Chapter 1 BF7707AMXX MCU general description

1.1. Features

- **Core: ARM Cortex-M0+**
  - Operating frequency: 48MHz, 32MHz, 24MHz, 12MHz
    Clock error: ±1% @-20°C~65°C, 5V; ±3% @-40°C~105°C, 5V
  - Single-cycle 32-bit hardware multiplier
- **Memory (FLASH)**
  - FLASH: 128K Bytes, support erase and write protection and read protection functions
  - DATA: 512 Bytes
  - SRAM: 12K Bytes
- **Clock source, reset and power management**
  - Internal low-speed RC oscillator: LIRC 32kHz,
    clock error is ±10% @25°C, 5V; ±25% @-40°C ~105°C, 5V
  - Internal high-speed RC oscillator: RC 1MHz
  - External crystal oscillator: XTAL 32768Hz/4MHz/8MHz passive crystal and active crystal(1MHz~48MHz)
  - 7 resets, power-down reset voltage (BOR):
    2.8V/3.3V/3.7V/4.2V(configured by option byte area)
  - Low voltage detection: 2.7V/3.0V/3.3V/3.6V/3.9V/4.2V/4.5V
- **IO**
  - Both support built-in pull-up resistor 12k
  - High current sink port (PA0~PA7)
  - Support IO function remapping
  - 16 external interrupt input channels, all EXTI support any I/O port mapping, support rising edge/ falling edge/ double edge trigger, edge wake up, support interrupt filtering
- **Communication module**
  - 5xUART communication module, UARTx supports remapping, TXD/RXD configuration, receiving idle, transmitter output polarity and receiver input polarity are optional
  - 1xIIC master-slave communication, both master and slave support 100kHz/400kHz/1MHz
  - 2xSPI master-slave communication, the master supports up to 8MHz, the slave supports up to 4MHz,support CRC hardware check
- **16-Bit PWM**
  - PWM0 supports one output and three mappings
  - PWM1 supports one output and three mappings
  - Support timing mode
- **DMA**
  - 3 channels, each channel has a separate interrupt enable and interrupt flag
  - Supports transmission between memory and memory, peripherals and memory
  - Two channel priorities can be matched
  - Supports peripherals TIM, ADC, SPI, IIC, and UART
- **Operating voltage: 2.6V ~5.5V**
- **Operating temperature: -40°C ~105°C**
  - Enhanced industrial grade, in line with the JESD industrial grade reliability certification standards
- **High precision 8/12-bit ADC, 1M sps ADC**
  - 18 GPIO analog input channels
  - 2 internal analog inputs: VREF channel (internal channel reference voltage channel), TS channel (built-in temperature sensor)
  - Trigger mode can be software register trigger or hardware event trigger
  - Reference voltage: VCC/2V/4V
  - Single conversion (including hardware average) /DMA continuous conversion mode
- **Interrupt**
  - 29 interrupt sources
- 4-level interrupt priority can be configured
- **Timer**
  - 16-bit advanced control timer: 1 (TIM0)
  - 16-bit universal timer: 7 (TIM1/2/3/4/5/6/7)
  - TIM0: 6 independent channels, 16 bit increment, decrement, increment/decrement automatic overload counter
  - TIM0 supports programmable dead-time complementary output and brake input
  - TIM1/2/3/4/5/7: 1 independent channel, 16-bit increment automatic overload counter
  - TIM6: 4 independent channels, 16 bit increment, decrement, increment/decrement automatic overload counter
  - TIM0 and TIM6 support incremental encoders and hall sensor circuits
  - TIM supports input capture (except for channel 5/6), output comparison, PWM generation, and one-pulse mode output
  - Watchdog timer: overflow time ranges from 18ms to 2.304s
  - SysTick Timer
- **Low power management**
  - Idle mode 0, power consumption 1.9mA@5V typical
  - Idle mode 1, power consumption 6.1μA@5V typical
- **Cyclic redundancy check unit**
  - CRC8/16/32, used for code check and data check
- **hardware divider**
  - 32-bit divider (signed and unsigned operations can be selected)
  - 16 clock cycles to complete the operation
- **128-bit chip unique identification code**
- **Serial two-wire debugging interface SWD, PGC/PGD programming**
- **Package**
  - LQFP44/LQFP52/LQFP64

1.2. Overview

BF7707AMXX is a 32-bit high performance microcontroller based on the ARM Cortex-M0+ core.The Cortex-M0+ core is based on the ARMv6-M architecture and supports the Thumb instruction set.

BF7707AMXX series working voltage is 2.6V~5.5V, working temperature is -40℃~+105℃, products provide a variety of different working modes, in order to cope with the power consumption needs in different situations. BF7707AMXX is embedded with 128KB Flash and 12KB SRAM, with a maximum operating frequency of48MHz.

The product offers 8 16-bit timers (1 advanced control timer, 7 universal timers).The product provides a variety of standard communication interfaces: 5-channels UART, 1-channel IIC, 2-channels SPI, 8/12 bit high-precision 1M high-speed ADC with optional resolution ratio, and with VCC/ 2V/4V reference ADC voltage function, its integrated temperature sensor can realize real-time monitoring of its own temperature. Powerful DMA data handling function, external interrupt function can be freely mapped to any port.

BF7707AMXX series provides CRC code check and data check of 8/16/32 bits, built-in watchdog timer, low voltage reset circuit, power failure detection circuit and other modules.Low power management of BF7707AMXX series can cope with power consumption requirements in different situations.

1.3. System architecture

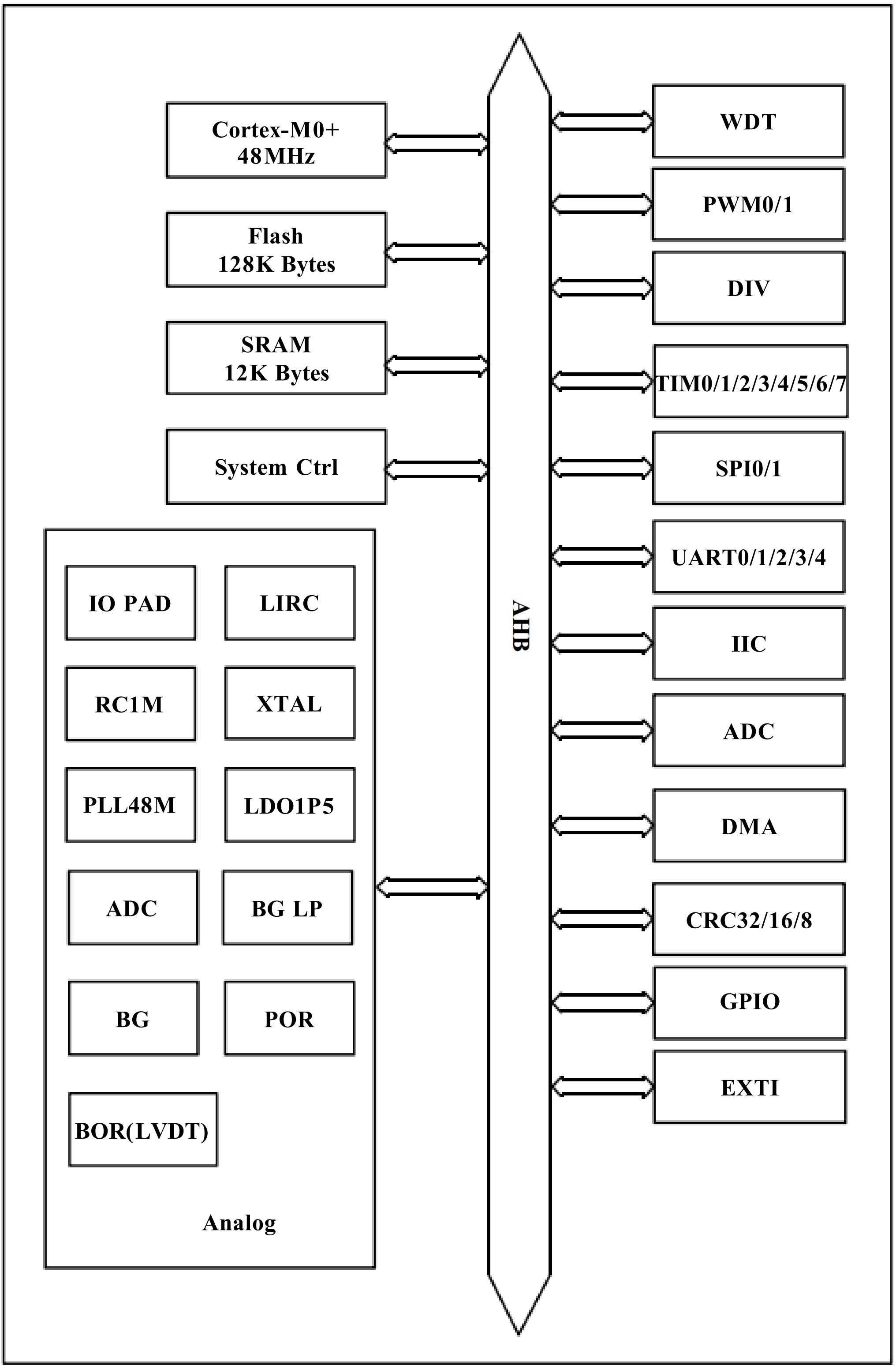


Figure 1.1 System architecture

1.4. Memory map

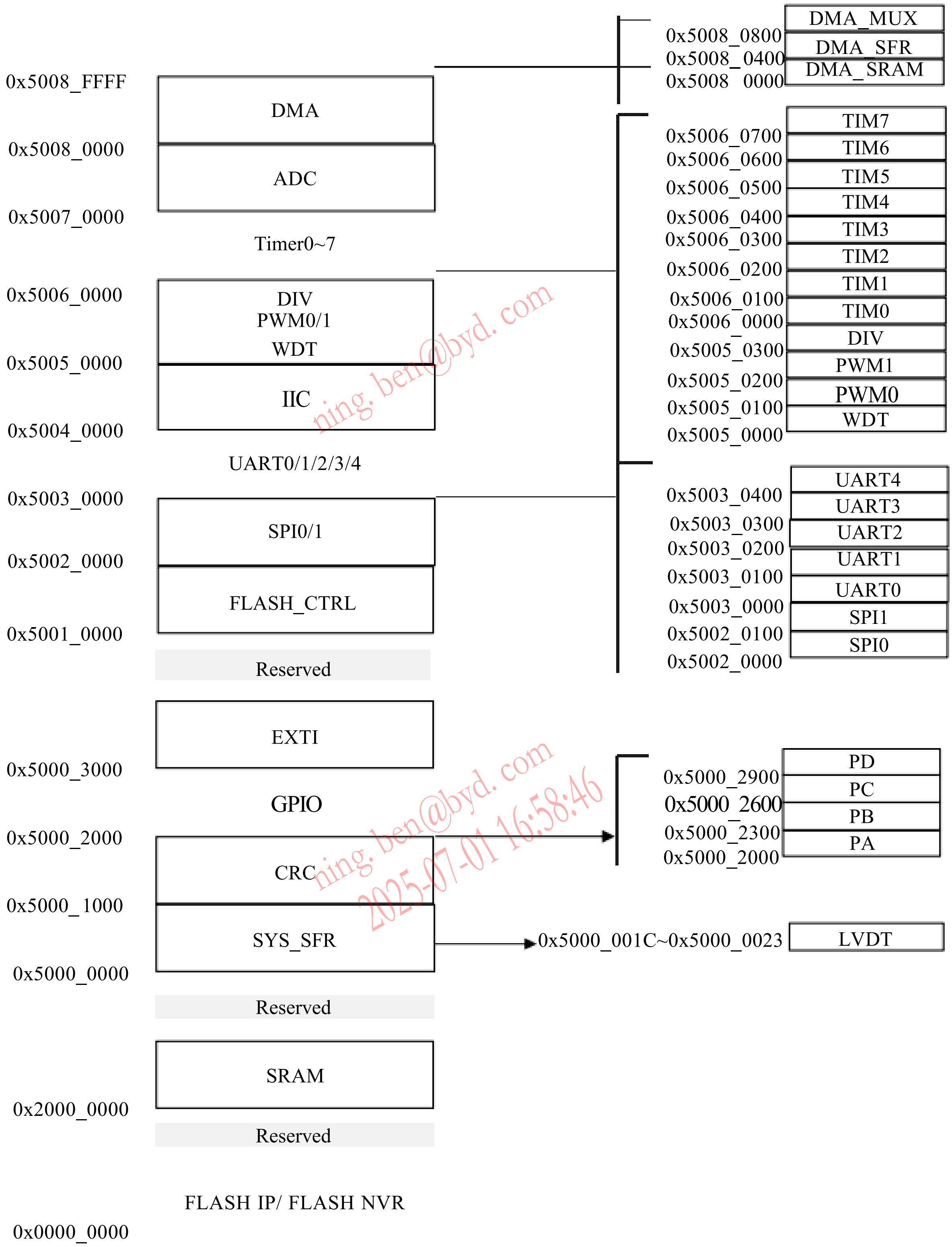


Figure 1.2 Address division map

Start address	End address	Size(Bytes)	Module
0x0000_0000	0x0002_07FF	130K	FLASH IP、FLASH NVR
0x2000_0000	0x2000_2FFF	12K	SRAM
0x5000_0000	0x5000_0057	88	SYS_SFR (Include LVDT)
0x5000_1000	0x5000_1013	20	CRC
0x5000_2000	0x5000_2087	136	GPIOA
0x5000_2300	0x5000_2387	136	GPIOB
0x5000_2600	0x5000_2687	136	GPIOC
0x5000_2900	0x5000_2987	136	GPIOD
0x5000_3000	0x5000_302B	44	EXTI
0x5001_0000	0x5001_002F	48	FLASH_CTRL
0x5002_0000	0x5002_001F	32	SPI0
0x5002_0100	0x5002_011F	32	SPI1
0x5003_0000	0x5003_001F	32	UART0
0x5003_0100	0x5003_011F	32	UART1
0x5003_0200	0x5003_021F	32	UART2
0x5003_0300	0x5003_031F	32	UART3
0x5003_0400	0x5003_041F	32	UART4
0x5004_0000	0x5004_0023	36	IIC
0x5005_0000	0x5005_0007	8	WDT
0x5005_0100	0x5005_010B	12	PWM0
0x5005_0200	0x5005_020B	12	PWM1
0x5005_0300	0x5005_0317	24	DIV
0x5006_0000	0x5006_006F	112	TIM0
0x5006_0100	0x5006_016F	112	TIM1
0x5006_0200	0x5006_026F	112	TIM2
0x5006_0300	0x5006_036F	112	TIM3
0x5006_0400	0x5006_046F	112	TIM4
0x5006_0500	0x5006_056F	112	TIM5
0x5006_0600	0x5006_066F	112	TIM6
0x5006_0700	0x5006_076F	112	TIM7
0x5007_0000	0x5007_002B	44	ADC
0x5008_0000	0x5008_03FF	1K	DMA_SRAM
0x5008_0400	0x5008_0463	100	DMA_SFR
0x5008_0800	0x5008_080B	12	DMA_MUX

Table 1.1 Address division

Note: LVDT address range: 0x5000_001C to 0x5000_0023

1.5. Clock block diagram

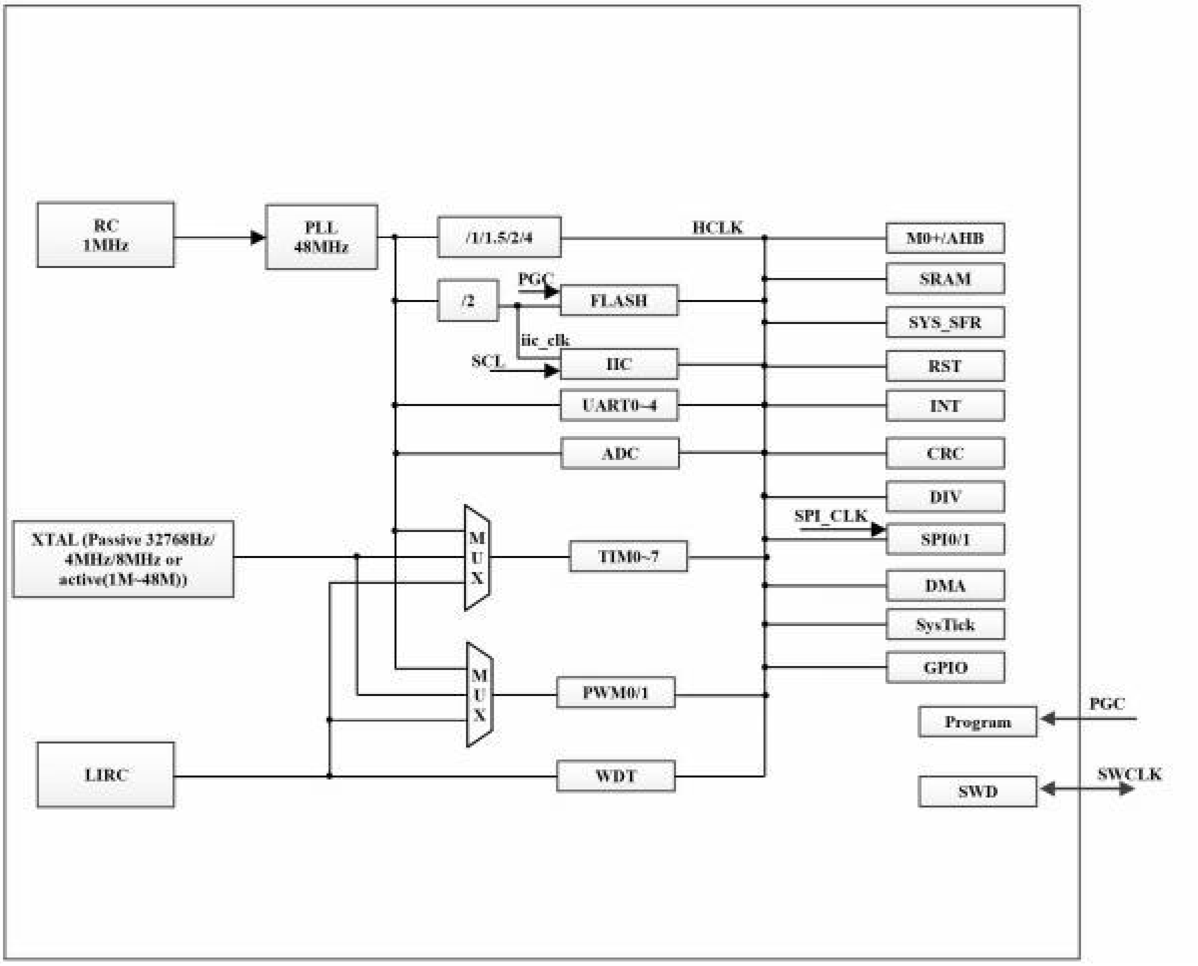


Figure 1.3 Clock block diagram

1.6. Selection list

Model		BF7707AM44-LJTX	BF7707AM52_LJTB	BF7707AM64-LJTA	BF7707AM64-LJTB
Operating voltage (V)		2.6~5.5	2.6~5.5	2.6~5.5	2.6~5.5
Operating temperature (℃)		-40~+105	-40~+105	-40~+105	-40~+105
System main frequency $_{max}$(Hz)		48M	48M	48M	48M
Core		ARM Cortex-M0+	ARM Cortex-M0+	ARM Cortex-M0+	ARM Cortex-M0+
Memory (Bytes)	FLASH	128K	128K	128K	128K
	DATA	512	512	512	512
	SRAM	12K	12K	12K	12K
Universal port	GPIO	42	50	60	60
	INT	42	50	60	60
	COM	8	8	8	8
Timer	Advanced timer	1	1	1	1
	Universal timer	7	7	7	7
	WDT	1	1	1	1
PWM module	PWM0 (16bit)	1ch	1ch	1ch	1ch
	PWM1 (16bit)	1ch	1ch	1ch	1ch
Communication module	IIC	1	1	1	1
	UART	5	5	5	5
	SPI	2	2	2	2
Analog module	ADC (12bit)	11ch	15ch	18ch	18ch
System correlation	DIV (32bit)	1	1	1	1
	DMA	3ch	3ch	3ch	3ch
	CRC	1	1	1	1
Package		LQFP44 (10mm*10mm，e=0.80mm)	LQFP52 (10mm*10mm，e=0.65mm)	LQFP64 (10mm*10mm，e=0.5mm)	LQFP64 (12mm*12mm，e=0.65mm)

1.7. Pin configuration

1.7.1. BF7707AM44-LJTX

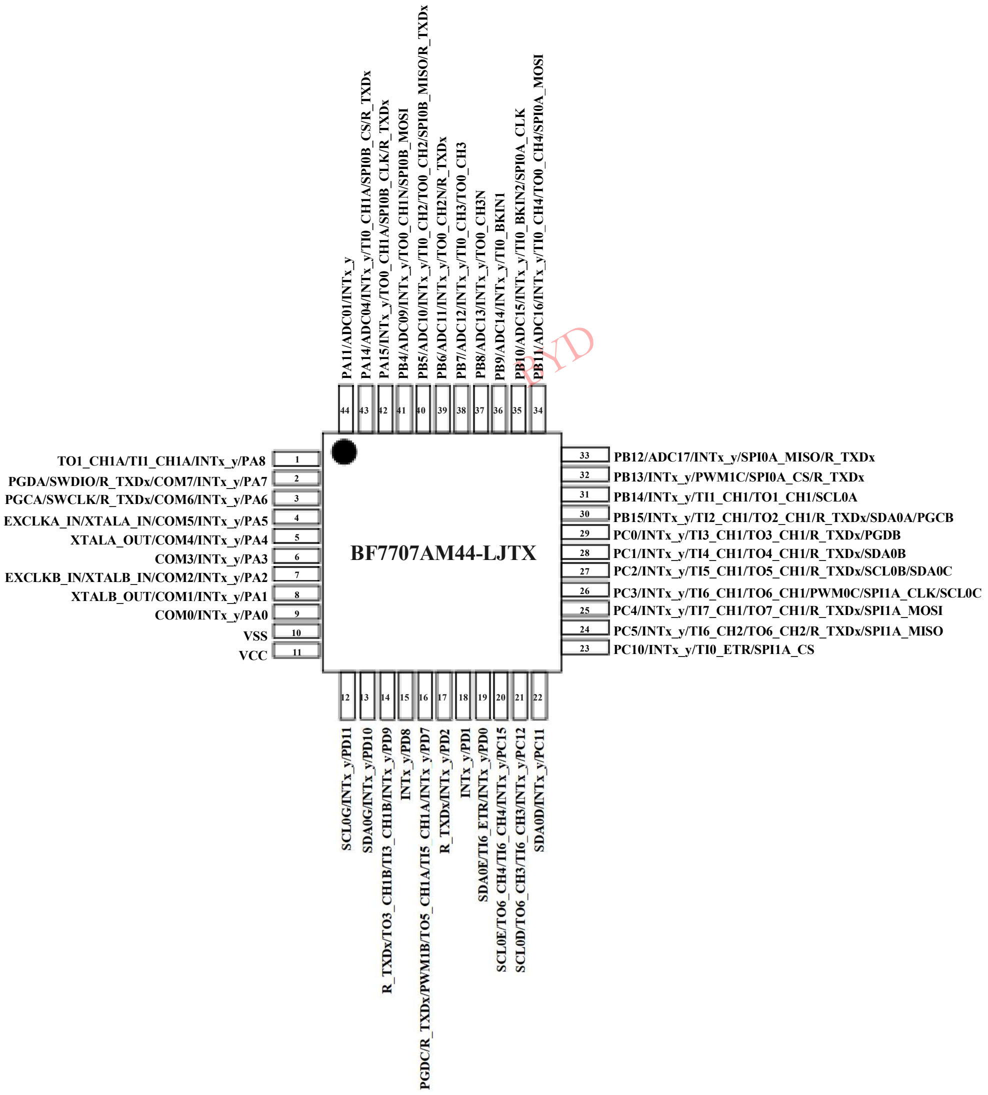

BF7707AM44-LJTX
1 TO1_CH1A/TI1_CH1A/INTx_y/PA8
2 PGDA/SWDIO/R_TXDx/COM7/INTx_y/PA7
3 PGCA/SWCLK/R_TXDx/COM6/INTx_y/PA6
4 EXCLKA_IN/XTALA_IN/COM5/INTx_y/PA5
5 XTALA_OUT/COM4/INTx_y/PA4
6 COM3/INTx_y/PA3
7 EXCLKB_IN/XTALB_IN/COM2/INTx_y/PA2
8 XTALB_OUT/COM1/INTx_y/PA1
9 COM0/INTx_y/PA0
10 VSS
11 VCC
12 SCL0G/INTx_y/PD11
13 SDA0G/INTx_y/PD10
14 R_TXDx/TO3_CH1B/TI3_CH1B/INTx_y/PD9
15 INTx_y/PD8
16 PGDC/R_TXDx/PWM1B/TO5_CH1A/TI5_CH1A/INTx_y/PD7
17 R_TXDx/INTx_y/PD2
18 INTx_y/PD1
19 SDA0E/TI6_ETR/INTx_y/PD0
20 SCL0E/TO6_CH4/TI6_CH4/INTx_y/PC15
21 SCL0D/TO6_CH3/TI6_CH3/INTx_y/PC12
22 SDA0D/INTx_y/PC11
23 PC10/INTx_y/TI0_ETR/SPI1A_CS
24 PC5/INTx_y/TI6_CH2/TO6_CH2/R_TXDx/SPI1A_MISO
25 PC4/INTx_y/TI7_CH1/TO7_CH1/R_TXDx/SPI1A_MOSI
26 PC3/INTx_y/TI6_CH1/TO6_CH1/PWM0C/SPI1A_CLK/SCL0C
27 PC2/INTx_y/TI5_CH1/TO5_CH1/R_TXDx/SCL0B/SDA0C
28 PC1/INTx_y/TI4_CH1/TO4_CH1/R_TXDx/SDA0B
29 PC0/INTx_y/TI3_CH1/TO3_CH1/R_TXDx/PGDB
30 PB15/INTx_y/TI2_CH1/TO2_CH1/R_TXDx/SDA0A/PGCB
31 PB14/INTx_y/TI1_CH1/TO1_CH1/SCL0A
32 PB13/INTx_y/PWM1C/SPI0A_CS/R_TXDx
33 PB12/ADC17/INTx_y/SPI0A_MISO/R_TXDx
34 PB11/ADC16/INTx_y/TI0_CH4/TO0_CH4/SPI0A_MOSI
35 PB10/ADC15/INTx_y/TI0_BKIN2/SPI0A_CLK
36 PB9/ADC14/INTx_y/TI0_BKIN1
37 PB8/ADC13/INTx_y/TO0_CH3N
38 PB7/ADC12/INTx_y/TI0_CH3/TO0_CH3
39 PB6/ADC11/INTx_y/TO0_CH2N/R_TXDx
40 PB5/ADC10/INTx_y/TI0_CH2/TO0_CH2/SPI0B_MISO/R_TXDx
41 PB4/ADC09/INTx_y/TO0_CH1N/SPI0B_MOSI
42 PA15/INTx_y/TO0_CH1A/SPI0B_CLK/R_TXDx
43 PA14/ADC04/INTx_y/TI0_CH1A/SPI0B_CS/R_TXDx
44 PA11/ADC01/INTx_y

Figure 1.4 LQFP44 package pin diagram

Note: The INT function can be mapped to INTx_y（x=0/1/2/3，y=0/1/2/3）, and the UART function can be mapped to R_TXDx（UART 的 RXD/TXD，x=0/1/2/3/4）. For more information, refer to "GPIO pin multiplexing configuration Table".

1.7.2. BF7707AM52-LJTB

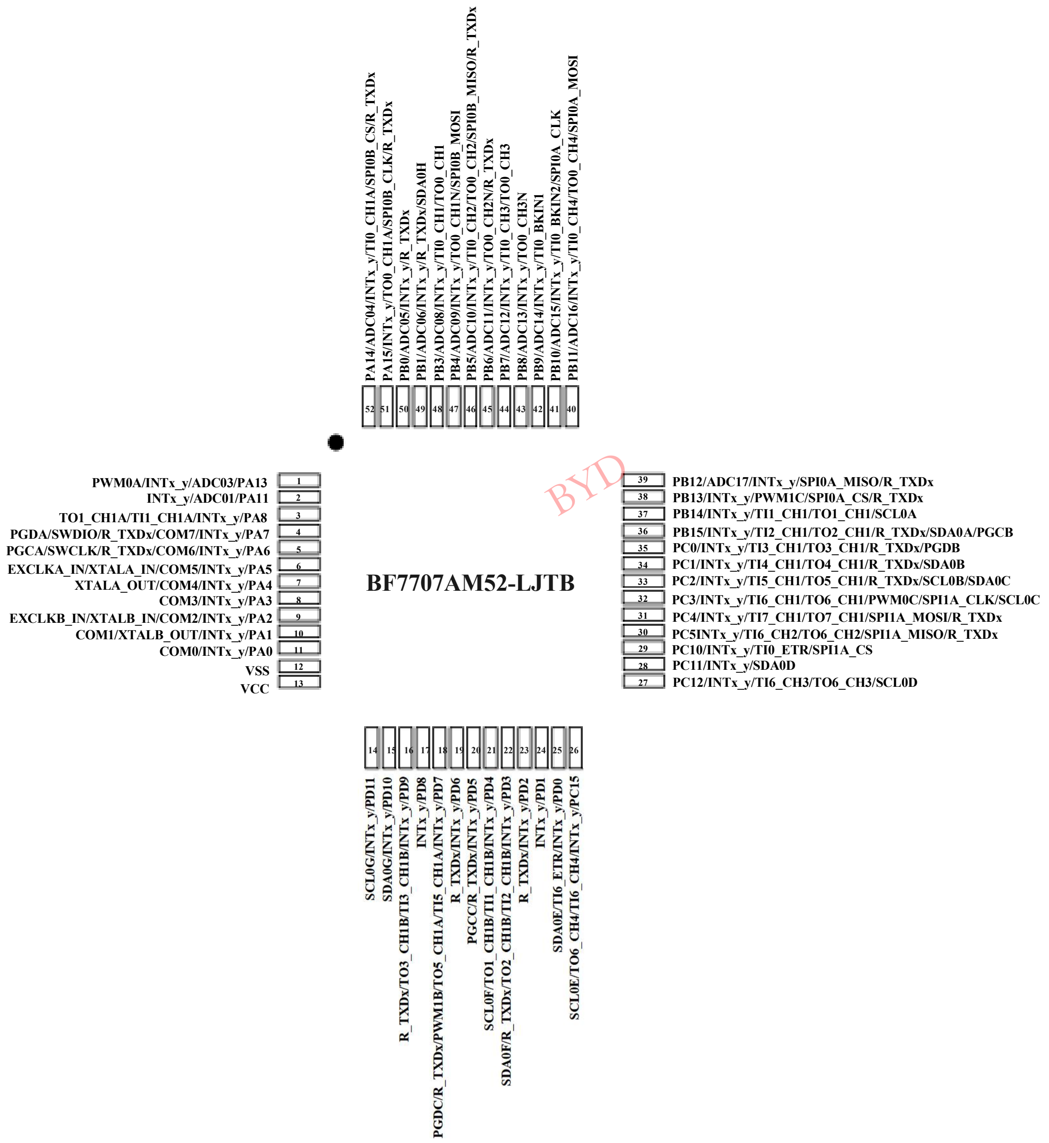

PA14/ADC04/INTx_y/TI0_CH1A/SPI0B_CS/R_TXDx
PA15/INTx_y/TO0_CH1A/SPI0B_CLK/R_TXDx
PB0/ADC05/INTx_y/R_TXDx
PB1/ADC06/INTx_y/R_TXDx/SDA0H
PB3/ADC08/INTx_y/TI0_CH1/TO0_CH1
PB4/ADC09/INTx_y/TO0_CH1N/SPI0B_MOSI
PB5/ADC10/INTx_y/TI0_CH2/TO0_CH2/SPI0B_MISO/R_TXDx
PB6/ADC11/INTx_y/TO0_CH2N/R_TXDx
PB7/ADC12/INTx_y/TI0_CH3/TO0_CH3
PB8/ADC13/INTx_y/TO0_CH3N
PB9/ADC14/INTx_y/TI0_BKIN1
PB10/ADC15/INTx_y/TI0_BKIN2/SPI0A_CLK
PB11/ADC16/INTx_y/TI0_CH4/TO0_CH4/SPI0A_MOSI
52
51
50
49
48
47
46
45
44
43
42
41
40
PWM0A/INTx_y/ADC03/PA13
1
INTx_y/ADC01/PA11
2
TO1_CH1A/TI1_CH1A/INTx_y/PA8
3
PGDA/SWDIO/R_TXDx/COM7/INTx_y/PA7
4
PGCA/SWCLK/R_TXDx/COM6/INTx_y/PA6
5
EXCLKA_IN/XTALA_IN/COM5/INTx_y/PA5
6
XTALA_OUT/COM4/INTx_y/PA4
7
COM3/INTx_y/PA3
8
EXCLKB_IN/XTALB_IN/COM2/INTx_y/PA2
9
COM1/XTALB_OUT/INTx_y/PA1
10
COM0/INTx_y/PA0
11
VSS
12
VCC
13
BF7707AM52-LJTB
39
PB12/ADC17/INTx_y/SPI0A_MISO/R_TXDx
38
PB13/INTx_y/PWM1C/SPI0A_CS/R_TXDx
37
PB14/INTx_y/TI1_CH1/TO1_CH1/SCL0A
36
PB15/INTx_y/TI2_CH1/TO2_CH1/R_TXDx/SDA0A/PGCB
35
PC0/INTx_y/TI3_CH1/TO3_CH1/R_TXDx/PGDB
34
PC1/INTx_y/TI4_CH1/TO4_CH1/R_TXDx/SDA0B
33
PC2/INTx_y/TI5_CH1/TO5_CH1/R_TXDx/SCL0B/SDA0C
32
PC3/INTx_y/TI6_CH1/TO6_CH1/PWM0C/SPI1A_CLK/SCL0C
31
PC4/INTx_y/TI7_CH1/TO7_CH1/SPI1A_MOSI/R_TXDx
30
PC5INTx_y/TI6_CH2/TO6_CH2/SPI1A_MISO/R_TXDx
29
PC10/INTx_y/TI0_ETR/SPI1A_CS
28
PC11/INTx_y/SDA0D
27
PC12/INTx_y/TI6_CH3/TO6_CH3/SCL0D
14
15
16
17
18
19
20
21
22
23
24
25
26
SCL0G/INTx_y/PD11
SDA0G/INTx_y/PD10
R_TXDx/TO3_CH1B/TI3_CH1B/INTx_y/PD9
INTx_y/PD8
PGDC/R_TXDx/PWM1B/TO5_CH1A/TI5_CH1A/INTx_y/PD7
R_TXDx/INTx_y/PD6
PGCC/R_TXDx/INTx_y/PD5
SCL0F/TO1_CH1B/TI1_CH1B/INTx_y/PD4
SDA0F/R_TXDx/TO2_CH1B/TI2_CH1B/INTx_y/PD3
R_TXDx/INTx_y/PD2
INTx_y/PD1
SDA0E/TI6_ETR/INTx_y/PD0
SCL0E/TO6_CH4/TI6_CH4/INTx_y/PC15

Figure 1.5 LQFP52 package pin diagram

Note: The INT function can be mapped to INTx_y（x=0/1/2/3，y=0/1/2/3）, and the UART function can be mapped to R_TXDx（UART 的 RXD/TXD，x=0/1/2/3/4）. For more information, refer to "GPIO pin multiplexing configuration Table".

1.7.3. BF7707AM64-LJTA

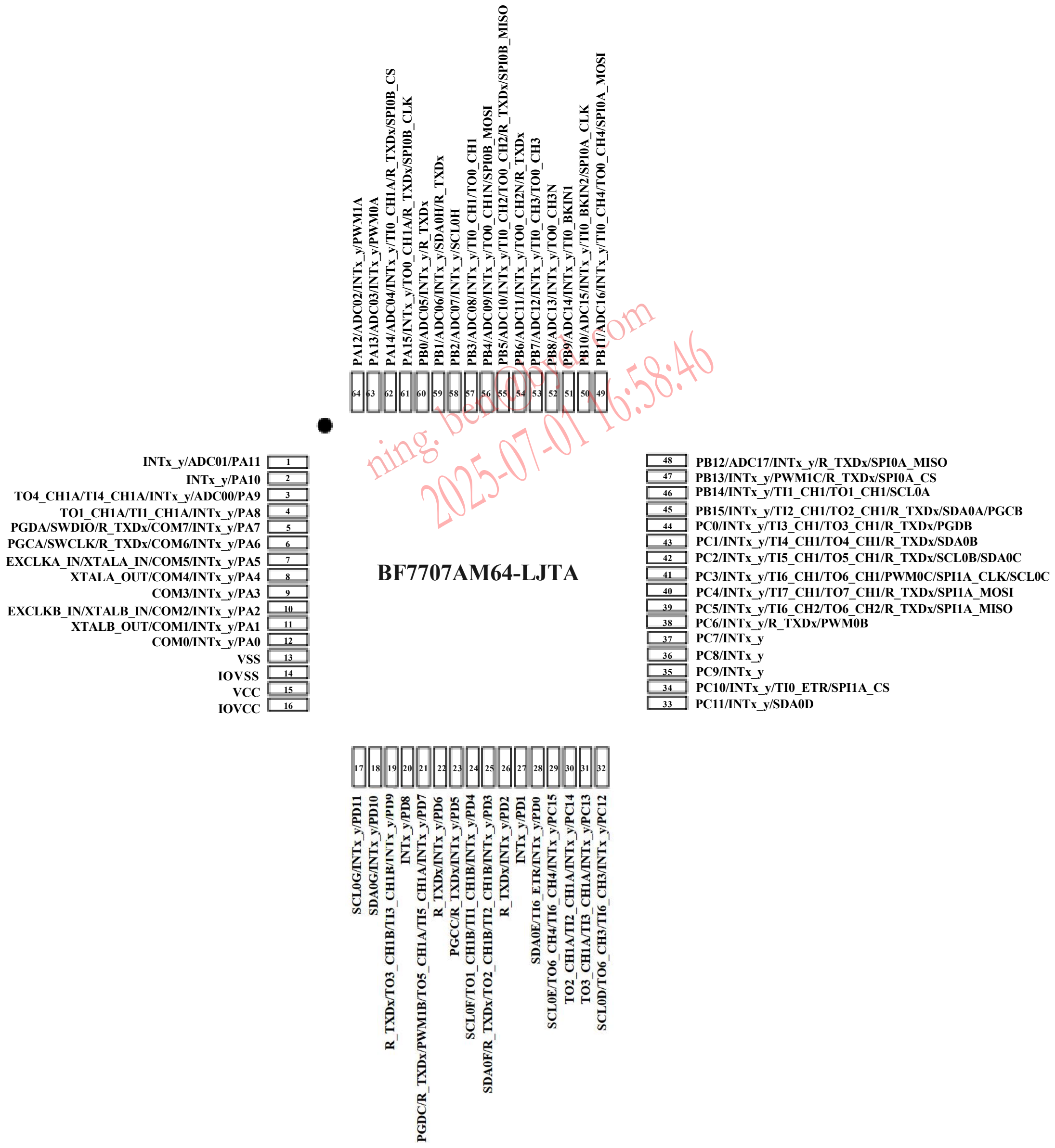

BF7707AM64-LJTA
1 INTx_y/ADC01/PA11
2 INTx_y/PA10
3 TO4_CH1A/TI4_CH1A/INTx_y/ADC00/PA9
4 TO1_CH1A/TI1_CH1A/INTx_y/PA8
5 PGDA/SWDIO/R_TXDx/COM7/INTx_y/PA7
6 PGCA/SWCLK/R_TXDx/COM6/INTx_y/PA6
7 EXCLKA_IN/XTALA_IN/COM5/INTx_y/PA5
8 XTALA_OUT/COM4/INTx_y/PA4
9 COM3/INTx_y/PA3
10 EXCLKB_IN/XTALB_IN/COM2/INTx_y/PA2
11 XTALB_OUT/COM1/INTx_y/PA1
12 COM0/INTx_y/PA0
13 VSS
14 IOVSS
15 VCC
16 IOVCC
17 SCL0G/INTx_y/PD11
18 SDA0G/INTx_y/PD10
19 R_TXDx/TO3_CH1B/TI3_CH1B/INTx_y/PD9
20 INTx_y/PD8
21 PGDC/R_TXDx/PWM1B/TO5_CH1A/TI5_CH1A/INTx_y/PD7
22 R_TXDx/INTx_y/PD6
23 PGCC/R_TXDx/INTx_y/PD5
24 SCL0F/TO1_CH1B/TI1_CH1B/INTx_y/PD4
25 SDA0F/R_TXDx/TO2_CH1B/TI2_CH1B/INTx_y/PD3
26 R_TXDx/INTx_y/PD2
27 INTx_y/PD1
28 SDA0E/TI6_ETR/INTx_y/PD0
29 SCL0E/TO6_CH4/TI6_CH4/INTx_y/PC15
30 TO2_CH1A/TI2_CH1A/INTx_y/PC14
31 TO3_CH1A/TI3_CH1A/INTx_y/PC13
32 SCL0D/TO6_CH3/TI6_CH3/INTx_y/PC12
33 PC11/INTx_y/SDA0D
34 PC10/INTx_y/TI0_ETR/SPI1A_CS
35 PC9/INTx_y
36 PC8/INTx_y
37 PC7/INTx_y
38 PC6/INTx_y/R_TXDx/PWM0B
39 PC5/INTx_y/TI6_CH2/TO6_CH2/R_TXDx/SPI1A_MISO
40 PC4/INTx_y/TI7_CH1/TO7_CH1/R_TXDx/SPI1A_MOSI
41 PC3/INTx_y/TI6_CH1/TO6_CH1/PWM0C/SPI1A_CLK/SCL0C
42 PC2/INTx_y/TI5_CH1/TO5_CH1/R_TXDx/SCL0B/SDA0C
43 PC1/INTx_y/TI4_CH1/TO4_CH1/R_TXDx/SDA0B
44 PC0/INTx_y/TI3_CH1/TO3_CH1/R_TXDx/PGDB
45 PB15/INTx_y/TI2_CH1/TO2_CH1/R_TXDx/SDA0A/PGCB
46 PB14/INTx_y/TI1_CH1/TO1_CH1/SCL0A
47 PB13/INTx_y/PWM1C/R_TXDx/SPI0A_CS
48 PB12/ADC17/INTx_y/R_TXDx/SPI0A_MISO
49 PB11/ADC16/INTx_y/TI0_CH4/TO0_CH4/SPI0A_MOSI
50 PB10/ADC15/INTx_y/TI0_BKIN2/SPI0A_CLK
51 PB9/ADC14/INTx_y/TI0_BKIN1
52 PB8/ADC13/INTx_y/TO0_CH3N
53 PB7/ADC12/INTx_y/TI0_CH3/TO0_CH3
54 PB6/ADC11/INTx_y/TO0_CH2N/R_TXDx
55 PB5/ADC10/INTx_y/TI0_CH2/TO0_CH2/R_TXDx/SPI0B_MISO
56 PB4/ADC09/INTx_y/TO0_CH1N/SPI0B_MOSI
57 PB3/ADC08/INTx_y/TI0_CH1/TO0_CH1
58 PB2/ADC07/INTx_y/SCL0H
59 PB1/ADC06/INTx_y/SDA0H/R_TXDx
60 PB0/ADC05/INTx_y/R_TXDx
61 PA15/INTx_y/TO0_CH1A/R_TXDx/SPI0B_CLK
62 PA14/ADC04/INTx_y/TI0_CH1A/R_TXDx/SPI0B_CS
63 PA13/ADC03/INTx_y/PWM0A
64 PA12/ADC02/INTx_y/PWM1A

Figure 1.6 LQFP64 package pin diagram

Note: The INT function can be mapped to INTx_y（x=0/1/2/3，y=0/1/2/3）, and the UART function can be mapped to R_TXDx（UART 的 RXD/TXD，x=0/1/2/3/4）. For more information, refer to "GPIO pin multiplexing configuration Table".

.

1.7.4. BF7707AM64-LJTB

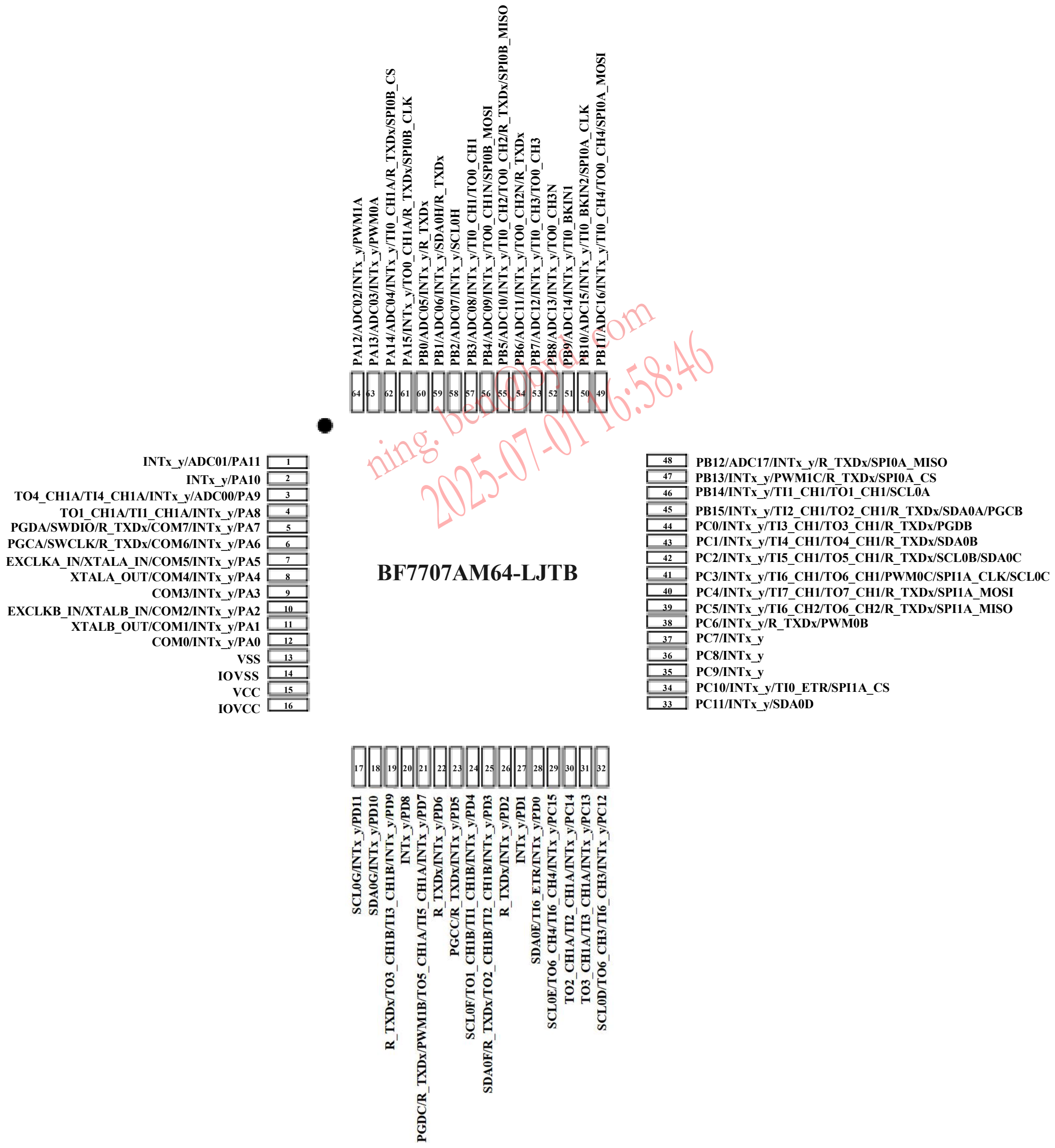

BF7707AM64-LJTB
1 INTx_y/ADC01/PA11
2 INTx_y/PA10
3 TO4_CH1A/TI4_CH1A/INTx_y/ADC00/PA9
4 TO1_CH1A/TI1_CH1A/INTx_y/PA8
5 PGDA/SWDIO/R_TXDx/COM7/INTx_y/PA7
6 PGCA/SWCLK/R_TXDx/COM6/INTx_y/PA6
7 EXCLKA_IN/XTALA_IN/COM5/INTx_y/PA5
8 XTALA_OUT/COM4/INTx_y/PA4
9 COM3/INTx_y/PA3
10 EXCLKB_IN/XTALB_IN/COM2/INTx_y/PA2
11 XTALB_OUT/COM1/INTx_y/PA1
12 COM0/INTx_y/PA0
13 VSS
14 IOVSS
15 VCC
16 IOVCC
17 SCL0G/INTx_y/PD11
18 SDA0G/INTx_y/PD10
19 R_TXDx/TO3_CH1B/TI3_CH1B/INTx_y/PD9
20 INTx_y/PD8
21 PGDC/R_TXDx/PWM1B/TO5_CH1A/TI5_CH1A/INTx_y/PD7
22 R_TXDx/INTx_y/PD6
23 PGCC/R_TXDx/INTx_y/PD5
24 SCL0F/TO1_CH1B/TI1_CH1B/INTx_y/PD4
25 SDA0F/R_TXDx/TO2_CH1B/TI2_CH1B/INTx_y/PD3
26 R_TXDx/INTx_y/PD2
27 INTx_y/PD1
28 SDA0E/TI6_ETR/INTx_y/PD0
29 SCL0E/TO6_CH4/TI6_CH4/INTx_y/PC15
30 TO2_CH1A/TI2_CH1A/INTx_y/PC14
31 TO3_CH1A/TI3_CH1A/INTx_y/PC13
32 SCL0D/TO6_CH3/TI6_CH3/INTx_y/PC12
33 PC11/INTx_y/SDA0D
34 PC10/INTx_y/TI0_ETR/SPI1A_CS
35 PC9/INTx_y
36 PC8/INTx_y
37 PC7/INTx_y
38 PC6/INTx_y/R_TXDx/PWM0B
39 PC5/INTx_y/TI6_CH2/TO6_CH2/R_TXDx/SPI1A_MISO
40 PC4/INTx_y/TI7_CH1/TO7_CH1/R_TXDx/SPI1A_MOSI
41 PC3/INTx_y/TI6_CH1/TO6_CH1/PWM0C/SPI1A_CLK/SCL0C
42 PC2/INTx_y/TI5_CH1/TO5_CH1/R_TXDx/SCL0B/SDA0C
43 PC1/INTx_y/TI4_CH1/TO4_CH1/R_TXDx/SDA0B
44 PC0/INTx_y/TI3_CH1/TO3_CH1/R_TXDx/PGDB
45 PB15/INTx_y/TI2_CH1/TO2_CH1/R_TXDx/SDA0A/PGCB
46 PB14/INTx_y/TI1_CH1/TO1_CH1/SCL0A
47 PB13/INTx_y/PWM1C/R_TXDx/SPI0A_CS
48 PB12/ADC17/INTx_y/R_TXDx/SPI0A_MISO
49 PB11/ADC16/INTx_y/TI0_CH4/TO0_CH4/SPI0A_MOSI
50 PB10/ADC15/INTx_y/TI0_BKIN2/SPI0A_CLK
51 PB9/ADC14/INTx_y/TI0_BKIN1
52 PB8/ADC13/INTx_y/TO0_CH3N
53 PB7/ADC12/INTx_y/TI0_CH3/TO0_CH3
54 PB6/ADC11/INTx_y/TO0_CH2N/R_TXDx
55 PB5/ADC10/INTx_y/TI0_CH2/TO0_CH2/R_TXDx/SPI0B_MISO
56 PB4/ADC09/INTx_y/TO0_CH1N/SPI0B_MOSI
57 PB3/ADC08/INTx_y/TI0_CH1/TO0_CH1
58 PB2/ADC07/INTx_y/SCL0H
59 PB1/ADC06/INTx_y/SDA0H/R_TXDx
60 PB0/ADC05/INTx_y/R_TXDx
61 PA15/INTx_y/TO0_CH1A/R_TXDx/SPI0B_CLK
62 PA14/ADC04/INTx_y/TI0_CH1A/R_TXDx/SPI0B_CS
63 PA13/ADC03/INTx_y/PWM0A
64 PA12/ADC02/INTx_y/PWM1A

Figure 1.7 LQFP64 package pin diagram

Note: The INT function can be mapped to INTx_y（x=0/1/2/3，y=0/1/2/3）, and the UART function can be mapped to R_TXDx（UART 的 RXD/TXD， x=0/1/2/3/4）. For more information, refer to "GPIO pin multiplexing configuration Table".

.

1.8. Pin description

BF7707AM64-LJTB	BF7707AM64-LJTA	BF7707AM52-LJTB	BF7707AM44-LJTX	Function description
1	1	2	44	Default function: GPIO <PA11> Other functions: ADC01: ADC channel 01 INTx_y: External interrupt
2	2	-	-	Default function: GPIO <PA10> Other functions: INTx_y: External interrupt
3	3	-	-	Default function: GPIO <PA9> Other functions: ADC00: ADC channel 00 INTx_y: External interrupt TI4_CH1A: TIM4 input capture channel 1 TO4_CH1A: TIM4 output compare channel 1
4	4	3	1	Default function: GPIO <PA8> Other functions: INTx_y: External interrupt TI1_CH1A: TIM1 input capture channel 1 TO1_CH1A: TIM1 output compare channel 1
5	5	4	2	Default function: GPIO <PA7> Other functions: INTx_y: External interrupt R_TXDx: Serial port transmission/reception SWDIO: Data input/output COM7: Large irrigation current port PGDA: Programming port
6	6	5	3	Default function: GPIO <PA6> Other functions: INTx_y: External interrupt R_TXDx: Serial port transmission/reception SWCLK: Clock signal COM6: Large irrigation current port PGCA: Programming port
7	7	6	4	Default function: GPIO <PA5> Other functions: INTx_y: External interrupt XTALA_IN: External crystal input EXCLKA_IN: External oscillator input

				COM5: Large irrigation current port
8	8	7	5	Default function: GPIO <PA4> Other functions: INTx_y: External interrupt XTALA_OUT: External crystal output COM4: Large irrigation current port
9	9	8	6	Default function: GPIO <PA3> Other functions: INTx_y: External interrupt COM3: Large irrigation current port
10	10	9	7	Default function: GPIO <PA2> Other functions: INTx_y: External interrupt XTALB_IN: External crystal input EXCLKB_IN: External oscillator input COM2: Large irrigation current port
11	11	10	8	Default function: GPIO <PA1> Other functions: INTx_y: External interrupt XTALB_OUT: External crystal output COM1: Large irrigation current port
12	12	11	9	Default function: GPIO <PA0> Other functions: INTx_y: External interrupt COM0: Large irrigation current port
13	13	12	10	Default function: GND<VSS>
14	14	-	-	Default function: IOVSS
15	15	13	11	Default function: Power supply<VCC>
16	16	-	-	Default function: IOVCC
17	17	14	12	Default function: GPIO <PD11> Other functions: INTx_y: External interrupt SCL0G: Serial clock line ofIIC
18	18	15	13	Default function: GPIO <PD10> Other functions: INTx_y: External interrupt SDA0G: Serial data line ofIIC
19	19	16	14	Default function: GPIO <PD9> Other functions: INTx_y: External interrupt TI3_CH1B: TIM3 input capture channel 1 TO3_CH1B: TIM3 output compare channel 1 R_TXDx: Serial port transmission/reception
20	20	17	15	Default function: GPIO <PD8> Other functions: INTx_y: External interrupt
21	21	18	16	Default function: GPIO <PD7> Other functions: INTx_y: External interrupt TI5_CH1A: TIM5 input capture channel 1 TO5_CH1A: TIM5 output compare channel 1 R_TXDx: Serial port transmission/reception

				PWM1B: PWM1 output port PGDC: Programming port
22	22	19	-	Default function: GPIO <PD6> Other functions: INTx_y: External interrupt R_TXDx: Serial port transmission/reception
23	23	20	-	Default function: GPIO <PD5> Other functions: INTx_y: External interrupt R_TXDx: Serial port transmission/reception PGCC: Programming port
24	24	21	-	Default function: GPIO <PD4> Other functions: INTx_y: External interrupt TI1_CH1B: TIM1 input capture channel 1 TO1_CH1B: TIM1 output compare channel 1 SCL0F: Serial clock line ofIIC
25	25	22	-	Default function: GPIO <PD3> Other functions: INTx_y: External interrupt TI2_CH1B: TIM2 input capture channel 1 TO2_CH1B: TIM2 output compare channel 1 R_TXDx: Serial port transmission/reception SDA0F: Serial data line ofIIC
26	26	23	17	Default function: GPIO <PD2> Other functions: INTx_y: External interrupt R_TXDx: Serial port transmission/reception
27	27	24	18	Default function: GPIO <PD1> Other functions: INTx_y: External interrupt
28	28	25	19	Default function: GPIO <PD0> Other functions: INTx_y: External interrupt TI6_ETRL: TIM6 external trigger pin SDA0E: Serial data line ofIIC
29	29	26	20	Default function: GPIO <PC15> Other functions: INTx_y: External interrupt TI6_CH4: TIM6 input capture channel 4 TO6_CH4: TIM6 output compare channel 4 SCL0E: Serial clock line ofIIC
30	30	-	-	Default function: GPIO <PC14> Other functions: INTx_y: External interrupt TI2_CH1A: TIM2 input capture channel 1 TO2_CH1A: TIM2 output compare channel 1
31	31	-	-	Default function: GPIO <PC13> Other functions: INTx_y: External interrupt TI3_CH1A: TIM3 input capture channel 1 TO3_CH1A: TIM3 output compare channel 1

32	32	27	21	Default function: GPIO <PC12> Other functions: INTx_y: External interrupt TI6_CH3: TIM6 input capture channel 3 TO6_CH3: TIM6 output compare channel 3 SCL0D: Serial clock line ofIIC
33	33	28	22	Default function: GPIO <PC11> Other functions: INTx_y: External interrupt SDA0D: Serial data line ofIIC
34	34	29	23	Default function: GPIO <PC10> Other functions: INTx_y: External interrupt TI0_ETR: TIM0 external trigger pin SPI1A_CS: SPI chip selection signal
35	35	-	-	Default function: GPIO <PC9> Other functions: INTx_y: External interrupt
36	36	-	-	Default function: GPIO <PC8> Other functions: INTx_y: External interrupt
37	37	-	-	Default function: GPIO <PC7> Other functions: INTx_y: External interrupt
38	38	-	-	Default function: GPIO <PC6> Other functions: INTx_y: External interrupt R_TXDx: Serial port transmission/reception PWM0B: PWM0 output port
39	39	30	24	Default function: GPIO <PC5> Other functions: INTx_y: External interrupt TI6_CH2: TIM6 input capture channel 2 TO6_CH2: TIM6 output compare channel 2 R_TXDx: Serial port transmission/reception SPI1A_MISO: SPI master data input
40	40	31	25	Default function: GPIO <PC4> Other functions: INTx_y: External interrupt TI7_CH1: TIM7 input capture channel 1 TO7_CH1: TIM7 output compare channel 1 R_TXDx: Serial port transmission/reception SPI1A_MOSI: SPI master data output
41	41	32	26	Default function: GPIO <PC3> Other functions: INTx_y: External interrupt TI6_CH1: TIM6 input capture channel 1 TO6_CH1: TIM6 output compare channel 1 SCL0C: Serial clock line ofIIC SPI1A_CLK: SPI clock line PWM0C: PWM0 output port
42	42	33	27	Default function: GPIO <PC2>

				Other functions: INTx_y: External interrupt TI5_CH1: TIM5 input capture channel 1 TO5_CH1: TIM5 output compare channel 1 R_TXDx: Serial port transmission/reception SDA0C: Serial data line ofIIC SCL0B: Serial clock line ofIIC
43	43	34	28	Default function: GPIO <PC1> Other functions: INTx_y: External interrupt TI4_CH1: TIM4 input capture channel 1 TO4_CH1: TIM4 output compare channel 1 R_TXDx: Serial port transmission/reception SDA0B: Serial data line ofIIC
44	44	35	29	Default function: GPIO <PC0> Other functions: INTx_y: External interrupt TI3_CH1: TIM3 input capture channel 1 TO3_CH1: TIM3 output compare channel 1 R_TXDx: Serial port transmission/reception PGDB: Programming port
45	45	36	30	Default function: GPIO <PB15> Other functions: INTx_y: External interrupt TI2_CH1: TIM2 input capture channel 1 TO2_CH1: TIM2 output compare channel 1 R_TXDx: Serial port transmission/reception SDA0A: Serial data line ofIIC PGCB: Programming port
46	46	37	31	Default function: GPIO <PB14> Other functions: INTx_y: External interrupt TI1_CH1: TIM1 input capture channel 1 TO1_CH1: TIM1 output compare channel 1 SCL0A: Serial clock line ofIIC
47	47	38	32	Default function: GPIO <PB13> Other functions: INTx_y: External interrupt R_TXDx: Serial port transmission/reception SPI0A_CS: SPI chip selection signal PWM1C: PWM1 output port
48	48	39	33	Default function: GPIO <PB12> Other functions: ADC17: ADC channel 17 INTx_y: External interrupt R_TXDx: Serial port transmission/reception SPI0A_MISO: SPI master data input
49	49	40	34	Default function: GPIO <PB11> Other functions: ADC16: ADC channel 16

				INTx_y: External interrupt TI0_CH4: TIM0 input capture channel 4 TO0_CH4: TIM0 output compare channel 4 SPI0A_MOSI: SPI master data output
50	50	41	35	Default function: GPIO <PB10> Other functions: ADC15: ADC channel 15 INTx_y: External interrupt TI0_BKIN2: TIM0 brake input pin SPI0A_CLK: SPI clock line
51	51	42	36	Default function: GPIO <PB9> Other functions: ADC14: ADC channel 14 INTx_y: External interrupt TI0_BKIN1: TIM0 brake input pin
52	52	43	37	Default function: GPIO <PB8> Other functions: ADC13: ADC channel 13 INTx_y: External interrupt TO0_CH3N: TIM0 complementary output compare channel 3
53	53	44	38	Default function: GPIO <PB7> Other functions: ADC12: ADC channel 12 INTx_y: External interrupt TI0_CH3: TIM0 input capture channel 3 TO0_CH3: TIM0 output compare channel 3
54	54	45	39	Default function: GPIO <PB6> Other functions: ADC11: ADC channel 11 INTx_y: External interrupt TO0_CH2N: TIM0 complementary output compare channel 2 R_TXDx: Serial port transmission/reception
55	55	46	40	Default function: GPIO <PB5> Other functions: ADC10: ADC channel 10 INTx_y: External interrupt TI0_CH2: TIM0 input capture channel 2 TO0_CH2: TIM0 output compare channel 2 R_TXDx: Serial port transmission/reception SPI0B_MISO: SPI master data input
56	56	47	41	Default function: GPIO <PB4> Other functions: ADC09: ADC channel 09 INTx_y: External interrupt TO0_CH1N: TIM0 complementary output compare channel 1 SPI0B_MOSI: SPI master data output

57	57	48	-	Default function: GPIO <PB3> Other functions: ADC08: ADC channel 08 INTx_y: External interrupt TI0_CH1: TIM0 input capture channel 1 TO0_CH1: TIM0 output compare channel 1
58	58	-	-	Default function: GPIO <PB2> Other functions: ADC07: ADC channel 07 INTx_y: External interrupt SCL0H: Serial clock line ofIIC
59	59	49	-	Default function: GPIO <PB1> Other functions: ADC06: ADC channel 06 INTx_y: External interrupt R_TXDx: Serial port transmission/reception SDA0H: Serial data line ofIIC
60	60	50	-	Default function: GPIO <PB0> Other functions: ADC05: ADC channel 05 INTx_y: External interrupt R_TXDx: Serial port transmission/reception
61	61	51	42	Default function: GPIO <PA15> Other functions: INTx_y: External interrupt TO0_CH1A: TIM0 output compare channel 1 R_TXDx: Serial port transmission/reception SPI0B_CLK: SPI clock line
62	62	52	43	Default function: GPIO <PA14> Other functions: ADC04: ADC channel 04 INTx_y: External interrupt TI0_CH1A: TIM0 input capture channel 1 R_TXDx: Serial port transmission/reception SPI0B_CS: SPI chip selection signal
63	63	1	-	Default function: GPIO <PA13> Other functions: ADC03: ADC channel 03 INTx_y: External interrupt PWM0A: PWM0 output port
64	64	-	-	Default function: GPIO <PA12> Other functions: ADC02: ADC channel 02 INTx_y: External interrupt PWM1A: PWM1 output port

Chapter 2 Electrical characteristics

2.1. Limit parameters

Symbol	Parameter	Test Condition		Min	Typical	Max	Unit
		VCC	Condition				
VCC	Supply voltage when operating	-	-	VSS+2.6	-	VSS+5.5	V
T_{STG}	Storage temperature	-	-	-40	-	125	℃
Ta	Operating temperature	-	-	-40	-	105	℃
Vin	I/O input voltage	-	-	VSS-0.5	-	VCC+0.5	V
I_{OLA}	I_{OL} total current	-	-	130			mA
I_{OHA}	I_{OH} total current	-	-	-130			mA
ESD(HBM)	Port electrostatic discharge voltage	-	-	-6	-	6	kV

Note: Exceeding the range specified by the limit parameters will cause damage to the chip. It is impossible to predict the working state of the chip outside the above marked range, and if it works under the conditions outside the marked range for a long time, it may affect the reliability of the chip.

2.2. AC characteristics

Symbol	Parameter	Test Conditions		Min	Typical	Max	Unit
		VCC	Temperature				
f_{RC1M}	Internal high-speed RC oscillator	5V	-20℃~65℃	-1%	1	+1%	MHz
			-40℃～105℃	-3%	1	+3%	
		2.6V~5.5V	25℃	-1%	1	+1%	
			-40℃～105℃	-3%	1	+3%	
f_{HCLK}	System clock	5V	-20℃~65℃	-1%	48/32/24/12	+1%	MHz
			-40℃～105℃	-3%	48/32/24/12	+3%	
		2.6V~5.5V	25℃	-1%	48/32/24/12	+1%	
			-40℃～105℃	-3%	48/32/24/12	+3%	
f_{LIRC}	Internal low speed RC oscillator	5V	25℃	-10%	32	+10%	kHz
			-40℃～105℃	-25%	32	+25%	
		2.6V~5.5V	25℃	-20%	32	+20%	
			-40℃～105℃	-30%	32	+30%	

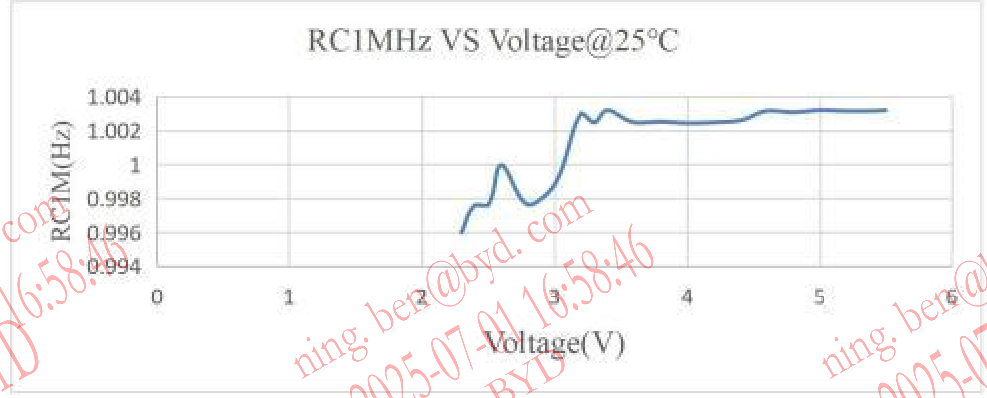


Figure 2.1 f_{RC1M} voltage graph

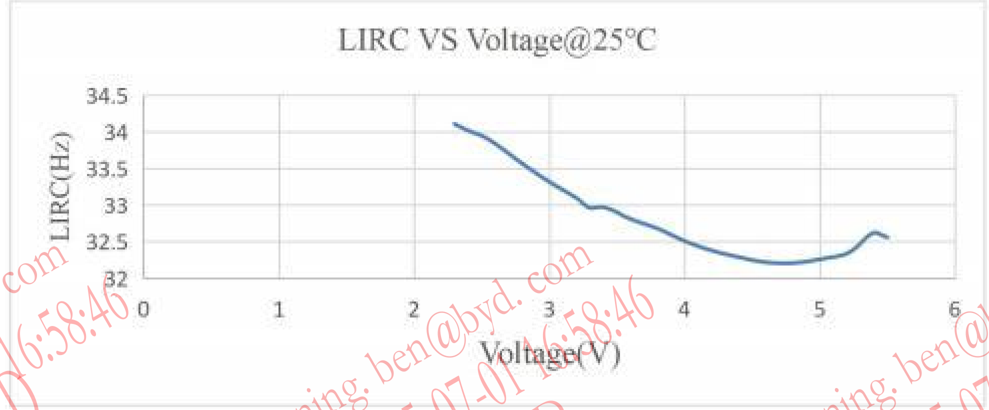


Figure 2.2 f_{LIRC} voltage graph

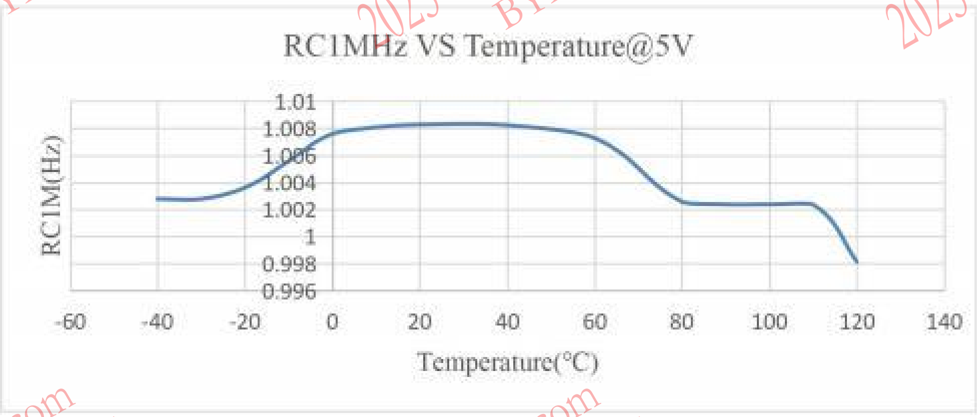


Figure 2.3 f_{RC1M} temperature graph

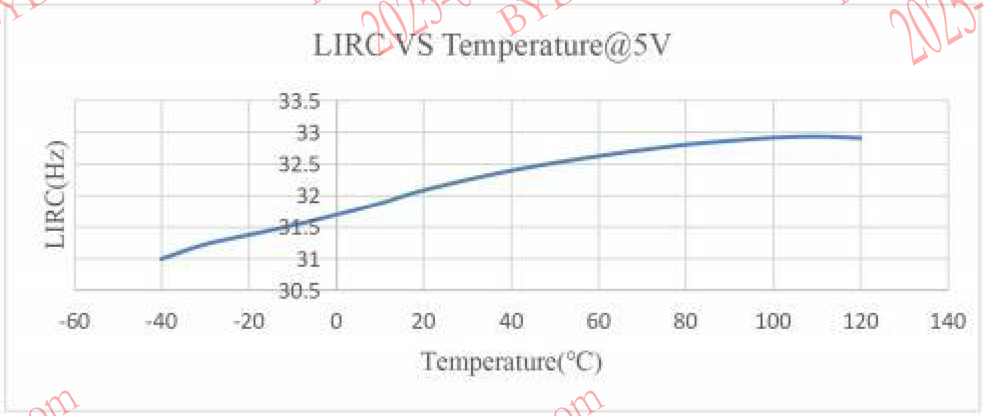


Figure 2.4 f_{LIRC} temperature graph

2.3. DC characteristics

Ta=25℃

Symbol	Parameter	Test Condition		Min	Typical	Max	Unit
		VCC	Condition				
VCC	Operating voltage	-	-	2.6	-	5.5	V
I_{OP}	Active mode current	3.3V	f_{RC1M}/PLL on, f_{HCLK}=48MHz, f_{LIRC} on, all peripherals off	-	5.4	7	mA
		5V		-	5.5	7.2	
		3.3V	f_{RC1M}/PLL on, f_{HCLK}=32MHz, f_{LIRC} on, all peripherals off	-	4.3	5.6	
		5V		-	4.4	5.7	
		3.3V	f_{RC1M}/PLL on, f_{HCLK}=24MHz, f_{LIRC} on, all peripherals off	-	3.9	5.1	
		5V		-	4.0	5.2	
		3.3V	f_{RC1M}/PLL on, f_{HCLK}= 12MHz, f_{LIRC} on, all peripherals off	-	3.0	3.9	
		5V		-	3.1	4.0	
I_{STB0}	Idle mode 0 current	3.3V	f_{RC1M}/PLL on, f_{HCLK} off, f_{LIRC} on, all peripherals off	-	1.8	2.3	mA
		5V		-	1.9	2.5	
I_{STB1}	Idle mode 1 current	3.3V	f_{RC1M}/PLL/f_{HCLK} on , f_{LIRC} on, all peripherals off	-	6.6	-	µA
		5V		-	5.6	-	
V_{IL}	Input low voltage	2.6~5.5V	-	-	-	0.3*VCC	V
V_{IH}	Input high voltage	2.6~5.5V	-	0.7*VCC	-	-	V
V_{INTL}	INT input low voltage	2.6~5.5V	-	-	-	0.3*VCC	V
V_{INTH}	INT input high voltage	2.6~5.5V	-	0.7*VCC	-	-	V
V_{OL}	Output low voltage	5V	I_{OL}=66mA	-	-	0.1*VCC	V
V_{OH}	Output high voltage	5V	I_{OH}= 16mA	0.9*VCC	-	-	V
I_{OL}	IO sink current	5V	V_{OL}=0.1VCC	49	66	91	mA

I_{OH}	IO source current	5V	V_{OH}=0.9VCC	12	16	22	mA
I_{COM}	PA0-PA7 high current	5V	V_{OL}=0. 1VCC	-	130	-	mA
I_{Leak}	Input leakage current	5V	-	-	1	5	µA
R_{PH}	IO/RST_N internal pull-up resistor	5V	-	8	12	16	kΩ

Symbol	Parameter	Test Condition		Min	Typical	Max	Unit
		VCC	Condition				
I_{ADC}	ADC operating current	5V	f_{HCLK}=48MHz, ADC clock select 12MHz, enable ADC, open one channel	-	4.4	-	mA
I_{PWM}	PWM operating current	5V	f_{HCLK}=48MHz, enable PWM0, output 4kHz waveform	-	0.35	-	mA
I_{LVDT}	LVDT operating current	5V	f_{HCLK}=48MHz, enable LVDT, configure detection parameter 2.7V	-	5.0	-	mA

2.4. ADC characteristics

Ta=25℃

Symbol	Parameter	Test Condition		Min	Typical	Max	Unit
		VCC	Condition				
V_{ADC}	Supply voltage	-	-	2.6	-	5.5	V
N_R	Precision	-	-	-	9	10	Bit
V_{ADCI}	ADC input voltage	-	-	VSS		V_{REF}	V
R_{ADCI}	ADC input resistance	5V	-	0.7	0.8	3.7	kΩ
I_{ADC}	ADC operating current	5V	f_{HCLK}=48MHz, ADC clock select 12MHz, enable ADC, open one channel	-	4.4		mA
I_{ADCI}	A/D input current	-	-	-	-	1	µA
DNL	Differential nonlinearity error	5V	-	-	±4	±6	LSB
INL	Integral nonlinearity error	5V	-	-	±4	±6	LSB

t2	ADC sampling time	-	-	0.212	-	-	µs
t_{ADC}	ADC conversion time	-	-	0.583	-	-	µs
RESO	Resolution	-	-	8/12			Bit
N_{ADC}	Input channel	-	-	-	-	18	Channel

2.5. Temperature sensor characteristics

Symbol	Parameter	Test Condition		Min	Typical	Max	Unit
		VCC	Condition				
Avg_slope	Temperature mean slope	-	-	-	3.227	-	mV/℃
T_L	Linearity of the temperature sensor	-	-	-	-	±10℃	℃

2.6. Memory characteristics

Ta=25℃

Symbol	Parameter	Test Condition		Min	Typical	Max	Unit
		VCC	Condition				
EC	Erase and write times	5V	-	100k	-	-	Cycle
t_{RET}	Data retention period	-	-	100	-	-	Year
t_{PROG}	Word programming time	5V	-	-	46	-	µs
t_{ERASE}	Page erase time	5V	-	-	4.5	-	ms
t_{MERASE}	Whole chip erase time	5V	-	-	30	-	ms
I_{ERASE}	Page erase current	5V	f_{HCLK}=48MHz, in the while, the main memory block is page erased, turn off all other functions. Total chip power consumption.	-	6.3	-	mA
I_{PROG}	Programming current	5V	f_{HCLK}=48MHz, in the while, a word is written to the main memory block, turn off all other functions. Total chip power consumption.	-	6.3	-	mA

Chapter 3 System control (SYS_CTRL)

3.1. Clock description

The clock control module mainly controls the system clock and peripheral clock. It can configure different system clock frequency divisions, and can enable or disable peripheral clocks.

Clock source:

- Internal high-speed RC oscillator: RC 1MHz
- Internal low-speed RC oscillator: LIRC 32kHz
- External crystal oscillator XTAL: Passive crystal oscillator (32768Hz/4MHz/8MHz) and active crystal oscillator (1MHz~48MHz)
- RC1M frequency multiplication to get PLL clock: PLL frequency multiplied clock

The BF7707AMXX series clocks are defined as follows:

- **RC1M:** Internal high-speed RC oscillator, frequency multiplication to get PLL48M clock, it provides the clock for the core and its peripheral equipment, and manages the low-power mode by clock gating and multistage prescaler.
- **PLL48M:** System clock source, TIM/ADC/PWM/UART module working clock, and external interrupt filtering clock source.
- **HCLK:** System clock, AHB bus peripheral clock, 48MHz/32MHz/24MHz/12MHz frequency optional.
- **PLL48M divide by 2:** FLASH programming clock, IIC digital filter clock and IIC master working clock(iic_clk).
- **LIRC 32kHz:** Internal low-speed RC oscillator, WDT/TIM/PWM module clock source, and external interrupt filtering clock source.
- **XTAL:** External crystal oscillator, TIM/PWM module clock source.
- **SCL:** The highest 1MHz clock is used as the IIC slave communication clock source.
- **SPI_CLK:** The highest 4MHz clock is used as the slave SPI communication clock source.
- **PGC:** The clock within 5M is used as the FLASH programming clock source.
- **SWCLK:** Clock within 20M is used as the clock source for DEBUG and FLASH programming.

