfer of data packets to memory. ERIC = 0: This bit is set only when the Rx DMA completely fills the receive buffer with data packet data. ERIC = 1: This bit is set every time data is burst-transferred from the Rx DMA to the buffer. Setting the RI bit will automatically clear this bit. 0: Early Receive Interrupt status not detected 1: Early Receive Interrupt status detected *Note: This bit has access restrictions, automatically set to 1 for internal events, writing a 1 clears it, writing a 0 has no effect.*
10	ETI	Early Transmit Interrupt. When this bit is set to 1, it indicates that the Tx DMA has completed the transfer of data packet data to the MTL Tx FIFO memory. ETIC = 0: This bit is set only when the Tx DMA has completed the transfer of a complete data packet to the MTL. ETIC = 1: This bit is set when the buffer in the transmit Descriptor with IOC = 1 completes the (partial) data packet transfer. 0: Early Transmit Interrupt status not detected 1: Early Transmit Interrupt status detected

Bit Field	Name	Description
		Note: This bit has access restrictions, automatically set to 1 for internal events, writing a 1 clears it, writing a 0 has no effect.
9	RWT	Receive Watchdog Timeout. This bit is set to 1 when a data packet longer than 2048 bytes (or 10240 bytes when jumbo frame mode is enabled) is received. 0: Receive Watchdog Timeout status not detected 1: Receive Watchdog Timeout status detected *Note: This bit has access restrictions, automatically set to 1 for internal events, writing a 1 clears it, writing a 0 has no effect.*
8	RPS	Receive Process Stopped. This bit is set to 1 when the receive process enters a stopped state. 0: Receive Process Stopped status not detected 1: Receive Process Stopped status detected *Note: This bit has access restrictions, automatically set to 1 for internal events, writing a 1 clears it, writing a 0 has no effect.*
7	RBU	Receive Buffer Unavailable. This bit indicates that the DMA is unable to access the next Descriptor in the receive list that the application owns. The Rx process is paused (suspended). To resume processing Rx Descriptors, the application should change the ownership of the Descriptor and issue a receive polling demand command. If this command is not issued, the Rx process will resume upon receiving the next identifiable incoming data packet. In loopback mode, the application should advance the receive Descriptor tail pointer register of the channel. This bit is set to 1 only when the DMA owns the previous Rx Descriptor. 0: Receive Buffer Unavailable status not detected 1: Receive Buffer Unavailable status detected *Note: This bit has access restrictions, automatically set to 1 for internal events, writing a 1 clears it, writing a 0 has no effect.*
6	RI	Receive Interrupt. This bit indicates that the data packet reception has been completed. After the completion of data packet reception, the 31st bit of RDES3 in the last Descriptor will be reset, and specific packet status information will be updated in the Descriptor. Reception is still in progress. 0: Receive Interrupt status not detected 1: Receive Interrupt status detected *Note: This bit has access restrictions, automatically set to 1 for internal events, writing a 1 clears it, writing a 0 has no effect.*
5:3	Reserved	Reserved, must be kept at reset value.
2	TBU	Transmit Buffer Unavailable. This bit indicates that the DMA is unable to access the next Descriptor in the transmit list that the application owns, causing the Tx process to pause (suspend). The TPS0

Bit Field	Name	Description
		field in the DMA debug status register explains the state transitions of the transmit process. To resume processing transmit Descriptors, the application should: ■ Change the ownership of the Descriptor by setting the 31st bit of TDES3. ■ Issue a transmit polling demand command. In loopback mode, the application should advance the transmit Descriptor tail pointer register of the channel. 0: Transmit Buffer Unavailable status not detected 1: Transmit Buffer Unavailable status detected *Note: This bit has access restrictions, automatically set to 1 for internal events, writing a 1 clears it, writing a 0 has no effect.*
1	TPS	Transmit Process Stopped. This bit is set to 1 when the transmit process enters a stopped state. 0: Transmit Process Stopped status not detected 1: Transmit Process Stopped status detected Note: *This bit has access restrictions, automatically set to 1 for internal events, writing a 1 clears it, writing a 0 has no effect.*
0	TI	Transmit Interrupt. This bit indicates that the data packet transmission has been completed. After the completion of data packet transmission, the 31st bit of TDES3 in the last Descriptor will be reset, and specific packet status information will be updated in the Descriptor. 0: Transmit Interrupt status not detected 1: Transmit Interrupt status detected *Note: This bit has access restrictions, automatically set to 1 for internal events, writing a 1 clears it, writing a 0 has no effect.*

35.6.3.21 ETH DMA CH0 Miss Frame Count register (ETH_DMACH0DPCNT)

Address offset: 0x1164

Reset value: 0x0000 0000

This register has the number of packet counter that got dropped by the DMA either due to Bus Error or due to programming RPF field in DMA CH0 Rx Control register.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DPCO	Reserved				DPC										
r					r										

Bit Field	Name	Description
31:16	Reserved	Reserved, must be kept at reset value.

Bit Field	Name	Description
15	DPCO	Overflow status of the DPC Counter. When this bit is set to 1, the DPC counter will no longer increment. This bit is cleared when a read operation is performed on this register. *Note: This bit has access restrictions. Read clears it. Automatically set to 1 for internal events.*
14:11	Reserved	Reserved, must be kept at reset value.
10:0	DPC	Dropped Packet Counters. This counter indicates the number of data packets that have been discarded by the DMA due to bus errors or the RPF field in the programmed DMA Channel 0 receive control register. The counter is cleared when a read operation is performed on this register. *Note: This Bit Field has access restrictions. Read clears it. Automatically set to 1 for internal events.*

35.6.3.22 ETH DMA CH0 RX ERI Count register (ETH_DMACH0RXERICNT)

Address offset: 0x116C

Reset value: 0x0000 0000

This register provides the count of the number of times ERI

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				ECNT											
				r											

Bit Field	Name	Description
31:12	Reserved	Reserved, must be kept at reset value.
11:0	ECNT	ERI Counter. When ERIC bit of DMA CH0 Rx Control register is set, this counter increments for burst transfer completed by the Rx DMA from the start of packet transfer. This counter is reset at the start of new packet.

36 Module Cascading

To meet application requirements, some peripherals support internal signal interconnection.

36.1 ADC Internal Trigger Sources

Both the ADC regular channels and the injected channels support multiple internal trigger sources, which are configured separately for each ADC module using the ADCx_CTRL2.EXTRSEL[5:0] and ADCx_CTRL2.EXTJSEL[5:0] registers (where x=1, 2, 3, 4), and each ADC can be configured independently. Please refer to the ADC chapter for detailed register descriptions.

The correspondence between the register values of ADCx_CTRL2.EXTRSEL[5:0] and ADCx_CTRL2.EXTJSEL[5:0] (where x=1, 2, 3, 4) and the trigger sources is shown in the table below.

Table 36-1 ADC Trigger Source

Register Values	Trigger Source	Description
0	ATIM1_CC4	ATIM1 channel 4 compare output event
1	ATIM1_TRGO	ATIM1 master mode trigger output
2	ATIM1_TRGO2	ATIM1 master mode trigger output 2
3	ATIM2_CC4	ATIM2 channel 4 compare output event
4	ATIM2_TRGO	ATIM2 master mode trigger output
5	ATIM2_TRGO2	ATIM2 master mode trigger output 2
6	ATIM3_CC4	ATIM3 channel 4 compare output event
7	ATIM3_TRGO	ATIM3 master mode trigger output
8	ATIM3_TRGO2	ATIM3 master mode trigger output 2
9	GTIM1_CC4	GTIM1 channel 4 compare output event
10	GTIM2_CC4	GTIM2 channel 4 compare output event
11	GTIM3_CC4	GTIM3 channel 4 compare output event
12	GTIM4_CC4	GTIM4 channel 4 compare output event
13	GTIM5_CC4	GTIM5 channel 4 compare output event