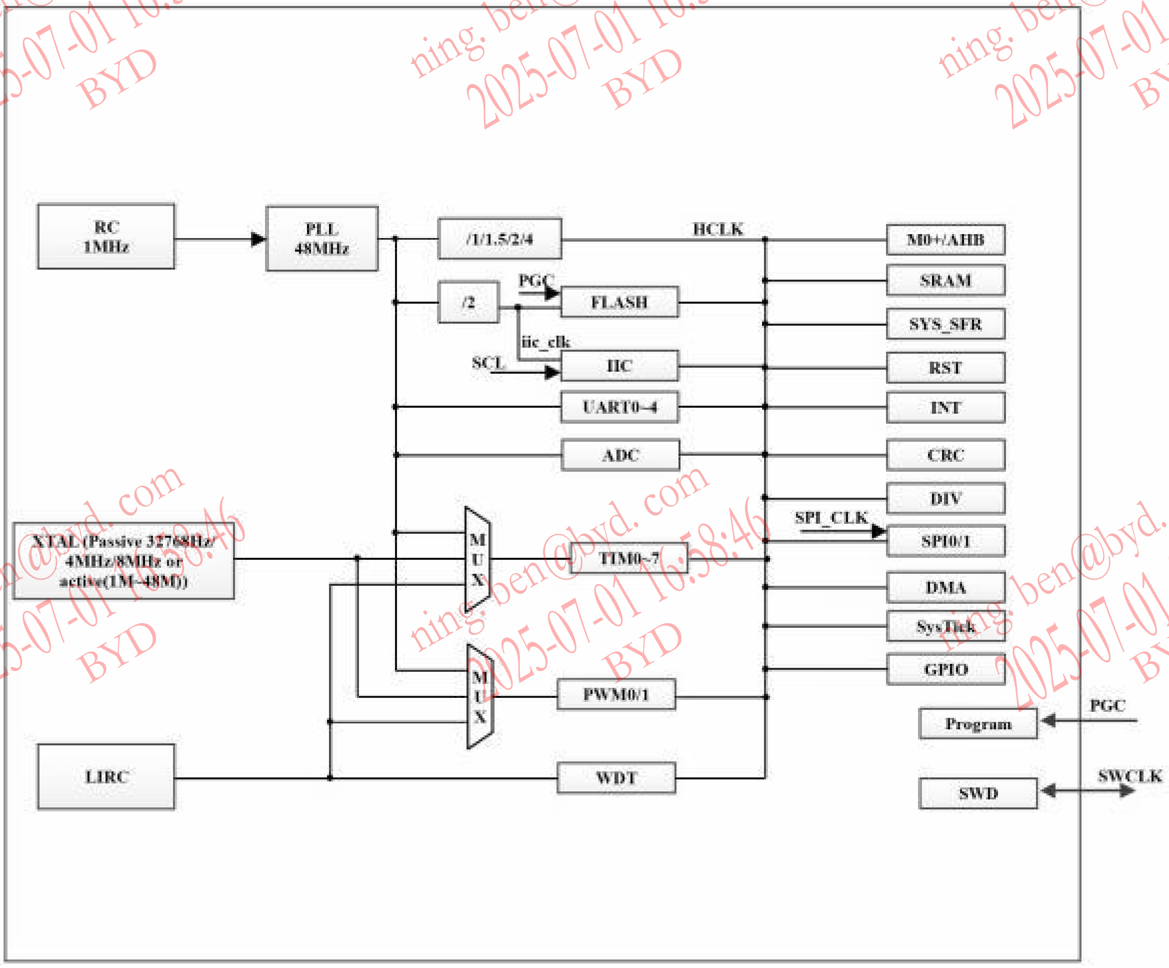


Figure 3.1 Clock block diagram

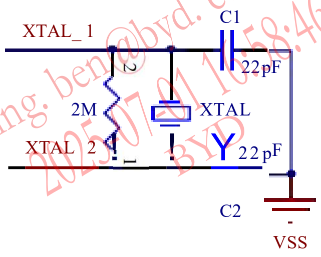


Figure 3.2 External passive crystal oscillator circuit reference

Note:

1. The external passive crystal oscillator circuit is for reference only, the actual parameters refer to the crystal oscillator specifications;
2. The excitation power of external passive 32768Hz crystal oscillator is recommended to be greater than 1μW;
3. The external passive 32768Hz crystal oscillator is recommended to have a parallel resistance of 2MΩ;
4. The external passive 4M/8M crystal oscillator is recommended to have a parallel resistance of 1MΩ;

3.2. Registers

Base address: 0x5000 0000

Address offset	Register	Description
0x00	CLK_CFG	Clock configuration register
0x04	RCU_EN	Peripheral module clock control register
0x08	RST_STATE	Reset flag register
0x0C	XTAL_HS_SEL	Hysteresis voltage selection of comparator in crystal oscillator register
0x10	ANA_CFG	Analog module switch register
0x14	BOR_CFG	BOR configuration register
0x18	WAKE_CFG	System wake-up configuration register
0x1C	LVDT_CTRL	LVDT control register
0x20	LVDT_STATE	LVDT status register
0x2C	COM_IO_SEL	COM port selection configuration register
0x34	ODRAIN_EN	Open drain output enable register
0x38	IIC_IO_CTRL	IIC control register
0x40	RCU_RSTEN	Peripheral module reset enable register
0x44	TIM_CLK_SEL	TIM clock selection register
0x48	CLK_SEL_RDY	Clock switch completion flag register
0x4C	PWM_CLK_SEL	PWM clock selection register
0x50	HSPEED_EN	SPI high-speed mode enable register
0x54	ADC_TRIG_SEL	ADC hardware trigger source selection register

3.2.1. Clock registers

3.2.1.1. Clock configuration register (CLK_CFG)

Address offset: 0x00

Reset value: 0x0000 0002

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

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

31:2	-	Reserved
1:0	HCLK_SEL	HCLK clock frequency division selection register 00: 48MHz 01: 32MHz 10: 24MHz 11: 12MHz

3.2.1.2. Peripheral module clock control register (RCU_EN)

Address offset: 0x04

Reset value: 0x0000 0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.				DIV_ CLKEN	GPIOD_ CLKEN	GPIOC_ CLKEN	GPIOB_ CLKEN	GPIOA_ CLKEN	DMA_ CLKEN	CRC_ CLKEN	ADC_ CLKEN	WDT_ CLKEN	TIM7_ CLKEN	TIM6_ CLKEN	TIM5_ CLKEN
				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
TIM4_ CLKEN	TIM3_ CLKEN	TIM2_ CLKEN	TIM1_ CLKEN	TIM0_ CLKEN	PWM1_ CLKEN	PWM0_ CLKEN	IIC_ CLKEN	UART4_ CLKEN	UART3_ CLKEN	UART2_ CLKEN	UART1_ CLKEN	UART0_ CLKEN	SPI1_ CLKEN	SPI0_ CLKEN	Res.
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	

31:28	-	Reserved
27	DIV_CLKEN	DIV module operation enable 1: Work 0: Off, the default is 0
26	GPIOD_CLKEN	GPIOD module operation enable 1: Work 0: Off, the default is 0
25	GPIOC_CLKEN	GPIOC module operation enable 1: Work, 0: Off, the default is 0
24	GPIOB_CLKEN	GPIOB module operation enable 1: Work 0: Off, the default is 0
23	GPIOA_CLKEN	GPIOA module operation enable 1: Work 0: Off, the default is 0
22	DMA_CLKEN	DMA(include DMA_SRAM/DMAMUX) module operation enable 1: Work 0: Off, the default is 0
21	CRC_CLKEN	CRC module operation enable 1: Work 0: Off, the default is 0
20	ADC_CLKEN	ADC module operation enable 1: Work 0: Off, the default is 0
19	WDT_CLKEN	WDT module operation enable 1: Work 0: Off, the default is 0
18	TIM7_CLKEN	TIM7 module operation enable

		1: Work 0: Off, the default is 0
17	TIM6_CLKEN	TIM6 module operation enable 1: Work 0: Off, the default is 0
16	TIM5_CLKEN	TIM5 module operation enable 1: Work 0: Off, the default is 0
15	TIM4_CLKEN	TIM4 module operation enable 1: Work 0: Off, the default is 0
14	TIM3_CLKEN	TIM3 module operation enable 1: Work 0: Off, the default is 0
13	TIM2_CLKEN	TIM2 module operation enable 1: Work 0: Off, the default is 0
12	TIM1_CLKEN	TIM1 module operation enable 1: Work 0: Off, the default is 0
11	TIM0_CLKEN	TIM0 module operation enable 1: Work 0: Off, the default is 0
10	PWM1_CLKEN	PWM1 module operation enable 1: Work 0: Off, the default is 0
9	PWM0_CLKEN	PWM0 module operation enable 1: Work 0: Off, the default is 0
8	IIC_CLKEN	IIC module operation enable 1: Work 0: Off, the default is 0
7	UART4_CLKEN	UART4 module operation enable 1: Work 0: Off, the default is 0
6	UART3_CLKEN	UART3 module operation enable 1: Work 0: Off, the default is 0
5	UART2_CLKEN	UART2 module operation enable 1: Work 0: Off, the default is 0
4	UART1_CLKEN	UART1 module operation enable

		1: Work 0: Off, the default is 0
3	UART0_CLKEN	UART0 module operation enable 1: Work 0: Off, the default is 0
2	SPI1_CLKEN	SPI1 module operation enable 1: Work 0: Off, the default is 0
1	SPI0_CLKEN	SPI0 module operation enable 1: Work 0: Off, the default is 0
0	-	Reserved

3.2.2. Reset flag register (RST_STATE)

Address offset: 0x08

Reset value: 0x0000 0001

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

15:8	7	6	5	4	3	2	1	0
Res.	PROG	ADDROF	Res.	LOCKUP	SYSRST	WDTRST	BOR	POR
	RC_W0	RC_W0		RC_W0	RC_W0	RC_W0	RC_W0	RC_W0

31:8	-	Reserved
7	PROG	FLASH programming reset flag, software can write 0 to clear the corresponding flag bit 1: FLASH programming reset occurs 0: No FLASH programming reset occurs
6	ADDROF	PC pointer overflow reset flag, software can write 0 to clear the corresponding flag bit 1: PC pointer overflow reset occurs 0: No PC pointer overflow reset occurs
5	-	Reserved
4	LOCKUP	CPU lock reset flag, software can write 0 to clear the corresponding flag bit 1: CPU lock reset occurred 0: No CPU lock reset occurs
3	SYSRST	CPU software reset flag, software can write 0 to clear the corresponding flag bit 1: CPU software reset occurred 0: No CPU software reset occurs
2	WDTRST	Watchdog reset flag, software can write 0 to clear the corresponding flag bit 1: Watchdog reset occurred

		0: No watchdog reset occurs
1	BOR	Power-down reset flag, software can write 0 to clear the corresponding flag bit 1: Power-down reset occurred 0: No power-down reset occurs
0	POR	Power-on reset flag, software can write 0 to clear the corresponding flag bit 1: Power-on reset occurred 0: No power-on reset occurred

3.2.3. Hysteresis voltage selection of comparator in crystal oscillator register (XTAL_HS_SEL)

Address offset: 0x0C

Reset value: 0x0000 0000

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

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

31:2	-	Reserved
1:0	HS_SEL	Hysteresis voltage selection of comparator in crystal oscillator 00: 300mV 01: 400mV 10: 500mV 11: 600mV

3.2.4. Analog module switch register (ANA_CFG)

Address offset: 0x10

Reset value: 0x0000 1850

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

15:13	12	11	10	9	8	7	6	5	4	3:2	1	0
Res.	PD_TEMP	PD_ADC	VREF_SEL	VREF_VOL_SEL	VREF_IN_ADC_SEL		PD_ADC_IN_VREF	XTAL_HFR_SEL	XTAL_SEL	XTAL_BYP_SEL	XTAL_EN_SEL	
	RW	RW	RW	RW	RW		RW	RW	RW	RW	RW	

31:13	-	Reserved
12	PD_TEMP	Analog ADC TEMP detection channel shutdown control register 0: Module works normally 1: Module does not work

11	PD_ADC	Analog ADC shutdown control register 0: ADC module works normally 1: ADC module does not work
10	VREF_SEL	Select output signal 0: Select VCC as the reference voltage 1: Select the output of the ADC_VREF module as the reference voltage
9	VREF_VOL_SEL	ADC_VREF output mode control signal, ADCK frequency must be ≤3MHz: 0: 2V is used as the ADC reference voltage 1: 4V is used as the reference voltage (VCC greater than 4.5V)
8:7	VREF_IN_ADC_SEL	Voltage input to ADC18 ADC18_VREF: 00: 1.3801V 01: 2.2832V 10: 3.1517V 11: 4.135V The default is 00
6	PD_ADC_IN_VREF	Analog ADC VREF detection channel shutdown control register 0: Module works normally 1: Module does not work
5	XTAL_HFR_SEL	Current configuration selection of analog high frequency crystal oscillator circuit 0: 175μA (4MHz crystal oscillator recommended) 1: 300μA (8MHz crystal oscillator recommended)
4	XTAL_SEL	When passive crystal oscillator circuit is available 0: Select 32 K 1: Select 4M/8M, default value is 1
3:2	XTAL_BYP_SEL	Active crystal oscillator path selection 00: Path closed, do not use active crystal oscillator 01: Select path A (PA5) for input 10: Select path B (PA2) for input 11: Reserved Default value is 00 Note: If any active crystal path is selected, the passive crystal oscillator starting circuit is closed at the same time.
1:0	XTAL_EN_SEL	Passive crystal oscillator path selection 00: Passive crystal oscillator circuit is closed, do not use passive crystal oscillator 01: Select channel A (PA5 and PA4). 10: Select channel B (PA2 and PA1) for input 11: Reserved When XTAL_BYP_SEL[1:0] = 00 and XTAL_EN_SEL[1:0]!= 00, turn on the passive crystal oscillator circuit, default value is 00. Note: Enabling passive crystal oscillator must be configured without enabling active crystal oscillator

3.2.5. BOR configuration register (BOR_CFG)

Address offset: 0x14

Reset value: 0x0000 0002

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

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

31:1	-	Reserved
0	DELAY_SEL	BOR's power down delay configuration register 0: Delay selection 0 1: Delay selection 1 See table "BOR threshold and delay selection"

3.2.6. System wake-up configuration register (WAKE_CFG)

Address offset: 0x18

Reset value: 0x0000 0007

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2:0
Res.													PLL_WAKE_TIM
													RW

31:3	-	Reserved
2:0	PLL_WAKE_TIM	Wake up PLL timing time 000: 0.2ms 001: 0.3ms 010: 0.4ms 011: 0.5ms 100: 0.6ms 101: 0.7ms 110: 0.9ms 111: 1ms

3.2.7. LVDT control register (LVDT_CTRL)

Address offset: 0x1C

Reset value: 0x0000 0004

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

15:8	7	6	5	4	3	2	1	0
Res.	VTH_SEL			DELAY_SEL		PD_LVDT	PO_INTEN	BO_INTEN
	RW			RW		RW	RW	RW

31:8	-	Reserved
7:5	VTH_SEL	LVDT threshold selection 000: Reserved 001: 2.7V 010: 3.0V 011: 3.3V 100: 3.6V 101: 3.9V 110: 4.2V 111: 4.5V For the specific corresponding threshold voltage, see the table "LVDT threshold and delay selection"
4:3	DELAY_SEL	LVDT power-down delay configuration 00: Power-down delay 1 01: Power-down delay 2 10: Power-down delay 3 11: Power-down delay 4
2	PD_LVDT	LVDT control register 1: Off 0: On. Off by default
1	PO_INTEN	LVDT low voltage boost interrupt enable 1: Enable 0: Disable
0	BO_INTEN	LVDT low voltage buck interrupt enable 1: Enable 0: Disable

3.2.8. LVDT status register (LVDT_STATE)

Address offset: 0x20

Reset value: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.														POF	BOF
														RC_W1	RC_W1

31:2	-	Reserved
1	POF	LVDT boost interrupt flag 1: Valid 0: Invalid When this interrupt occurs, the hardware sets Bit, and the software writes 1 to clear 0
0	BOF	LVDT buck interrupt flag 1: Valid 0: Invalid When this interrupt occurs, the hardware sets Bit, and the software writes 1 to clear 0

3.2.9. COM port selection configuration register (COM_IO_SEL)

Address offset: 0x2C

Reset value: 0x0000 0000

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

15:8	7	6	5	4	3	2	1	0
Res.	COM7	COM6	COM5	COM4	COM3	COM2	COM1	COM0
	RW	RW	RW	RW	RW	RW	RW	RW

31:8	-	Reserved
7:0	COM[7:0]	COM port selection configuration register, bit[0]~bit[7] correspond to PA0~PA7 ports 1: Select COM port function 0: Select common IO port function

3.2.10. Open drain output enable register (ODRAIN_EN)

Address offset: 0x34

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															OD16
															RW

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OD15	OD14	OD13	OD12	OD11	OD10	OD9	OD8	OD7	OD6	OD5	OD4	OD3	OD2	OD1	OD0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

31:17	-	Reserved
16	OD16	PD8 port open drain output enable 1: Open drain output 0: CMOS output
15	OD15	PD9 port open drain output enable 1: Open drain output 0: CMOS output
14	OD14	PB1 port open drain output enable 1: Open drain output 0: CMOS output
13	OD13	PB2 port open drain output enable 1: Open drain output 0: CMOS output
12	OD12	PC1 port open drain output enable 1: Open drain output 0: CMOS output
11	OD11	PB14 port open drain output enable 1: Open drain output 0: CMOS output
10	OD10	PB15 port open drain output enable 1: Open drain output 0: CMOS output
9	OD9	PC2 port open drain output enable 1: Open drain output 0: CMOS output
8	OD8	PC3 port open drain output enable 1: Open drain output 0: CMOS output
7	OD7	PC11 port open drain output enable 1: Open drain output 0: CMOS output
6	OD6	PC12 port open drain output enable 1: Open drain output 0: CMOS output
5	OD5	PC15 port open drain output enable 1: Open drain output 0: CMOS output
4	OD4	PD0 port open drain output enable

		1: Open drain output 0: CMOS output
3	OD3	PD3 port open drain output enable 1: Open drain output 0: CMOS output
2	OD2	PD4 port open drain output enable 1: Open drain output 0: CMOS output
1	OD1	PD10 port open drain output enable 1: Open drain output 0: CMOS output
0	OD0	PD11 port open drain output enable 1: Open drain output 0: CMOS output

Note: When IIC is enabled, the open drain output of the corresponding port is automatically enabled, and 8 channels ofIIC mapping

3.2.11. IIC control register (IIC_IO_CTRL)

Address offset: 0x38

Reset value: 0x0000 0001

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

15:2	1	0
Res.	DFIL_SEL	AFIL_SEL
	RW	RW

31:2	-	Reserved
1	DFIL_SEL	IIC function digital filter enable 1: Enable 0: Disable. Default is 0
0	AFIL_SEL	IIC function analog filter enable 1: Enable 0: Disable. Default is 1

3.2.12. Peripheral module reset enable register(RCU_RSTEN)

Address offset: 0x40

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.				DIV RST	GPIOD RST	GPIOC RST	GPIOB RST	GPIOA RST	DMA RST	CRC RST	ADC RST	WDT RST	TIM7 RST	TIM6 RST	TIM5 RST

				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
TIM4 RST	TIM3 RST	TIM2 RST	TIM1 RST	TIM0 RST	PWM1 RST	PWM0 RST	IIC RST	UART4 RST	UART3 RST	UART2 RST	UART1 RST	UART0 RST	SPI1 RST	SPI0 RST	Res.
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	

31:28	-	Reserved
27	DIVRST	DIV module reset enable 1: Module reset 0: No reset. Write 0 to clear the reset status. Default value is 0. After the bit is written 1, the module is always in the reset state. Need to write 0 to exit the reset
26	GPIODRST	GPIOD module reset enable 1: Module reset 0: No reset. Write 0 to clear the reset status. Default value is 0. After the bit is written 1, the module is always in the reset state. Need to write 0 to exit the reset
25	GPIOCRST	GPIOC module reset enable 1: Module reset 0: No reset. Write 0 to clear the reset status. Default value is 0. After the bit is written 1, the module is always in the reset state. Need to write 0 to exit the reset
24	GPIOBRST	GPIOB module reset enable 1: Module reset 0: No reset. Write 0 to clear the reset status. Default value is 0. After the bit is written 1, the module is always in the reset state. Need to write 0 to exit the reset
23	GPIOARST	GPIOA module reset enable 1: Module reset 0: No reset. Write 0 to clear the reset status. Default value is 0. After the bit is written 1, the module is always in the reset state. Need to write 0 to exit the reset
22	DMARST	DMA(include DMA_SFR/DMAMUX) module reset enable 1: Module reset 0: No reset. Write 0 to clear the reset status. Default value is 0. After the bit is written 1, the module is always in the reset state. Need to write 0 to exit the reset
21	CRCRST	CRC module reset enable 1: Module reset 0: No reset. Write 0 to clear the reset status. Default value is 0. After the bit is written 1, the module is always in the reset state. Need to write 0 to exit the reset

20	ADCRST	ADC module reset enable 1: Module reset 0: No reset. Write 0 to clear the reset status. Default value is 0. After the bit is written 1, the module is always in the reset state. Need to write 0 to exit the reset
19	WDTRST	WDT module reset enable 1: Module reset 0: No reset. Write 0 to clear the reset status. Default value is 0. After the bit is written 1, the module is always in the reset state. Need to write 0 to exit the reset
18	TIM7RST	TIM7 module reset enable 1: Module reset 0: No reset. Write 0 to clear the reset status. Default value is 0. After the bit is written 1, the module is always in the reset state. Need to write 0 to exit the reset
17	TIM6RST	TIM6 module reset enable 1: Module reset 0: No reset. Write 0 to clear the reset status. Default value is 0. After the bit is written 1, the module is always in the reset state. Need to write 0 to exit the reset
16	TIM5RST	TIM5 module reset enable 1: Module reset 0: No reset. Write 0 to clear the reset status. Default value is 0. After the bit is written 1, the module is always in the reset state. Need to write 0 to exit the reset
15	TIM4RST	TIM4 module reset enable 1: Module reset 0: No reset. Write 0 to clear the reset status. Default value is 0. After the bit is written 1, the module is always in the reset state. Need to write 0 to exit the reset
14	TIM3RST	TIM3 module reset enable 1: Module reset 0: No reset. Write 0 to clear the reset status. Default value is 0. After the bit is written 1, the module is always in the reset state. Need to write 0 to exit the reset
13	TIM2RST	TIM2 module reset enable 1: Module reset 0: No reset. Write 0 to clear the reset status. Default value is 0. After the bit is written 1, the module is always in the reset state. Need to write 0 to exit the reset
12	TIM1RST	TIM1 module reset enable 1: Module reset

		0: No reset. Write 0 to clear the reset status. Default value is 0. After the bit is written 1, the module is always in the reset state. Need to write 0 to exit the reset
11	TIM0RST	TIM0 module reset enable 1: Module reset 0: No reset. Write 0 to clear the reset status. Default value is 0. After the bit is written 1, the module is always in the reset state. Need to write 0 to exit the reset
10	PWM1RST	PWM1 module reset enable 1: Module reset 0: No reset. Write 0 to clear the reset status. Default value is 0. After the bit is written 1, the module is always in the reset state. Need to write 0 to exit the reset
9	PWM0RST	PWM0 module reset enable 1: Module reset 0: No reset. Write 0 to clear the reset status. Default value is 0. After the bit is written 1, the module is always in the reset state. Need to write 0 to exit the reset
8	IICRST	IIC module reset enable 1: Module reset 0: No reset. Write 0 to clear the reset status. Default value is 0. After the bit is written 1, the module is always in the reset state. Need to write 0 to exit the reset
7	UART4RST	UART4 module reset enable 1: Module reset 0: No reset. Write 0 to clear the reset status. Default value is 0. After the bit is written 1, the module is always in the reset state. Need to write 0 to exit the reset
6	UART3RST	UART3 module reset enable 1: Module reset 0: No reset. Write 0 to clear the reset status. Default value is 0. After the bit is written 1, the module is always in the reset state. Need to write 0 to exit the reset
5	UART2RST	UART2 module reset enable 1: Module reset 0: No reset. Write 0 to clear the reset status. Default value is 0. After the bit is written 1, the module is always in the reset state. Need to write 0 to exit the reset
4	UART1RST	UART1 module reset enable 1: Module reset 0: No reset. Write 0 to clear the reset status. Default value is 0. After the bit is written 1, the module is always in the reset state. Need to write

		0 to exit the reset
3	UART0RST	UART0 module reset enable 1: Module reset 0: No reset. Write 0 to clear the reset status. Default value is 0. After the bit is written 1, the module is always in the reset state. Need to write 0 to exit the reset
2	SPI1RST	SPI1 module reset enable 1: Module reset 0: No reset. Write 0 to clear the reset status. Default value is 0. After the bit is written 1, the module is always in the reset state. Need to write 0 to exit the reset
1	SPI0RST	SPI0 module reset enable 1: Module reset 0: No reset. Write 0 to clear the reset status. Default value is 0. After the bit is written 1, the module is always in the reset state. Need to write 0 to exit the reset
0	-	Reserved

3.2.13. TIM clock selection register(TIM_CLK_SEL)

Address offset: 0x44

Reset value: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIM7CLK		TIM6CLK		TIM5CLK		TIM4CLK		TIM3CLK		TIM2CLK		TIM1CLK		TIM0CLK	
RW		RW		RW		RW		RW		RW		RW		RW	

31:16	-	Reserved
15:14	TIM7CLK	TIM7 clock selection: 00: Select PLL48M clock; 01: Select LIRC 32K clock 1x: Select external crystal clock XTAL, passive crystal (32768Hz/4MHz/8MHz) and active crystal (1MHz~48MHz)
13:12	TIM6CLK	TIM6 clock selection: 00: Select PLL48M clock; 01: Select LIRC 32K clock 1x: Select external crystal clock XTAL, passive crystal (32768Hz/4MHz/8MHz) and active crystal (1MHz~48MHz)
11:10	TIM5CLK	TIM5 clock selection: 00: Select PLL48M clock;

		01: Select LIRC 32K clock 1x: Select external crystal clock XTAL, passive crystal (32768Hz/4MHz/8MHz) and active crystal (1MHz~48MHz)
9:8	TIM4CLK	TIM4 clock selection: 00: Select PLL48M clock; 01: Select LIRC 32K clock 1x: Select external crystal clock XTAL, passive crystal (32768Hz/4MHz/8MHz) and active crystal (1MHz~48MHz)
7:6	TIM3CLK	TIM3 clock selection: 00: Select PLL48M clock; 01: Select LIRC 32K clock 1x: Select external crystal clock XTAL, passive crystal (32768Hz/4MHz/8MHz) and active crystal (1MHz~48MHz)
5:4	TIM2CLK	TIM2 clock selection: 00: Select PLL48M clock; 01: Select LIRC 32K clock 1x: Select external crystal clock XTAL, passive crystal (32768Hz/4MHz/8MHz) and active crystal (1MHz~48MHz)
3:2	TIM1CLK	TIM1 clock selection: 00: Select PLL48M clock; 01: Select LIRC 32K clock 1x: Select external crystal clock XTAL, passive crystal (32768Hz/4MHz/8MHz) and active crystal (1MHz~48MHz)
1:0	TIM0CLK	TIM0 clock selection: 00: Select PLL48M clock; 01: Select LIRC 32K clock 1x: Select external crystal clock XTAL, passive crystal (32768Hz/4MHz/8MHz) and active crystal (1MHz~48MHz)

3.2.14. Clock switch completion flag register(CLK_SEL_RDY)

Address offset: 0x48

Reset value: 0x0000 03FF

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

15:10	9	8	7	6	5	4	3	2	1	0
Res.	TIM7RDY	TIM6RDY	TIM5RDY	TIM4RDY	TIM3RDY	TIM2RDY	TIM1RDY	TIM0RDY	PWM1RDY	PWM0RDY
	R	R	R	R	R	R	R	R	R	R

31:10	-	Reserved
9	TIM7RDY	TIM7 clock switch complete flag

		0: Is switching; 1: Clock switch complete
8	TIM6RDY	TIM6 clock switch complete flag 0: Is switching; 1: Clock switch complete
7	TIM5RDY	TIM5 clock switch complete flag 0: Is switching; 1: Clock switch complete
6	TIM4RDY	TIM4 clock switch complete flag 0: Is switching; 1: Clock switch complete
5	TIM3RDY	TIM3 clock switch complete flag 0: Is switching; 1: Clock switch complete
4	TIM2RDY	TIM2 clock switch complete flag 0: Is switching; 1: Clock switch complete
3	TIM1RDY	TIM1 clock switch complete flag 0: Is switching; 1: Clock switch complete
2	TIM0RDY	TIM0 clock switch complete flag 0: Is switching; 1: Clock switch complete
1	PWM1RDY	PWM1 clock switch complete flag 0: Is switching; 1: Clock switch complete
0	PWM0RDY	PWM0 clock switch complete flag 0: Is switching; 1: Clock switch complete

3.2.15. PWM clock selection register(PWM_CLK_SEL)

Address offset: 0x4C

Reset value: 0x0000 0000

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

15:12	11:10	9:8	7:4	3:0
Res.	PWM1SEL	PWM0SEL	PWM1DIV	PWM0DIV
	RW	RW	RW	RW

31:12	-	Reserved
11:10	PWM1SEL	PWM1 clock selection: 00: Select PLL 48M clock; 01: Select LIRC 32K clock 1x: Select external crystal clock XTAL, passive crystal (32768Hz/4MHz/8MHz) and active crystal (1MHz~48MHz)
9:8	PWM0SEL	PWM0 clock selection: 00: Select PLL 48M clock; 01: Select LIRC 32K clock 1x: Select external crystal clock XTAL, passive crystal (32768Hz/4MHz/8MHz) and active crystal (1MHz~48MHz)

7:4	PWM1DIV	PWM1 Clock frequency division selection: Used for clock frequency division selection register after PWM1 clock source is selected, 0~ 15: 2^(0~ 15) frequency division
3:0	PWM0DIV	PWM0 Clock frequency division selection: Used for clock frequency division selection register after PWM0 clock source is selected, 0~ 15: 2^(0~ 15) frequency division

3.2.16. SPI high-speed mode enable register(HSPEED_EN)

Address offset: 0x50

Reset value: 0x00000000

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

15:9	8	7	6	5	4	3	2	1	0
Res.	HSSEL8	HSSEL7	HSSEL6	HSSEL5	HSSEL4	HSSEL3	HSSEL2	HSSEL1	HSSEL0
	RW	RW	RW	RW	RW	RW	RW	RW	RW

31:9	-	Reserved
8	HSSEL8	The high-speed mode is enabled for the SPI communication port PA15 1: High-speed mode; 0: Normal mode
7	HSSEL7	The high-speed mode is enabled for the SPI communication port PB4 1: High-speed mode; 0: Normal mode
6	HSSEL6	The high-speed mode is enabled for the SPI communication port PB5 1: High-speed mode; 0: Normal mode
5	HSSEL5	The high-speed mode is enabled for the SPI communication port PB10 1: High-speed mode; 0: Normal mode
4	HSSEL4	The high-speed mode is enabled for the SPI communication port PB11 1: High-speed mode; 0: Normal mode
3	HSSEL3	The high-speed mode is enabled for the SPI communication port PB12 1: High-speed mode; 0: Normal mode
2	HSSEL2	The high-speed mode is enabled for the SPI communication port PC3 1: High-speed mode; 0: Normal mode
1	HSSEL1	The high-speed mode is enabled for the SPI communication port PC4 1: High-speed mode; 0: Normal mode
0	HSSEL0	The high-speed mode is enabled for the SPI communication port PC5 1: High-speed mode; 0: Normal mode

Note: This register is enabled when the SPI communication frequency is greater than 1M.

3.2.17. ADC hardware trigger source selection register(ADC_TRIG_SEL)

Address offset: 0x54

Reset value: 0x0000 0000

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

15:3	2:0
Res.	TRIG_SEL
	RW

31:4	-	Reserved
2:0	TRIG_SEL	ADC hardware trigger source selection: 000: TIM0 event 001: TIM1 event 010: TIM2 event 011: TIM3 event 100: TIM4 event 101: TIM5 event 110: TIM6 event 111: TIM7 event

3.3. SysTick timer

3.3.1. System timer description

SysTick timer, used to generate periodic interrupt requests.

Functional characteristics:

- 24 bit loop decrement count
- Automatic loading count initial value
- Configurable interrupts
- The count clock is the HCLK clock

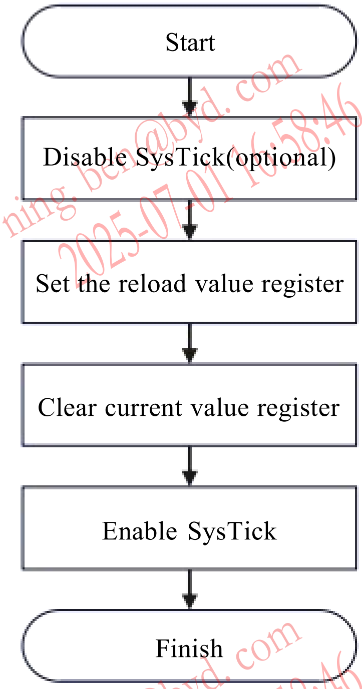
SysTick cannot wake up standby mode.

SysTick is a 24-bit timer and counts down. After the timer count is reduced to 0, a programmable value will be reloaded and a SysTick exception will be generated at the same time. SysTick can be operated by polling or interruption. Programs that use polling can read the SysTick control and status registers and check COUNTFLAG. If the flag is set, it means that the SysTick count has been reduced to 0.

Directly operate register to control SysTick:

```
SysTick->CTRL = 0;          // Disable SysTick
SysTick->LOAD = 999;       // Count down from 999 to 0
SysTick->VAL = 0;           // Clear the current value to 0
SysTick->CTRL = 0x7;       // Enable SysTick
```

The process of setting SysTick is as follows:


Start
Disable SysTick(optional)
Set the reload value register
Clear current value register
Enable SysTick
Finish

3.3.2. Registers

Address	Register	CMSIS symbol	Description
0xE000E010	SYST_CSR	SysTick->CTRL	SysTick control and status register
0xE000E014	SYST_RVR	SysTick->LOAD	SysTick reload value register
0xE000E018	SYST_CVR	SysTick->VAL	SysTick current value register

3.3.2.1. SysTick control and status register (SYST_CSR)

Bit	Bit symbol	Description	RW	Reset value
31:17	Res.	-	-	-
16	COUNTFLAG	SysTick timer overflow flag, read-only bit 1: Timer counts to 0 0: Timer did not count to 0 Reading register will be cleared	R	0
15:3	Res.	-	-	-
2	CLKSOURCE	SysTick clock source 1: Use HCLK clock 0: Reserved	R/W	0
1	TICKINT	SysTick interrupt enable 1: Enable interrupt 0: Prohibit interrupt	R/W	0
0	ENABLE	SysTick timer enable 1: Enable SysTick 0: Prohibit SysTick	R/W	0

3.3.2.2. SysTick reload value register (SYST_RVR)

Bit	Bit symbol	Description	RW	Reset value
31:24	Res.	-	-	-
23:0	RELOAD	Reload initial value of Systick timer	R/W	Undefined

3.3.2.3. SysTick current value register (SYST_CVR)

Bit	Bit symbol	Description	RW	Reset value
31:24	Res.	-	-	-
23:0	CURRENT	The current value of the Systick timer, writing any value will clear the register, and COUNTFLAG will also be cleared (does not cause an exception to the Systick timer)	R/W	Undefined

Chapter 4 FLASH

4.1. FLASH features

- Main storage block: 128K Bytes, 256 pages
- NVR1/2/3/4: 512 Bytes per NVR, 1 page
- Support 32-bit whole word programming
- Support whole chip erasing and page erasing
- After erasure, the data is all 1
- Support IAP upgrade
- Has erase and programming protection state to prevent accidental write operations
- Support read protection function and write configuration word protection function
- Support security protection status, which can prevent illegal read access to code or data
- Erasing/programming times: At least 100000 times@25℃
- Data retention period: 100 years@25℃

4.2. FLASH memory allocation

Main block is used to store the chip running program, the address range is 0x0000 to 0x1FFFF

NVR1 (system block) addresses 0x201F4, 0x201F8, 0x201FC and NVR3 addresses 0x205FC~0x205E0 are option byte areas. Users can configure according to specific application requirements.

NVR2 (information block): Used to store information about the factory configuration, which cannot be changed by the user.

NVR4 (DATA): User data storage area.

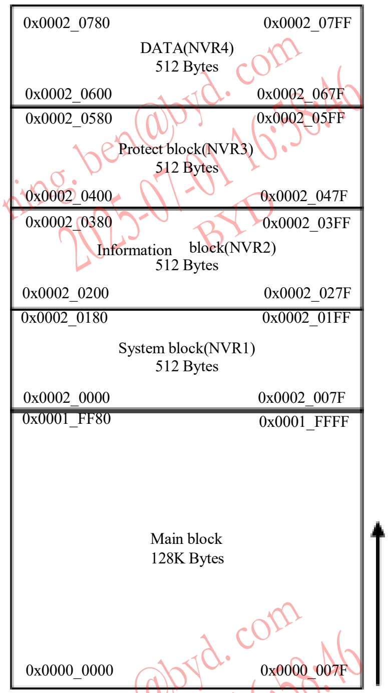


Figure 4.1 FLASH IP memory allocation diagram

Note: About page concept, mainly used for page erasing.

Module	Address	Size(Bytes)	Page
Main block	0x0000_0000 ~ 0x0000_01FF	512	Page 0
	0x0000_0200 ~ 0x0000_03FF	512	Page 1
	0x0000_0400 ~ 0x0000_05FF	512	Page 2
	0x0000_0600 ~ 0x0000_07FF	512	Page 3
	…	…	…
	0x0001_FE00~ 0x0001_FFFF	512	Page 255
NVR1(System block)	0x0002_0000 ~ 0x0002_01FF	512	1 page
NVR2 (Information block)	0x0002_0200 ~ 0x0002_03FF	512	1 page
NVR3(Protect block)	0x0002_0400 ~ 0x0002_05FF	512	1 page
NVR4(DATA)	0x0002_0600 ~ 0x0002_07FF	512	1 page

Table 4.1 Address allocation table

4.3. FLASH function overview

When the FLASH is operated and the erasing and programming instructions are executed, the AHB bus is occupied. When the erasing and programming instructions are completed, the bus is released and the program continues to execute. If an interrupt occurs, wait for the erasing and programming instructions to complete, and the FMC_STATE register is cleared to 0, then execute the interrupt service routine.

4.3.1. Key value

In order to enhance security, when performing an operation, it is necessary to write specific values to a bit to verify whether it is a safe operation. These values are called key values. The FLASH of BF7707AMXX has three key values:

RDP = 0xA5, used to release read protection;

KEY1 = 0x45670123, used to release the Flash lock;

KEY2 = 0xCDEF89AB, used to release the Flash lock.

4.3.2. FLASH unlock

After reset, the FMC module is protected and the FMC control register (FMC_CTRL) does not allow write operations. The FMC module can be opened by writing a specific sequence to the FMC_KEY register. The unlock sequence is as follows:

1. Write KEY1 = 0x45670123 to the FMC key register (FMC_KEY)
2. Write KEY2 = 0xCDEF89AB to the FMC key register (FMC_KEY)

Wrong sequence of operations will lock the FMC module and the FMC_CTRL register before the next reset, and writing the wrong key sequence will also generate a bus error. A bus error occurs when either of the following conditions occurs:

- The first write is not KEY1
- The first write is KEY1 but the second write is not KEY2

The first 2 pages of FLASH and NVR are still protected after writing the key value to the FMC key register (FMC_KEY). For erasing and programming of the first 2 pages of FLASH, PER0KEY is also required to unlock. The unlocking process consists of two write operations:

1. Write 0x45670123 to the FLASH first 2 page key register (FMC_PER0KEY)
2. Write 0xCDEF89AB to the FLASH first 2 page key register (FMC_PER0KEY)

For the unlocking steps of NVR1/3/4, see "NVR1/3/4 unlocked", and for the unlocking steps of NVR2, see "NVR2 read".

The program can set the LOCK bit in the FMC control register (FMC_CTRL) to lock the FMC module and the FMC control register (FMC_CTRL).

4.3.3. Erase

4.3.3.1. Main memory block page erase

The page erase function ofFMC initializes the page content of the main storage flash memory to high level. Each page can be erased independently without affecting the contents of other pages. FMC erase page steps are as follows:

1. Make sure that the FMC_CTRL register is not in a locked state;
2. Check the BUSY bit of the FMC_STATE register to determine whether the flash memory is in the erase and write access state, if the BUSY bit is 1, you need to wait for the operation to end, and the BUSY bit becomes 0;
3. Set the PER bit of the FMC_CTRL register;
4. Writes the absolute address of the page to be erased to the FMC_ADDR register;
5. Send the page erase command to FMC by setting the START bit of the FMC_CTRL register to 1;
6. Wait for the erase command to be executed, and the BUSY bit of the FMC_STATE register is cleared to 0;
7. If necessary, the CPU read can verify whether the page has been successfully erased;
8. After completing the relevant operations, re-lock the FMC control register to prevent FLASH from being misoperated.

When the page erase is successfully performed, the END bit of the FMC_STATE register is set. The user needs to make sure that the correct erase address is written. Otherwise, when the address of the page to be erased is used to fetch instructions or access data, the software will run away. In this case, the FMC does not provide any notification of the error. At the same time, erasing a page with erase protection will have no effect, and the WPERR bit of the FMC_STATE register will be set. The figure below shows the page erase operation process.

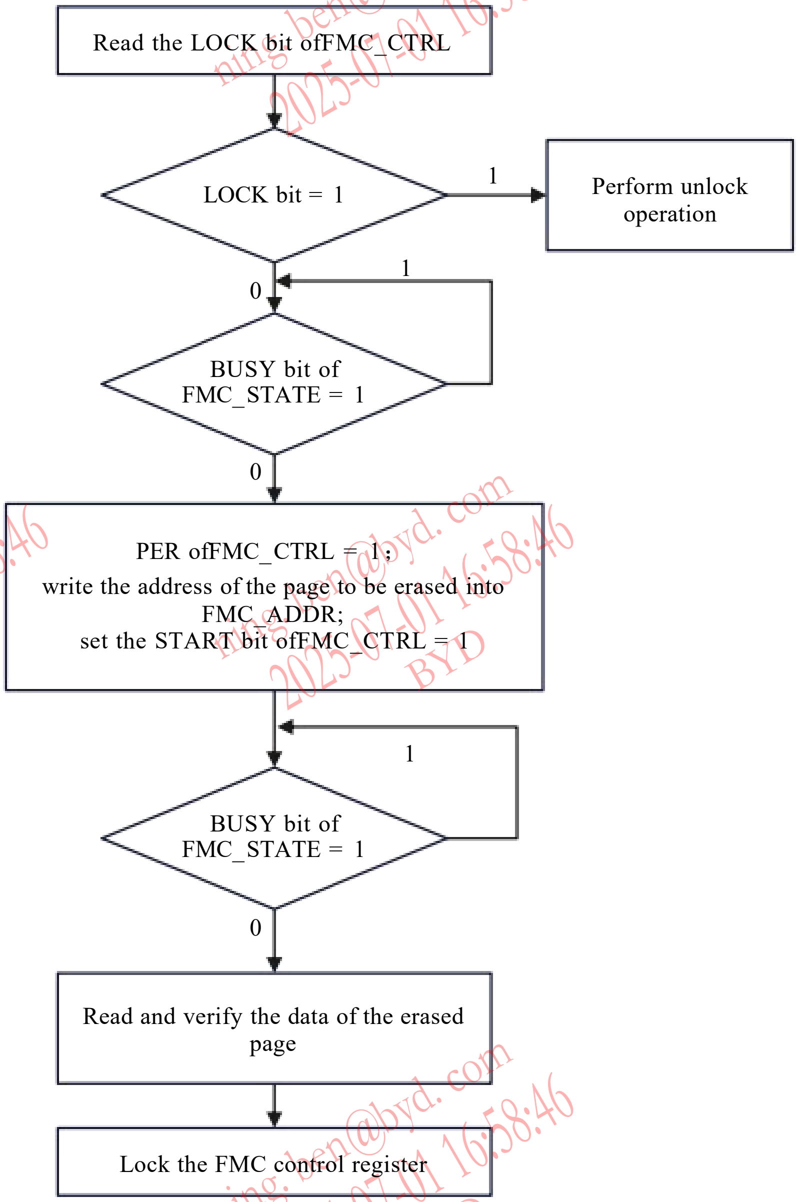

Read the LOCK bit ofFMC_CTRL
LOCK bit = 1
1
Perform unlock operation
0
1
BUSY bit of FMC_STATE = 1
0
PER ofFMC_CTRL = 1;
write the address of the page to be erased into FMC_ADDR;
set the START bit ofFMC_CTRL = 1
1
BUSY bit of FMC_STATE = 1
0
Read and verify the data of the erased page
Lock the FMC control register

Figure 4.2 Page erase operation process

4.3.3.2. Main memory block full chip erase

The chip erase function of the BF7707AMXX initializes the contents ofthe main memory block to a high level. Chip Erase does not affect the NVR. For chip erase operation, the specific steps of register setting are as follows:

1. Make sure that the FMC_CTRL register is not in a locked state;
2. Wait for the BUSY bit of the FMC_STATE register to become 0;
3. Set the MER bit in the FMC_CTRL register;
4. Send the entire chip erase command to FMC by setting the START bit of the FMC_CTRL register to 1;
5. Wait for the erase command to be executed and the BUSY bit of the FMC_STATE register is cleared to 0;
6. If necessary, it can be verified by CPU read whether the programming is successful ;
7. After completing the relevant operations, re-lock the FMC control register to prevent FLASH from being misoperated.

When the entire chip erase is successfully executed, the END bit of the FMC_STATE register is set. Since all the flash memory data will be reset to 0xFFFF_FFFF, the entire chip erase operation can be realized by directly accessing the FMC register by a program running in SRAM or using a debugging tool.

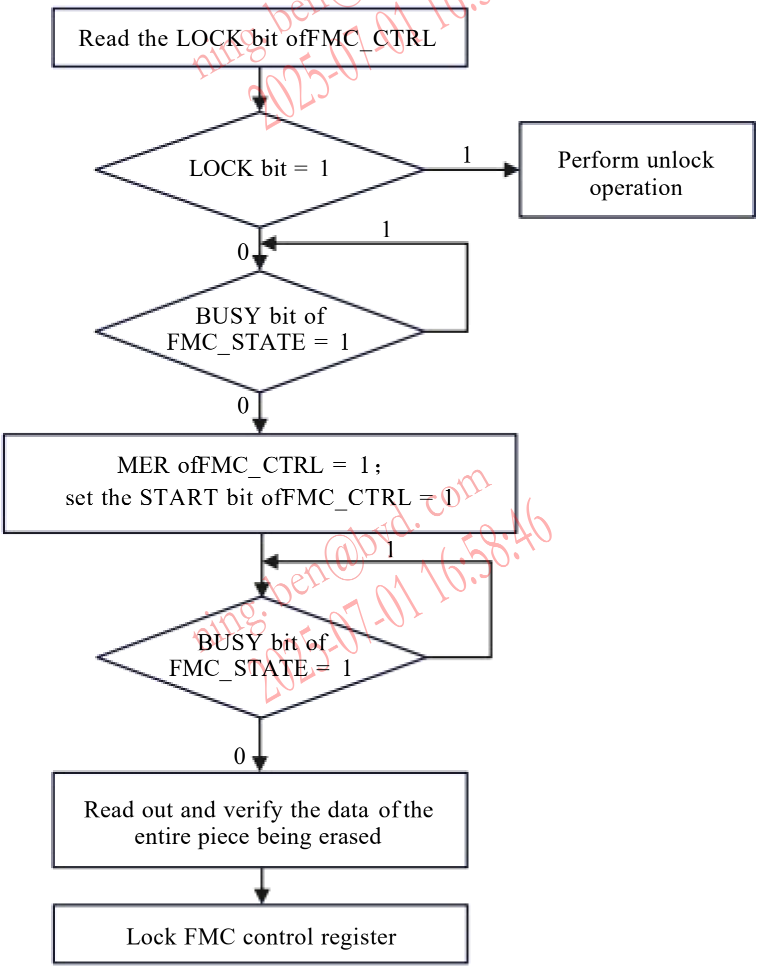

Read the LOCK bit ofFMC_CTRL
LOCK bit = 1
1
Perform unlock operation
1
0
BUSY bit of FMC_STATE = 1
0
MER ofFMC_CTRL = 1;
set the START bit ofFMC_CTRL = 1
1
BUSY bit of FMC_STATE = 1
0
Read out and verify the data of the entire piece being erased
Lock FMC control register

Figure 4.3 Chip erase operation process

4.3.4. Main memory block programming

FMC provides a 32-bit whole word programming function to modify the contents ofthe main memory block. The programming operation using each register flow is as follows:

1. Make sure that the FMC_CTRL register is not in a locked state;
2. Wait for the BUSY bit of the FMC_STATE register to become 0;
3. Set the PG bit of the FMC_CTRL register;
4. The CPU writes a 32-bit whole word to the destination absolute address (0x000X_XXXX);
5. Wait for the completion of the programming instruction, and the BUSY bit of the FMC_STATE register is cleared to 0;
6. Clear the PG bit of the FMC_CTRL register;
7. If necessary, it can be verified by CPU read whether the programming is successful (little endian mode);
8. After completing the relevant operations, re-lock the FMC control register to prevent FLASH from being misoperated.

When the main memory block programming is successfully executed, the END bit of the FMC_STATE register is set.

Programming errors: The following errors can be detected.

- **PGERR: Programming error**

  When performing a whole word programming operation, the hardware checks whether the data at the destination address is 0xFFFF. When the data at the address is not 0xFFFF, a programming error occurs, PGERR is set to 1, and programming is aborted.

- **WPERR: Erase/Program protection error**

  During an erase/program operation on a protected page, The WPERR bit of FMC_STATE is set to 1 by hardware and the erase/program is aborted. The following figure shows the flow of main memory block programming.

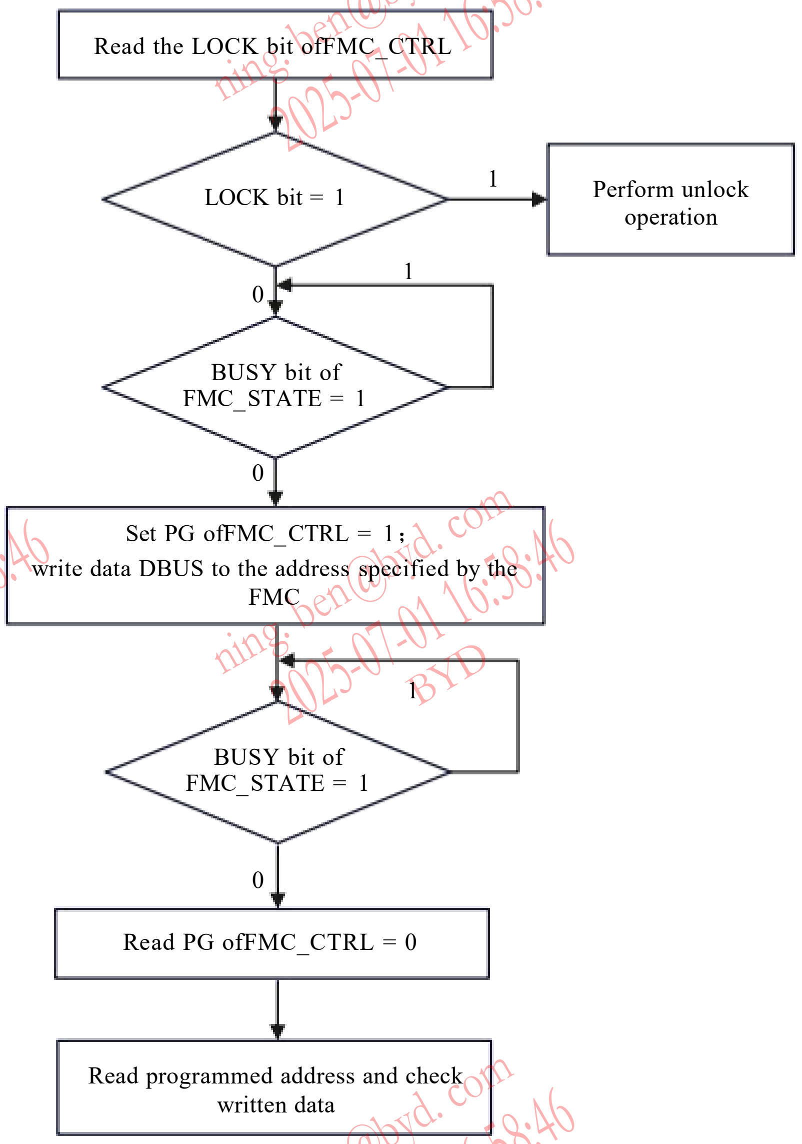

Read the LOCK bit ofFMC_CTRL
LOCK bit = 1
1
Perform unlock operation
0
1
BUSY bit of FMC_STATE = 1
0
Set PG ofFMC_CTRL = 1;
write data DBUS to the address specified by the FMC
1
BUSY bit of FMC_STATE = 1
0
Read PG ofFMC_CTRL = 0
Read programmed address and check written data

Figure 4.4 Main memory block programming operation process

4.3.5. Protect

The user code area in the flash prevents illegal reading. The pages of the flash are protected to prevent them from being accidentally changed when the program runs away. The basic unit of write protection is: 2 pages.

4.3.5.1. Main memory read protection

You can activate read protection by setting the RDP option byte and then resetting the system to load the new RDP option byte.

Read protection turned on:

1. Only allow the read operation of the main flash memory from the user code.
2. All the functions of loading and executing code to the built-in SRAM through SWD are still valid, and it can also be started from the built-in SRAM through SWD. This function can be used to release the read protection.
3. When the read-protected option byte is changed to the unprotected value of the memory, the entire chip erase process will be executed.

Steps to release read protection:

1. Erase the entire option byte area, the read protection code (RDP) will become 0xFF, at this time the read protection is still valid;
2. Write the correct nRDP (0x5A) and RDP (0xA5) to release the protection of the memory. This operation will firstly cause the entire chip erase operation of the main flash memory;
3. Perform a reset to reload the option byte (and the new RDP code), at this time the read protection is released**.**

4.3.5.2. Main memory write protection

Write protection is implemented in units of every 2 pages.

If you try to program or erase a protected page, a protection error flag will be returned in the FMC status flag register (FMC_STATE).

The configuration option byte WRP[15:0] will set the write protection, and the subsequent system reset will load the new WRPx option byte. Page 0~1 are automatically written-protected, and other parts of the memory can be programmed through the code executed in the main memory (to achieve data storage functions), but it is not allowed to perform write or erase operations in the debug mode or after the internal SRAM is started (except for whole-chip erase).

Release write protection:

1. Erase the entire option byte area;
2. Reload option bytes (including new WRP[15:0] bytes);
3. Perform a system reset and the write protection is released.

4.4. NVR1/3/4

4.4.1. NVR1/3 option bytes

NVR1 (system block) addresses 0x201F4, 0x201F8, 0x201FC (Fixed value is 0xFF880077) and NVR3 addresses 0x205FC~0x205E0 of BF7707AMXX are option byte areas. Each option byte space has 512 bytes, the option bytes are configured by the user according to specific application requirements. Each 32-bit word in the option byte is divided into the following format:

Bit[31:16]	Bit[15:0]
Inversion of the option byte 0	Option byte 0

The option byte block must be 32-bit word programming, Bit[31:16] is the inverse of Bit[15:0], otherwise the default value is restored to 0xFFFF_FFFF.

NVR1 option byte description:

FLASH address	Bit	Bit symbol	Description	Defaults
0x201F4	[3:1]	BOR_VTH_SEL	BOR power-down threshold configuration register 000/001/010: 4.2V 011: 3.7V 100: 3.3V 101: 2.8V 110/111: Reserved, prohibition of use After power-on reset, the BOR threshold defaults to 2.8V, and the program configures the above gears. See table "BOR Threshold and delay selection"	101
	[0]	BOR_EN	BOR control register 1: Enable, 0: Disable. Enable by default	1

FLASH address	Bit	Bit symbol	Description	Defaults
0x201F8	[10]	PD_WDT_EN	0: Do not allow the program to turn off the watchdog function 1: Allow the program to turn off the watchdog function	1
	[9]	WDTRST_SD	0: When entering idle mode 1, a watchdog reset is generated 1: Reserved, cannot be configured	0
	[8]	WDTRST_SP	0: When entering idle mode 0, a watchdog reset is generated	0

			1: Reserved, cannot be configured	
	[7:0]	MAIN_READ_ PROTECT	Code read protection option byte. Unprotected state (can be read): RDP=0xA5 Protected state (cannot be read): RDP!=0xA5	0xA5

NVR3 option byte description: The basic unit of write protection is: 2 pages.
Each representative: 0: Write protection is valid; 1: Write protection is invalid

FLASH address	Bit	Bit symbol	Description	Defaults
0x205FC	[15:0]	WRP0	Write protection for page 0~31 of the main memory block Bit[0]: Write protection for page 0 and 1 Bit[1]: Write protection for page 2 and 3 … Bit[14]: Write protection for page 28 and 29 Bit[15]: Write protection for page 30 and 31	0xFFFF
0x205F8	[15:0]	WRP1	Write protection for page 32~63 of the main memory block Bit[0]: Write protection for page 32 and 33 … Bit[15]: Write protection for page 62 and 63	0xFFFF
0x205F4	[15:0]	WRP2	Write protection for page 64~95 of the main memory block Bit[0]: Write protection for page 64 and 65 … Bit[15]: Write protection for page 94 and 95	0xFFFF
0x205F0	[15:0]	WRP3	Write protection for page 96~ 127 of the main memory block Bit[0]: Write protection for page 96 and 97 … Bit[15]: Write protection for page 126 and 127	0xFFFF
0x205EC	[15:0]	WRP4	Write protection for page 128~ 159 of the main memory block Bit[0]: Write protection for page 128 and 129 … Bit[15]: Write protection for page 158 and 159	0xFFFF
0x205E8	[15:0]	WRP5	Write protection for page 160~ 191 of the main memory block Bit[0]: Write protection for page 160 and 161 … Bit[15]: Write protection for page 190 and 191	0xFFFF
0x205E4	[15:0]	WRP6	Write protection for page 192~223 of the main memory block Bit[0]: Write protection for page 192 and 193	0xFFFF

			… Bit[15]: Write protection for page 222 and 223	
0x205E0	[15:0]	WRP7	Write protection for page 224~255 of the main memory block Bit[0]: Write protection for page 224 and 225 … Bit[15]: Write protection for page 254 and 255	0xFFFF

4.4.2. NVR4 (DATA area)

The NVR4 of BF7707AMXX includes a DATA storage area for storing user data. The address range of the DATA area is 0x0002_0600 ~ 0x0002_07FF, one page, 512 bytes.

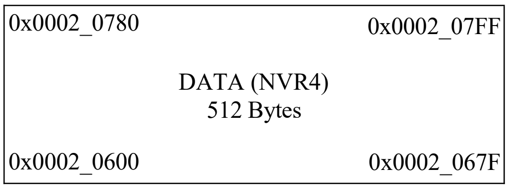


Figure 4.5 DATA area address map

4.4.3. NVR1/3/4 unlocked

By default, NVR1/3/4 are always readable and write-protected.

Write operation (program/erase) to NVR1/3/4 must first write the correct key sequence in the FMC option key register, then allow write operation to NVR1/3/4, OPTWRE bit ofFMC control register (FMC_CTRL) indicates that writing is allowed, clearing this bit will disable writing.

The unlock sequence is as follows:

1. Write KEY1 = 0x45670123 to the FMC key register (FMC_KEY);
2. Write KEY2 = 0xCDEF89AB to the FMC key register (FMC_KEY);
3. Write 0x45670123 to the FMC option key register (FMC_OPTKEY);
4. Write 0xCDEF89AB to the FMC Option Key Register (FMC_OPTKEY).

4.4.4. NVR1/3/4 erase

After the erase operation is completed, the page content of the NVR is initialized to a high level.

The NVR1/3/4 erasing steps are as follows.

1. Unlock according to NVR1/3/4 unlocking sequence;
2. Make sure the FMC_CTRL register is not locked;
3. Wait for the BUSY bit of the FMC_STATE register to become 0;
4. Wait for the OPTWRE bit of the FMC_CTRL register to be 1;
5. Set the OPTER bit ofthe FMC_CTRL register;
6. Write the absolute address of the NVR to be erased to the FMC_ADDR register;

7. Send erase command to FMC by setting the START bit of the FMC_CTRL register to 1;
8. Wait for the erase command to be executed and the BUSY bit of the FMC_STATE register is cleared to 0;
9. If necessary, it can be verified by CPU read whether the programming is successful;
10. After completing the relevant operations, re-lock the FMC control register to prevent FLASH from being misoperated.
11. When the NVR1/3/4 erase is successfully executed, the END bit of the FMC_STATE register is set.

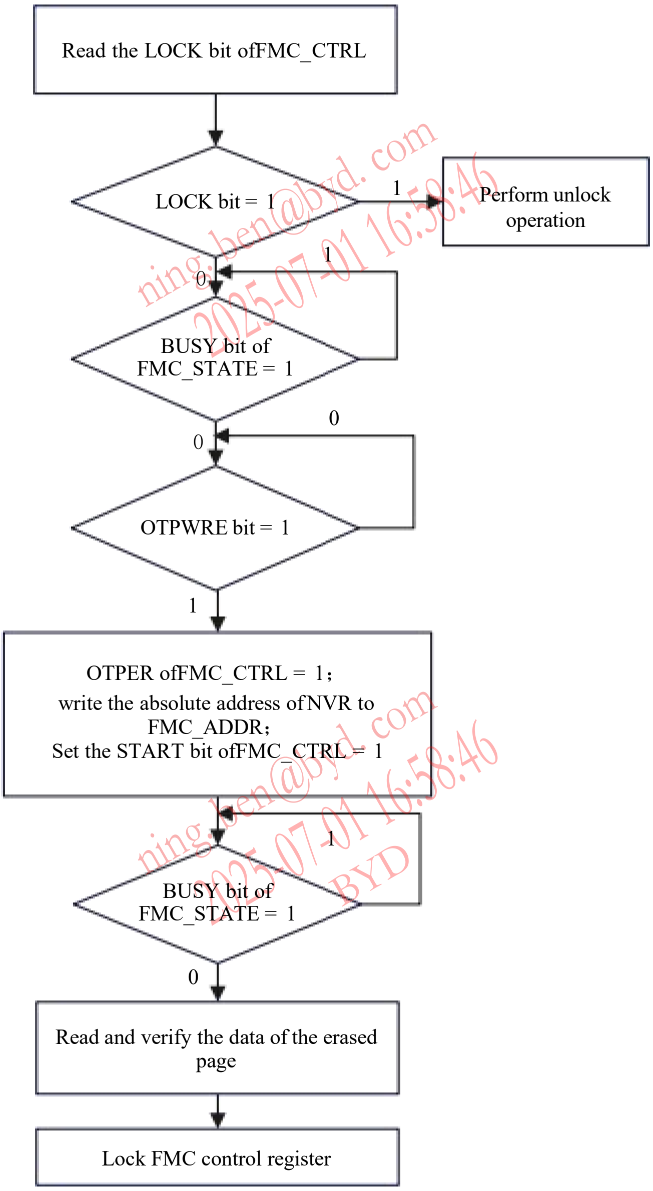

Read the LOCK bit ofFMC_CTRL
LOCK bit = 1
1
Perform unlock operation
0
1
BUSY bit of FMC_STATE = 1
0
0
OTPWRE bit = 1
1
OTPER ofFMC_CTRL = 1；
write the absolute address of NVR to FMC_ADDR；
Set the START bit ofFMC_CTRL = 1
1
BUSY bit of FMC_STATE = 1
0
Read and verify the data of the erased page
Lock FMC control register

Figure 4.6 NVR erase erasing steps

4.4.5. NVR1/3/4 programming

FMC provides a 32-bit whole word programming function, which can be used to modify the content of NVR1/3/4.

Modifying any option byte: First erase the option byte page, then program all option bytes from the option byte page. When programming the option byte, Bit[31:16] must be the inverse of Bit[15:0].

Note: When modifying the address 0x201F8 or 0x201F4, first read the option byte page, then execute eraseing option byte page, then write 0xFF880077 to address 0x201FC, write the read value or the value to be configured to address 0x201F4, and finally write the address 0x201F8.

The NVR1/3/4 programming operation steps are as follows.

1. Unlock according to NVR1/3/4 unlocking sequence;
2. Make sure the FMC_CTRL register is not locked;
3. Wait for the BUSY bit of the FMC_STATE register to become 0;
4. Wait for the OPTWRE bit of the FMC_CTRL register to be 1;
5. Set the OPTPG bit of the FMC_CTRL register;
6. The CPU writes a 32-bit whole word to the destination address;
7. Wait for the completion of the programming instruction, and the BUSY bit of the FMC_STATE register is cleared to 0;
8. Clear the OPTPG bit of the FMC_CTRL register;
9. If necessary, it can be verified by CPU read whether the programming is successful (little endian mode);
10. After completing the relevant operations, re-lock the FMC control register to prevent FLASH from being misoperated.

When the NVR1/3/4 programming is successfully executed, the END bit of the FMC_STATE register is set.

Programming error

When performing a whole word programming operation, the hardware checks whether the data of the destination address is 0xFFFF. When the data ofthe address is not 0xFFFF, a programming error occurs, PGERR is set to 1, and programming is aborted.

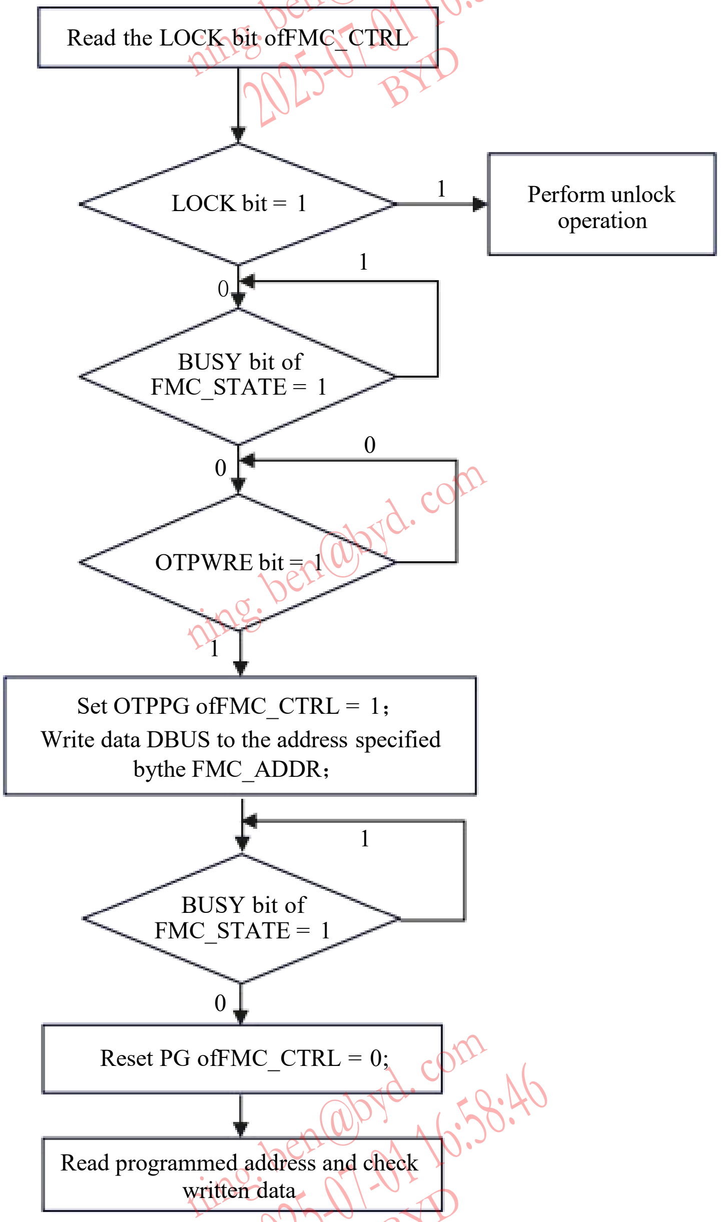

Read the LOCK bit ofFMC_CTRL
LOCK bit = 1
1
Perform unlock operation
0
1
BUSY bit of FMC_STATE = 1
0
0
OTPWRE bit = 1
1
Set OTPPG ofFMC_CTRL = 1;
Write data DBUS to the address specified bythe FMC_ADDR;
1
BUSY bit of FMC_STATE = 1
0
Reset PG ofFMC_CTRL = 0;
Read programmed address and check written data

Figure 4.7 NVR programming operation process

4.4.6. NVR1/3/4 protection

By default, the option byte block is always readable and write protected. To write to the option byte block (programming/erasing), first write the correct sequence of keys to the OPTKEY, then allow write operation to the option byte block. OPTWRE bit ofFMC control register (FMC_CTRL) indicates that writing is allowed, clearing this bit will disable writing.

4.5. NVR2 read

The NVR2 (information block) module stores the configuration word, ADC calibration value and electronic signature of the device, etc. It needs to be unlocked to read, and the user cannot change it. The Information block key register (FMC_INFKEY) is used to unlock the read protection.

The reading steps are as follows:

1. Write KEY1 = 0x45670123 to the FMC key register (FMC_KEY);
2. Write KEY2 = 0xCDEF89AB to the FMC key register (FMC_KEY);
3. Write 0x896DBA23 to the Information block key register (FMC_INFKEY);
4. Write 0x7EA56C8F to the Information block key register (FMC_INFKEY);
5. Read the data;
6. Configure the FMC control register FMC_CTRL.INFRDE = 0 to lock the NVR2 module.

4.6. Registers

Base address: 0x5001 0000

Address offset	Register	Description
0x00	FMC_KEY	FMC key register
0x04	FMC_OPTKEY	FMC option key register
0x08	FMC_STATE	FMC status flag register
0x0C	FMC_CTRL	FMC control register
0x10	FMC_ADDR	FMC address register
0x14	FMC_PER0KEY	The first 2 pages of FLASH key register
0x18	FMC_INFKEY	Information block key register
0x1C	FMC_CFG	CFG register
0x20	FMC_WRP0	Write protection register 0
0x24	FMC_WRP1	Write protection register 1
0x28	FMC_WRP2	Write protection register 2
0x2C	FMC_WRP3	Write protection register 3

4.6.1. FMC key register (FMC_KEY)

Address offset: 0x00

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">KEY[31:16]</td></tr>
<tr><td colspan="16">W</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">KEY[15:0]</td></tr>
<tr><td colspan="16">W</td></tr>
</table>

31:0	KEY[31:0]	Used to unlock FLASH erasing and programming The register is write-only, the read back data is 0 Unlock method: Write KEY1 = 0x45670123, KEY2 = 0xCDEF89AB in sequence Note: Only after configuring this register successfully, other FMC registers can be configured

4.6.2. FMC option key register (FMC_OPTKEY)

Address offset: 0x04

Reset value: 0x0000 0000

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															

31:0	OPTKEY[31:0]	Used to unlock NVR1/3/4 erasing and programming The register is write-only, the read back data is 0 Unlock method: Write 0x45670123, 0xCDEF89AB in sequence

4.6.3. FMC status flag register (FMC_STATE)

Address offset: 0x08

Reset value: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.										END	WPERR	Res.	PGERR	Res.	BUSY
										RC_W1	RC_W1		RC_W1		R

31:6	-	Reserved
5	END	Operation end flag After the operation is successfully executed, this bit is set to 1 by hardware Software write 1 to clear 0
4	WPERR	Erase/program protection error flag During an erase/program operation on a protected page, this bit is set to 1 by hardware Software write 1 to clear 0
3	-	Reserved
2	PGERR	Programming error flag When the state of the programmed area is not 0xFFFF, program the flash , this bit is set to 1 by hardware Software write 1 to clear 0
1	-	Reserved
0	BUSY	Busy, this bit indicates that the flash operation is in progress At the beginning of the flash operation, this bit is set to 1 At the end of the operation or an error occurs, this bit is cleared to 0

4.6.4. FMC control register (FMC_CTRL)

Address offset: 0x0C

Reset value: 0x0000 0080

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

15:12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	INFRDE	Res.	OPTWRE	PER0WRE	LOCK	START	OPTER	OPTPG	Res.	MER	PER	PG
	RC_W0		RC_W0	RC_W0	RS	RS	RW	RW		RW	RW	RW

31:12	-	Reserved
11	INFRDE	Allow reading information block NVR2: When this bit is ' 1', the read operation of the information block is allowed When the correct key sequence is written in the FMC_INFKEY register, this bit is set to ' 1' Software can write 0 to clear this bit
10	-	Reserved
9	OPTWRE	Allow writing NVR1/NVR3/NVR4 When this bit is ' 1', the NVR1/NVR3/NVR4 can be programmed When the correct key sequence is written in the FLASH_OPTKEY register, this bit is set to ' 1' Software can write 0 to clear this bit
8	PER0WRE	Allow writing the first 2 pages of flash When this bit is ' 1', the first 2 pages of flash can be programmed When the correct key sequence is written in the FLASH_PER0KEY register, this bit is set to ' 1' Software can write 0 to clear this bit
7	LOCK	Lock: Only ' 1' can be written. When this bit is ' 1', it means the FLASH_CTRL register is locked This bit is cleared by hardware after a correct unlock sequence is detected If the unlock operation fails, this bit remains set to 1 until the next system reset
6	START	Send erase command bit to FLASH Software set to 1 can send erase command to FLASH When the BUSY bit is cleared to 0, this bit is cleared to 0 by hardware When there is no valid erase command, after START writing 1, it can only be cleared when writing FLASH command is completed or an error occurred
5	OPTER	Erase NVR1/NVR3/NVR4 1: Effective Set and cleared by software
4	OPTPG	Programming NVR1/NVR3/NVR4

		1: Effective Set and cleared by software
3	-	Reserved
2	MER	Select erase all pages of main memory block 1: Effective Set and cleared by software
1	PER	Select erase main memory block pages 1: Effective Set and cleared by software
0	PG	Main memory block programming operation 1: Effective Set and cleared by software

4.6.5. FMC address register (FMC_ADDR)

Address offset: 0x10

Reset value: 0x0000 0000

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

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

31:0	ADDR[31:0]	Flash erase address, the bit is set by software The ADDR bit is the address of the flash erase command

4.6.6. The first 2 pages of FLASH key register (FMC_PER0KEY)

Address offset: 0x14

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PER0KEY [31:16]															
W															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PER0KEY [15:0]															
W															

31:0	PER0KEY	Used to unlock erasing and programming the first 2 pages of FLASH The register is write-only, the read back data is 0

		Unlock method: Write 0x45670123, 0xCDEF89AB in sequence

4.6.7. Information block key register (FMC_INFKEY)

Address offset: 0x18

Reset value: 0x0000 0000

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

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

31:0	INFKEY[31:0]	Used to unlock the read protection of the information block (NVR2) 0x0002_0200~0x0002_03FF The register is write-only, the read back data is 0 Unlock method: Write 0x896DBA23, 0x7EA56C8F in sequence

4.6.8. FMC CFG register (FMC_CFG)

Address offset: 0x1C

Reset value: 0xFFFF FFFF

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

15:5	4	3	2	1	0
Res.	WDT_EN	WDTRST_SD	WDTRST_SP	RDPRT	CFGERR
	R	R	R	R	R

31:5	-	Reserved
4	WDT_EN	User option bytes loaded by NVR1: 0: Do not allow the program to turn off the watchdog function 1: Allow program to turn off watchdog function
3	WDTRST_SD	User option bytes loaded by NVR1: 0: When entering idle mode 1, a watchdog reset is generated 1: Reserved
2	WDTRST_SP	User option bytes loaded by NVR1: 0: When entering idle mode 0, a watchdog reset is generated 1: Reserved
1	RDPRT	Read protection, read-only bit 1: Indicates that the flash memory is read protected

		0: Indicates that the flash memory is not read-protected
0	CFGERR	Reset read configuration word error, read-only bit When this bit is 1, it means that the configuration word and its inverse code do not match

4.6.9. Write protection register 0 (FMC_WRP0)

Address offset: 0x20
Reset value: 0xFFFF FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
WRP1															
R															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WRP0															
R															

31:16	WRP1	Write protection for page 0~63 of the main memory block Bit[0]: Write protection for page 0 and 1 Bit[1]: Write protection for page 2 and 3 … Bit[31]: Write protection for page 62 and 63 0: The corresponding two-pages write protection is valid 1: The corresponding two-pages write protection is invalid This register contains the write protection option byte loaded by NVR3
15:0	WRP0	

4.6.10. Write protection register 1 (FMC_WRP1)

Address offset: 0x24
Reset value: 0xFFFF FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
WRP3															
R															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WRP2															
R															

31:16	WRP3	Write protection for page 64~ 127 of the main memory block Bit[0]: Write protection for page 64 and 65 Bit[1]: Write protection for page 66 and 67

15:0	WRP2	… Bit[31]: Write protection for page 126 and 127 0: The corresponding two-pages write protection is valid 1: The corresponding two-pages write protection is invalid This register contains the write protection option byte loaded by NVR3

4.6.11. Write protection register 2 (FMC_WRP2)

Address offset: 0x28

Reset value: 0xFFFF FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
WRP5															
R															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WRP4															
R															

31:16	WRP5	Write protection for page 128~191 of the main memory block Bit[0]: Write protection for page 128 and 129 Bit[1]: Write protection for page 130 and 131 … Bit[31]: Write protection for page 190 and 191 0: The corresponding two-pages write protection is valid 1: The corresponding two-pages write protection is invalid This register contains the write protection option byte loaded by NVR3
15:0	WRP4	

4.6.12. Write protection register 3 (FMC_WRP3)

Address offset: 0x2C

Reset value: 0xFFFF FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
WRP7															
R															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WRP6															
R															

31:16	WRP7	Write protection for page 192~255 of the main memory block Bit[0]: Write protection for page 192 and 193 Bit[1]: Write protection for page 194 and 195 … Bit[31]: Write protection for page 254 and 255 0: The corresponding two-pages write protection is valid 1: The corresponding two-pages write protection is invalid This register contains the write protection option byte loaded by NVR3
15:0	WRP6	

Chapter 5 Reset

There are 7 reset sources in BF7707AMXX: Power-on reset, power-down reset, watchdog reset, CPU software reset, CPU lock reset, PC pointer overflow reset, and FLASH programming reset. As long as any one of these resets occurs, the system's global reset signal will reset the entire chip. The reset flag register (RST_STATE) can be used to determine what kind of reset the chip has performed. The reset flag bit needs to be cleared by software.

Peripherals mounted on the AHB can be reset separately by configuring the RCU_RSTEN register. For example, setting the SPI0RST bit ofthe RCU_RSTEN register to 1 generates a reset of SPI0, needs to write 0 to exit the reset state.

Note: The software reset generated by RCU_RSTEN only affects the corresponding peripheral module.

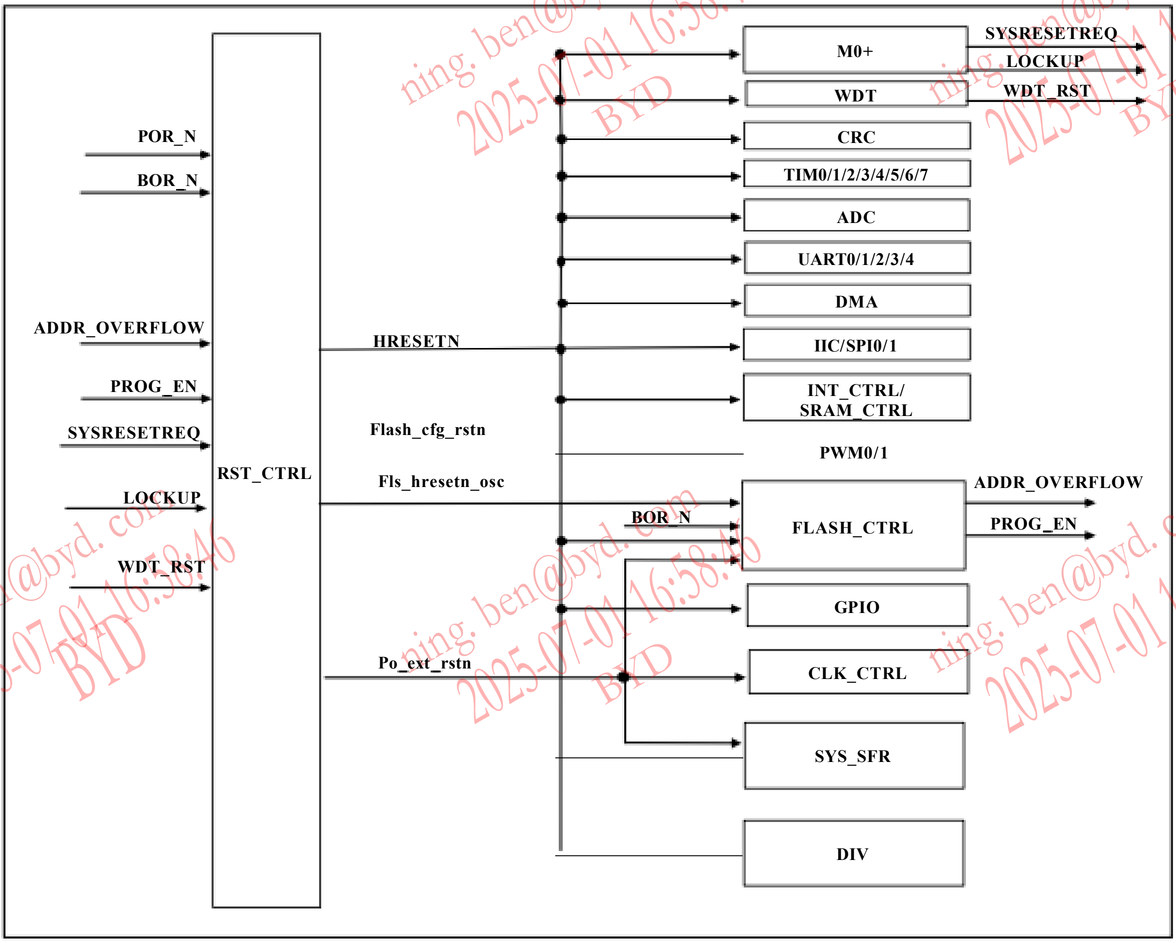


Figure 5.1 Reset block diagram

5.1. Reset introduction

5.1.1. Power-on/Power-down reset

POR_N: After the system is powered on, the POR reset module generates a POR_N reset signal, the global reset signal is generated and lasts for 93ms, and after the global reset signal continues to be valid for 7ms, the system exits the reset mode.

BOR_N: After the system is powered off, the BOR reset module generates a BOR_N reset signal, the global reset signal is valid for 7ms.

The configuration of the module enable and power down threshold of the BOR module is controlled by the address 0x201F4 in the option byte space NVR1; 0x201F4 [3:1] BOR_VTH_SEL bit configuration BOR power-off threshold, 0x201F4 [0] BOR_ EN bit configuration BOR module enabled (default enabled). For specific details, please refer to "4.4.1. NVR1/3 Option Bytes".

Power-up/power-down sequence:

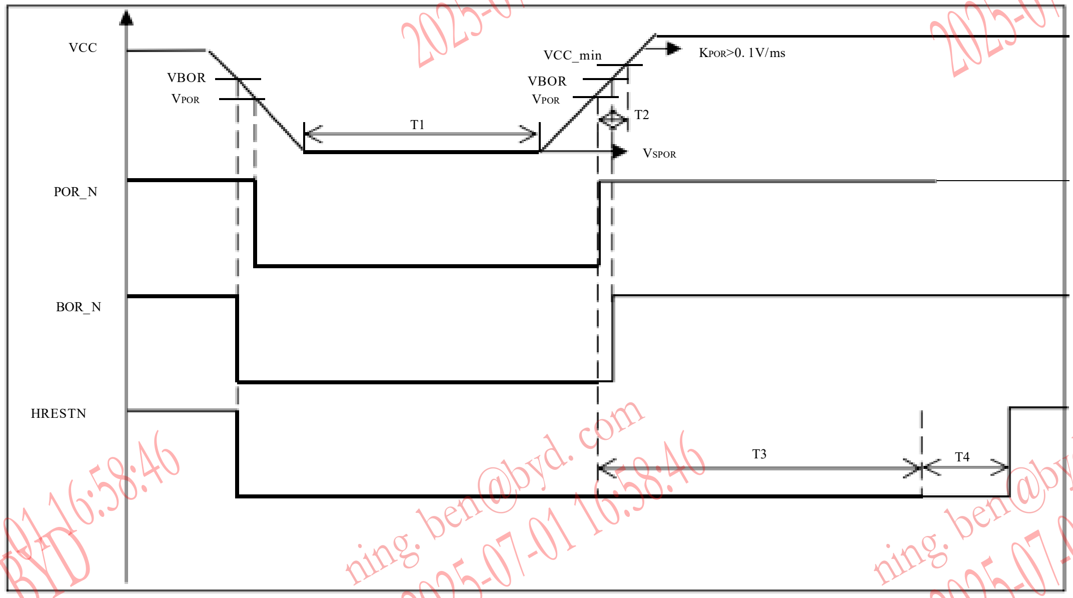


Figure 5.2 Power-on reset schematic

Power-on/power-down reset parameters:

Symbol	Parameter	Test Conditions		Min	Typical	Max	Unit
		VCC	Temperature				
VSPOR	Power-on reset start voltage	-	25°C	-	-	300	mV
KPOR	Power-on reset voltage rate	-	25°C	0.1	-	-	V/ms
VPOR	Power-on reset voltage, 0.3V hysteresis	-	25°C	1.1	1.4	2.0	V
VBOR	Power-down reset voltage (±10%), 0.2V hysteresis	-	25°C	-	VBOR	-	V
VCC_min	Minimum working voltage	-	25°C	2.6	-	-	V
T1	VCC hold VSPOR time	-	25°C	0.1	-	-	ms
T2	VPOR to VCC_min time	-	25°C	-	-	0.6*T3	ms

T3	Analog POR module delay time	-	25℃	46	93	140	ms
T4	Global reset valid time	-	25℃	-	7	-	ms

Table 5.1 Power-on reset characteristic parameter table

Note: The time required for VDD power on from 0V to 2.6V is 500us<$T_{\text{power\_on time}}$<46ms (@ full temperature conditions), which may cause the MCU to fail to operate if it is not within the specified time range.

5.1.2. Watchdog timer overflow reset

WDT_RST: After the watchdog timer overflows, the wdt_rst signal changes to high, and sys_clk detects the high level ofwdt_rst (valid for 1 clock cycle) along the clock rising edge, and then makes a global reset for 7ms. After 7ms, the system exits the reset mode.

5.1.3. CPU software reset

SYSRESETREQ: The soft reset signal is valid by writing the CPU register, and the global reset signal is valid for 7ms. After 7ms, the system exits the reset mode (AIRCR. SYSRESETREQ, writing 1 will activate the external SYSRESETREQ signal).

5.1.4. CPU lockup reset

LOCKUP: If there is an error in program execution, the CPU is locked to make the reset signal valid, and the global reset signal is valid for 7ms. After 7ms, the system exits the reset mode.

5.1.5. PC pointer overflow reset

ADDR_OVERFLOW: If the PC pointer exceeds the valid address range of the flash when the MCU addresses the program memory, the addr_overflow signal becomes high, and the rising edge of the sys_clk clock detects the high level of addr_overflow (it takes 1 clock cycle) , and then makes a global reset for 7ms, and the reset signal will clear the addr_overflow signal to 0, after 7ms, the system exits the reset mode.

5.1.6. FLASH programming reset

PROG_EN: When PROG_EN is high, it is the programming mode of FLASH. At this time, the global reset signal is valid. After it becomes low, the global reset signal continues to be valid for 7ms.After 7ms, the system exits the reset mode.

5.2. Reset timing diagram

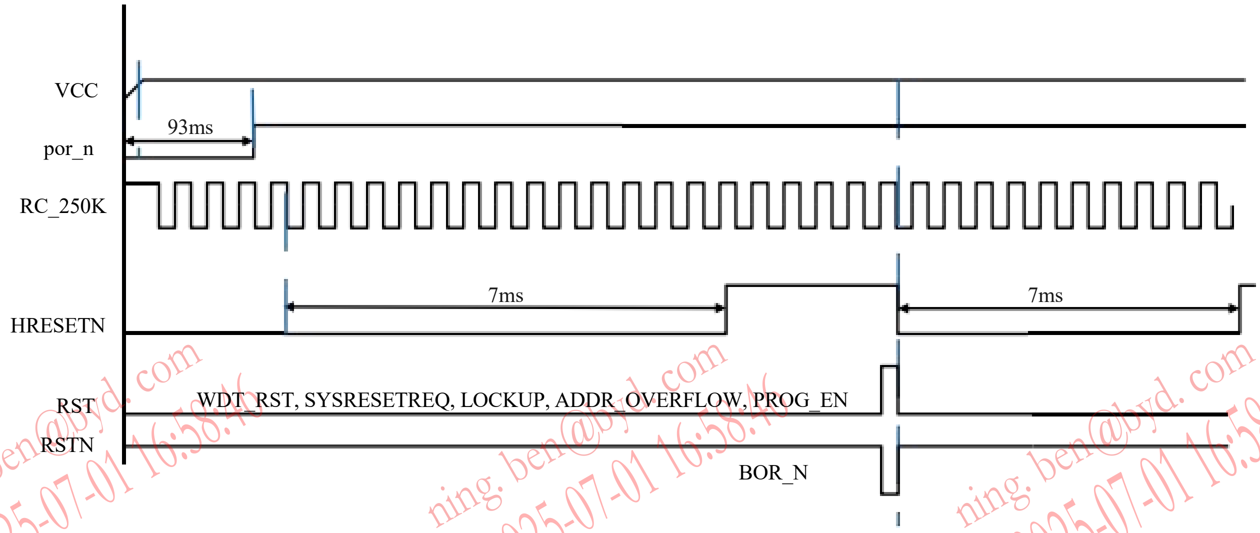


Figure 5.3 Reset sequence schematic

Reset sequence description:

1. When the chip has a power-on reset, the analog POR module delays for 93ms, and PO_N is pulled high.
2. The programmer sends instructions to make the chip enter the programming mode (PROG_EN is pulled high), after completing the programming, exit the programming mode. After a delay of 7ms, and HRESETN is pulled high. The chip enters normal operation.
3. In normal operation, when any one of watchdog reset, address overflow reset, soft reset, and LOCKUP occurs, HRESETN is pulled low, after a delay of 7ms, HRESETN is pulled high. The chip enters normal operation.
4. In normal operation, you can no longer enter the programming mode.
5. In normal operation, after BOR_N occurs, HRESETN is pulled low, after BOR_N is pulled high, and after a delay of 7ms, HRESETN is pulled high. The chip enters normal operation.

5.3. Registers

Base address: 0x5000 0000

Address offset	Register	Description
0x08	RST_STATE	Reset flag register
0x14	BOR_CFG	BOR configuration register
0x40	RCU_RSTEN	Peripheral module reset enable register

5.3.1. Reset flag register (RST_STATE)

Address offset: 0x08

Reset value: 0x0000 0001

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

15:8	7	6	5	4	3	2	1	0
Res.	PROG	ADDROF	Res.	LOCKUP	SYSRST	WDTRST	BOR	POR
	RC_W0	RC_W0		RC_W0	RC_W0	RC_W0	RC_W0	RC_W0

31:8	-	Reserved
7	PROG	FLASH programming reset flag, software can write 0 to clear the corresponding flag bit 1: FLASH programming reset occurs 0: No FLASH programming reset occurs
6	ADDROF	PC pointer overflow reset flag, software can write 0 to clear the corresponding flag bit 1: PC pointer overflow reset occurs 0: No PC pointer overflow reset occurs
5	-	Reserved
4	LOCKUP	CPU lock reset flag, software can write 0 to clear the corresponding flag bit 1: CPU lock reset occurs 0: No CPU lock reset occurs
3	SYSRST	CPU software reset flag, software can write 0 to clear the corresponding flag bit 1: CPU software reset occurs 0: No CPU software reset occurs
2	WDTRST	Watchdog reset flag, software can write 0 to clear the corresponding flag bit 1: Watchdog reset occurs 0: No watchdog reset occurs
1	BOR	Power-down reset flag, software can write 0 to clear the corresponding flag bit 1: Power-down reset occurs 0: No power-down reset occurs

0	POR	Power-on reset flag, software can write 0 to clear the corresponding flag bit 1: Power-on reset occurs 0: No power-on reset occurs

5.3.2. BOR configuration register (BOR_CFG)

Address offset: 0x14

Reset value: 0x0000 0001

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

15:1	0
Res.	DELAY_SEL
	RW

31:1	-	Reserved
0	DELAY_SEL	BOR's power-Down delay configuration register 0: Delay option 0 1: Delay option 1 See table "BOR threshold and delay selection"

5.3.3. Peripheral module reset enable register(RCU_RSTEN)

Address offset: 0x40

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.				DIV RST	GPIOD RST	GPIOC RST	GPIO BRST	GPIOA RST	DMA RST	CRC RST	ADC RST	WDT RST	TIM7 RST	TIM6 RST	TIM5 RST
				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
TIM4 RST	TIM3 RST	TIM2 RST	TIM1 RST	TIM0 RST	PWM1 RST	PWM0 RST	IIC RST	UART4 RST	UART3 RST	UART2 RST	UART1 RST	UART0 RST	SPI1 RST	SPI0 RST	Res.
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	

31:28	-	Reserved
27	DIVRST	DIV module reset enable 1: Module reset, 0: No reset. Write 0 to clear the reset state.The default value is 0. After this bit is written 1, the module is always in the reset state. You need to write 0 to exit the reset
26	GPIODRST	GPIOD module reset enable 1: Module reset, 0: No reset. Write 0 to clear the reset state.The default value is 0.

		After this bit is written 1, the module is always in the reset state. You need to write 0 to exit the reset
25	GPIOCRST	GPIOC module reset enable 1: Module reset, 0: No reset. Write 0 to clear the reset state.The default value is 0. After this bit is written 1, the module is always in the reset state. You need to write 0 to exit the reset
24	GPIOBRST	GPIOB module reset enable 1: Module reset, 0: No reset. Write 0 to clear the reset state.The default value is 0. After this bit is written 1, the module is always in the reset state. You need to write 0 to exit the reset
23	GPIOARST	GPIOA module reset enable 1: Module reset, 0: No reset. Write 0 to clear the reset state.The default value is 0. After this bit is written 1, the module is always in the reset state. You need to write 0 to exit the reset
22	DMARST	DMA(Include DMA_SFR/DMAMUX) module reset enable 1: Module reset, 0: No reset. Write 0 to clear the reset state.The default value is 0. After this bit is written 1, the module is always in the reset state. You need to write 0 to exit the reset
21	CRCRST	CRC module reset enable 1: Module reset, 0: No reset. Write 0 to clear the reset state.The default value is 0. After this bit is written 1, the module is always in the reset state. You need to write 0 to exit the reset
20	ADCRST	ADC module reset enable 1: Reset, 0: No reset. Write 0 to clear the reset state.The default value is 0. After this bit is written 1, the module is always in the reset state. You need to write 0 to exit the reset
19	WDTRST	WDT module reset enable 1: Module reset, 0: No reset. Write 0 to clear the reset state.The default value is 0. After this bit is written 1, the module is always in the reset state. You need to write 0 to exit the reset
18	TIM7RST	TIM7 module reset enable 1: Module reset, 0: No reset. Write 0 to clear the reset state.The default value is 0. After this bit is written 1, the module is always in the reset state. You need to write 0 to exit the reset

17	TIM6RST	TIM6 module reset enable 1: Module reset, 0: No reset. Write 0 to clear the reset state.The default value is 0. After this bit is written 1, the module is always in the reset state. You need to write 0 to exit the reset
16	TIM5RST	TIM5 module reset enable 1: Module reset, 0: No reset. Write 0 to clear the reset state.The default value is 0. After this bit is written 1, the module is always in the reset state. You need to write 0 to exit the reset
15	TIM4RST	TIM4 module reset enable 1: Module reset, 0: No reset. Write 0 to clear the reset state.The default value is 0. After this bit is written 1, the module is always in the reset state. You need to write 0 to exit the reset
14	TIM3RST	TIM3 module reset enable 1: Module reset, 0: No reset. Write 0 to clear the reset state.The default value is 0. After this bit is written 1, the module is always in the reset state. You need to write 0 to exit the reset
13	TIM2RST	TIM2 module reset enable 1: Module reset, 0: No reset. Write 0 to clear the reset state.The default value is 0. After this bit is written 1, the module is always in the reset state. You need to write 0 to exit the reset
12	TIM1RST	TIM1 module reset enable 1: Module reset, 0: No reset. Write 0 to clear the reset state.The default value is 0. After this bit is written 1, the module is always in the reset state. You need to write 0 to exit the reset
11	TIM0RST	TIM0 module reset enable 1: Module reset, 0: No reset. Write 0 to clear the reset state.The default value is 0. After this bit is written 1, the module is always in the reset state. You need to write 0 to exit the reset
10	PWM1RST	PWM1 module reset enable 1: Module reset, 0: No reset. Write 0 to clear the reset state.The default value is 0. After this bit is written 1, the module is always in the reset state. You need to write 0 to exit the reset
9	PWM0RST	PWM0 module reset enable 1: Module reset,

		0: No reset. Write 0 to clear the reset state.The default value is 0. After this bit is written 1, the module is always in the reset state. You need to write 0 to exit the reset
8	IICRST	IIC module reset enable 1: Module reset, 0: No reset. Write 0 to clear the reset state.The default value is 0. After this bit is written 1, the module is always in the reset state. You need to write 0 to exit the reset
7	UART4RST	UART4 module reset enable 1: Module reset, 0: No reset. Write 0 to clear the reset state.The default value is 0. After this bit is written 1, the module is always in the reset state. You need to write 0 to exit the reset
6	UART3RST	UART3 module reset enable 1: Module reset, 0: No reset. Write 0 to clear the reset state.The default value is 0. After this bit is written 1, the module is always in the reset state. You need to write 0 to exit the reset
5	UART2RST	UART2 module reset enable 1: Module reset, 0: No reset. Write 0 to clear the reset state.The default value is 0. After this bit is written 1, the module is always in the reset state. You need to write 0 to exit the reset
4	UART1RST	UART1 module reset enable 1: Module reset, 0: No reset. Write 0 to clear the reset state.The default value is 0. After this bit is written 1, the module is always in the reset state. You need to write 0 to exit the reset
3	UART0RST	UART0 module reset enable 1: Module reset, 0: No reset. Write 0 to clear the reset state.The default value is 0. After this bit is written 1, the module is always in the reset state. You need to write 0 to exit the reset
2	SPI1 RST	SPI1 module reset enable 1: Module reset, 0: No reset. Write 0 to clear the reset state.The default value is 0. After this bit is written 1, the module is always in the reset state. You need to write 0 to exit the reset
1	SPI0 RST	SPI0 module reset enable 1: Module reset, 0: No reset. Write 0 to clear the reset state.The default value is 0. After this bit is written 1, the module is always in the reset state. You need to

		write 0 to exit the reset
0	-	Reserved

5.4. BOR threshold and delay selection

Ta ＝ 25℃

Power down delay configuration DELAY_SEL	Power down threshold configuration BOR_VTH_SEL	Power down threshold (V)	Recovery threshold (V)	Hysteresis (mV)	Delay (μs)
				Typical	Typical
0	000	4.2	4.3	129.02	239
	001				
	010				
	011	3.7	3.8	120.43	234
	100	3.3	3.4	144.03	199
	101	2.8	2.9	139.91	207
1	000	4.2	4.3	134.37	116
	001				
	010				
	011	3.7	3.8	125.3	112
	100	3.3	3.4	148.46	107
	101	2.8	2.9	143.7	116

Table 5.2 BOR threshold and delay selection

Chapter 6 System working mode

6.1. Working mode description

The working mode of BF7707AMXX: Active mode, debug mode, standby mode.

- **Active Mode**

The RC1M, PLL, HCLK, LIRC work, XTAL depends on software settings. The core is running, the peripherals keep working normally, and the functions of each peripheral are controlled by software configuration.

- **Debug Mode**

Debugging is performed by transmitting commands through the SWD port.

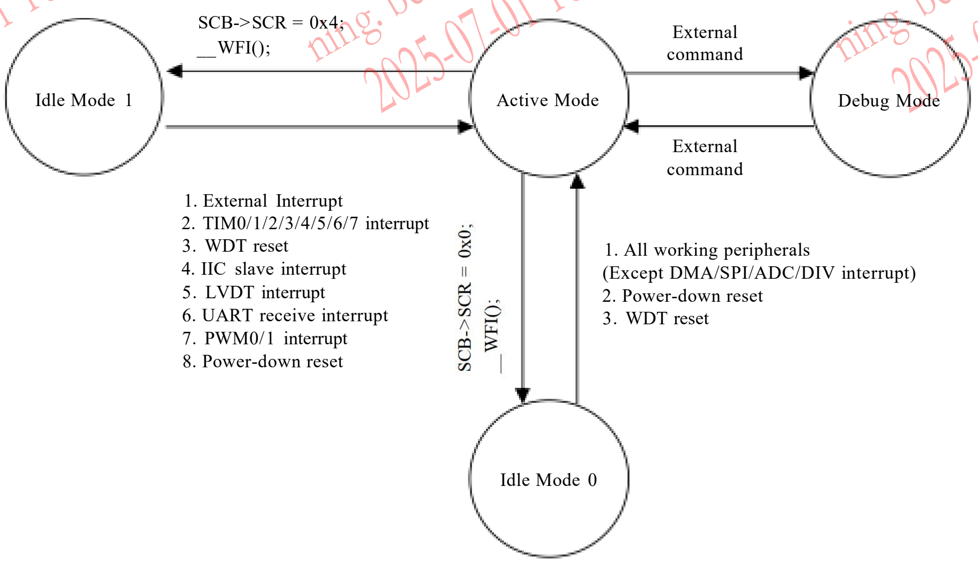
- **Standby mode is divided into idle mode 0 and idle mode 1**
  - Idle Mode 0

    Only the HCLK clock is turned off, RC1M, PLL and LIRC work, XTAL depends on software settings. The core stops running, and the rest ofthe peripherals that do not need HCLK as the working clock can work.
  - Idle Mode 1

    RC1M, PLL and HCLK are off, LIRC works, XTAL depends on software settings. The core stops running, and the peripherals using the LIRC clock work fine.

The working status of the clock source:

Work mode	Entry conditions	Effects on the clock	
Active Mode	Wake-up from power-on reset/standby mode	RC1M	Work
		PLL	Work
		HCLK	Work
		LIRC	Work
		XTAL	Depends on software configuration
Idle Mode 0	SCB->SCR = 0x0; __WFI();	RC1M	Work
		PLL	Work
		HCLK	Off
		LIRC	Work
		XTAL	Depends on software configuration
Idle Mode 1	SCB->SCR = 0x4; __WFI();	RC1M	Off
		PLL	Off
		HCLK	Off
		LIRC	Work
		XTAL	Depends on software configuration



Working mode conversion diagram

6.1.1. Active mode

Reduce system clock speed

In active mode, the system clock (HCLK) speed can be reduced by configuring the clock configuration register (CLK_CFG).

Peripheral clock gating

In active mode, peripherals can be stopped at any time to reduce power consumption.

To further reduce power consumption in idle mode, peripheral clocks can be disabled before executing a WFI instruction. Peripheral clock gating is controlled by the peripheral module clock control register (RCU_EN).

6.1.2. Debug mode

Debugging is performed by transmitting commands through the SWD port.

If the user application puts the MCU in standby mode, debugging will break because the core clock is stopped.

6.1.3. Low power management

The BF7707AMXX saves all CPU states before entering standby mode, SRAM and register contents are preserved, and GPIOs remain in run-time state.

Enter idle mode 0

SCB->SCR = 0x0; __WFI();

Enter idle mode 1

SCB->SCR = 0x4; __WFI();

Status of peripherals in standby mode:

Module	Clock	Standby mode	
		Idle mode 0	Idle mode 1
Cortex-M0+	HCLK	×	×
SRAM	HCLK	×	×
SYS_SFR	HCLK	×	×
SysTick	HCLK	×	×
FLASH	HCLK/Divide by 2 of PLL_48M/PGC	×	×
DMA	HCLK	×	×
UART0/1/2/3/4	PLL_48M	√	×
PWM0/1	LIRC/PLL_48M/XTAL	√	√
WDT	LIRC	√	√
TIM0/1/2/3/4/5/6/7	PLL_48M/LIRC/XTAL	√	√
ADC	PLL_48M	√	×
SPI	HCLK/SPI_CLK	×	×
IIC	Divide by 2 of PLL_48M/SCL	√	×(√)
DIV	HCLK	×	×
CRC	HCLK	×	×
GPIO	HCLK	×	×

Note: In the figure, √ means support, according to the program configuration. × means unavailable.

Exit idle mode 0

WDT interrupt, external interrupt, IIC slave interrupt, UART0/1/2/3/4 interrupt, Timer0/1/2/3/4/5/6/7 interrupt, PWM0/1 interrupt, LVDT interrupt, any of which can wake up the chip, exit idle mode 0, and the CPU executes the interrupt service routine. External reset, WDT reset can also wake up the chip to exit idle mode 0.

Exit idle mode 1

WDT interrupt, external interrupt, IIC slave interrupt, Timer 0/1/2/3/4/5/6/7 interrupt, LVDT interrupt, UART receive interrupt, PWM0/1 interrupt, any of which can wake up the chip, exit idle mode 1, and the CPU executes the interrupt service routine. Power-down reset, and WDT reset can also wake up the chip to exit idle mode 1.

6.2. Register

Enter idle mode 0 or idle mode 1, which is determined by the register SCR.SLEEPDEEP bit.

Address	Register	Description
0xE000ED10	SCR	System control register

Bit	Bit symbol	Description	RW	Reset value
31:3	Res.	-	-	-
2	SLEEPDEEP	Select standby mode register 0: Enter idle mode 0 1: Enter idle mode 1	R/W	0
1	SLEEPONEXIT	0: When exiting exception handling and returning to the program thread, do not enter standby mode (WFI) 1: When exiting exception handling and returning to the program thread, the processor automatically enters standby mode (WFI)	R/W	0
0	Res.	-	-	-

Chapter 7 WDT

7.1. Overview

The watchdog timer is timed according to the system configuration and generates a timing overflow signal.

Watchdog timed overflow reset: reset generated in normal mode; In standby mode, the option byte is selected by default to generate a reset. If a watchdog timing overflow occurs, the overflow signal is the watchdog overflow reset signal, which affects the global reset. At this time, the system implements a global reset action and reloads the configuration information.

The watchdog module is a timing counting module, with an internal low-speed clock LIRC as the counting clock. Its timing reset signal is composed of global reset and configuration reset. This signal is synchronously released by the watchdog timing clock in the reset module; For the reset action, configure the WDT overflow value configuration register WDT every time the CPU is configured WDT_TIME_SEL or watchdog timer enable configuration register WDT_EN occurs, the watchdog restarts timing; At the same time, the watchdog counter has a watchdog count enable control. When the count enable is effective, after the watchdog generates a timing overflow (reset), as long as the watchdog count enable is not turned off, the watchdog counter will restart counting.

7.2. Features

- The WDT clock is provided by internal low-speed clock LIRC 32kHz, which can work in standby mode
- Timing range: 18ms~2.304s
- Option byte controls WDT_EN, enabled by default
- In debug mode, it can be configured to automatically pause at the current value and continue counting after exiting

7.3. Registers

Base address: 0x5000 0000

Address offset	Register	Description
0x04	RCU_EN	Peripheral module clock control register

Base address: 0x5001 0000

Address offset	Register	Description
0x1C	FMC_CFG	CFG register

Base address: 0x5005 0000

Address offset	Register	Description
0x00	WDT_EN	Watchdog timer enable configuration register
0x04	WDT_TIME_SEL	Watchdog overflow timing configuration register

7.3.1. Peripheral module clock control register (RCU_EN)

Address offset: 0x04

Reset value: 0x0000 0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.				DIV_ CLKEN	GPIOD_ CLKEN	GPIOC_ CLKEN	GPIOB_ CLKEN	GPIOA_ CLKEN	DMA_ CLKEN	CRC_ CLKEN	ADC_ CLKEN	WDT_ CLKEN	TIM7_ CLKEN	TIM6_ CLKEN	TIM5_ CLKEN
				RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

19	WDT_CLKEN	WDT module operation enable 1: Work 0: Off, the default is 0

7.3.2. FMC CFG register (FMC_CFG)

Address offset: 0x1C

Reset value: 0xFFFF FFFF

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

15:5	4	3	2	1	0
Res.	WDT_EN	WDTRST_SD	WDTRST_SP	RDPRT	CFGERR
	R	R	R	R	R

31:5	-	Reserved
4	WDT_EN	User option bytes loaded by NVR1 0: The program is not allowed to turn off the watchdog function 1: Allow the program to turn off the watchdog function
3	WDTRST_SD	User option bytes loaded by NVR1

		0: When entering idle mode 1, a watchdog reset is generated 1: Reserved
2	WDTRST_SP	User option bytes loaded by NVR1 0: When entering idle mode 0, a watchdog reset is generated 1: Reserved

7.3.3. Watchdog timing enable configuration register (WDT_EN)

Address offset: 0x00

Reset value: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.								EN[7:0]							
								RW							

31:8	-	Reserved
7:0	EN[7:0]	Watchdog timer enable configuration register When the configuration value of this register is 0x55, the watchdog is turned off, and when it is configured to other values, the watchdog is turned on Configuring this register will clear the WDT counter Option Byte controls whether the enable is closed

7.3.4. Watchdog overflow timing configuration register (WDT_TIME_SEL)

Address offset: 0x04

Reset value: 0x0000 0007

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

15:3	2	1	0
Res.	TIME_SEL[2:0]		
	RW		

31:3	-	Reserved
2:0	TIME_SEL[2:0]	Watchdog overflow timing configuration register, the timing length is as follows: 000: 18ms 001: 36ms 010: 72ms 011: 144ms 100: 288ms 101: 576ms 110: 1152ms 111: 2304ms

Chapter 8 GPIO port

8.1. GPIO port description

Some pins ofthe GPIO port are multiplexed with the peripheral functions of the device, and they cannot be configured for multiple functions at the same time, otherwise it will cause functional disorder.The priority of special functions is higher than that of GPIO functions

8.2. GPIO function feature

Configure the corresponding registers to achieve the following features:

- Port mode selection (Px_MODR)
  Each pin function is defined with 2 bits; The pin configuration 00 is configured as input; The pin configuration 01 is configured as output; The pin configuration 10 is configured in multiplexing mode; The pin configuration 11 is configured in analog mode
- Output high and low level selection (Px_PDOR)
  Configuration 1 configures the corresponding pin to output high, write 0 to configure the corresponding pin to output low.
- Output high and low level read (Px_PDIR)
  Read is 1, the pin logic level is high; read is 0, the pin logic level is low.
- Internal pull-up resistor (Px_PPUR)
  The corresponding pin pull-up resistor of configuration 1 is enabled, and the corresponding pin of configuration 0 do not enable the pull-up resistor, and the pull-up resistor is 12k.
- Port setting output (Px_PSOR)
  The register is a write-only register. The configuration of the register can update the content of the corresponding bit in Px_PDOR.If the value is set to 1, set the corresponding bit in Px_PDOR to logical level 1.If the value is set to 0, the corresponding bit in Px_PDOR does not change.
- Port clearing output (Px_PCOR)
  The register is a write-only register. The configuration of the register can update the content of the corresponding bit in Px_PDOR.If the value is set to 1, set the corresponding bit in Px_PDOR to logical level 0.If the value is set to 0, the corresponding bit in Px_PDOR does not change.
- Port switching output (Px_PTOR)
  The register is a write-only register. The configuration of the register can update the content of the corresponding bit in Px_PDOR.If the value is set to 1, the corresponding bit in Px_PDOR is set to the opposite state of its existing logical state.If the value is set to 0, the corresponding bit in Px_PDOR does not change.
- Port multiplexing function selection (Px_AFSEL0/1)
  Each pin function defines the mode with 4 bits, and configuration 0~3 corresponds to

AF0~AF3

- Open drain output (ODRAIN_EN)

PD11/PD10/PD4/PD3/PD0/PC15/PC12/PC11/PC3/PC2/PB15/PB14/PC1/PB2/PB1/PD9/PD8 have drain function.Configure the corresponding pin in 1 to enable open drain output. Clear to disable open drain output. Enable the open drain output automatically after the IIC function is enabled.An external pull-up resistor is recommended for IIC/UART

- Support 8 large current drive function of GPIO port PA0 ~ PA7 (COM_IO_SEL)

Set this parameter to 1, select COM function. Set this parameter to 0, select common I/O function.

- All ports can provide external interrupt, and each interrupt can be configured as rising edge trigger, falling edge trigger, double edge trigger, see chapter "External interrupt" for details.

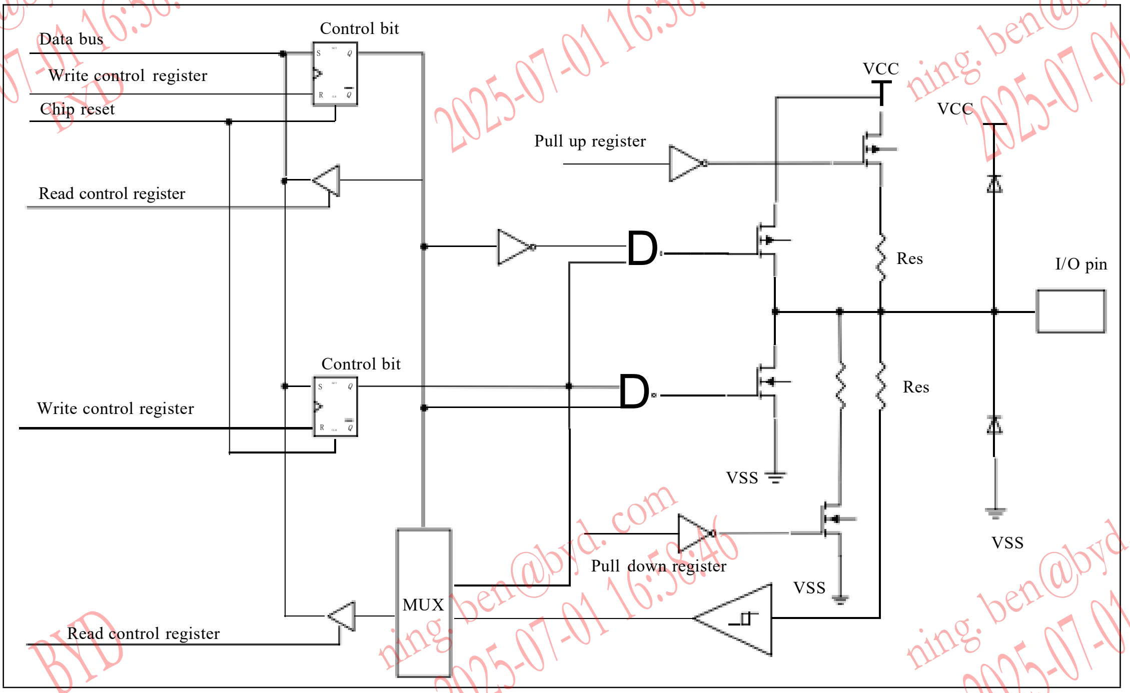


Figure 8.1 Common IO structure diagram

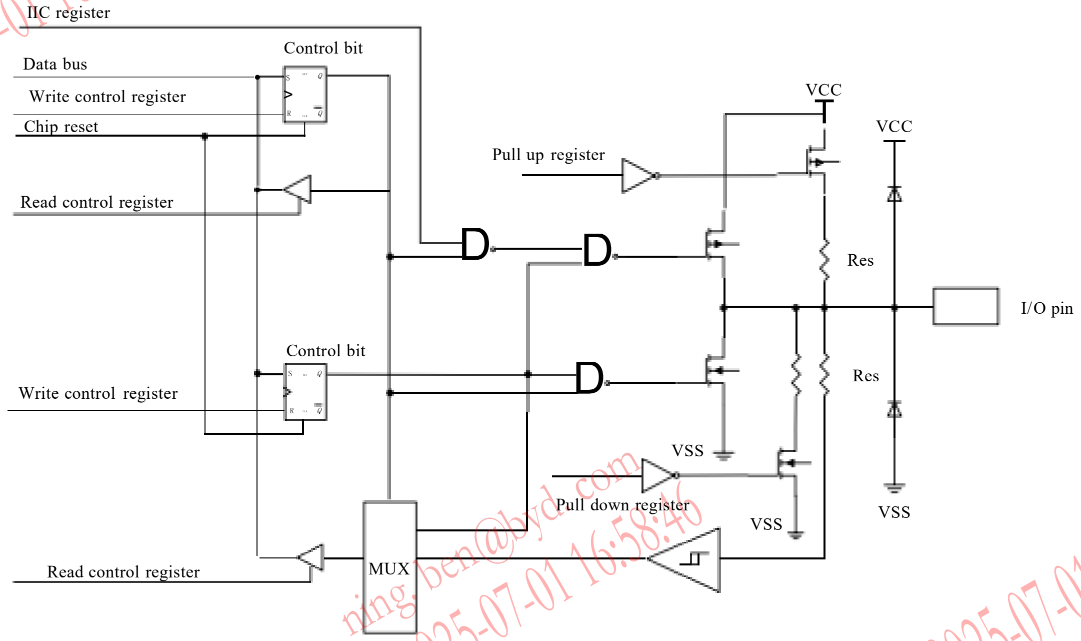


Figure 8.2 Open drain output IO structure diagram

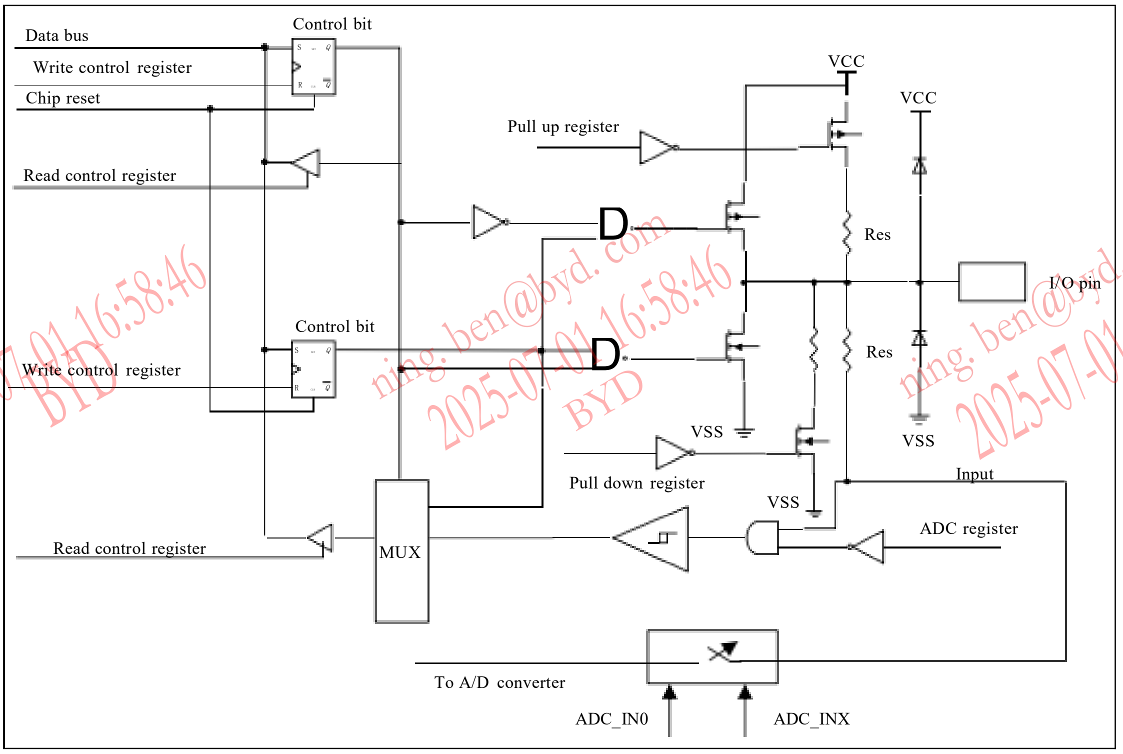


Figure 8.3 ADC IO structure diagram

8.3. GPIO pin multiplexing configuration

GPIO	AF0	AF1	AF2	AF3	EXTI	Analog input ADC	Other functions
PA0	-	-	-	-	INTx_y	-	
PA1	-	-		-	INTx_y	-	XTALB_OUT
PA2	-	-	-	-	INTx_y	-	EXCLKB_IN/XTALB_IN
PA3	-	-	-		INTx_y	-	
PA4	-	-	-	-	INTx_y	-	XTALA_OUT
PA5	-	-	-	-	INTx_y	-	EXCLKA_IN/XTALA_IN
PA6	SWCLK	R_TXDx	-	-	INTx_y	-	PGCA
PA7	SWDIO	R_TXDx	-	-	INTx_y	-	PGDA
PA8	TI1_CH1A/TO1_CH1A	-	-	-	INTx_y	-	-
PA9	TI4_CH1A/TO4_CH1A	-	-	-	INTx_y	ADC00	-
PA10	-	-	-	-	INTx_y	-	-
PA11	-	-	-	-	INTx_y	ADC01	-
PA12	PWM1A	-	-	-	INTx_y	ADC02	-
PA13	PWM0A	-	-	-	INTx_y	ADC03	-
PA14	TI0_CH1A	R_TXDx	SPI0B_CS	-	INTx_y	ADC04	-
PA15	TO0_CH1A	R_TXDx	SPI0B_CLK	-	INTx_y	-	-
PB0	-	R_TXDx	-	-	INTx_y	ADC05	-
PB1	SDA0H	R_TXDx	-	-	INTx_y	ADC06	-
PB2	SCL0H	-	-	-	INTx_y	ADC07	-
PB3	TI0_CH1/TO0_CH1	-	-	-	INTx_y	ADC08	-
PB4	TO0_CH1N	-	SPI0B_MOSI	-	INTx_y	ADC09	-
PB5	TI0_CH2/TO0_CH2	R_TXDx	SPI0B_MISO	-	INTx_y	ADC10	-
PB6	TO0_CH2N	R_TXDx			INTx_y	ADC11	
PB7	TI0_CH3/TO0_CH3	-	-	-	INTx_y	ADC12	-
PB8	TO0_CH3N	-	-	-	INTx_y	ADC13	-
PB9	TI0_BKIN1	-	-	-	INTx_y	ADC14	
PB10	TI0_BKIN2	-	SPI0A_CLK	-	INTx_y	ADC15	
PB11	TI0_CH4/TO0_CH4	-	SPI0A_MOSI	-	INTx_y	ADC16	-
PB12	-	R_TXDx	SPI0A_MISO	-	INTx_y	ADC17	-
PB13	PWM1C	R_TXDx	SPI0A_CS	-	INTx_y	-	-
PB14	-	TI1_CH1/TO1_CH1	SCL0A	-	INTx_y	-	-
PB15	TI2_CH1/TO2_CH1	R_TXDx	SDA0A	-	INTx_y	-	PGCB
PC0	TI3_CH1/TO3_CH1	R_TXDx	--	-	INTx_y	-	PGDB
PC1	TI4_CH1/TO4_CH1	R_TXDx	SDA0B	-	INTx_y	-	-
PC2	TI5_CH1/TO5_CH1	R_TXDx	SCL0B	SDA0C	INTx_y	-	-
PC3	TI6_CH1/TO6_CH1	PWM0C	SPI1A_CLK	SCL0C	INTx_y	-	-
PC4	TI7_CH1/TO7_CH1	R_TXDx	SPI1A_MOSI	-	INTx_y	-	-
PC5	TI6_CH2/TO6_CH2	R_TXDx	SPI1A_MISO	-	INTx_y		-
PC6	PWM0B	R_TXDx	-	-	INTx_y	-	-
PC7	-	-	-	-	INTx_y	-	-
PC8	-	-	-	-	INTx_y	-	-
PC9	-	-	-	-	INTx_y	-	

PC10	TI0_ETR	-	SPI1A_CS	-	INTx_y	-	-
PC11	-	-	SDA0D	-	INTx_y	-	-
PC12	TI6_CH3/TO6_CH3	-	SCL0D	-	INTx_y	-	-
PC13	TI3_CH1A/TO3_CH1A	-	-	-	INTx_y	-	-
PC14	TI2_CH1A/TO2_CH1A	-	-	-	INTx_y	-	-
PC15	TI6_CH4/TO6_CH4	SCL0E	-	-	INTx_y	-	-
PD0	TI6_ETR	SDA0E	-	-	INTx_y	-	-
PD1	-	-	-	-	INTx_y	-	-
PD2	-	R_TXDx		-	INTx_y	-	-
PD3	TI2_CH1B/TO2_CH1B	R_TXDx	SDA0F	-	INTx_y	-	-
PD4	TI1_CH1B/TO1_CH1B	-	SCL0F	-	INTx_y	-	-
PD5	-	R_TXDx	-	-	INTx_y	-	PGCC
PD6	-	R_TXDx	-	-	INTx_y	-	-
PD7	TI5_CH1A/TO5_CH1A	R_TXDx	PWM1B	-	INTx_y	-	PGDC
PD8	-	-	-	-	INTx_y	-	-
PD9	TI3_CH1B/TO3_CH1B	R_TXDx	-	-	INTx_y	-	-
PD10	SDA0G	-	-	-	INTx_y	-	-
PD11	SCL0G	-	-	-	INTx_y	-	-

Table 8.1 GPIO pin multiplexing configuration table

Note:

1. External interrupt INTx_y can be mapped from all IO ports, where x=0/1/2/3, y=0/1/2/3, and the EXTIx interrupt source selects the register EXTIx_SEL, determine INTx_y Which IO port mapped to.
2. Support GPIO for R_TXDx（x = 0/1/2/3/4）, available through Px_UART register selects which TXD or RXD of the UART to reuse.
3. Support GPIO of TIx_CHy/TOx_CHy can be achieved through TIMx_DIR register configuration selection multiplexed to TIx_CHy or TOx_CHy.

8.4. Registers

Base address: 0x5000 0000

Address offset	Register	Description
0x04	RCU_EN	Peripheral module clock control register
0x2C	COM_IO_SEL	COM port selection configuration register
0x34	ODRAIN_EN	Port open drain output enable register

PA base address: 0x5000_2000; PB base address: 0x5000_2300
PC base address: 0x5000_2600; PD base address: 0x5000_2900 x=A/B/C/D

Address offset	Register	Description
0x00	Px_MODR	Px port mode register
0x04	Px_PDIR	Px port data input register
0x08	Px_PDOR	Px port data output register
0x0C	Px_PPUR	Px port pull-up resistor control register
0x10	Px_PSOR	Px port setting output register
0x14	Px_PCOR	Px port clearing output register
0x18	Px_PTOR	Px port switching output register
0x1C	Px_UART	Px port UART control register
0x80	Px_AFSEL0	Px port multiplexing function register 0
0x84	Px_AFSEL1	Px port multiplexing function register 1

8.4.1. Peripheral module clock control register (RCU_EN)

Address offset: 0x04

Reset value: 0x0000 0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.				DIV_ CLKEN	GPIOD_ CLKEN	GPIOC_ CLKEN	GPIOB_ CLKEN	GPIOA_ CLKEN	DMA_ CLKEN	CRC_ CLKEN	ADC_ CLKEN	WDT_ CLKEN	TIM7_ CLKEN	TIM6_ CLKEN	TIM5_ CLKEN
				RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

26	GPIOD_CLKEN	GPIOD module operation enable 1: Work 0: Off, the default is 0
25	GPIOC_CLKEN	GPIOC module operation enable 1: Work, 0: Off, the default is 0
24	GPIOB_CLKEN	GPIOB module operation enable 1: Work 0: Off, the default is 0
23	GPIOA_CLKEN	GPIOA module operation enable 1: Work 0: Off, the default is 0

8.4.2. COM port selection register(COM_IO_SEL)

Address offset: 0x2C

Reset value: 0x0000 0000

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

15:8	7	6	5	4	3	2	1	0
Res.	COM7	COM6	COM5	COM4	COM3	COM2	COM1	COM0
	RW	RW	RW	RW	RW	RW	RW	RW

31:8	-	Reserved
7:0	COM[7:0]	COM port selection configuration register, Bit[0]~Bit[7] corresponds to PA0~PA7 port 1: Select the COM function 0: Select the common I/O function

8.4.3. Open drain output enable register(ODRAIN_EN)

Address offset: 0x34

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14
Res.															OD16	OD15	OD14
															RW	RW	RW

13	12	11	10	9	8	7	6	5	4	3	2	1	0
OD13	OD12	OD11	OD10	OD9	OD8	OD7	OD6	OD5	OD4	OD3	OD2	OD1	OD0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

31:17	-	Reserved
16	OD16	PD8 port open drain output enable. 1: Open drain output; 0: CMOS output.
15	OD15	PD9 port open drain output enable. 1: Open drain output; 0: CMOS output.
14	OD14	PB1 port open drain output enable. 1: Open drain output; 0: CMOS output.
13	OD13	PB2 port open drain output enable. 1: Open drain output; 0: CMOS output.
12	OD12	PC1 port open drain output enable. 1: Open drain output; 0: CMOS output.
11	OD11	PB14 port open drain output enable. 1: Open drain output; 0: CMOS output.

10	OD10	PB15 port open drain output enable. 1: Open drain output; 0: CMOS output.
9	OD9	PC2 port open drain output enable. 1: Open drain output; 0: CMOS output.
8	OD8	PC3 port open drain output enable. 1: Open drain output; 0: CMOS output.
7	OD7	PC11 port open drain output enable. 1: Open drain output; 0: CMOS output.
6	OD6	PC12 port open drain output enable. 1: Open drain output; 0: CMOS output.
5	OD5	PC15 port open drain output enable. 1: Open drain output; 0: CMOS output.
4	OD4	PD0 port open drain output enable. 1: Open drain output; 0: CMOS output.
3	OD3	PD3 port open drain output enable. 1: Open drain output; 0: CMOS output.
2	OD2	PD4 port open drain output enable. 1: Open drain output; 0: CMOS output.
1	OD1	PD10 port open drain output enable. 1: Open drain output; 0: CMOS output.
0	OD0	PD11 port open drain output enable. 1: Open drain output; 0: CMOS output.

Note: When the IIC is enabled, the open drain output of the corresponding port is automatically enabled, and 8 channels ofIIC mapping are enabled.

8.4.4. Px port mode register (Px_MODR) (x=A/B/C/D)

Address offset: 0x00

Reset value: 0x0000 A000 (PA), 0x0000 0000 (Other ports)

31 30	29 28	27 26	25 24	23 22	21 20	19 18	17 16
MODE15	MODE14	MODE13	MODE12	MODE11	MODE10	MODE9	MODE8
RW	RW	RW	RW	RW	RW	RW	RW

15 14	13 12	11 10	9 8	7 6	5 4	3 2	1 0
MODE7	MODE6	MODE5	MODE4	MODE3	MODE2	MODE1	MODE0
RW	RW	RW	RW	RW	RW	RW	RW

31:0	MODEy	Port mode register The bit width [31:0] corresponds to 16 pins respectively, each pin function is defined with 2 bits. MODEy[1:0]: The port is configured with I/O pin y(y = 0 ~ 15) These bits are written by software to configure the I/O mode.

		00: Input mode 01: Universal output mode 10: Reuse function mode 11: Analog mode note: 1. x=A/B/C/D, PD0~PD11corresponds to PD_MODR[0:23], PD_MODR[24:31] bit reserved. 2. The lock analog mode configuration is invalid after the SW mode is entered.

8.4.5. Px port data input register (Px_PDIR) (x=A/B/C/D)

Address offset: 0x04

Reset value: 0x0000 xxxx

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PDI15	PDI14	PDI13	PDI12	PDI11	PDI10	PDI9	PDI8	PDI7	PDI6	PDI5	PDI4	PDI3	PDI2	PDI1	PDI0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

31:16	-	Reserved
15:0	PDIy	Port data input, I/O pin y (y = 15 ~ 0) 0: The pin logic level is 0 1: The pin logic level is 1

8.4.6. Px port data output register (Px_PDOR) (x=A/B/C/D)

Address offset: 0x08

Reset value: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PDO15	PDO14	PDO13	PDO12	PDO11	PDO10	PDO9	PDO8	PDO7	PDO6	PDO5	PDO4	PDO3	PDO2	PDO1	PDO0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

31:16	-	Reserved
15:0	PDOy	Port data output, I/O pin y (y = 15 ~ 0) 0: If the port is configured for general output mode, drive logic level is 0 on the pin 1: If the port is configured for general output mode, drive logic level is 1 on the pin

8.4.7. Px port pull-up resistor control register (Px_PPUR) (x=A/B/C/D)

Address offset: 0x0C

Reset value: 0x0000 00C0(PA)0x0000 0000(Other ports)

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PPU15	PPU14	PPU13	PPU12	PPU11	PPU10	PPU9	PPU8	PPU7	PPU6	PPU5	PPU4	PPU3	PPU2	PPU1	PPU0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

31:16	-	Reserved
15:0	PPUy	Port pull-up resistor control register, I/O port y(y = 15 ~ 0) 1: Pull-up resistor is enable 0: Pull-up resistor is disable Note: PA6 is PGCA and PA7 is PGDA. Pull-up resistor is enabled by default

8.4.8. Px port set output register (Px_PSOR) (x=A/B/C/D)

Address offset: 0x10

Reset value: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PTSO15	PTSO14	PTSO13	PTSO12	PTSO11	PTSO10	PTSO9	PTSO8	PTSO7	PTSO6	PTSO5	PTSO4	PTSO3	PTSO2	PTSO1	PTSO0
WR0	WR0	WR0	WR0	WR0	WR0	WR0	WR0	WR0	WR0	WR0	WR0	WR0	WR0	WR0	WR0

31:16	-	Reserved
15:0	PTSOy	Port set output, I/O port y(y = 15 ~ 0) Write this register to update the corresponding bit in register Px_PDOR, only write the register. 0: The corresponding bit in Px_PDOR remains unchanged 1: The corresponding bit in Px_PDOR is set to logical level 1

8.4.9. Px port clear output register(Px_PCOR) (x=A/B/C/D)

Address offset: 0x14

Reset value: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PTCO15	PTCO14	PTCO13	PTCO12	PTCO11	PTCO10	PTCO9	PTCO8	PTCO7	PTCO6	PTCO5	PTCO4	PTCO3	PTCO2	PTCO1	PTCO0
WR0	WR0	WR0	WR0	WR0	WR0	WR0	WR0	WR0	WR0	WR0	WR0	WR0	WR0	WR0	WR0

31:16	-	Reserved
15:0	PTCOy	Port clear output, I/O port y(y = 15 ~ 0) Write this register to update the corresponding bit in register Px_PDOR, only write the register. 0: The corresponding bit in Px_PDOR remains unchanged 1: The corresponding bit in Px_PDOR is set to logical level 0

8.4.10. Px port switch output register (Px_PTOR) (x=A/B/C/D)

Address offset: 0x18

Reset value: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PTTO15	PTTO14	PTTO13	PTTO12	PTTO11	PTTO10	PTTO9	PTTO8	PTTO7	PTTO6	PTTO5	PTTO4	PTTO3	PTTO2	PTTO1	PTTO0
WR0	WR0	WR0	WR0	WR0	WR0	WR0	WR0	WR0	WR0	WR0	WR0	WR0	WR0	WR0	WR0

31:16	-	Reserved
15:0	PTTOy	Port switch output Write this register to update the corresponding bit in register Px_PDOR, only write the register, read back to 0 0: The corresponding bit in Px_PDOR remains unchanged 1: The corresponding bit in Px_PDOR is set to the opposite of its existing logical state

8.4.11. PA port UART control register (PA_UART)

Address offset: 0x1C

Reset value: 0x0000 0000

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

15	14:12	11	10:8	7	6:4	3	2:0
TXRX_SEL3	UART_SEL3	TXRX_SEL2	UART_SEL2	TXRX_SEL1	UART_SEL1	TXRX_SEL0	UART_SEL0
RW	RW	RW	RW	RW	RW	RW	RW

31:16	-	Reserved
15	TXRX_SEL3	When PA15 select the UART function, TXD/RXD pin selection,

		1: TXD function 0: RXD function
14:12	UART_SEL3	Used to select the UART module for PA15 port 000: UART0 001: UART1 010: UART2 011: UART3 1xx: UART4
11	TXRX_SEL2	When PA14 select the UART function, TXD/RXD pin selection, 1: TXD function 0: RXD function
10:8	UART_SEL2	Used to select the UART module for PA14 port: 000: UART0 001: UART1 010: UART2 011: UART3 1xx: UART4
7	TXRX_SEL1	When PA7 select the UART function, TXD/RXD pin selection, 1: TXD function 0: RXD function
6:4	UART_SEL1	Used to select the UART module for PA7 port: 000: UART0 001: UART1 010: UART2 011: UART3 1xx: UART4
3	TXRX_SEL0	When PA6 select the UART function, TXD/RXD pin selection, 1: TXD function 0: RXD function
2:0	UART_SEL0	Used to select the UART module for PA6 port: 000: UART0 001: UART1 010: UART2 011: UART3 1xx: UART4

8.4.12. PB port UART control register (PB_UART)

Address offset: 0x1C

Reset value: 0x0000 0000

31:28	27	26:24	23	22:20	19	18:16
Res.	TXRX_SEL6	UART_SEL6	TXRX_SEL5	UART_SEL5	TXRX_SEL4	UART_SEL4
	RW	RW	RW	RW	RW	RW

15	14:12	11	10:8	7	6:4	3	2:0
RXTX_SEL3	UART_SEL3	TXRX_SEL2	UART_SEL2	TXRX_SEL1	UART_SEL1	TXRX_SEL0	UART_SEL0
RW	RW	RW	RW	RW	RW	RW	RW

31:28	-	Reserved
27	TXRX_SEL6	When PB15 select the UART function, TXD/RXD pin selection, 1: TXD function, 0: RXD function
26:24	UART_SEL6	Used to select the UART module for PB15 port: 000: UART0 001: UART1 010: UART2 011: UART3 1xx: UART4

23	TXRX_SEL5	When PB13 select the UART function, TXD/RXD pin selection, 1: TXD function, 0: RXD function
22:20	UART_SEL5	Used to select the UART module for PB13 port: 000: UART0 001: UART1 010: UART2 011: UART3 1xx: UART4
19	TXRX_SEL4	When PB12 select the UART function, TXD/RXD pin selection, 1: TXD function, 0: RXD function
18:16	UART_SEL4	Used to select the UART module for PB12 port: 000: UART0 001: UART1 010: UART2 011: UART3 1xx: UART4
15	RXTX_SEL3	When PB6 select the UART function, TXD/RXD pin selection, 1: TXD function, 0: RXD function
14:12	UART_SEL3	Used to select the UART module for PB6 port: 000: UART0 001: UART1 010: UART2 011: UART3 1xx: UART4
11	TXRX_SEL2	When PB5 select the UART function, TXD/RXD pin selection, 1: TXD function, 0: RXD function
10:8	UART_SEL2	Used to select the UART module for PB5 port: 000: UART0 001: UART1 010: UART2 011: UART3 1xx: UART4
7	TXRX_SEL1	When PB1 select the UART function, TXD/RXD pin selection, 1: TXD function, 0: RXD function
6:4	UART_SEL1	Used to select the UART module for PB1 port: 000: UART0 001: UART1 010: UART2 011: UART3 1xx: UART4
3	TXRX_SEL0	When PB0 select the UART function, TXD/RXD pin selection, 1: TXD function, 0: RXD function
2:0	UART_SEL0	Used to select the UART module for PB0 port: 000: UART0 001: UART1 010: UART2 011: UART3 1xx: UART4

8.4.13. PC port UART control register (PC_UART)

Address offset: 0x1C

Reset value: 0x0000 0000

31:24	23	22:20	19	18:16
Res.	TXRX_SEL5	UART_SEL5	TXRX_SEL4	UART_SEL4
	RW	RW	RW	RW

15	14:12	11	10:8	7	6:4	3	2:0
TXRX_SEL3	UART_SEL3	TXRX_SEL2	UART_SEL2	TXRX_SEL1	UART_SEL1	TXRX_SEL0	UART_SEL0
RW	RW	RW	RW	RW	RW	RW	RW

31:24		Reserved
23	TXRX_SEL5	When PC6 select yhe UART function, TXD/RXD pin selection, 1: TXD function, 0: RXD function
22:20	UART_SEL5	Used to select the UART module for PC6 port: 000: UART0 001: UART1 010: UART2 011: UART3 1xx: UART4
19	TXRX_SEL4	When PC5 select the UART function, TXD/RXD pin selection, 1: TXD function, 0: RXD function
18:16	UART_SEL4	Used to select the UART module for PC5 port: 000: UART0 001: UART1 010: UART2 011: UART3 1xx: UART4
15	TXRX_SEL3	When PC4 select the UART function, TXD/RXD pin selection, 1: TXD function, 0: RXD function
14:12	UART_SEL3	Used to select the UART module for PC4 port: 000: UART0 001: UART1 010: UART2 011: UART3 1xx: UART4
11	TXRX_SEL2	When PC2 select the UART function, TXD/RXD pin selection, 1: TXD function, 0: RXD function
10:8	UART_SEL2	Used to select the UART module for PC2 port: 000: UART0 001: UART1 010: UART2 011: UART3 1xx: UART4
7	TXRX_SEL1	When PC1 select the UART function, TXD/RXD pin selection, 1: TXD function, 0: RXD function
6:4	UART_SEL1	Used to select the UART module for PC1 port: 000: UART0 001: UART1 010: UART2 011: UART3 1xx: UART4
3	TXRX_SEL0	When PC0 select the UART function, TXD/RXD pin selection, 1: TXD function, 0: RXD function
2:0	UART_SEL0	Used to select the UART module for PC0 port: 000: UART0 001: UART1 010: UART2 011: UART3 1xx: UART4

8.4.14. PD port UART control register (PD_UART)

Address offset: 0x1C
Reset value: 0x0000 0000

31:24	23	22:20	19	18:16
Res.	TXRX_SEL5	UART_SEL5	TXRX_SEL4	UART_SEL4
	RW	RW	RW	RW

15	14:12	11	10:8	7	6:4	3	2:0

TXRX_SEL3	UART_SEL3	TXRX_SEL2	UART_SEL2	TXRX_SEL1	UART_SEL1	TXRX_SEL0	UART_SEL0
RW	RW	RW	RW	RW	RW	RW	RW

31:24		Reserved
23	TXRX_SEL5	When PD9 select the UART function, TXD/RXD pin selection, 1: TXD function, 0: RXD function
22:20	UART_SEL5	Used to select the UART module for PD9 port: 000: UART0 001: UART1 010: UART2 011: UART3 1xx: UART4
19	TXRX_SEL4	When PD7 select the UART function, TXD/RXD pin selection, 1: TXD function, 0: RXD function
18:16	UART_SEL4	Used to select the UART module for PD7 port: 000: UART0 001: UART1 010: UART2 011: UART3 1xx: UART4
15	TXRX_SEL3	When PD6 select the UART function, TXD/RXD pin selection, 1: TXD function, 0: RXD function
14:12	UART_SEL3	Used to select the UART module for PD6 port: 000: UART0 001: UART1 010: UART2 011: UART3 1xx: UART4
11	TXRX_SEL2	When PD5 select the UART function, TXD/RXD pin selection, 1: TXD function, 0: RXD function
10:8	UART_SEL2	Used to select the UART module for PD5 port: 000: UART0 001: UART1 010: UART2 011: UART3 1xx: UART4
7	TXRX_SEL1	When PD3 select the UART function, TXD/RXD pin selection, 1: TXD function, 0: RXD function
6:4	UART_SEL1	Used to select the UART module for PD3 port: 000: UART0 001: UART1 010: UART2 011: UART3 1xx: UART4
3	TXRX_SEL0	When PD2 select the UART function, TXD/RXD pin selection, 1: TXD function, 0: RXD function
2:0	UART_SEL0	Used to select the UART module for PD2 port: 000: UART0 001: UART1 010: UART2 011: UART3 1xx: UART4

8.4.15. Px port multiplexing function register 0(Px_AFSEL0)

Address offset: 0x80

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AFSEL7				AFSEL6				AFSEL5				AFSEL4			
RW				RW				RW				RW			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AFSEL3				AFSEL2				AFSEL1				AFSEL0			
RW				RW				RW				RW			

31:0	AFSELy	Port multiplexing function register 0 Bit width [31:0] corresponds to 8 pins, each pin function is defined with 4 bits. AFSELy[3:0] : Multiplexing function selection on port x pin y (y = 0 ~ 7) These bits are written by software to configure multiplexed functional I/O. AFSELy selection: 0000: AF0 0001: AF1 0010: AF2 0011: AF3 Other: Reserved For the port multiplexing function, see "GPIO Pin Multiplexing Configuration Table".

8.4.16. Px port multiplexing function register 1(Px_AFSEL1)

Address offset: 0x84

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AFSEL15				AFSEL14				AFSEL13				AFSEL12			
RW				RW				RW				RW			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AFSEL11				AFSEL10				AFSEL9				AFSEL8			
RW				RW				RW				RW			

31:0	AFSELy	Port multiplexing function register 1 Bit width [31:0] corresponds to 8 pins, each pin function is defined with 4 bits. AFSELy[3:0] : Multiplexing function selection on port x pin y (y = 8 ~ 15) These bits are written by software to configure multiplexed functional I/O. AFSELy selection: 0000: AF0 0001: AF1 0010: AF2 0011: AF3 Other: Reserved Note: PD8 ~ PD11 correspond to PD_AFSELy[15:0], and PD_AFSELy[31:16] bit reserved For the port multiplexing function, see "GPIO Pin Multiplexing Configuration Table".

8.5. GPIO configuration process

PA6 and PA7 are SWD download and debugging function ports by default. You can configure the register PA_MODR to be set to other modes.

8.5.1. GPIO input configuration

1. Configure the peripheral clock;
2. Configure the Px_MODR register to select the input mode;
3. Configure the Px_PPUR pull-up resistor register as required;
4. Read the Px_PDIR register to obtain the IO state.

8.5.2. GPIO output configuration

1. Configure the peripheral clock;
2. Configure the Px_MODR register to select the output mode;
3. Configure the Px_PPUR pull-up resistance register as required;
4. Configure Px_PDOR register to control IO output level;
5. Configure Px_PSOR, Px_PCOR, Px_PTOR these three registers can change the output state ofthe I/O port.

8.5.3. GPIO multiplexing function configuration

1. Configure the peripheral clock;
2. Configure the Px_MODR register to select the multiplexing mode;
3. Configure Px_AFSEL0/1 To enable port multiplexing.

Port multiplexing is configured for TIM function:

Select PA8 to the TI1_CH1/TO1_CH1 function of TIM1

1. Configure PA_MODR[7:6]= 0x2, set the port mode of PA8 to multiplexing function mode
2. Configure PA_AFSEL1[3:0]=0x0, Select AF0 (TI1_CH1/TO1_CH1) function for PA8

Note: The TIM1_DIR register controls whether TI1_CH1 is input or TO1_CH1 is output

Configure port reuse as SPI function:

Select PC3 configuration as SPI1A_ SCLK, select PC4 to configure as SPI1A_ MOSI, select PC5 to configure as SPI1A_ MISO, select PC10 to configure as SPI1A_ CS.

1. Configure PC_MODR[7:6]= 0x2, configure PC3 port mode as multiplexing function mode
2. Configure PC_MODR[9:8]= 0x2, configure PC4 port mode as multiplexing function mode
3. Configure PC_MODR[11:10]= 0x2, configure PC5 port mode as the multiplexing function mode
4. Configure PC_MODR[21:20]= 0x2, configure PC10 port mode as multiplexing function mode
5. Configure PC_AFSEL0[15:12]=0x2, select the AF2 (SPI1A_CLK) function of PC3
6. Configure PC_AFSEL0[19:16]=0x2, select the AF2 (SPI1A MOSI) function of PC4

7. Configure PC_AFSEL0[23:20]=0x2, select the AF2 (SPI1A-MISO) function of PC5
8. Configure PC_AFSEL1[11:8]=0x2, select the AF2 (SPI1A_CS) function of PC10

Configure port reuse as UART function:

Select RXD with PA6 configured as UART0 and TXD with PA7 configured as UART0

1. Configure PA_ MODR [13:12]=0x2, configure PA6 port mode as multiplexing function mode
2. Configure PA_ MODR [15:14]=0x2, configure PA7 port mode as multiplexing function mode
3. Configure PA_ AFSEL0 [27:24]=0x1, select the AF1 (RXD/TXD) function of PA6
4. Configure PA_AFSEL0[31:28]=0x1, select the AF1 (RXD/TXD) function of PA7
5. Configure PA_ UART [2:0]=0x0, configure PA6 as UART0 function
6. Configure PA_ UART [3]=0x0, configure PA6 as RXD function
7. Configure PA_ UART [6:4]=0x0, configure PA7 as UART0 function
8. Configure PA_ UART [7]=0x1, configure PA7 as TXD function

The port reuse configuration is IIC function:

Select clock line SCL with PB14 configured as IIC and data line SDA with PB15 configured as IIC

1. Configure PB_ MODR [29:28]=0x2, configure PB14 port mode as multiplexing function mode
2. Configure PB_ MODR [31:30]=0x2, configure PB15 port mode as multiplexing function mode
3. Configure PB_ AFSEL1 [27:24]=0x2, select the AF2 (SCL0A) function of PB14
4. Configure PB_ AFSEL1 [31:28]=0x2, select the AF2 (SDA0A) function of PB15.

Port multiplexing is configured as PWM:

Select PA12 to configure as PWM output

1. Configure PA_ MODR [25:24]=0x2, configure PA12 port mode as multiplexing function mode.
2. Configure PA_ AFSEL1 [19:16]=0x00, select the AF0 (PWM1A) function of PA12.

8.5.4. GPIO simulation function configuration

1. Configure peripheral clock;
2. Configure Px_ MODR register selection Analog mode;

Chapter 9 Interrupt controller(NVIC)

9.1. Interrupt source and entry address

There are 29 interrupt sources in BF7707AMXX. If multiple IO are used as External interrupt to multiplex the same interrupt entry, you should refer to the interrupt source settings.

The built-in simple 24-bit down-counting timer (SysTick) is used as the tick timer of the real-time operating system (RTOS), or as a simple counter. SysTick counts down from a preset value, and when it reaches zero, a system interrupt is generated. For details, see the "SysTick timer" chapter.

Interrupt priority	Interrupt number	CPU interrupt list	Interrupt vector address	Description
-2	-14	NMI	0x08	Reserved
-1	-13	HardFault	0x0C	Hardware error
Programmable	-5	SVCall	0x2C	Request management call realized by SVC
Programmable	-2	PendSV	0x38	Suspendable system service request
Programmable	-1	SysTick	0x3C	System tick timer
Programmable	0	IRQ0	0x40	INT_DIV
Programmable	1	IRQ1	0x44	INT_LVDT
Programmable	2	IRQ2	0x48	INT_EXTI0
Programmable	3	IRQ3	0x4C	INT_EXTI1
Programmable	4	IRQ4	0x50	INT_EXTI2
Programmable	5	IRQ5	0x54	INT_EXTI3
Programmable	6	IRQ6	0x58	INT_IIC
Programmable	7	IRQ7	0x5C	INT_UART0
Programmable	8	IRQ8	0x60	INT_UART1
Programmable	9	IRQ9	0x64	INT_UART2
Programmable	10	IRQ10	0x68	INT_UART3
Programmable	11	IRQ11	0x6C	INT_UART4
Programmable	12	IRQ12	0x70	INT_SPI0
Programmable	13	IRQ13	0x74	INT_SPI1
Programmable	14	IRQ14	0x78	INT_TIM0_BC
Programmable	15	IRQ15	0x7C	INT_TIM0_CC
Programmable	16	IRQ16	0x80	INT_TIM1
Programmable	17	IRQ17	0x84	INT_TIM2
Programmable	18	IRQ18	0x88	INT_TIM3
Programmable	19	IRQ19	0x8C	INT_TIM4
Programmable	20	IRQ20	0x90	INT_TIM5

Programmable	21	IRQ21	0x94	INT_TIM6
Programmable	22	IRQ22	0x98	INT_TIM7
Programmable	23	IRQ23	0x9C	INT_ADC
Programmable	24	IRQ24	0xA0	INT_PWM0
Programmable	25	IRQ25	0xA4	INT_PWM1
Programmable	26	IRQ26	0xA8	INT_DMA_CH0
Programmable	27	IRQ27	0xAC	INT_DMA_CH1
Programmable	28	IRQ28	0xB0	INT_DMA_CH2

Note:

1. INT_TIM0_BC: TIMER0 break、update、trigger and commutation interrupt.
2. INT_TIM0_CC: TIMER0 capture compare interrupt

9.2. Interrupt function

9.2.1. Features

- Flexible interrupt management
- Support nested interrupt
- Programmable priority

9.2.2. Interrupt enable and interrupt clear

The interrupt control register is programmable and used to control the enable and disable of interrupt requests.

For example: Enable or clear interrupt 2:

```
*( (volatile unsigned long* ) (0xE000E100 ) ) = 0x4;    // Enable interrupt 2
*( (volatile unsigned long* ) (0xE000E180 ) ) = 0x4;    //Disable interrupt 2
```

Address	Register	Description
0xE000E100	SETENA	Interrupt set enable register
0xE000E180	CLRENA	Interrupt clear enable register

9.2.2.1. Interrupt set-enable register (SETENA)

Bit	Description	RW	Reset value
31:0	Set enable interrupt 0 to 28, write 1 to set the bit to 1, writing 0 has no effect Bit[0]: Used for interrupt 0 Bit[1]: Used for interrupt 1 …… Bit[28]: Used for interrupt 28 Bit[29]~[31: Reserved	R/W	0x00000000

9.2.2.2. Interrupt clear enable register (CLRENA)

Bit	Description	RW	Reset value
31:0	Clear enable interrupt 0 to 28, write 1 to set the bit to 1, writing 0 has no effect Bit[0]: Used for interrupt 0 Bit[1]: Used for interrupt 1 ... ... Bit[28]: Used for interrupt 28 Bit[29]~[31]: Reserved	R/W	0x00000000

9.2.3. Interrupt suspension and clear suspension

If an interrupt occurs but cannot be processed immediately, the interrupt request will be suspended. Access or modify the interrupt pending status by operating the interrupt to set the pending register and the interrupt clear-pending register.

The interrupt pending status register allows software to trigger interrupts. If the interrupt has been enabled and has not been shielded, and there is no higher priority interrupt currently running, the interrupt service routine will be executed immediately.

For example: Trigger interrupt 2:

```
* ( (volatile unsigned long* ) (0xE000E100 ) ) = 0x4;     // Enable interrupt 2
* ( (volatile unsigned long* ) (0xE000E200 ) ) = 0x4;   // Pending interrupt 2
```

Clear the suspended state of interrupt 2:

```
* ( (volatile unsigned long* ) (0xE000E280 ) ) = 0x4;// Clear the suspended state of interrupt 2
```

Address	Register	Description
0xE000E200	SETPEND	Interrupt set pending register
0xE000E280	CLRPEND	Interrupt clear pending register

9.2.3.1. Interrupt set pending register(SETPEND)

Bit	Description	RW	Reset value
31:0	Set the suspended state of interrupts 0 to 28, write 1 to set the bit to 1, writing 0 has no effect Write, 1: Change interrupt pending state; 0: No effect Read, 1: Interrupt pending; 0: Interrupt not pending Bit[0]: Used for interrupt 0 Bit[1]: Used for interrupt 1 …… Bit[28]: Used for interrupt 28 Bit[29]~[31]: Reserved The read value indicates the current suspended state	R/W	0x00000000

9.2.3.2. Interrupt clear-pending register (CLRPEND)

Bit	Description	RW	Reset value
31:0	Clear the suspended state of interrupts 0 to 28,write 1 to set the bit to 1, writing 0 has no effect Bit[0]: Used for interrupt 0 Bit[1]: Used for interrupt 1 …… Bit[28]: Used for interrupt 28 Bit[29]~[31]: Reserved The read value indicates the current suspended state	R/W	0x00000000

9.2.4. Interrupt Priority

The CPU contains 8 32-bit register configurations. Each interrupt entry corresponds to an 8-bit priority register, of which only the highest two bits [7:6] are valid, and the priority levels that can be used are 0x00 (highest), 0x40, 0x80 and 0xC0 (lowest). The default priority is interrupt 0 to 28.

If two interrupts occur at the same time and their priority is the same, the interrupt with the smaller interrupt number will be executed first. The ongoing interrupt service routine can only be interrupted by high-priority interrupt requests.

Address	[31:30]	[29:24]	[23:22]	[21:16]	[15:14]	[13:8]	[7:6]	[5:0]
0xE000E41C	Reserved	-	Reserved	-	Reserved	-	IRQ28	-
0xE000E418	IRQ27	-	IRQ26	-	IRQ25	-	IRQ24	-
0xE000E414	IRQ23	-	IRQ22	-	IRQ21	-	IRQ20	-
0xE000E410	IRQ19	-	IRQ18	-	IRQ17	-	IRQ16	-
0xE000E40C	IRQ15	-	IRQ14	-	IRQ13	-	IRQ12	-
0xE000E408	IRQ11	-	IRQ10	-	IRQ9	-	IRQ8	-
0xE000E404	IRQ7	-	IRQ6	-	IRQ5	-	IRQ4	-
0xE000E400	IRQ3	-	IRQ2	-	IRQ1	-	IRQ0	-

9.3. External interruption

9.3.1. Overview

There are 16 external interrupt, INT0_0 ~ INT0_3、INT1_0 ~ INT1_3、INT2_0 ~ INT2_3、INT3_0 ~ INT3_3, Each of the four external interrupts shares an interrupt number. The interrupt mode can trigger the rising、falling edge、double edge, and the corresponding mark can be read. Each external interrupt has a separate interrupt enable, can be enabled at the same time, there is a separate interrupt flag; Each interrupt can be mapped from all IO ports, there are 60 IO port: PA0~PA15、PB0~PB15、PC0~PC15 and PD0~PD11, The interrupt source for each interrupt is selected through register EXTIx_SEL(x=0/1/2/3).

In standby mode, the edge trigger mode can wake up the chip.

The filtering clock of the external interrupt is optional, selected by register EXTI_FIL_CLK_SEL, 0: select LIRC 32K, 1: select system clock HCLK, default is LIRC 32K. Filtering time is 5clock, if select LIRC 32K, filtering time: 5*LIRC 32K(±30%)=112us~208us, if select system clock, HCLK filtering time: 5*HCLK(±3%)=101ns~107ns(HCLK=48MHz), Calculate frequency division based on system frequency.

9.3.2. Registers

Base address: 0x5000 3000

Address offset	Register	Description
0x00	EXTI_EN	EXTI interrupt enable register
0x04	EXTI_RTSR	EXTI rising edge trigger enable register
0x08	EXTI_FTSR	EXTI falling edge trigger enable register
0x0C	EXTI_RPR	EXTI rising edge trigger request flag register
0x10	EXTI_FPR	EXTI falling edge trigger request flag register
0x14	EXTI0_SEL	EXTI0 interrupt source selection register
0x18	EXTI1_SEL	EXTI1 interrupt source selection register
0x1C	EXTI2_SEL	EXTI2 interrupt source selection register
0x20	EXTI3_SEL	EXTI3 interrupt source selection register
0x24	EXTI_FIL_EN	EXTI filtering enable register
0x28	EXTI_FIL_CLK_SEL	EXTI filtering clock selection register

9.3.2.1. EXTI interrupt enable register(EXTI_EN)

Address offset: 0x00
Reset value: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTI3_3	EXTI3_2	EXTI3_1	EXTI3_0	EXTI2_3	EXTI2_2	EXTI2_1	EXTI2_0	EXTI1_3	EXTI1_2	EXTI1_1	EXTI1_0	EXTI0_3	EXTI0_2	EXTI0_1	EXTI0_0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

31:16	-	Reserved
15:0	EXTIx_y	EXTIx_y interrupt enable, x=0/1/2/3, y=0/1/2/3: 1: Enable 0: Disable

9.3.2.2. EXTI rising edge trigger enable register (EXTI_RTSR)

Address offset: 0x04
Reset value: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RT3_3	RT3_2	RT3_1	RT3_0	RT2_3	RT2_2	RT2_1	RT2_0	RT1_3	RT1_2	RT1_1	RT1_0	RT0_3	RT0_2	RT0_1	RT0_0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

31:16	-	Reserved
15:0	RTx_y	EXTIx_y rising edge trigger, x=0/1/2/3, y=0/1/2/3: 1: rising edge trigger effect 0: rising edge trigger invalid

9.3.2.3. EXTI falling edge trigger enable register(EXTI_FTSR)

Address offset: 0x08
Reset value: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FT3_3	FT3_2	FT3_1	FT3_0	FT2_3	FT2_2	FT2_1	FT2_0	FT1_3	FT1_2	FT1_1	FT1_0	FT0_3	FT0_2	FT0_1	FT0_0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

31:16	-	Reserved
15:0	FTx_y	EXTIx_y falling edge trigger enable, x=0/1/2/3, y=0/1/2/3: 1: falling edge trigger effect 0: falling edge trigger invalid

9.3.2.4. EXTI rising edge trigger request flag register (EXTI_RPR)

Address: 0x0C

Reset value: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RPIF3_3	RPIF3_2	RPIF3_1	RPIF3_0	RPIF2_3	RPIF2_2	RPIF2_1	RPIF2_0	RPIF1_3	RPIF1_2	RPIF1_1	RPIF1_0	RPIF0_3	RPIF0_2	RPIF0_1	RPIF0_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

31:16	-	Reserved
15:0	RPIFx_y	EXTIx_y rising edge trigger request, x=0/1/2/3, y=0/1/2/3: 1: The rising edge trigger request occurred 0: No rising edge trigger request occurred

9.3.2.5. EXTI falling edge trigger request flag register(EXTI_FPR)

Address: 0x10

Reset value: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FPIF3_3	FPIF3_2	FPIF3_1	FPIF3_0	FPIF2_3	FPIF2_2	FPIF2_1	FPIF2_0	FPIF1_3	FPIF1_2	FPIF1_1	FPIF1_0	FPIF0_3	FPIF0_2	FPIF0_1	FPIF0_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

31:16	-	Reserved
15:0	FPIFx_y	EXTIx_y falling edge trigger request, x=0/1/2/3, y=0/1/2/3: 1: The falling edge trigger request occurred 0: No falling edge trigger request occurred

9.3.2.6. EXTI0 interrupt source selection register (EXTI0_SEL)

Address offset: 0x14

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EXTI0_SS3								EXTI0_SS2							
RW								RW							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTI0_SS1								EXTI0_SS0							
RW								RW							

31:24	EXTI0_SS3	EXTI0_x interrupt source selection: Define EXTI0_x interrupt source selection every 8 bits, x=0/1/2/3: 00000000~00001111: PA[0]~PA[15] 00010000~00011111: PB[0]~PB[15] 00100000~00101111: PC[0]~PC[15] 00110000~00111011: PD[0]~PD[11] 00111100~ 11111111: Reserved
23:16	EXTI0_SS2	
15:8	EXTI0_SS1	
7:0	EXTI0_SS0	

9.3.2.7. EXTI1 interrupt source selection register (EXTI1_SEL)

Address offset: 0x18

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EXTI1_SS3								EXTI1_SS2							
RW								RW							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTI1_SS1								EXTI1_SS0							
RW								RW							

31:24	EXTI1_SS3	EXTI1_x interrupt source selection: Define EXTI1_x interrupt source selection every 8 bits, x=0/1/2/3: 00000000~00001111: PA[0]~PA[15] 00010000~00011111: PB[0]~PB[15] 00100000~00101111: PC[0]~PC[15] 00110000~00111011: PD[0]~PD[11] 00111100~ 11111111: Reserved
23:16	EXTI1_SS2	
15:8	EXTI1_SS1	
7:0	EXTI1_SS0	

9.3.2.8. EXTI2 interrupt source selection register (EXTI2_SEL)

Address offset: 0x1C

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EXTI2_SS3								EXTI2_SS2							

RW								RW							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTI2_SS1								EXTI2_SS0							
RW								RW							

31:24	EXTI2_SS3	EXTI2x interrupt source selection: Define EXTI2_x interrupt source selection every 8 bits, x=0/1/2/3: 00000000~00001111: PA[0]~PA[15] 00010000~00011111: PB[0]~PB[15] 00100000~00101111: PC[0]~PC[15] 00110000~00111011: PD[0]~PD[11] 00111100~ 11111111: Reserved
23:16	EXTI2_SS2	
15:8	EXTI2_SS1	
7:0	EXTI2_SS0	

9.3.2.9. EXTI3 interrupt source selection register (EXTI3_SEL)

Address offset: 0x20
Reset value: 0x0000 0000

31:24	23:16	15:8	7:0
EXTI3_SS3	EXTI3_SS2	EXTI3_SS1	EXTI3_SS0
RW	RW	RW	RW

31:24	EXTI3_SS3	EXTI3_x interrupt source selection: Define EXTI3_x interrupt source selection every 8 bits, x=0/1/2/3: 00000000~00001111: PA[0]~PA[15] 00010000~00011111: PB[0]~PB[15] 00100000~00101111: PC[0]~PC[15] 00110000~00111011: PD[0]~PD[11] 00111100~ 11111111: Reserved
23:16	EXTI3_SS2	
15:8	EXTI3_SS1	
7:0	EXTI3_SS0	

9.3.2.10. EXTI filtering enable register (EXTI_FIL_EN)

Address offset: 0x24
Reset value: 0x0000 0000

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

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

31:1	-	Reserved
0	FIL_EN	EXTI filtering enable: 1: Filter output for external interrupts 0: No filtering.(The bit is used for all interrupts) Note: Select LIRC 32k filtering time: 5*LIRC 32k(±30%)=112us~208us

		Select system clock HCLK filtering time:5*HCLK(±3%) 101ns~107ns (HCLK=48MHz), Based on system frequency division. Note: When in standby mode, if the system clock HCLK is selected, there is no filtering; If LIRC 32k clock is selected, filtering can be turned on. When in normal operating mode, selecting HCLK and LIRC 32K can turn on filtering

9.3.2.11. EXTI filtering clock selection register (EXTI_FIL_CLK_SEL)

Address offset: 0x28

Reset value: 0x0000 0002

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

15:2	1	0
Res.	FIL_CLK_RDY	FIL_CLK
	R	RW

31:2	-	Reserved
1	FIL_CLK_RDY	Clock switch completion flag: 0: Switching in progress 1: Clock switching completed
0	FIL_CLK	External interrupt filtering clock selection: 0: Select LIRC 32k 1: Select system clock HCLK Note: Only when both interrupt enable and filter enable are turned on simultaneously can there be a filtering clock. If filtering is always switched on, it is necessary to read the FIL_CLK_RDY signal as high

9.3.3. Configuration process

1. Configure trigger type.
2. Configure interrupt source selection register.
3. Configure filtering clock selection register.
4. Configure filtering enable Register.
5. Waiting FIL_CLK_RDY to pull up.
6. Configure interrupt enable register.

9.3.4. Note

1. Double edge mode requires configuration of rising edge trigger enable and falling edge trigger enable
2. Only when both interrupt enable and filter enable are turned on simultaneously can there be a filter clock. If the filter clock is switched, it is necessary to read the FIL_CLK_RDY signal as high to obtain a filter clock.
3. After configuring the filtering clock, add a 1ms delay, wait for the filtering clock to stabilize, and then enable the interrupt.

Chapter 10 Direct memory access controller(DMA)

10.1. Overview

The Direct Memory Access (DMA) controller is a master bus system peripheral used for programmable high-speed data transfer between the peripheral and memory or between memory and memory, thereby reducing the burden on the CPU. The DMA controller adopts a single AHB master bus architecture. The DMA controller has three independent channels, each used to manage requests for memory access from peripherals, and an arbiter to coordinate the priority of DMA requests.

10.2. Functional characteristics

- The configuration of the channel data structure is stored in a separate SRAM space (1K)
- Each channel has two levels of priority that can be matched, and with the same priority, the lower the channel number, the higher the priority
- Each channel has a separate interrupt enable: transmission completion interrupt, bus error interrupt
- Supports three DMA transmission data bit widths: word, half word, and byte
- The number of transfers in a single DMA cycle can be configured as 1-1024, depending on the transmission data bit width
- Source and destination address increments support bytes, half words, words, and no increments
- Supported DMA transmission types: memory to memory, memory to peripheral, peripheral to memory
- Support multiple DMA working modes:
  Invalid data structure mode, basic mode, automatic request mode, ping pong mode, memory dispersion/aggregation mode, and peripheral dispersion/aggregation mode

Notes:

1. In this chapter, N is N_ MINUS_ 1+1, configured transmission data length;
2. DMA_ DONE [C] indicates the completion of DMA transmission in channel C.

10.3. Function description

10.3.1. Arbitration

DMA transmission arbitration rate is determined by CHNLx_CONTROL register's R_power bit configuration.

When N>2^{R_power}, the controller transmits 2^{R_power} data, arbitration occurs, pausing the current channel data transmission and waiting for the next request.

10.3.2. Interrupt

The DMA module generates interrupts when either channel completes transmission or bus errors occur. Each channel has a separate interrupt enable, transmission completion interrupt flag, and error interrupt flag.

DMA transmission completion interrupt, channel interrupt enable is effective. When the channel completes the configured transmission quantity N, a transmission completion interrupt is generated, and the corresponding channel transmission completion interrupt flag bit is raised.

DMA error interrupt, channel interrupt enable is effective. When DMA accesses an undefined address space, an error interrupt is generated, and the corresponding channel transmission error interrupt flag bit is raised, causing the hardware to close the channel.

10.3.3. Data width/alignment

When channel CHNLx_CONTROL register is configured with different source data bit widths (SRC_SIZE) and destination data bit widths (DST_SIZE), and the destination data bit width is transmitted according to the source data bit width. The controller initiates AHB transmission in bytes or half words, and unused data bits on the 32-bit AHB master data bus will repeat the transmitted data. For example:

- Write Half Word 0xABCD: HWDATA=0xABCDABCD
- Write Bytes 0xAB: HWDATA=0xhABABABAB

10.3.4. Working mode

The DMA operating mode is controlled by the channel data control register CHNLx_CONTROL of CYCLE_CTRL bit configuration. DMA working modes include invalid data structure mode, basic mode, Ping-Pong mode, automatic request mode, memory dispersed aggregation mode, and peripheral dispersed aggregation mode.