14	GTIM6_CC4	GTIM6 channel 4 compare output event
15	GTIM7_CC4	GTIM7 channel 4 compare output event
16	GTIM8_CC4	GTIM8 channel 4 compare output event
17	GTIM9_CC4	GTIM9 channel 4 compare output event
18	GTIM10_CC4	GTIM10 channel 4 compare output event
19	GTIM1_TRGO	GTIM1 master mode trigger output
20	GTIM2_TRGO	GTIM2 master mode trigger output
21	GTIM3_TRGO	GTIM3 master mode trigger output
22	GTIM4_TRGO	GTIM4 master mode trigger output
23	GTIM5_TRGO	GTIM5 master mode trigger output
24	GTIM6_TRGO	GTIM6 master mode trigger output
25	GTIM7_TRGO	GTIM7 master mode trigger output
26	GTIM8_TRGO	GTIM8 master mode trigger output
27	GTIM9_TRGO	GTIM9 master mode trigger output
28	GTIM10_TRGO	GTIM10 master mode trigger output
29	SHRTIM1_ADC_TRG1	SHRTIM ADC trigger output 1
30	SHRTIM1_ADC_TRG2	SHRTIM ADC trigger output 2
31	SHRTIM1_ADC_TRG3	SHRTIM ADC trigger output 3
32	SHRTIM1_ADC_TRG4	SHRTIM ADC trigger output 4
33	SHRTIM1_ADC_TRG5	SHRTIM ADC trigger output 5
34	SHRTIM1_ADC_TRG6	SHRTIM ADC trigger output 6
35	SHRTIM1_ADC_TRG7	SHRTIM ADC trigger output 7
36	SHRTIM1_ADC_TRG8	SHRTIM ADC trigger output 8
37	SHRTIM1_ADC_TRG9	SHRTIM ADC trigger output 9

38	SHRTIM1_ADC_TRG10	SHRTIM ADC trigger output 10
39	LPTIM1_OUT	LPTIM1 compare output
40	LPTIM2_OUT	LPTIM1 compare output
41	EXTI0～15	Any channel in external interrupt channels 0 to 15
42	SWSTRRCH/SWSRTJCH	Software trigger (regular/injected channels)

36.2 DAC Internal Trigger Source

The DAC's pseudo-noise, triangle wave, and sawtooth wave update/reset, and sawtooth wave step support multiple internal trigger sources. They are configured separately for each DAC module using the DACx_SEL_CTRL.TySEL[5:0] and DACx_SEL_CTRL.STINCTRGy[5:0] registers (where x=14, y=12), and each DAC can be configured independently. Please refer to the DAC chapter for detailed register descriptions.

The correspondence between the register values of DACx_SEL_CTRL.TySEL [5:0] (where x=14, y=12) and the pseudo-noise, triangle wave, and sawtooth wave reset trigger sources is shown in the table below.

Table 36-2 ADC Update/Reset Trigger Source

Register Values	Update/Reset Trigger Source	Description	Comments
0	SW	Software trigger	
1	ATIM1_TRGO	ATIM1 master mode trigger output	
2	ATIM2_TRGO	ATIM2 master mode trigger output	
3	ATIM3_TRGO	ATIM3 master mode trigger output	
5	GTIM1_TRGO	GTIM1 master mode trigger output	
6	GTIM2_TRGO	GTIM2 master mode trigger output	
7	GTIM3_TRGO	GTIM3 master mode trigger output	
8	GTIM4_TRGO	GTIM4 master mode trigger output	
9	GTIM5_TRGO	GTIM5 master mode trigger output	
10	GTIM6_TRGO	GTIM6 master mode trigger output	
11	GTIM7_TRGO	GTIM7 master mode trigger output	
12	GTIM8_TRGO	GTIM8 master mode trigger output	

13	GTIM9_TRGO	GTIM9 master mode trigger output	
14	GTIM10_TRGO	GTIM10 master mode trigger output	
15	EXTI5	External interrupt channel 5	
16	EXTI7	External interrupt channel 7	
17	EXTI9	External interrupt channel 9	
18	SHRTIM1_DAC_RESET_TRG1	SHRTIM1 DAC reset event output 1	
19	SHRTIM1_DAC_RESET_TRG2	SHRTIM1 DAC reset event output 2	
20	SHRTIM1_DAC_RESET_TRG3	SHRTIM1 DAC reset event output 3	
21	SHRTIM1_DAC_RESET_TRG4	SHRTIM1 DAC reset event output 4	
22	SHRTIM1_DAC_RESET_TRG5	SHRTIM1 DAC reset event output 5	
23	SHRTIM1_DAC_RESET_TRG6	SHRTIM1 DAC reset event output 6	
24	SHRTIM1_DAC_TRG1	SHRTIM1 DAC update event output 1	Only applicable to DAC1/2/7/8
	SHRTIM1_DAC_TRG2	SHRTIM1 DAC update event output 2	Only applicable to DAC3/4
	SHRTIM1_DAC_TRG3	SHRTIM1 DAC update event output 3	Only applicable to DAC5/6

The correspondence between the register values of DACx_SEL_CTRL.STINCTRG1y[5:0] and the sawtooth wave step trigger sources is shown in the table below.

Table 36-3 DAC Step Trigger Source

Register Values	**Step Trigger Source**	**Description**

0	SW	Software trigger
1	ATIM1_TRGO	ATIM1 master mode trigger output
2	ATIM2_TRGO	ATIM2 master mode trigger output
3	ATIM3_TRGO	ATIM3 master mode trigger output
5	GTIM1_TRGO	GTIM1 master mode trigger output
6	GTIM2_TRGO	GTIM2 master mode trigger output
7	GTIM3_TRGO	GTIM3 master mode trigger output
8	GTIM4_TRGO	GTIM4 master mode trigger output
9	GTIM5_TRGO	GTIM5 master mode trigger output
10	GTIM6_TRGO	GTIM6 master mode trigger output
11	GTIM7_TRGO	GTIM7 master mode trigger output
12	GTIM8_TRGO	GTIM8 master mode trigger output
13	GTIM9_TRGO	GTIM9 master mode trigger output
14	GTIM10_TRGO	GTIM10 master mode trigger output
15	EXTI6	External interrupt channel 6
16	EXTI8	External interrupt channel 8
17	EXTI10	External interrupt channel 10
18	SHRTIM1_STEP_TRG1	SHRTIM1 DAC step event output 1
19	SHRTIM1_STEP_TRG2	SHRTIM1 DAC step event output 2
20	SHRTIM1_STEP_TRG3	SHRTIM1 DAC step event output 3
21	SHRTIM1_STEP_TRG4	SHRTIM1 DAC step event output 4
22	SHRTIM1_STEP_TRG5	SHRTIM1 DAC step event output 5
23	SHRTIM1_STEP_TRG6	SHRTIM1 DAC step event output 6

36.3 ATIM/GTIM Internal Trigger Input

Timers ATIMx (x=13) and GTIMx (x=110) each support 16 internal trigger input signals tim_itr[15:0], which can be used as a reference clock in slave mode, and each timer can be configured independently.

The correspondence between ATIM/GTIM tim_itr[15:0] and trigger input signal sources is shown in the table below:

Table 36-4 ATIM/GTIM Internal Trigger Input Sources

Trigger Input	Signal Sources	Description	Comments
tim_itr0	ATIM1_TRGO	ATIM1 master mode trigger output	Not applicable to ATIM1
tim_itr1	GTIM1_TRGO	GTIM1 master mode trigger output	Not applicable to GTIM1
tim_itr2	GTIM2_TRGO	GTIM2 master mode trigger output	Not applicable to GTIM2
tim_itr3	GTIM3_TRGO	GTIM3 master mode trigger output	Not applicable to GTIM3
tim_itr4	GTIM4_TRGO	GTIM4 master mode trigger output	Not applicable to GTIM4
tim_itr5	ATIM2_TRGO	ATIM2 master mode trigger output	Not applicable to ATIM2
tim_itr6	GTIM8_TRGO	GTIM8 master mode trigger output	Not applicable to GTIM8
tim_itr7	GTIM9_TRGO	GTIM9 master mode trigger output	Not applicable to GTIM9
tim_itr8	GTIM10_TRGO	GTIM10 master mode trigger output	Not applicable to GTIM10
tim_itr9	ATIM3_TRGO	ATIM3 master mode trigger output	ATIM3_TRGO
tim_itr10	SHRTIM1_OUT_SYNC2	SHRTIM1 synchronous event output	
tim_itr11	USB_SOF_SYNC	USB HS SOF pulse output	Only applicable to GTIM1
	ETH_PTP	ETH PTP trigger output	Only applicable to GTIM7

tim_itr12	GTIM5_TRGO	GTIM5 master mode trigger output	Not applicable to GTIM5
tim_itr13	GTIM6_TRGO	GTIM6 master mode trigger output	Not applicable to GTIM6
tim_itr14	GTIM7_TRGO	GTIM7 master mode trigger output	Not applicable to GTIM7
tim_itr15	Reserved		