Working mode	Mode characteristics	Transmission data length (words/half words/bytes)	Transmission type
Invalid data structure	Invalid data structure, no data transmission;	-	-
Basis	Enable hardware to close the channel after a transmission is completed;	Max=1024	Transfer between peripherals and storage: After arbitration occurs, the peripheral needs to send a request signal again to continue transmission.
Ping-Pong	Alternate the use of primary and backup data structures until an invalid data structure is read or the channel is enabled by software programming; After generating the transmission completion interrupt, it is necessary to reconfigure the channel data structure configuration register.	Can be greater than 1024	
Peripheral dispersed aggregation	The configuration value of the channel backup data structure register is stored in the system SRAM; After each DMA (N data) transmission is completed, the controller automatically configures backup data structure registers through the main data structure transmission.	Can be greater than 1024	
Automatic request	After a single transmission (N data) is completed, the hardware closes the channel to enable it.	Max=1024	Memory to memory transfer: A request signal can complete a DMA transmission, and the controller automatically generates a request signal after arbitration occurs.
Memory dispersed aggregation	The configuration value of the channel backup data structure register is stored in the system SRAM; After each DMA (N data) transmission is completed, the controller automatically configures backup data structure registers through the main data structure transmission.	Can be greater than 1024	

10.3.4.1. Invalid data structure pattern

Invalid data structure, this mode does not perform data transmission; When responding to a DMA transmission request and the corresponding channel DMA working mode is invalid data structure, the hardware turns off the channel to enable it.

10.3.4.2. Basic mode

The transfer mode between peripherals and memory, with a maximum data length of 1024 (in words/half words/bytes) for a DMA transfer.

In this operating mode, after the controller receives a DMA transmission request (with the highest priority):

1. Controller executes 2^{R_power} or N (based on the smaller value) DMA data transmission;
2. Small N: The controller performs N data transfers, and after the transfer is completed, the controller generates a transmission completion interrupt and sets the channel transmission completion interrupt flag CHNL_ DONE_ ISR set to 1;
3. Small 2^{R_power}: DMA controller pause transmission after executes 2^{R_power} data transmission;
   a) If a higher priority channel is requesting a response, the controller responds to that channel;
   b) The peripheral device or software of this channel sends a request to the controller.

      The controller responds to the request and continues the previous transmission, repeating step 3 until N data transfers are completed.

Once DMA N data transfers are completed, a DMA transfer completion interrupt is generated, the hardware closes the channel to enable it, and the hardware clears the channel CHNLx_CONTROL register's N_MINUS_ 1 and Cycle_ CTRL bit.

10.3.4.3. Ping-Pong mode

When the transmission data length between the peripheral and the memory is greater than 1024 (unit: word/half word/byte optional), this mode can be selected. In this working mode, the DMA controller alternates between the main data structure and the backup data structure. After completing N DMA data transfers at a time, a transfer completion interrupt (channel interrupt enabled) is generated, and the channel data structure configuration register can be reprogrammed by software; The next DMA transmission hardware will automatically switch to another data structure. After each DMA (N data) transmission is completed, the hardware will not disable the channel until an invalid data structure is read (hardware disabled channel enabled) or software programming disabled channel enabled.

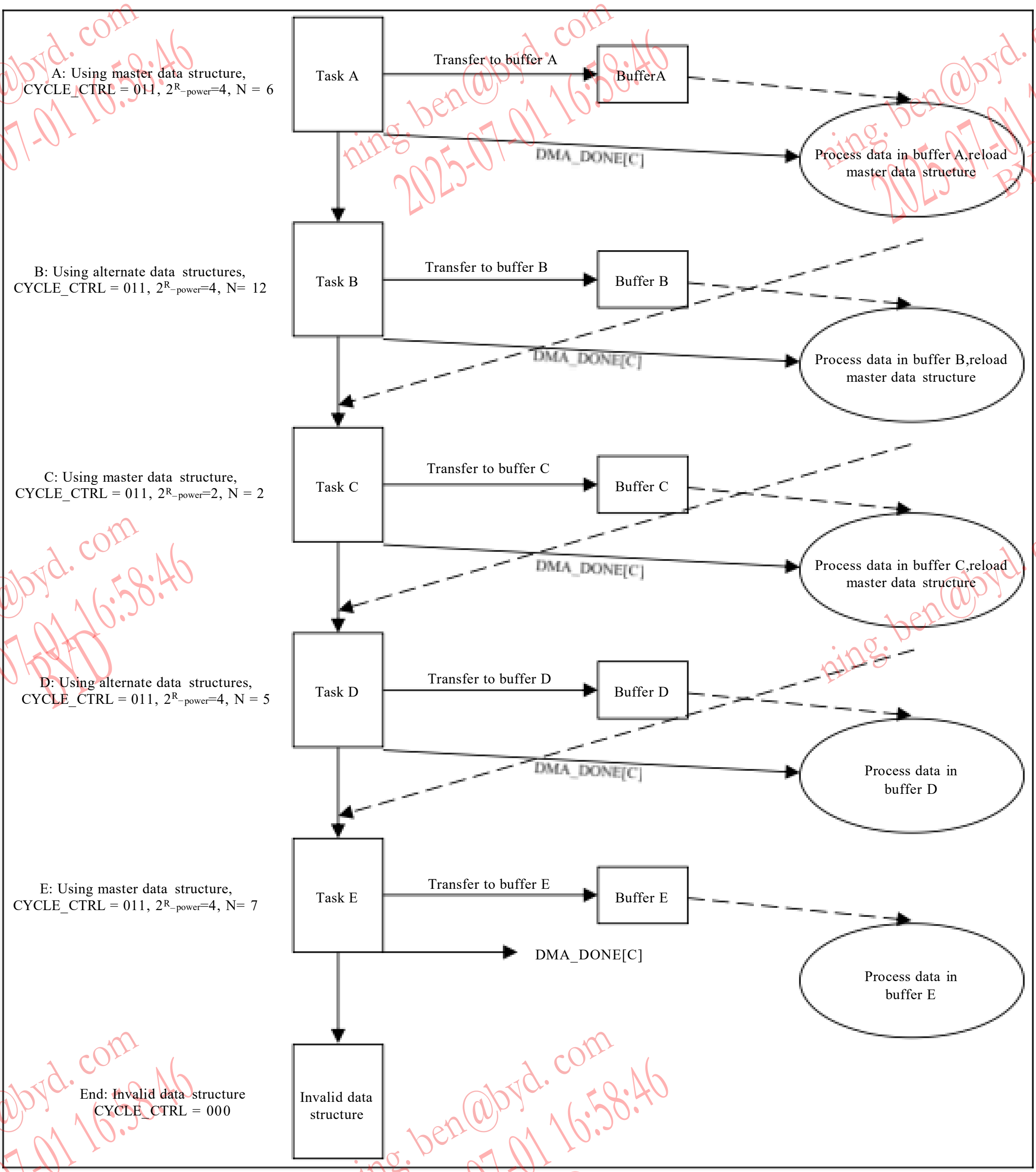

A: Using master data structure,
CYCLE_CTRL = 011, $2^{R}{}_{-power}=4$, N = 6
Task A
Transfer to buffer A
BufferA
DMA_DONE[C]
Process data in buffer A,reload master data structure
B: Using alternate data structures,
CYCLE_CTRL = 011, $2^{R}{}_{-power}=4$, N= 12
Task B
Transfer to buffer B
Buffer B
DMA_DONE[C]
Process data in buffer B,reload master data structure
C: Using master data structure,
CYCLE_CTRL = 011, $2^{R}{}_{-power}=2$, N = 2
Task C
Transfer to buffer C
Buffer C
DMA_DONE[C]
Process data in buffer C,reload master data structure
D: Using alternate data structures,
CYCLE_CTRL = 011, $2^{R}{}_{-power}=4$, N = 5
Task D
Transfer to buffer D
Buffer D
DMA_DONE[C]
Process data in buffer D
E: Using master data structure,
CYCLE_CTRL = 011, $2^{R}{}_{-power}=4$, N= 7
Task E
Transfer to buffer E
Buffer E
DMA_DONE[C]
Process data in buffer E
End: Invalid data structure
CYCLE_CTRL = 000
Invalid data structure

Figure 10.1 Ping-Pong mode example

Initialization:

According to Task A, Configure the CYCLE_CTRL=011 (Ping-Pong mode) of the channel data control register (CHNLx_CONTROL)for the master data structure of channel C, R_power = 0010, N_minus_1 = 0000000101; Other registers are configured according to specific requirements.

According to Task B, Configure the CYCLE_CTRL=011 (Ping-Pong mode) of the channel data control register (CHNLx_CONTROL)for the backup data structure of channel C, R_power = 0010, N_minus_1 = 0000010001; Other registers are configured according to specific requirements.

Configure CHNL_ PRI_ ALT_ CLR, configure channel C to select the master data structure.

When the controller receives a DMA request, if the channel has the highest priority, the process continues:

Task A (Channel C uses the master data structure):

1. Controller executes 2^{R_power} = 4 DMA data transfers;
2. Controller arbitration: When the controller receives a DMA request again, if the channel has the highest priority, the process continues.
3. The controller performs the remaining 2 DMA data transfers;
4. Controller setting DMA_ DONE [C] lasts for one HCLK cycle and enters an alternating process.

Note: After Task A is completed, the processor hardware is set to CHNL_ PRI_ ALT_ SET selects the backup data structure, and software programming corresponds to the channel data structure configuration register of the channel main data structure, which is used to control Task C.

After the controller receives a new request for this channel (with the highest priority), Task B begins:

Task B (Channel C uses backup data structures):

1. The controller performs 4 DMA data transfers;
2. Controller arbitration: After receiving a request from this channel, if the channel has the highest priority, the process continues;
3. The controller receives a request and performs four DMA data transfers;
4. Controller arbitration: After receiving a request from this channel, if the channel has the highest priority, the process continues;
5. The controller performs the remaining 4 DMA data transfers;
6. Controller setting DMA_ DONE [C] continues for one HCLK cycle and enters an alternating process.

Note: After Task B is completed, the processor hardware is set to CHNL_PRI_ALT_CLR selects the main data structure, and software programming corresponds to the channel data structure configuration register of the channel backup data structure, which is used to control Task D.

After the controller receives a new request for this channel (with the highest priority), Task C begins:

Task C (Channel C uses the master data structure):

1. The controller performs 2 DMA data transfers.
2. Controller setting DMA_ DONE [C] continues for one HCLK cycle and enters an

alternating process.

Note: After Task C is completed, the processor hardware is set to CHNL_ PRI_ ALT_ SET selects the backup data structure, and software programming corresponds to the channel data structure configuration register of the channel main data structure, which is used to control task E.

After the controller receives a new request for this channel (with the highest priority), Task D begins:

Task D (Channel C uses backup data structures):

1. The controller performs 4 DMA data transfers;
2. Controller arbitration: After receiving a request from this channel, if the channel has the highest priority, the process continues;
3. The controller performs the remaining 1 DMA data transmission;
4. Controller setting DMA_ DONE [C] continues for one HCLK cycle and enters an alternating process.

After the controller receives a new request for this channel (with the highest priority), then Task E begins:

Task E (Channel C uses the master data structure):

1. The controller performs 4 DMA data transfers;
2. Controller arbitration: After receiving a request from this channel, if the channel has the highest priority, the process continues;
3. The controller performs the remaining 3 DMA data transfers;
4. Controller setting DMA_ DONE [C] continues for one HCLK cycle and enters an alternating process.

After receiving a new request from this channel (with the highest priority), start the next task:

End

Because the processor did not reconfigure the channel backup data structure register, after the Task D transfer was completed, the controller configured the channel backup data structure (CHNLx_CONTROL) register CYLLE_CTRL is 000B (invalid data structure), so data transmission will not be carried out. The hardware will disable the corresponding channel to enable it, and at this time, the channel will be the main data structure.

Note: If after Task C is completed, Task E Channel Data Structure CHNLx_CONTROL register CYCLE_CTRL configured as 001, Task E is a basic DMA cycle type, then after the transfer of Task E is completed, set DMA_ DONE [C] is high and lasts for one HCLK cycle, and the corresponding channel is enabled by hardware shutdown. At this time, the channel is a backup data structure.

10.3.4.4. Peripheral dispersion/aggregation mode

When the data transfer between the peripheral and the memory is greater than 1024 (unit: word/half word/byte optional), this mode can be selected for use. In this operating mode, the DMA controller uses the backup data structure of the channel for data transfer between peripherals and memory, and the configuration values of the backup data structure register of the channel are stored in the system SRAM (The CHNLx_SRC_END_PTR register of the master data structure points to this address), The controller configures backup data structure registers for the same channel through the main data structure of the channel.

Notes:

1. Peripheral dispersion/aggregation mode, where the channel performs DMA transmission through the main data structure to configure the backup data structure register of the channel. Therefore, the configuration of the main data structure register must comply with the following requirements:
   a) CHNLx_ Control register R_power configuration is 0010, and N is a multiple of 4;
   b) CHNLx_ DST_ END_PTR register lower 4 bits are configured as 0xC (the address of the backup data control register pointing to channel x);
2. Backup data control register address for channel x = 0x5008_0200+{Channel x*0x10+0x0C}(x*0x10+0x0C is a reserved address);
3. The main data structure performs DMA transmission, written in the order of 0x0, 0x4, 0x8, and 0xC;
4. The last backup data structure must be configured as the basic mode. After the entire DMA transmission is completed, the controller generates a DMA transmission completion interrupt.
5. For the peripheral dispersed aggregation DMA cycle type, after the main data structure transmission is completed, the hardware switches to select the backup data structure. After all DMA transfers specified by the backup data structure are completed, the hardware switches to select the main data structure again.
6. If you want to start the transmission again, you need to first configure CHNL_PRI_ALT_CLR corresponds bit to 1 and switches back to the main data structure.

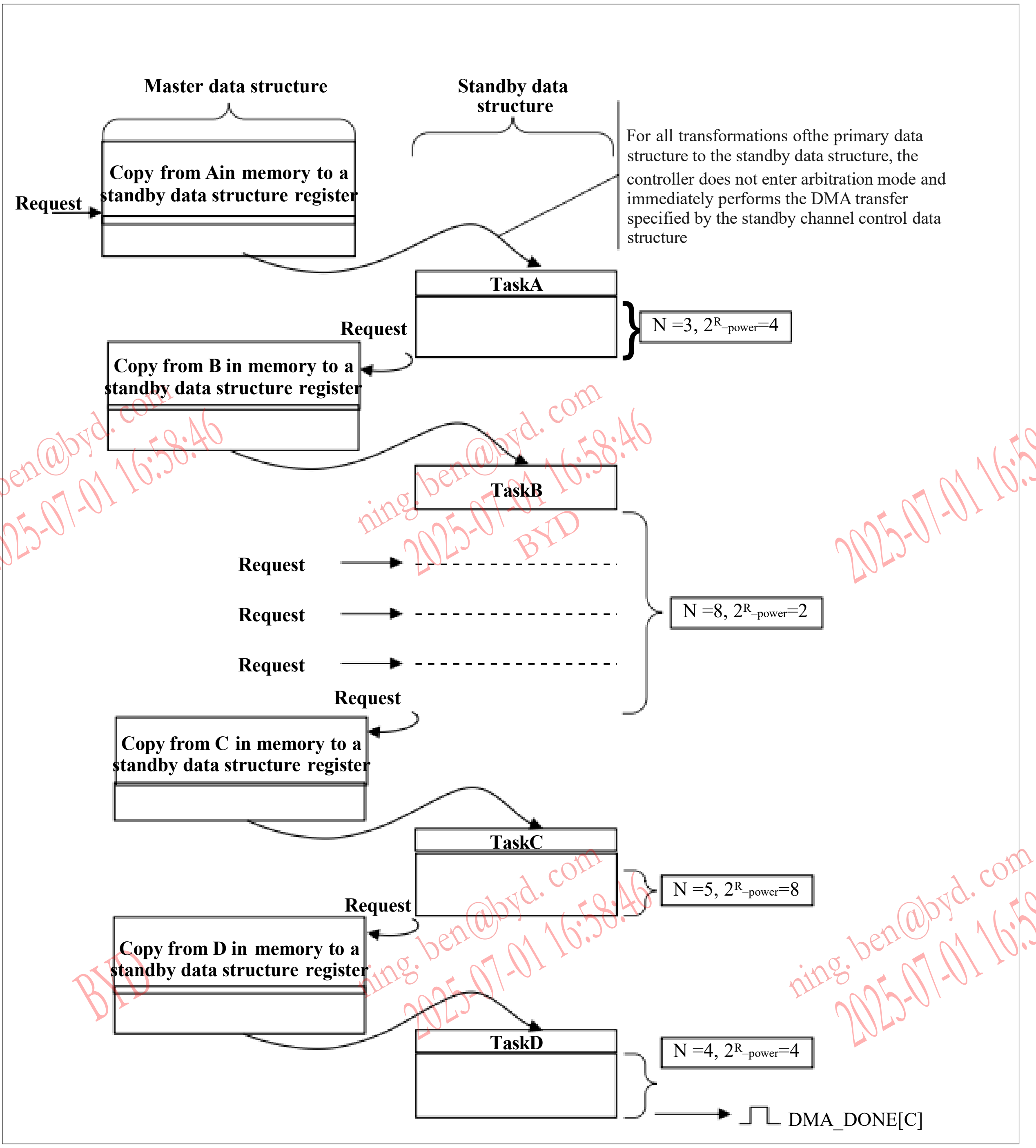

Master data structure
Standby data structure
For all transformations ofthe primary data structure to the standby data structure, the controller does not enter arbitration mode and immediately performs the DMA transfer specified by the standby channel control data structure
Request
Copy from Ain memory to a standby data structure register
TaskA
N =3, 2R_power=4
Request
Copy from B in memory to a standby data structure register
TaskB
Request
Request
Request
N =8, 2R_power=2
Request
Copy from C in memory to a standby data structure register
TaskC
N =5, 2R_power=8
Request
Copy from D in memory to a standby data structure register
TaskD
N =4, 2R_power=4
DMA_DONE[C]

Figure 10.2 Peripheral dispersion/aggregation mode

Initialize

The host processor configures the channel master data structure:

1. Configure CHNLx_CONTROL register CYCLE_CTRL= 110(Peripheral dispersion/Aggregation model),the minimum R_power configuration is 0010.In this case,There are four tasks, so set N_MINUS_1 to 0000001111,Source data bit width and target data bit width SRC_SIZE＝10,DST_SIZE＝10(word);
2. Configure the CHNLx_SRC_END_PTR register as the end address of the memory space

to store the configured value of the standby data structure register;

3. Configure the CHNLx_DST_END_PTR register as the end address of the secondary channel structure register:
4. Alternate channel structure register end address = 0x5008_0200+{Channel number x*0x10+0x0C}
5. The processor writes the data structure configuration value used by tasks A, B, C, and D to the storage address space configured in the CHNLx_SRC_END_PTR register of the main data structure of the channel.
6. DMA and corresponding channels are enabled on the processor.

When the controller receives a DMA request, the peripheral distributed/aggregated transmission begins. The transmission process is as follows:

Primary,copy A

The controller performs four DMA data transfers that write backup data structure registers to Task A.

No arbitration occurs and the controller continues to execute Task A.

Task A,Standby data structure

The controller performs 3 DMA data transfers to complete the transfer.

After the peripheral issues a new request (highest priority), the transfer process continues:

Primary,copy B

The controller performs four DMA transfers that write backup data structure registers to Task B.

No arbitration occurs and the controller continues to execute Task B.

Task B,Standby data structure

1. The controller performs 2 DMA transfers and then arbitrates.
2. When the controller receives a DMA request (highest priority), the controller performs 2 DMA transfers and then arbitrates.
3. When the controller receives a DMA request (highest priority), the controller performs 2 DMA transfers and then arbitrates.
4. When the controller receives a DMA request (highest priority), the controller performs the remaining 2 DMA transfers to complete the transfer.

Notes: In order for the controller to complete this Task, the peripheral must issue three requests.

After the peripheral issues a new request (highest priority), the transfer process continues:

Primary,copy C

The controller performs 4 DMA transfers that write backup data structure registers to Task C.

No arbitration occurs and the controller continues to execute Task C

Task C,Standby data structure

The controller performs 5 DMA transfers to complete the transfer.

After the peripheral issues a new request (highest priority), the transfer process continues:

Primary,copy D

The controller performs four DMA transfers that write alternate data structures to Task D.

No arbitration occurs and the controller continues to execute task D:

Task D,Standby data structure

The controller executes Task D using a basic mode DMA cycle type: The controller performs four DMA transfers, sets DMA_Done to high duration for one HCLK cycle, and turns off channel enable.

10.3.4.5. Automatic Request Mode

Used for transfer mode between memory. The maximum data length for DMA transmission is 1024(Unit: word/half word/byte Optional),Configure the software to trigger transfer requests to start DMA transfers. N data transfers are completed at one DMA,generates a DMA transfer completion interrupt,the hardware close channel was enabled,hardware clear channel The N_MINUS_ 1 and CYCLE_CTRL bits ofthe CHNLx_CONTROL register.

10.3.4.6. Memory Dispersion/Aggregation Mode Mode

It can be used for transmission or block transmission of data length greater than 1024 (unit: word/half word/byte optional) between memory,configure software requests to initiate DMA transfers. In this mode of operation,DMA controllers use channel standby data structures for data transfer between memory. The channel alternate data structure register configuration values are stored in the system SRAM (the primary data structure's CHNLx_SRC_END_PTR register points to this address).After each DMA (N data) transfer,The controller automatically configures alternate data structure registers ofthe same channel through the primary data structure transmission of the channel.

Notes:

1. The channel performs DMA transfers through the primary data structure to configure the standby data structure register. Therefore, the master data structure register configuration must comply with the following requirements:
   1) The CHNLx_CONTROL register R_power is set to 0010 and N is a multiple of 4;
   2) The low 4 bits ofthe CHNLx_DST_END_PTR register are configured to 0xC (to the alternate control data structure register address of the channel);
2. Alternate channel structure register end address = 0x5008_0200+{Channel number x*0x10+0x0C} (x*0x10+0x0C is reserved address);
3. The master data structure performs DMA transfers, written in the order 0x0, 0x4, 0x8, 0xC;
4. The last time the standby data structure must be configured for automatic request operation mode, the controller generates a DMA transfer completion interrupt after the entire DMA

transfer is complete. If 2^R_power=N, the transmission can be correctly configured in the basic mode, and the controller generates a DMA transmission completion interrupt after the transmission is completed.

5. For the memory distributed aggregation DMA cycle type, the primary data structure is transferred, the hardware switches to select the alternate data structure, and after all DMA transfers specified by the alternate data structure are completed, the hardware switches again to select the primary data structure.

If you want to start the transfer again, you need to configure the CHNL_PRI_ALT_CLR bit to 1 and switch back to the main data structure.

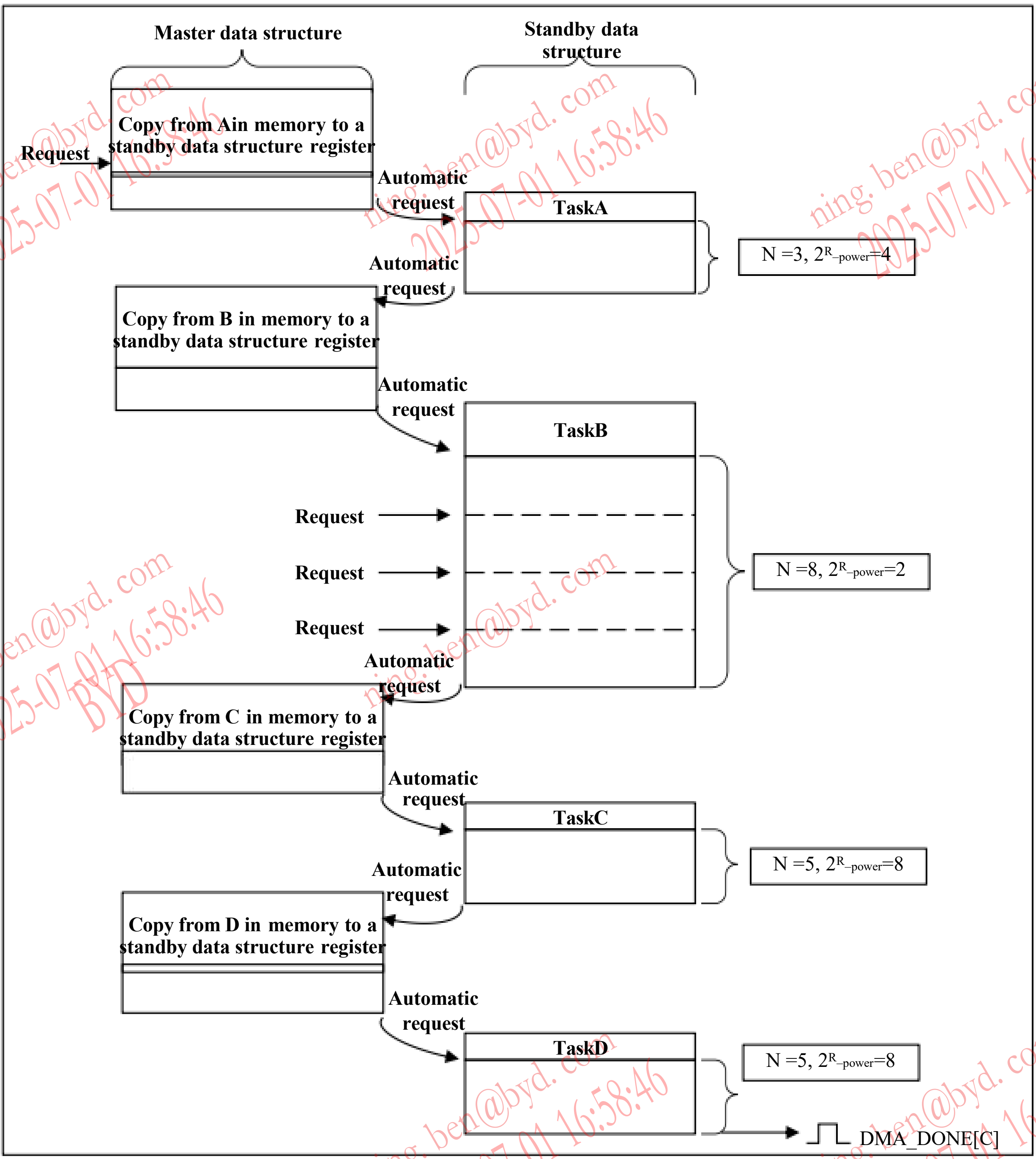


Figure 10.3 Memory dispersion/aggregation mode

Initialize:

1. Host processor configures channel master data structure:
2. Configure the CHNLx_CONTROL register CYCLE_CTRL=100 (memory dispersion/aggregation mode),and R_power is set to 0010. In this example, there are 4 tasks so N_minus_ 1 is configured to 0x00F, source and target data bit width size=10, inc=10;
3. Configure the CHNLx_SRC_END_PTR register as the end address of the memory space to store the configured value of the standby data structure register;
4. Configure the CHNLx_DST_END_PTR register as the end address of the secondary channel structure register:
5. Alternate channel structure register end address = alternate data control register base address pointer+{Channel number x*0x10+0x0C}
6. The host processor writes the data structure configuration value used by tasks A, B, C, and D to the storage address space configured in the CHNLx_SRC_END_PTR register of the main data structure of the channel.
7. DMA and corresponding channels are enabled on the host processor

When the controller receives a DMA request (highest priority), the peripheral decentralized/aggregated transfer begins, and the transfer process is as follows:

Primary,copy A

1. The controller performs four DMA transfers that write backup data structure registers to Task A.
2. The controller generates automatic requests for channels and then arbitrates.

If the current channel has the highest priority, the transmission process continues:

Task A,Standby data structure

1. The controller performs 3 DMA transfers, the controller generates channel automatic requests, and then arbitrates.

If the current channel has the highest priority, the transmission process continues;

Primary,copy B

1. The controller performs four DMA transfers that write backup data structure registers to Task B.
2. The controller generates automatic requests for channels and then arbitrates.

If the current channel has the highest priority, the transmission process continues:

Task B,Standby data structure

1. The controller performs 2 DMA transfers, generating channel automatic requests, which are then arbitrated.
2. If the current channel has the highest priority, the controller continues to perform 2 DMA transfers, producing an automatic request for the channel, and then arbitrates.
3. If the current channel has the highest priority, the controller continues to perform 2 DMA

transfers, producing an automatic request for the channel, and then arbitrates.

4. If the current channel has the highest priority, the controller continues to perform the remaining 2 DMA transfers, completing the transfer and generating the channel automatic request.

If the current channel has the highest priority, the transmission process continues:

Primary,copy C

1. The controller performs four DMA transfers that write backup data structure registers to task C.
2. The controller generates automatic requests for channels and then arbitrates.

If the current channel has the highest priority, the transmission process continues:

Task C,Standby data structure

1. The controller performs 5 DMA transfers, completes the transfer, and the controller generates a channel automatic request.

If the current channel has the highest priority, the transmission process continues:

Primary(Master),copy D

1. The controller performs four DMA transfers that write backup data structure registers to Task D.
2. The controller generates automatic requests for channels and then arbitrates.

If the current channel has the highest priority, the transmission process continues:

Task D,Standby data structure

The controller uses a basic period type to execute Task D: The controller performs four DMA transfers. After the transfers are complete, set DMA_DONE[C] to high duration for one HCLK cycle and disable the channel.

10.3.5. DMAMUX request source

DMA channel number	DMA request source number	Feature
Channel 0~2	0	Fixed low level
	1	ADC
	2	SPI0_RX
	3	SPI0_TX
	4	SPI1_RX
	5	SPI1_TX
	6	IIC_RX
	7	IIC_TX
	8	UART0_RX
	9	UART0_TX
	10	UART1_RX
	11	UART1_TX
	12	UART2_RX
	13	UART2_TX
	14	UART3_RX
	15	UART3_TX
	16	UART4_RX
	17	UART4_TX
	18	TIM0_CH1_CC
	19	TIM0_CH2_CC
	20	TIM0_CH3_CC
	21	TIM0_CH4_CC
	22	TIM0_TRIG_COM
	23	TIM0_UP
	24	TIM1_CH1_CC
	25	TIM1_UP
	26	TIM2_CH1_CC
	27	TIM2_UP
	28	TIM3_CH1_CC
	29	TIM3_UP
	30	TIM4_CH1_CC
	31	TIM4_UP
	32	TIM5_CH1_CC
	33	TIM5_UP
	34	TIM6_CH1_CC
	35	TIM6_CH2_CC
	36	TIM6_CH3_CC
	37	TIM6_CH4_CC
	38	TIM6_TRIG_COM
	39	TIM6_UP
	40	TIM7_CH1_CC
	41	TIM7_UP

10.4. DMA register

Base address: 0x5000 0000

Address offset	Register	Description
0x04	RCU_EN	Peripheral module clock control register

Channel data structure configuration register stored in DMA_SRAM (0x5008 0000-0x5008 03FF)
Primary data structure configures register base address: 0x5008 0000
Alternate data structure configuration register base address: 0x5008 0200   x = 0/1/2,x is the channel number

Address offset	Register	Description
0x00+0x10*x	CHNLx_SRC_END_PTR	Channel x Source data end address
0x04+0x10*x	CHNLx_DST_END_PTR	Channel x Destination data end address
0x08+0x10*x	CHNLx_CONTROL	Channel x channel data control register

DMA_SFR Register
Base address: 0x5008 0400

Address offset	Register	Description
0x00	DMA_STATUS	DMA status register
0x04	DMA_CFG	DMA configuration register
0x08	CTRL_BASE_PTR	Channel master data structure base address register
0x10	DMA_WAITONREQ_STATUS	Channel wait request status register
0x14	CHNL_SW_REQUEST	Channel software request register
0x20	CHNL_REQ_MASK_SET	Channel request mask setting register
0x24	CHNL_REQ_MASK_CLR	Channel request mask clear register
0x28	CHNL_EN_SET	Channel enable set register
0x2C	CHNL_EN_CLR	Channel enables clear register
0x30	CHNL_PRI_ALT_SET	Channel primary standby data structure setting register
0x34	CHNL_PRI_ALT_CLR	Channel master standby data structure clear register
0x38	CHNL_PRI_SET	Channel priority setting register
0x3C	CHNL_PRI_CLR	Channel priority clear register
0x50	CHNL_INT_EN	DMA interrupt enable register
0x54	CHNL_DONE_ISR	Channel transfer completion flag register
0x58	CHNL_DONE_IFCR	Channel transfer completion flag clear register
0x5C	CHNL_ERR_ISR	Channel transfer error flag register
0x60	CHNL_ERR_IFCR	Channel transfer error flag Clear register

DMAMUX Register
Base address: 0x5008 0800

Address offset	Register	Description
0x00	DMAMUX_SEL00	DMA channel 0 mapping request source select register
0x04	DMAMUX_SEL01	DMA channel 1 mapping request source select register
0x08	DMAMUX_SEL02	DMA channel 2 mapping request source select register

10.4.1. Peripheral module clock control register(RCU_EN)

Address offset: 0x04

Reset value: 0x0000 0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.				DIV_ CLKEN	GPIOD_ CLKEN	GPIOC_ CLKEN	GPIOB_ CLKEN	GPIOA_ CLKEN	DMA_ CLKEN	CRC_ CLKEN	ADC_ CLKEN	WDT_ CLKEN	TIM7_ CLKEN	TIM6_ CLKEN	TIM5_ CLKEN
				RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

22	DMA_CLKEN	The DMA module (including DMA_SRAM/DMAMUX) was enabled: 1: Work 0: Off. The default is 0

10.4.2. Channel x source data end address register(CHNLx_SRC_END_PTR)

Address offset: 0x00+0x10*x(x = 0/1/2,x is the channel number)

Reset value: 0x0000 0000

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

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

31: 0	SRC_END_PTR	End address of the source data

The controller must configure the source data end address before performing a DMA transfer; The source data start address depends on the source data bit width (SRC_SIZE) of the control data configuration register (CHNLx_CONTROL), the source data address increment (SRC_INC) and the total number of DMA transfers (N_MINUS_1).

Source data start address = Source data end address - INC * Total number of DMA transfers (N_MINUS_1);

When the source data address increment is no increment, byte, half word, word, INC corresponds to 0, 1,2,4;

10.4.3. Channel x destination data end address register(CHNLx_DST_END_PTR)

Address offset: 0x04+0x10*x(x = 0/1/2,x is the channel number)

Reset value: 0x0000 0000

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

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

31: 0	DST_END_PTR	Destination end address of data

The controller must configure the destination data end address before performing a DMA transfer;

The destination data start address depends on the source data bit width (DST_SIZE) of the control data configuration register (CHNLx_CONTROL), the destination data address increment (DST_INC) and the total number of DMA transfers (N_MINUS_1).

Destination data start address = Destination data end address - INC * N_MINUS_1;

When the destination data address increment is set to no increment, byte, half word, word, INC corresponds to 0, 1,2,4;

10.4.4. Channel x channel data control register(CHNLx_CONTROL)

Address offset: 0x08+0x10*x(x = 0/1/2,x is the channel number)

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18
DST_INC		DST_SIZE		SRC_INC		SRC_SIZE		Res.					
RW		RW		RW		RW							

17: 14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R_power	N_MINUS_ 1										Res.	CYCLE_CTRL		
RW	RW											RW		

31: 30	DST_INC	Destination address increment: The destination data bit width is byte 00: Byte 01: Half-word 10: Word 11: No increment The destination data bit width is half - word 00/01: Half-word 10: Word 11: No increment The destination data bit width is word 00/01/10: Word 11: No increment
29: 28	DST_SIZE	Destination data bit width: 00: Byte 01: Half-word 10/11: Word Notes: DST_SIZE and SRC_SIZE must be set to the same value
27: 26	SRC_INC	Source address increment:

		00: Byte 01: Half-word 10: Word 11: No increment The destination data bit width is half - word 00/01: Half-word 10: Word 11: No increment The destination data bit width is word 00/01/10: Word 11: No increment
25: 24	SRC_SIZE	Bit width of the source data: 00: Byte 01: Half-word 10/11: Word
23: 18	-	Reserved
17: 14	R_power	Arbitration rate setting: 0000: 1 piece of data is transferred per request 0001: 2 pieces of data is transferred per request 0010: 4 pieces of data is transferred per request 0011: 8 pieces of data is transferred per request 0100: 16 pieces of data is transferred per request 0101: 32 pieces of data is transferred per request 0110: 64 pieces of data is transferred per request 0111: 128 pieces of data is transferred per request 1000: 256 pieces of data is transferred per request 1001: 512 pieces of data is transferred per request 1010~ 1111: 1024 pieces of data is transferred per request For ADC/UART/IIC/SPI modules, R_power must be configured to 0
13: 4	N_MINUS_ 1	Each DMA transfer length is N_minus_ 1+1: 0000000000: The length of the transmitted data is 1 0000000001: The length of the transmitted data is 2 0000000010: The length of the transmitted data is 3 0000000011: The length of the transmitted data is 4 0000000100: The length of the transmitted data is 5 …… 1111111111: The length of the transmitted data is 1024 The arbitration process takes precedence over the controller. In the DMA module, the amount of data transmitted by the channel is represented by N: N= N_MINUS_ 1+1
3	-	Reserved
2: 0	CYCLE_CTRL	DMA working mode selection: 000: Invalid data structure 001: Basics 010: Automatic request: During the arbitration process, the controller automatically inserts a request

		011: Ping-pong: The controller uses one data structure to execute a DMA cycle, and after the DMA cycle is completed, the next DMA cycle is executed using the other data structure. 100: Memory dispersion/aggregation (using master data structure) 101: Memory dispersion/aggregation (using alternate data structures) The two cycle types are used together, for details see the working mode of DMA. 110: Peripheral dispersion/aggregation (using master data structure) 111: Peripheral dispersion/aggregation (using alternate data structure) The two cycle types are used together, for details see the working mode of DMA

10.4.5. DMA status register(DMA_STATUS)

Address offset: 0x00

Reset value: 0x0000 0000

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

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

31: 1	-	Reserved
0	STATE	DMA controller status: 0: DMA controller is disabled 1: DMA controller is enabled

10.4.6. DMA configuration register(DMA_CFG)

Address offset: 0x04

Reset value: 0x0000 0000

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

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

31: 1	-	Reserved

0	DMA_EN	DMA controller enabled: 0: Disable DMA controller 1: Enable DMA controller

10.4.7. DMA channel master data structure base address register (CTRL_BASE_PTR)

Address offset: 0x08

Reset value: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CTRL_BASE_PTR[15: 7]									CTRL_BASE_PTR[6: 0]						
RW									R						

31: 7	CTRL_BASE_PTR	Primary data structure configuration register base address (configured to point to DMA_SRAM): Notes: Set the register to 0x5008_0000
6: 0		

10.4.8. DMA channel wait request status register (DMA_WAITONREQ_STATUS)

Address offset: 0x10

Reset value: 0x00000000

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

31: 3	2	1	0
Res.	WAITONREQ_S2	WAITONREQ_S1	WAITONREQ_S0
	R	R	R

31: 3	-	Reserved
2	WAITONREQ_Sx	DMA channel x waiting for request status: 0: The channel x waiting request signal is low 1: The channel x waiting request signal is high
1		
0		

10.4.9. DMA channel software request register (CHNL_SW_REQUEST)

Address offset: 0x14

Reset value: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.													SW_REQ2	SW_REQ1	SW_REQ0
													W	W	W

31: 3	-	Reserved
2	SW_REQx	Channel x software request signal: Write 1 generates a software DMA request corresponding to the DMA channel When a DMA channel is disabled, writing the corresponding bit does not generate a DMA request for that channel
1		
0		

10.4.10. DMA channel request mask set register (CHNL_REQ_MASK_SET)

Address offset: 0x20

Reset value: 0x0000 0000

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

15: 3	2	1	0
Res.	REQ_MASK_SET2	REQ_MASK_SET1	REQ_MASK_SET0
	RS	RS	RS

31: 3	-	Reserved
2	REQ_MASK_SETx	Mask status of DMA channel x request signal: Read: 0: Enable peripheral hardware requests for channel x 1: Disable peripheral hardware requests for channel x Write: 0: Invalid 1: Disable peripheral hardware requests for channel x
1		
0		

10.4.11. DMA channel request mask clear register (CHNL_REQ_MASK_CLR)

Address offset: 0x24

Reset value: 0x0000 0000

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

15: 3	2	1	0
Res.	REQ_MASK_CLR2	REQ_MASK_CLR1	REQ_MASK_CLR0
	RS	RS	RS

31: 3	-	Reserved
2	REQ_MASK_CLRx	The masking state of the DMA request signal: Read: 0: Enable peripheral hardware requests for channel x 1: Disable peripheral hardware requests for channel x Write: 0: Invalid 1: Enable peripheral hardware requests for channel x
1		
0		

10.4.12. DMA channel enable set register (CHNL_EN_SET)

Address offset: 0x28

Reset value: 0x0000 0000

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

15: 3	2	1	0
Res.	EN_SET2	EN_SET1	EN_SET0
	RS	RS	RS

31: 3	-	Reserved
2	EN_SETx	Enable status of channel x: Read: 0: Channel x is disabled 1: Channel x is enabled Write: 0: Invalid 1: Enable channel x Notes: When the channel transfer is complete, the hardware deactivates the channel enable by reading an invalid data structure or a channel error.
1		
0		

10.4.13. DMA channel enable clear registers (CHNL_EN_CLR)

Address offset: 0x2C

Reset value: 0x0000 0000

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

15: 3	2	1	0
Res.	EN_CLR2	EN_CLR1	EN_CLR0
	RS	RS	RS

31: 3	-	Reserved
2	EN_CLRx	Enable status of channel x: Read: 0: Channel x is disabled 1: Channel x is enabled Write: 0: Invalid 1: Channel x is disabled
1		
0		

10.4.14. DMA channel master standby setup register (CHNL_PRI_ALT_SET)

Address offset: 0x30

Reset value: 0x0000 0000

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

15: 3	2	1	0
Res.	PRI_ALT_SET2	PRI_ALT_SET1	PRI_ALT_SET0
	RS	RS	RS

31: 3	-	Reserved
2	PRI_ALT_SETx	Channel x controls the data structure state: Read: 0: Channel x uses the master data structure 1: Channel x uses the alternate data structure Write: 0: Invalid 1: Channel x selects the alternate data structure
1		
0		

10.4.15. DMA channel master and standby clear register (CHNL_PRI_ALT_CLR)

Address offset: 0x34

Reset value: 0x0000 0000

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

15: 3	2	1	0
Res.	PRI_ALT_CLR2	PRI_ALT_CLR1	PRI_ALT_CLR0
	RS	RS	RS

31: 3	-	Reserved
2	PRI_ALT_CLRx	Channel x controls the data structure state: Read: 0: Channel x uses the master data structure 1: Channel x uses the alternate data structure Write: 0: Invalid 1: Channel x selects the master data structure
1		
0		

10.4.16. DMA channel priority setting register (CHNL_PRI_SET)

Address offset: 0x38

Reset value: 0x0000 0000

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

15: 3	2	1	0
Res.	PRI_SET2	PRI_SET1	PRI_SET0
	RS	RS	RS

31: 3	-	Reserved
2	PRI_SETx	Channel x priority status: 0: Invalid 1: Channel x uses a high priority
1		
0		

10.4.17. DMA channel priority clear register(CHNL_PRI_CLR)

Address offset: 0x3C

Reset value: 0x0000 0000

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

15: 3	2	1	0
Res.	PRI_CLR2	PRI_CLR1	PRI_CLR0
	RS	RS	RS

31: 3	-	Reserved
2	PRI_CLRx	Channel x priority status: 0: Invalid 1: Channel x uses the default low priority
1		
0		

10.4.18. DMA interrupt enable register(CHNL_INT_EN)

Address offset: 0x50

Reset value: 0x0000 0000

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

15: 3	2	1	0
Res.	INT_EN2	INT_EN1	INT_EN
	RW	RW	RW

31: 3	-	Reserved
2	INT_EN2	Channel 2 interrupt enable register: 0: Channel 2 interrupt is disabled 1: Channel 2 interrupt is enabled
1	INT_EN1	Channel 1 interrupt enable register: 0: Channel 1 interrupt is disabled 1: Channel 1 interrupt is enabled
0	INT_EN0	Channel 0 interrupt enable register: 0: Channel 0 interrupt is disabled 1: Channel 0 interrupt is enabled

10.4.19. DMA channel transfer completion flag register (CHNL_DONE_ISR)

Address offset: 0x54

Reset value: 0x0000 0000

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

15: 3	2	1	0
Res.	DONE_F2	DONE_F1	DONE_F0
	R	R	R

31: 3	-	Reserved
2	DONE_F2	Channel 2 transmission complete flag: 0: Channel 2 transmission incomplete 1: Channel 2 transmission completion interrupt flag
1	DONE_F1	Channel 1 transmission complete flag: 0: Channel 1 transmission incomplete 1: Channel 1 transmission completion interrupt flag
0	DONE_F0	Channel 0 transmission complete flag: 0: Channel 0 transmission incomplete 1: Channel 0 transmission completion interrupt flag

10.4.20. DMA channel transfer completion flag clear register (CHNL_DONE_IFCR)

Address offset: 0x58

Reset value: 0x0000 0000

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

15: 3	2	1	0
Res.	DONE_FC2	DONE_FC1	DONE_FC0
	W	W	W

31: 3	-	Reserved
2	DONE_FC2	Channel 2 Transmission complete interrupt flag cleared: 1: Clear the channel 2 transmission completion interrupt flag DONE_F2 0: Invalid
1	DONE_FC1	Channel Transmission complete interrupt flag cleare: 1: Clear the channel 2 transmission completion interrupt flag DONE_F1 0: Invalid
0	DONE_FC0	Channel 0 Transmission complete interrupt flag cleare: 1: Clear the channel 2 transmission completion interrupt flag

		DONE_F0 0: Invalid

10.4.21. DMA channel transfers error flag register (CHNL_ERR_ISR)

Address offset: 0x5C

Reset value: 0x0000 0000

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

15: 3	2	1	0
Res.	ERR_F2	ERR_F1	ERR_F0
	R	R	R

31: 3	-	Reserved
2	ERR_F2	Channel 2 transmission error interrupt flag: 0: Channel 2 no transmission error interrupts 1: Channel 2 transmission error interrupt flag
1	ERR_F1	Channel 1 transmission error interrupt flag: 0: Channel 1 no transmission error interrupts 1: Channel 1 transmission error interrupt flag
0	ERR_F0	Channel 0 transmission error interrupt flag: 0: Channel 0 no transmission error interrupts 1: Channel 0 transmission error interrupt flag

10.4.22. DMA channel transmission error flag clear register (CHNL_ERR_IFCR)

Address offset: 0x60

Reset value: 0x0000 0000

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

15: 3	2	1	0
Res.	ERR_FC2	ERR_FC1	ERR_FC0
	W	W	W

31: 3	-	Reserved
2	ERR_FC2	Channel 2 Error interrupt flag cleared: 1: Clear the channel 2 transmission error interrupt flag ERR_F2 0: Invalid

1	ERR_FC1	Channel 1 Error interrupt flag cleared: 1: Clear the channel 1 transmission error interrupt flag ERR_F1 0: Invalid
0	ERR_FC0	Channel 0 Error interrupt flag cleared: 1: Clear the channel 0 transmission error interrupt flag ERR_F0 0: Invalid

10.4.23. DMA channel 0 mapping request source select register (DMAMUX_SEL00)

Address offset: 0x00

Reset value: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.										SEL_CH00					
										RW					

31: 6	-	Reserved
5: 0	SEL_CH00	Select the request source mapped to DMA channel 0: 000000: Fixed connection to the low level 000001: Request source 1 000010: Request source 2 000011: Request source 3 ...... 101001: Request source 41 Other: Invalid Notes: The request source numbers are described in section DMAMUX request Sources

10.4.24. DMA channel 1 mapping request source select register (DMAMUX_SEL01)

Address offset: 0x04

Reset value: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.										SEL_CH01					
										RW					

31: 6	-	Reserved
5: 0	SEL_CH01	Select the request source mapped to DMA channel 1: 000000: Fixed connection to the low level 000001: Request source 1 000010: Request source 2 000011: Request source 3 ...... 101001: Request source 41 Other: Invalid Notes: The request source numbers are described in section DMAMUX request Sources

10.4.25. DMA channel 2 mapping request source select register (DMAMUX_SEL02)

Address offset: 0x08

Reset value: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.										SEL_CH02					
										RW					

31: 6	-	Reserved
5: 0	SEL_CH02	Select the request source mapped to DMA channel 1: 000000: Fixed connection to the low level 000001: Request source 1 000010: Request source 2 000011: Request source 3 ...... 101001: Request source 41 Other: Invalid Notes: The request source numbers are described in section DMAMUX request Sources

10.5. DMA configuration process

Peripherals to memory transfer:

1. Configure DMA register:
   a) CTRL_BASE_PTR register: Configure the pointing channel control data structure base address(0x5008 0000);
   b) DMA_CFG register: Configuration enables DMA controller;
   c) CHNL_EN_SET register: Configuration enables DMA channel ;
   d) CHNL_INT_EN register: Configuration enables channel interrupts;
   e) CHNL_PRI_SET register: Set the channel priority, default is low priority;
   f) In basic and Ping-Pong mode, you can configure the CHNL_PRI_ALT_SET register and select the channel data structure. The primary data structure is used by default;
   g) Configure channel control data structure registers (stored in DMA SRAM space):

      CHNLx_SRC_END_PTR register: Configuration points to the end address of the source data;

      CHNLx_DST_END_PTR register: Configuration points to the end address of the destination data;

      CHNLx_CONTROL register: Configure the DMA working mode,Each DMA transfer length,Arbitration rate,Source data/destination data bit width,Source address/destination address increment;
2. CYCLE_CTRL can be configured as the basis,Ping-pong or peripheral dispersion/aggregation cycle type(If a transmission is greater than 1024 (optional unit: word, half word, or byte), you can select the peripheral dispersion/aggregation period type);
3. Configure peripheral DMA enable;
4. After the DMA transmission is completed, the transmission completion interrupt is generated, and the channel completion flag bit can be Read to judge the transmission completion channel; To transfer again, you need to reconfigure the channel enablement and channel data structure register(DMA transmission length, working mode).

Memory to memory transfer:

1. Configure DMA register:
   a) CTRL_BASE_PTR register: Configure the pointing channel data structure base address(0x5008 0000);
   b) DMA_CFG register: Configuration enables DMA controller;
   c) CHNL_EN_SET register: Configuration enables DMA channel;
   d) CHNL_INT_EN register: Configuration enables interrupts;
   e) CHNL_PRI_SET register: Set the priority, default is low priority;
   f) In automatic request mode, you can configure the CHNL_PRI_ALT_SET register, select the channel data structure, and use the main data structure by default;
2. Configure channel control data structure registers(Stored in DMA_SRAM space)
   a) CHNLx_SRC_END_PTR register: Configuration points to the end of the source data
   b) CHNLx_DST_END_PTR register: Configuration points to the destination data end

address;

c) CHNLx_CONTROL register: Configure the DMA working mode,Each DMA transfer length,Arbitration rate,Source data/destination data bit width,Source address/destination address increment;

3. CYCLE_CTRL can be configured as automatic request or memory dispersion/aggregation cycle type(If a transmission is greater than 1024 (optional unit: word, half word, or byte), you can select the memory dispersion/aggregation period type);
4. Configure the CHNL_SW_REQUEST register to generate a DMA software request signal;
5. After the completion of DMA transmission, the transmission completion interrupt is generated, and the Read channel completion flag bit can be used to determine the transmission completion channel. To transfer data again, you need to reconfigure the channel enablement and channel data structure(DMA transmission length, working mode).

10.6. Note

1. To disable DMA, disable DMA interrupt, disable DMA module, and then clear the interrupt flag register.
2. When multiple channels are used at the same time, if a transmission error occurs on one channel, the channel will be disabled by the hardware, which will affect the transmission of other channels, and DMA needs to be reconfigured and the current transmission is ignored.
3. If multiple channels are used at the same time, evaluate the request interval (recommended to be greater than 20 HCLK). If the channel requests are frequent, the requests will be lost.
4. When multiple channels are enabled, the read-back check is performed after the interrupt flag is cleared. If the channel is not cleared, clear it again.
5. If the data bit width is a word, the source data end address and destination data end address must be aligned with 4 bytes. If the data bit width is half a word, the end address of the source data and the end address of the destination data should be aligned by 2/4 bytes. Data is transmitted forward from the start address to the end address.
6. DMA does not support erasing or write FLASH.
7. For DMA data transmission of the SPI/UART/IIC/ADC module, the arbitration rate R_power must be set to 0:

   If the configuration is incorrect, the communication is interrupted if it is not completed. The CHNL_DONE_ISR identifier for channel transmission completion is set to 1, and the CHNL_ERR_ISR identifier for channel transmission error is set to 0. If the channel control register and channel enable are not reconfigured, the DMA corresponding channel will no longer respond to module requests; If the module is sending data, it will no longer send or send the previous data; If the module is receiving data, pull the overflow flag bit higher after the overflow.
8. DMA transfer of TIM module:
   a) When only one data transfer is required at a time, the R_power must be equal to 0, and the data address can be directed to the destination register, or the register mapping can be controlled via TIM DMA.
   b) When multiple data transfers are requested at once, control register mapping via TIM DMA, and note that DBL is configured in the same way as R_power(The DMA arbitration rate can only be an exponential of 2, and the DBL can only be an even number).
9. Switch the CHNL_PRI_ALT_SET/CHNL_PRI_ALT_CLR value when the controller completes the following conditions:
   a) For memory decentralized aggregation mode or peripheral decentralized aggregation mode, after the main data structure is transferred, the hardware switch selects the standby data structure, and after all DMA transfers specified by the standby data structure are completed, the hardware switch selects the main data structure.
   b) For Ping-Pong mode, after all DMA transfers specified by the master or alternate data structure are completed, the hardware switches to the other data structure.

Chapter 11 Advanced control timer(TIM0)

11.1. TIM0 overview

The Advanced Control Timer (TIM0) contains a 16-bit automatic overload counter that is driven by a programmable predivider. The secondary mode controller allows the selection of the counter's clock source, input trigger signal and output signal.

11.2. TIM0 features

- 16-bit increment, decrement, increment/decrement automatically reload counters
- 16-bit programmable prescaler (support readification). The frequency division factor of the counter clock frequency is any value between 1 and 65536
- Up to 6 separate channels, support:
  - ○ Input capture (except channel 5/6)
  - ○ Output comparison
  - ○ PWM generation(edge and center alignment patterns)
  - ○ Single pulse mode output
- Support for programmable dead-time complementary outputs
- Support the use of timer and external signal control can realize multiple timers (TIM6) interconnected synchronous circuit
- A repeat counter used to update the timer register after a given number of counter cycles
- 2 brake input and output signal is used to the timer is put in user optional security configuration
- An interrupt /DMA request is generated when the following events occur:
  - ○ Update: Counter overflow/underflow, counter initialization (triggered by software or internally/externally)
  - ○ Trigger event (counter start, stop, initialize, or trigger count via internal/external)
  - ○ Input capture
  - ○ Output comparison
  - ○ brake input (interrupt request)
- Support for incremental (quadrature) encoder and Hall sensor circuits for positioning
- The trigger input is used as an external clock

11.3. TIM0 functional description

11.3.1. TIM0 structure diagram

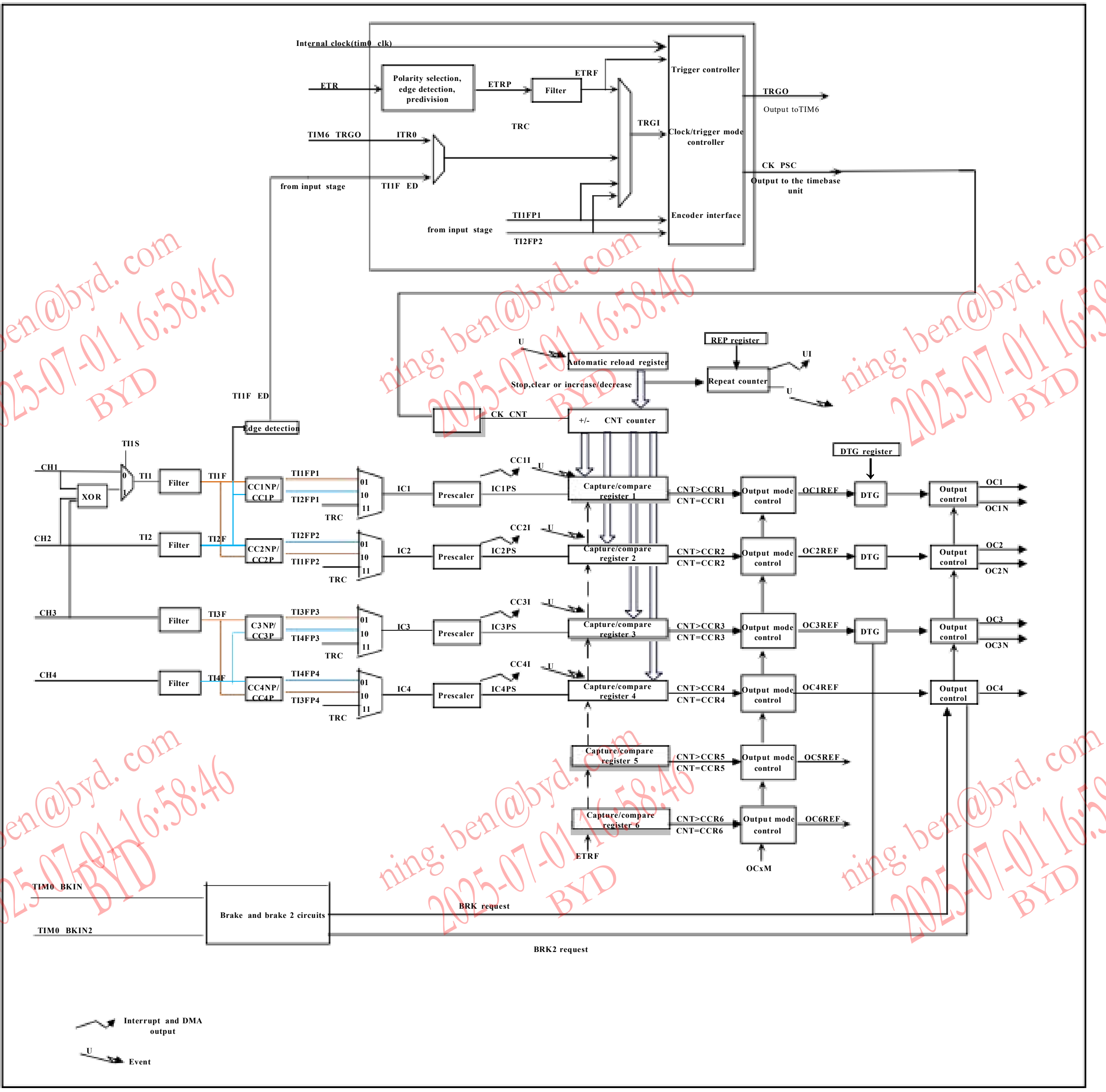


Figure 11.1 Advanced timer (TIM0) structure block diagram

11.3.2. Clock selection

Counter clocks can be provided by the following clock sources:

- Internal clock TIM0_CLK: PLL_48MHz/LIRC 32K/XTAL(Passive crystal oscillator(32768Hz/4MHz/8MHz)and Active crystal oscillator(1MHz~48MHz))
- External clock mode 1: External input pin or internal trigger input (ITR0)
- External clock mode 2: External trigger input ETR
- Encoder mode

The slave mode controller allows the user to select the clock source of the counter, the input trigger signal and the output signal.

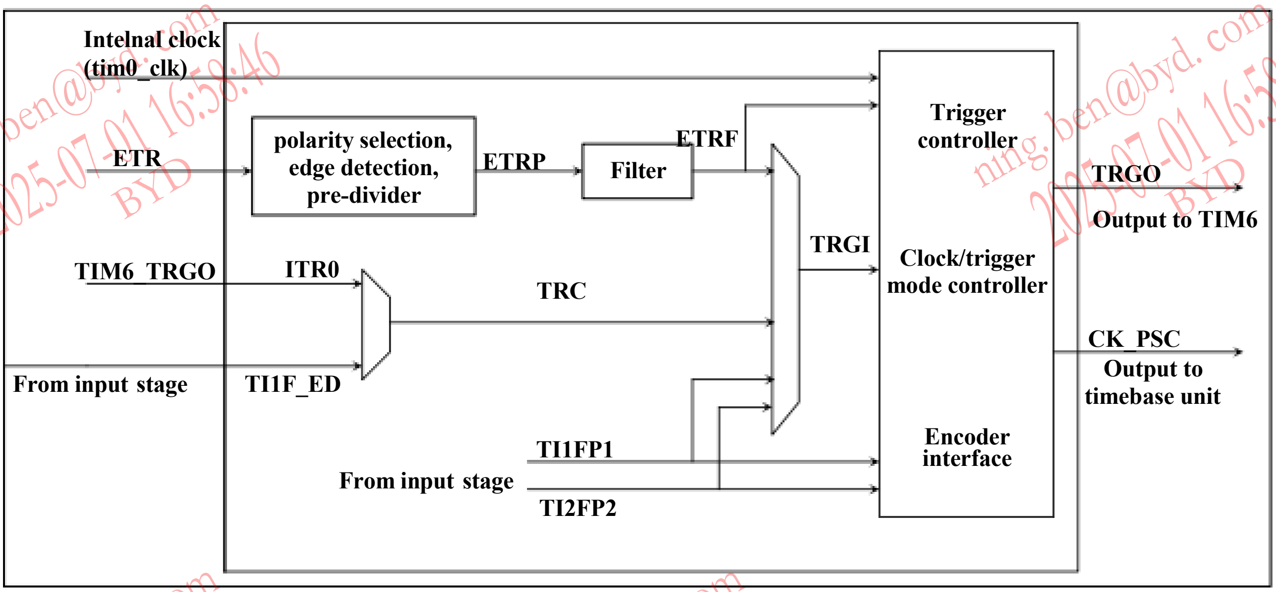


Figure 11.2 Slave mode controller block diagram

11.3.2.1. Internal clock(TIM0_CLK)

If slave mode controller is disabled (SMS = 0000), the counter enable CEN bit, counter count direction DIR bit (TIM0_CR1 register) and UG bit (TIM0_EGR register) are actual control bits and can only be changed by software (except UG, which remains auto-reset). When writing 1 to the CEN bit, the clock of the predivider is directly provided by the internal clock TIM0_CLK.

11.3.2.2. External clock mode 1

When SMS=0111 in the TIM0_SMCR register, select external clock source mode 1; The counter counts when a rising or falling edge appears on the selected input signal.

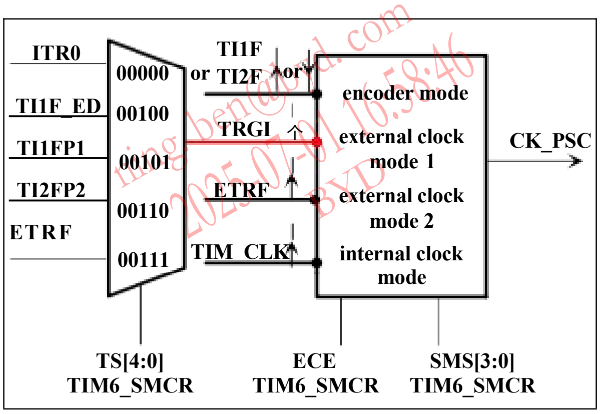


Figure 11.3 External clock mode 1 function block diagram

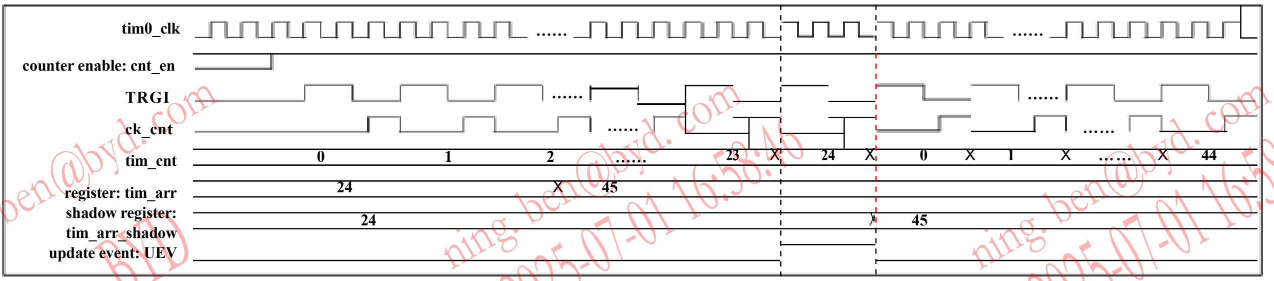


Figure 11.4 External clock source mode 1 Count timing diagram

For example, to make the incrementing counter count when the TI2 input has a rising edge, perform the following steps

1. Configure the TIM0_CCMR1 register CC2S = 01 and configure channel 2 as input so that it can detect the rising edge of the TI2 input;
2. Configure the TIM0_CCMR1 register IC2F[3: 0] and configure the input filtering bandwidth as required;
3. Configure the TIM0_CCER registers CC2P = 0 and CC2NP = 0 to select the rising edge polarity;
4. Configure the TIM0_SMCR register SMS[3: 0] = 0111 to make the timer work in external clock mode 1;
5. Configure the TIM0_SMCR register TS[4: 0] = 00110, and select TI2FP2 after TI2 filtering as the trigger input source;
6. Configure the TIM0_CR1 register CEN = 1 to enable the counter;

11.3.2.3. External clock mode 2

When ECE= 1 in the TIM0_SMCR register, select external clock source mode 2. The counter can count when the external trigger input ETR has a rising edge or falling edge, and the ETP controls the choice of the polarity of the external trigger input.

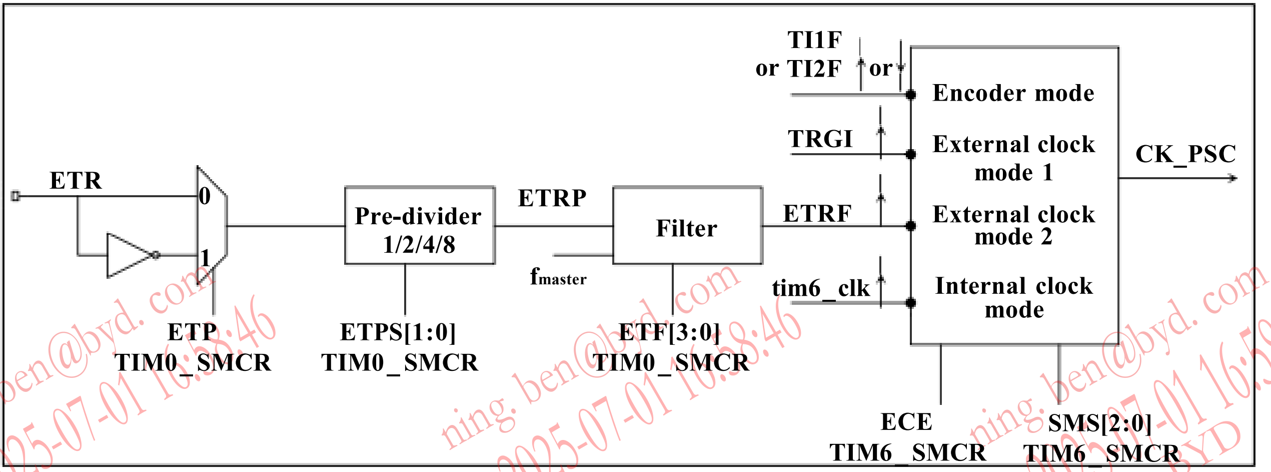


Figure 11.5External trigger input module

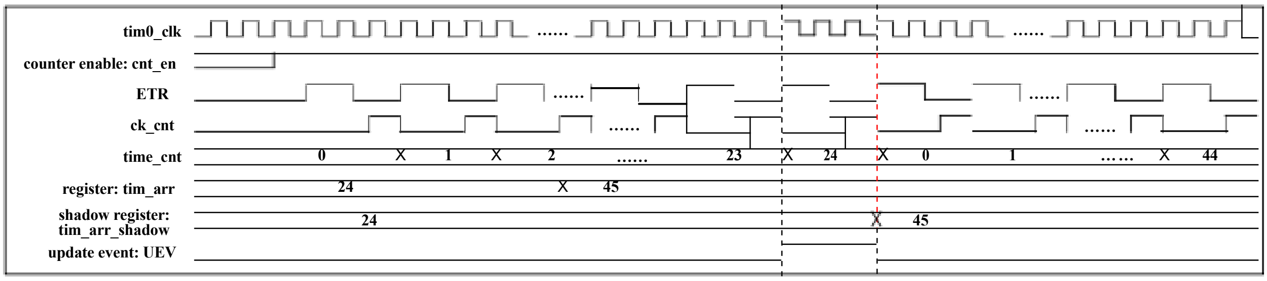


Figure 11.6 External clock source mode 2 Count timing diagram

For example, to make the incrementing counter count every time 1 falling edge occurs on the ETR, perform the following steps:

1. Configure the TIM0_SMCR register ECE = 1 and select external clock 2 mode;
2. Configure the TIM0_SMCR register ETF[3: 0] and configure the external trigger input filtering as needed;
3. Configure the TIM0_SMCR register ETPS[0: 1] = 00 and turn off the predivider;
4. Configure the TIM0_SMCR register ETP = 1 and select the falling edge detection of the ETR pin;
5. Configure the TIM0_CR1 register CEN = 1 to enable the counter;

11.3.2.4. Encoder mode

Write SMS=0001 into the TIM0_SMCR register, select encoder mode 1, and count at the TI1F edge; Write SMS=0010, select encoder mode 2 and count at TI2F edge; Write SMS=0011, select encoder mode 3, and the counter counts at both TI1F and TI2F edges.

11.3.3. Time base unit

Programmable timer advanced control TIM0 main module is a 16-bit counters and other related auto reload register. Counters can be increasing counting, decreasing counting or alternation of increasing and decreasing. The clock of the counter can be divided by a predivider. Counters, automatic overload registers, and predivider registers can be read or write by software, even when the counter is running.

Time base units include:

- Predivider register(TIM0_PSC)
- Counter register(TIM0_CNT)
- Automatic reload register(TIM0_ARR)
- Repeat counter register(TIM0_RCR)

11.3.3.1. Predivider PSC

The predivider PSC has an input clock CK_PSC and an output clock CK_CNT. The input clock comes from the controller part, and different CK_CNT can be obtained by setting the value of the pre-division frequency,and It actually calculates the formula: $CK\_CNT = F_{CK\_PSC}/(PSC[15:0]+1)$. Because the TIM0_PSC control register has a buffer (shadow register) that can change its value while it is running, the new predivision frequency value will take effect at the next update event.

11.3.3.2. Counter CNT

The counter can be incremented, decremented, or alternately incremented and decremented. The counter is started only when CEN in the TIM0_CR1 register is set to 1.

Notes: The counter will start counting after one clock cycle at the CEN position 1 moment of the TIM0_CR1 register.

11.3.3.3. Automatic reload register ARR

The automatically loaded register is preloaded and is accessed during Write in or Read fetch operations. The contents ofthe TIM0_ARR register can be transferred either directly to the shadow register or each time an update event occurs, depending on the automatic Reload Preload enable bit (ARPE) in the TIM0_CR1 register. An update event is sent when the counter reaches an overflow value (or an underflow value when decrement counts) and the UDIS bit in the TIM0_CR1 register is 0. The update event can also be generated by software.

11.3.3.4. Repeat counter RCR

When the timer overflows or underflows, the value of the repeat counter is reduced continuously. The value of the repeat counter is reduced once the count overflows. An update event is generated only when the repeat counter is 0.

11.3.4. Counter mode

11.3.4.1. Incremental count mode

In incremental count mode, the counter counts from 0 to an automatically overloaded value (the contents ofthe TIM0_ARR register), then counts again from 0 and generates a counter overflow event.

If a repeat counter is used, an update event (UEV) is generated when the number of repetitions of the incremental count reaches the number of repetitions set in the repeat counter register plus one (TIM0_RCR + 1). Otherwise, an update event occurs each time the counter overflows.

An update event is also generated when the UG position 1 of the TIM0_EGR register is changed (either by software or using a slave mode controller).

The UEV update event can be disabled by placing UDIS position 1 in the TIM0_CR1 register by software; This avoids updating the shadow register when new values are written to the preloaded register; No update event occurs until the UDIS bit is written to 0, however, both the counter and the predivider counter start counting again from 0 (while the predivider ratio remains the same).

In addition, if the URS bit in the TIM0_CR1 register (the source of the update request) is set to 1, placing UG at position 1 generates the update event UEV but does not set the update interrupt flag UIF (therefore, no interrupt or DMA request is sent) to avoid both update and capture interrupts when the counter is cleared in capture mode.

When an update event occurs, all registers are updated and the update flag (UIF bit in the TIM0_SR register) is set to 1 (depending on the URS bit):

- Repeat counter reloads the contents ofthe TIM0_RCR register
- The automatic overload shadow register will be updated with the preloaded value (TIM0_ARR)
- The predivider's buffer will be reloaded with the preloaded value (the contents ofthe TIM0_PSC register)

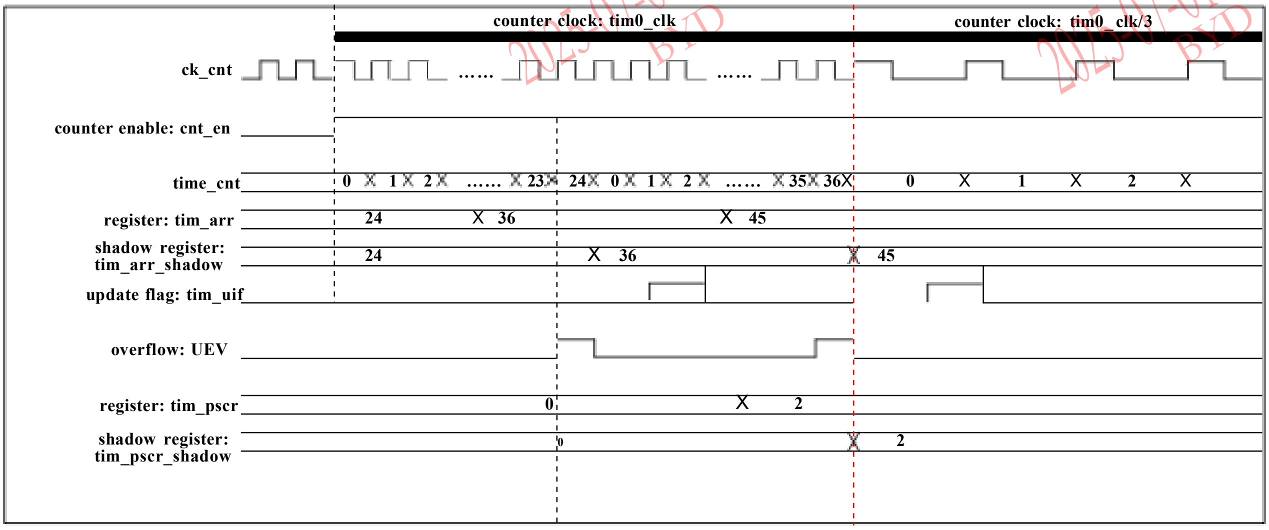


Figure 11.7 Incremental counting sequence diagram

11.3.4.2. Decrement count mode

In decrement count mode, the counter decrement the count from the automatic overload value (the contents ofthe TIM0_ARR register) to 0, then counts again from the automatic overload value and generates a counter underflow event.

f a repeat counter is used, an update event (UEV) is generated when the number of repeats of the decrement count reaches the number of times programmed in the repeat counter register plus one (TIM0_RCR + 1). Otherwise, an update event occurs each time the counter underflows.

An update event is also generated when the UG position 1 of the TIM0_EGR register is changed (either by software or using a slave mode controller).

Update events can be disabled by placing UDIS position 1 in the TIM0_CR1 register by software. It can avoid the preloaded writing new value update shadow registers. No update events occur until the UDIS bit is written to 0. However, the counter starts counting from the current automatic overload value, while the predivider counter starts counting again from 0 (while the predivider ratio remains the same).

In addition, if the URS bit (update request selection) in the TIM0_CR1 register is set to 1, the UG position 1 generates an update event UEV, but does not set the update interrupt flag UIF to 1 (therefore, no interrupt or DMA requests are sent). This is to avoid both update and capture interruptions when the counter is cleared in capture mode.

When an update event occurs, all registers are updated and the update flag (UIF bit in the TIM0_SR register) is set to 1 (depending on the URS bit):

- Repeat counter reloads the contents ofthe TIM0_RCR register
- The automatic overload shadow register will be updated with the preloaded value (TIM0_ARR)
- The predivider's buffer will be reloaded with the preloaded value (the contents ofthe TIM0_PSC register)

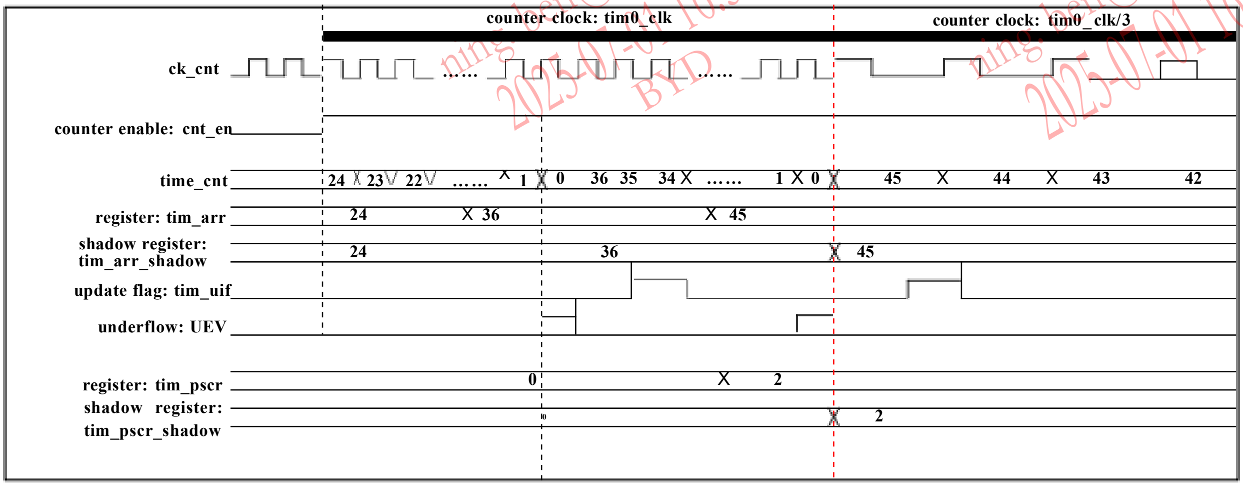


Figure 11.8 Decrement count sequence diagram

11.3.4.3. Center Alignment Mode (Increment/decrement count mode)

In center alignment mode, the counter counts from 0 to the automatically overloaded value (the contents ofthe TIM0_ARR register) -1, generating a counter overflow event; It then counts down to 1 starting from the automatic overloaded value and generates a counter underflow event; Then count again from 0.

The center alignment mode is valid when the CMS bit in the TIM0_CR1 register is not "00". When a channel is configured in output mode, its output comparison interrupt flag is set to 1 in the following modes: counter decrement count (center alignment mode 1, CMS = "01"), counter increment count (center alignment mode 2, CMS = "10"), and counter increment/decrement count (center alignment mode 3, CMS = "11").

In this mode, you cannot Write the DIR direction bit in TIM0_CR1, which is updated by the hardware and indicates the current count direction.

Update events are generated each time a counter overflows and underflows occur; Update events can also be generated by setting the UG bit in the TIM0_EGR register (software or using the slave mode controller). Then, the counter starts counting again from 0, and the pre-divider starts counting again from 0.

Setting the UDIS bit in the TIM0_CR1 register disables UEV events. This avoids updating the shadow register when new values are written to the preloaded register. Therefore, no update event occurs until the UDIS bit is cleared to 0. However, the counter continues to count up or down depending on the current automatic reload value.

In addition, if the URS bit (update request selection) in the TIM0_CR1 register is set, setting the UG flag1 will generate an update event UEV but will not set the UIF flag to 1 (thus generating no interrupt and DMA request), in order to avoid generating both update and capture interrupts when a capture event occurs and the counter is cleared.

When an update event occurs, all registers are updated and the update flag (UIF bit in the TIM0_SR register) is set to 1 (depending on the URS bit):

- Repeat counter reloads the contents ofthe TIM0_RCR register
- The automatic overload shadow register will be updated with the preloaded value (TIM0_ARR)
- The predivider's buffer will be reloaded with the preloaded value (the contents ofthe TIM0_PSC register)

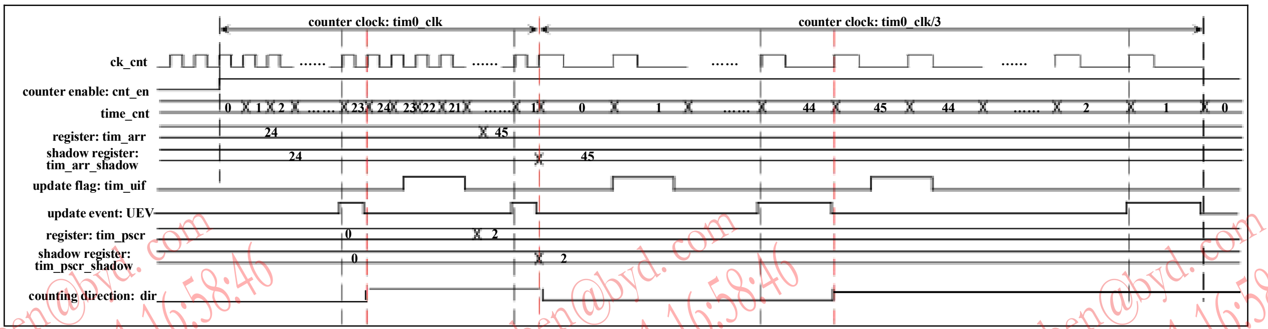


Figure 11.9 Center alignment counting sequence diagram

11.3.5. Repeat counter

In fact, update events are generated only when the repeat counter reaches zero. This means that whenever N+1 counter overflows or underflows occur (where N is the value in the TIM0_RCR repeat counter register), Data will then be transferred from the pre-loaded register to the shadow register (TIM0_ARR automatic reload register, TIM0_PSC pre-divider register, and TIM0_CCRx capture/compare register in comparison mode).

The repeat counter decays in the following cases:

- Each counter overflows in incremental count mode
- Each counter underflows in decremental count mode
- Center alignment mode for each counter overflow and counter underflow. Although this allows the maximum repetition to be no more than 32,768 PWM cycles, the duty cycle can be updated twice per PWM cycle. When the comparison register is refreshed only once per PWM cycle in center alignment mode, the maximum resolution is 2*T_{ck} due to the symmetry of the mode.

The repeat counter is an automatic reload type; The repetition rate is the value defined in the TIM0_RCR register. When the update event is generated by software (by setting the UG position of the TIM0_EGR register to 1) or hardware (by coming from the pattern controller), the update event occurs immediately regardless of the value of the repeat counter, and the contents ofthe TIM0_RCR register are reloaded in the repeat counter.