36.4 ATIM/GTIM Internal Trigger Clear Effective Reference Level Input

Timers ATIMx (x=13) and GTIMx (x=110) each support 16 internal trigger clear effective reference level signals tim_ocref_clr015 to clear the timer's compare output reference signal OCxREF (setting OCxREF to a low level, where x is the output channel number). Each timer can be configured independently, and each channel can be individually enabled. The correspondence between tim_ocref_clr015 and trigger input signal sources is shown in the table below:

Table 36-5 ATIM/GTIM Internal Trigger Clear Effective Reference Level Input

Trigger Input	Signal Sources	Description
tim_ocref_clr0	COMP1_OUT	Comparator 1 output
tim_ocref_clr0	COMP2_OUT	Comparator 2 output
tim_ocref_clr0	COMP3_OUT	Comparator 3 output
tim_ocref_clr0	COMP4_OUT	Comparator 4 output
tim_ocref_clr0	COMP5_OUT	Comparator 5output
tim_ocref_clr0	COMP6_OUT	Comparator 6 output
tim_ocref_clr0	COMP7_OUT	Comparator 7 output
tim_ocref_clr[15:7]	Reserved	

36.5 ATIM/GTIM External Trigger Input

Timers ATIMx (x=1-3) and GTIMx (x=1-10) each support 16 external trigger input signals tim_etr[15:0], which can be used as a reference clock in slave mode or as trigger input, and each timer can be configured independently.

The correspondence between ATIM/GTIM tim_etr[15:0] and trigger input signal sources is shown below:

Table 36-6 ATIM1 External Trigger Input Signal Sources

Trigger Input	Signal Sources	Description
tim_etr0	ATIM1_ETR	ATIM1 external ETR pin input
tim_etr1	COMP1_OUT	Comparator 1 output
tim_etr2	COMP2_OUT	Comparator 2 output
tim_etr3	COMP3_OUT	Comparator 3 output
tim_etr4	COMP4_OUT	Comparator 4 output
tim_etr5	COMP5_OUT	Comparator 5 output
tim_etr6	COMP6_OUT	Comparator 6 output
tim_etr7	COMP7_OUT	Comparator 7 output
tim_etr8	ADC1_AWD1	ADC1 analog watchdog 1 output
tim_etr9	ADC1_AWD2	ADC1 analog watchdog 2 output
tim_etr10	ADC1_AWD3	ADC1 analog watchdog 3 output
tim_etr11	ADC4_AWD1	ADC4 analog watchdog 1 output
tim_etr12	ADC4_AWD2	ADC4 analog watchdog 2 output
tim_etr13	ADC4_AWD3	ADC4 analog watchdog 3 output
tim_etr[14:15]	Reserved	

Table 36-7 ATIM2 External Trigger Input Signal Sources

Trigger Input	Signal Sources	Description
tim_etr0	ATIM2_ETR	ATIM2 external ETR pin input
tim_etr1	COMP1_OUT	Comparator 1 output
tim_etr2	COMP2_OUT	Comparator 2 output
tim_etr3	COMP3_OUT	Comparator 3 output

tim_etr4	COMP4_OUT	Comparator 4 output
tim_etr5	COMP5_OUT	Comparator 5 output
tim_etr6	COMP6_OUT	Comparator 6 output
tim_etr7	COMP7_OUT	Comparator 7 output
tim_etr8	ADC2_AWD1	ADC2 analog watchdog 1 output
tim_etr9	ADC2_AWD2	ADC2 analog watchdog 2 output
tim_etr10	ADC2_AWD3	ADC2 analog watchdog 3 output
tim_etr11	ADC3_AWD1	ADC3 analog watchdog 1 output
tim_etr12	ADC3_AWD2	ADC3 analog watchdog 2 output
tim_etr13	ADC3_AWD3	ADC3 analog watchdog 3 output
tim_etr[14:15]	Reserved	

Table 36-8 ATIM3 External Trigger Input Signal Sources

Trigger Input	Signal Sources	Description
tim_etr0	ATIM3_ETR	ATIM3 external ETR pin input
tim_etr1	COMP1_OUT	Comparator 1 output
tim_etr2	COMP2_OUT	Comparator 2 output
tim_etr3	COMP3_OUT	Comparator 3 output
tim_etr4	COMP4_OUT	Comparator 4 output
tim_etr5	COMP5_OUT	Comparator 5 output
tim_etr6	COMP6_OUT	Comparator 6 output
tim_etr7	COMP7_OUT	Comparator 7 output
tim_etr8	ADC3_AWD1	ADC3 analog watchdog 1 output
tim_etr9	ADC3_AWD2	ADC3 analog watchdog 2 output

tim_etr10	ADC3_AWD3	ADC3 analog watchdog 3 output
tim_etr11	ADC1_AWD1	ADC1 analog watchdog 1 output
tim_etr12	ADC1_AWD2	ADC1 analog watchdog 2 output
tim_etr13	ADC1_AWD3	ADC1 analog watchdog 3 output
tim_etr[14:15]	Reserved	

Table 36-9 GTIM1 External Trigger Input Signal Sources

Trigger Input	Signal Sources	Description
tim_etr0	GTIM1_ETR	GTIM1 external ETR pin input
tim_etr1	COMP1_OUT	Comparator 1 output
tim_etr2	COMP2_OUT	Comparator 2 output
tim_etr3	COMP3_OUT	Comparator 3 output
tim_etr4	COMP4_OUT	Comparator 4 output
tim_etr5	COMP5_OUT	Comparator 5 output
tim_etr6	COMP6_OUT	Comparator 6 output
tim_etr7	COMP7_OUT	Comparator 7 output
tim_etr8	GTIM2_ETR	GTIM2 external ETR pin input
tim_etr9	GTIM8_ETR	GTIM8 external ETR pin input
tim_etr10	ADC1_AWD3	ADC1 analog watchdog 3 output
tim_etr11	LSE	LSE clock
tim_etr12	ADC1_AWD1	ADC1 analog watchdog 1 output
tim_etr13	ADC1_AWD2	ADC1 analog watchdog 2 output
tim_etr[14:15]	Reserved	

Table 36-10 GTIM2 External Trigger Input Signal Sources

Trigger Input	Signal Sources	Description
tim_etr0	GTIM2_ETR	GTIM2 external ETR pin input
tim_etr1	COMP1_OUT	Comparator 1 output
tim_etr2	COMP2_OUT	Comparator 2 output
tim_etr3	COMP3_OUT	Comparator 3 output
tim_etr4	COMP4_OUT	Comparator 4 output
tim_etr5	COMP5_OUT	Comparator 5 output
tim_etr6	COMP6_OUT	Comparator 6 output
tim_etr7	COMP7_OUT	Comparator 7 output
tim_etr8	GTIM1_ETR	GTIM1 external ETR pin input
tim_etr9	GTIM3_ETR	GTIM3 external ETR pin input
tim_etr10	GTIM9_ETR	GTIM9 external ETR pin input
tim_etr11	ADC2_AWD1	ADC2 analog watchdog 1 output
tim_etr12	ADC2_AWD2	ADC2 analog watchdog 2 output
tim_etr13	ADC2_AWD3	ADC2 analog watchdog 3 output
tim_etr[14:15]	Reserved	

Table 36-11 GTIM3 External Trigger Input Signal Sources

Trigger Input	Signal Sources	Description
tim_etr0	GTIM3_ETR	GTIM3 external ETR pin input
tim_etr1	COMP1_OUT	Comparator 1 output
tim_etr2	COMP2_OUT	Comparator 2 output
tim_etr3	COMP3_OUT	Comparator 3 output

tim_etr4	COMP4_OUT	Comparator 4 output
tim_etr5	COMP5_OUT	Comparator 5 output
tim_etr6	COMP6_OUT	Comparator 6 output
tim_etr7	COMP7_OUT	Comparator 7 output
tim_etr8	GTIM2_ETR	GTIM2 external ETR pin input
tim_etr9	GTIM4_ETR	GTIM4 external ETR pin input
tim_etr10	GTIM10_ETR	GTIM10 external ETR pin input
tim_etr11	ADC3_AWD1	ADC3 analog watchdog 1 output
tim_etr12	ADC3_AWD2	ADC3 analog watchdog 2 output
tim_etr13	ADC3_AWD3	ADC3 analog watchdog 3 output
tim_etr[14:15]	Reserved	

Table 36-12 GTIM4 External Trigger Input Signal Sources

Trigger Input	Signal Sources	Description
tim_etr0	GTIM4_ETR	GTIM4 external ETR pin input
tim_etr1	COMP1_OUT	Comparator 1 output
tim_etr2	COMP2_OUT	Comparator 2 output
tim_etr3	COMP3_OUT	Comparator 3 output
tim_etr4	COMP4_OUT	Comparator 4 output
tim_etr5	COMP5_OUT	Comparator 5 output
tim_etr6	COMP6_OUT	Comparator 6 output
tim_etr7	COMP7_OUT	Comparator 7 output
tim_etr8	GTIM1_ETR	GTIM1 external ETR pin input
tim_etr9	GTIM2_ETR	GTIM2 external ETR pin input

tim_etr10	GTIM8_ETR	GTIM8 external ETR pin input
tim_etr11	ADC4_AWD1	ADC4 analog watchdog 1 output
tim_etr12	ADC4_AWD2	ADC4 analog watchdog 2 output
tim_etr13	ADC4_AWD3	ADC4 analog watchdog 3 output
tim_etr[14:15]	Reserved	

Table 36-13 GTIM5 External Trigger Input Signal Sources

Trigger Input	Signal Sources	Description
tim_etr0	GTIM5_ETR	GTIM5 external ETR pin input
tim_etr1	COMP1_OUT	Comparator 1 output
tim_etr2	COMP2_OUT	Comparator 2 output
tim_etr3	COMP3_OUT	Comparator 3 output
tim_etr4	COMP4_OUT	Comparator 4 output
tim_etr5	COMP5_OUT	Comparator 5 output
tim_etr6	COMP6_OUT	Comparator 6 output
tim_etr7	COMP7_OUT	Comparator 7 output
tim_etr8	GTIM6_ETR	GTIM6 external ETR pin input
tim_etr9	GTIM7_ETR	GTIM7 external ETR pin input
tim_etr10	GTIM9_ETR	GTIM9 external ETR pin input
tim_etr11	ADC1_AWD1	ADC1 analog watchdog 1 output
tim_etr12	ADC1_AWD2	ADC1 analog watchdog 2 output
tim_etr13	ADC1_AWD3	ADC1 analog watchdog 3 output
tim_etr[14:15]	Reserved	

Table 36-14 GTIM6 External Trigger Input Signal Sources

Trigger Input	Signal Sources	Description
tim_etr0	GTIM6_ETR	GTIM6 external ETR pin input
tim_etr1	COMP1_OUT	Comparator 1 output
tim_etr2	COMP2_OUT	Comparator 2 output
tim_etr3	COMP3_OUT	Comparator 3 output
tim_etr4	COMP4_OUT	Comparator 4 output
tim_etr5	COMP5_OUT	Comparator 5 output
tim_etr6	COMP6_OUT	Comparator 6 output
tim_etr7	COMP7_OUT	Comparator 7 output
tim_etr8	GTIM5_ETR	GTIM5 external ETR pin input
tim_etr9	GTIM7_ETR	GTIM7 external ETR pin input
tim_etr10	GTIM10_ETR	GTIM10 external ETR pin input
tim_etr11	ADC2_AWD1	ADC2 analog watchdog 1 output
tim_etr12	ADC2_AWD2	ADC2 analog watchdog 2 output
tim_etr13	ADC2_AWD3	ADC2 analog watchdog 3 output
tim_etr[14:15]	Reserved	

Table 36-15 GTIM7 External Trigger Input Signal Sources

Trigger Input	Signal Sources	Description
tim_etr0	GTIM7_ETR	GTIM7 external ETR pin input
tim_etr1	COMP1_OUT	Comparator 1 output
tim_etr2	COMP2_OUT	Comparator 2 output
tim_etr3	COMP3_OUT	Comparator 3 output

tim_etr4	COMP4_OUT	Comparator 4 output
tim_etr5	COMP5_OUT	Comparator 5 output
tim_etr6	COMP6_OUT	Comparator 6 output
tim_etr7	COMP7_OUT	Comparator 7 output
tim_etr8	GTIM5_ETR	GTIM5 external ETR pin input
tim_etr9	GTIM6_ETR	GTIM6 external ETR pin input
tim_etr10	GTIM8_ETR	GTIM8 external ETR pin input
tim_etr11	ADC3_AWD1	ADC3 analog watchdog 1 output
tim_etr12	ADC3_AWD2	ADC3 analog watchdog 2 output
tim_etr13	ADC3_AWD3	analog watchdog 3 output
tim_etr[14:15]	Reserved	

Table 36-16 GTIM8 External Trigger Input Signal Sources

Trigger Input	Signal Sources	Description
tim_etr0	GTIM8_ETR	GTIM8 external ETR pin input
tim_etr1	COMP1_OUT	Comparator 1 output
tim_etr2	COMP2_OUT	Comparator 2 output
tim_etr3	COMP3_OUT	Comparator 3 output
tim_etr4	COMP4_OUT	Comparator 4 output
tim_etr5	COMP5_OUT	Comparator 5 output
tim_etr6	COMP6_OUT	Comparator 6 output
tim_etr7	COMP7_OUT	Comparator 7 output
tim_etr8	GTIM9_ETR	GTIM9 external ETR pin input
tim_etr9	GTIM10_ETR	GTIM10 external ETR pin input

tim_etr10	GTIM1_ETR	GTIM1 external ETR pin input
tim_etr11	ADC4_AWD1	ADC4 analog watchdog 1 output
tim_etr12	ADC4_AWD2	ADC4 analog watchdog 2 output
tim_etr13	ADC4_AWD3	ADC4 analog watchdog 3 output
tim_etr[14:15]	Reserved	

Table 36-17 GTIM9 External Trigger Input Signal Sources

Trigger Input	Signal Sources	Description
tim_etr0	GTIM9_ETR	GTIM9 external ETR pin input
tim_etr1	COMP1_OUT	Comparator 1 output
tim_etr2	COMP2_OUT	Comparator 2 output
tim_etr3	COMP3_OUT	Comparator 3 output
tim_etr4	COMP4_OUT	Comparator 4 output
tim_etr5	COMP5_OUT	Comparator 5 output
tim_etr6	COMP6_OUT	Comparator 6 output
tim_etr7	COMP7_OUT	Comparator 7 output
tim_etr8	GTIM8_ETR	GTIM8 external ETR pin input
tim_etr9	GTIM10_ETR	GTIM10 external ETR pin input
tim_etr10	GTIM2_ETR	GTIM2 external ETR pin input
tim_etr11	ADC1_AWD1	ADC1 analog watchdog 1 output
tim_etr12	ADC1_AWD2	ADC1 analog watchdog 2 output
tim_etr13	ADC1_AWD3	ADC1 analog watchdog 3 output
tim_etr[14:15]	Reserved	

Table 36-18 GTIM10 External Trigger Input Signal Sources

Trigger Input	Signal Sources	Description
tim_etr0	GTIM10_ETR	GTIM10 external ETR pin input
tim_etr1	COMP1_OUT	Comparator 1 output
tim_etr2	COMP2_OUT	Comparator 2 output
tim_etr3	COMP3_OUT	Comparator 3 output
tim_etr4	COMP4_OUT	Comparator 4 output
tim_etr5	COMP5_OUT	Comparator 5 output
tim_etr6	COMP6_OUT	Comparator 6 output
tim_etr7	COMP7_OUT	Comparator 7 output
tim_etr8	GTIM8_ETR	GTIM8 external ETR pin input
tim_etr9	GTIM9_ETR	GTIM9 external ETR pin input
tim_etr10	GTIM3_ETR	GTIM3 external ETR pin input
tim_etr11	ADC2_AWD1	ADC2 analog watchdog 1 output
tim_etr12	ADC2_AWD2	ADC2 analog watchdog 2 output
tim_etr13	ADC2_AWD3	ADC2 analog watchdog 3 output
tim_etr[14:15]	Reserved	

36.6 ATIM/GTIM Input Channels

ATIM/GTIM input channels support external pin inputs as well as internal signals. Each input channel supports up to 16 input signal sources tim_tix_in[15:0].

The correspondence between each channel's tim_tix_in[15:0] and the input signal sources is shown below.