In center alignment mode, if the RCR value is odd, the update event will occur on overflow or underflow, depending on when the RCR register is written and when the counter is started: IfRCR is written before the counter is started, UEV occurs during an overflow. If the RCR is written after the counter is started, the UEV occurs during underflow.

11.3.6. Timer synchronization

TIM0 timers are connected internally to synchronize or link timers, and they can be synchronized in several modes: reset mode, gated mode, and trigger mode.

11.3.6.1. Slave mode: Reset mode

The counter and its predivider can be re-initialized when the trigger input signal changes. In addition, if the URS bit in the TIM0_CR1 register is 0, the update event UEV is generated. Then, all the preloaded registers (TIM0_ARR and TIM0_CCRx) are updated.

Start the counter by writing CEN= 1 in the TIM0_CR1 register. The counter starts counting against the internal clock until a trigger input (TRGI) rising edge appears, when the counter clears and starts counting again from 0. At the same time, the trigger interrupt flag (TIF bit in the TIM0_SR register) is set to 1. After the interrupt or DMA is enabled, the trigger interrupt or DMA request can also be sent (depending on the TIE and TDE bits in the TIM0_DIER register).

In the following example, the increment counter clears when a rising edge appears on the TI1 input:

1. Configure the CC1S = 01 of the TIM0_CCMR1 register and configure the CC1 channel as input.
2. Configure the CC1P = 0 and CC1NP = 0 of the TIM0_CCER register, and select the rising edge contact detection of TI1.
3. Configure the SMS = 0100 of the TIM0_SMCR register to set the timer to reset mode.
4. Configure the TS = 00101 of the TIM0_SMCR register and select TI1FP1 filtered by TI1 as the input source.
5. Configure the CH1DIR = 1 of the TIM0_DIR register , and configure TIM0_CH1 as input.
6. Write CEN = 1 to the TIM0_CR1 register to start the counter.

11.3.6.2. Slave mode: Gated mode

The level of the input signal can be used to enable the counter; As long as the TRGI is high, the counter starts counting against the internal clock and stops counting until the TRGI becomes low. When the counter is started or stopped, the trigger interrupt flag TIF in the TIM0_SR register is set to 1.

In gated mode, if CEN=0, the counter does not start regardless of the trigger input level.

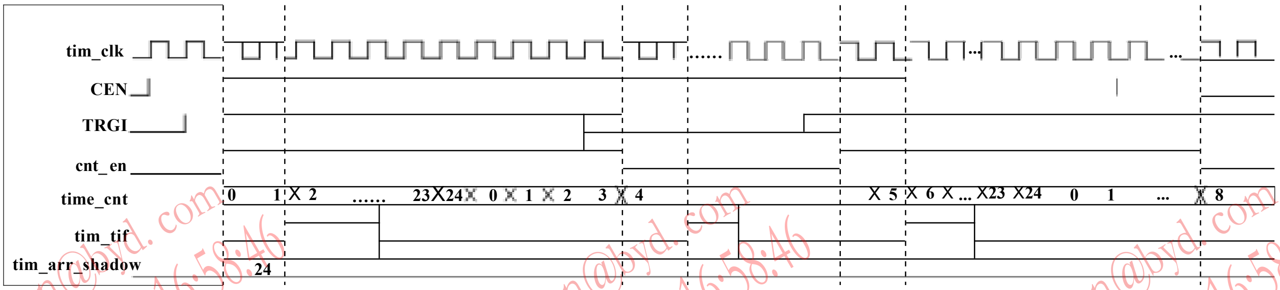


Figure 11.10 Gated mode timing diagram

In the following example, the counter counts at TI1 input low level:

1. Configure the CC1S = 01 of the TIM0_CCMR1 register, configure the CC1 channel as

input, and detect TI1.

2. Configure the CC1P = 1 and CC1NP = 0 of the TIM0_CCER register to detect low TI1 levels.
3. Configure the SMS = 0101 of the TIM0_SMCR register to set the timer to gated mode.
4. Configure the TS = 00101 of the TIM0_SMCR register and select TI1FP1 filtered by TI1 as the input source.
5. Configure the CH1DIR = 1 of the TIM0_DIR register , and configure TIM0_CH1 as input.
6. Write CEN = 1 to the TIM0_CR1 register to start the counter.

11.3.6.3. Slave mode: Trigger mode

The counter can be started when the selected trigger input signal appears. When TRGI appears with a rising edge, the counter starts counting against the internal clock and the TIF flag is set to 1.

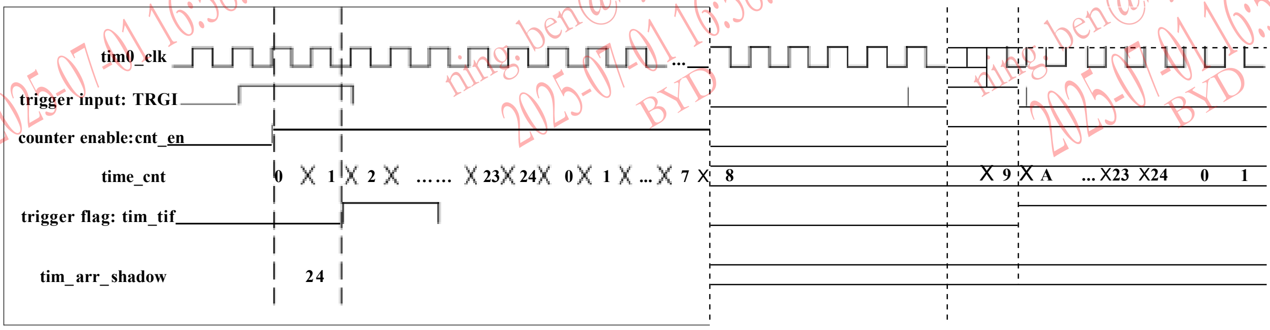


Figure 11.11 Trigger mode timing diagram

In the following example, the counter starts when a rising edge appears on the TI2 input:

1. Configure the CC2S = 01 of the TIM0_CCMR1 register, configure the CC2 channel as input, and detect TI2
2. Configure the CC2P = 0 and CC2NP = 0 of the TIM0_CCER register to detect the rising edge of TI2.
3. Configure the SMS = 0110 of the TIM0_SMCR register to set the timer to trigger mode.
4. Configure the TS = 00110 of the TIM0_SMCR register and select TI2FP2 after TI2 filtering as the input source.
5. Configure the CH1DIR = 1 of the TIM0_DIR register , and configure TIM0_CH1 as input.

11.3.6.4. Slave mode: Combined reset + trigger mode

When the selected trigger input (TRGI) rise edge appears, the counter is reinitialized, a register update event is generated, and the counter is started. This mode is used for monopulse mode.

11.3.6.5. Slave mode: External clock mode 2+ trigger mode

External clock mode 2 can be used in combination with another slave mode (except external clock mode 1 and encoder mode). In this case, the ETR signal is used as the external clock input, and another input is selected as the trigger input TRGI(in reset mode, gated mode, or trigger mode). The TS bit in the TIM0_SMCR register disables the selection of ETR as TRGI.

In the following example, when a rising edge appears in TI1, the counter counts at each rising edge of the ETR signal:

1. Configure the external clock mode 2 input circuit to configure the TIM0_SMCR register: ETF = 0000(No filter),ETPS = 00(Pre-dividers are forbidden),ETP = 0(The rising edge of the ETR is detected),ECE = 1(Enable external clock mode 2)
2. Configure channel 1 to detect the rising edge of TI1: configure CC1S = 01 of the TIM0_CCMR1 register, configure CC1 channel as input, and detect TI1; Configure the CC1P = 0 and CC1NP = 0 of the TIM0_CCER register to determine polarity (detect rising edge).
3. Configure the SMS = 0110 of the TIM0_SMCR register to set the timer to trigger mode.
4. Configure the TS = 00101 of TIM0_SMCR register and select TI1P1 after TI1 wave as input source.

The counter is enabled and the TIF flag is set to 1 when a rising edge appears in TRGI. The counter then counts at each rising edge of the ETR signal.

11.3.7. ADC synchronization

The timer can generate ADC trigger events from a variety of internal signals, such as reset, enable, or compare events; Also can be generated by the internal issue of pulse edge detector, such as, rising and falling of OC4REF along, rising of OC5REF along or falling of OC6REF along.

There are 16 possible events that are triggered on the TRGO2 internal circuit that is redirected to the ADC and can be selected by the MMS2[3: 0] bit in the TIM0_CR2 register.

Notes:

1. **The clock of the slave peripheral (timer, ADC, etc.) that receives the TRGO or TRGO2 signal must be enabled before the event can be received from the master timer. And when receiving the trigger signal from the master timer, the clock frequency must not be changed in real time (predivider).**
2. **The ADC clock must be enabled before events can be received from the main timer. The ADC clock must not be changed in real time when the trigger signal is received from the timer.**

11.3.8. Capture/Compare channels

Each capture/comparison channel is built around a capture/comparison register (including a shadow register), a capture input part (digital filtering, multiplexing, and predivider, except for channels 5 and 6), and an output part (comparator and output control).

In the input phase, the corresponding TIx input is sampled and a filtered signal TIxF is generated. The edge detector with polarity selection then generates a signal (TIxFPx) that can be used either as a trigger input from the mode controller or as a capture command. The signal is first predivided (ICxPS) and then entered into the capture register.

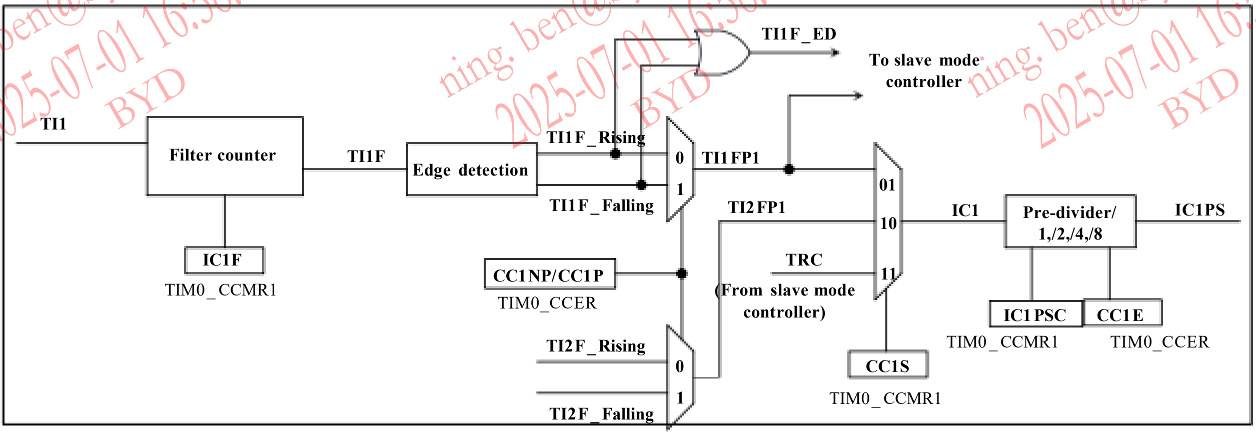


Figure 11.12 Capture/compare the input phase of the channel(Example: Channel 1 input phase)

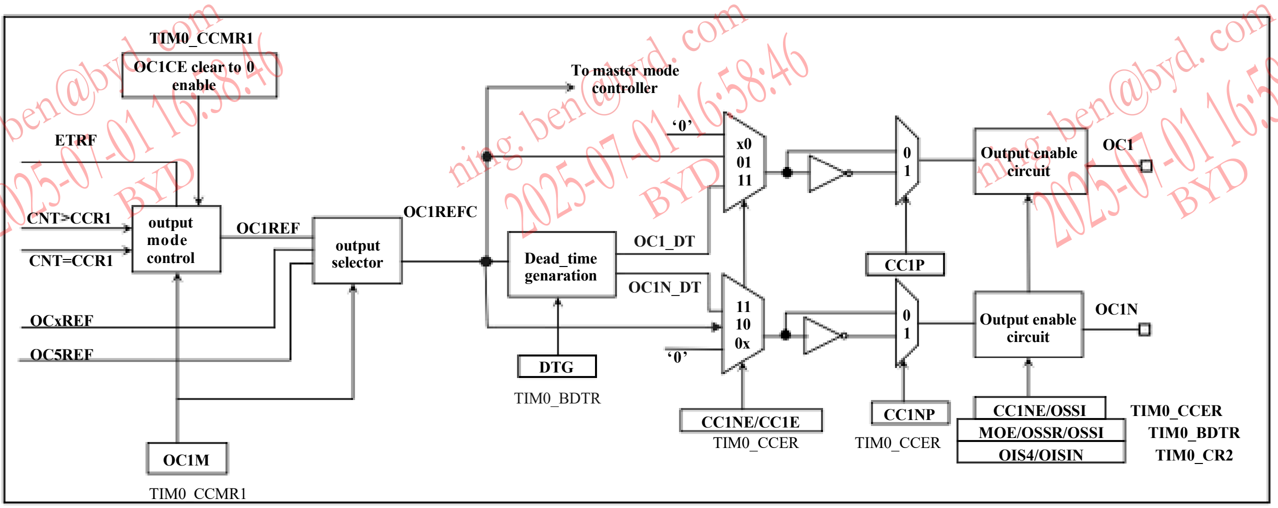


Figure 11.13 Capture/compare the output phase of the channel(Channel 1, 2, and 3)

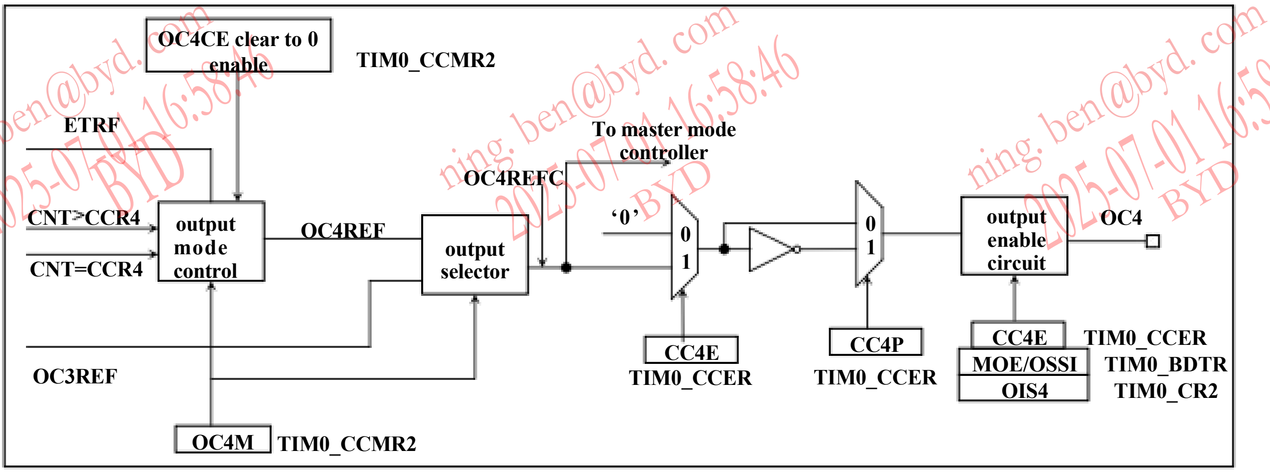


Figure 11.14 Capture/compare the output phase of the channel(Channel 4)

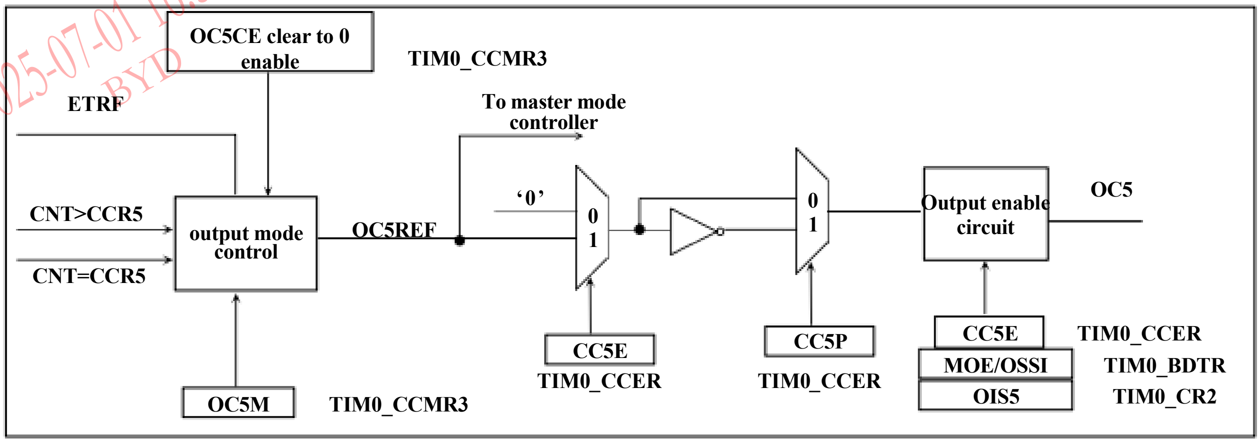


Figure 11.15 Capture/compare the output phase of the channel(Channel 5and Channel 6)

The capture/compare module consists of a preloaded register and a shadow register. Preloaded registers are always accessible by read and write operations.

In capture mode, the capture actually takes place in the shadow register, and the contents ofthe shadow register are then copied into the preloaded register.

In comparison mode, the contents ofthe preloaded register are copied into the shadow register, and then the contents ofthe shadow register are compared with the counter.

11.3.9. Input capture mode

In input capture mode, the capture/comparison register (TIM0_CCRx) is used to latch the counter value when the corresponding ICx signal detects a jump edge. When a capture event occurs, the corresponding capture/compare interrupt flag CCxIF (TIM0_SR register) is set to 1 and an interrupt or DMA request can be sent (if CCxIE and CCxDE are enabled). If the CCxIF flag is already high when a capture event occurs, the CCxOF (TIM0_SR register) is set to 1. The CCxIF can be reset by software either by writing a "0" to it or by reading the captured data stored in the TIM0_CCRx register. CCxOF clears zero when "0" is written.

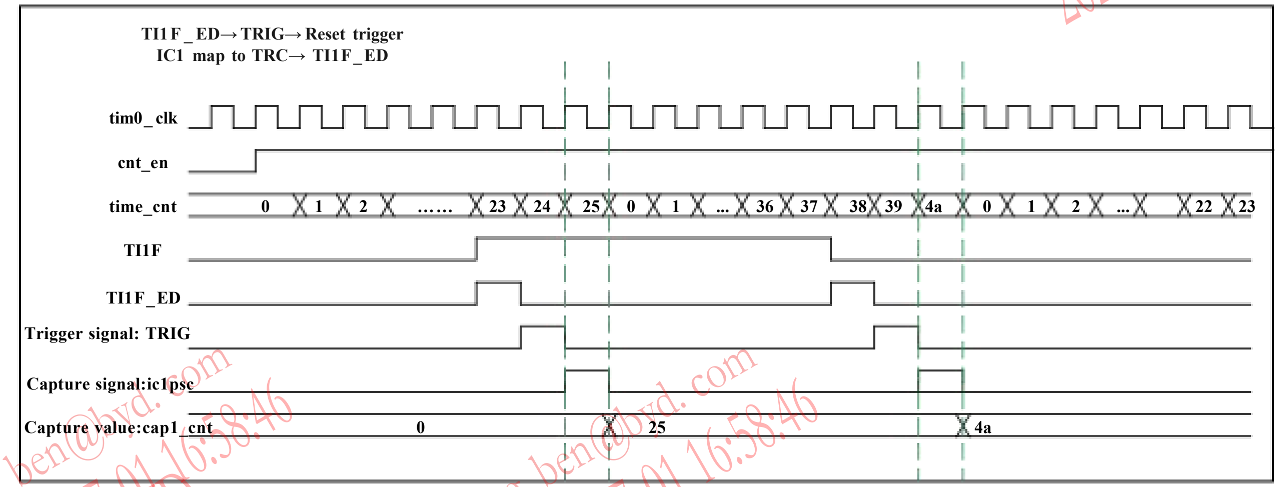


Figure 11.16 Input capture timing diagram

The following example shows how to capture the counter's value into TIM0_CCR1 when the TI1 input has a rising edge:

1. Select valid input: Configure the CC1S bit of the TIM0_CCMR1 register to write 01, the CC1 channel is configured as input, and IC1 is mapped to TI1.
2. Configure the input filter to the required bandwidth based on the signal connected to the timer (in this case, if the input is TI1, configure the IC1F bit ofthe TIM0_CCMR1 register). It is assumed that when the signal edge changes, the input signal jitter occurs within a maximum of 5 internal clock cycles. Therefore, the filter bandwidth needs to be set to more than 5 internal clock cycles, and the jump edge on TI1 can be determined after eight consecutive samples with new levels are detected (sampled at the f_{TIM0_CLK} frequency). Then write 0011 to the IC1F bit ofTIM0_CCMR1.
3. Configure the CC1P and CC1NP bits ofthe TIM0_CCER register to write 0 and select the effective conversion edge rise edge on TI1.
4. Configure the IC1PSC = 00 of the TIM0_CCMR1 register to disable the input predivider and execute the capture operation every time the capture input detects a valid edge.
5. Configure the CC1E = 1 of the TIM0_CCER register to allow the value of the counter to be captured into the capture register.
6. As needed, the input capture 1 interrupt can be enabled by configuring the CC1IE bit of TIM0_DIER to 1, and/or by configuring the CC1DE bit of this register to 1 to enable DMA requests

When input capture occurs:

- When a valid jump edge occurs, the TIM0_CCRx register gets the value of the counter.
- Set the CCxIF flag to 1 (interrupt flag). If at least two consecutive catches occur, but the CCxIF flag is not cleared to zero, then the CCxOF capture overflow flag is set to 1.
- Generates interrupts based on the CCxIE bit.
- Generate DMA requests based on CCxDE bits.

To handle repeated captures, it is recommended to read the data before reading the capture overflow flag. This can avoid the loss of duplicate capture information that may occur after reading the capture overflow flag and before reading the data.

Notes:

1. **IC interrupts and/or DMA requests can be generated by software placing the corresponding CCxG position 1 in the TIM0_EGR register.**
2. **Configure the TIM0_CCMRx register twice: Configure the CCxS bit in the TIM0_CCMRx register for the first time, and select the mapping of the input mode; The ICxF and ICxPSC of the TIM0_CCMRx register are configured for the second time to select filtering and predivision coefficients.**

11.3.10. PWM input mode

This pattern is a special case of the input capture pattern, and the implementation steps are basically the same as the input capture pattern, with the following differences:

- Two ICx signals are mapped to the same TIx input;
- The two ICx signals are effective at the edge, but of opposite polarity;
- Select one of the two TIxFP signals as the trigger input and configure the slave mode controller to reset mode;

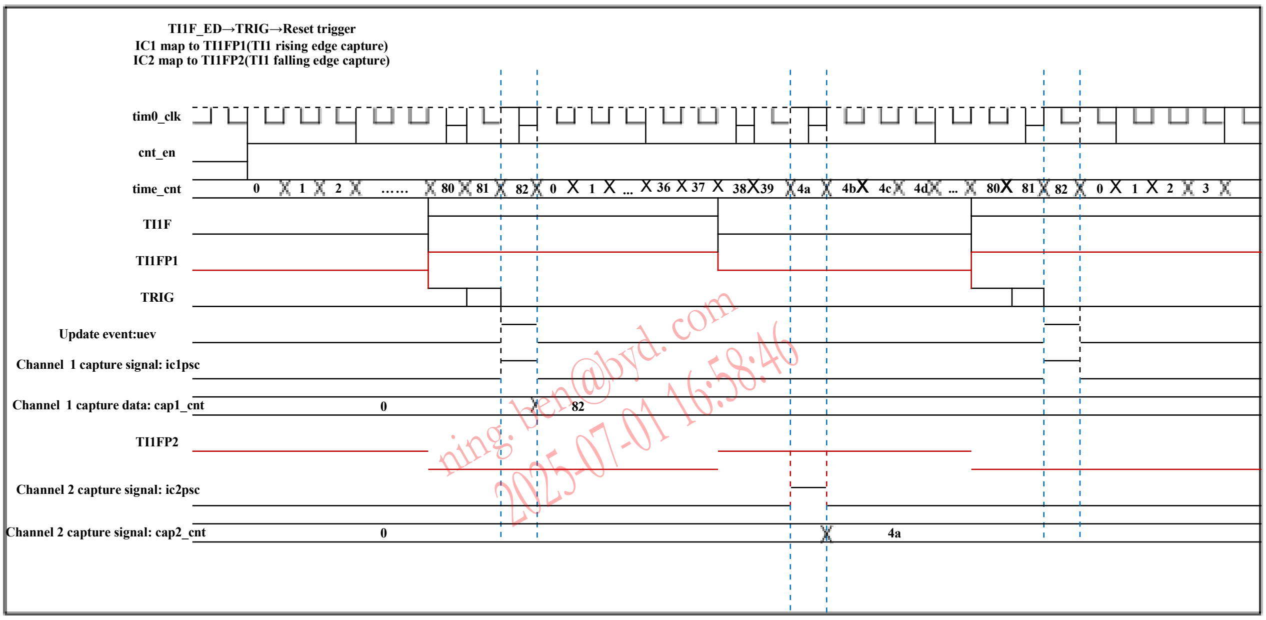

TI1F_ED→TRIG→Reset trigger
IC1 map to TI1FP1(TI1 rising edge capture)
IC2 map to TI1FP2(TI1 falling edge capture)
tim0_clk
cnt_en
time_cnt
0
1
2
......
80
81
82
0
1
...
36
37
38
39
4a
4b
4c
4d
...
80
81
82
0
1
2
3
TI1F
TI1FP1
TRIG
Update event:uev
Channel 1 capture signal: ic1psc
Channel 1 capture data: cap1_cnt
0
82
TI1FP2
Channel 2 capture signal: ic2psc
Channel 2 capture signal: cap2_cnt
0
4a

Figure 11.17 PWM input mode timing

The following example shows measurements corresponding to the PWM period (in the TIM0_CCR1 register) and duty cycle (in the TIM0_CCR2 register) for TI1 (depending on the TIM0_CLK frequency and the value of the predivider):

1. Select the valid input for TIM0_CCR1: Configure CC1S = 01 in the TIM0_CCMR1 register (select TI1), configure the CC1 channel as input, and map IC1 to TI1.
2. Select the effective polarity of TI1FP1 (for capture and counter clearing in TIM0_CCR1): Configure the CC1P bit and CC1NP bit of the TIM0_CCER register to 0 (valid for rising edge).
3. Select the valid input for TIM0_CCR2: Configure CC2S = 10 in the TIM0_CCMR1 register (select TI1), configure the CC2 channel as input, and map IC2 to TI1.
4. Select effective polarity of TI1FP2 (for capture and counter clearing in TIM0_CCR2): Configure the CC2P bit of the TIM0_CCER register to be 1 and the CC2NP bit to be 0 (valid for falling edge).
5. Select a valid trigger input: Configure TS = 00101 in the TIM0_SMCR register (select TI1FP1).
6. Configures the slave mode control register to reset mode: Configure SMS = 0100 in the TIM0_SMCR register.
7. Select the direction of TIM0_CH1: Configure the TIM0_DIR register CH1DIR = 1, and configure TIM0_CH1 as input.
8. Enable capture: Configure the CC1E bit and CC2E bit in the TIM0_CCER register to be 1

11.3.11. Force output mode

In output mode (CCxS bit = 00 in the TIM0_CCMRx register), each output comparison signal (OCxREF and OCx/OCxN) can be forcibly set to an active or Invalid level directly by the software, regardless of any comparison results between the output comparison register and the counter.

To force the output comparison signals (OCxREF and OCx/OCxN) to an active level, the user simply writes 0101 to the OCxM bit in the corresponding TIM0_CCMRx register. OCxREF is then forced to be set to a high level (OCxREF is always valid for high levels), and OCx takes the opposite value of the CCxP polarity bit.
For example: CCxP=0 (the OCx high level is valid) => Forcibly set the OCx high level.

The OCxREF signal can be forcibly set to a low level by writing 0100 to the OCxM bit in the TIM0_CCMRx register.

In any case, the comparison between the TIM0_CCRx shadow register and the counter is still performed, and the flag is allowed to be set to 1. The corresponding interrupt and DMA requests can therefore be sent

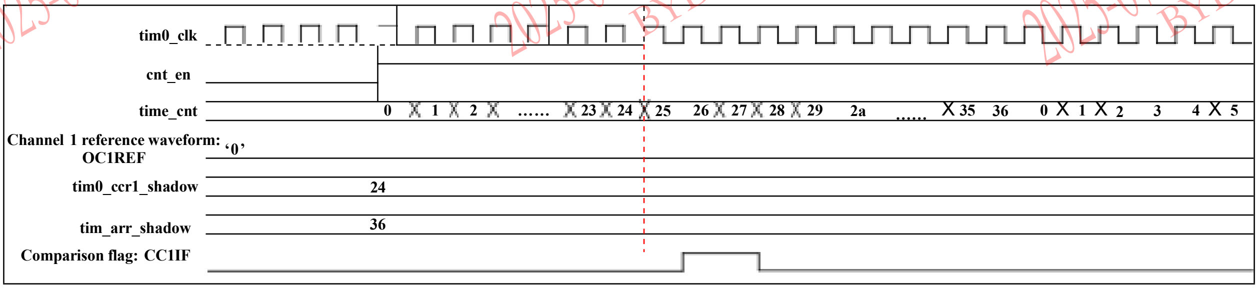


Figure 11.18 Force output Invalid level timing diagram(OCxM=0100)

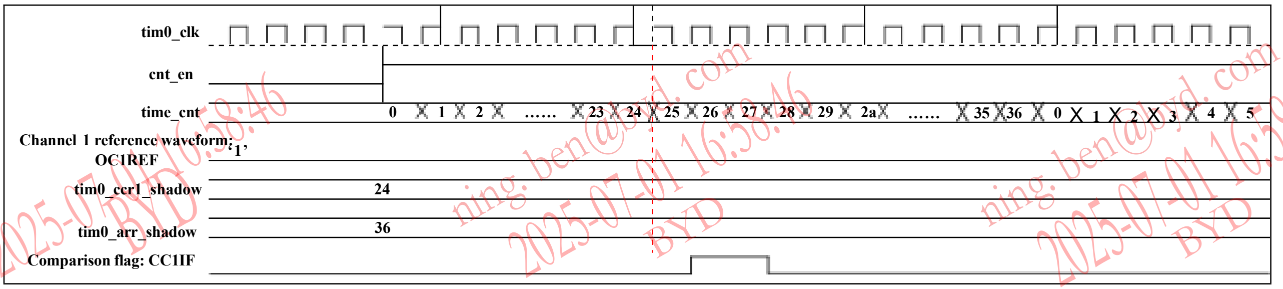


Figure 11.19 Force output active level timing diagram(OCxM=0101)

11.3.12. Output comparison mode

The output comparison mode is used to control the output waveform or indicate that a certain period of time has passed. Channel 1 to channel 4 can be used as the output, and channel 5 and 6 can only be used inside the device (for example, used to produce hybrid waveforms or trigger an ADC).

When there is a match between the capture/compare register and the counter, the output compare function is as follows:

- A programmable value is assigned to the corresponding output pin, defined by the output comparison mode (OCxM bits in the TIM0_CCMRx register) and the output polarity (CCxP bits in the TIM0_CCER register). When the counter value matches the CCR value, the output pin can either hold its level (OCxM=0000), be set to active (OCxM=0001), Invalid (OCxM=0010), or be flipped (OCxM=0011).
- Set the capture/compare interrupt flag to 1 (CCxIF bit).
- If the corresponding interrupt enable bit (CCxIE bit) is set to 1, an interrupt is generated.
- If the corresponding DMA request enable bit (CCxDE bit) is set to 1, a DMA request will be generated.

Using the OCxPE bit in the TIM0_CCMRx register, the TIM0_CCRx register can be configured with or without the preloaded register.

In output comparison mode, updating event UEV has no effect on OCxREF and OCx output. The accuracy of synchronization can reach one count cycle of the counter. Output comparison mode can also be used to output monopulse (in monopulse mode).

Application step:

1. Select counter clock (internal, external, predivider).
2. Write the required data in the TIM0_ARR and TIM0_CCRx registers.
3. If you want to generate an interrupt request, you need to place CCxIE at 1.
4. Select output mode. Such as:
   a) Write OCxM = 0011 and flip the OCx output pin when CNT matches CCRx
   b) Write OCxPE = 0 to disable the preload register
   c) Write CCxP = 0 to select the high level effective polarity
   d) Write CCxE = 1 to enable output
5. Enable the counter by placing the CEN position 1 in the TIM0_CR1 register.

The TIM0_CCRx register can be updated at any time through the software to control the output waveform, provided that the pre-loaded register is not enabled (OCxPE=0, otherwise the TIM0_CCRx shadow register will only be updated when the next update event UEV occurs).

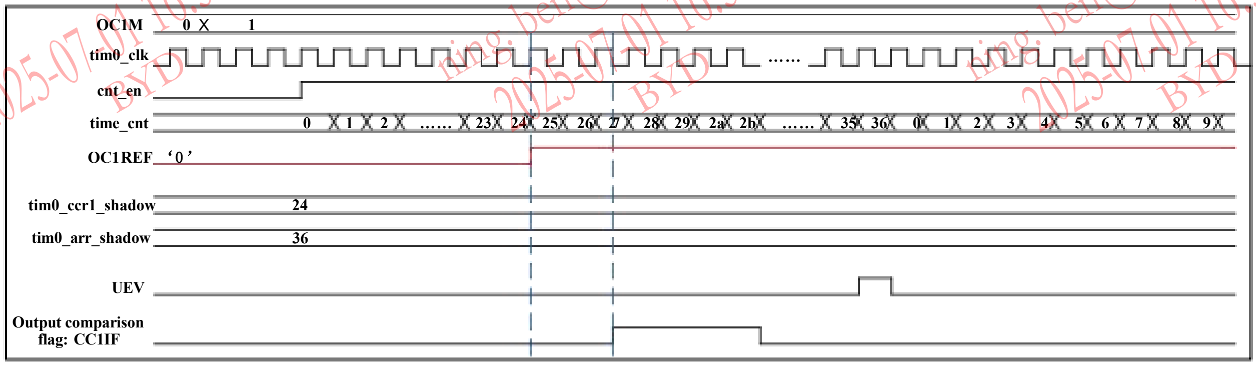

OC1M
0
1
tim0_clk
cnt_en
time_cnt
0 1 2 …… 23 24 25 26 27 28 29 2a 2b …… 35 36 0 1 2 3 4 5 6 7 8 9
OC1REF
'0'
tim0_ccr1_shadow
24
tim0_arr_shadow
36
UEV
Output comparison flag: CC1IF

Figure 11.20 Output comparison mode(OCxM=0001)

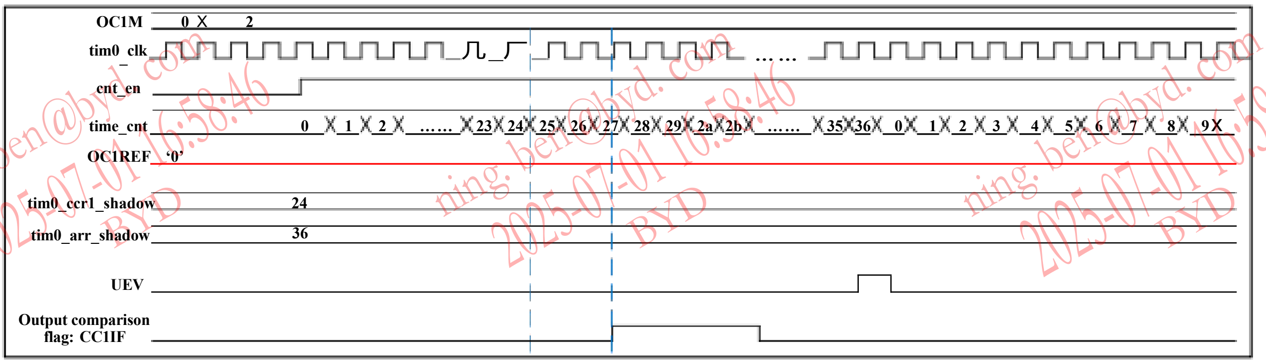

OC1M
0
2
tim0_clk
cnt_en
time_cnt
0 1 2 …… 23 24 25 26 27 28 29 2a 2b …… 35 36 0 1 2 3 4 5 6 7 8 9
OC1REF
'0'
tim0_ccr1_shadow
24
tim0_arr_shadow
36
UEV
Output comparison flag: CC1IF

Figure 11.21 Output comparison mode(OCxM=0010)

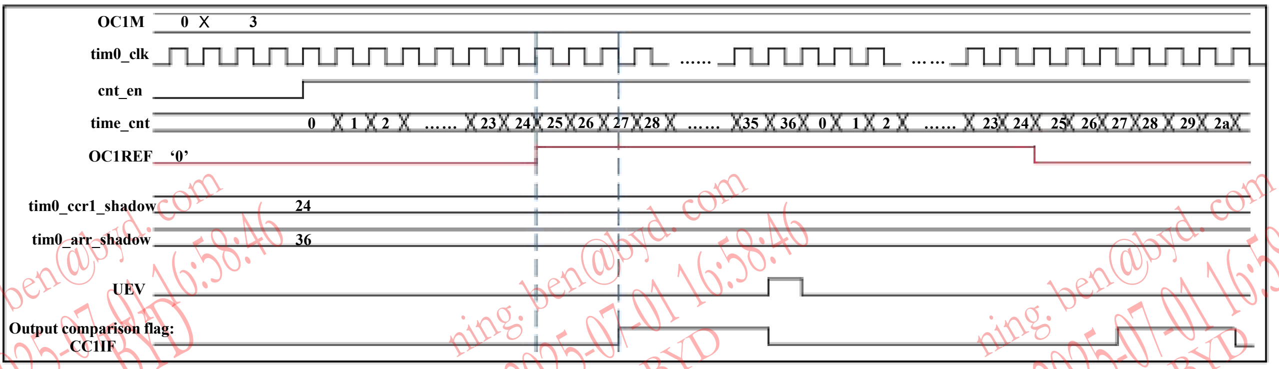

OC1M
0
3
tim0_clk
cnt_en
time_cnt
0 1 2 …… 23 24 25 26 27 28 …… 35 36 0 1 2 …… 23 24 25 26 27 28 29 2a
OC1REF
'0'
tim0_ccr1_shadow
24
tim0_arr_shadow
36
UEV
Output comparison flag: CC1IF

Figure 11.22 Output comparison mode(OCxM=0011)

11.3.13. PWM mode

The PWM mode generates a signal with a programmable frequency and duty cycle determined by the TIM0_ARR register value and the duty cycle determined by the TIM0_CCRx register value.

Each channel can independently select the PWM mode (each OCx output corresponds to a PWM), just Write "0110" (PWM mode 1) or "0111" (PWM mode 2) to the OCxM bit of the TIM0_CCMRx register.The corresponding preloaded register must be enabled by enabling OCxPE position 1 in the TIM0_CCMRx register, and finally by enabling ARPE position 1 in the TIM0_CR1 register to automatically reload the preloaded register (in up-count or center alignment mode).

Since the preloaded registers are only transferred to the shadow registers when an update event occurs, all registers must be initialized by placing UG position 1 in the TIM0_EGR register before starting the counter.

The polarity of OCx can be set by software in the CCxP bit in the TIM0_CCER register. It can be set to high active or low active. OCx output is enabled through a combination of CCxE, CCxNE, MOE, OSSI, and OSSR bits (TIMx_CCER and TIM0_BDTR registers).

PWM edge alignment

Select edge alignment mode when CMS = 00 in the TIM0_CR1 register.

- Incremental count configuration

  Increment counts are performed when DIR = 0 in the TIM0_CR1 register. The following is an example of PWM mode 1. As long as TIM0_CNT < TIM0_CCRx, the PWM reference signal OCxREF is high, otherwise it is low. If the comparison value in TIM0_CCRx is greater than the automatic overload value (in TIM0_ARR), OCxREF remains "1". If the comparison value is 0, OCxREF remains "0".
- Decrement count configuration

  Decrement is performed when DIR = 1 in the TIM0_CR1 register. In PWM mode 1, as long as TIM0_CNT > TIM0_CCRx, the reference signal OCxREF is low, otherwise it is high. If the comparison value in TIM0_CCRx is greater than the automatic overload value in TIM0_ARR, OCxREF remains "1". It is not possible to generate a PWM waveform with a duty cycle of 0% in this mode.

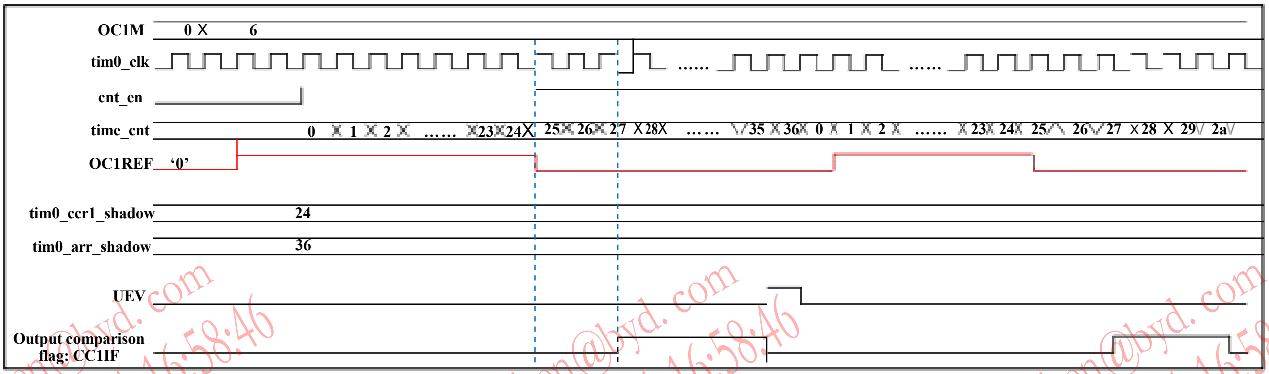

OC1M
0
6
tim0_clk
......
cnt_en
time_cnt
0 1 2 ...... 23 24 25 26 27 28 ...... 35 36 0 1 2 ...... 23 24 25 26 27 28 29 2a
OC1REF
'0'
tim0_ccr1_shadow
24
tim0_arr_shadow
36
UEV
Output comparison flag: CC1IF

Figure 11.23 PWM edge alignment mode 1(OCxM=0110)

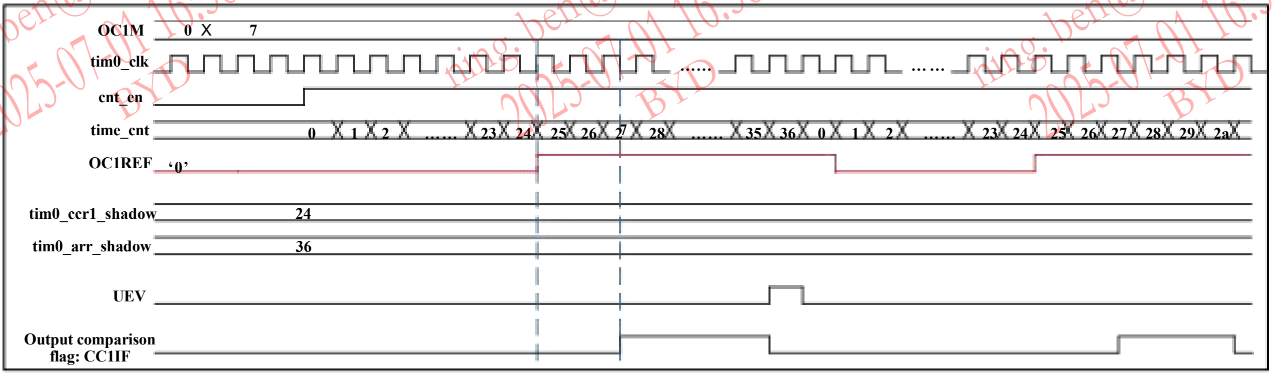

OC1M
0 X 7
tim0_clk
......
......
cnt_en
time_cnt
0 1 2 ...... 23 24 25 26 27 28 ...... 35 36 0 1 2 ...... 23 24 25 26 27 28 29 2a
OC1REF
'0'
tim0_ccr1_shadow
24
tim0_arr_shadow
36
UEV
Output comparison flag: CC1IF

Figure 11.24 PWM edge alignment mode 2(OCxM=0111)

PWM center alignment

When the CMS bit in the TIM0_CR1 register is not "00" (all other configurations have the same effect on the OCxREF/OCx signal), the center alignment mode takes effect. Depending on the configuration of the CMS bits, the comparison flag can be set to 1 when the counter is incremented, decrement, or both. The direction bit (DIR) in the TIM0_CR1 register is updated by hardware and cannot be changed by software.

Center alignment mode use recommendations:

- The current increment/decrement count configuration is used when starting Center alignment mode. This means that the counter increments or deciles the count based on the value written to the DIR bit in the TIM0_CR1 register.
- The safest way to use the center alignment mode is to generate a software update before starting the counter (to UG position 1 in the TIM0_EGR register) and not to write to the counter while it is running.

11.3.14. Asymmetric PWM mode

In asymmetric mode, a programmable phase shift is allowed between the two center-aligned PWM signals generated. The frequency is determined by the value of the TIM0_ARR register, while the duty cycle and phase shift are determined by a pair of TIM0_CCRx registers. Two registers control the PWM during increment count and decrement count, so that the PWM is adjusted once every half PWM cycle:

- OC1REFC (or OC2REFC) is controlled by TIM0_CCR1 and TIM0_CCR2
- OC3REFC (or OC4REFC) is controlled by TIM0_CCR3 and TIM0_CCR4

The two channels can independently select the asymmetric PWM mode (one OCx output per pair ofCCR registers) by writing "1110" (asymmetric PWM mode 1) or "1111" (asymmetric PWM mode 2) to the OCxM bit of the TIM0_CCMRx register.

When a given channel is used as an asymmetric PWM channel, its complementary channels can also be used. For example, if an OC1REFC signal is generated on channel 1 (asymmetric PWM mode 1), then either an OC2REF signal or an OC2REFC signal can be output on channel 2 due to asymmetric PWM mode 1.

11.3.15. Combined PWM mode

In combined PWM mode, programmable delay and phase shift are allowed between individual pulses ofthe generated two edge-aligned or center-aligned PWM signals. The frequency is determined by the value of the TIM0_ARR register, while the duty cycle and delay are determined by the two TIM0_CCRx registers. The resulting signal OCxREFC consists of two logic or operations that reference PWM or a combination of logic and operations.

- OC1REFC (or OC2REFC) is controlled by TIM0_CCR1 and TIM0_CCR2
- OC3REFC (or OC4REFC) is controlled by TIM0_CCR3 and TIM0_CCR4

The two channels can independently select the combined PWM mode (one OCx output per pair ofCCR registers) by writing "1100" (combined PWM mode 1) or "1101" (combined PWM mode 2) to the OCxM bit of the TIM0_CCMRx register.When a given channel is used as a combined PWM channel, its complementary channels must be configured in opposite PWM modes (for example, one channel is configured in combined PWM mode 1 and the other in combined PWM mode 2).

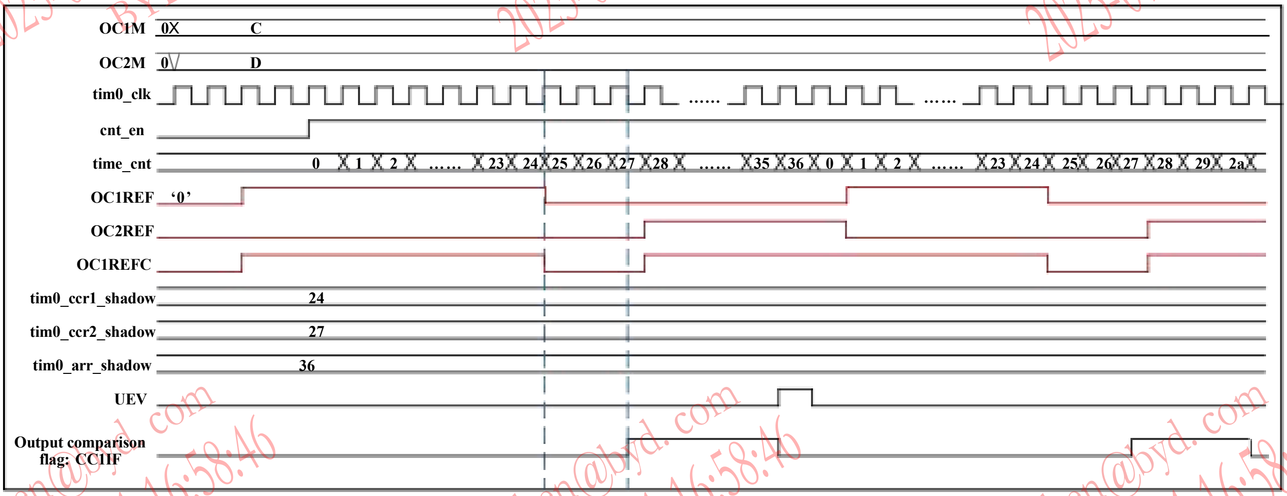

OC1M 0X C
OC2M 0 D
tim0_clk
cnt_en
time_cnt 0 1 2 ...... 23 24 25 26 27 28 ...... 35 36 0 1 2 ...... 23 24 25 26 27 28 29 2a
OC1REF '0'
OC2REF
OC1REFC
tim0_ccr1_shadow 24
tim0_ccr2_shadow 27
tim0_arr_shadow 36
UEV
Output comparison flag: CC1IF

Figure 11.25 Combined PWM mode 1(OCxM=1100)

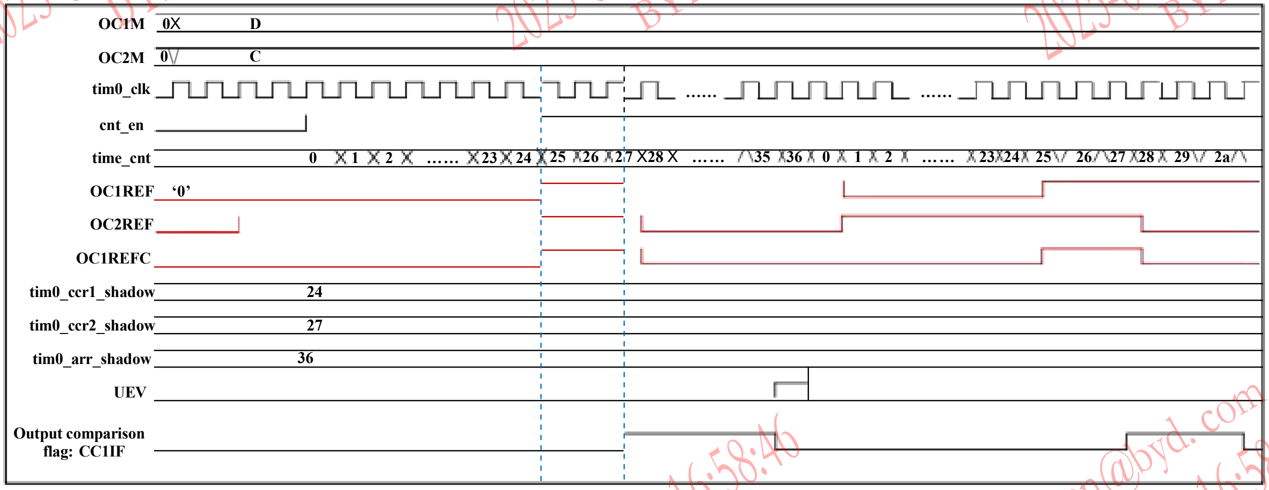

OC1M 0X D
OC2M 0 C
tim0_clk
cnt_en
time_cnt 0 1 2 ...... 23 24 25 26 27 28 ...... 35 36 0 1 2 ...... 23 24 25 26 27 28 29 2a
OC1REF '0'
OC2REF
OC1REFC
tim0_ccr1_shadow 24
tim0_ccr2_shadow 27
tim0_arr_shadow 36
UEV
Output comparison flag: CC1IF

Figure 11.26 Combined PWM Mode 2(OCxM=1101)

11.3.16. Combined Three-phase Mode

In the combined three-phase PWM mode, the resulting one to three center-aligned PWM signals and a programmable signal allow logic and operations in the middle of the pulse. The OC5REF signal is used to define the resulting combined signal. With the 3-bit GC5C[3: 1] in TIM0_CCR5, you can choose which reference signal the OC5REF is combined with. The resulting signal OCxREFC consists of a combination of two logic and operations that reference PWM.

- If GC5C1 is set to 1, OC1REFC is controlled by TIM0_CCR1 and TIM0_CCR5
- If GC5C2 is set to 1, OC2REFC is controlled by TIM0_CCR2 and TIM0_CCR5
- If GC5C3 is set to 1, OC3REFC is controlled by TIM0_CCR3 and TIM0_CCR5

The combined three-phase PWM mode can be selected independently from channel 1 to channel 3 by placing at least one of the 3-bit GC5C[3: 1] in position 1.

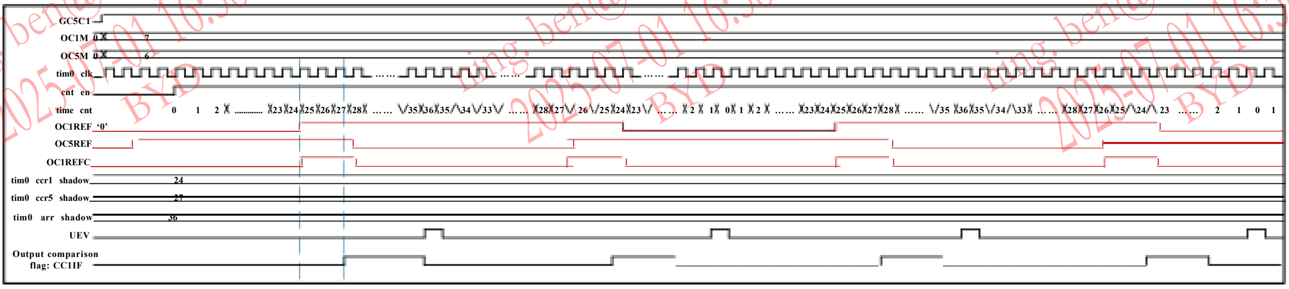


Figure 11.27 Combined three-phase PWM mode(GC5C1=1)

11.3.17. One pulse mode

One pulse mode(OPM)is a special case of the above mode. In this mode, the counter can be activated at the trigger of an excitation signal and can produce a pulse with programmable pulse width after a programmable delay.

Counters can be started from the mode controller. Waveforms can be generated in output comparison mode or PWM mode. Select monopulse mode by placing the OPM position in the TIM0_CR1 register. In this way, the counter will automatically stop when the next update event UEV occurs.

A pulse can be generated correctly only if the comparison value is different from the initial counter value. Before starting the timer (when the timer is waiting for triggering), you must perform the following configurations:

- Incremental count time: CNT<CCRx<=ARR(Special attention,0<CCRx)
- Decrement count time: CNT>CCRx

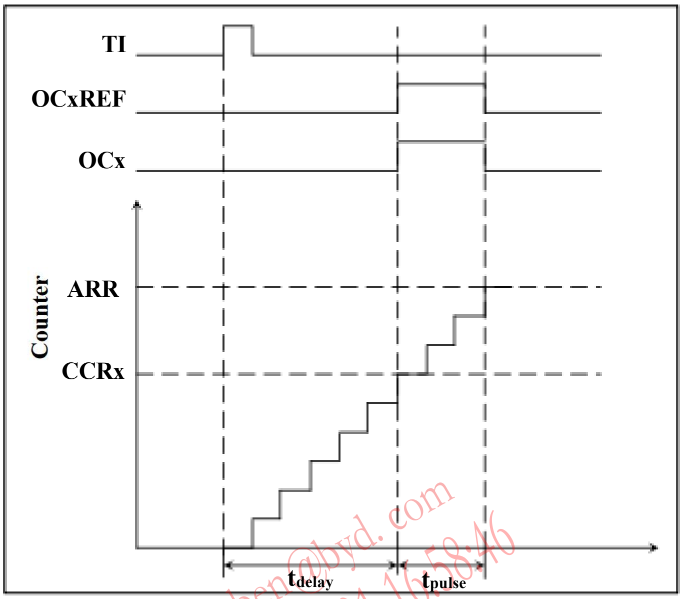


Figure 11.28 One pulse mode

Special case: The OCx function is quickly enabled:

In monopulse mode, edge detection ofthe TIx input sets the CEN position to 1, indicating enable counter. The output is then flipped when a comparison occurs between the counter value and the comparison value. However, completing these operations requires multiple clock cycles, which limits the minimum possible delay (t_{DELAY} minimum).

If you want to output the waveform with minimal delay, you can put the OCxFE position 1 in the TIM0_CCMRx register. This forces OCxRef (and OCx) to respond to the excitation signal regardless of the comparison. Its new level is the same as when the comparison match occurred. OCxFE works only when the channel is configured in PWM1 or PWM2 mode.

11.3.18. Retrigger One pulse mode(OPM)

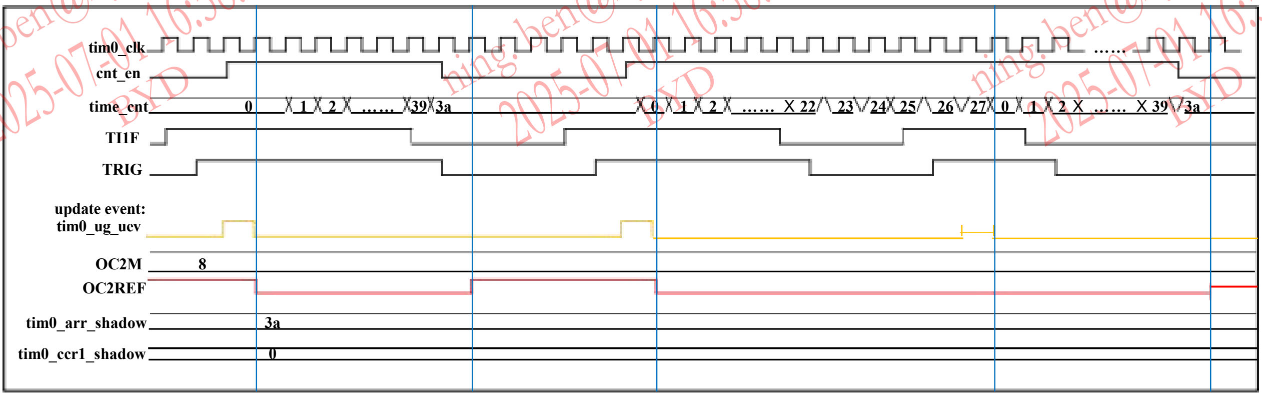
This mode allows the counter to be activated at the trigger of an excitation signal and can produce pulses of programmable length, but with the following differences from the non-reflexible monopulse mode:

- When triggered, the pulse is generated immediately (no programmable delay)
- If a new trigger occurs before the previous trigger completes, the pulse will be extended

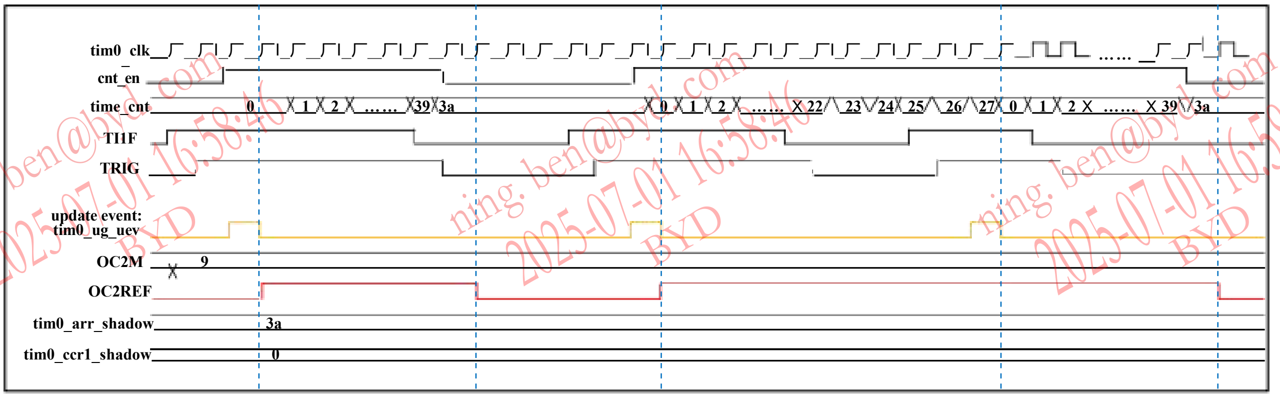
Timer must be in slave mode, bit SMS[3: 0] in the TIM0_SMCR register = "1000" (combined reset + trigger mode), OCxM[3: 0] bit set to "1000" or "1001" for retrigger OPM mode 1 or Mode 2.

When the timer is configured in incremental count mode, the corresponding CCRx must be set to 0 (the ARR register sets the pulse length). If the timer is configured in decreasing count mode, CCRx must be higher than or equal to ARR.

Notes: This mode cannot be used with the center-aligned PWM mode. CMS[1: 0]=00 must be set in TIM0_CR1.



Retrigger OPM Mode 1(OCxM=1000)



Retrigger OPM Mode 2(OCxM=1001)

11.3.19. Complementary output and dead-zone insertion

Advanced control timer (TIM0) can output two complementary signals and manage the turn-off and turn-on moments ofthe output. This period is often referred to as the dead time and must be adjusted according to the device connected to the output and its characteristics(the inherent delay of the level converter、 the delay generated by the switching device).

Complementary signals OCx and OCxN are activated by a combination of the following control bits: the CCxE and CCxNE bits in the TIM0_CCER register and the MOE, OISx, OISxN, OSSI, and OSSR bits in the TIM0_BDTR and TIM0_CR2 registers.

Dead-zone insertion is enabled when both CCxE and CCxNE bits are set to 1 and MOE position 1 (if brakes are present). Each channel has a 10-bit dead zone generator. Two output OCx and OCxN are generated based on the reference waveform OCxREF. Valid if OCx and OCxN are high:

- The output signal OCx is the same as the reference signal, except that its rising edge is delayed relative to the reference rising edge.
- The output signal OCxN is opposite to the reference signal, and its rising edge is delayed relative to the reference falling edge.

If the delay time is greater than the width of the effective output (OCx or OCxN), no corresponding pulse is generated.

The following figure shows the relationship between the output signal of the dead-zone generator and the reference signal OCxREF,Suppose CCxP = 0,CCxNP = 0,MOE = 1,CCxE = 1,CCxNE = 1.

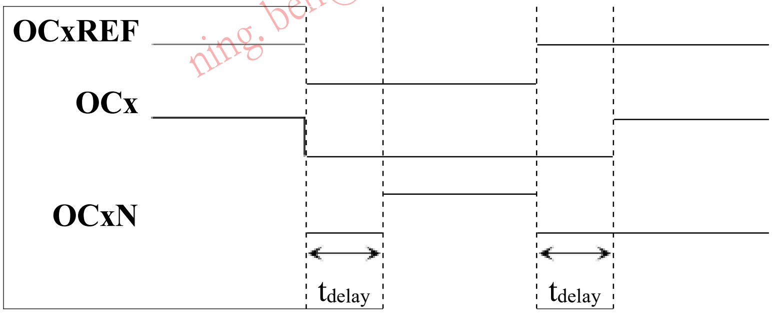


Figure 11.29 Complementary output with dead-zone insertion

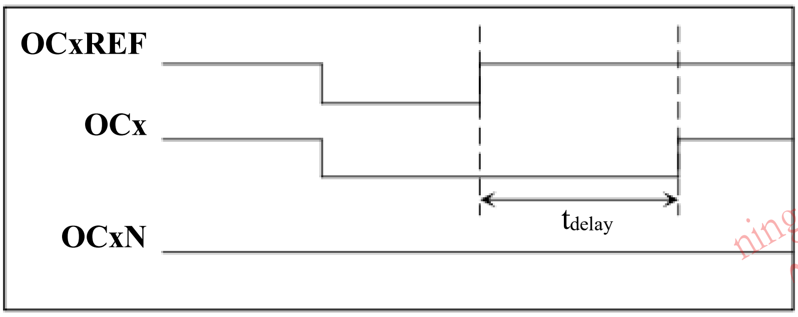


Figure 11.30 Dead-time waveform with delay time greater than negative pulse width

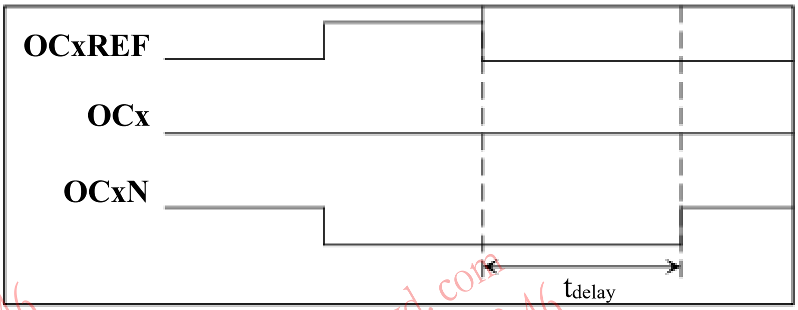


Figure 11.31 Dead-time waveform with delay time greater than positive pulse width

The dead-time delay is the same for all channels and can be programmed with the DTG bits in the TIM0_BDTR register.

Redirect OCxREF to OCx or OCxN

In output mode (forced output mode, output compare mode, or PWM mode), the OCxREF can be redirected to the OCx output or OCxN output by configuring the CCxE and CCxNE bits in the TIM0_CCER register.

With this feature, you can send a specific waveform (such as PWM or static active level) on one output while leaving the complementary output at its Invalid level. Alternatively, make both outputs Invalid at the same time, or make both outputs valid at the same time, both complementary and with dead-zone.

Notes: If only OCxN is enabled (CCxE=0, CCxNE=1), the two are not complementary. Once OCxREF is high, OCxN becomes effective. For example, if CCxNP=0, then OCxN=OCxRef. On the other hand, if both OCx and OCxN are enabled (CCxE=CCxNE=1), OCx becomes effective at high OCxREF, while OCxN is complementary and effective at low OCxREF.

11.3.20. Use the brake function

The purpose of the braking function is to protect the power switch driven by the PWM signal generated by the TIM0 timer. The two brake inputs are usually connected to the power stage and the fault output of the three-phase inverter. When activated, the brake circuit turns off the PWM output and forces it to a predefined safety state.

There are two brake channels. Brake channel sources include only application faults (from input pins) and can force the output to a predefined level (valid or Invalid) after the dead zone duration. The brake 2 channel includes only application failures and can force the output to Invalid.

The output enable signal and output level during braking depend on multiple control bits:

- The MOE bit in the TIM0_BDTR register, which allows the output to be enabled/disabled by software, to reset in the event of brake and brake 2 events.
- The OSSI bit in the TIM0_BDTR register that defines whether the timer will hold the output in Invalid state or release control of the GPIO controller (usually leaving it in high resistance mode)
- The OISx and OISxN bits in the TIM0_CR2 register set the output to the off level (valid or Invalid). Cannot both OCx and OCxN outputs be set to active at a given time, regardless of the OISx and OISxN values.

The source of the brake (BRK) channel is an external source (set by the AF1 controller) connected to the BKIN pin, with polarity selection and optional digital filtering.

The brake 2 (BRK2) source is an external source (set by the AF2 controller) connected to the BK2IN pin, with polarity selection and optional digital filtering.

Brake events can also be generated by software via BG and B2G bits in the TIM0_EGR register. Regardless of the value of the BKE and BK2E enable bits, the brake can be generated by software using BG and B2G.

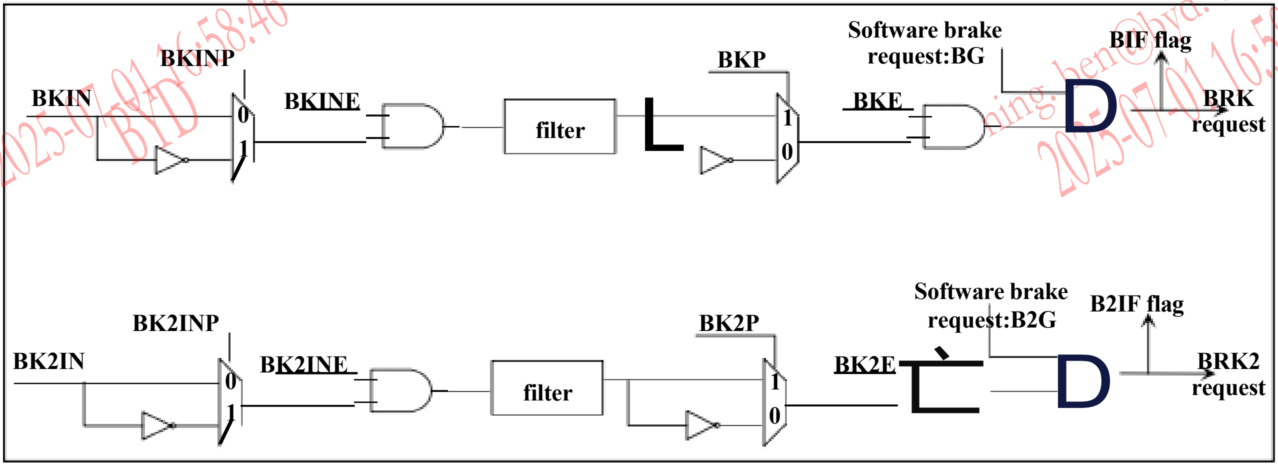


Figure 11.32 Brake and brake 2 circuits

When one of the brakes occurs (the selected level appears on one of the brake inputs):

- MOE bit asynchronously clear, leaving output in Invalid, idle or even releasing control to GPIO controller (selected by OSSI bit). This feature is enabled even if the MCU oscillator is turned off.

- When MOE=0, each output channel is driven at a level programmed in the OISx bit of the TIM0_CR2 register. If OSSI=0, the timer releases the output control (which is taken over by the GPIO controller), otherwise enables the output to remain high.
- When using complementary output:
  - ○ The output is first placed in the Invalid state (depending on polarity). This is an asynchronous operation, so it works even if no clock is provided for the timer.
  - ○ If the timer clock is still present, the dead zone generator is reactivated and the output is driven at a level programmed in OISx and OISxN bits after the dead zone. Even in this case, OCx and OCxN cannot be driven to their active levels at the same time.

    Note that the MOE resynchronizes, so the dead zone lasts a little longer than usual.
  - ○ If OSSI=0, the timer releases the output control (taken over by the GPIO controller that forces the high resistance state), otherwise the enable output will remain high or immediately change to high when one of the CCxE or CCxNE bits is high.
- Set the brake status flag (BIF and B2IF bits in the TIM0_SR register) to 1. If BIE position 1 is in the TIM0_DIER register, an interrupt is generated.
- If the AOE position in the TIM0_BDTR register is 1, the MOE bit is automatically set to 1 again when the next update event (UEV) occurs.

Notes:

1. **Brake input is level active. Therefore, the MOE position 1 cannot be changed (automatically or via software) when the braking input is active. At the same time, the status flags BIF and B2IF cannot be cleared.**
2. **When CCxE and CCxNE are both 0, the brake Invalid is valid, both OCx and OCxN disable output, and the I/O port is released to the GPIO controller.**

The following figure shows an example of the output in response to braking:

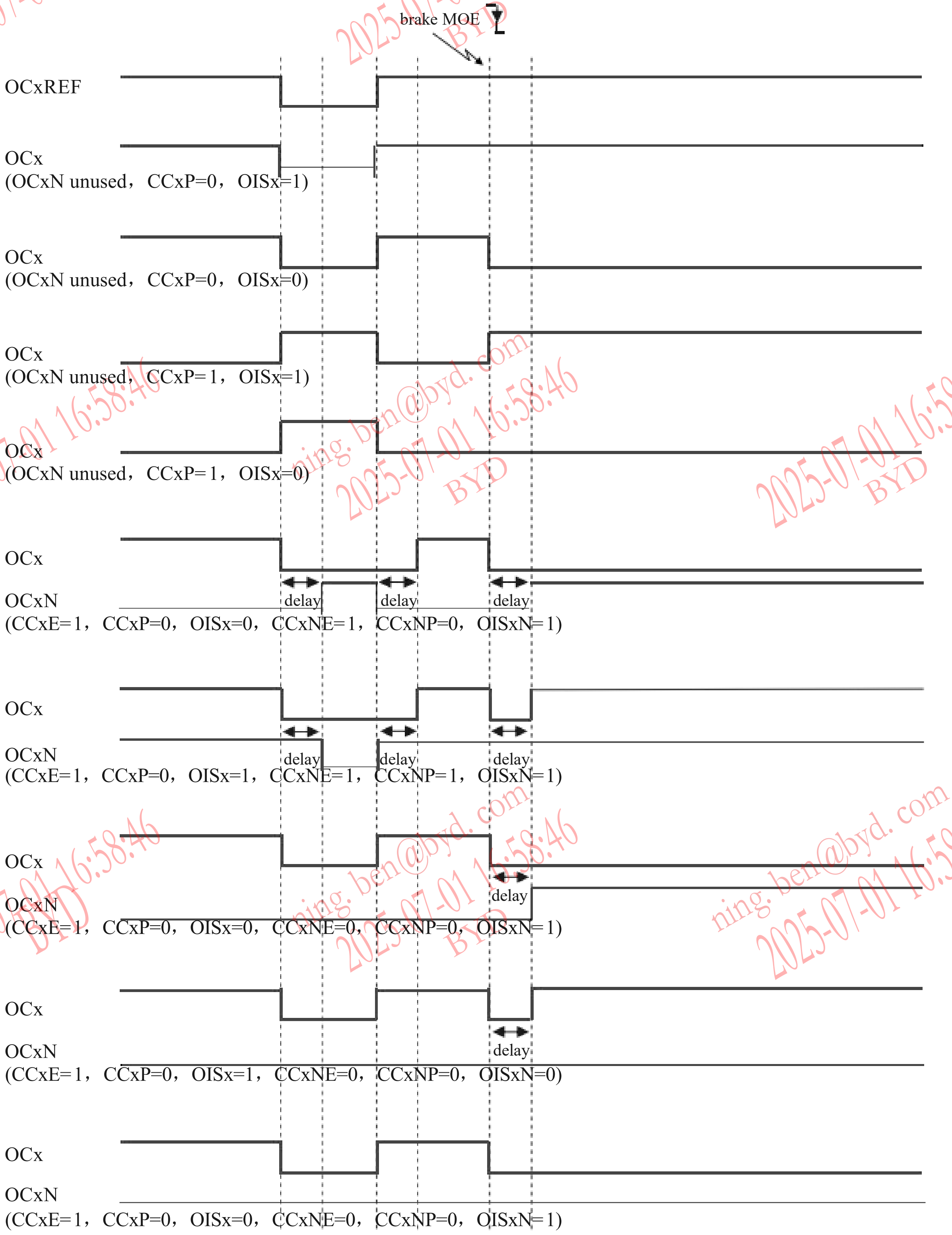

brake MOE
OCxREF
OCx
(OCxN unused，CCxP=0，OISx=1)
OCx
(OCxN unused，CCxP=0，OISx=0)
OCx
(OCxN unused，CCxP=1，OISx=1)
OCx
(OCxN unused，CCxP=1，OISx=0)
OCx
OCxN
delay
delay
delay
(CCxE=1，CCxP=0，OISx=0，CCxNE=1，CCxNP=0，OISxN=1)
OCx
OCxN
delay
delay
delay
(CCxE=1，CCxP=0，OISx=1，CCxNE=1，CCxNP=1，OISxN=1)
OCx
OCxN
delay
(CCxE=1，CCxP=0，OISx=0，CCxNE=0，CCxNP=0，OISxN=1)
OCx
OCxN
delay
(CCxE=1，CCxP=0，OISx=1，CCxNE=0，CCxNP=0，OISxN=0)
OCx
OCxN
(CCxE=1，CCxP=0，OISx=0，CCxNE=0，CCxNP=0，OISxN=1)

Figure 11.33 Output of brake request

Output control bits of complementary channels OCx and OCxN with brake function

Control bit					Output state	
MOE	OSSI	OSSR	CCxE	CCxNE	OCx Output state	OCxN Output state
1	x	x	0	0	Output disable (not driven by timer: high resistance) OCx=0、OCxN=0	
		0	0	1	Output disable (not driven by timer: high resistance) OCx=0	OCxREF + polarity OCxN = OCxREF xor CCxNP
		0	1	0	OCxREF + polarity OCx = OCxREF xor CCxP	Output disable (not driven by timer: high resistance) OCxN=0
		x	1	1	OCxREF + Polarity + Dead zone	OCxREF Inverting + Polarity + Dead zone
		1	0	1	Closed (output enabled and the Invalid level) OCx = OCxP	OCxREF + Polarity OCx = OCxREF xor CCxNP
		1	1	0	OCxREF + Polarity OCx = OCxREF xor CCxP	Closed (output enabled and the Invalid level) OCxN = OCxNP
0	0	x	0	0	Output disable (not driven by timer). The output state is defined by the GPIO controller and can be high, low, or high resistance	
			0	1		
			1	0		
			1	1		
	1		0	0	Closed (output for Invalid state) Asynchronous: OCx=CCxP、OCxN=CCxNP(If BRK or BRK2 is triggered). Subsequently (effective only if BRK is triggered),If a clock exists: OCx=OISx and OCxN=OISxN after the dead zone, assuming that OISx and OISxN are not both set to OCx and OCxN active levels (otherwise a short circuit may be caused when driving the switch in a half-bridge configuration).	
			0	1		
			1	0		
			1	1		

In addition to brake input and output management, write protection is implemented inside the brake circuit to protect the safety of the application. With this feature, users can freeze multiple parameter configurations (dead zone duration, OCx/OCxN polarity and status when disabled, OCxM configuration, brake enable and polarity). Applications can choose from three protection levels via the LOCK bit in the TIM0_BDTR register. After the MCU resets, only one write operation can be performed on the LOCK bit.

The two brake inputs have different behaviors for the timer output:

- The BRK input can disable the PWM output (Invalid state) or force the PWM output to a predefined security state.
- BRK2 can only disable (Invalid state) PWM output.

The BRK input takes precedence over the BRK2 input,shown in the following table:

BRK	BRK2	Timer output state	Typical use cases	
			OCxN output (Lower arm switch)	OCx output (upper arm switch)
Valid	x	Invalid. Then force to output state (after a dead time) If OSSI=0, output is disabled (GPIO logic takes over control)	Start after dead zone insertion	Closed
Invalid	Valid	Invalid	Closed	Closed

11.3.21. Two-way brake input

The TIM0 has bidirectional brake I/O and uses the BKBID and BK2BID bits ofthe TIM0_BDTR register to configure the brake and brake 2 inputs in bidirectional mode. You can use the LOCK bit in the TIM0_BDTR register to lock the BKBID programming bit into Read only mode (lock level 1 or higher).

Bidirectional mode can be used for brake and brake-2 inputs, requiring I/O to be configured as an open drain mode with low effective polarity (configured using BKINP, BKP, BK2INP, and BK2P bits). Any brake request from a system (such as CSS), an on-chip peripheral, or a brake input forces the brake input to be set to a low level to notify that a failure event has occurred. If the polarity bit (high level effective polarity) is not set correctly, bidirectional mode is prohibited for safety purposes.

Software brake events (BG and B2G) also cause the brake I/O to be forced to "0", which indicates to the external component that the timer has entered the brake state, but only if the brake is enabled (BK(2)E= 1). When a software brake event is generated and BK(2)E=0, the output is placed in a safe state and the brake flag is set to 1, but there is no effect on brake (2) I/O.

When BKDSRM(BK2DSRM) is at position 1, the brake output is released to clear the fault signal, thus enabling the system to regain protection.

The brake protection circuit cannot be disabled under the following conditions:

- The brake input path is always valid: The braking event is still in effect even ifBKDSRM (BK2DSRM) is in position 1 and the leak control is released. This prevents the PWM output from restarting during braking.
- After the output (MOE position 1) is enabled, the BK(2)DSRM bit cannot remove the brake protection

Notes: When configuring registers, the low level effective polarity should be configured first, that is, the BKINP, BKP, BK2INP, and BK2P bits should be configured first; Then configure the BKBID and BK2BID bits ofthe TIM0_BDTR register to configure the brake and brake 2 inputs in bidirectional mode. If the I/O is not of a low effective polarity, the BKBID and BK2BID bit configurations ofthe TIM0_BDTR register cannot be written in, and cannot be configured in bidirectional mode.

11.3.21.1. Start and restart the brake circuit

By default (peripherals reset configuration) will start the brake circuit (in input or double mode).

After a brake (brake 2) event, the protection must be restarted by following these steps:

- The BKDSRM(BK2DSRM) must be placed at position 1 to release the output control
- The software must poll the BKDSRM (BK2DSRM) bit until the bit is cleared by the hardware (when the applied brake condition disappears)

The brake circuit is then activated and the PWM output can be re-enabled by placing the MOE at position 1.

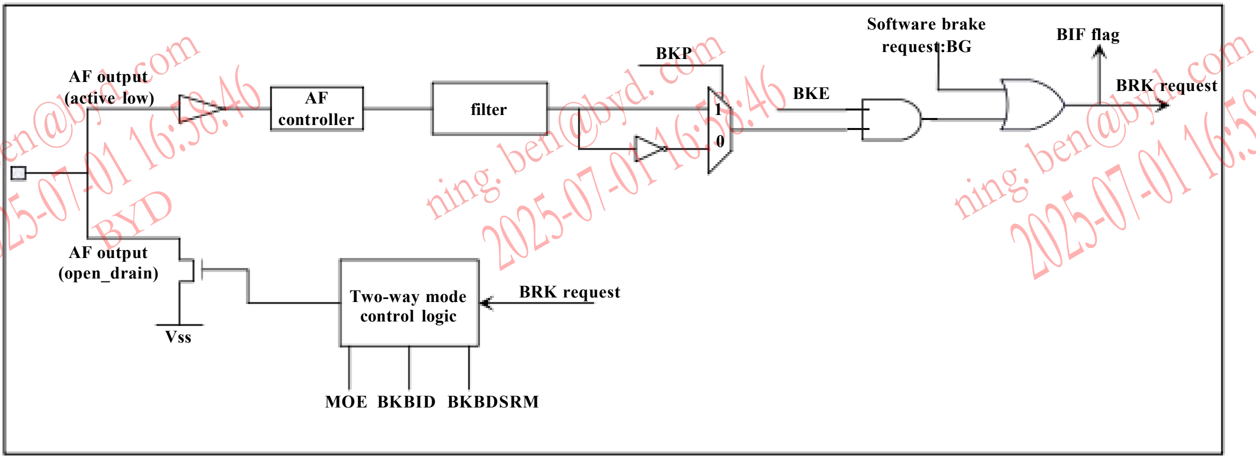


Figure 11.34 Output redirection (BRK2 request not shown in figure)

11.3.22. Clear the OCxREF signal when an external event occurs

For a given channel, applying a high level (corresponding to OCxCE enable position 1 in the TIM0_CCMRx register) to the ocref_clr_int (ETR after ETRF filter) input clears the OCxREF signal to zero. The OCxREF signal will remain low until the next update event (UEV) occurs. This function can only be used in output comparison mode and PWM mode, and does not work in forced mode.