Table 36-19 ATIM/GTIM Input Channel 1 Signal Sources

Input	**Signal Sources**	**Description**	**Comments**

tim_ti1_in0	CH1	Timer input channel 1 external pin	
tim_ti1_in1	COMP1_OUT	Comparator 1 output	
tim_ti1_in2	COMP2_OUT	Comparator 2 output	
tim_ti1_in3	COMP3_OUT	Comparator 3 output	
tim_ti1_in4	COMP4_OUT	Comparator 4 output	
tim_ti1_in5	LSI	LSI clock	Only applicable to GTIM5
	HSE/128	Divided by 128 from the HSE clock	Only applicable to GTIM7
tim_ti1_in6	RTC_WAKE_UP	RTC automatic wake-up event output	Only applicable to GTIM6
tim_ti1_in[15:7]	Reserved		

Table 36-20 ATIM/GTIM Input Channel 2 Signal Sources

Input	Signal Sources	Description	Comments
tim_ti2_in0	CH2	Timer input channel 2 external pin	
tim_ti2_in1	GTIM7_CC1	GTIM7 channel 1 compare output event	Only applicable to ATIM1
	GTIM7_CC2	GTIM7 channel 2 compare output event	Only applicable to ATIM2
	GTIM7_CC3	GTIM7 channel 3 compare output event	Only applicable to ATIM3
	GOMP5_OUT	Comparator 5 output	Only applicable to GTIM1~10
tim_ti2_in2	COMP6_OUT	Comparator 6 output	Only applicable to GTIM1~10
tim_ti2_in3	COMP7_OUT	Comparator 7 output	Only applicable to GTIM1~10

tim_ti2_in4	GTIM6_CC1	GTIM6 channel 1 compare output event	Only applicable to GTIM1
	GTIM6_CC2	GTIM6 channel 2 compare output event	Only applicable to GTIM2
	GTIM6_CC3	GTIM6 channel 3 compare output event	Only applicable to GTIM3
	GTIM6_CC4	GTIM6 channel 4 compare output event	Only applicable to GTIM4
	GTIM7_CC3	GTIM7 channel 3 compare output event	Only applicable to GTIM8/GTIM10
	GTIM7_CC4	GTIM7 channel 4 compare output event	Only applicable to GTIM9
	LSE	LSE clock	Only applicable to GTIM5
	LSI	LSI clock	Only applicable to GTIM7
	HSE/128	Divided by 128 from the HSE clock	Only applicable to GTIM6
tim_ti2_in[15:5]	Reserved		

Table 36-21 ATIM/GTIM Input Channel 3 Signal Sources

Input	**Signal Sources**	**Description**	**Comments**
tim_ti3_in0	CH3	Timer input channel 3 external pin	
tim_ti3_in1	COMP1_OUT	Comparator 1 output	Only applicable to GTIM1
	COMP2_OUT	Comparator 2 output	Only applicable to GTIM2
	COMP3_OUT	Comparator 3 output	Only applicable to GTIM3

	COMP4_OUT	Comparator 4 output	Only applicable to GTIM4
	COMP5_OUT	Comparator 5 output	Only applicable to GTIM8
	COMP6_OUT	Comparator 6 output	Only applicable to GTIM9
	COMP7_OUT	Comparator 7 output	Only applicable to GTIM10
	LSI	LSI clock	Only applicable to GTIM6
	HSE/128	Divided by 128 from the HSE clock	Only applicable to GTIM5
tim_ti3_in[15:2]	Reserved		

Table 36-22 ATIM/GTIM Input Channel 4 Signal Sources

Input	Signal Sources	Description	Comments
tim_ti4_in0	CH4	Timer input channel 4 external pin	
tim_ti4_in1	COMP1_OUT	Comparator 1 output	Only applicable to GTIM1
	COMP2_OUT	Comparator 2 output	Only applicable to GTIM2
	COMP3_OUT	Comparator 3 output	Only applicable to GTIM3
	COMP4_OUT	Comparator 4 output	Only applicable to GTIM4
	COMP5_OUT	Comparator 5 output	Only applicable to GTIM8
	COMP6_OUT	Comparator 6 output	Only applicable to GTIM9

	COMP7_OUT	Comparator 7 output	Only applicable to GTIM10
	MCO	MCO Clock Output	Only applicable to GTIM6
	MCO2	MCO2 Clock Output	Only applicable to GTIM7
tim_ti4_in3	RTC_WAKE_UP	RTC automatic wake-up event output	Only applicable to GTIM5
tim_ti4_in[15:3]	Reserved		

36.7 ATIM/GTIM Brake Signal Input

ATIM1/2/3, GTIM8/9/10 brake signal inputs (tim_brk1_comp[7:1], tim_brk2_comp[7:1]) support internal interconnection with comparator outputs, as shown in the table below.

Table 36-23 ATIM/GTIM Brake Signal Input Interconnected with Comparator

Brake Signal Input	Signal Source	Description	Comments
tim_brk1_comp1	COMP1_OUT	Comparator 1 output	
tim_brk1_comp2	COMP2_OUT	Comparator 2 output	
tim_brk1_comp3	COMP3_OUT	Comparator 3 output	
tim_brk1_comp4	COMP4_OUT	Comparator 4 output	
tim_brk1_comp5	COMP5_OUT	Comparator 5 output	
tim_brk1_comp6	COMP6_OUT	Comparator 6 output	
tim_brk1_comp7	COMP7_OUT	Comparator 7 output	
tim_brk2_comp1	COMP1_OUT	Comparator 1 output	Only applicable to ATIM1/2/3
tim_brk2_comp2	COMP2_OUT	Comparator 2 output	Only applicable to ATIM1/2/3
tim_brk2_comp3	COMP3_OUT	Comparator 3 output	Only applicable to ATIM1/2/3

tim_brk2_comp4	COMP4_OUT	Comparator 4 output	Only applicable to ATIM1/2/3
tim_brk2_comp5	COMP5_OUT	Comparator 5 output	Only applicable to ATIM1/2/3
tim_brk2_comp6	COMP6_OUT	Comparator 6 output	Only applicable to ATIM1/2/3
tim_brk2_comp7	COMP7_OUT	Comparator 7 output	Only applicable to ATIM1/2/3

36.8 COMP Blanking Signal Input

Multiple timers can be internally interconnected with comparators as blanking signal input sources, configured through the COMP module's COMPx_CTRL.BLANKING [2:0] (where x=17). Each comparator can be configured independently.

The correspondence between the COMPx_CTRL.BLANKING [2:0] (where x=17) register values and the blanking signal input sources is shown in the table below.

Table 36-24 COMP Blank Signal Input Source

Register Value	Input Sources	Description
1	ATIM1_OC5	ATIM1 channel 5 compare reference output
2	ATIM2_OC5	ATIM2 channel 5 compare reference output
3	ATIM3_OC5	ATIM3 channel 5 compare reference output
4	GTIM8_OC5	GTIM8 channel 5 compare reference output
5	GTIM9_OC5	GTIM9 channel 5 compare reference output
6	GTIM10_OC5	GTIM10 channel 5 compare reference output

36.9 SHRTIM Synchronous Input

SHRTIM supports 4 synchronous signal inputs shrtim1_in_sync[4:1], configured through the SHRTIM_MCTRL.SYNCIN[2:0] register. The correspondence is shown in the table below:

Table 36-25 SHRTIM Synchronous Input Signal Sources

Synchronous Input	Signal Source	Description
shrtim1_in_sync1	ATIM1_TRGO	ATIM1 master mode trigger output
shrtim1_in_sync2	ATIM2_TRGO	ATIM2 master mode trigger output
shrtim1_in_sync3	ATIM3_TRGO	ATIM3 master mode trigger output
shrtim1_in_sync4	SHRTIM1_SCIN	SHRTIM Synchronous Input Pin

36.10 SHRTIM External Event Input

SHRTIM supports 10 external event inputs SHRTIM_EXEVx (x=1-10), with each external event input supporting multiple signal sources. They can be configured using the SHRTIM_EXEVCTRL1.EXEVxSRC (x=1-5) and SHRTIM_EXEVCTRL2.EXEVxSRC (x=6-10) registers. The correspondence between the register values and input signal sources is shown in the table below:

Table 36-26 SHRTIM External Event Input Signal Sources

Register Value	Signal Source	Description	Comments
0	SHRTIM_EXEVx (x=1~10)	External Event Input Pin SHRTIM1_EXEVx	External event input pin, can be mapped to any IO port, selected by registers AFIO_SHRT_EXEV_CFGy.SHRT1_EXEVx[6:0] (y=1-3, x=1-10)
1	COMPx_OUT(x=1~7)	Comparator Output	Can be mapped to any comparator, selected by register SHRTIM_EXEVCTRL5.EXEVxCSEL[2:0]
2	ATIM1_TRGO	ATIM1 master mode trigger output	Only applicable to SHRTIM_EXEV1 event
	ATIM2_TRGO	ATIM2 master mode trigger output	Only applicable to SHRTIM_EXEV2 event

	ATIM3_TRGO	ATIM3 master mode trigger output	Only applicable to SHRTIM_EXEV3 event
	BTIM2_TRGO	BTIM2 master mode trigger output	Only applicable to SHRTIM_EXEV7 event
	GTIM8_TRGO	GTIM8 master mode trigger output	Only applicable to SHRTIM_EXEV9 event
	GTIM9_TRGO	GTIM9 master mode trigger output	Only applicable to SHRTIM_EXEV10 event
3	ADC1_AWD1	ADC1 analog watchdog 1 output	Only applicable to SHRTIM_EXEV1 event
	ADC1_AWD2	ADC1 analog watchdog 2 output	Only applicable to SHRTIM_EXEV2 event
	ADC1_AWD3	ADC1 analog watchdog 3 output	Only applicable to SHRTIM_EXEV3 event
	ADC2_AWD1	ADC2 analog watchdog 1 output	Only applicable to SHRTIM_EXEV4 event
	ADC2_AWD2	ADC2 analog watchdog 2 output	Only applicable to SHRTIM_EXEV5 event
	ADC2_AWD3	ADC2 analog watchdog 3 output	Only applicable to SHRTIM_EXEV6 event
	ADC3_AWD2	ADC3 analog watchdog 2 output	Only applicable to SHRTIM_EXEV7 event
	ADC3_AWD3	ADC3 analog watchdog 3 output	Only applicable to SHRTIM_EXEV8 event
	ADC4_AWD2	ADC4 analog watchdog 2 output	Only applicable to SHRTIM_EXEV9 event
	ADC4_AWD3	ADC4 analog watchdog 3 output	Only applicable to SHRTIM_EXEV10 event

36.11 SHRTIM Fault Input

SHRTIM supports 6 fault inputs SHRTIM1_FALTx (x=1-6), with each fault input supporting multiple signal sources. They can be configured using the FALTxSRC[1:0] (x=1-6) registers in SHRTIM_FALTIN1 and SHRTIM_FALTIN2. The correspondence between the register values and input signal sources is shown in the table below:

Table 36-27 SHRTIM Fault Input Signal Sources

Register Value	Signal Source	Description	Comments
0	SHRTIM1_FALTx(x=1~6)	fault input pin SHRTIM1_FALTx	
1	COMPx_OUT(x=1~7)	Comparator Output	Can be mapped to any comparator, selected by register HRT_FLTINR5.FALTxCSEL[2:0]
2	SHRTIM_EXEVx (x=1~10)	External event input pin SHRTIM1_EXEVx	External Event Input Pin, can be mapped to any IO port, selected by registers AFIO_SHRT_EXEV_CFGy.SHRT1_EXEVx[6:0] (y=1-3, x=1-10)

37 Debug Support (DBG)

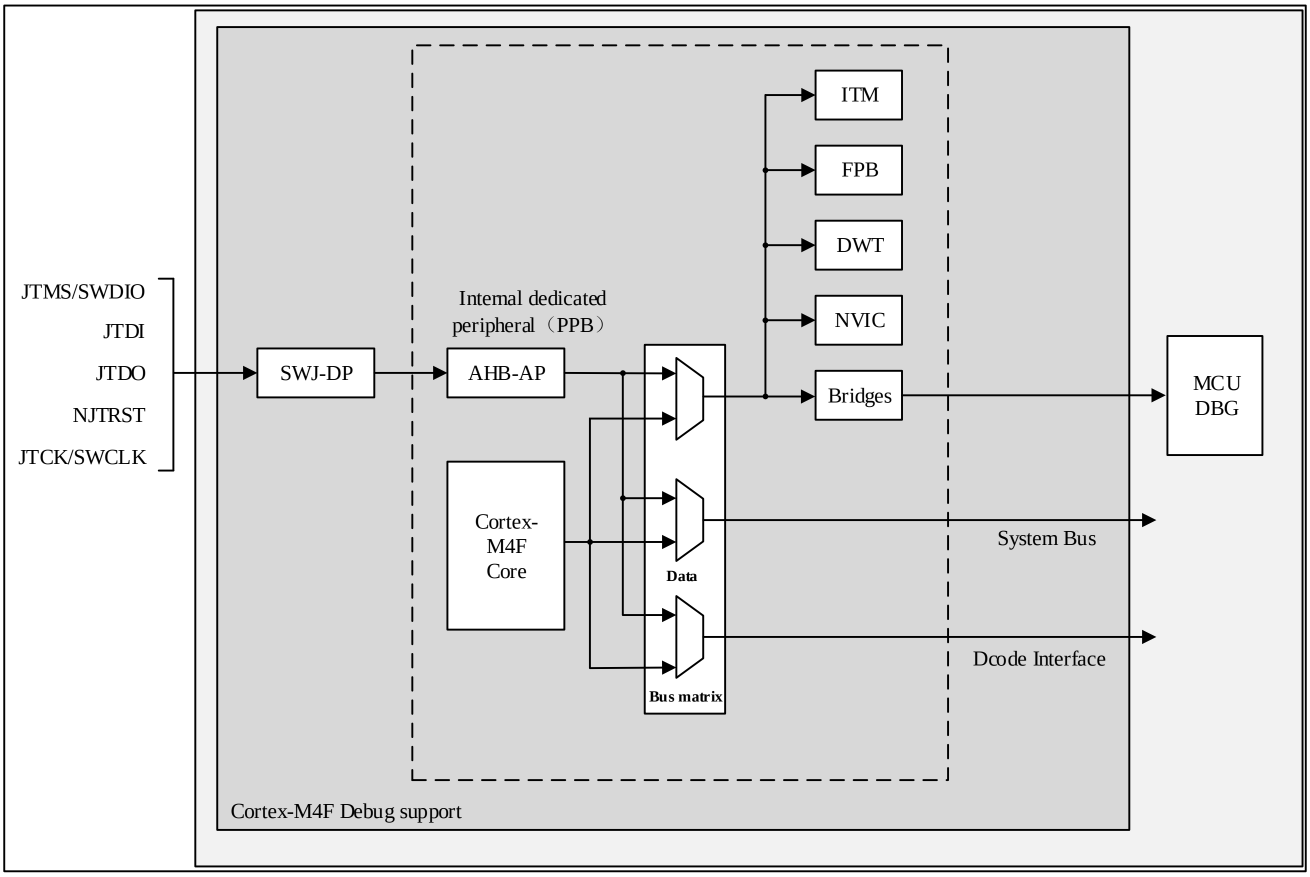
37.1 Overview

The N32H47x and N32H48x series use Cortex^{TM}-M4F core, which integrates hardware debugging module. Supporting instruction breakpoint (stop when instruction fetches value) and data breakpoint (stop when data access). When the kernel is stopped, the user can view the internal state of the kernel and the external state of the system. After the user's query operation is completed, the kernel and peripherals can be restored, and the corresponding program can continue to be executed. The hardware debugging module of the kernel can be used when they are connected to the debugger (when it is not disabled).

Supports the following debugging interfaces:

- Serial wire
- JTAG debugging interface

Figure 37-1 Chip Level And CortexTM-M4F Level Debugging Block Diagram



The ARM Cortex^{TM}-M4F core hardware debugging module can provide the following debugging functions:

- SWJ-DP: Serial /JTAG debug port
- AHP-AP:AHB access port
- ITM: Instrumentation trace macrocell

- FPB: Flash patch breakpoint
- DWT: Data watchpoint trigger

Reference:

- Cortex^{TM}-M4F Technical Reference Manual (TRM)
- ARM debugging interface V5 structure specification
- ARM CoreSight development tool set (r1p0 version) technical reference manua

The system supports low-power mode debugging and debugging of some peripherals. The peripherals supporting debugging include: I2C interface and TIMER, WWDG and IWDG modules. The user needs to set the corresponding bit of the debug control register (DBG_CTRL) to 1 when debugging with low power consumption or peripherals.