When OCxCE is enabled, the ETR must be configured as follows:

- The external trigger predivider must be turned off: TIM0_SMCR register ETPS[1: 0] position "00".
- External clock mode 2 must be disabled: ECE position "0" in the TIM0_SMCR register.
- External trigger polarity (ETP) and external trigger filter (ETF) can be configured as required.

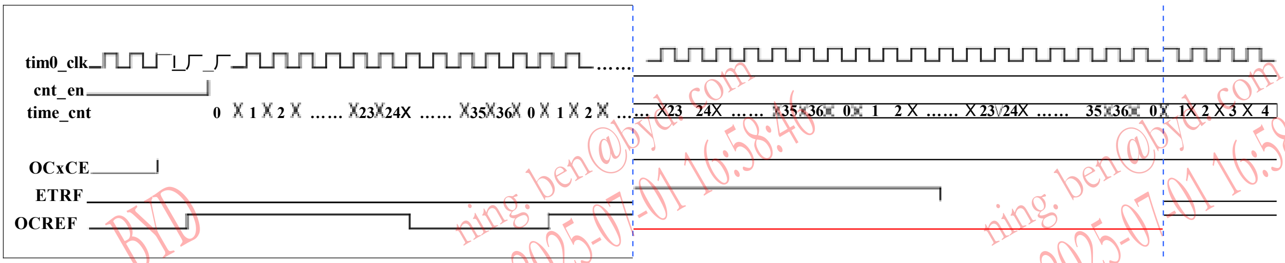


Figure 11.35 Clear the OCxREF of TIMER0

11.3.23. Generate the six-step PWM

When channels use complementary outputs, preloaded load bits are provided on OCxM, CCxE, and CCxNE bits. When a COM reversing event occurs, these preloaded bits are transferred to shadow bits. Therefore, the next configuration can be pre-programmed and the configuration of all channels can be changed at the same time. COM can be generated by software by placing COMG position 1 in the TIM0_EGR register, or by hardware at the rising edge of TRGI.

When a COM event occurs, a flag bit (the COMIF bit in the TIM0_SR register) will be set to 1. At this point, if the COMIE position 1 is in the TIM0_DIER register, an interrupt will be generated; If COMDE position 1 is in the TIM0_DIER register, a DMA request will be generated.

11.3.24. Encoder interface mode

When selecting encoder interface mode, Write SMS= "0001" in the TIM0_SMCR register if the counter counts only at TI1 edges; If the counter counts only at the edge of TI2, Write SMS= "0010"; If the counter counts at both TI1 and TI2 edges, Write SMS= "0011".

Select TI1 and TI2 polarity by programming the CC1P and CC2P bits ofthe TIM0_CCER register. If necessary, the input filter can also be programmed. CC1NP and CC2NP must be kept low.

Two inputs TI1 and TI2 are used to connect the orthogonal encoder. If the counter is enabled (Write " 1" in the CEN bit of the TIM0_CR1 register), the clock for the counter is provided by each active signal conversion on TI1FP1 or TI2FP2. TI1FP1 and TI2FP2 are the signals ofTI1 and TI2 after input filter and polarity selection, if not filtered and reversed, TI1FP1=TI1, TI2FP2=TI2. Counting pulses and directional signals are generated based on a sequence of signal conversions from the two inputs. According to the signal conversion sequence, the counter increments or deciles the count accordingly, and the hardware modifies the DIR bit of the TIM0_CR1 register accordingly. The DIR bit is calculated when any input (TI1 or TI2) undergoes a signal conversion, whether the counter counts only at the edge of TI1 or TI2, or at both TI1 and TI2.

The encoder interface mode is equivalent to an external clock with direction selection. This means that the counter only counts continuously between 0 and the automatically overloaded value in the TIM0_ARR register (increasing the count from 0 to ARR, or decreasing the count from ARR to 0, depending on the specific direction). Therefore, TIM0_ARR must be configured before startup. Again, the capture, compare, repeat counter, and trigger output functions continue to work properly. Encoder mode and external clock mode 2 are incompatible and therefore cannot be selected at the same time.

Notes: When encoder mode is enabled, the predivider must be set to zero.

In this mode, the counter is automatically modified according to the speed and direction of the orthogonal encoder, so its contents always represent the position of the encoder. The counting direction corresponds to the rotation direction of the connected sensor. The following table summarizes the possible combinations (assuming TI1 and TI2 do not switch at the same time).

Effective edge	Level of the relative signal (TI1FP1 corresponds to TI2, TI2FP2 corresponds to TI1)	TI1FP1 signal		TI2FP2 signal	
		Rising	Falling	Rising	Falling
Count only in TI1	High	Count down	Count up	No count	No count
	Low	Count up	Count down	No count	No count
Count only in TI2	High	No count	No count	Count up	Count down
	Low	No count	No count	Count down	Count up
Count on TI1 and TI2	High	Count down	Count up	Count up	Count down
	Low	Count up	Count down	Count down	Count up

11.3.25. UIF remapping

The UIFREMAP bit in the TIM0_CR1 register forces the update interrupt flag UIF to be continuously copied to bit 31(TIM0_CNT[31]) in the timer counter register. This automatically reads the counter value and the potential reversal condition emitted by the UIFCPY flag. In certain cases, this simplifies calculations by avoiding race conditions when processing is shared between background tasks (counter Read) and interrupts (update interrupts). There is no delay between enabling the UIF and UIFCPY flags.

11.3.26. Timer input XOR function

The three input pins TIM0_CH1 to TIM0_CH3 are combined by connecting the input filter of channel 1 to the output of the XOR gate through the TI1S bit in the TIM0_CR2 register. Xor outputs can be used with all timer input functions such as trigger or input capture.

11.3.27. Connect the hall sensor

The connection to the Hall sensor can be achieved through an advanced control timer (TIM0) for generating the motor driven PWM signal, and another timer, TIM6, called an "interface timer". The three timer input pins (CC1, CC2, and CC3) are connected to the TI1 input channel via an XOR gate (selected by placing the TI1S position 1 in the TIM6_CR2 register) and are captured by an "interface timer".

The slave mode controller is configured to reset mode; The slave input is TI1F_ED. This way, whenever one of the three inputs is switched, the counter is re-counted from 0. This will produce a time base triggered by any change in Hall's input.

On the "Interface Timer", capture/compare channel 1 is configured in capture mode and the capture signal is TRC. The captured value corresponds to the interval between the last two changes in the input and provides information related to the motor speed.

The "Interface timer" can be used to generate pulses in output mode to change the configuration of the individual channels ofthe Advanced Control Timer (TIM0) by triggering a COM event. The TIM0 timer is used to generate the motor drive PWM signal. For this, the interface timer channel must be programmed to produce a positive pulse after a programmed delay (in output comparison or PWM mode). The pulse is sent to the advanced control timer (TIM0) through the TRGO output.

Example: you want to change the PWM configuration of your advanced-control timer TIM0 after a programmed delay each time a change occurs on the Hall inputs connected the TIM6 timers.

- Configure 3 timer inputs ORed to the TI1 input channel by writing the TI1S bit in the TIM6_CR2 register to ‘ 1 ’;
- Program the time base: write the TIM6_ARR to the max value (the counter must be cleared by the TI1 change. Set the prescaler to get a maximum counter period longer than the time between 2 changes on the sensors;
- Program channel 1 in capture mode (TRC selected): write the CC1S bits in the TIM6_CCMR1 register to ‘ 11 ’. You can also program the digital filter if needed;
- Program channel 2 in PWM 2 mode with the desired delay: write the OC2M bits to ‘ 111 ’ and the CC2S bits to ‘00’ in the TIM6_CCMR1 register;
- Select OC2REF as trigger output on TRGO: write the MMS bits in the TIM6_CR2 register to ‘ 101 ’.

In the advanced control timer TIM0, the correct ITR input must be selected as the trigger input, the timer is programmed to generate a PWM signal, capture/compare the control signal for pre-loading (CCPC=1 of the TIM0_CR2 register), and the COM event is controlled by the trigger input (CCUS=1 of the TIM0_CR2 register). After the COM event occurs, the next configuration is written in the PWM control bits (CCxE, OCxM), which can be done in the interrupt subroutine generated by the OCxREF rising edge of the "interface timer".

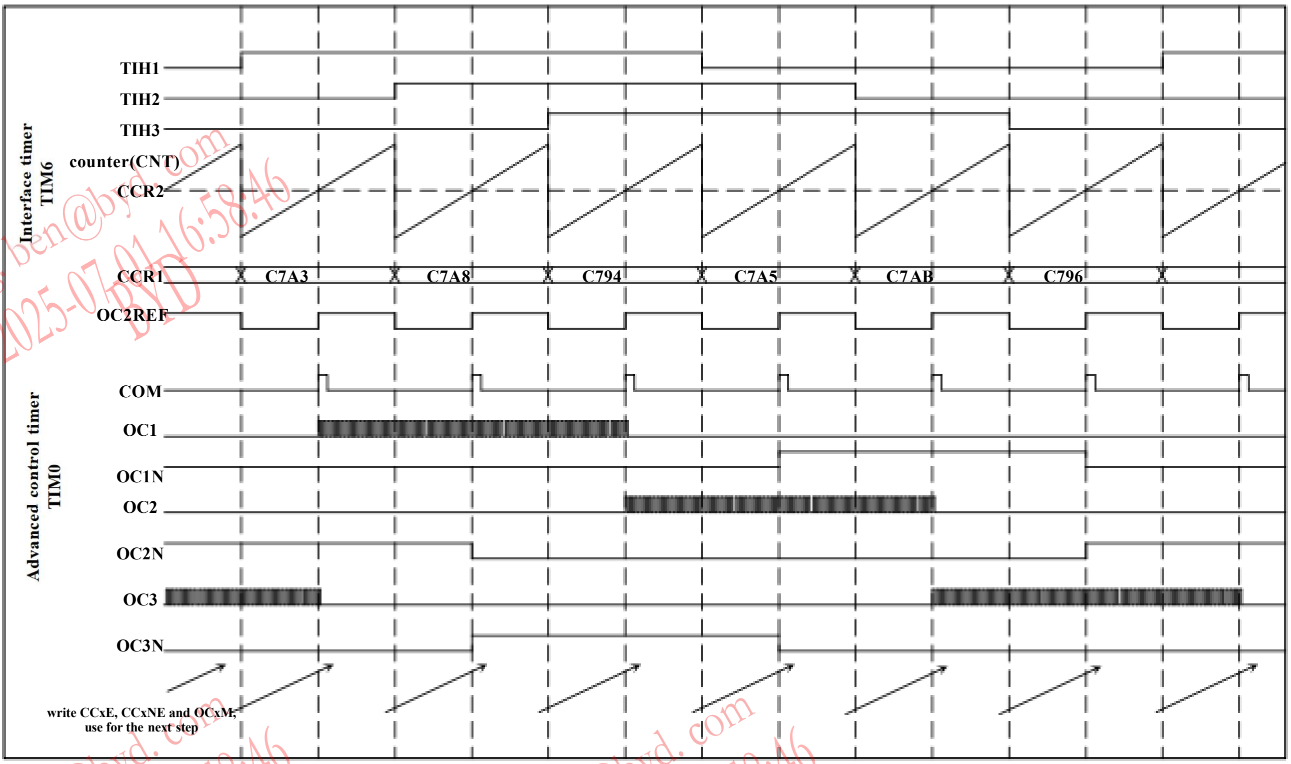


Figure 11.35 Example of a Hall sensor interface

11.3.28. DMA continuous transfer mode

The TIM0 timer can generate multiple DMA requests based on a single event. The main purpose is to be able to reprogram a portion of the timer multiple times without software overhead, but can also be used to periodically read multiple registers in a row.

The DMA controller target is unique and must point to the virtual register TIM0_DMAR. When a given timer event occurs, the timer initiates a DMA request sequence (burst). Every time write TIM0_DMAR registers will be redirected to one of the timer registers.

The DBL[4: 0] bit in the TIM0_DCR register sets the DMA continuous transfer length. When read or write access is made to the TIM0_DMAR address, the timer performs one consecutive transmission, that is, the number of transmission (in half words or bytes).

The DBA[4: 0] bit in the TIM0_DCR register defines the DMA base address of the DMA transfer (when Read/Write access is performed via the TIM0_DMAR address). DBA is defined as the offset calculated from the TIM0_CR1 register address:

Examples:

00000: TIM0_CR1
00001: TIM0_CR2
00010: TIM0_SMCR

For example, the timer DMA continuous transfer function is used to update the contents ofthe CCRx register (x =1, 2, 3, 4) to multiple half-words transferred to the CCRx register via DMA after an update event occurs.

The specific steps are as follows:

1. Configure the corresponding DMA channels as follows:
   a) The destination data end address of the DMA channel is the DMAR register address.
   b) The DMA channel source data end address is the RAM buffer address that contains the data to be transferred to the CCRx register via DMA.
   c) Amount of data to be transferred = 4
   d) R_power = 0x02
   e) Select the request source TIM0_UP
2. Configure the DCR register by configuring the DBA and DBL bit fields as follows: DBL=4, DBA=0xD.
3. Enable TIM0 to update DMA requests (UDE position 1 in DIER register).
4. Enable TIM0
5. Enable DMA channel

This example applies if each CCRx register is updated only once. If each CCRx register is updated twice, the amount of data to be transferred should be 8. The RAM buffers containing data1, data2, data3, data4, data5, data6, data7, and data8 are used as examples. The data is transferred to the CCRx register as follows: During the first update DMA request, data1 transfers to CCR1, data2 transfers to CCR2, data3 transfers to CCR3, and data4 transfers to CCR4; During the second update DMA request, data5 transfers to CCR1, data6 transfers to CCR2, data7 transfers to CCR3, and data8 transfers to CCR4.

11.4. TIM0 register

Base address: 0x5000 0000

Address offset	register	Description
0x04	RCU_EN	Peripheral module clock control register
0x10	ANA_CFG	Analog module switch register
0x44	TIM_CLK_SEL	TIM clock select register
0X48	CLK_SEL_RDY	Clock switch completion flag register

Base address: 0x5006 0000

Address offset	register	Description
0x00	TIM0_CR1	TIM0 control register 1
0x04	TIM0_CR2	TIM0 control register 2
0x08	TIM0_SMCR	TIM0 slave mode control register
0x0C	TIM0_DIER	TIM0 DMA/ interrupt enable register
0x10	TIM0_SR	TIM0 status register
0x14	TIM0_EGR	TIM0 event generation register
0x18	TIM0_CCMR1	TIM0 capture/compare mode register 1
0x1C	TIM0_CCMR2	TIM0 capture/compare mode register 2
0x20	TIM0_CCER	TIM0 capture/compare enable register
0x24	TIM0_CNT	TIM0 counter
0x28	TIM0_PSC	TIM0 predivider
0x2C	TIM0_ARR	TIM0 automatic reload register
0x30	TIM0_RCR	TIM0 repeat counter register
0x34	TIM0_CCR1	TIM0 capture/compare register 1
0x38	TIM0_CCR2	TIM0 capture/compare register 2
0x3C	TIM0_CCR3	TIM0 capture/compare register 3
0x40	TIM0_CCR4	TIM0 capture/compare register 4
0x44	TIM0_BDTR	TIM0 brake and dead-zone register
0x48	TIM0_DCR	TIM0 DMA control register
0x4C	TIM0_DMAR	TIM0 full transmission DMA address
0x54	TIM0_CCMR3	TIM0 capture/compare mode register 3
0x58	TIM0_CCR5	TIM0 capture/compare register 5
0x5C	TIM0_CCR6	TIM0 capture/compare register 6
0x60	TIM0_AF1	TIM0 reuse function option register 1
0x64	TIM0_AF2	TIM0 reuse function option register 2
0x68	TIM0_DIR	TIM0 input/output control register
0x6C	TIM0_EN	TIM0 module enable register

11.4.1. Peripheral module clock control register(RCU_EN)

Address offset: 0x04

Reset value: 0x0000 0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.				DIV_ CLKEN	GPIOD_ CLKEN	GPIOC_ CLKEN	GPIOB_ CLKEN	GPIOA_ CLKEN	DMA_ CLKEN	CRC_ CLKEN	ADC_ CLKEN	WDT_ CLKEN	TIM7_ CLKEN	TIM6_ CLKEN	TIM5_ CLKEN
				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
TIM4_ CLKEN	TIM3_ CLKEN	TIM2_ CLKEN	TIM1_ CLKEN	TIM0_ CLKEN	PWM1_ CLKEN	PWM0_ CLKEN	IIC_ CLKEN	UART4_ CLKEN	UART3_ CLKEN	UART2_ CLKEN	UART1_ CLKEN	UART0_ CLKEN	SPI1_ CLKEN	SPI0_ CLKEN	Res.
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	

26	GPIOD_CLKEN	GPIOD module work enabled: 1: Work 0: Close. Default is 0
25	GPIOC_CLKEN	GPIOC module work enabled: 1: Work 0: Close. Default is 0
24	GPIOB_CLKEN	GPIOB module work enabled: 1: Work 0: Close. Default is 0
23	GPIOA_CLKEN	GPIOA module work enabled: 1: Work 0: Close. Default is 0
22	DMA_CLKEN	DMA(including DMA_SRAM/DMAMUX) module work enabled: 1: Work 0: Close. Default is 0
11	TIM0_CLKEN	TIM0 module work enabled: 1: Work 0: Close. Default is 0

11.4.2. Analog module switch register(ANA_CFG)

Address offset: 0x10

Reset value: 0x0000 1850

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

15:13	12	11	10	9	8: 7	6	5	4	3 2	1: 0
Res.	PD_TEMP	PD_ADC	VREF_SEL	VREF_VOL_SEL	VREF_IN_ADC_SEL	PD_ADC_IN_VREF	XTAL_HFR_SEL	XTAL_SEL	XTAL_BYP_SEL	XTAL_EN_SEL
	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

5	XTAL_HFR_SEL	Current configuration selection of analog high frequency crystal oscillator circuit: 0: 175μA(Recommend 4 MHZ crystal vibration) 1: 300μA(Recommend 8 MHZ crystal vibration)
4	XTAL_SEL	When a passive crystal oscillator circuit is available: 0: Select 32K 1: Select 4M/8M. Default value 1
3: 2	XTAL_BYP_SEL	Active crystal oscillator path selection: 00: Channel closed, active crystal oscillator is not used 01: Select channel A (PA5) input 10: Select channel B (PA2) input 11: Reserved Notes: Select any active crystal path, and the passive crystal vibration circuit is turned off at the same time. Default value is 00.
1: 0	XTAL_EN_SEL	Passive crystal path selection: 00: The passive crystal oscillator is turned off and no passive crystal oscillator is used 01: Select channel A (PA5, PA4) for input 10: Select path B (PA2, PA1) for input 11: Reserved When XTAL_BYP_SEL[1: 0] = 00 and XTAL_EN_SEL[1: 0]! = 00, turn on the passive crystal vibration circuit, the default value is 00. Notes: Passive crystal oscillator must be enabled when active crystal oscillator is not enabled.

11.4.3. TIM clock select register(TIM_CLK_SEL)

Address offset: 0x44

Reset value: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIM7CLK		TIM6CLK		TIM5CLK		TIM4CLK		TIM3CLK		TIM2CLK		TIM1CLK		TIM0CLK	
RW		RW		RW		RW		RW		RW		RW		RW	

1: 0	TIM0CLK	TIM0 clock select: 00: Select the PLL48M clock 01: Select the LIRC 32K clock 1x: Select XTAL,Passive crystal oscillator(32768Hz/4MHz/8MHz)and active crystal oscillator (1MHz~48MHz)

11.4.4. Clock switch completion flag register(CLK_SEL_RDY)

Address offset: 0x48

Reset value: 0x0000 03FF

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

15: 10	9	8	7	6	5	4	3	2	1	0
Res.	TIM7RDY	TIM6RDY	TIM5RDY	TIM4RDY	TIM3RDY	TIM2RDY	TIM1RDY	TIM0RDY	PWM1RDY	PWM0RDY
	R	R	R	R	R	R	R	R	R	R

2	TIM0RDY	TIM0 clock switch completion flag: 0: Be switching 1: Clock switching completed

11.4.5. TIM0 control register1 (TIM0_CR1)

Address offset: 0x00

Reset value: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.				UIFREMAP	Res.	CKD[1:0]		ARPE	CMS[1:0]		DIR	OPM	URS	UDIS	CEN
				RW		RW		RW	RW		RW	RW	RW	RW	RW

31: 12	-	Reserved
11	UIFREMAP	UIF status bit remapping: 0: No remapping. The UIF status bit is not copied to bit 31 of the TIM0_CNT register 1: Enable remapping. The UIF status bit is copied to bit 31 of the TIM0_CNT register
10	-	Reserved
9: 8	CKD[1: 0]	Clock frequency division factor: Defines the frequency division ratio between the timer clock (TIM0_CLK) frequency and the dead band and sampling clock (t_{DTS}) used by the timer clock (TIM0_CLK) and the digital filter (ETR, TIx) 00: $t_{DTS} = t_{TIM0\_CLK}$ 01: $t_{DTS} = 2*t_{TIM0\_CLK}$ 10: $t_{DTS} = 4*t_{TIM0\_CLK}$ 11: Reserved, do not set this value Notes: This bit is configured before the configuration of the TIM0_EN register
7	ARPE	Automatic reload preload enable: 0: TIM0_ARR register with no buffering 1: TIM0_ARR register buffers
6: 5	CMS[1: 0]	Center alignment mode selection: 00: Edge alignment mode. The counter increments or deciles the count according to the direction bit (DIR) 01: Center alignment mode 1. The counter alternates between increasing and decreasing counts. The output comparison interrupt flag for the channel configured for output (CCxS=00 in the TIM0_CCMRx register) is set to 1 only when the counter decrement the count 10: Center alignment mode 2. The counter alternates between increasing and decreasing counts. The output comparison interrupt flag for the channel configured for output (CCxS=00 in the TIM0_CCMRx register) is set to 1 only when the counter incremented the count 11: Center alignment mode 3. The counter alternates between increasing and decreasing counts. When the counter increments or decays the count, the output comparison interrupt flag is set to 1 for the channel configured to output (CCxS=00 in the TIM0_CCMRx register) Notes: As long as the counter is enabled (CEN=1), it cannot be switched from edge alignment mode to center alignment mode. This bit is configured before the configuration of the TIM0_EN register
4	DIR	Direction: 0: Counter increment count 1: Counter decrement count Notes: When the timer is configured in center alignment mode or encoder mode, the bit is read-only
3	OPM	Single pulse mode: 0: The counter does not stop counting when an update event occurs

		1: The counter stops counting at the next update event (clears the CEN bit to zero)
2	URS	Update request source: This bit is set to 1 and cleared by the software to select the UEV event source 0: When enabled, all of the following events generate update interrupts or DMA requests. Such incidents include: – Counter overflow/underflow – The UG position 1 – Update events generated from the slave controller 1: When enabled, only counter overflows/underflows generate update interrupts or DMA requests
1	UDIS	Update forbidden: This bit is set to 1 and cleared by the software to enable/disable UEV event generation. 0: Enable UEV. An updated UEV event can be generated by one of the following events: – Counter overflow – The UG position 1 – Update events generated from the slave controller Then update the value of the shadow register. 1: Ban UEV. No update event is generated, and the values of each shadow register (ARR, PSC, and CCRx) remain unchanged. However, if the UG position 1, or hardware reset is received from the slave mode controller, the counter and predivider are reinitialized.
0	CEN	Counter enable: 0: Ban the counter 1: Enable counter Notes: External clock, gated mode and encoder mode can only be used if CEN position 1 is set by software in advance. While the trigger mode can automatically position the CEN 1 through the hardware

11.4.6. TIM0 control register2 (TIM0_CR2)

Address offset: 0x04

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.								MMS2[3:0]				Res.	OIS6	Res.	OIS5
								RW					RW		RW

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	OIS4	OIS3N	OIS3	OIS2N	OIS2	OIS1N	OIS1	TI1S	MMS[2:0]			Res.	CCUS	Res.	CCPC
	RW	RW	RW	RW	RW	RW	RW	RW	RW				RW		RW

31: 24	-	Reserved,The reset value must be reserved.

23: 20	MMS2[3: 0]	Master mode selection 2: These bits select the information that will be sent to the ADC for synchronization(TRGO2) 0000: Reset——The UG bit in the TIM0_EGR register is used as a trigger output (TRGO2) 0001: Enable——The counter enable signal CNT_EN is used as a trigger output (TRGO2) 0010: Update——Select the update event as the trigger output (TRGO2) 0011: Comparison pulse——When the CC1IF flag is set to 1 (even if it is already high), the trigger output (TRGO2) sends a positive pulse whenever a capture or comparison match occurs 0100: Comparison——The OC1REF signal is used as a trigger output (TRGO2) 0101: Comparison——The OC2REF signal is used as a trigger output (TRGO2) 0110: Comparison——The OC3REF signal is used as a trigger output (TRGO2) 0111: Comparison——The OC4REFsignal is used as a trigger output (TRGO2) 1000: Comparison——The OC5REF signal is used as a trigger output (TRGO2) 1001: Comparison——The OC6REF signal is used as a trigger output (TRGO2) 1010: Comparison pulse——When OC4REF rises or falls along the edge, pulses are generated on TRGO2 1011: Comparison pulse——When OC6REF rises or falls along the edge, pulses are generated on TRGO2 1100: Comparison pulse——When OC4REF or OC6REF rises, a pulse is generated on TRGO2 1101: Comparison pulse——When OC4REF rising edge or OC6REF falling edge, pulse is generated on TRGO2 1110: Comparison pulse——When OC5REF or OC6REF rises, a pulse is generated on TRGO2 1111: Comparison pulse——When OC5REF rising edge or OC6REF falling edge, pulse is generated on TRGO2 Notes: The ADC's clock must be enabled before events can be received from the master timer; The ADC clock must not be changed in real time when the trigger signal is received from the master timer. This bit is configured before the configuration of the TIM0_EN register
19	-	Reserved
18	OIS6	Output idle state 6 (OC6 output): Please refer to OIS1bit
17	-	Reserved
16	OIS5	Output idle state 5(OC5 output): Please refer to OIS1bit
15	-	Reserved
14	OIS4	Output idle state 4(OC4 output): Please refer to OIS1bit

13	OIS3N	Output idle state 3(OC3N output): Please refer to OIS1N bit
12	OIS3	Output idle state 3(OC3 output): Please refer to OIS1bit
11	OIS2N	Output idle state 2(OC2N output): Please refer to OIS1N bit
10	OIS2	Output idle state 2(OC2 output): Please refer to OIS1bit
9	OIS1N	Output idle state 1(OC1N output): 0: When MOE=0, (after dead time if OC1N is valid)OC1N=0 1: When MOE=0, (after dead time if OC1N is valid)OC1N= 1 Notes: This bit cannot be modified as long as LOCK (the LOCK bit in the TIM0_BDTR register) level 1, 2, or 3 is programmed. This bit is configured before the configuration of the TIM0_EN register
8	OIS1	Output idle state 1(OC1 output): 0: When MOE=0, (after dead time if OC1 is valid) OC1=0 1: When MOE=0, (after dead time if OC1 is valid) OC1= 1 Notes: This bit cannot be modified as long as LOCK (the LOCK bit in the TIM0_BDTR register) level 1, 2, or 3 is programmed. This bit is configured before the configuration of the TIM0_EN register
7	TI1S	TI1 select: 0: TIM0_CH1 pin is connected to the TI1 input 1: TIM0_CH1, CH2, and CH3 pins connected to TI1 inputs (XOR combination)
6: 4	MMS[2: 0]	Master mode selection: These bits select the information to be sent to the slave timer in master mode for synchronization(TRGO) 000: Reset——The UG bit in the TIM0_EGR register is used as a trigger output(TRGO) 001: Enable——Counter Enable signal CNT_EN used as trigger output (TRGO). The trigger output can be used to start multiple timers at the same time, or to control enabling the slave timer over a period of time. In gated mode, the counter enable signal is the logic and generation of the CEN control bit and trigger input signal. There is a delay in TRGO when the counter enable signal is controlled by the trigger input 010: Update——Select the update event as the trigger output(TRGO) 011: Comparison pulse——Once an input capture or comparison match event occurs, the triggered output sends a positive pulse (TRGO) when the CC1IF flag is set to 1(even if it is high). 100: Comparison——OC1REF signal used as trigger output (TRGO) 101: Comparison——OC2REF signal used as trigger output (TRGO) 110: Comparison——OC3REF signal used as trigger output (TRGO) 111: Comparison——OC4REF signal used as trigger output (TRGO) Notes: Before receiving events from the master timer, the clock of the slave timer must be enabled; When the trigger signal is received from the master timer, the clock of the

		slave timer must not be changed in real time. This bit is configured before the configuration of the TIM0_EN register
3	-	Reserved
2	CCUS	Capture/compare control update selection: 0: If the capture/compare control bits are preloaded (CCPC= 1), these bits are updated only by setting the COMG position to 1 1: If the capture/compare control bits are preloaded (CCPC= 1), these bits can be updated by COMG position 1 or the rising edge of TRGI Notes: This bit is only valid for channels with complementary outputs. Notes: This bit is configured before the configuration of the TIM0_EN register
1	-	Reserved
0	CCPC	Capture/compare preloaded controls: 0: The CCxE, CCxNE, and OCxM bits are not preloaded 1: The CCxE, CCxNE, and OCxM bits are preloaded, and after they are written, they are updated only when a commutation event (COM) occurs (COMG position 1 or a rising edge is detected on the TRGI, depending on the CCUS bit) Notes: This bit is only valid for channels with complementary outputs Notes: This bit is configured before the configuration of the TIM0_EN register

11.4.7. TIM0 slave mode control register (TIM0_SMCR)

Address offset: 0x08

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.										TS[4:3]		Res.			SMS[3]
										RW					RW

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETP	ECE	ETPS[1:0]		ETF[3:0]				MSM	TS[2:0]			Res.	SMS[2:0]		
RW	RW	RW		RW				RW	RW				RW		

31: 22	-	Reserved
21: 20	TS[4: 3]	Trigger selection: This bit [4: 3] is the bit of TS[4: 0],annotation refer to TS[4: 0]
19: 17	-	Reserved
16	SMS[3]	slave mode selection: This bit[3] is the bit of SMS[3: 0],annotation refer to SMS[3: 0]
15	ETP	External trigger polarity: This bit selects whether the ETR or the ETR reverse phase is used to trigger the operation 0: ETR is not reversed, high or rising edge effective

		1: ETR is reversed, low level or falling edge effective
14	ECE	External clock enabled: This bit enables external clock mode 2 0: Disable the external clock mode 2 1: Enable the external clock mode2 Notes: 1、Setting ECE position 1 has the same effect as selecting external clock mode 1 and connecting TRGI to ETRF (SMS= 111 and TS=00111) 2、External clock mode 2 can be used in conjunction with the following slave modes: Reset mode, gate mode, and trigger mode. However, in this case, TRGI must not connect to ETRF (TS bit must not be 00111) 3、External clock mode 1 and external clock mode 2 cannot be used simultaneously
13: 12	ETPS[1: 0]	Externally triggered predivider: The external trigger signal ETRP frequency must not exceed 1/4 of the TIM0_CLK frequency. The ETRP frequency can be reduced by enabling the predivider. This approach is useful when entering a fast external clock. 00: The predivider is off 01: 2 frequency division ETRP frequency 10: 4 frequency division ETRP frequency 11: 8 frequency division ETRP frequency Notes: This bit is configured before the configuration of the TIM0_EN register
11:8	ETF[3:0]	External trigger filter: This bitfield defines the sampling frequency of ETRP signals and the bandwidth of digital filters applicable to ETRP. The digital filter consists of an event counter, where each N consecutive events is considered a valid output edge: 0000: No filter, sampling at f_{DTS} frequency 0001: $f_{SAMPLING} = f_{TIM0\_CLK}$, N=2 0010: $f_{SAMPLING} = f_{TIM0\_CLK}$, N=4 0011: $f_{SAMPLING} = f_{TIM0\_CLK}$, N=8 0100: $f_{SAMPLING} = f_{DTS}/2$, N=6 0101: $f_{SAMPLING} = f_{DTS}/2$, N=8 0110: $f_{SAMPLING} = f_{DTS}/4$, N=6 0111: $f_{SAMPLING} = f_{DTS}/4$, N=8 1000: $f_{SAMPLING} = f_{DTS}/8$, N=6 1001: $f_{SAMPLING} = f_{DTS}/8$, N=8 1010: $f_{SAMPLING} = f_{DTS}/16$, N=5 1011: $f_{SAMPLING} = f_{DTS}/16$, N=6 1100: $f_{SAMPLING} = f_{DTS}/16$, N=8 1101: $f_{SAMPLING} = f_{DTS}/32$, N=5 1110: $f_{SAMPLING} = f_{DTS}/32$, N=6 1111: $f_{SAMPLING} = f_{DTS}/32$, N=8 Note: This bit is set before the TIM0_EN register is configured.

7	MSM	Master/slave mode: 0:No operation is performed 1:The action of the trigger input event (TRGI) of the current timer is deferred so that the current timer is perfectly synchronized with its slave timer (by TRGO). This setting is useful when multiple timers are synchronized by a single external event Note: This bit is set before the TIM0_EN register is configured
6:4	TS[2:0]	Trigger selection:The bits ofTS bit [4:3] is located at the bits of TIM0_SMCR bit [21:20] This bit field selects the trigger input to be used to synchronize the counter 00000:Internal trigger 0 (ITR0) --------->TIM6_TRGO 00100:TI1 Edge detector (TI1F_ED) 00101:Filtered timer input1 (TI1FP1) 00110:Filtered timer input2 (TI2FP2) 00111:External trigger input(ETRF) Other values: reserved Note: These bits can only be changed when not in use (for example, SMS=0000) to avoid erroneous edge detection during conversion. This bit is configured before the configuration of the TIM0_EN register
3	-	reserved
2:0	SMS[2:0]	slave mode selection: The SMS bit[3] is located at the TIM0_SMCR bit[16] When selecting an external signal, the effective edge of the trigger signal (TRGI) is related to the polarity selected on the external input. 0000:Disable slave mode ---- if CEN = "1", the pre-divider clock is supplied directly by the internal clock 0001:Encoder mode 1 ---- The counter increments/decays the count along the edge of TI1FP1 according to the TI2FP2 level 0010:Encoder mode 2 ---- The counter increments/decays the count along the edge of TI2FP2 according to the TI1FP1 level 0011:Encoder mode 3 ---- The counter counts at the edge of TI1FP1 and TI2FP2, the direction of the count depends on the level of the other input 0100:Reset mode ---- Reinitializes the counter and generates a register update event when the selected trigger input (TRGI) rise edge appears 0101:Gated mode ---- Trigger Input (TRGI) enables counter clock at high power level. As soon as the trigger input becomes low, the counter immediately stops counting (but no reset). The start and stop of the counter are controlled 0110:Trigger mode ---- Start the counter (but no reset) when the trigger signal TRGI appears on a rising edge. Only the startup of the counter is controlled 0111:External clock mode 1 ---- The counter clock is provided by the rising edge of the selected trigger signal (TRGI) 1000:Combined reset + trigger mode ---- When the selected trigger input (TRGI) rise edge appears, the counter is reinitialized, a register update event is generated, and the counter is started

		Other: Reserved Note: If TI1F_ED is selected as the trigger input (TS=00100), gating mode cannot be used. In fact, each time TI1F is converted, TI1F_ED outputs a pulse, and the gated mode checks the level of the trigger signal Note: The clock of the slave peripheral (timer, ADC, etc.) that receives the TRGO or TRGO2 signal must be enabled before the event can be received from the master timer; And when receiving the trigger signal from the master timer, the clock frequency must not be changed in real time (pre-divider). Note: This bit is set before the TIM0_EN register is configured

11.4.8. TIM0 DMA/Interrupt enable register (TIM0_DIER)

Address offset: 0x0C

Reset value:0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	TDE	COMDE	CC4DE	CC3DE	CC2DE	CC1DE	UDE	BIE	TIE	COMIE	CC4IE	CC3IE	CC2IE	CC1IE	UIE
	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

31:15	-	Reserved
14	TDE	Trigger DMA request enable: 0:Triggering DMA requests is disabled 1:Enable the triggering of DMA requests
13	COMDE	COM DMA request enable: 0:Disables COM DMA requests 1:Enable the COM DMA request
12	CC4DE	Enable Capture/compare 4 DMA requests 0:Disable CC4 DMA requests 1:Enable the CC4 DMA request
11	CC3DE	Enable Capture/compare 3 DMA requests 0:Disable CC3 DMA requests 1:Enable the CC3 DMA request
10	CC2DE	Enable Capture/compare 2 DMA requests 0:Disable CC2 DMA requests 1:Enable the CC2 DMA request
9	CC1DE	Enable Capture/compare 1 DMA requests 0:Disable CC1 DMA requests 1:Enable the CC1 DMA request
8	UDE	Update DMA request enable: 0:Disable updating DMA requests

		1:The update DMA request was enabled
7	BIE	Brake interrupt enable: 0:Disable Brake interrupt 1:Enable Brake interrupt
6	TIE	Trigger interrupt enable: 0:Disable trigger interrupt 1:Enable trigger interrupt
5	COMIE	COM interrupt enable: 0:Disable COM interrupt 1:Enable COM interrupt
4	CC4IE	Capture/compare 4 Interrupt enable: 0:Disable CC4 interrupt 1:Enable CC4 interrupt
3	CC3IE	Capture/compare 3 Interrupt enable: 0:Disable CC3 interrupt 1:Enable CC3 interrupt
2	CC2IE	Capture/compare 2 Interrupt enable: 0:Disable CC2 interrupt 1:Enable CC2 interrupt
1	CC1IE	Capture/compare 1 Interrupt enable: 0:Disable CC1 interrupt 1:Enable CC1 interrupt
0	UIE	Update interrupt enable: 0:Disable update interrupt 1:Enable update interrupt

11.4.9. TIM0 status register (TIM0_SR)

Address offset: 0x10

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.													RDCNTF	CC6IF	CC5IF
													RW	RC_W0	RC_W0

15:13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	CC4OF	CC3OF	CC2OF	CC1OF	B2IF	BIF	TIF	COMIF	CC4IF	CC3IF	CC2IF	CC1IF	UIF
	RC_W0	RC_W0	RC_W0	RC_W0	RC_W0	RC_W0	RC_W0	RC_W0	RC_W0	RC_W0	RC_W0	RC_W0	RC_W0

31:18	-	Reserved
18	RDCNTF	TIM0_CNT register read flag bit: 1:Software write 1, counter count value latch enable, latch counter count value 0:The counter will clear the bit of hardware after the value latch is completed. This bit can

		also be written or cleared by software.
17	CC6IF	Compare 6 Interrupt flags: Refer to CC1IF description (note: Channel 6 can only be configured as output)
16	CC5IF	Compare 5 Interrupt flags: Refer to CC1IF description(note: Channel 5 can only be configured as output)
15:13	-	Reserved
12	CC4OF	Capture/compare 4 Repeat capture flag: Refer to CC1OF description
11	CC3OF	Capture/compare 3 Repeat capture flag: Refer to CC1OF description
10	CC2OF	Capture/compare 2 Repeat capture flag: Refer to CC1OF description
9	CC1OF	Capture/compare 1 Repeat capture flag: This flag bit is set to 1 by the hardware only if the corresponding channel is configured in input capture mode. This bit can be cleared by writing "0" to the software 0:No duplicate capture was detected 1:The counter value is captured in the TIM0_CCR1 register and the CC1IF flag is set to 1
8	B2IF	Brake 2 interrupt flag: As soon as the brake input 2 becomes active, this flag is set to 1 by the hardware. Brake 2 If the brake input 2 is invalid, it can be cleared by software 0:No braking events occurred 1:Active level detected on brake input 2. If BIE= 1 in the TIM0_DIER register, an interrupt is generated
7	BIF	Brake interrupt flag: As soon as the brake input becomes active, this flag is set to 1 by the hardware. If the brake input is invalid, it can be cleared by software. 0:No braking events occurred 1:Active level detected on brake input. If BIE= 1 in the TIM0_DIER register, an interrupt is generated
6	TIF	Trigger interrupt flag: In all modes except gated mode, the flag is set to 1 by the hardware when a valid edge is detected on the TRGI input after the mode controller is enabled. When gating mode is selected, the flag is set to 1 when the counter is started or stopped. But it needs to be cleared by software. 0:No triggering event occurred 1:The triggered interrupt is suspended
5	COMIF	COM interrupt flag: This flag is set to 1 by the hardware in the event of a COM event (when the capture/compare control bits CCxE, CCxNE, and OCxM have been updated). But it needs to be cleared by software. 0:No COM event occurred 1:COM interrupt is suspended

4	CC4IF	Capture/compare 4 Interrupt flags: Refer to the CC1IF description
3	CC3IF	Capture/compare 3 Interrupt flags: Refer to the CC1IF description
2	CC2IF	Capture/compare 2 Interrupt flags: Refer to the CC1IF description
1	CC1IF	Capture/compare 1 Interrupt flag: **If channel CC1 is configured to output:** This flag is set to 1 by hardware when the counter matches the comparison value, except in center alignment mode. But it needs to be cleared by software. 0:mismatch 1:The value of the TIM0_CNT counter matches the value of the TIM0_CCR1 register. When the value of TIM0_CCR1 is greater than the value of TIM0_ARR, the CC1IF bit becomes high when the counter overflows (in increasing and decreasing count modes) or underflows (in decreasing count modes) **If channel CC1 is configured as input:** this bit will be set to 1 by the hardware when a capture event occurs. Clear this bit to zero by software or reading the TIM0_CCR1 register 0:No input capture event occurred 1:Counter values captured in the TIM0_CCR1 register (edges matching the selected polarity detected on IC1)
0	UIF	Update interrupt flag: This bit is set to 1 by the hardware and cleared to 0 by the software when an update event occurs 0:No update occurred 1:Update interrupt suspended. This bit is set to 1 by the hardware when updating the register in the following cases: —UDIS=0 in the TIM0_CR1 register and when the repeat counter value overflows or underflows (updated when the repeat counter =0); —URS=0 and UDIS=0 in the TIM0_CR1 register,And CNT is re-initialized by software using UG bit in TIM0_EGR register; —URS=0 and UDIS=0 in the TIM0_CR1 register, and CNT is reinitialized by the trigger event;

11.4.10. TIM0 event generation register (TIM0_EGR)

Address offset: 0x14

Reset value: 0x0000 0000

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

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

Res.	B2G	BG	TG	COMG	CC4G	CC3G	CC2G	CC1G	UG
	RW	RW	RW	RW	RW	RW	RW	RW	RW

31:9	-	Reserved
8	B2G	Brake 2 generation: This bit is set to 1 by the software to generate events and automatically cleared by the hardware. 0:No operation is performed 1:Generate brake 2 event. The MOE bit is cleared to 0 and the B2IF flag is set to 1. After this function is enabled, related interruptions can occur
7	BG	Brake generation: This bit is set to 1 by the software to generate events and automatically cleared by the hardware. 0:No operation is performed 1:Generate brake events. The MOE bit is cleared to 0 and the BIF flag is set to 1. After this function is enabled, related interrupts or DMA transfer events can occur
6	TG	Trigger generation: This bit is set to 1 by the software to generate events and automatically cleared by the hardware. 0:No operation is performed 1:The TIF flag in the TIM0_SR register is set to 1. After this function is enabled, related interrupts or DMA transfer events can occur
5	COMG	Capture/compare control update generation: This bit can be set to 1 by software and automatically cleared by hardware 0:No operation is performed 1:When the CCPC position is 1, the CCxE, CCxNE, and OCxM bits can be updated Note: This bit is only valid for channels with complementary outputs
4	CC4G	Capture/compare 4 Generated: Refer to the CC1G description
3	CC3G	Capture/compare 3 Generated: Refer to the CC1G description
2	CC2G	Capture/compare 2 Generated: Refer to the CC1G description
1	CC1G	Capture/compare 1 Generated: This bit is set to 1 by the software to generate events and automatically cleared by the hardware 0:No operation is performed 1:Generate capture/compare events on channel 1: **If channel CC1 is configured as output:** When enabled, the CC1IF flag is set to 1 and the corresponding interrupt or DMA request is sent **If channel CC1 is configured as input:**

		The TIM0_CCR1 register will capture the current value of the counter. When enabled, the CC1IF flag is set to 1 and the corresponding interrupt or DMA request is sent. If the CC1IF flag is already high, the CC1OF flag is set to 1
0	UG	Update generation: This bit can be set to 1 by software and automatically cleared by hardware. 0:No operation is performed 1:Reinitializes the counter and generates a register update event. Note that the pre-divider counter will also be cleared to 0 (but the pre-divider ratio will not be affected). If you select center alignment mode or DIR=0 (incrementing count), the counter clears to 0; If DIR=1 (decrement count), the counter will use the automatically overloaded value (TIM0_ARR)

11.4.11. TIM0 capture/compare mode register 1[Reuse](TIM0_CCMR1)

Address offset: 0x18

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
Res.							OC2M[3]	Res.							OC1M[3]
							RW								RW

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IC2F[3:0]				IC2PSC[1:0]		CC2S[1:0]		IC1F[3:0]				IC1PSC[1:0]		CC1S[1:0]	
OC2CE	OC2M[3:0]			OC2PE	OC2FE	CC2S[1:0]		OC1CE	OC1M[3:0]			OC1PE	OC1FE	CC1S[1:0]	
RW	RW			RW	RW	RW		RW	RW			RW	RW	RW	

Enter the capture mode:

31:16	-	Reserved
15:12	IC2F[3:0]	Input capture 2 filter: Refer to IC1F[3:0] description
11:10	IC2PSC[1:0]	Input capture 2 pre-divider: Refer to OC1PSC[1:0] description
9:8	CC2S[1:0]	Capture/Compare 2 Select: This bit field defines the channel direction (input/output) and the input pins used 00:The CC2 channel is configured as output 01:The CC2 channel is configured as input, and IC2 is mapped to TI2 10:The CC2 channel is configured as input, and IC2 is mapped to TI1 11:The CC2 channel is configured as input, and IC2 is mapped to the TRC. This mode is only valid if the internal trigger input (ITR0) is selected via the TS bit (TIM0_SMCR register) Note: 1. Data can be written to the CC2S bit only when the channel is closed (CC2E=0, CC2NE=0 in TIM0_CCER and has been updated). This bit is configured before the

		configuration of the TIM0_EN register.
7:4	IC1F[3:0]	Input capture 1 filter: This bit field defines the sampling frequency and digital filter bandwidth of the TI1 signal. The digital filter consists of an event counter, where each N consecutive events is considered a valid output edge: 0000: No filter, sampling at f_{DTS} frequency 0001: $f_{SAMPLING} = f_{TIM0\_CLK}$, N=2 0010: $f_{SAMPLING} = f_{TIM0\_CLK}$, N=4 0011: $f_{SAMPLING} = f_{TIM0\_CLK}$, N=8 0100: $f_{SAMPLING} = f_{DTS}/2$, N=6 0101: $f_{SAMPLING} = f_{DTS}/2$, N=8 0110: $f_{SAMPLING} = f_{DTS}/4$, N=6 0111: $f_{SAMPLING} = f_{DTS}/4$, N=8 1000: $f_{SAMPLING} = f_{DTS}/8$, N=6 1001: $f_{SAMPLING} = f_{DTS}/8$, N=8 1010: $f_{SAMPLING} = f_{DTS}/16$, N=5 1011: $f_{SAMPLING} = f_{DTS}/16$, N=6 1100: $f_{SAMPLING} = f_{DTS}/16$, N=8 1101: $f_{SAMPLING} = f_{DTS}/32$, N=5 1110: $f_{SAMPLING} = f_{DTS}/32$, N=6 1111: $f_{SAMPLING} = f_{DTS}/32$, N=8 Note: The configuration is completed before the TIM0_EN register configuration
3:2	IC1PSC[1:0]	Input capture 1 pre-divider: This bit field defines the pre-division ratio of the CC1 input (IC1). As soon as CC1E= "0" (TIM0_CCER register), the pre-divider is reset immediately 00:Without a pre-divider, capture is performed on every edge detected on the capture input 01:A capture is performed for every 2 events 10:A capture is performed for every 4 events 11:A capture is performed for every 8 events Note: The configuration is completed before the TIM0_EN register configuration
1:0	CC1S[1:0]	Capture/Compare 1 Select: This bit field defines the channel direction (input/output) and the input pins used 00:The CC1 channel is configured as output 01:The CC1 channel is configured as input, and IC1 is mapped to TI1 10:The CC1 channel is configured as input, and IC1 is mapped to TI2 11:The CC1 channel is configured as the input, and IC1 is mapped to the TRC. This mode is only valid if the internal trigger input (ITR0) is selected via the TS bit (TIM0_SMCR register) Note: 1. Data can be written to the CC1S bit only when the channel is closed (CC1E=0, CC1NE=0 in TIM0_CCER and has been updated). This bit is configured before the

		configuration of the TIM0_EN register

Output comparison mode:

31:25	-	Reserved
24	OC2M[3]	Output comparison 2 mode: This bit is the third bit of OC2M[3:0], note reference OC2M[3:0]
23:17	-	Reserved
16	OC1M[3]	Output comparison 1 mode: This bit is the third bit of OC1M[3:0], note reference OC1M[3:0]
15	OC2CE	Output comparison 2 Clear enable: Refer to the OC1CE description
14:12	OC2M[3:0]	Output comparison 2 mode: The third bit of OC2M is at the 24-bit of TIM0_CCMR1(output comparison) Refer to the OC1M[3:0] description
11	OC2PE	Output comparison 2 Preload enable: Refer to the OC1PE description
10	OC2FE	Output Comparison 2 Quick enable: Refer to the OC1FE description
9:8	CC2S[1:0]	Capture/Compare 2 Select: This bit field defines the direction of the channel (input/output) and the input used 00:The CC2 channel is configured as output 01:The CC2 channel is configured as input, and IC2 is mapped to TI2 10:The CC2 channel is configured as input, and IC2 is mapped to TI1 11:The CC2 channel is configured as the input, and IC2 is mapped to the TRC. This mode is only valid if the internal trigger input is selected via the TS bit (TIM0_SMCR register) Note: Data can be written to the CC2S bit only when the channel is closed (CC2E =0, CC2NE=0 in TIM0_CCER and has been updated)
7	OC1CE	Output comparison 1 Clear enable: 0:OC1REF is not affected by ocref_clr_int signal 1:When high voltage is detected on the ocref_clr_int signal (ETRF input), the OC1REF clears immediately
6:4	OC1M[3:0]	Output comparison 1 mode: The third bit of OC1M is located at 16 bits of TIM0_CCMR1(output comparison) These bits define the behavior of OC1REF, the output reference signal for OC1 and OC1N. OC1REF is active at a high level, while the active levels of OC1 and OC1N depend on the CC1P bit and CC1NP bit. 0000:Freeze ---- Comparing the output comparison register TIM0_CCR1 with the counter TIM0_CNT has no effect on the output. (This pattern is used to generate the time base) 0001:Set channel 1 to the output active level when matched. When the counter TIM0_CNT matches the capture/comparison register 1(TIM0_CCR1), the OC1REF signal is forced to a high level

		0010:Set channel 1 to output invalid level when matched. When the counter TIM0_CNT matches the capture/comparison register 1 (TIM0_CCR1), the OC1REF signal is forced to a low level 0011:Flip ---- When TIM0_CNT= TIM0_CCR1, OC1REF flips 0100:Forced to invalid level ---- OC1REF forced to low level 0101:Forced to active level ---- OC1REF forced to high level 0110:PWM Mode 1 ----In increasing count mode, as long as TIM0_CNT is in decreasing count mode, TIM0_CNT>TIM0_CCR1, channel 1 is invalid (OC1REF= "0"); otherwise, channel 1 is valid (OC1REF= "1"). 0111:Retrigger PWM mode 2 ---- In incremental count mode, as long as TIM0_CNT<TIM0_CCR1, channel 1 is invalid, otherwise it is valid.In decrement mode, as long as TIM0_CNT> TIM0_CCR1, channel 1 is valid, otherwise it is invalid. 1000:OPM mode 1 - in incremental count mode, the channel is in a valid state until a trigger event is detected (on the TRGI signal). The comparison is then made in PWM mode 1, and the channel becomes active again at the next update. In decrement count mode, the channel is in an invalid state until a trigger event is detected (on the TRGI signal). The comparison is then made in PWM mode 1, and the channel becomes invalid again at the next update 1001:Retrigger PWM mode 2 ----In incremental count mode, the channel is in an invalid state until a trigger event is detected (on the TRGI signal). Then, when compared in PWM mode 2, the channel becomes invalid again at the next update. In decreasing count mode, the channel is in a valid state until a trigger event is detected (on the TRGI signal). The comparison is then made in PWM mode 1, and the channel becomes active again at the next update 1010/1011:Reserved 1100:Combined PWM Mode 1 ---- OC1REF behaves the same as in PWM mode 1. OC1REFC is the OR result of OC1REF and OC2REF 1101:Combined PWM Mode 2 ---- OC1REF behaves the same as in PWM mode 2. OC1REFC is the result of OC1REF and OC2REF 1110:Asymmetric PWM Mode 1 ---- OC1REF behaves the same as in PWM mode 1.When the counter is incremented, OC1REFC outputs OC1REF; When the counter decays the count, OC1REFC outputs OC2REF 1111:Asymmetric PWM Mode 2 ---- OC1REF behaves the same as in PWM mode 2. When the counter is incremented, OC1REFC outputs OC1REF; When the counter decays the count, OC1REFC outputs OC2REF Note: 1.As long as LOCK (LOCK bit in TIM0_BDTR register) level 3 is programmed and CC1S= "00" (channel configuration as output), these bits cannot be modified. 2.In PWM mode, the OCREF level changes only when the comparison result changes or the output comparison mode switches from "Freeze" mode to "PWM" mode 3.This bit field will be preloaded on channels with complementary outputs. If the CCPC position is 1 in the TIM0_CR2 register, the OC1M significant bit gets a new value from

		the preloaded bit only when a COM event is generated.
3	OC1PE	Output comparison 1 Preloaded Enable: 0:Disable preloaded registers associated with TIM0_CCR1. Data can be written to TIM0_CCR1 at any time, and the new value will be used immediately after writing 1:Enable the preloaded register associated with TIM0_CCR1. Read/write access to the preloaded register. The TIM0_CCR1 preloaded value is loaded into the current register each time an update event is generated. Note: 1. As long as LOCK (LOCK bit in TIM0_BDTR register) level 3 is programmed and CC1S= "00" (channel configuration as output), these bits cannot be modified. 2. PWM mode (The OPM bit in the TIM0_CR1 register is set to 1) can only be used in monopulse mode without verifying the pre-loaded register. In other cases, the behavior is not guaranteed.
2	OC1FE	Output Comparison 1 Quick enable: This bit is used to speed up the impact of triggered input events on CC output. 0:Even if triggered to turn on, CC1 will work properly according to the counter and CCR1 value. The minimum delay time to activate the CC1 output when the trigger input appears edging is 5 clock cycles 1:Triggering an effective edge on the input corresponds to a comparative match on the CC1 output. Subsequently, OC is set to the comparison level regardless of the comparison result. The OCFE works only when the channel is configured in PWM1 or PWM2 mode
1:0	CC1S[1:0]	Capture/Compare 1 Select: This bit field defines the direction of the channel (input/output) and the input used 00:The CC1 channel is configured as output 01:The CC1 channel is configured as input, and IC1 is mapped to TI1 10:The CC1 channel is configured as input, and IC1 is mapped to TI2 11:The CC1 channel is configured as input, and IC1 is mapped to the TRC. This mode is only valid if the internal trigger input (ITR0) is selected via the TS bit (TIM0_SMCR register) Note: Data can be written to the CC1S bit only when the channel is closed (CC1E=0, CC1NE=0 in TIM0_CCER and has been updated).

11.4.12. TIM0 capture/compare mode register 2[Reuse] (TIM0_CCMR2)

Address offset: 0x1C

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
Res.							OC4M[3]	Res.							OC3M[3]
							RW								RW

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IC4F[3:0]				IC4PSC[1:0]		CC4S[1:0]		IC3F[3:0]				IC3PSC[1:0]		CC3S[1:0]	
OC4CE	OC4M[3:0]			OC4PE	OC4FE	CC4S[1:0]		OC3CE	OC3M[3:0]			OC3PE	OC3FE	CC3S[1:0]	
RW	RW			RW	RW	RW		RW	RW			RW	RW	RW	

Enter the capture mode:

31:16	-	Reserved
15:12	IC4F[3:0]	Input capture 4 filter: Refer to IC1F[3:0] description
11:10	IC4PSC[1:0]	Input capture 4 Pre-divider: Refer to OC1PSC[1:0] description
9:8	CC4S[1:0]	Capture/Compare 4 Select: This bit field defines the direction of the channel (input/output) and the input used 00:The CC4 channel is configured as output 01:The CC4 channel is configured as input, and IC4 is mapped to TI4 10:The CC4 channel is configured as input, and IC4 is mapped to TI3 11:The CC4 channel is configured as input, and the IC4 is mapped to the TRC. This mode is only valid if the internal trigger input is selected via the TS bit (TIM0_SMCR register) Note: 1. Data can be written to the CC4S bit only when the channel is closed (CC4E=0 in TIM0_CCER) 2. This bit is set before the configuration of the TIM0_EN register
7:4	IC3F[3:0]	Input capture 3 filters: Refer to IC1F[3:0] description
3:2	IC3PSC[1:0]	Input capture 3 pre-divider Refer to OC1PSC[1:0] description
1:0	CC3S[1:0]	Capture/compare 3 Select: Define the channel direction (input/output) and the input pins to be used 00:The CC3 channel is configured as output 01:The CC3 channel is configured as input, and IC4 is mapped to TI3 10:The CC3 channel is configured as input, and IC4 is mapped to TI4 11:The CC3 channel is configured as input, and IC3 is mapped to the TRC. This mode is only valid if the internal trigger input is selected via the TS bit (TIM0_SMCR register) Note: 1.Data can be written to the CC3S bit only when the channel is closed (CC3E=0, CC3NE=0 in TIM0_CCER and has been updated) 2. This bit is set before the configuration of the TIM0_EN register

Output comparison mode:

31:25	-	Reserved
24	OC4M[3]	Output comparison 4 mode: This bit is the third bit of OC4M[3:0], Note Reference OC4M[3:0]
23:17	-	Reserved

16	OC3M[3]	Output comparison 3 mode: This bit is the third bit of OC3M[3:0], Note Reference OC3M[3:0]
15	OC4CE	Refer to the OC1CE description
14:12	OC4M[3:0]	In output comparison 4 mode, the third bit of OC4M is located at the 24 bit of TIM0_CCMR2(output comparison) Refer to the OC1M[3:0] description
11	OC4PE	Output comparison 4 Preloaded Enable: Refer to the OC1PE description
10	OC4FE	Output Comparison 4 Quick enable: Refer to the OC1FE description
9:8	CC4S[1:0]	Capture/Compare 4 Select: This bit field defines the direction of the channel (input/output) and the input used 00:The CC4 channel is configured as output 01:The CC4 channel is configured as input, and IC4 is mapped to TI4 10:The CC4 channel is configured as input, and IC4 is mapped to TI3 11:The CC4 channel is configured as input, and the IC4 is mapped to the TRC. This mode is only valid if the internal trigger input is selected via the TS bit (TIM0_SMCR register) Note: Data can be written to the CC4S bit only when the channel is closed (CC4E =0 in TIM0_CCER)
7	OC3CE	Output comparison 3 Cleared enable: Refer to the OC1CE description
6:4	OC3M[3:0]	Output comparison 3 mode: The third bit of OC3M is in the 16 bit of TIM0_CCMR2(output comparison) Refer to the OC1M[3:0] description
3	OC3PE	Output comparison 3 Preload Enable: Refer to the OC1PE description
2	OC3FE	Output Comparison 3 Quick enable: Refer to the OC1FE description
1:0	CC3S[1:0]	Capture/Compare 3 Select: This bit field defines the direction of the channel (input/output) and the input used 00:The CC3 channel is configured as output 01:The CC3 channel is configured as input, and IC4 is mapped to TI3 10:The CC3 channel is configured as input, and IC4 is mapped to TI4 11:The CC3 channel is configured as input, and IC3 is mapped to the TRC. This mode is only valid if the internal trigger input is selected via the TS bit (TIM0_SMCR register) Note: Data can be written to the CC3S bit only when the channel is closed (CC3E=0, CC3NE=0 in TIM0_CCER and has been updated)

11.4.13. TIM0 capture/compare enable registers (TIM0_CCER)

Address offset: 0x20

Reset value: 0x0000 0000

31:22	21	20	19	18	17	16
Res.	CC6P	CC6E	Res.		CC5P	CC5E
	RW	RW			RW	RW

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CC4NP	Res.	CC4P	CC4E	CC3NP	CC3NE	CC3P	CC3E	CC2NP	CC2NE	CC2P	CC2E	CC1NP	CC1NE	CC1P	CC1E
RW		RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

31:22	-	Reserved
21	CC6P	Capture/compare 6 Output polarity: Refer to the CC1P description
20	CC6E	Capture/compare 6 Output Enable: Refer to the CC1E description
19:18	-	Reserved
17	CC5P	Capture/compare 5 Output polarity: Refer to the CC1P description
16	CC5E	Capture/compare 5 Output Enable: Refer to the CC1E description
15	CC4NP	Capture/compare 4 Complementary output polarity: Refer to the CC1NP description
14	-	Reserved
13	CC4P	Capture/compare 4 output polarity: Refer to the CC1Pdescription
12	CC4E	Capture/compare 4 Output Enable: Refer to the CC1E description
11	CC3NP	Capture/compare 3 Complementary output polarity: Refer to the CC1NP description
10	CC3NE	Capture/compare 3 Complementary output Enable: Refer to the CC1NE description
9	CC3P	Capture/compare 3 Output polarity: Refer to the CC1P description
8	CC3E	Capture/compare 3 Output Enable: Refer to the CC1E description
7	CC2NP	Capture/compare 2 Complementary output polarity: Refer to the CC1NP description
6	CC2NE	Capture/compare 2 Complementary output Enable: Refer to the CC1NE description
5	CC2P	Capture/compare 2 Output polarity:

		Refer to the CC1P description
4	CC2E	Capture/compare 2 Output Enable: Refer to the CC1E description
3	CC1NP	Capture/compare 1 Complementary output polarity: **The CC1 channel is configured as output:** 0:OC1N High level is valid 1:OC1N Low level is valid **The CC1 channel is configured as input:** This bit is used in conjunction with CC1P to define the polarity of TI1FP1 and TI2FP1. Refer to the CC1P description
2	CC1NE	Capture/compare 1 Complementary output Enable: 0:Off---- OC1N is not activated. The OC1N level is a function of the MOE, OSSI, OSSR, OIS1, OIS1N, and CC1E bits 1:On ---- Outputs the OC1N signal on the corresponding output pin, depending on the MOE, OSSI, OSSR, OIS1, OIS1N, and CC1E bits. This bit will be preloaded on channels with complementary outputs. If the CCPC position is 1 in the TIM0_CR2 register, the CC1NE significant bit gets a new value from the preloaded bit only when the commutation event is generated
1	CC1P	Capture/compare 1 output polarity: **The CC1 channel is configured as output:** 0:OC1 High level is valid 1:OC1 Low level is valid **The CC1 channel is configured as an input:** the CC1NP/CC1P bit selects the effective polarity of TI1FP1 and TI2FP1 for trigger or capture operations. [CC1NP: CC1P]: 00:Non-inverting/rising edge trigger. The circuit acts on the rising edge of TIxFP1 (performs capture or trigger operations in reset mode, external clock mode, or trigger mode), and TIxFP1 is not inverted (performs trigger operations in gated mode or encoder mode). 01:Reverse phase/falling edge trigger. The circuit acts on the falling edge ofTIxFP1 (performs capture or trigger operations in reset mode, external clock mode, or trigger mode), and TIxFP1 inverts (performs trigger operations in gated mode or encoder mode). 10:Reserved 11:Non-inverting/both rising and falling edges are triggered. The circuit acts on the rising and falling edges ofTIxFP1 (performs capture or trigger operations in reset mode, external clock mode, or trigger mode), and TIxFP1 is not inverted (performs trigger operations in gated mode). This configuration cannot be used in encoder mode. Note: As soon as LOCK (the LOCK bit in the TIM0_BDTR register) level 2 or 3 is programmed, this bit immediately becomes unwritable. This bit will be preloaded on channels with complementary outputs. If the CCPC position is 1 in the TIM0_CR2 register, the CC1P significant gets a new value from the

		preloaded bit only when the commutation event is generated
0	CC1E	Capture/compare 1 output Enable: **The CC1 channel is configured as output:** 0:Off -- OC1 is not activated. The OC1 level is a function of the MOE, OSSI, OSSR, OIS1, OIS1N, and CC1NE bits. 1:On - The OC1 signal is output to the corresponding output pin, depending on the MOE, OSSI, OSSR, OIS1, OIS1N, and CC1NE bits **The CC1 channel is configured as an input:** the counter value is captured into the capture/compare register 1(TIM0_CCR1) 0:No capture 1:Enable capture This bit will be preloaded on channels with complementary outputs. If the CCPC position is 1 in the TIM0_CR2 register, the CC1E significant bit gets a new value from the preloaded bit only when the commutation event is generated

11.4.14. TIM0 counter (TIM0_CNT)

Address offset: 0x24

Reset value: 0x0000 0000

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

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

31	UIFCPY	UIF copy: This bit is a read-only copy of the UIF bit in the TIM0_SR register If the UIFREMAP bit in TIM0_CR1 is reset, bit 31 Reserved is, read it as 0
30:16	-	Reserved
15:0	CNT[15:0]	Count the value. Write to the register is prohibited during the counter count When the counter has not started counting, you can modify the initial counter value by writing the TIM0_CNT register. Reading process: 1. The RDCNTF bit ofthe TIM0_SR register is written to 1, and the counter will be latched to count the value. 2. Read the TIM0_SR register and check the RDCNTF bit. When the RDCNTF bit is 0, it indicates that the counter count value latch is complete. 3. The TIM0_CNT register is read, and the value read is the value desired by the current read flow.

11.4.15. TIM0 pre-divider (TIM0_PSC)

Address offset: 0x28

Reset value: 0x0000 0000

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

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

31:16	-	Reserved
15:0	PSC[15:0]	Pre-divider value: Counter clock frequency (CK_CNT) = f_{CK_PSC}/(PSC[15:0] +1) The PSC contains the value to be loaded into the active predivider register each time an update event occurs (including when the counter is cleared by the UG bit in the TIM0_EGR register or by triggering the controller when configured in "reset mode")

11.4.16. TIM0 automatic reload register (TIM0_ARR)

Address offset: 0x2C

Reset value: 0x0000 FFFF

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

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

31:16	-	Reserved
15:0	ARR[15:0]	Automatic overload values: ARR is the value to be loaded into the actual automatic reload register The counter does not work when the automatic overload value is empty

11.4.17. TIM0 repeat counter register (TIM0_RCR)

Address offset: 0x30

Reset value: 0x0000 0000

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

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

REP[15:0]
RW

31:16	-	Reserved
15:0	REP[15:0]	Repeat counter value: When the pre-loaded register is enabled, the user can set the update frequency of the comparison register through these bits (that is, periodically transfer data from the pre-loaded register to the active register). When the update interrupt is enabled, you can also set the generation rate of the update interrupt. Each time the subtracting counter associated with REP_CNT counts to 0, an update event is generated and the counter starts counting again from the REP value. Since REP_CNT overloads the REP value only when the repeat update event U_RC is generated, it does not matter what value is written to the TIM0_RCR register until the next repeat update event is generated. This means that PWM mode (REP+1) is equivalent to: The number of PWM cycles in edge-aligned mode. The number of PWM half-cycles in center alignment mode.

11.4.18. TIM0 capture/compare register 1 (TIM0_CCR1)

Address offset: 0x34

Reset value: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR1[15:0]															
RW															

31:16	-	Reserved
15:0	CCR1[15:0]	Capture/compare 1 values: **If channel CC1 is configured to output:** CCR1 is the value to be loaded into the valid capture/compare 1 register (the preloaded value). If the OC1PE bit in the TIM0_CCMR1 register is not used to preload the function, the value takes effect immediately. Otherwise it only takes effect when an update event occurs (copied to the actual active capture/compare register 1) The actual capture/compare register contains the value to be compared to the counter TIM0_CNT and signaled on the OC1 output. **If channel CC1 is configured as an input:** CCR1 captures the counter value for the previous input when the 1 event (IC1) occurred. The TIM0_CCR1 register can only be read and cannot be programmed

11.4.19. TIM0 capture/compare register 2(TIM0_CCR2)

Address offset: 0x38

Reset value: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR2[15:0]															
RW															

31:16	-	Reserved
15:0	CCR2[15:0]	Capture/compare 2 values: **If the channel CC2 is configured to output:** CCR2 is the value to be loaded into the valid capture/compare 2 register (the preloaded value). If the OC2PE bit in the TIM0_CCMR1 register is not used to preload the function, the value takes effect immediately. Otherwise it only takes effect when an update event occurs (copied to the actual active capture/compare register 2) The actual capture/compare register contains the value to be compared to the counter TIM0_CNT and signaled on the OC2 output. **If channel CC2 is configured as an input:** CCR2 captures the counter value for the previous input when the 2 event (IC2) occurred. The TIM0_CCR2 register can only be read and cannot be programmed

11.4.20. TIM0 capture/compare register 3 (TIM0_CCR3)

Address offset: 0x3C

Reset value: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR3[15:0]															
RW															

31:16	-	Reserved
15:0	CCR3[15:0]	Capture/compare 3 values: **If the channel CC3 is configured to output:** CCR3 is the value to be loaded into the valid capture/compare 3 register (the preloaded value). If the OC3PE bit in the TIM0_CCMR2 register is not used to preload the function, the value takes effect immediately. Otherwise it only takes effect when an update event occurs (copied to the actual active capture/compare register 3) The actual

		capture/compare register contains the value to be compared to the counter TIM0_CNT and signaled on the OC3 output. **If channel CC3 is configured as an input:** CCR3 captures the counter value for the previous input when the 3 event (IC3) occurred. The TIM0_CCR3 register can only be read and cannot be programmed

11.4.21. TIM0 capture/compare register 4 (TIM0_CCR4)

Address offset: 0x40

Reset value: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR4[15:0]															
RW															

31:16	-	Reserved
15:0	CCR4[15:0]	Capture/compare 4 values: **If the channel CC4 is configured to output:** CCR4 is the value to be loaded into the valid capture/compare 4 register (the preloaded value). If the OC4PE bit in the TIM0_CCMR2 register is not used to preload the function, the value takes effect immediately. Otherwise it only takes effect when an update event occurs (copied to the actual active capture/compare register 4) The actual capture/compare register contains the value to be compared to the counter TIM0_CNT and signaled on the OC4 output. **If channel CC4 is configured as an input:** CCR4 captures the counter value for the previous input when the 4 event (IC4) occurred. The TIM0_CCR4 register can only be read and cannot be programmed

11.4.22. TIM0 brake and dead zone registers (TIM0_BDTR)

Address offset: 0x44

Reset value: 0x0000 0000

31 30	29	28	27	26	25	24	23 22 21 20	19 18 17 16
Res.	BK2BID	BKBID	BK2DSRM	BKDSRM	BK2P	BK2E	BK2F	BKF
	RW	RW	RW	RW	RW	RW	RW	RW

15	14	13	12	11	10	9 8	7 6 5 4 3 2 1 0
MOE	AOE	BKP	BKE	OSSR	OSSI	LOCK	DTG
RW	RW	RW	RW	RW	RW	RW	RW

31:30	-	Reserved
29	BK2BID	Brake 2 bidirectional: Refer to BKBID description
28	BKBID	Brake bidirectional: 0:The brake input BRK is the input mode 1:The brake input BRK is bidirectional In bidirectional mode (BKBID=1), the brake input is configured as input mode and open-drain output mode. Any activated brake event causes the brake input to be logically low to indicate to external devices that an internal brake event has occurred. Note: As long as LOCK (LOCK bit in TIM0_BDTR register) level 1 is programmed, this bit cannot be modified. Note: Any write operation to this bit takes effect only after a delay of 1 AHB clock cycle.
27	BK2DSRM	Brake 2 Release: Refer to BKDSRM description
26	BKDSRM	Brake release: 0:Activate brake input BRK 1:Release brake input BRK When the brake source is activated, this bit is cleared by the hardware. The BKDSRM position 1 must be set by software to release the bidirectional output control (the open leakage output is in a high resistance state), and then the bit is constantly polling until it is reset by the hardware, indicating that the fault condition has disappeared Note: Any write operation to this bit takes effect only after a delay of 1 AHB clock cycle.
25	BK2P	Brake 2 polarity: 0:Brake input BRK2 is active at low level 1:Brake input BRK2 is active at high level Note: As long as LOCK (LOCK bit in TIM0_BDTR register) level 1 is programmed, this bit cannot be modified. Note: Any write operation to this bit takes effect only after a delay of 1 AHB clock cycle
24	BK2E	Brake 2 Enable: 0:Brake input BRK2 is prohibited 1:Enable brake input BRK2 Note: BRK2 must only be used when OSSR=OSSI=1. Note: After the LOCK (LOCK bit in the TIM0_BDTR register) level 1 is programmed, this bit cannot be modified. Note: Any write operation to this bit takes effect only after a delay of 1 AHB clock cycle.
23:20	BK2F	Brake 2 filter: This bit field defines the sampling frequency of the BKR2 signal and the bandwidth of the digital filter applicable to BKR2. The digital filter consists of an event counter, where each N consecutive events is considered a valid output edge: 0000: No filter, sampling at f_{DTS} frequency 0001: $f_{SAMPLING} = f_{TIM0\_CLK}$, N=2 0010: $f_{SAMPLING} = f_{TIM0\_CLK}$, N=4

		0011: $f_{SAMPLING} = f_{TIM0\_CLK}$, N=8 0100: $f_{SAMPLING} = f_{DTS}/2$, N=6 0101: $f_{SAMPLING} = f_{DTS}/2$, N=8 0110: $f_{SAMPLING} = f_{DTS}/4$, N=6 0111: $f_{SAMPLING} = f_{DTS}/4$, N=8 1000: $f_{SAMPLING} = f_{DTS}/8$, N=6 1001: $f_{SAMPLING} = f_{DTS}/8$, N=8 1010: $f_{SAMPLING} = f_{DTS}/16$, N=5 1011: $f_{SAMPLING} = f_{DTS}/16$, N=6 1100: $f_{SAMPLING} = f_{DTS}/16$, N=8 1101: $f_{SAMPLING} = f_{DTS}/32$, N=5 1110: $f_{SAMPLING} = f_{DTS}/32$, N=6 1111: $f_{SAMPLING} = f_{DTS}/32$, N=8 Note: After the LOCK (LOCK bit in the TIM0_BDTR register) level 1 is programmed, this bit cannot be modified. Note: This bit is set before the TIM0_EN register is configured
19:16	BKF	Brake filter: This bit field defines the sampling frequency of the BRK signal and the bandwidth of the digital filter applicable to the BRK. The digital filter consists of an event counter, where each N consecutive events is considered a valid output edge: 0000: No filter, sampling at f_{DTS} frequency 0001: $f_{SAMPLING} = f_{TIM0\_CLK}$, N=2 0010: $f_{SAMPLING} = f_{TIM0\_CLK}$, N=4 0011: $f_{SAMPLING} = f_{TIM0\_CLK}$, N=8 0100: $f_{SAMPLING} = f_{DTS}/2$, N=6 0101: $f_{SAMPLING} = f_{DTS}/2$, N=8 0110: $f_{SAMPLING} = f_{DTS}/4$, N=6 0111: $f_{SAMPLING} = f_{DTS}/4$, N=8 1000: $f_{SAMPLING} = f_{DTS}/8$, N=6 1001: $f_{SAMPLING} = f_{DTS}/8$, N=8 1010: $f_{SAMPLING} = f_{DTS}/16$, N=5 1011: $f_{SAMPLING} = f_{DTS}/16$, N=6 1100: $f_{SAMPLING} = f_{DTS}/16$, N=8 1101: $f_{SAMPLING} = f_{DTS}/32$, N=5 1110: $f_{SAMPLING} = f_{DTS}/32$, N=6 1111: $f_{SAMPLING} = f_{DTS}/32$, N=8 Note: After the LOCK (LOCK bit in the TIM0_BDTR register) level 1 is programmed, this bit cannot be modified. Note: This bit is set before the TIM0_EN register is configured
15	MOE	Main output enable: This bit is cleared asynchronously by the hardware as long as the brake input (BRK or BRK2) is in a valid state. This bit is set to 1 by the software, or can be automatically set to 1

		according to the AOE bit status. This bit is valid only for channels configured as output. 0:Responds to braking events (2). When OC and OCN outputs are disabled in response to braking events or when 0 is written to MOE: OC and OCN outputs are disabled or forced to idle, depending on the OSSI bit. 1:If the corresponding enable bits for OC and OCN outputs (CCxE and CCxNE bits in the TIM0_CCER register) are set to 1, OC and OCN outputs are enabled.
14	AOE	Automatic output enable: 0:MOE can only be set to 1 by software 1:MOE can be set to 1 by the software or automatically when the next update event occurs (this bit is invalid if the brake inputs BRK and BRK2 are valid) Note: As long as LOCK (LOCK bit in TIM0_BDTR register) level 1 is programmed, this bit cannot be modified.
13	BKP	Brake polarity: 0:Brake input BRK is active at low level 1:Brake input BRK is active at high level Note: As long as LOCK (LOCK bit in TIM0_BDTR register) level 1 is programmed, this bit cannot be modified. Note: Any write operation to this bit takes effect only after a delay of one APB clock cycle.
12	BKE	Brake enable: This bit enables complete brake protection (including all sources connected to the bk_acth and corresponding BKIN sources, as shown in the brake and brake 2 circuits). 0:Disable braking function 1:The brake function was enabled Note: After the LOCK (LOCK bit in the TIM0_BDTR register) level 1 is programmed, this bit cannot be modified. Note: Any write operation to this bit takes effect only after a delay of 1 AHB clock cycle.
11	OSSR	Select the closed state in running mode: This bit applies to channels configured for output mode and with complementary outputs when MOE= 1. If there is no complementary output in the timer, there is no OSSR 0:In an invalid state, disable OC/OCN output (the timer releases output control and the GPIO logic that forces the high resistance state takes over). 1:In an invalid state, once CCxE= 1 or CCxNE= 1, the OC/OCN output is enabled and set to an invalid level (the output is still controlled by the timer). Note: After the LOCK (LOCK bit in the TIM0_BDTR register) level 2 is programmed, this bit cannot be modified.
10	OSSI	Off status selection in idle mode: When MOE=0 due to a braking event or software write operation, this bit acts on the channel configured as output. 0:In an invalid state, disable OC/OCN output (the timer releases output control and the GPIO logic that forces the high resistance state takes over). 1:When in an invalid state, first force the OC/OCN output to its invalid level, and then force it to a predefined state after the dead zone. The timer always controls the output.