37.2 JTAG/SWD Function

The debugging tool can call the debugging function through the SWD debugging interface or JTAG debugging interface mentioned above.

37.2.1 Switching JTAG/SWD Interface

The chip uses JTAG debug interface by default. If you need to switch the debug interface, you can switch between SWD interface and JTAG interface through the following operations:

JTAG debug to SWD debug switch:

1. Sending JTMS = 1 signals with more than 50 JTCK cycles;
2. Send 16-bit JTMS = 1110011110011110(0xE79E LSB) signal;
3. Send JTMS = 1 signal with more than 50 JTCK cycles.

Switch from SWD debugging to JTAG debugging:

1. Sending JTMS = 1 signals with more than 50 JTCK cycles;
2. Send 16-bit JTMS = 1110011100111100(0xE73C LSB) signal;
3. Send JTMS = 1 signal with more than 50 JTCK cycles.

37.2.2 Pin Assignment

JTAG debugging interface includes five pins: JTCK(JTAG clock pin), JTMS(JTAG mode selection pin), JTDI(JTAG data input pin), JTDO(JTAG data output pin) and NJTRST(JTAG data reset pin, low level reset pin).

SWD (serial debugging) interface includes two pins: SWCLK (clock pin) and SWDIO (data input and output pin), which provide the interface of two pins: data input and output pin (SWDIO) and clock pin (SWCLK).

See the following Table for the pin allocation of JTAG debugging interface and SWD debugging interface (SWDIO is multiplexed with JTMS, SWCLK is multiplexed with JTCK):

Table 37-1 Debug Port Pin

Debug Port	Pin Assignment
JTMS/SWDIO	PA13
JTCK/SWCLK	PA14
JTDI	PA15
JTDO	PB3
NJTRST	PB4

- When both JTAG debugging interface and SWD debugging interface are enabled, the 5-wire JTAG debugging interface will be used by default after reset.
- When using JTAG interface, users can not use NJTRST pin. In this case, NJTRST pin (PB4, internal hardware pull-up) can be used as a general-purpose GPIO.
- When SWD interface is used, three pins JTDI(PA15), JTDO(PB3/PG15) and NJTRST(PB4) can be used as general GPIO.
- When the debugging function is not used, the above five pins can be used as general-purpose GPIO.

37.3 MCU Debug Function

37.3.1 Low-Power Mode Debug Support

Provide various low-power modes (refer to Chapter 3 for details). When debugging, ensure that the FCLK and HCLK of the kernel are on, and provide the necessary clock for kernel debugging. Users can debug MCU in low power mode according to specific operation (ensuring the output of FCLK or HCLK in low power mode).

If users want to debug MCU in low power mode, they first need the debugger to configure registers related to low power mode:

- DBG_SLEEP mode:

  The DBG_CTRL.SLEEP bit needs to be configured to provide HCLK with the same clock as provided to FCLK (ie: the original configured system clock).

- DBG_STOP mode:

  The DBG_CTRL.STOP bit needs to be configured to start the internal RC oscillator to provide the clock for HCLK and FCLK.

- DBG_STANDBY mode:

  The DBG_CTRL.STDBY bit needs to be configured to start the internal RC oscillator to provide the clock for HCLK and FCLK.

37.3.2 Peripheral Debug Support

When the corresponding bit of the peripheral control bit in the DBG_CTRL register is set to 1, the corresponding peripheral enters the debugging state after the kernel stops:

- Timer peripheral: the timer counter stops and debugs;
- I2C peripheral: the SMBUS of I2C keeps the state and carries out debugging;
- WWDG/IWDG peripheral: WWDG/IWDG counter clock stops and debugs;

37.4 DBG Registers

37.4.1 DBG Register Overview

These peripheral registers must be operated as words (32 bits). The base address of the register is 0xE004 2000.

Table 37-2 DBG Register Overview

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
000h	DBG_ID	SRAM[3:0]				Reserved				DEV_NUM_L[3:0]				FLASH[3:0]				DEV_NUM_H[3:0]				DEV_NUM_M[3:0]				REV_NUM_H[3:0]				REV_NUM_L[3:0]			
	Reset Value	x	x	x	x					x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
004h	DBG_CTRL	Reserved		SHRTIM1_STOP	GTIM10_STOP	GTIM9_STOP	GTIM8_STOP	GTIM7_STOP	GTIM6_STOP	GTIM5_STOP	GTIM4_STOP	GTIM3_STOP	GTIM2_STOP	GTIM1_STOP	BTIM2_STOP	BTIM1_STOP	ATIM3_STOP	ATIM2_STOP	ATIM1_STOP	I2C4SMBUS_TIMEOUT	I2C3SMBUS_TIMEOUT	I2C2SMBUS_TIMEOUT	I2C1SMBUS_TIMEOUT	WWDG_STOP	IWDG_STOP	Reserved					STDBY	STOP	SLEEP
	Reset Value			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0						0	0	0

37.4.2 ID Register (DBG_ID)

Address offset: 0x00

Only 32-bit access is supported, and fixed values cannot be modified

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SRAM[3:0]				Reserved				DEV_NUM_L[3:0]				FLASH[3:0]			
r								r				r			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DEV_NUM_H[3:0]				DEV_NUM_M[3:0]				REV_NUM_H[3:0]				REV_NUM_L[3:0]			
r				r				r				r			

Bit Field	Name	Description
31:28	SRAM[3:0]	SRAM capacity. The chip SRAM capacity is (SRAM[3:0] + 1) * 16KB
27:24	Reserved	Reserved, must keep the reset value.
23:20	DEV_NUM_L[3:0]	Lower 4 digits of device model. Device model consists of 12 bits, including high, medium and low, representing the model of MCU. The values are as follows:

Bit Field	Name	Description
		● 0x473: Basic N32H473 ● 0x474: Enhanced N32H474 ● 0x475: Interconnected N32H475 ● 0x473: Basic N32H473 ● 0x480: Basic N32H480 ● 0x482: Enhanced N32H482 ● 0x488: Enhanced N32H488 ● 0x487: Interconnected N32H487
19:16	FLASH[3:0]	FLASH capacity. Chip FLASH capacity is FLASH[3:0] * 64KB
15:12	DEV_NUM_H[3:0]	The upper 4 digits of the device model. See the description of DEV_NUM_L[3:0].
11:8	DEV_NUM_M[3:0]	The middle 4 digits of the device model. See the description of DEV_NUM_L[3:0].
7:4	REV_NUM_H[3:0]	High 4 bits of MCU version number
3:0	REV_NUM_L[3:0]	Low 4 bits of MCU version number

37.4.3 Debug Control Register (DBG_CTRL)

Address offset: 0x04

POR reset value: 0x0000 0000 (not reset by system reset)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		SHRTIM1_STOP	GTIM10_STOP	GTIM9_STOP	GTIM8_STOP	GTIM7_STOP	GTIM6_STOP	GTIM5_STOP	GTIM4_STOP	GTIM3_STOP	GTIM2_STOP	GTIM1_STOP	BTIM2_STOP	BTIM1_STOP	ATIM3_STOP
		rw	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ATIM2_STOP	ATIM1_STOP	I2C4SMBUS_TIMEOUT	I2C3SMBUS_TIMEOUT	I2C2SMBUS_TIMEOUT	I2C1SMBUS_TIMEOUT	WWDG_STOP	IWDG_STOP	Reserved					STDBY	STOP	SLEEP
rw	rw	rw	rw	rw	rw	rw	rw						rw	rw	rw

Bit Field	Name	Description
31:22	Reserved	Reserved, the reset value must be maintained.
29	SHRTIM1_STOP	When the core is halted, the counter of high precision timer 1 stops operating. It can be set to 1 or cleared by software. 0: The counter of the selected high precision timer 1 continues to operate normally. 1: The counter of the selected high precision timer 1 stops operating.
28:19	GTIMx_STOP	When the core is halted, the counter of the general purpose timer stops operating (x=10, 9, 8, 7, 6, 5, 4, 3, 2, 1). It can be set to 1 or cleared by software. 0: The counter of the selected general purpose timer continues to operate normally. 1: The counter of the selected general purpose timer stops operating.