		Note: After the LOCK (LOCK bit in the TIM0_BDTR register) level 2 is programmed, this bit cannot be modified.
9:8	LOCK	Lock configuration: These bits are used to provide write protection against software errors. 00:Disable lock ---- do not provide write protection for any bits 01:Lock level 1, The OISx and OISxN bits in the TIM0_CR2 register and the BK2BID, BKBID, BK2DSRM, BKDSRM, BK2P, BK2E, BK2F[3:0], BKF[3:0], AOE, BKP, BKE, and DTG bits in the TIM0_BDTR register cannot be executed at this time Write operation. 10:At lock level 2, write operations cannot be performed on the applicable bits in lock level 1, the CC polarity bits (CCxP/CCxNP bits in the TIM0_CCER register, as long as the relevant channel is configured to output via the CCxS bits), and the OSSR and OSSI bits. 11:Lock level 3, at this time cannot write to each of the applicable members in lock level 2, CC control bits (OCxM and OCxPE bits in the TIM0_CCMRx register, as long as the relevant channels are configured as outputs through the CCxS bits). Note: Write to the LOCK bit only once after reset. After writing to the TIM0_BDTR register, its contents are frozen until the next reset.
7:0	DTG	Configure dead zone generator: This bit field defines the duration of the dead zone inserted between complementary outputs. DT corresponds to this duration. DTG[7:5]=0xx=>DT=DTG[7:0] *t_{dtg}, t_{dtg}=t_{DTS}. DTG[7:5]=10x=>DT=(64+DTG[5:0]) *t_{dtg}, t_{dtg}=2*t_{DTS}. DTG[7:5]=110=>DT=(32+DTG[4:0]) *t_{dtg}, t_{dtg}=8*t_{DTS}. DTG[7:5]=111=>DT=(32+DTG[4:0]) *t_{dtg}, t_{dtg}= 16*t_{DTS}. Example: If t_{DTS}= 125ns(8MHz), the possible dead zone values are: 0 to 15875ns (step size is 125ns) 16μs to 31750ns (step size is 250ns) 32μs to 63μs (step size is 1μs) 64μs to 126μs (step size is 2μs) Note: As long as LOCK (the LOCK bit in the TIM0_BDTR register) level 1, 2, or 3 is programmed, this bit field cannot be modified. Note: This bit is set before the TIM0_EN register is configured

11.4.23. TIM0 DMA control register (TIM0_DCR)

Address offset: 0x48

Reset value: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.			DBL					Res.			DBA				
			RW								RW				

31:13	-	Reserved
12:8	DBL	DMA continuous transmission length: This 5-bit vector defines the DMA transfer length (when a read or write access is made to the TIM0_DMAR address, the timer makes one successive pass), that is, the number of passes. Can be transmitted by half word or byte 00000:Once transmission 00001:Double transmission 00010:Tertiary transmission ... 10001:18 transmission Example: DBL = 7 bytes and DBA = TIM0_CR1 ---- If DBL = 7 bytes and DBA = TIM0_CR1 represents the address of the bytes to be transferred, the address to be transferred should be given by the following formula: (TIM0_CR1address) + DBA + (DMA index),And DMA index = DBL In this case, add 7 bytes to (TIM0_CR1 address) + DBA to get the source/destination address of the data to be copied. In this case, data is sent to seven registers starting at the following address: (TIM0_CR1 address) + DBA Depending on the configuration of the DMA data size, several things can happen: ---- If the DMA data size is configured as a half-word, 16 bits of data are transferred to each of the seven registers ---- If you configure the DMA data size in bytes, you will also send data to seven registers: the first contains the first MSB byte, the second contains the first LSB byte, and so on. Therefore, when using the transfer timer, you must also specify the size of the data to be transferred by DMA
7:5	-	Reserved
4:0	DBA	DMA base address: The 5-bit vector defines the base address of the DMA transfer (for read/write access via the TIM0_DMAR address) as the offset calculated from the TIM0_CR1 register address: 00000:TIM0_CR1 00001:TIM0_CR2 00010:TIM0_SMCR …

11.4.24. TIM0 full transport DMA address(TIM0_DMAR)

Address offset: 0x4C

Reset value: 0x0000 0000

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

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

31:0	DMAB[31:0]	DMA continuous transfer register: Reading or writing to the DMAR register will access registers at the following addresses: (TIM0_CR1 address) + (DBA + DMA index) * 4 The TIM0_CR1 address is the address of control register 1; DBA is the DMA base address configured in the TIM0_DCR register The DMA index is automatically controlled by the DMA transfer and ranges from 0 to DBL (DBL configured in the TIM0_DCR register)

11.4.25. TIM0 capture/compare mode register 3(TIM0_CCMR3)

Address offset: 0x54

Reset value: 0x0000 0000

31 30 29 28 27 26 25	24	23 22 21 20 19 18 17	16
Res.	OC6M[3]	Res.	OC5M[3]
	RW		RW

15	14 13 12	11	10	9 8	7	6 5 4	3	2	1 0
OC6CE	OC6M[3:0]	OC6PE	OC6FE	Res.	OC5CE	OC5M[3:0]	OC5PE	OC5FE	Res.
RW	RW	RW	RW		RW	RW	RW	RW	

Output comparison mode:

31:25	-	Reserved
24	OC6M[3]	Output comparison 6 mode: This bit is the third bit of OC6M[3:0], refer to OC6M[3:0] description
23:17	-	Reserved
16	OC5M[3]	Output comparison 5 mode: This bit is the third bit of OC5M[3:0], refer to OOC5M[3:0] description
15	OC6CE	Output comparison 6 Clear enable: refer to OC1CE description
14:12	OC6M[3:0]	Output comparison 6 mode: The third bit of OC6M is in the 24 bit of TIM0_CCMR3(output comparison) refer to OC1M[3:0] description
11	OC6PE	Output comparison 6 Preloaded Enable: refer to OC1PE description
10	OC6FE	Output Comparison 6 Quick enable: refer to OC1FE description
9:8	-	Reserved
7	OC5CE	Output comparison 5 Clear Enable: refer to OC5CE description

6:4	OC5M[3:0]	Output comparison 5 mode: The third bit of the OC5M is located at the 16-bit of TIM0_CCMR3(output comparison) refer to OC1M[3:0] description
3	OC5PE	Output comparison 5 Preloaded Enable: refer to OC1PE description
2	OC5FE	Output Comparison 5 Quick enable: refer to OC1FE description
1:0	-	Reserved

11.4.26. TIM0 capture/compare register 5(TIM0_CCR5)

Address offset: 0x58

Reset value: 0x0000 0000

31	30	29	28:16
GC5C3	GC5C2	GC5C1	Res.
RW	RW	RW	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR5															
RW															

31	GC5C3	Channel 5 and channel 3 groups: Distortion on channel 3 output: 0:OC5REF has no effect on OC3REFC 1:OC3REFC is the logical and operational result of OC3REFC and OC5REF This bit can take effect immediately, or it can be preloaded and executed after the update event (if preloaded is selected in TIM0_CCMR2). Note: This distortion can be applied to combined PWM signals
30	GC5C2	Channel 5 and channel 2 groups: Distortion on channel 2 output: 0:OC5REF has no effect on OC2REFC 1:OC2REFC is the logical and operational result of OC2REFC and OC5REF This bit can take effect immediately, or it can be preloaded and executed after the update event (if preloaded is selected in TIM0_CCMR1). Note: This distortion can be applied to combined PWM signals
29	GC5C1	Channel 5 and channel 1 groups: Distortion on channel 1 output: 0:OC5REF has no effect on OC1REFC 1:OC1REFC is the logical and operational result of OC1REFC and OC5REF This bit can take effect immediately, or it can be preloaded and executed after the update event (if preloaded is selected in TIM0_CCMR1). Note: This distortion can be applied to combined PWM signals

28:16	-	Reserved
15:0	CCR5	Capture/compare 5 values: CCR5 is the preloaded value of the capture/compare register 5. If the OC5PE bit in the TIM0_CCMR3 register is not used to preload the function, the value takes effect immediately. Otherwise it only takes effect when an update event occurs (copy to a valid capture/compare register 5).The effective capture/compare register contains the value to be compared with the counter TIM0_CNT and signaled on the OC5 output.

11.4.27. TIM0 capture/compare register 6 (TIM0_CCR6)

Address offset: 0x5C

Reset value: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR6[15:0]															
RW															

31:16	-	Reserved
15:0	CCR6	Capture/compare 6 values: CCR5 is the preloaded value of capture/compare register 6 If the OC6PE bit in the TIM0_CCMR3 register is not used to preload the function, the value takes effect immediately. Otherwise it only takes effect when an update event occurs (copy to a valid capture/compare register 6). The effective capture/compare register contains the value to be compared with the counter TIM0_CNT and signaled on the OC6 output.

11.4.28. TIM0 reuse function option register 1 (TIM0_AF1)

Address offset: 0x60

Reset value: 0x0000 0001

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.						BKINP	Res.								BKINE
						RW									RW

31:10	-	Reserved
9	BKINP	BRK BKIN Input polarity:

		This bit selects the input level sensitivity of the BKIN multiplexing function and must be programmed with the BKP polarity bit. 0:The BKIN input is active at a high level 1:The BKIN input is active at a low level Note: As long as LOCK (LOCK bit in TIM0_BDTR register) level 1 is programmed, this bit cannot be modified
8:1	-	Reserved
0	BKINE	BRK BKIN Input Enable: This bit enables the BKIN multiplexing function of the BRK input of the timer. The BKIN input performs "or" operations with other BRK sources. 0:BKIN input disable 1:BKIN input enable Note: As long as LOCK (LOCK bit in TIM0_BDTR register) level 1 is programmed, this bit cannot be modified

11.4.29. TIM0 reuse function option register 2(TIM0_AF2)

Address offset: 0x64

Reset value: 0x0000 0001

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.						BK2INP	Res.								BK2INE
						RW									RW

31:10	-	Reserved
9	BK2INP	BRK2 BKIN Input polarity: This bit selects the BKIN2 multiplexed function input level sensitivity and must be programmed with the BKP2 polarity bit. 0:The BKIN2 input is active at a high level 1:The BKIN2 input is active at a low level Note: As long as LOCK (LOCK bit in TIM0_BDTR register) level 1 is programmed, this bit cannot be modified
8:1	-	Reserved
0	BK2INE	BRK2 Input Enable: This bit enables the BKIN2 multiplexing function of the BRK2 input of the timer. The BKIN2 input performs "or" operations with other BRK2 sources. 0:BKIN2 input disable 1:BKIN2 input enable Note: As long as LOCK (LOCK bit in TIM0_BDTR register) level 1 is programmed, this bit cannot be modified

11.4.30. TIM0 input/output control register (TIM0_DIR)

Address offset: 0x68

Reset value: 0x0000 0070

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

15:7	6	5	4	3	2	1	0
Res.	CH3NE	CH2NE	CH1NE	CH4DIR	CH3DIR	CH2DIR	CH1DIR
	RW	RW	RW	RW	RW	RW	RW

31:7	-	Reserved
6	CH3NE	TIM0_CH3N Output control enable: 0:Output enable 1:Output disable
5	CH2NE	TIM0_CH2N Output control enable: 0:Output enable 1:Output disable
4	CH1NE	TIM0_CH1N Output control enable: 0:Output enable 1:Output disable
3	CH4DIR	TIM0_CH4 direction control: 0:Output 1:Input
2	CH3DIR	TIM0_CH3 direction control: 0:Output 1:Input
1	CH2DIR	TIM0_CH2 direction control: 0:Output 1:Input
0	CH1DIR	TIM0_CH1 direction control: 0:Output 1:Input

Note: This register is set before the TIM0_EN register is configured.

11.4.31. TIM0 module enable register (TIM0_EN)

Address offset: 0x6C

Reset value: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.															TIM0EN
															RW

31:1	-	Reserved
0	TIM0EN	TIM0 module work enable: 1:Module work 0:module Stop working

11.5. Note

1. When the IO port is reused as the brake function, the state is unstable (floating input), and a peripheral circuit must provide a stable level.

Chapter 12 Universal timer(TIMx)(x = 1/2/3/4/5/7)

12.1. TIMx Overview

The Universal timer (TIMx) contains a 16-bit automatic overload counter that is driven by a programmable pre-divider.

12.2. TIMx functional characteristics

- 16-bit incrementing automatic reload counter
- 16-bit programmable (can be modified in real time) pre-divider, counter clock frequency divider factor is 1 ~ 65536 arbitrary value
- 1 independent channel that can be used for:
  - Input capture
  - Output comparison
  - PWM generation (edge mode)
  - Monopulse mode output
- An interrupt /DMA request is generated when the following events occur:
  - Update: Counter overflow, counter initialization (by software)
  - Input capture
  - Output comparison

12.3. TIMx function description

12.3.1. TIMx block diagram

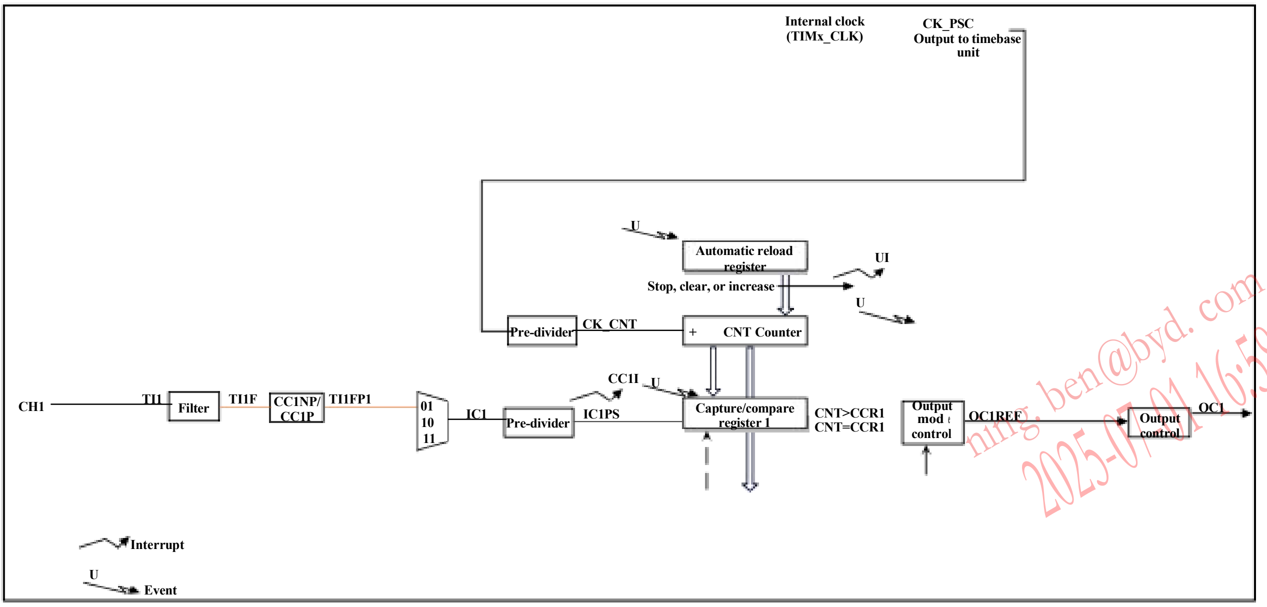


Figure 12.1 TIMx timer block diagram

12.3.2. Clock selection

Counter clocks can be supplied by the following clock sources:

- Internal clock TIMx_CLK: PLL_48MHz/LIRC 32K/XTAL(Passive crystal (32768Hz/4MHz/8MHz) and active crystal (1MHz~48MHz))

12.3.2.1. Internal clock(TIMx_CLK)

When 1 is written to the CEN bit, the clock of the pre-divider is provided by the internal clock TIMx_CLK.

12.3.3. Time base unit

The main module of the universal control timer TIMx is a 16-bit counter and its associated auto-reload register. The counter counts incrementally. The clock of the counter can be divided by a pre-divider. Counters, auto-reload registers, and pre-divider registers can be read and written by software, even when the counter is running.

Time base units include:

- Pre-divider register(TIMx_PSC)
- Counter register(TIMx_CNT)
- Automatic reload register(TIMx_ARR)

12.3.3.1. Pre-divider PSC

The pre-divider PSC has an input clock CK_PSC and an output clock CK_CNT. The input clock comes from the controller part. By setting the value of the pre-division frequency, different CK_CNT can be obtained. The actual calculation formula is as follows: CK_CNT = F_{CK_PSC}/(PSC[15:0]+1).

Because the TIMx_PSC control register has a buffer (shadow register), its value can be changed during the run, and the new predivision frequency value will take effect when the event is updated next.

12.3.3.2. Counter CNT

The counter counts incrementally. The counter is started only when CEN in the TIMx_CR1 register is set to 1.

12.3.3.3. Automatic reload register ARR

The automatically loaded register is preloaded and is accessed when a write or read operation is performed on it.

The contents ofthe TIMx_ARR register can be transferred either directly to the shadow register or each time an update event occurs, depending on the auto-reload preload enable bit (ARPE) in the TIMx_CR1 register. When the counter reaches an overflow value and the UDIS bit in the TIMx_CR1 register is 0, an update event is sent. The update event can also be generated by software.

12.3.4. Counter mode

12.3.4.1. Incremental count mode

In incremental count mode, the counter counts from 0 to an automatically overloaded value (the contents ofthe TIMx_ARR register), then counts again from 0 and generates a counter overflow event.

Update events are also generated when the TIMx_EGR register is changed to UG position 1 (either by software or using a slave mode controller).

UEV events can be disabled by software at UDIS position 1 in the TIMx_CR1 register, which avoids updating the shadow register when a new value is written to the preloaded register. No update events occur until the UDIS bit is written to 0; However, both the counter and the pre-divider counter start counting again from 0 (while the pre-divider ratio remains the same). In addition, ifthe URS bit (update request selection) in the TIMx_CR1 register is set to 1, the UG position 1 generates the update event UEV, but does not set the UIF flag to 1 (therefore, no interrupt is sent); This is to avoid both update and capture interruptions when the counter is cleared in capture mode.

When an update event occurs, all registers are updated and the update flag (UIF bit in the TIMx_SR register) is set to 1 (depending on the URS bit) :

- The automatically overloaded shadow register is updated with the pre-loaded value (TIMx_ARR)
- The preload value (the contents ofthe TIMx_PSC register) will be reloaded in the buffer of the predivider.

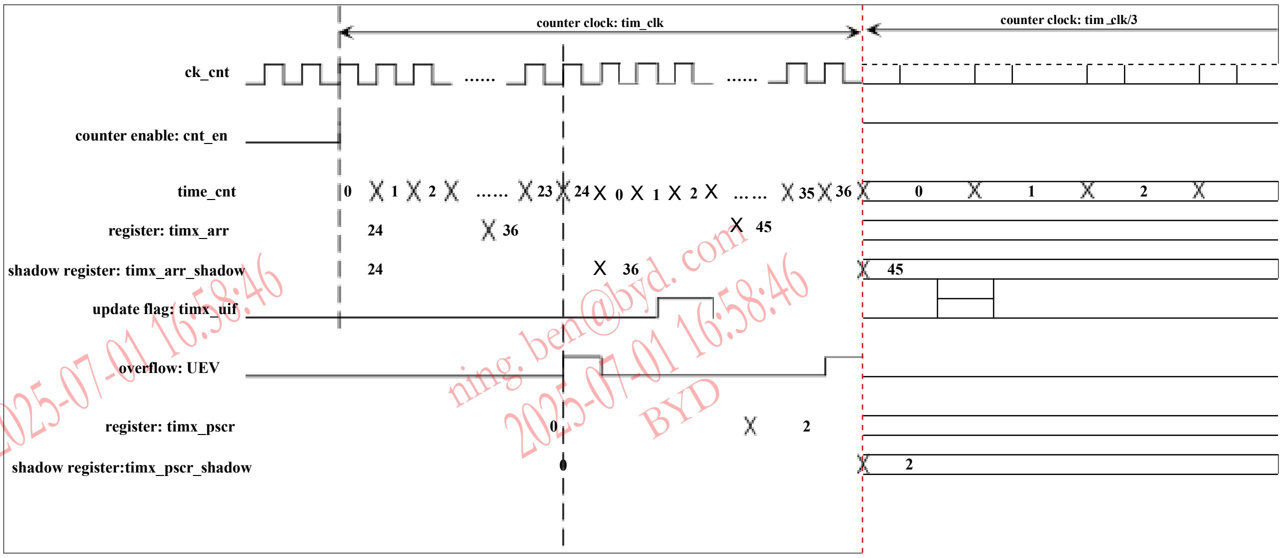


Figure 12.2 Incremental counting sequence diagram

12.3.5. ADC synchronization

The timer can generate ADC trigger events from a variety of internal signals, such as reset, enable, or compare events.

The trigger signal is emitted on the TRGO2 internal line that redirects to the ADC, and there are a total of five possible events, which can be selected by the MMS2[2:0] bit in the TIMx_CR2 register.

Note:

1. The clock of the slave peripheral (timer, ADC, etc.) that receives TRGO or TRGO2 signals must be enabled before the event can be received from the master timer; And when the trigger signal is received from the master timer, the clock frequency (pre-divider) must not be changed in real time.

2. The ADC clock must be enabled before receiving events from the master timer; The ADC clock must not be changed in real time when the trigger signal is received from the timer.

12.3.6. Capture/compare channels

Each capture/comparison channel is built around a capture/comparison register (including a shadow register), a capture input part (digital filter, multiplexing, and pre-divider), and an output part (comparator and output control).

In the input phase, the corresponding TIx input is sampled and a filtered signal TIxF is generated. The edge detector with polarity selection then generates a signal (TIxFPx) that can be used either as a trigger input from the mode controller or as a capture command; The signal is first pre-divided (ICxPS) and then entered into the capture register.

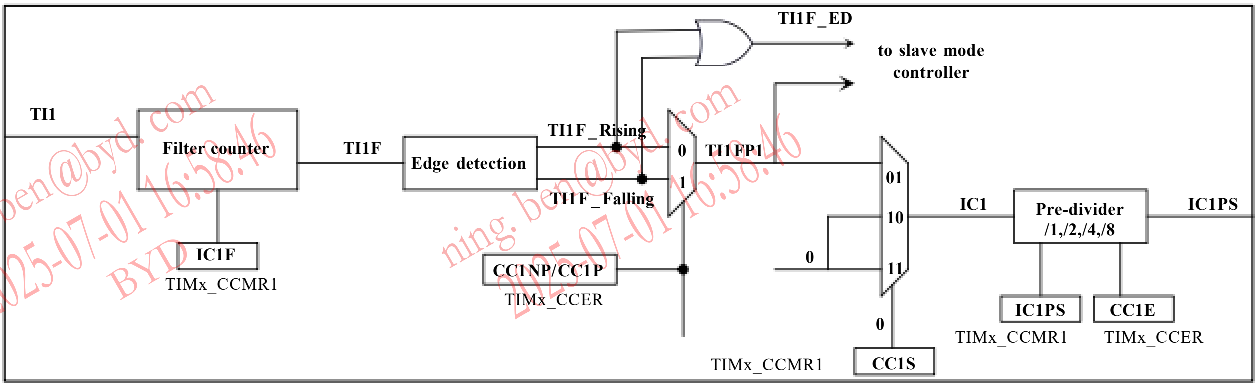


Figure 12.3 Channel 1 input phase

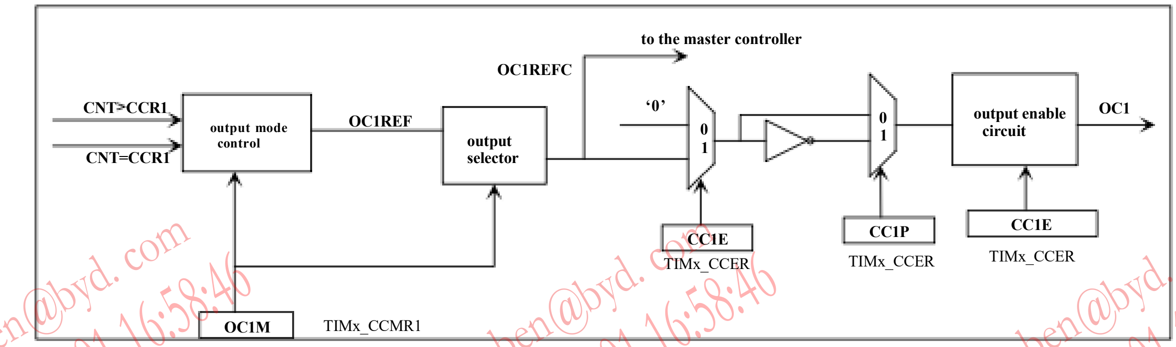


Figure 12.4 Capture/compare the output phase of a channel (channel 1)

The capture/comparison module consists of a preloaded register and a shadow register, which can always be accessed through read and write operations.

In capture mode, the capture actually takes place in the shadow register, and the contents ofthe shadow register are then copied into the preloaded register.

In comparison mode, the contents ofthe pre-loaded register are copied into the shadow register, and then the contents ofthe shadow register are compared with the counter.

12.3.7. Input capture mode

In input capture mode, the capture/compare register (TIMx_CCRx) is used to latch the counter value when the corresponding ICx signal detects a jump edge. When a capture event occurs, the corresponding CCxIF flag (TIMx_SR register) is set to 1 and an interrupt (if enabled) can be sent.If the CCxIF flag is already high when a capture event occurs, the CCxOF (TIMx_SR register) is set to 1. CCxIF clears by software by writing "0" to CCxIF or reading the captured data stored in the TIMx_CCRx register; CCxOF clears when "0" is written.

The following example shows how to capture the counter's value into TIMx_CCR1 when a rising edge appears in the TI1 input:

1. Select valid input: Configure the CC1S bit of the TIMx_CCMR1 register to write 01, configure the CC1 channel as input, and map IC1 to TI1.
2. Depending on the signal connected to the timer, configure the input filter to the required bandwidth (in this case, if the input is TI1, configure the IC1F bit ofthe TIMx_CCMR1 register).It is assumed that when the signal edge changes, the input signal jitter occurs within a maximum of 5 internal clock cycles. Therefore, the filter bandwidth needs to be set to more than 5 internal clock cycles, and the jump edge on TI1 can be determined after 8 consecutive samples with new levels are detected (sampled at the f_{TIMx_CLK} frequency). Then write 0011 to the IC1F bit ofTIMx_CCMR1.
3. Configure the CC1P and CC1NP bits ofthe TIMx_CCER register to write 0, and select the effective conversion edge rising edge on TI1
4. Configure the IC1PSC = 00 of the TIMx_CCMR1 register to disable the input of the pre-divider, and perform the capture operation when the capture input detects a valid edge.
5. Configuring the CC1E = 1 of the register TIMx_CCER allows the value of the counter to be captured into the capture register.
6. As needed, the input capture 1 interrupt can be enabled by configuring the CC1IE bit of TIMx_DIER to 1, and/or by configuring the CC1DE bit of this register to 1 to enable DMA requests.

When input capture occurs:

- When a valid jump edge occurs, the TIMx_CCRx register takes the value of the counter.
- Set the CCxIF flag to 1 (interrupt flag). If at least two consecutive catches occur, but the CCxIF flag is not cleared, then the CCxOF capture overflow flag is set to 1.
- Generates interrupts based on the CCxIE bit.
- Generate DMA requests based on CCxDE bits.

To handle duplicate captures, it is recommended to read the data before reading the capture overflow flag to avoid losing duplicate capture information that may occur after reading the capture overflow flag and before reading the data.

Note:

1. **IC interrupt can be generated by corresponding CCxG position 1 in TIMx_EGR register through software.**
2. **Configure the TIMx_CCMRx register in two configurations: configure the CCxS bit in the TIMx_CCMRx register for the first time and select the mapping of the input mode; The ICxF and ICxPSC of the TIMx_CCMRx register are configured for the second time to select filtering and predivision coefficients.**

12.3.8. Forced output mode

In the output mode (CCxS bit =00 in the TIMx_CCMRx register), each output comparison signal (OCxREF and OCx) can be forcibly set to an active or invalid level directly by the software, without considering any comparison results between the output comparison register and the counter.

To force the output comparison signal (OCxREF/OCx) to be set to an active level, 0101 is written to the OCxM bit in the corresponding TIMx_CCMRx register, OCxREF is then forced to be set to a high level (OCxREF is always active at a high level), and OCx gets the opposite value of the CCxP polarity bit.

For example, CCxP=0 (the OCx high level is valid) => Force the OCx high level.

The OCxREF signal can be forcibly set to a low level by writing 0100 to the OCxM bit in the TIMx_CCMRx register.

In any case, the comparison between the TIMx_CCRx shadow register and the counter is still performed, and the flag is allowed to be set to 1, so the corresponding interrupt and DMA requests can be sent.

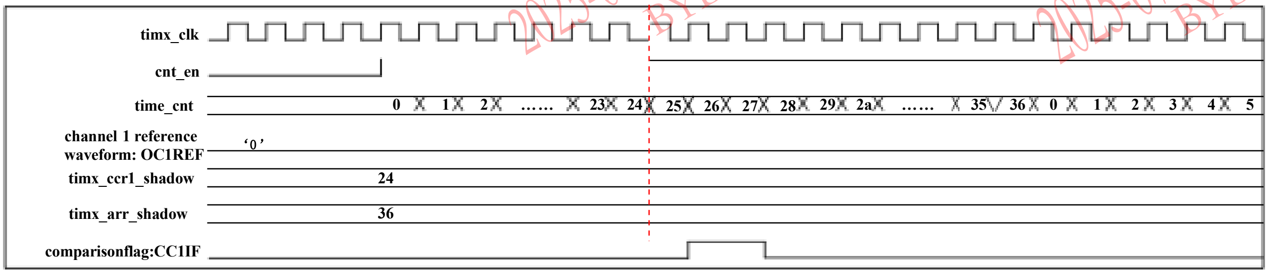


Figure 12.5 Forced output invalid level timing diagram (OCxM=0100)

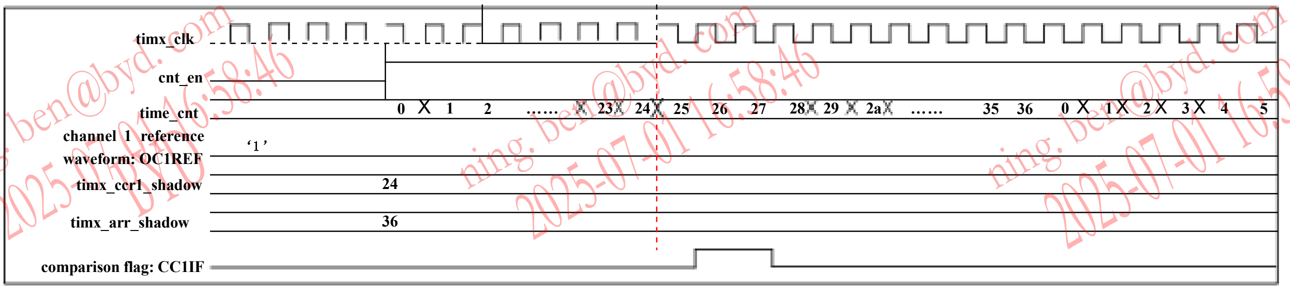


Figure 12.6 Forced output active level timing diagram (OCxM=0101)

12.3.9. Output comparison mode

This function is used to control the output waveform or indicate that a certain period of time has passed.

When there is a match between the capture/compare register and the counter, the output compare function is as follows:

- A programmable value is assigned to the corresponding output pin, defined by the output comparison mode (OCxM bits in the TIMx_CCMRx register) and the output polarity (CCxP bits in the TIMx_CCER register).When matched, the output pin can either hold its level (OCxM=0000), be set to active (OCxM=0001), invalid (OCxM=0010), or be flipped (OCxM=0011).

- Set the flag in the interrupt status register to 1 (the CCxIF bit in the TIMx_SR register).
- If the corresponding interrupt enable bit (CCxIE bit in the TIMx_DIER register) is set to 1, an interrupt is generated.
- If the corresponding enable bit (the CCxDE bit of the TIMx_DIER register) is set to 1, a DMA request is sent.

By configuring the OCxPE bit in the TIMx_CCMRx register, you can enable or disable the pre-loaded register related to the TIMx_CCRx register.

In output comparison mode, updating event UEV has no effect on OCxREF and OCx output. The accuracy of synchronization can reach one count cycle of the counter. The output comparison mode can also be used to output monopulse (in monopulse mode).

Application steps:

1. Select the counter clock (internal, external, pre-divider).
2. Write the required data in the TIMx_ARR and TIMx_CCRx registers.
3. If you want to generate an interrupt request, you need to set the CCxIE position to 1.
4. Select an output mode. For example:
   a) When CNT matches CCRx, write OCxM = 0011 to flip the OCx output pin
   b) Write OCxPE = 0 to disable the preload register
   c) Write CCxP = 0 to select the high level effective polarity
   d) Write CCxE = 1 to enable output
5. Enable the counter by placing the CEN position 1 in the TIMx_CR1 register.

The TIMx_CCRx register can be updated at any time through the software to control the output waveform, provided that the pre-loaded register is not enabled (OCxPE= "0", otherwise the TIMx_CCRx shadow register will only be updated when the next update event UEV occurs).

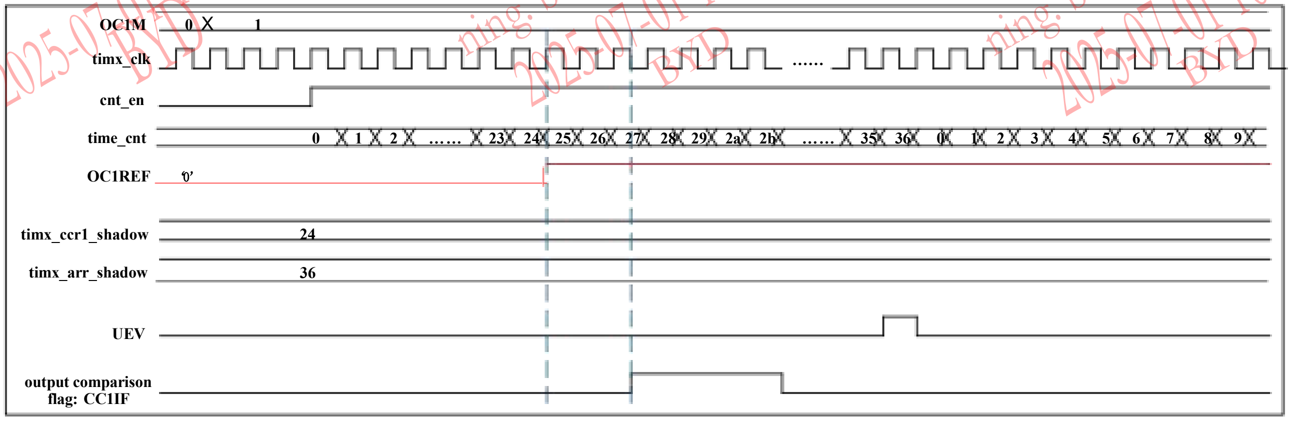

OC1M
0
1
timx_clk
......
cnt_en
time_cnt
0 1 2 ...... 23 24 25 26 27 28 29 2a 2b 35 36 0 1 2 3 4 5 6 7 8 9
OC1REF
'0'
timx_ccr1_shadow
24
timx_arr_shadow
36
UEV
output comparison flag: CC1IF

Figure 12.7 Output comparison mode (OCxM=0001)

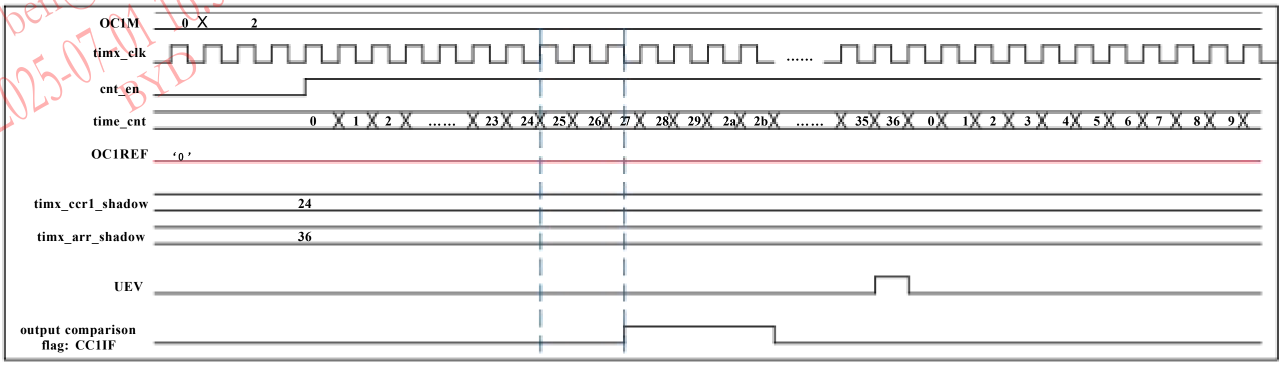

OC1M
0
2
timx_clk
......
cnt_en
time_cnt
0 1 2 ...... 23 24 25 26 27 28 29 2a 2b 35 36 0 1 2 3 4 5 6 7 8 9
OC1REF
'0'
timx_ccr1_shadow
24
timx_arr_shadow
36
UEV
output comparison flag: CC1IF

Figure 12.8 Output comparison mode (OCxM=0010)

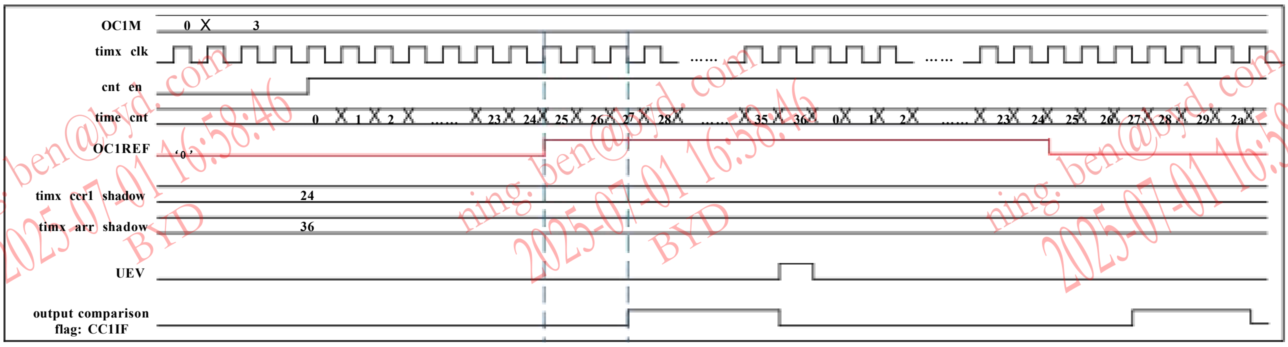

OC1M
0
3
timx clk
......
cnt en
time cnt
0 1 2 ...... 23 24 25 26 27 28 ...... 35 36 0 1 2 ...... 23 24 25 26 27 28 29 2a
OC1REF
'0'
timx ccr1 shadow
24
timx arr shadow
36
UEV
output comparison flag: CC1IF

Figure 12.9 Output comparison mode (OCxM=0011)

12.3.10. PWM mode

The pulse width modulation mode can generate a signal whose frequency is determined by the TIMx_ARR register value and whose duty cycle is determined by the TIMx_CCRx register value.

Each channel can independently select the PWM mode (each OCx output corresponds to a PWM), just write "0110" (PWM mode 1) or "0111" (PWM mode 2) to the OCxM bit of the TIMx_CCMRx register; The corresponding preload register must be enabled by enabling the OCxPE position 1 in the TIMx_CCMRx register, and finally by enabling the automatic reload register to be preloaded by enabling the ARPE position 1 in the TIMx_CR1 register.

Since the preloaded registers are only transferred to the shadow registers when an update event occurs, all registers must be initialized by placing UG position 1 in the TIMx_EGR register before starting the counter.

The polarity of OCx can be set by software in the CCxP bit in the TIMx_CCER register. It can be set to high active or low active, and OCx output can be enabled via CCxE.

12.3.10.1. PWM edge alignment mode

Incremental count configuration

The following takes PWM mode 1 as an example. As long as TIMx_CNT < TIMx_CCRx, PWM reference signal OCxREF is high level, otherwise it is low level. If the comparison value in TIM_CCRx is greater than the automatic overload value (in TIMx_ARR), OCxREF remains "1"; If the comparison value is 0, OCxREF remains "0".

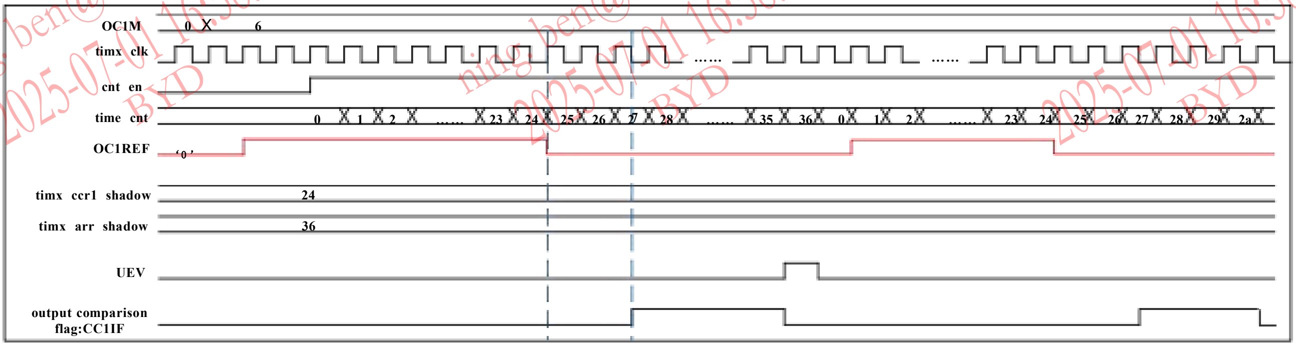

OC1M
0
6
timx clk
……
cnt en
time cnt
0 1 2 …… 23 24 25 26 27 28 …… 35 36 0 1 2 …… 23 24 25 26 27 28 29 2a
OC1REF
'0'
timx ccr1 shadow
24
timx arr shadow
36
UEV
output comparison flag:CC1IF

Figure 12.10 PWM edge alignment mode 1 (OCxM=0110)

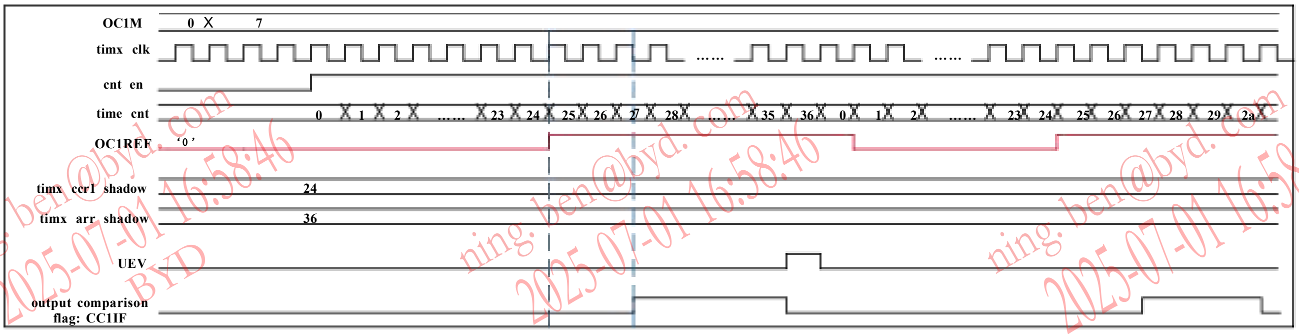

OC1M
0
7
timx clk
……
cnt en
time cnt
0 1 2 …… 23 24 25 26 27 28 …… 35 36 0 1 2 …… 23 24 25 26 27 28 29 2a
OC1REF
'0'
timx ccr1 shadow
24
timx arr shadow
36
UEV
output comparison flag: CC1IF

Figure 12.11 PWM edge alignment mode 2 (OCxM=0111)

12.3.11. Single pulse mode

Monopulse mode (OPM) is a special case of the above mode. In this mode, the counter can be activated at the trigger of an excitation signal and can produce a pulse with programmable pulse width after a programmable delay.

Can be started by CEN bit control counter; The waveform can be generated in output comparison mode or PWM mode by selecting monopulse mode at OPM position 1 in the TIMx_CR1 register. In this way, the counter will automatically stop when the next update event UEV occurs.

Only when the comparison value is different from the initial value of the counter, can a pulse be correctly generated, and before starting (when the timer is waiting to trigger), the following configuration must be performed:

- When counting incremented: CNT<CCRx<=ARR(note, 0<CCRx)

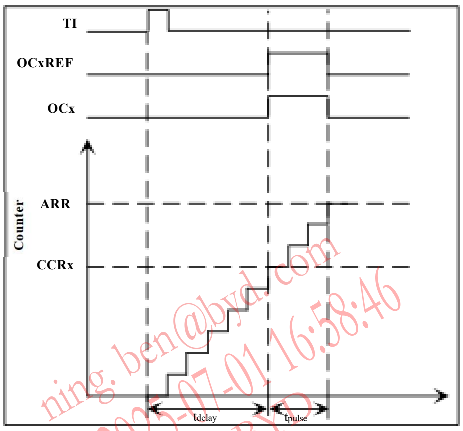


Figure 12.12 Single pulse mode

Special cases: OCx quick enable:

In monopulse mode, the CEN position is 1, indicating that the counter is enabled, and then the output is flipped when a comparison occurs between the counter value and the comparison value. However, multiple clock cycles are required to complete these operations, which limits the minimum possible delay (t_{delay} minimum).

If you need to output the waveform with minimal delay, you can place the OCxFE position 1 in the TIMx_CCMRx register, which forces the OCxRER (and OCx) to respond to the excitation signal regardless of the result of the comparison, with the same new level as when the comparison match occurred. OCxFE works only when the channel is configured in PWM1 or PWM2 mode.

12.3.12. UIF bit remapping

The UIFREMAP bit in the TIMx_CR1 register forces the update interrupt flag UIF to be continuously copied to bit 31(TIMx_CNT[31]) in the timer counter register.This automatically reads the counter value and the potential reversal condition emitted by the UIFCPY flag. In certain cases, this simplifies calculations by avoiding race conditions when processing is shared between background tasks (counter reads) and interrupts (update interrupts). There is no delay between enabling the UIF and UIFCPY flags.

12.3.13. DMA continuous transfer mode

The TIMx timer can generate multiple DMA requests based on a single event. The main purpose is to be able to reprogram a portion of the timer multiple times without software overhead, but can also be used to periodically read multiple registers in a row.

The DMA controller target is unique and must point to the virtual register TIMx_DMAR. When a given timer event occurs, the timer initiates a DMA request sequence (burst). Each write to the TIMx_DMAR register redirects to one of the timer registers.

The DBL[4:0] bit in the TIMx_DCR register sets the DMA continuous transfer length. When the TIMx_DMAR address is read or written, the timer carries out a continuous transmission, that is, the number of transfers (by half a word or byte).

The DBA[4:0] bit in the TIMx_DCR register defines the DMA base address of the DMA transfer (when read/write access is performed via the TIM6_DMAR address). DBA is defined as the offset calculated from the TIMx_CR1 register address:

Example:

- 00000:TIMx_CR1
- 00001:TIMx_CR2
- 00010:TIMx_SMCR

For example, the timer DMA continuous transfer function is used to update the contents of the CCRx register (x=1) to half words that are transferred to the CCRx register via DMA after an update event occurs.

The specific steps are as follows:

1. The specific steps are as follows:
   a) The destination data end address of the DMA channel is the DMAR register address
   b) The DMA channel source data end address is the RAM buffer address containing the data to be transferred to the CCRx register via DMA
   c) Amount of data to be transferred = 1
   d) R_power = 0x00
   e) Select the request source TIMx_UP
2. Configure the DCR register by configuring the DBA and DBL bit fields as follows: DBL= 1, DBA= 0xD.
3. Enables TIMx update DMA requests (UDE position 1 in the DIER register).
4. Enable TIMx
5. Enable the DMA channel

This example is applicable when the CCRx register is updated only once. If each CCRx register is updated twice, the amount of data to be transferred should be 2. The following uses a RAM buffer containing data1 and data2 as an example. Data is transferred to the CCRx register as follows: During the first update DMA request, data1 is transferred to CCR1; During the second update DMA request, data2 is transferred to CCR1.

12.4. TIMx register

Base address: 0x5000 0000

Address offset	Register	Description
0x04	RCU_EN	Peripheral module clock control register
0x10	ANA_CFG	Analog module switch register
0x44	TIM_CLK_SEL	TIM clock select register
0x48	CLK_SEL_RDY	Clock switch completion flag register

Base address: TIM1: 0x5006 0100 TIM2: 0x5006 0200
TIM3: 0x5006 0300 TIM4: 0x5006 0400
TIM5: 0x5006 0500 TIM7: 0x5006 0700

Address offset	Register	Description
0x00	TIMx_CR1	TIMx control register 1
0x04	TIMx_CR2	TIMx control register 2
0x0C	TIMx_DIER	TIMx DMA/ interrupt enable register
0x10	TIMx_SR	TIMx status register
0x14	TIMx_EGR	TIMx event generation register
0x18	TIMx_CCMR1	TIMx Capture/compare mode register 1
0x20	TIMx_CCER	TIMx Capture/compare enable register
0x24	TIMx_CNT	TIMx counter
0x28	TIMx_PSC	TIMx pre-divider
0x2C	TIMx_ARR	TIMx automatic reload register
0x34	TIMx_CCR1	TIMx capture/compare register 1
0x48	TIMx_DCR	TIMx DMA control register
0x4C	TIMx_DMAR	TIMx full transport DMA address
0x68	TIMx_DIR	TIMx input/output control register
0x6C	TIMx_EN	TIMx module enable register

12.4.1. Peripheral module clock control register(RCU_EN)

Address offset: 0x04

Reset value: 0x0000 0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.				DIV_ CLKEN	GPIOD_ CLKEN	GPIOC_ CLKEN	GPIOB_ CLKEN	GPIOA_ CLKEN	DMA_ CLKEN	CRC_ CLKEN	ADC_ CLKEN	WDT_ CLKEN	TIM7_ CLKEN	TIM6_ CLKEN	TIM5_ CLKEN
				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
TIM4_ CLKEN	TIM3_ CLKEN	TIM2_ CLKEN	TIM1_ CLKEN	TIM0_ CLKEN	PWM1_ CLKEN	PWM0_ CLKEN	IIC_ CLKEN	UART4_ CLKEN	UART3_ CLKEN	UART2_ CLKEN	UART1_ CLKEN	UART0_ CLKEN	SPI1_ CLKEN	SPI0_ CLKEN	Res.
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	

26	GPIOD_CLKEN	GPIOD module work enable:

		1:on 0:Close. Default is 0
25	GPIOC_CLKEN	GPIOC module work enable: 1:on 0:Close. Default is 0
24	GPIOB_CLKEN	GPIOB module work enable: 1:on 0:Close. Default is 0
23	GPIOA_CLKEN	GPIOA module work enable: 1:on 0:Close. Default is 0
22	DMA_CLKEN	DMA(including DMA_SRAM/DMAMUX) module work enable: 1:on 0:Close. Default is 0
18	TIM7_CLKEN	TIM7 module work enable: 1:on 0:Close. Default is 0
16	TIM5_CLKEN	TIM5 module work enable: 1:on 0:Close. Default is 0
15	TIM4_CLKEN	TIM4 module work enable: 1:on 0:Close. Default is 0
14	TIM3_CLKEN	TIM3 module work enable: 1:on 0:Close. Default is 0
13	TIM2_CLKEN	TIM2 module work enable: 1:on 0:Close. Default is 0
12	TIM1_CLKEN	TIM1 module work enable: 1:on 0:Close. Default is 0

12.4.2. Analog module switch register(ANA_CFG)

Address offset: 0x10

Reset value: 0x0000 1850

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

15:13	12	11	10	9	8:7	6	5	4	3	2	1:0
Res.	PD_TEMP	PD_ADC	VREF_SEL	VREF_VOL_SEL	VREF_IN_ADC_SEL	PD_ADC_IN_VREF	XTAL_HFR_SEL	XTAL_SEL	XTAL_BYP_SEL		XTAL_EN_

										SEL
	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

5	XTAL_HFR_SEL	Analog high frequency crystal oscillator circuit current configuration selection: 0:175μA(4MHz crystal oscillator recommended) 1:300μA(8MHz crystal oscillator recommended)
4	XTAL_SEL	When a passive crystal vibration circuit is available: 0:Select 32K 1:Select 4M/8M. Default value 1
3:2	XTAL_BYP_SEL	Active crystal oscillator path selection: 00:Channel closed, active crystal oscillator is not used 01:Select channel A (PA5) input 10:Select channel AB(PA2) input 11:Reserved Note: Select any active crystal path, and the passive crystal vibration circuit is turned off at the same time. The default value is 00.
1:0	XTAL_EN_SEL	Passive crystal path selection: 00:The passive crystal oscillator is turned off and passive crystal oscillator is not used 01:Select path A (PA5, PA4) 10:Select path B (PA2, PA1) input 11:Reserved When XTAL_BYP_SEL[1:0] = 00 and XTAL_EN_SEL[1:0]! = 00, turn on the passive crystal vibration circuit, the default value is 00. Note: Passive crystal oscillator must be enabled when active crystal oscillator is not enabled.

12.4.3. TIM clock select register(TIM_CLK_SEL)

Address offset: 0x44

Reset value: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIM7CLK		TIM6CLK		TIM5CLK		TIM4CLK		TIM3CLK		TIM2CLK		TIM1CLK		TIM0CLK	
RW		RW		RW		RW		RW		RW		RW		RW	

31:16	-	Reserved
15:14	TIM7CLK	TIM7 Clock selection: 00:Select the PLL48M clock 01:Select the LIRC 32K clock

		1x:Choose XTAL (passive oscillator (32768Hz/4MHz/8MHz) and active oscillator (1MHz~48MHz))
11:10	TIM5CLK	TIM5 Clock selection: 00:Select the PLL48M clock 01:Select the LIRC 32K clock 1x:Choose XTAL (passive oscillator (32768Hz/4MHz/8MHz) and active oscillator (1MHz~48MHz))
9:8	TIM4CLK	TIM4 Clock selection: 00:Select the PLL48M clock 01:Select the LIRC 32K clock 1x:Choose XTAL (passive oscillator (32768Hz/4MHz/8MHz) and active oscillator (1MHz~48MHz))
7:6	TIM3CLK	TIM3 Clock selection: 00:Select the PLL48M clock 01:Select the LIRC 32K clock 1x:Choose XTAL (passive oscillator (32768Hz/4MHz/8MHz) and active oscillator (1MHz~48MHz))
5:4	TIM2CLK	TIM2 Clock selection: 00:Select the PLL48M clock 01:Select the LIRC 32K clock 1x:Choose XTAL (passive oscillator (32768Hz/4MHz/8MHz) and active oscillator (1MHz~48MHz))
3:2	TIM1CLK	TIM1 Clock selection: 00:Select the PLL48M clock 01:Select the LIRC 32K clock 1x:Choose XTAL (passive oscillator (32768Hz/4MHz/8MHz) and active oscillator (1MHz~48MHz))

12.4.4. Clock switch completion flag register(CLK_SEL_RDY)

Address offset: 0x48

Reset value: 0x0000 03FF

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

15:10	9	8	7	6	5	4	3	2	1	0
Res.	TIM7RDY	TIM6RDY	TIM5RDY	TIM4RDY	TIM3RDY	TIM2RDY	TIM1RDY	TIM0RDY	PWM1RDY	PWM0RDY
	R	R	R	R	R	R	R	R	R	R

31:10	-	Reserved
9	TIM7RDY	TIM7 Clock switchover complete flag: 0:Be switching 1:Clock switching is complete

7	TIM5RDY	TIM5 Clock switchover complete flag: 0:Be switching 1:Clock switching is complete
6	TIM4RDY	TIM4 Clock switchover complete flag: 0:Be switching 1:Clock switching is complete
5	TIM3RDY	TIM3 Clock switchover complete flag: 0:Be switching 1:Clock switching is complete
4	TIM2RDY	TIM2 Clock switchover complete flag: 0:Be switching 1:Clock switching is complete
3	TIM1RDY	TIM1 Clock switchover complete flag:: 0:Be switching 1:Clock switching is complete

12.4.5. TIMx control register 1 (TIMx_CR1) (x = 1/2/3/4/5/7)

Address offset: 0x00

Reset value: 0x0000 0000

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

15:12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	UIFREMAP	Res.	CKD[1:0]		ARPE	Res.			OPM	URS	UDIS	CEN
	RW		RW		RW				RW	RW	RW	RW

31:12	-	Reserved
11	UIFREMAP	UIF status bit remapping: 0:No remapping, UIF status bits are not copied to bit 31 of the TIMx_CNT register 1:Remapping is enabled. The UIF status bit is copied to bit 31 of the TIMx_CNT register
10	-	Reserved
9:8	CKD[1:0]	Clock frequency division factor: The frequency division ratio between the timer clock (TIMx_CLK) frequency and the dead band and sampling clock (t_{DTS}) used by the digital filter (TIx) 00: $t_{DTS}=t_{TIMx\_CLK}$ 01: $t_{DTS}=2*t_{TIMx\_CLK}$ 10: $t_{DTS}=4*t_{TIMx\_CLK}$ 11: Reserved Note: This bit is set before the TIMx_EN register is configured
7	ARPE	Automatic reload preload allowance: 0:TIMx_ARR registers are not buffered 1:TIMx_ARR register buffering
6:4	-	Reserved
3	OPM	Single pulse mode: 0:The counter does not stop counting when an update event occurs 1:The counter stops counting at the next update event (clears the CEN bit)

2	URS	Update request source: This bit is set to 1 and cleared by the software to select the UEV event source 0:When enabled, all of the following events generate update interrupts or DMA requests. Such incidents include: – Counter overflow – UG is set to 1 1:When enabled, only counter overflows generate update interrupts or DMA requests
1	UDIS	Disable update: This bit is set to 1 and cleared by the software to enable/disable UEV event generation. 0:The UEV function was enabled. Update UEV events can be generated by one of the following events: – Counter overflow – UG is set to 1 Then update the value of the shadow register 1:Prohibit UEV. No update event is generated, and the values of each shadow register (ARR, PSC, and CCRx) remain unchanged. However, if the UG position is 1, the counter and pre-divider are reinitialized
0	CEN	Counter enable: 0:Disable counter 1:Enable counter

12.4.6. TIMx control register2 (TIMx_CR2) (x = 1/2/3/4/5/7)

Address offset: 0x04

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.									MMS2[2:0]			Res.			
									RW						

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

31:23	-	Reserved
22:20	MMS2[2:0]	Master mode selection: These bits select the information that will be sent to the ADC for synchronization (TRGO2). The combination of these bits is as follows: 000:Reset, the UG bit in the TIMx_EGR register is used as the trigger output (TRGO2). 001:Enable, the counter enable signal CNT_EN is used as a trigger output (TRGO2). 010:Update, select the update event as the trigger output (TRGO2). 011:When the CC1IF flag is set to 1 (even if it is already high), the trigger output (TRGO2) sends a positive pulse whenever a capture or comparison match occurs. 100:For comparison, the OC1REF signal is used as a trigger output (TRGO2)

		Others: Reserved Note: The ADC clock must be enabled before events can be received from the master timer. The ADC clock must not be changed in real time when the trigger signal is received from the master timer. Note: This bit is set before the TIMx_EN register is configured
19:0	-	Reserved

12.4.7. TIMx DMA/ interrupt enable register (TIMx_DIER) (x = 1/2/3/4/5/7)

Address offset: 0x0C

Reset value: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.						CC1DE	UDE	Res.						CC1IE	UIE
						RW	RW							RW	RW

31:10	-	Reserved
9	CC1DE	Capture/compare 1 DMA request enable: 0:Disable CC1 DMA requests 1:Enable CC1 DMA requests
8	UDE	Update DMA request enable: 0:Disable updating DMA requests 1:Enable updating DMA requests
7:2	-	Reserved
1	CC1IE	Capture/compare 1 Interrupt Enable: 0:Disable CC1 interrupt 1:Enable CC1 interrupt
0	UIE	Update interrupt enable: 0:Disable update interrupt 1:Enable update interrupt

12.4.8. TIMx status register (TIMx_SR) (x = 1/2/3/4/5/7)

Address offset: 0x10

Reset value: 0x0000 0000

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

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

Res.	CC1OF	Res.	CC1IF	UIF
	RC_W0		RW	RW

31:19	-	Reserved
18	RDCNTF	TIMx_CNT register read flag bits: 1:Software write 1, counter count value latch enabled, latch counter count value 0:After the counter counts the value of the latch is completed, the hardware will clear the bit , and the bit also can be written or cleared by software.
17:10	-	Reserved
9	CC1OF	Capture/compare 1 Repeat capture flag: This flag bit is set to 1 by the hardware only if the corresponding channel is configured in input capture mode. This bit can be cleared by writing "0" to the software. 0:No duplicate capture was detected 1:The counter value is captured in the TIMx_CCR1 register and the CC1IF flag is set to 1
8:2	-	Reserved
1	CC1IF	Capture/compare 1 Interrupt flag: **If the channel CC1 is configured as an output:** When the counter matches the comparison value, this flag is set to 1 by hardware, but needs to be cleared by software. 0:mismatch 1:The value of the TIMx_CNT counter matches the value of the TIMx_CCR1 register. When the value of TIMx_CCR1 is greater than the value of TIMx_ARR, the CC1IF bit becomes high when the counter overflows (in increasing count mode). **If channel CC1 is configured as input:** this bit will be set to 1 by the hardware when a capture event occurs. Clear this bit by software or reading the TIMx_CCR1 register. 0:No input capture event occurred 1:Counter values captured in the TIMx_CCR1 register (edges matching the selected polarity detected on IC1)
0	UIF	Update interrupt flag: This bit is set to 1 by hardware when an update event occurs, but needs to be cleared by software. 0:No update occurred. 1:Update interrupt suspended. This bit is set to 1 by the hardware when updating the register in the following cases: – UDIS=0 in the TIMx_CR1 register, and the counter overflows. – URS = 0 and UDIS = 0 in the TIMx_CR1 register, and CNT is re-initialized by the software using the UG bit in the TIMx_EGR register.

12.4.9. TIMx event generation register (TIMx_EGR) (x = 1/2/3/4/5/7)

Address offset: 0x14

Reset value: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.														CC1G	UG
														W	W

31:2	-	Reserved
1	CC1G	Capture/compare 1 Generation: This bit is set to 1 by the software to generate events and automatically cleared by the hardware 0:No operation is performed 1:Generate capture/compare events on channel 1: **If channel CC1 is configured as output:** When enabled, the CC1IF flag is set to 1 and the corresponding interrupt or DMA request is sent. **If channel CC1 is configured as input:** The current counter value is captured in the TIMx_CCR1 register. When enabled, the CC1IF flag is set to 1 and the corresponding interrupt or DMA request is sent. If the CC1IF flag is already high, the CC1OF flag is set to 1
0	UG	Update generation: This bit can be set to 1 by software and automatically cleared by hardware 0:No operation is performed 1:Reinitializes the counter and generates a register update event. Please note that the pre-divider counter will also be cleared to zero (but the pre-divider ratio will not be affected)

12.4.10. TIMx capture/compare mode register 1[reuse] (TIMx_CCMR1) (x = 1/2/3/4/5/7)

Address offset: 0x18

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
Res.															OC1M[3]
															RW

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.								IC1F[3:0]				IC1PSC[1:0]		CC1S[1:0]	
								Res.	OC1M[2:0]			OC1PE	OC1FE	CC1S[1:0]	
								RW				RW		RW	

Enter the capture mode:

31:8	-	Reserved
7:4	IC1F[3:0]	Input capture 1 filter: This bit field defines the sampling frequency and digital filter bandwidth of the TI1 signal. The digital filter consists of an event counter, where each N consecutive events is considered a valid output edge: 0000: No filter, sampling at f_{DTS} frequency 0001: $f_{SAMPLING} = f_{TIMx\_CLK}$, N=2 0010: $f_{SAMPLING} = f_{TIMx\_CLK}$, N=4 0011: $f_{SAMPLING} = f_{TIMx\_CLK}$, N=8 0100: $f_{SAMPLING} = f_{DTS}/2$, N=6 0101: $f_{SAMPLING} = f_{DTS}/2$, N=8 0110: $f_{SAMPLING} = f_{DTS}/4$, N=6 0111: $f_{SAMPLING} = f_{DTS}/4$, N=8 1000: $f_{SAMPLING} = f_{DTS}/8$, N=6 1001: $f_{SAMPLING} = f_{DTS}/8$, N=8 1010: $f_{SAMPLING} = f_{DTS}/16$, N=5 1011: $f_{SAMPLING} = f_{DTS}/16$, N=6 1100: $f_{SAMPLING} = f_{DTS}/16$, N=8 1101: $f_{SAMPLING} = f_{DTS}/32$, N=5 1110: $f_{SAMPLING} = f_{DTS}/32$, N=6 1111: $f_{SAMPLING} = f_{DTS}/32$, N=8 Note: The configuration is completed before the configuration of the TIMx_EN register
3:2	IC1PSC[1:0]	Input capture 1 predivider: These bits define the pre-division ratio of the CC1 input (IC1). As long as CC1E= "0" (TIMx_CCER register), the pre-divider is reset immediately. 00:Without a pre-divider, capture is performed on every edge detected on the capture input 01: A capture is performed for every 2 events 10:A capture is performed for every 4 events 11:A capture is performed for every 8 events Note: This bit is set before the TIMx_EN register is configured
1:0	CC1S[1:0]	Capture/Compare 1 Select: Define the channel direction (input/output) and the input pins to be used 00:The CC1 channel is configured as output 01:The CC1 channel is configured as input, and IC1 is mapped to TI1 Other values: Reserved Note: 1. Data can be written to the CC1S bit only when the channel is closed (CC1E = "0" in TIMx_CCER) 2. The configuration is completed before the configuration of the TIMx_EN register

Output comparison mode:

31:17	-	Reserved
16	OC1M[3]	Output comparison 1 mode: This bit is the third bit of OC1M[3:0]. Note refer to OC1M[3:0]
15:7	-	Reserved
6:4	OC1M[3:0]	Output comparison 1 mode: The third bit of OC1M is in the 16 bit of TIMx_CCMR1(output comparison) These bits define the behavior of OC1REF, the output reference signal for OC1 and OC1N. OC1REF is active at a high level, while the active levels of OC1 and OC1N depend on the CC1P bit and CC1NP bit. 0000:Frozen, the output comparison register TIMx_CCR1 is compared with the counter TIMx_CNT without any effect on the output (this mode is used to generate the timebase). 0001:Set channel 1 to the output active level when matched. When the counter TIMx_CNT matches the capture/comparison register 1(TIMx_CCR1), the OC1REF signal is forced to a high level. 0010:Set channel 1 to output invalid level when matched. When the counter TIMx_CNT matches the capture/comparison register 1 (TIMx_CCR1), the OC1REF signal is forced to a low level. 0011:When TIMx_CNT=TIMx_CCR1, the OC1REF is flipped. 0100:The OC1REF is forcibly changed to an invalid level and the OC1REF is forcibly changed to a low level. 0101:The OC1REF is forcibly changed to an active level and the OC1REF is forcibly changed to a high level. 0110:In PWM mode 1, in incremental count mode, as long as TIMx_CNT<TIMx_CCR1, channel 1 is in the valid state, otherwise it is invalid. 0111:In PWM mode 2, in increasing count mode, as long as TIMx_CNT<TIMx_CCR1, channel 1 is invalid state, otherwise it is valid state. Other values: Reserved Note: In PWM mode, the OCREF level changes only when the comparison result changes or the output comparison mode switches from "Freeze" mode to "PWM" mode
3	OC1PE	Output comparison 1 Preloaded Enable: 0:The pre-loaded register related to TIMx_CCR1 is disabled. Data can be written to TIMx_CCR1 at any time, and the new value will be used immediately after writing. 1:Enable the pre-loaded register related to TIMx_CCR1, and enable read/write access to the pre-loaded register. The TIMx_CCR1 preloaded value is loaded into the current register each time an update event is generated. Note: PWM mode (OPM position 1 in the TIMx_CR1 register) can only be used in monopulse mode without verifying the pre-loaded register, otherwise this behavior is not guaranteed.
2	OC1FE	Output comparison 1 quick enable: This bit is used to speed up the impact of triggered input events on CC output.

		0:Even if triggered to turn on, CC1 will work properly according to the counter and CCR1 value. The minimum delay time to activate the CC1 output when the trigger input appears edging is 5 clock cycles. 1:Triggering an effective edge on the input corresponds to a comparative match on the CC1 output. Subsequently, OC is set to the comparison level regardless of the comparison result. The OCFE works only when the channel is configured in PWM1 or PWM2 mode
1:0	CC1S[1:0]	Capture/Compare 1 Select: This bit field defines the channel direction (input/output) and the input pins used 00:The CC1 channel is configured as output 01:The CC1 channel is configured as input, and IC1 is mapped to TI1 Other values: Reserved Note: Data can be written to the CC1S bit only when the channel is closed (CC1E = "0" in TIMx_CCER)

12.4.11. TIMx capture/compare enable registers(TIMx_CCER) (x = 1/2/3/4/5/7)

Address offset: 0x20

Reset value: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.												CC1NP	Res.	CC1P	CC1E
												RW		RW	RW

31:4	-	Reserved
3	CC1NP	Capture/compare 1 Complementary output polarity: **The CC1 channel is configured as output:** 0:OC1N High level is valid 1:OC1N Low level is valid **The CC1 channel is configured as input:** This bit is used in conjunction with CC1P to define the polarity of TI1FP1. Refer to CC1P description
2	-	Reserved
1	CC1P	Capture/compare 1 output polarity: **The CC1 channel is configured as output:** 0:OC1 High level is valid 1:OC1 Low level is valid **The CC1 channel is configured as an input:** the CC1NP/CC1P bit selects the effective polarity of TI1FP1 for trigger or capture operations.

		00:Non-inverting/rising edge trigger, the circuit acts on the rising edge of TI1FP1, TI1FP1 is not inverting 01:Reversed phase/falling edge trigger, the circuit acts on the falling edge of TI1FP1, TI1FP1 reversed phase 10:Reserved 11:Non-inverting/both rising and falling edges are triggered
0	CC1E	Capture/compare 1 Output Enable: **The CC1 channel is configured as output:** 0:Off -- OC1 is not activated 1:On - The OC1 signal is output to the corresponding output pin **The CC1 channel is configured as an input:** the counter value is captured into the capture/compare register 1 (TIMx_CCR1) 0:No capture 1:Enable capture

12.4.12. TIMx counter (TIMx_CNT) (x = 1/2/3/4/5/7)

Address offset: 0x24

Reset value: 0x0000 0000

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

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

31	UIFCPY	UIF copy: This bit is a read-only copy of the UIF bit in the TIMx_ISR register If the UIFREMAP bit in TIMx_CR1 is reset, bit 31Reserved is read as 0
30:16	-	Reserved
15:0	CNT[15:0]	Count the value. Write to the register is prohibited during the counter count. When the counter has not started counting, the initial counter value can be modified by writing the TIMx_CNT register. Reading process: 1.TIMx_SR register bit[18] is written to 1, then the counter will be latched to count the value; 2.Read the TIMx_SR register and check bit[18]. When bit[18] is 0, it indicates that the counter count value latch is complete. 3.The TIMx_CNT register is read, and the value read is the value desired by the current read flow.

12.4.13. TIMx prescaler (TIMx_PSC) (x = 1/2/3/4/5/7)

Address offset: 0x28

Reset value: 0x0000 0000

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

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

31:16	-	Reserved
15:0	PSC[15:0]	Pre-divider value: Counter clock frequency (CK_CNT) = f_{CK_PSC}/(PSC[15:0] +1) The PSC contains the value to be loaded into the active pre-divider register each time an update event occurs, including when the counter is cleared by UG bit in the TIMx_EGR register

12.4.14. TIMx automatic reload register (TIMx_ARR) (x = 1/2/3/4/5/7)

Address offset: 0x2C

Reset value: 0x0000 FFFF

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

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

31:16	-	Reserved
15:0	ARR[15:0]	Automatic reload value: ARR is the value to be loaded into the actual automatic reload register The counter does not work when the automatic overload value is empty

12.4.15. TIMx capture/compare register 1 (TIMx_CCR1) (x = 1/2/3/4/5/7)

Address offset: 0x34

Reset value: 0x0000 0000

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

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

CCR1[15:0]
RW

31:16	-	Reserved
15:0	CCR1[15:0]	Capture/compare 1 values: **If channel CC1 is configured to output:** CCR1 is the value to be loaded into the valid capture/compare 1 register (the preloaded value). If the OC1PE bit in the TIMx_CCMR1 register is not used to preload the function, the value takes effect immediately. Otherwise it only takes effect when an update event occurs (copied to the actual active capture/compare register 1) The actual capture/compare register contains the value to be compared to the counter TIMx_CNT and signaled on the OC1 output. **If channel CC1 is configured as an input:** CR1 captures the counter value for the previous input when the 1 event (IC1) occurred. The TIMx_CCR1 register can only be read and cannot be programmed.

12.4.16. TIMx DMA control register (TIMx_DCR) (x = 1/2/3/4/5/7)

Address offset: 0x48

Reset value: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.			DBL[4:0]					Res.			DBA[4:0]				
			RW								RW				

31:13	-	Reserved
12:8	DBL[4:0]	DMA continuous transfer length: The 5 bits define the DMA transfer length (when the TIMx_DMAR address is read or written, the timer makes a continuous transfer), that is, the number of transfers. Can be transmitted by half word or byte 00000:once transmission 00001:Secondary transmission 00010:Tertiary transmission … 10001:18 transmissions example: DBL = 7 bytes and DBA = TIMx_CR1 – If DBL = 7 bytes and DBA = TIMx_CR1 represents the address of the bytes to be transferred, the address to be transferred should be given by the following formula: (TIMx_CR1 address) + DBA + (DMA index), where DMA index = DBL In this case, add 7 bytes to (TIMx_CR1 address) + DBA to get the source/destination

		address of the data to be copied. In this case, data is passed to seven registers starting at the following address: (TIMx_CR1 address) + DBA Depending on the configuration of the DMA data size, several things can happen: – If the DMA data size is configured as a half-word, 16 bits of data are transferred to each of the seven registers – If you configure the DMA data size in bytes, you will also send data to seven registers: the first contains the first MSB byte, the second contains the first LSB byte, and so on. Therefore, when using the transfer timer, you must also specify the size of the data to be transferred by DMA
7:5	-	Reserved
4:0	DBA[4:0]	DMA base Address: The 5-bit vector defines the base address of the DMA transfer (for read/write access via the TIMx_DMAR address) as the offset calculated from the TIMx_CR1 register address: 00000:TIMx_CR1 00001:TIMx_CR2 00010:Reserved 00011:TIMx_DIER

12.4.17. TIMx full transport DMA address (TIMx_DMAR) (x = 1/2/3/4/5/7)

Address offset: 0x4C

Reset value: 0x0000 0000

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

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

31:0	DMAB[31:0]	DMA continuous transfer register: Reading or writing to the DMAR register will access the register at the following address: (TIMx_CR1 address) + (DBA + DMA index) x 4, where: TIMx_CR1 address is the address of control register 1; DBA is the DMA base address configured in the TIMx_DCR register. The DMA index is automatically controlled by the DMA transfer and ranges from 0 to DBL (DBL configured in the TIMx_DCR register)

12.4.18. TIMx input/output control register (TIMx_DIR) (x = 1/2/3/4/5/7)

Address offset: 0x68

Reset value: 0x0000 0000

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

15:1	0
Res.	CH1DIR
	RW

31:1	-	Reserved
0	CH1DIR	TIMx_CH1 direction control: 0:output 1:input Note: This register is configured before the TIMx_EN register is configured

12.4.19. TIMx module enable register (TIMx_EN) (x = 1/2/3/4/5/7)

Address offset: 0x6C

Reset value: 0x0000 0000

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

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

31:1	-	Reserved
0	TIMxEN	TIMx module work enables: 1:Module work 0:Module not working

Chapter 13 Universal timer(TIM6)

13.1. TIM6 summarize

The universal timer (TIM6) contains a 16-bit automatic overload counter that is driven by a programmable pre-divider.

13.2. TIM6 functional characteristics

TIM6 timer main features:

- 16-bit increment, decrement, increment/decrement automatically reload counters
- 16-bit programmable (can be modified in real time) pre-divider, counter clock frequency divider factor of any value between 1 and 65536
- Up to 4 separate channels can be used for:
  - Input capture
  - Output Comparison
  - PWM generation (edge and center alignment mode)
  - Monopulse mode output
- A synchronous circuit that uses external signals to control timers and enables multiple timers to interconnect
- An interrupt /DMA request is generated when the following events occur:
  - Update: Counter overflow/underflow, counter initialization (triggered by software or internal/external)
  - Trigger events (counters start, stop, initialize, or count by internal/external trigger)
  - Input capture
  - Output Comparison
- Incremental (orthogonal) encoders and Hall sensor circuits for positioning are supported
- Trigger input used as external clock

13.3. TIM6 function description

13.3.1. TIM6 block diagram

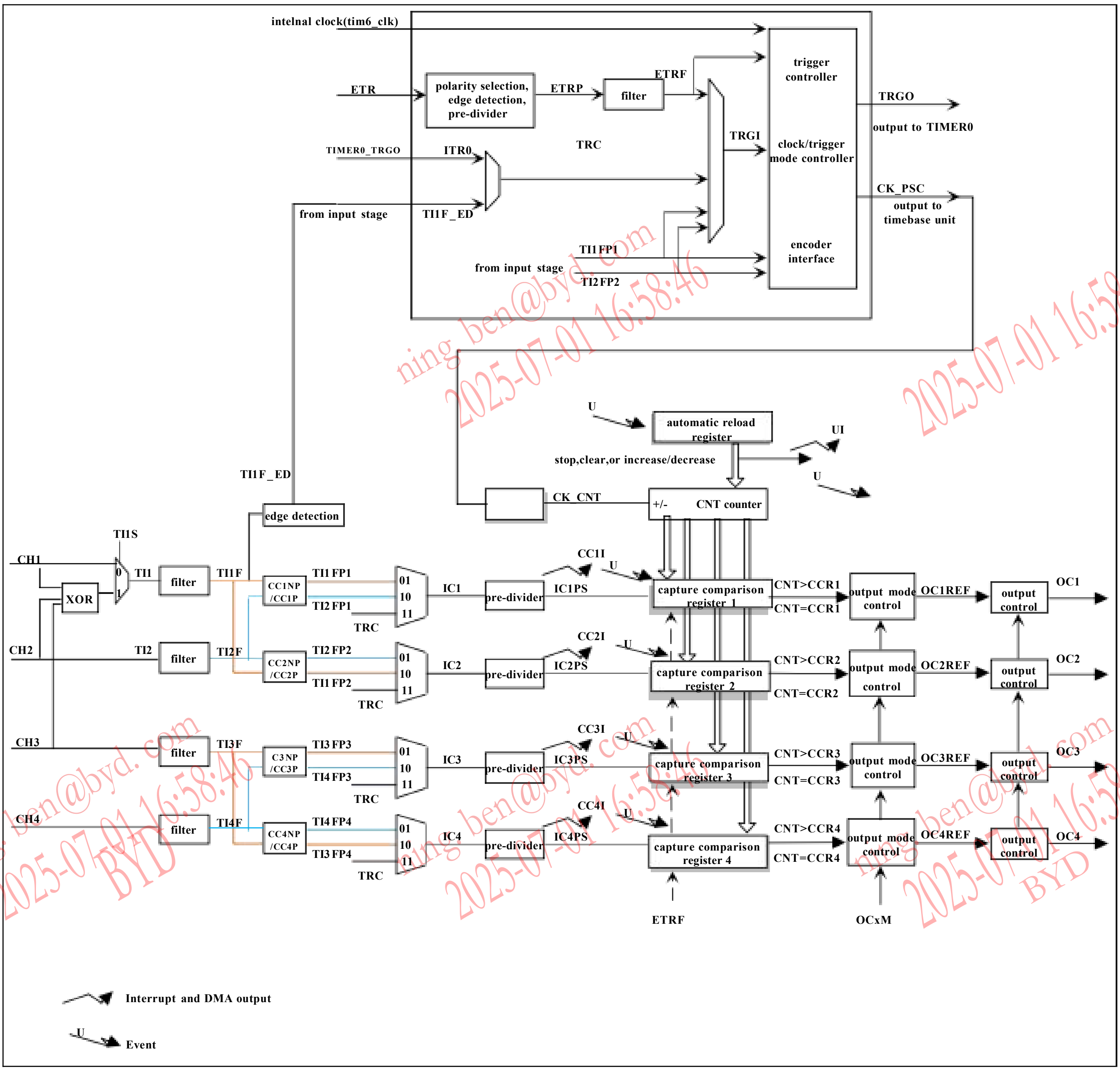


Figure 13.1 Advanced control timer block diagram

13.3.2. Clock selection

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

- TIM6_CLK: PLL_48MHz/LIRC 32K/XTAL(passive crystal oscillator (32768Hz/4MHz/8MHz) and active crystal oscillator (1MHz~48MHz))
- External clock Mode 1: External input pin or internal trigger input (ITRx)
- External clock Mode 2: Externally triggered input ETR
- encoder mode

The slave mode controller allows the user to select the clock source of the counter, the input trigger signal and the output signal.

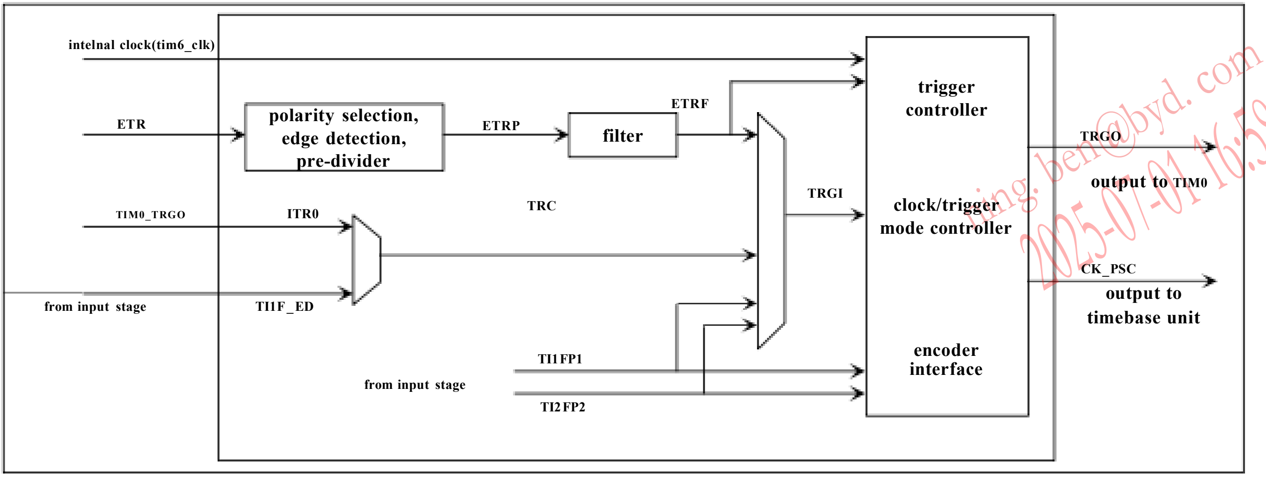


Figure 13.2 Slave mode controller block diagram

13.3.2.1. Internal clock sourceTIM6_CLK)

If the slave mode controller (SMS=000) is disabled, the counter enables the CEN bit, the counter counter direction DIR bit (TIM0_CR1 register), and the UG bit (TIM0_EGR register) to be the actual control bit and can only be changed by software (except UG, which still keeps auto-clear). When 1 is written to the CEN bit, the clock of the pre-divider is provided by the internal clock TIM6_CLK.

13.3.2.2. External clock source mode 1

This mode can be selected when SMS=0111 in the TIM6_SMCR register, and the counter can count when a rising edge appears on the selected input signal.

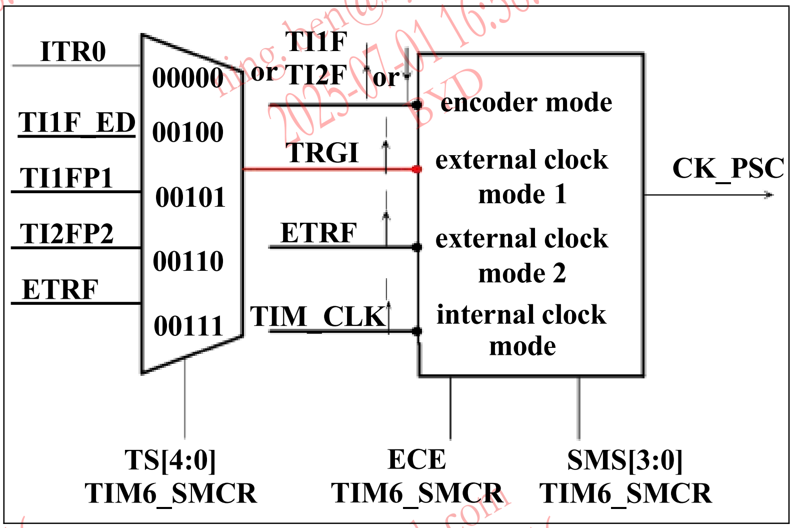


Figure 13.3 External clock mode 1

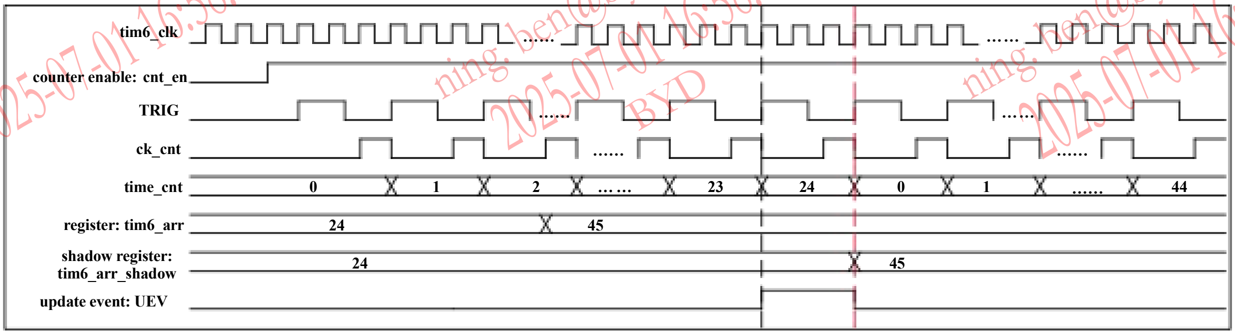


Figure 13.4 External clock source mode 1 Count timing diagram

For example, to make the incrementing counter count when the TI2 input has a rising edge, perform the following steps:

1. Configure the TIM6_CCMR1 register CC2S = 01, configure channel 2 as input, so that it can detect the rising edge of TI2 input;
2. Configure the TIM6_CCMR1 register IC2F[3:0], and configure the input filtering bandwidth as required;
3. Configure the TIM6_CCER registers CC2P = 0 and CC2NP = 0 to select the polarity of the rising edge;
4. Configure the TIM6_SMCR register SMS[3:0] = 0111 to make the timer work in external clock mode 1;
5. Configure the TIM6_SMCR register TS[4:0] = 00110 and select TI2FP2 after TI2 filtering as the trigger input source;
6. Set the TIM6_CR1 register CEN = 1 to enable the counter;

13.3.2.3. External clock mode 2

This mode can be selected by writing ECE= 1 in the TIM6_SMCR register. The counter can count when the external trigger input ETR appears with a rising or falling edge.