Bit Field	Name	Description
18:17	BTIMx_STOP	When the core is halted, the counter of the basic timer stops operating (x=2, 1). It can be set to 1 or cleared by software. 0: The counter of the selected basic timer continues to operate normally. 1: The counter of the selected basic timer stops operating.
16:14	ATIMx_STOP	When the core is halted, the counter of the advanced timer stops operating (x=3, 2, 1). It can be set to 1 or cleared by software. 0: The counter of the selected advanced timer continues to operate normally. 1: The counter of the selected advanced timer stops operating.
13:10	I2CxSMBUS_TIMEOUT	When the core stops, the SMBus timeout mode stops (x=4, 3, 2, 1). It can be set to 1 or cleared by software. 0: Operates the same as normal mode. 1: Freezes the SMBus timeout control.
9	WWDG_STOP	When the core enters debug state, the debug window watchdog stops operating. It can be set to 1 or cleared by software. 0: The window watchdog counter continues to operate normally. 1: The window watchdog counter stops operating.
8	IWDG_STOP	When the core enters debug state, the watchdog stops operating. It can be set to 1 or cleared by software. 0: The watchdog counter continues to operate normally. 1: The watchdog counter stops operating.
7:3	Reserved	Reserved, the reset value must be maintained.
2	STDBY	Debug standby mode. Set or cleared by software. 0: (FCLK OFF, HCLK OFF) The whole digital circuit is powered off. From the software point of view, exiting the STANDBY mode is the same as resetting (except that some status bits indicate that the microcontroller has just exited from the STANDBY state). 1: (FCLK ON, HCLK ON) The digital circuit part is not powered down, and the FCLK and HCLK clocks are clocked by the internal RLD oscillator. In addition, it is the same as resetting that the microcontroller exits the STANDBY mode by generating a system reset.
1	STOP	Debug stop mode. Set or cleared by software. 0: (FCLK OFF, HCLK OFF) In stop mode, the clock controller disables all clocks (including HCLK and FCLK). When exiting from STOP mode, the configuration of the clock is the same as that after reset (the microcontroller is clocked by the 8MHz internal RC oscillator (HSI)). Therefore, the software must reconfigure the clock control system to start PLL, crystal oscillator, etc.

Bit Field	Name	Description
		1: (FCLK ON, HCLK ON) In stop mode, the FCLK and HCLK clocks are provided by the internal RC oscillator. When exiting the stop mode, the software must reconfigure the clock system to start PLL, crystal oscillator, etc. (the same operation as when this bit is set to 0).
0	SLEEP	Debug sleep mode. Set or cleared by software. 0: (FCLK is ON, HCLK is OFF) In sleep mode, FCLK is provided by the previously configured system clock, while HCLK is off. Since sleep mode does not reset the configured clock system, the software does not need to reconfigure the clock system when exiting from sleep mode. 1: (FCLK ON, HCLK ON) In sleep mode, both the FCLK and HCLK clocks are provided by the previously configured system clock.

38 Unique Device Serial Number (UID)

38.1 Introduction

MCU series products have two built-in unique device serial numbers with different lengths, namely 96-bit UID (Unique device ID) and 128-bit UCID (Unique Customer ID). These two device serial numbers are stored in the system configuration block of the flash memory, and the information is programmed during manufacture, and any MCU microcontroller is guaranteed to be unique under any circumstances. It can be read by user applications or external devices through CPU or SWD interface and cannot be modified.

UID is 96 bits, which is usually used as serial number or password. When writing flash memory, this unique identifier is combined with software encryption and decryption algorithm to further improve the security of code in flash memory.

UCID is 128 bits and complies with the definition of the Nations Technologies chip serial number. It contains information about chip production and version.

38.2 UID Register

Start address: 0x1FFF_C760, 96 bits in length.

38.3 UCID Register

Start address: 0x1FFF_C740, 128 bits in length.

39 Version History

Version	**Date**	**Changes**
V1.0.0	2024/10/10	Initial version
V1.1.0	2025/4/21	1. Optimize Figure 17-28 2. Chapter 27.3.20 modify year description 3. Chapter 34.2 modify clock source select of USBHS 4. 7.4.12 Chapter Adding MEMORY to MEMORY Mode Usage Notes 5. Optimize chapter 3.1.2.3 BOR enable description 6. 2.2.1.4.4 Chapter ROW Programming Add Note 7. Chapter 29 Adding SAC Operating Clock Notes 8. 28.5.5 Optimizing XSPI Register Descriptions 9. Optimization Figure 7-1 DMA Block Diagram 10. Chapter 17.9.4.1 Add notes on the use of crossover mode 11. Modified analog signal channel naming and interpretation in sections 5.4.7 through 5.4.10. 12. 3.5.6 Section PWR_CTRL4 Add Register Bit Descriptions 13. Section 4.2 Adding Notes on APB2 Frequency Configuration
V1.2.0	2025/8/18	1. Modified Table 10-16 Table 12-14 2. Delete the note of TIMx_CTRL1. ASYMMETRIC 3. Add USB clock usage restrictions in section 4.2.6 4. Add SDIO/USBHS/ETH Clock Usage Restrictions to HCLKPRESS Bit in section 4.3.4 5. Add LPTIM1-2 clock usage restrictions in section 4.2.1 6. The order of bits for system faults in SHRTIM_FALTIN2 has been modified 7. Modified the description of the synchronous output function of SHRTIM in 9.3.19.1 and 9.4.1.12 8. RCC_SHRTPLLCTRLx (x=1,2,3) Register Add Reset Clear Instructions 9. Optimize Section 4.2.10 CSS Interrupt Explanation

		10. Add SHRTPLL clock lock control bit to Chapter 4.3.22 11. Section 3.5.3: Precautions for Adding PVDO Logo 12. Added PWR_NRSTCTRL register and description in section 3.5.7 13. Chapter 2.2.3 Add SRAM One Click Initialization Process 14. 2.2.1.4 Chapter Adding Flash Operation Precautions 15. Section 17.9.7: Adding Precautions for Use 16. Add PGA input channel switching considerations in Chapter 20.2 17. Add attention points to OREF position in Chapter 23.7.5 18. Attention points for adding ARA logo in Chapter 25.5.16 19. 31.7.2.2 Chapter CLKDIV Bit Addition and Usage Precautions 20. Add Wakeup signal wake-up points to Chapter 33.4.5.22 21. Section 26.5.4: Precautions for Adding the Undercover Logo 22. Attention Points for Adding I2C Slave Address Configuration in Chapter 24.3.2.6 23. Section 27.2.2 adds RTC_OUT2 (PB2) usage issues 24. Section 4.3.16 Optimisation of HSECSS Register Descriptions

40 Disclaimer

This document is the exclusive property of NSING TECHNOLOGIES PTE. LTD. (Hereinafter referred to as NSING). This document, and the product of NSING described herein (Hereinafter referred to as the Product) are owned by NSING under the laws and treaties of Republic of Singapore and other applicable jurisdictions worldwide. The intellectual properties of the product belong to NSING Technologies Inc. and NSING Technologies Inc. does not grant any third party any license under its patents, copyrights, trademarks, or other intellectual property rights. Names and brands of third party may be mentioned or referred thereto (if any) for identification purposes only. NSING reserves the right to make changes, corrections. enhancements, modifications, and improvements to this document at any time without notice. Please contact NSING and obtain the latest version of this document before placing orders. Although NSING has attempted to provide accurate and reliable information, NSING assumes no responsibility for the accuracy and reliability of this document. It is the responsibility of the user of this document to properly design, program, and test the functionality and safety of any application made of this information and any resulting product. In no event shall NSING be liable for any direct, indirect, incidental, special, exemplary, or consequential damages arising in any way out of the use of this document or the Product. NSING Products are neither intended nor warranted for usage in systems or equipment, any malfunction or failure of which may cause loss of human life, bodily injury or severe property damage. Such applications are deemed, Insecure Usage'. Insecure usage includes, but is not limited to: equipment for surgical implementation, atomic energy control instruments, airplane or spaceship instruments, all types of safety devices, and other applications intended to supporter sustain life. All Insecure Usage shall be made at user's risk. User shall indemnify NSING and hold NSING harmless from and against all claims, costs, damages, and other liabilities, arising from or related to any customer's Insecure Usage Any express or implied warranty with regard to this document or the Product, including, but not limited to. The warranties of merchantability, fitness for a particular purpose and non-infringement are disclaimed to the fullest extent permitted by law. Unless otherwise explicitly permitted by NSING, anyone may not use, duplicate, modify, transcribe or otherwise distribute this document for any purposes, in whole or in part.