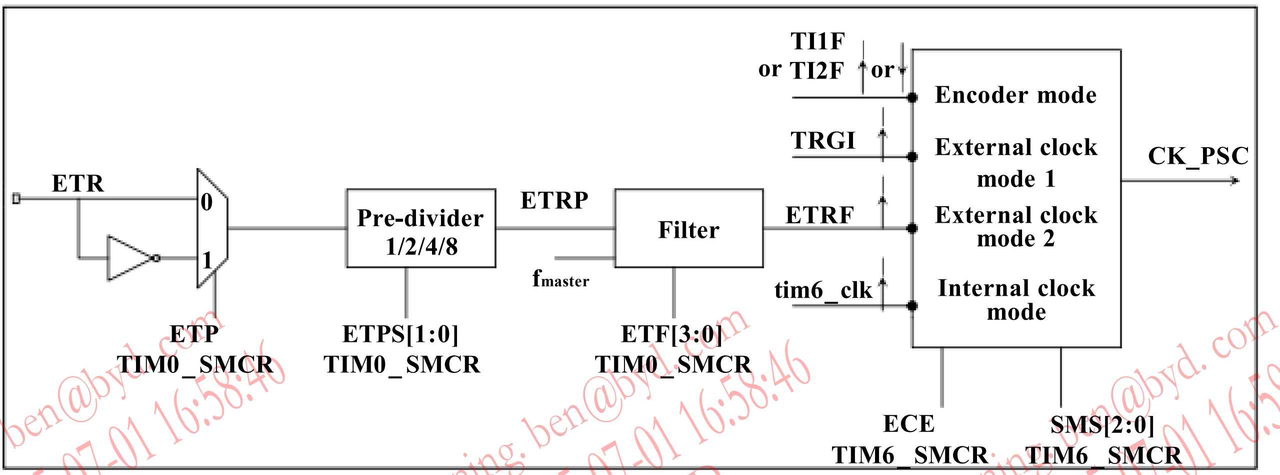


Figure 13.5 External trigger input module

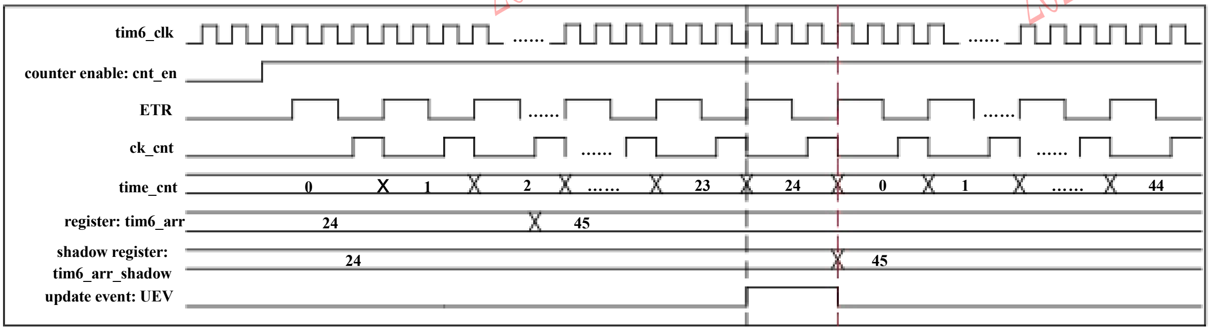


Figure 13.6 External clock source mode 2 Count timing diagram

For example, to make the incrementing counter count every 2 rising edges in the ETR, perform the following steps:

1. Configure TIM0_SMCR register ECE = 1, select external clock 2 mode;
2. Configure TIM0_SMCR register ETF[3:0], and configure external trigger input filtering as required;
3. Configure TIM0_SMCR register ETPS[0:1] = 01, set 2 frequency division;
4. Set TIM0_SMCR register ETP = 0, select the rising edge detection of ETR pin;
5. Set the TIM0_CR1 register CEN = 1 to enable the counter;

13.3.2.4. Encoder mode

Write SMS=0001 in TIM6_SMCR register, select encoder mode 1, count at TI1F edge; Write SMS=0010, select encoder mode 1, count at TI2F edge; Write SMS=0011, select encoder mode 3, and the counter counts at both TI1F and TI2F edges.

13.3.3. Time base unit

The main module of the universal control timer TIM6 is a 16-bit counter and its associated auto-reload register. The counter can be incremented, decremented, or alternately incremented and decremented. The clock of the counter can be divided by a pre-divider. Counters, auto-reload registers, and pre-divider registers can be read and written by software, even when the counter is running.

Time base units include:

- Pre-divider register (TIM6_PSC)
- Counter register (TIM6_CNT)
- Automatic reload register (TIM6_ARR)

13.3.3.1. Pre-divider PSC

The pre-divider PSC has an input clock CK_PSC and an output clock CK_CNT. The input clock comes from the controller part. By setting the value of pre-division, different CK_CNT can be obtained. The actual calculation formula is CK_CNT = F_{CK_PSC}/(PSC[15:0]+1).

Because the TIM6_PSC control register has a buffer (shadow register) that can change its value while it is running, the new predivision frequency value will take effect at the next update event.

13.3.3.2. Counter CNT

The counter can be incremented, decremented, or alternately incremented and decremented. The counter is started only when CEN in the TIM6_CR1 register is set to 1.

13.3.3.3. Automatic reload register ARR

The automatically loaded register is preloaded and is accessed when a write or read operation is performed on it. The contents ofthe TIM6_ARR register can be transferred either directly to the shadow register or each time an update event occurs, depending on the auto-reload preload enable bit (ARPE) in the TIM6_CR1 register. An update event is sent when the counter reaches an overflow value (or an underflow value when decrement counts) and the UDIS bit in the TIM6_CR1 register is 0. The update event can also be generated by software.

13.3.4. Counter mode

13.3.4.1. Incremental count mode

In incremental count mode, the counter counts from 0 to an automatically overloaded value (the contents of the TIM6_ARR register), then counts again from 0 and generates a counter overflow event.

An update event is generated each time a counter overflows, or when the UG position 1 of the TIM1_EGR register is changed (either by software or using a slave mode controller).

UEV events can be disabled by placing UDIS position 1 in the TIM6_CR1 register by software. This avoids updating the shadow register when a new value is written to the preloaded register and does not generate any update events until the UDIS bit is written to 0. However, both the counter and the pre-divider counter start counting again from 0 (while the pre-divider ratio remains the same). In addition, if the URS bit (update request selection) in the TIM6_CR1 register is set to 1, the UG position 1 generates the update event UEV, but does not set the UIF flag to 1 (therefore, no interrupt or DMA requests are sent). This is to avoid both update and capture interruptions when the counter is cleared in capture mode.

When an update event occurs, all registers are updated and the update flag (UIF bit in the TIM6_SR register) is set to 1 (depending on the URS bit) :

- The automatically overloaded shadow register is updated with the preloaded value (TIM6_ARR)
- The preload value will be reloaded in the buffer of the predivider (TIM6_PSC register)

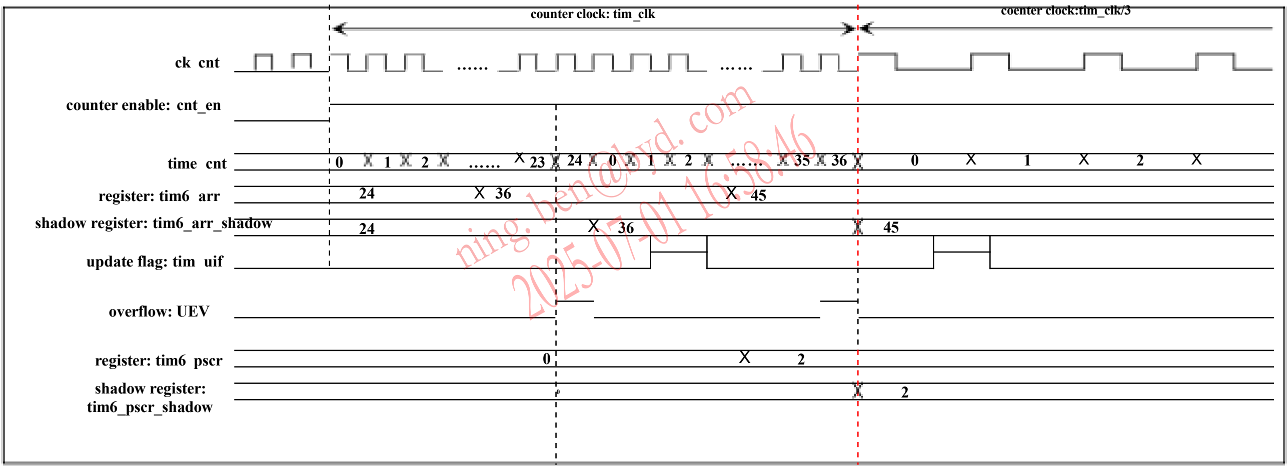


Figure 13.7 Incremental counting sequence diagram

13.3.4.2. Decrement count mode

In decrement count mode, the counter decrement the count from the automatic overload value (the contents of the TIM6_ARR register) to 0, then counts again from the automatic overload value and generates a counter underflow event.

An update event is generated each time a counter underflow occurs, or when the UG position 1 of the TIM6_EGR register is changed (either by software or using a slave mode controller).

The UEV update event can be disabled by placing UDIS position 1 in the TIM6_CR1 register by software. This avoids updating the shadow register when a new value is written to the preloaded register and does not generate any update events until the UDIS bit is written to 0. However, the counter starts counting again from the current automatic overload value, while the pre-divider counter starts counting again from 0 (but the pre-divider ratio remains the same).

In addition, if the URS bit (update request selection) in the TIM6_CR1 register is set to 1, the UG position 1 generates the update event UEV, but does not set the UIF flag to 1 (therefore, no interrupt or DMA requests are sent). This is to avoid both update and capture interruptions when a capture event occurs and the counter is cleared.

When an update event occurs, all registers are updated and the update flag (UIF bit in the TIM6_SR register) is set to 1 (depending on the URS bit) :

- The automatically overloaded shadow register is updated with the preloaded value (TIM6_ARR)
- The buffer of the pre-divider will be reloaded

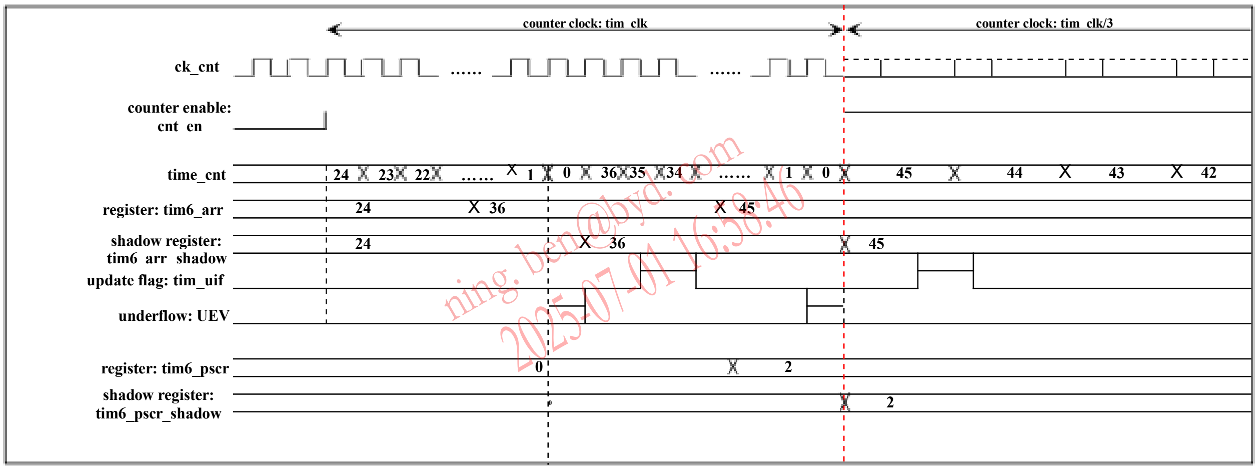


Figure 13.8 Decrement count sequence diagram

13.3.4.3. Center aligned count (increasing/decreasing count)

In the center alignment mode, the counter counts from 0 to the automatic overload value (the contents ofthe TIM6_ARR register) -1, generating the counter overflow event; It then counts down to 1 from the automatically overloaded value and generates a counter underflow event, after which it re-counts from 0.

The center alignment mode is valid when the CMS bit in the TIM6_CR1 register is not "00". When a channel is configured in output mode, its output comparison interrupt flag is set to 1 in the following mode, i.e. : Counter decrement count (center alignment mode 1, CMS = "01"), counter increment count (center alignment mode 2, CMS = " 10"), and counter increment/decrement count (center alignment mode 3, CMS = "11").

In this mode, the DIR direction bit of the TIM6_CR1 register cannot be written to a value, but is updated by the hardware and indicates the current counter direction.

Update events are generated each time a counter overflows and underflows occur, or the UG position 1 in the TIM6_EGR register (either by software or using a slave mode controller) can also generate an update event. In this case, the counter and the pre-divider counter will start counting again from 0.

UEV update events can be disabled by software simply by placing UDIS position 1 in the TIM6_CR1 register; This avoids updating the shadow register when a new value is written to the preloaded register; No update events occur until the UDIS bit is written to 0; However, the counter will still increment and decrement the count based on the current automatically overloaded value.

In addition, if the URS bit (update request selection) in the TIM6_CR1 register is set to 1, the UEV update event is generated at UG position 1, but the UIF flag is not set to 1 (therefore, no interrupt or DMA requests are sent); This is to avoid both update and capture interruptions when a capture event occurs and the counter is cleared.

When an update event occurs, all registers are updated and the update flag (UIF bit in the TIM6_SR register) is set to 1 (depending on the URS bit) :

- The auto-reload shadow register is updated with the preloaded value (TIM6_ARR)
- The preload value (contents ofthe TIM6_PSC register) will be reloaded in the buffer of the predivider

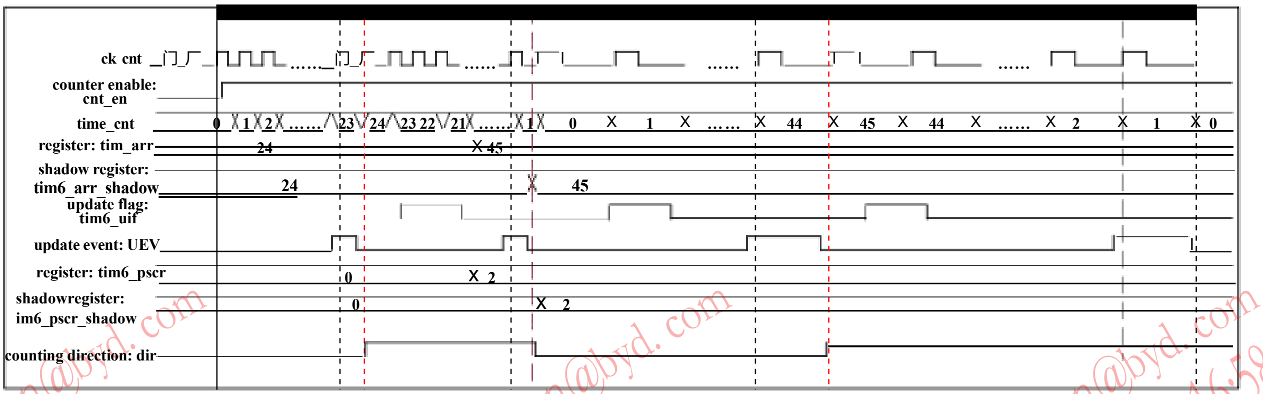


Figure 13.9 Center align count timing chart

13.3.5. Timer synchronization

TIM6 timers are connected together internally to achieve timer synchronization or linking. They can be synchronized in several modes: reset mode, gated mode, and trigger mode.

13.3.5.1. Slave mode: Reset mode

The counter and its pre-divider can be re-initialized when the trigger input signal changes. In addition, if the URS bit in the TIM6_CR1 register is low, an update event UEV is generated; Then, all the preloaded registers (TIM6_ARR and TIM6_CCRx) will be updated.

The counter starts counting against the internal clock and then operates normally until a trigger input (TRGI) rise occurs. When TRGI appears a rising edge, the counter clears to zero, and then starts counting again from 0. At the same time, the trigger flag (TIF bit in the TIM6_SR register) is set to 1, and after the interrupt or DMA is enabled, an interrupt or DMA request can be sent (depending on the TIE and TDE bits in the TIM6_DIER register).

In the following example, the increment counter clears when a rising edge appears on the TI1 input:

1. Configure the CC1S = 01 of the TIM6_CCMR1 register and configure the CC1 channel as input.
2. Configure the CC1P = 0 and CC1NP = 0 of the TIM6_CCER register, and select the rising edge contact detection of TI1.
3. Configure SMS = 0100 of the TIM6_SMCR register and set the timer to reset mode.
4. Configure TS = 00101 of the TIM6_SMCR register and select TI1 as the input source.
5. Configure the TIM6_DIR register CH1DIR = 1, and set TIM0_CH1 as the input.
6. Write CEN = 1 in the TIM6_CR1 register to start the counter.

13.3.5.2. Slave mode: Gated mode

The level of the input signal can be used to power the counter. As long as the TRGI is high, the counter starts counting against the internal clock and stops counting until the TRGI becomes low. When the counter is started or stopped, the TIF flag in the TIM6_SR register is set to 1. In gated mode, if CEN=0, the counter does not start regardless of the trigger input level.

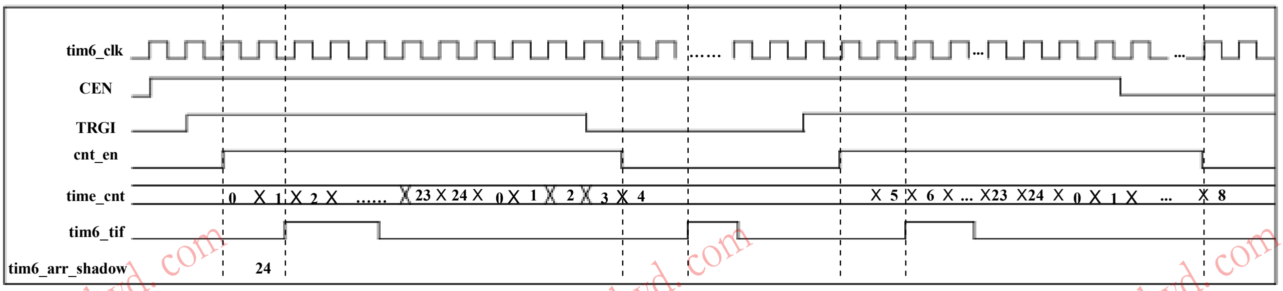


Figure 13.10 Gating mode timing diagram

In the following example, the counter counts at TI1 when the input is low voltage:

1. Configure the CC1S = 01 of the TIM6_CCMR1 register, configure the CC1 channel as input, and detect TI1.

2. Configure the CC1P = 1 and CC1NP = 0 of the TIM6_CCER register to detect TI1 low level.
3. Configure the SMS = 0101 of the TIM6_SMCR register and set the timer to the gated mode.
4. Configure TS = 00101 of the TIM6_SMCR register and select TI1 as the input source.
5. Configure the TIM6_DIR register CH1DIR = 1, and set TIM0_CH1 as the input.
6. Write CEN = 1 in the TIM6_CR1 register to start the counter.

13.3.5.3. Slave mode: Trigger mode

The counter can be started when an event occurs on the selected input. When TRGI appears with a rising edge, the counter starts counting against the internal clock and the TIF flag is set to 1.

In the following example, the counter starts when a rising edge appears on the TI2 input:

1. Configure the CC2S = 01 of the TIM0_CCMR1 register, configure the CC2 channel as input, and detect TI2
2. Configure the CC2P = 0 and CC2NP = 0 of the TIM0_CCER register to detect the rising edge of TI2.
3. Configure the SMS = 0110 of the TIM0_SMCR register to set the timer to trigger mode.
4. Configure the TS = 00110 of the TIM0_SMCR register and select TI2 as the input source.
5. Configure the TIM0_DIR register CH1DIR = 1, and configure TIM0_CH1 as input.

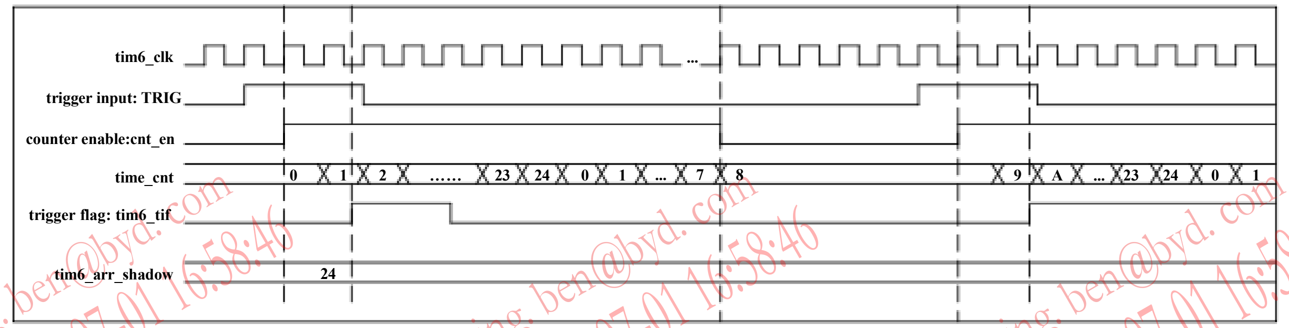


Figure 13.11 Trigger mode timing diagram

13.3.5.4. Slave mode: Combined reset + trigger mode

In this case, when the selected trigger input (TRGI) rise edge appears, the counter is reinitialized, a register update event is generated, and the counter is started. This mode is used for monopulse mode.

13.3.5.5. Slave mode: External clock mode 2+ trigger mode

External clock mode 2 can be used in combination with another slave mode (except external clock mode 1 and encoder mode). In this case, the ETR signal is used as an external clock input, and another input can be selected as the trigger input when operating in reset mode, gated mode, or trigger mode. The TS bit in the TIM6_SMCR register disallows the selection of ETR as TRGI. The counter is enabled when TRGI appears with a rising edge and the TIF flag is set to 1, then the counter counts when ETR appears with a rising edge. External clock mode 2 can be used in combination with another slave mode (except external clock mode 1 and encoder mode). In this case, the ETR signal is used as an external clock input, and another input can be selected as the trigger input when operating in reset mode, gated mode, or trigger mode. The TS bit in the TIM6_SMCR register disallows the selection of ETR as TRGI. The counter is enabled when TRGI appears with a rising edge and the TIF flag is set to 1, then the counter counts when ETR appears with a rising edge.

In the following example, when a rising edge appears in TI1, the counter counts at each rising edge of the ETR signal:

1. Configure the external clock mode 2 input circuit, configure the TIM0_SMCR register: ETF = 0000 (no filter), ETPS = 00 (disable pre-splitter), ETP = 0 (detect the rising edge of ETR), ECE = 1 (enable external clock mode 2)
2. Configure channel 1 to detect the rising edge of TI1: configure CC1S = 01 of the TIM0_CCMR1 register, configure CC1 channel as input, and detect TI1; Configure the CC1P = 0 and CC1NP = 0 of the TIM0_CCER register to determine polarity (detect rising edges).
3. Configure the SMS = 0110 of the TIM0_SMCR register to set the timer to trigger mode.
4. Configure the TS = 00101 of the TIM0_SMCR register and select TI1 as the input source.

The counter is enabled and the TIF flag is set to 1 when a rising edge appears in TRGI. The counter then counts at each rising edge of the ETR signal.

13.3.6. ADC synchronization

The timer can generate ADC trigger events through various internal signals, such as reset, enable, or compare events; Pulses emitted by the internal edge detector can also be generated, for example, rising and falling edges of OC4REF.

The trigger signal is emitted on the TRGO2 internal line that redirects to the ADC, and there are a total of nine possible events, which can be selected by the MMS2[3:0] bit in the TIM6_CR2 register.

Note:

1. **The clock of the slave peripheral (timer, ADC, etc.) that receives TRGO or TRGO2 signals must be enabled before the event can be received from the master timer; And when the trigger signal is received from the master timer, the clock frequency (pre-divider) must not be changed in real time.**
2. **The ADC clock must be enabled before events can be received from the master timer. The ADC clock must not be changed in real time when the trigger signal is received from the timer.**

13.3.7. Capture/compare channels

Each capture/comparison channel is built around a capture/comparison register (including a shadow register), a capture input part (digital filter, multiplexing, and pre-divider), and an output part (comparator and output control).

In the input phase, the corresponding TIx input is sampled and a filtered signal TIxF is generated. The edge detector with polarity selection then generates a signal (TIxFPx) that can be used either as a trigger input from the mode controller or as a capture command. The signal is first pre-divided (ICxPS) and then entered into the capture register.

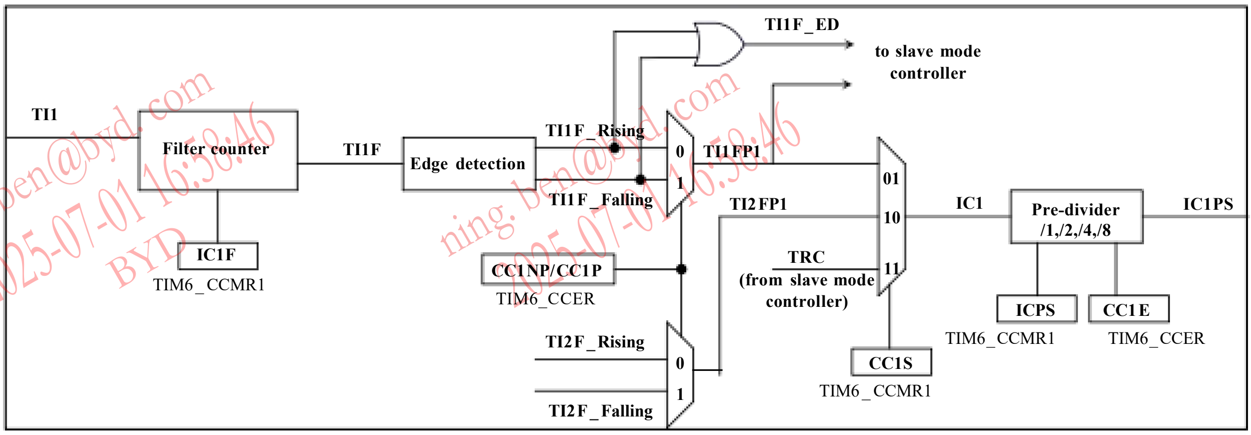


Figure 13.12 Capture/compare the input phase of a channel (example: Channel 1 input phase)

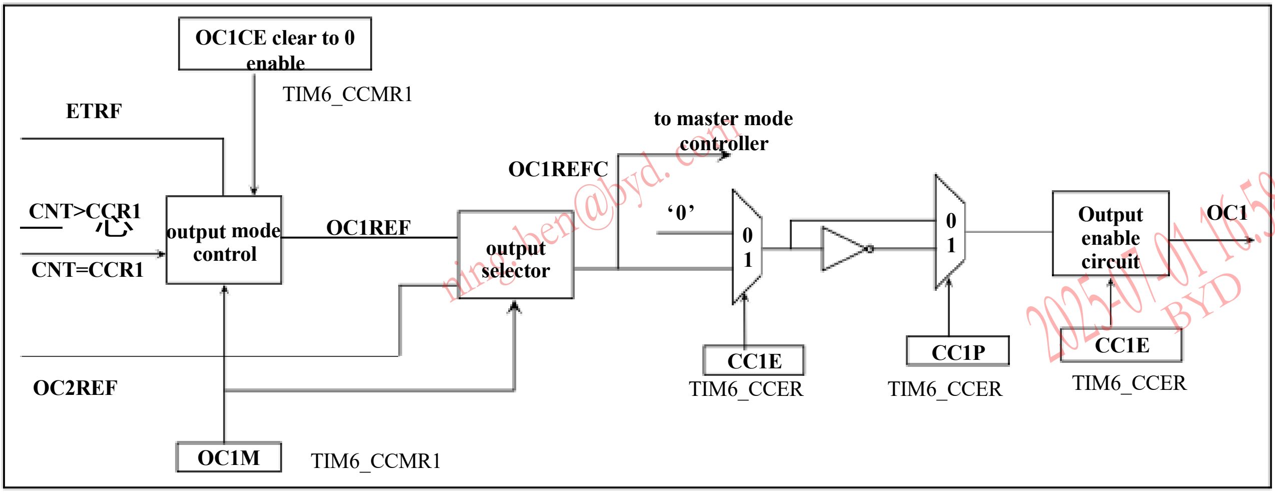


Figure 13.13 Capture/compare the output phases of channels (channels 1, 2, 3, and 4)

The capture/comparison module consists of a preloaded register and a shadow register, which can always be accessed through read and write operations.

In capture mode, the capture actually takes place in the shadow register, and the contents ofthe shadow register are then copied into the preloaded register.

In comparison mode, the contents ofthe pre-loaded register are copied into the shadow register, and then the contents ofthe shadow register are compared with the counter.

13.3.8. Input capture mode

In input capture mode, the capture/compare register (TIM6_CCRx) is used to latch the counter value when the corresponding ICx signal detects a jump edge. When a capture event occurs, the corresponding CCxIF flag (the TIM6_SR register) is set to 1 and an interrupt or DMA request, if enabled, can be sent. If the CCxIF flag is already high when a capture event occurs, the repeated capture flag CCxOF (TIM6_SR register) is set to 1. The CCxIF can be zeroed by software either by writing a "0" to the CCxIF or by reading the captured data stored in the TIM6_CCRx register. CCxOF clears zero when "0" is written.

The following example shows how to capture the counter's value into TIM6_CCR1 when a rising edge appears in the TI1 input:

1. Select valid input: Configure the CC1S bit of the TIM6_CCMR1 register to write 01, the CC1 channel is configured as the input, and IC1 is mapped to TI1.
2. According to the signal connected to the timer, configure the input filter to the required bandwidth (in this example, if the input is TI1, configure the IC1F bit ofthe TIM6_CCMR1 register). It is assumed that when the signal edge changes, the input signal jitter occurs within a maximum of 5 internal clock cycles. Therefore, the filter bandwidth needs to be set to more than 5 internal clock cycles, and the jump edge on TI1 can be determined after eight consecutive samples with new levels are detected (sampled at the fTIM6_CLK frequency). Then write 0011 to the IC1F bit ofTIM6_CCMR1.
3. Configure the CC1P and CC1NP bits ofthe TIM6_CCER register to write 0, and select the effective conversion edge rising edge on TI1.
4. Configure the IC1PSC = 00 of the TIM6_CCMR1 register, prohibit the input of the pre-divider, and perform the capture operation when the capture input detects a valid edge.
5. Configure the CC1E = 1 of register TIM6_CCER to allow the counter value to be captured into the capture register.
6. As needed, the input capture 1 interrupt can be enabled by configuring the CC1IE bit of TIM6_DIER to 1, and/or by configuring the CC1DE bit of this register to 1 to enable DMA requests.

When input capture occurs:

- When a valid jump edge occurs, the TIM6_CCRx register gets the value of the counter.
- Set the CCxIF flag to 1 (interrupt flag), so that the CCxOF capture overflow flag is set to 1 if at least two consecutive catches have occurred but the CCxIF flag has not been cleared to zero.
- generates interrupts based on the CCxIE bit.
- Generates a DMA request from the CCxDE bit.

To handle repeated capture, it is recommended to read the data before reading the capture overflow flag; This avoids the loss of duplicate capture information that may occur after reading the capture overflow flag and before reading the data.

Note:

1. **The corresponding CCxG position 1 in the TIM6_EGR register can be generated by software to generate IC interrupt and/or DMA request.**
2. **When configuring the TIM6_CCMRx register, configure the CCxS bit in the TIM6_CCMRx register twice. Select the mapping of the input mode. The ICxF and ICxPSC of the TIM6_CCMRx register are configured for the second time to select filtering and predivision coefficients.**

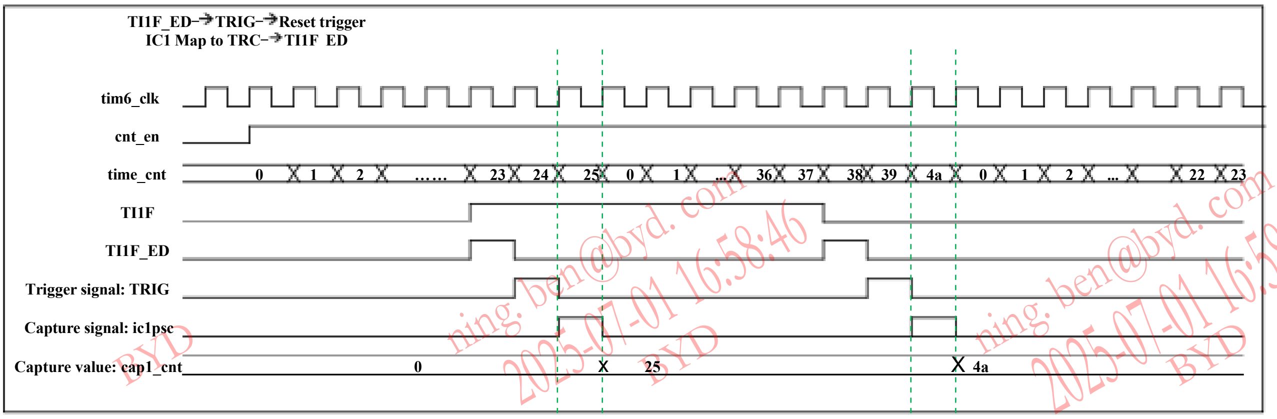


Figure 13.14 Enter capture timing diagram

13.3.9. PWM input mode

This pattern is a special case of the input capture pattern. The implementation steps are basically the same as the input capture mode, with the following differences:

- Two ICx signals are mapped to the same TIx input.
- The two ICx signals are effective at the edge, but of opposite polarity.
- Select one of the two TIxFP signals as the trigger input and configure the slave mode controller to reset mode.

The following example shows measurements corresponding to the PWM period (in the TIM6_CCR1 register) and duty cycle (in the TIM6_CCR2 register) for TI1 (depending on the CK_INT frequency and the value of the pre-divider) :

1. Use the valid input of TIM6_CCR1: Configure CC1S = 01 in the TIM6_CCMR1 register (select TI1), configure the CC1 channel as the input, and map IC1 to TI1.
2. Select the effective polarity of TI1FP1 (for capture and counter clearing in TIM0_CCR1) : Configure the CC1P bit and CC1NP bit of the TIM6_CCER register to 0 (effective for rising edge).
3. Select the valid input of TIM6_CCR2: Configure the CC2S = 10 in the TIM6_CCMR1 register (select TI1), configure the CC2 channel as the input, and map IC2 to TI1.
4. Select the effective polarity of TI1FP2 (for capture and counter clearing in TIM6_CCR2) : Configure the CC2P bit of the TIM6_CCER register to be 1 and the CC2NP bit to be 0 (effective for falling edge).
5. Select valid trigger input: Configure TS = 00101 in the TIM6_SMCR register (select TI1FP1).

6. Configure the slave mode control register to reset mode: configure SMS = 0100 in the TIM6_SMCR register.
7. Select the direction of TIM6_CH1: Configure the TIM6_DIR register CH1DIR = 1, and configure TIM6_CH1 as the input.
8. Enable capture: Set the CC1E bit and CC2E bit in the TIM6_CCER register to 1

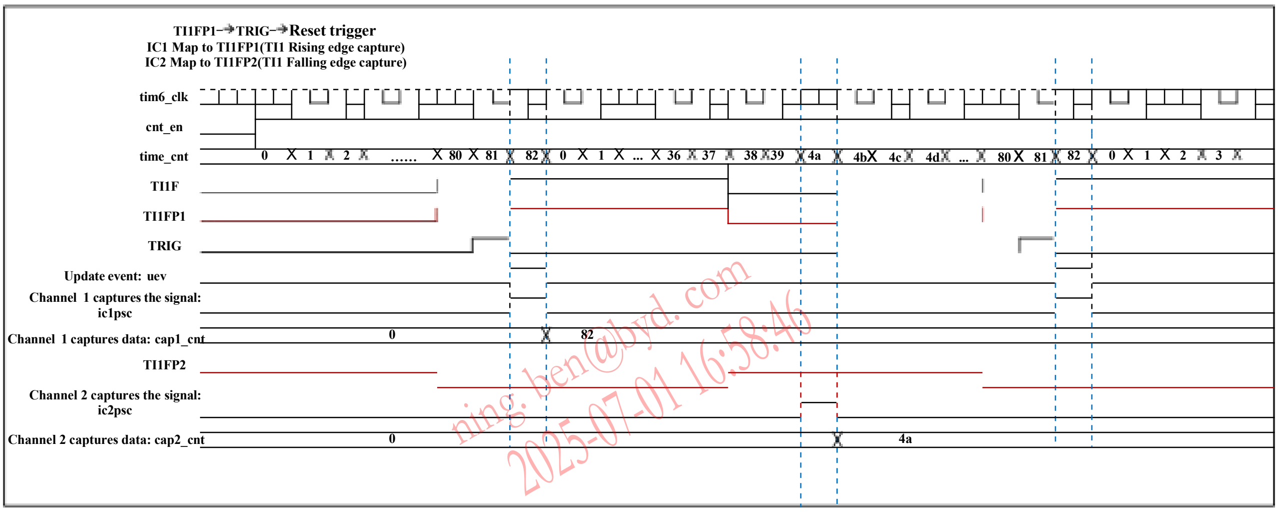

TI1FP1→TRIG→Reset trigger
IC1 Map to TI1FP1(TI1 Rising edge capture)
IC2 Map to TI1FP2(TI1 Falling edge capture)
tim6_clk
cnt_en
time_cnt
0 1 2 ...... 80 81 82 0 1 ... 36 37 38 39 4a 4b 4c 4d ... 80 81 82 0 1 2 3
TI1F
TI1FP1
TRIG
Update event: uev
Channel 1 captures the signal: ic1psc
Channel 1 captures data: cap1_cnt
0
82
TI1FP2
Channel 2 captures the signal: ic2psc
Channel 2 captures data: cap2_cnt
0
4a

Figure 13.15 PWM input mode timing

13.3.10. Forced output mode

In the output mode (CCxS bit =00 in the TIM6_CCMRx register), each output comparison signal (OCxREF and OCx) can be forcibly set to an active or invalid level directly by the software, regardless of any comparison results between the output comparison register and the counter.

To force the output comparison signal (OCxREF/OCx) to an active level, the user simply writes 0101 to the OCxM bit in the corresponding TIM6_CCMRx register; OCxREF then forces a high level (OCxREF is always active at a high level), and OCx takes the opposite value of the CCxP polarity bit. For example, CCxP=0 (the OCx high level is valid) => Forcibly set the OCx high level.

The OCxREF signal can be forcibly set to a low level by writing 0100 to the OCxM bit in the TIM6_CCMRx register.

In any case, the comparison between the TIM6_CCRx shadow register and the counter is still performed, and the flag is allowed to be set to 1, so the corresponding interrupt and DMA requests can be sent.

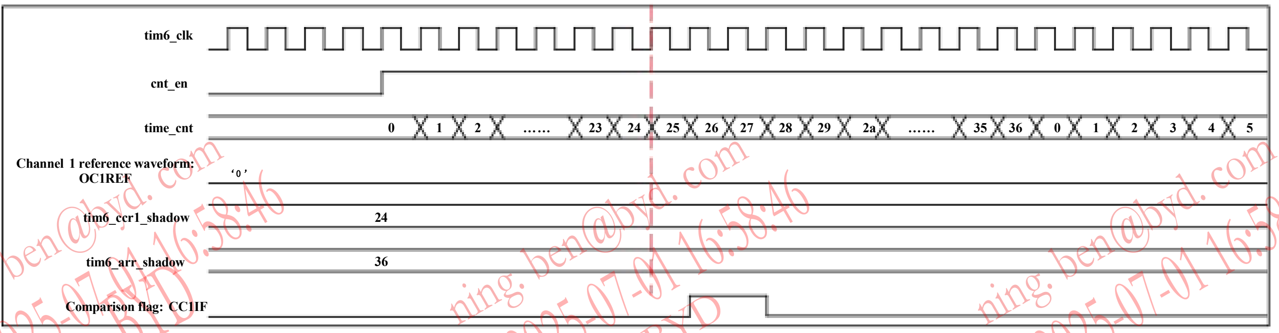

tim6_clk
cnt_en
time_cnt
0 1 2 ...... 23 24 25 26 27 28 29 2a 35 36 0 1 2 3 4 5
Channel 1 reference waveform: OC1REF
'0'
tim6_ccr1_shadow
24
tim6_arr_shadow
36
Comparison flag: CC1IF

Figure 13.16 Force output invalid level timing diagram(OCxM=0100)

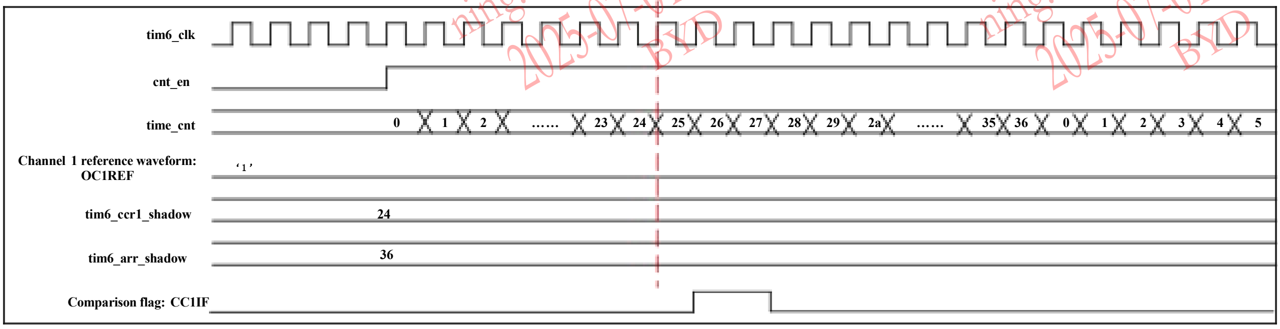


Figure 13.17 Forced output activel timing diagram(OCxM=0101)

13.3.11. Output comparison mode

This function is used to control the output waveform or indicate that a certain period of time has passed. Channels 1 through 4 can be used as outputs.

When there is a match between the capture/compare register and the counter, the output comparison function:

- assigns a programmable value to the corresponding output pin, defined by the output comparison mode (OCxM bits in the TIM6_CCMRx register) and the output polarity (CCxP bits in the TIM6_CCER register). When matched, the output pin can either hold its level (OCxM=0000), be set to active (OCxM=0001), invalid (OCxM=0010), or be flipped (OCxM=0011).
- sets the flag in the interrupt status register to 1 (the CCxIF bit in the TIM6_SR register).
- If the corresponding interrupt enable bit (CCxIE bit in the TIM6_DIER register) is set to 1, an interrupt is generated.
- If the corresponding enable bit (the CCxDE bit of the TIM6_DIER register) is set to 1, a DMA request is sent.

Using the OCxPE bit in the TIM6_CCMRx memory, the TIM6_CCRx register can be configured with or without the preloaded register.

In output comparison mode, updating event UEV has no effect on OCxREF and OCx output. The accuracy of synchronization can reach one count cycle of the counter. The output comparison mode can also be used to output monopulse (in monopulse mode).

Application steps:

1. Select the counter clock (internal, external, pre-divider).
2. Write the required data in the TIM6_ARR and TIM6_CCRx registers.
3. If you want to generate an interrupt request, you need to set the CCxIE position to 1.
4. Select an output mode. For example:
    a) When CNT matches CCRx, write OCxM = 0011 to flip the OCx output pin
    b) Write OCxPE = 0 to disable the preload register
    c) Write CCxP = 0 to select the high level effective polarity
    d) Write CCxE = 1 to enable output
5. Enable the counter by placing the CEN position 1 in the TIM6_CR1 register.

The TIM6_CCRx register can be updated at any time through the software to control the output waveform, provided that the pre-loaded register is not enabled (OCxPE= "0", otherwise the TIM6_CCRx shadow register will only be updated when the next update event UEV occurs).

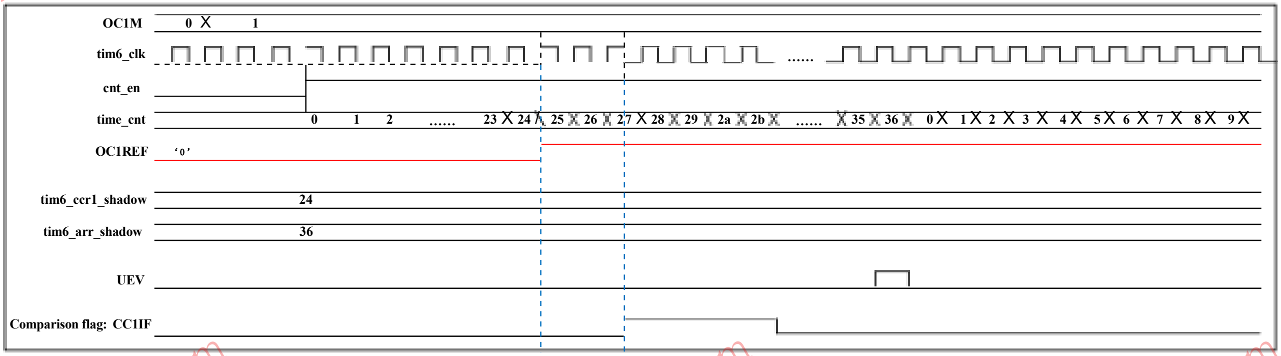


Figure 13.18 Output comparison mode(OCxM=0001)

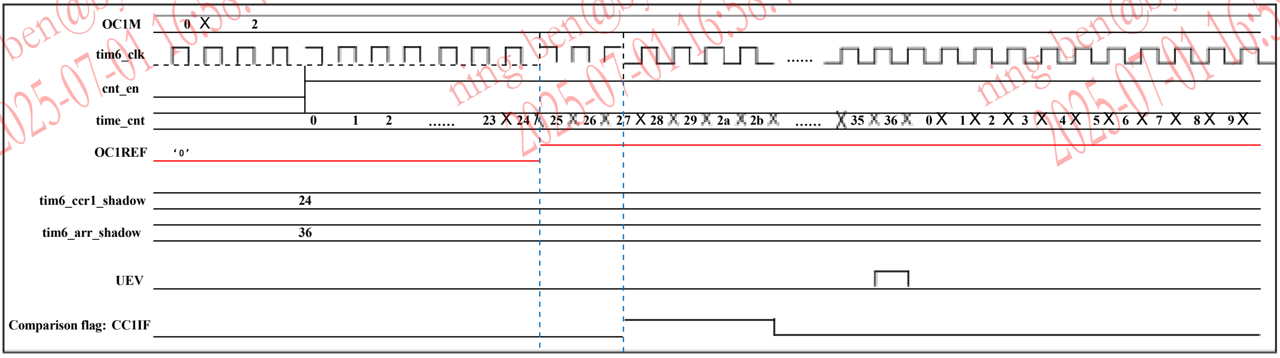


Figure 13.19 Output comparison mode(OCxM=0010)

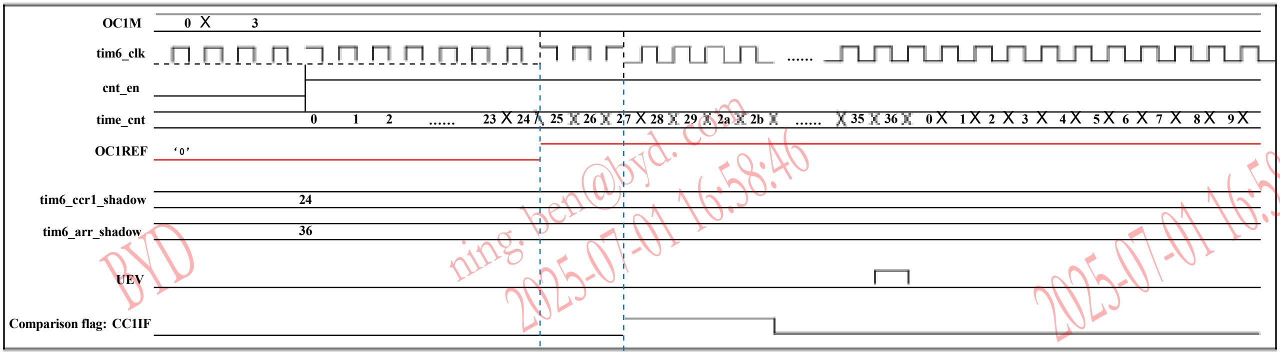


Figure 13.20 Output comparison mode(OCxM=0011)

13.3.12. PWM mode

The pulse width modulation mode can generate a signal whose frequency is determined by the TIM6_ARR register value, and whose duty cycle is determined by the TIM6_CCRx register value.

Each channel can independently select the PWM mode (one PWM for each OCx output) by writing "0110" (PWM mode 1) or "0111" (PWM mode 2) to the OCxM bit of the TIM6_CCMRx register. The corresponding preloaded register must be enabled by enabling OCxPE position 1 in the TIM6_CCMRx register, and finally by enabling ARPE position 1 in the TIM6_CR1 register to automatically reload the preloaded register (in up-count or center alignment mode).

Since the preloaded registers are only transferred to the shadow registers when an update event occurs, all registers must be initialized by placing UG position 1 in the TIM6_EGR register before starting the counter. The polarity of the OCx can be set by software in the CCxP bit in the TIM6_CCER register, which can be set to be active high or active low. The OCx output is enabled by placing the CCxE position 1 in the TIM6_CCER register.

13.3.12.1. PWM edge alignment mode

- Incremental count configuration

An incremental count is performed when the DIR bit in the TIM6_CR1 register is low. The following takes PWM mode 1 as an example, as long as TIM6_CNT<TIM6_CCRx, PWM reference signal OCxREF is high level, otherwise it is low level. If the comparison value in TIM6_CCRx is greater than the automatic overload value (in TIM6_ARR), OCxREF remains " 1"; If the comparison value is 0, OCxREF remains "0".

- Decrement count configuration

A decrement count is performed when the DIR bit in the TIM6_CR1 register is high. In PWM mode 1, as long as TIM6_CNT>TIM6_CCRx, the reference signal OCxREF is low, otherwise it is high. If the comparison value in TIM6_CCRx is greater than the automatic overload value in TIM6_ARR, the OCxREF remains "1" and it is not possible to produce a PWM waveform with a duty cycle of 0% in this mode.

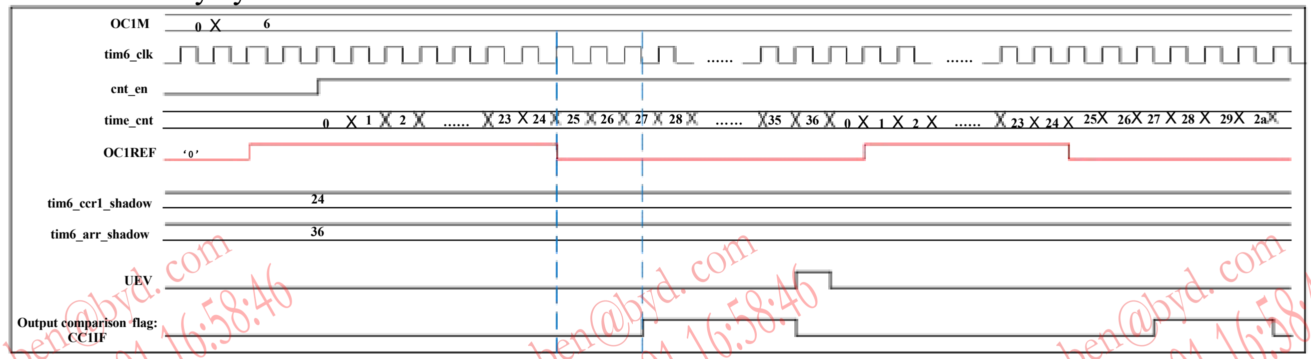

OC1M
0
6
tim6_clk
......
cnt_en
time_cnt
0 1 2 ...... 23 24 25 26 27 28 ...... 35 36 0 1 2 ...... 23 24 25 26 27 28 29 2a
OC1REF
'0'
tim6_ccr1_shadow
24
tim6_arr_shadow
36
UEV
Output comparison flag: CC1IF

Figure 13.21 PWM edge alignment mode 1(OCxM=0110)

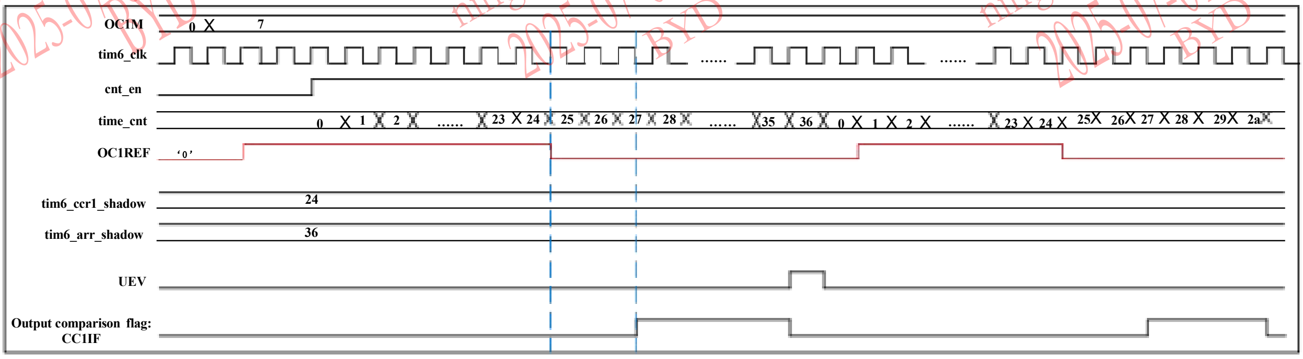

OC1M
0
7
tim6_clk
......
cnt_en
time_cnt
0 1 2 ...... 23 24 25 26 27 28 ...... 35 36 0 1 2 ...... 23 24 25 26 27 28 29 2a
OC1REF
'0'
tim6_ccr1_shadow
24
tim6_arr_shadow
36
UEV
Output comparison flag: CC1IF

Figure 13.22 PWM edge alignment mode 2(OCxM=0111)

13.3.12.2. PWM center alignment mode

When the CMS bit in the TIM6_CR1 register is not "00" (all other configurations have the same effect on the OCxRef/OCx signal), the center alignment mode takes effect. Depending on the configuration of the CMS bits, the comparison flag can be set to 1 when the counter is incremented, decrement, or both. The direction bit (DIR) in the TIM6_CR1 register is updated by hardware and cannot be changed by software.

Suggestions for using Center alignment mode:

- Center alignment mode is started with the current increment/decrement count configuration, which means that the counter will increment or decrement the count based on the value written to the DIR bit in the TIM6_CR1 register. In addition, you must not modify the DIR and CMS bits through software at the same time.
- Do not write to the counter while running center alignment mode, otherwise unexpected results will occur. In particular:
  - If a value greater than the automatically overloaded value is written in the counter (TIM6_CNT>TIM6_ARR), the direction is not updated. For example, if the counter is incrementing the count, continue incrementing the count.
  - If a value of 0 or TIM6_ARR is written to the counter, the count direction is updated, but no update event UEV is generated.
- The safest way to use the center alignment mode is to generate a software update before starting the counter (to UG position 1 in the TIM6_EGR register) and not to write to the counter while it is running.

13.3.13. Asymmetric PWM mode

In asymmetric mode, a programmable phase shift is allowed between the two center-aligned PWM signals generated. The frequency is determined by the value of the TIM6_ARR register, while the duty cycle and phase shift are determined by a pair of TIM6_CCRx registers. Two registers control the PWM during increasing count and decreasing count respectively, so that the PWM is adjusted once every half PWM cycle:

- OC1REFC (or OC2REFC) is controlled by TIM6_CCR1 and TIM6_CCR2
- OC3REFC (or OC4REFC) is controlled by TIM6_CCR3 and TIM6_CCR4

The two channels can independently select the asymmetric PWM mode (one OCx output per pair ofCCR registers) by simply writing "1110" (asymmetric PWM mode 1) or "1111" (asymmetric PWM mode 2) to the OCxM bit of the TIM6_CCMRx register.

When a given channel is used as an asymmetric PWM channel, its complementary channels can also be used. For example, if an OC1REFC signal is generated on channel 1 (asymmetric PWM mode 1), then either an OC2REF signal or an OC2REFC signal can be output on channel 2 due to asymmetric PWM mode 1.

13.3.14. Combined PWM mode

In combined PWM mode, programmable delay and phase shift are allowed between individual pulses ofthe generated two edge-aligned or center-aligned PWM signals. The frequency is determined by the value of the TIM6_ARR register, while the duty cycle and delay are determined by the two TIM6_CCRx registers, and the resulting signal OCxREFC consists of two logic or operations that reference PWM or a combination of logic and operations.

- OC1REFC (or OC2REFC) is controlled by TIM6_CCR1 and TIM6_CCR2

- OC3REFC (or OC4REFC) is controlled by TIM6_CCR3 and TIM6_CCR4

The two channels can independently select the combined PWM mode (one OCx output per pair ofCCR registers) by simply writing "1100" (combined PWM mode 1) or "1101" (combined PWM mode 2) to the OCxM bit of the TIM6_CCMRx register.

When a given channel is used as a combined PWM channel, its complementary channels must be configured in opposite PWM modes (for example, one channel is configured in combined PWM mode 1 and the other channel is configured in combined PWM mode 2).

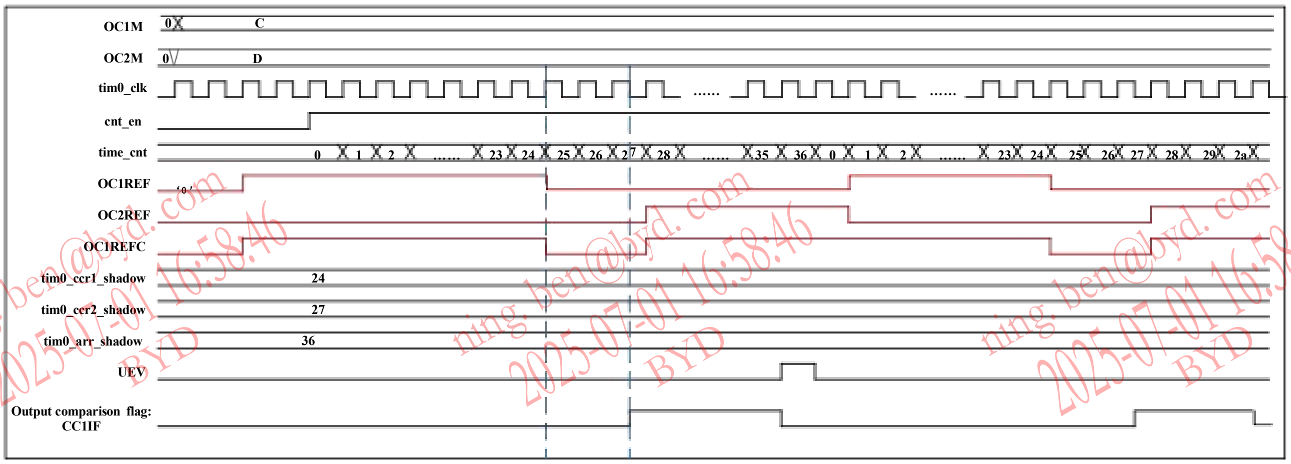


Figure 13.23 Combined PWM mode 1(OCxM=1100)

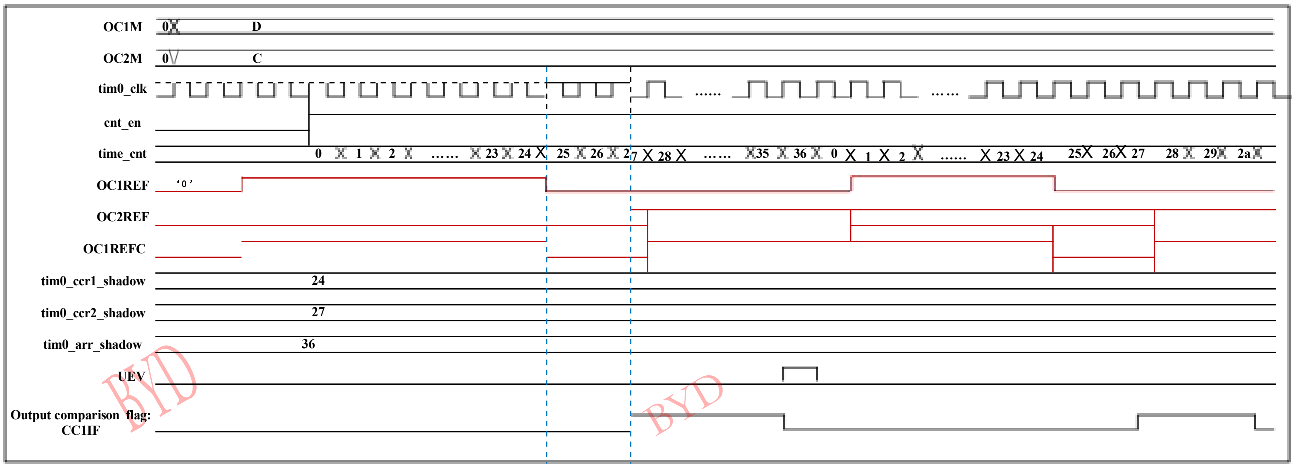


Figure 13.24 Combined PWM mode 2(OCxM=1101)

13.3.15. Single pulse mode

Monopulse mode (OPM) is a special case of the above mode. In this mode, the counter can be activated at the trigger of an excitation signal and can produce a pulse with programmable pulse width after a programmable delay.

Counters can be started from the mode controller. Waveforms can be generated in output comparison mode or PWM mode. Select the monopulse mode by placing the OPM position in the TIM6_CR1 register. In this way, the counter will automatically stop when the next update event UEV occurs.

A pulse can be generated correctly only if the comparison value is different from the initial counter value. Before starting (when the timer is waiting to be triggered), you must perform the following configurations:

- For increasing counts: CNT<CCRx<=ARR(note, 0<CCRx)
- When the count is decrement, CNT>CCRx

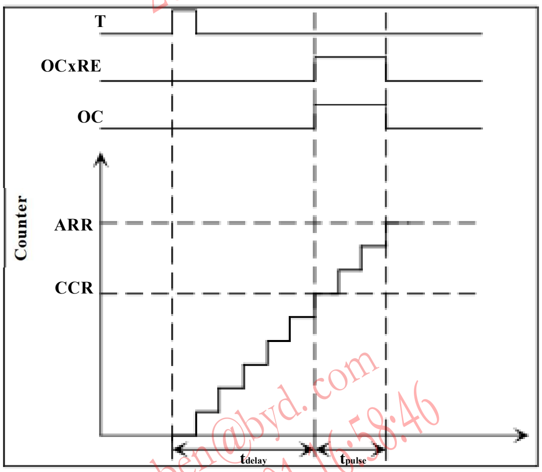


Figure 13.25 Single pulse mode

Special case: OCx fast enable:

In monopulse mode, edge detection ofthe TIx input sets the CEN position to 1, indicating enable counter. The output is then flipped when a comparison occurs between the counter value and the comparison value. However, multiple clock cycles are required to complete these operations, which limits the minimum possible delay (tDELAY minimum).

If you want to output the waveform with minimal delay, you can put the OCxFE position 1 in the TIM6_CCMRx register. This forces OCxRef (and OCx) to respond to the excitation signal regardless of the comparison. Its new level is the same as when the comparison match occurred. OCxFE works only when the channel is configured in PWM1 or PWM2 mode.

13.3.16. Retrigger monopulse mode (OPM)

This mode allows the counter to be activated at the trigger of an excitation signal, and can produce pulses of programmable length, but with the following differences from the non-reflexible monopulse mode:

- When triggered, the pulse is generated immediately (no programmable delay)
- If a new trigger occurs before the previous trigger completes, the pulse will be extended

The timer must be in slave mode, bit SMS[3:0] in the TIM6_SMCR register = "1000" (combined reset + trigger mode), and the OCxM[3:0] bit is set to "1000" or "1001" for retrigger OPM mode 1 or mode 2.

When the timer is configured in incremental count mode, the corresponding CCRx must be set to 0 (the ARR register sets the pulse length). If the timer is configured in decreasing count mode, CCRx must be higher than or equal to ARR.

Note: This mode cannot be used with the center-aligned PWM mode. CMS[1:0]=00 must be set in TIM6_CR1.

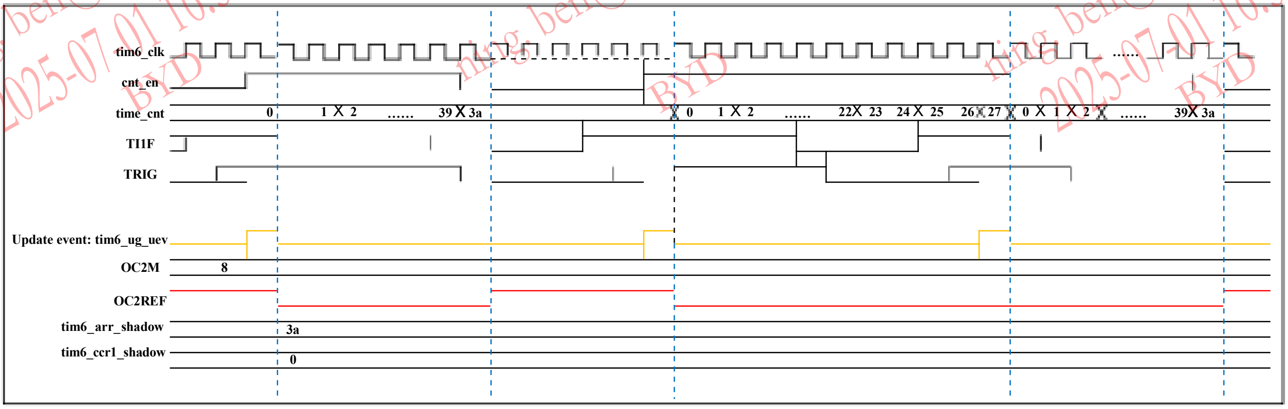


Figure 13.26 OPM mode 1 can be triggered again(OCxM=1000)

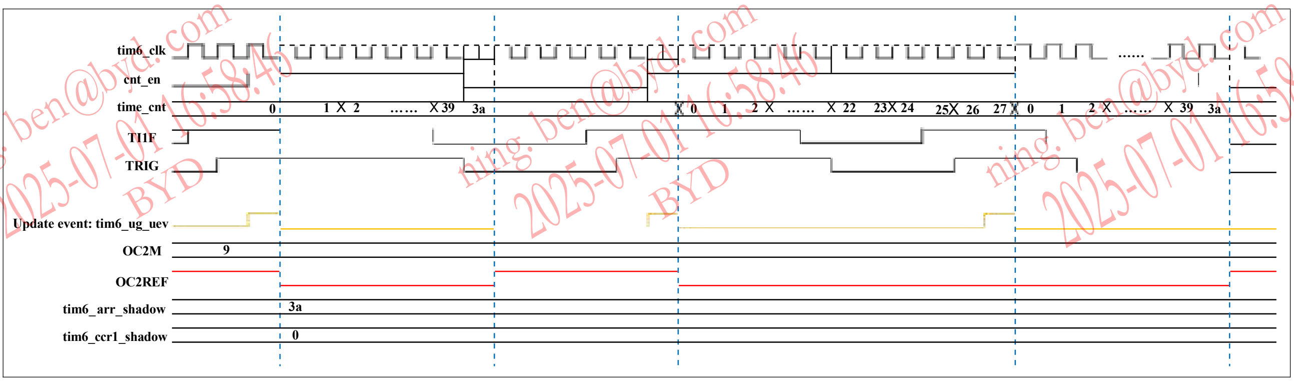


Figure 13.27 OPM mode 2 can be triggered again(OCxM=1001)

13.3.17. Clear the OCxREF signal when an external event occurs

For a given channel, applying a high level (corresponding to OCxCE enable position 1 in the TIM6_CCMRx register) to the ocref_clr_int (ETR after ETRF filter) input clears the OCxREF signal to zero. The OCxREF signal will remain low until the next update event (UEV) occurs. This function can only be used in output comparison mode and PWM mode. Does not work in forced mode.

When OCxCE is enabled, the ETR must be configured as follows:

1. The external trigger pre-divider must be turned off: ETPS[1:0] position "00" in the TIM6_SMCR register.
2. ECE position "0" in the external clock mode 2: TIM6_SMCR register must be disabled.
3. External trigger polarity (ETP) and external trigger filter (ETF) can be configured according to user needs.

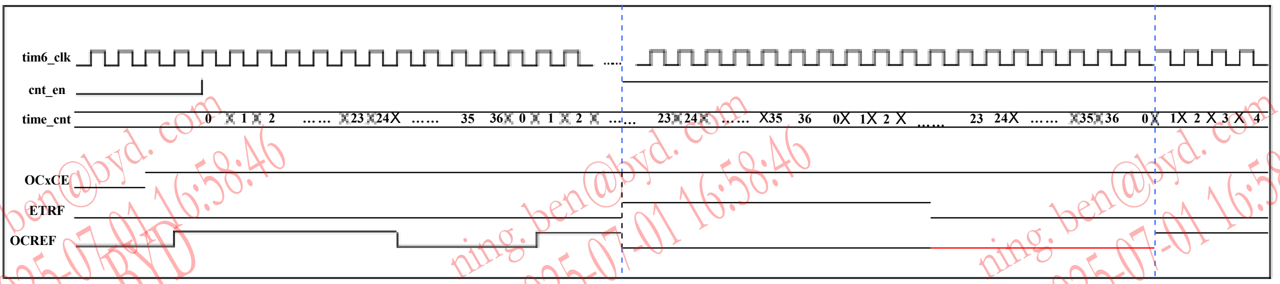


Figure 13.28 Clear OCxREF of TIM6

13.3.18. Encoder interface mode

When selecting the encoder interface mode, if the counter counts only at the TI1 edge, write SMS= 0001 in the TIM6_SMCR register; If the counter counts only at the TI2 edge, write SMS = 0010; If the counter counts at both TI1 and TI2 edges, SMS = 0011 is written.

Select TI1 and TI2 polarity by programming the CC1P and CC2P bits ofthe TIM6_CCER register. If necessary, the input filter can also be programmed. CC1NP and CC2NP must be kept low.

Two inputs TI1 and TI2 are used to connect the orthogonal encoder. If the counter is enabled (write "1" in the CEN bit of the TIM6_CR1 register), the clock of the counter is provided by every active signal conversion on TI1FP1 or TI2FP2. TI1FP1 and TI2FP2 are the signals ofTI1 and TI2 after input filter and polarity selection, if not filtered and reversed, TI1FP1=TI1, TI2FP2=TI2. Counting pulses and directional signals are generated based on a sequence of signal conversions from the two inputs. According to the signal conversion sequence, the counter increments or decrement the count accordingly, and the hardware modifies the DIR bit of the TIM6_CR1 register accordingly. The DIR bit is calculated when any input (TI1 or TI2) undergoes a signal conversion, whether the counter counts only at the edge of TI1 or TI2, or at both TI1 and TI2.

The encoder interface mode is equivalent to an external clock with direction selection. This means that the counter only counts continuously between 0 and the automatically overloaded value in the TIM6_ARR register (increasing the count from 0 to ARR, or decreasing the count from ARR to 0, depending on the specific direction). Therefore, TIM6_ARR must be configured before startup. Again, the capture, compare, repeat counter, and trigger output functions continue to work properly. Encoder mode and external clock mode 2 are incompatible and therefore cannot be selected at the same time.

Note: When encoder mode is enabled, the predivider must be set to zero.

In this mode, the counter is automatically modified according to the speed and direction of the orthogonal encoder, so its contents always represent the position of the encoder. The counting direction corresponds to the rotation direction of the connected sensor. The following table summarizes the possible combinations (assuming TI1 and TI2 do not switch at the same time).

Effective edge	**The level of the relative signal (TI1FP1 for TI2, TI2FP2 for TI1)**	**TI1FP1 signal**		**TI2FP2 signal**	
		rise	**descend**	**rise**	**descend**
Count only in TI1	high	Down count	Up count	No count	No count
	low	Up count	Down count	No count	No count
Count only in TI2	high	No count	No count	Up count	Down count
	low	No count	No count	Down count	Up count
Count on TI1 and TI2	high	Down count	Up count	Up count	Down count
	low	Up count	Down count	Down count	Up count

13.3.19. UIF bit remapping

The UIFREMAP bit in the TIM6_CR1 register forces the update interrupt flag UIF to be continuously copied to bit 31(TIM6_CNT[31]) in the timer counter register. This automatically reads the counter value and the potential reversal condition emitted by the UIFCPY flag. In certain cases, this simplifies calculations by avoiding race conditions when processing is shared between background tasks (counter reads) and interrupts (update interrupts).

There is no delay between enabling the UIF and UIFCPY flags.

13.3.20. Timer input XOR function

Through the TI1S bit in the TIM6_CR2 register, the input filter of channel 1 can be connected to the output of the XOR gate, thus combining the three input pins TIM6_CH1 to TIM6_CH3.

Xor outputs can be used with all timer input functions such as trigger or input capture.

13.3.21. DMA continuous transfer mode

The TIM6 timer can generate multiple DMA requests based on a single event. The main purpose is to be able to reprogram a portion of the timer multiple times without software overhead, but can also be used to periodically read multiple registers in a row.

The DMA controller target is unique and must point to the virtual register TIM6_DMAR. When a given timer event occurs, the timer initiates a DMA request sequence (burst). Each write to the TIM6_DMAR register redirects to one of the timer registers.

The DBL[4:0] bit in the TIM6_DCR register sets the DMA continuous transfer length. When a read or write access is made to the TIM6_DMAR address, the timer makes one successive transfer, that is, the number of transfers (by half a word or byte).

The DBA[4:0] bit in the TIM6_DCR register defines the DMA base address for the DMA transfer (when read/write access is performed via the TIM6_DMAR address). DBA is defined as the offset calculated from the TIM6_CR1 register address:

Example:

- 00000: TIM6_CR1
- 00001: TIM6_CR2
- 00010: TIM6_SMCR

For example, the timer DMA continuous transfer function is used to update the contents ofthe CCRx register (x=1, 2, 3, 4) to multiple half-words transferred to the CCRx register via DMA after an update event occurs.

The specific steps are as follows:

1. Configure the corresponding DMA channel as follows:
    a) The destination data end address of the DMA channel is the DMAR register address.
    b) The DMA channel source data end address is the RAM buffer address containing the data to be transferred to the CCRx register via DMA.
    c) Amount of data to be transferred = 4
    d) R_power = 0x02

e) Select the request source TIM6_UP

2. Configure the DCR register by configuring the DBA and DBL bit fields as follows: DBL=3, DBA= 0xD.
3. Enable TIM6 update DMA request (UDE position 1 in DIER register).
4. Enable TIM6
5. Enable the DMA channel

This example applies if each CCRx register is updated only once. If each CCRx register is updated twice, the amount of data to be transferred should be 8. The RAM buffers containing data1, data2, data3, data4, data5, data6, data7, and data8 are used as examples. The data is transferred to the CCRx register as follows: during the first update DMA request, data1 is transferred to CCR1, data2 to CCR2, data3 to CCR3, and data4 to CCR4; During the second update DMA request, data5 transfers to CCR1, data6 transfers to CCR2, data7 transfers to CCR3, and data8 transfers to CCR4.

13.4. TIM6 register

Base address: 0x5000 0000

Offset address	Register	Description
0x04	RCU_EN	Peripheral module clock control register
0x10	ANA_CFG	Analog module switch register
0x44	TIM_CLK_SEL	TIM clock select register
0X48	CLK_SEL_RDY	Clock switch completion flag register

Base address: TIM6: 0x5006 0600

Offset address	Register	Description
0x00	TIM6_CR1	Control register 1
0x04	TIM6_CR2	Control register 2
0x08	TIM6_SMCR	Slave mode control register
0x0C	TIM6_DIER	DMA/ interrupt enable register
0x10	TIM6_SR	Status register
0x14	TIM6_EGR	Event generation register
0x18	TIM6_CCMR1	Capture/compare mode register 1
0x1C	TIM6_CCMR2	Capture/compare mode register 2
0x20	TIM6_CCER	Capture/compare mode register
0x24	TIM6_CNT	counter
0x28	TIM6_PSC	A pre-divider
0x2C	TIM6_ARR	Automatic reload register
0x34	TIM6_CCR1	Capture/compare register 1
0x38	TIM6_CCR2	Capture/compare register 2
0x3C	TIM6_CCR3	Capture/compare register 3
0x40	TIM6_CCR4	Capture/compare register 4
0x48	TIM6_DCR	DMA control register
0x4C	TIM6_DMAR	Full transport DMA address
0x68	TIM6_DIR	Input/output control register
0x6C	TIM6_EN	The TIM6 module is enabled

13.4.1. Peripheral module clock control register (RCU_EN)

Address offset: 0x04

Reset value: 0x0000 0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.				DIV_ CLKEN	GPIOD_ CLKEN	GPIOC_ CLKEN	GPIOB_ CLKEN	GPIOA_ CLKEN	DMA_ CLKEN	CRC_ CLKEN	ADC_ CLKEN	WDT_ CLKEN	TIM7_ CLKEN	TIM6_ CLKEN	TIM5_ CLKEN
				RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

26	GPIOD_CLKEN	GPIOD module working enable:

		1: Work 0: Off, the default is 0
25	GPIOC_CLKEN	GPIOC module working enable: 1: Work 0: Off, the default is 0
24	GPIOB_CLKEN	GPIOB module working enable:: 1: Work 0: Off, the default is 0
23	GPIOA_CLKEN	GPIOA module working enable: 1: Work 0: Off, the default is 0
22	DMA_CLKEN	DMA(include DMA_SRAM/DMAMUX) module working enable: 1: Work 0: Off, the default is 0
17	TIM6_CLKEN	TIM6 module working enable: 1: Work 0: Off, the default is 0

13.4.2. Analog module switch register (ANA_CFG)

Address offset: 0x10

Reset value: 0x0000 1850

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

15:13	12	11	10	9	8:7	6	5	4	3 2	1:0
Res.	PD_TEMP	PD_ADC	VREF_SEL	VREF_VOL_SEL	VREF_IN_ADC_SEL	PD_ADC_IN_VREF	XTAL_HFR_SEL	XTAL_SEL	XTAL_BYP_SEL	XTAL_EN_SEL
	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

5	XTAL_HFR_SEL	Analog high frequency crystal oscillator circuit current configuration selection: 0: 175μA(4MHz crystal oscillator is recommended) 1: 300μA(8MHz crystal oscillator is recommended)
4	XTAL_SEL	When a passive crystal oscillator circuit is available 0: Select 32K 1: Select 4M/8M. Default value 1
3:2	XTAL_BYP_SEL	Active crystal oscillator path selection: 00: Channel closed, do not use active crystal oscillator 01: Select channel A (PA5) for input 10: Select channel B (PA2) for input 11: Reserved Note: Select any active crystal path, and the passive crystal vibration circuit is turned

		off at the same time. The default value is 00.
1:0	XTAL_EN_SEL	Passive crystal path selection: 00: The passive crystal vibration circuit is turned off and no passive crystal vibration is used 01: Select path A (PA5, PA4) 10: Select channel B (PA2, PA1) for input 11: Reserved When XTAL_BYP_SEL[1:0] = 00 and XTAL_EN_SEL[1:0]! = 00, turn on the passive crystal vibration circuit, the default value is 00. Note: Passive crystal oscillator must be enabled when active crystal oscillator is not enabled.

13.4.3. TIM clock selector register (TIM_CLK_SEL)

Address offset: 0x44

Reset value: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIM7CLK		TIM6CLK		TIM5CLK		TIM4CLK		TIM3CLK		TIM2CLK		TIM1CLK		TIM0CLK	
RW		RW		RW		RW		RW		RW		RW		RW	

13:12	TIM6CLK	TIM6 Clock selection: 00: Select the PLL48M clock 01: Select the LIRC 32K clock 1x: Select XTAL (passive crystal (32768Hz/4MHz/8MHz) and active crystal (1MHz~48MHz))

13.4.4. Clock switch completion flag register (CLK_SEL_RDY)

Address offset: 0x48

Reset value: 0x0000 03FF

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

15:10	9	8	7	6	5	4	3	2	1	0
Res.	TIM7RDY	TIM6RDY	TIM5RDY	TIM4RDY	TIM3RDY	TIM2RDY	TIM1RDY	TIM0RDY	PWM1RDY	PWM0RDY
	R	R	R	R	R	R	R	R	R	R

8	TIM6RDY	TIM6 Clock switchover completion sign: 0: The clock is being switched. 1: The clock switchover is complete

13.4.5. TIM6 Control register 1(TIM6_CR1)

Address offset: 0x00

Reset value: 0x0000 0000

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

15:12	11	10	9 8	7	6 5	4	3	2	1	0
Res.	UIFREMAP	Res.	CKD[1:0]	ARPE	CMS[1:0]	DIR	OPM	URS	UDIS	CEN
	RW		RW	RW	RW	RW	RW	RW	RW	RW

31:12		Reserved
11	UIFREMAP	UIF status bit remapping: 0: no remapping. The UIF status bit is not copied to bit 31 of the TIM6_CNT register. 1: enables remapping. The UIF status bit is copied to bit 31 of the TIM6_CNT register.
10	-	Reserved
9:8	CKD[1:0]	Clock frequency division: This bit field indicates the frequency division ratio between the timer clock (TIM6_CLK) frequency and the dead band and sampling clock (t_{DTS}) used by the timer clock (TIM6_CLK) and digital filters (ETR, TIx) 00: $t_{DTS}=t_{TIM6\_CLK}$ 01: $t_{DTS}=2*t_{TIM6\_CLK}$ 10: $t_{DTS}=4*t_{TIM6\_CLK}$ 11: Reserved Note: This bit is configured before the TIM6_EN register is configured
7	ARPE	Automatic reload preload enable: 0: The TIM6_ARR register is not buffered 1: TIM6_ARR register for buffering
5:6	CMS[1:0]	Center alignment mode selection: 00: Edge alignment mode. The counter increments or deciles the count according to the direction bit (DIR). 01: Center alignment mode 1. The counter alternates between increasing and decreasing counts. The output comparison interrupt flag for the channel configured for output (CCxS=00 in the TIM6_CCMRx register) is set to 1 only when the counter decrement the count. 10: Center alignment mode 2. The counter alternates between increasing and decreasing counts. The output comparison interrupt flag for the channel configured for output (CCxS=00 in the TIM6_CCMRx register) is set to 1 only when the counter increases the count. 11: Center alignment mode 3. The counter alternates between increasing and decreasing counts. When the counter increments or decays the count, the output comparison interrupt flag is set to 1 for the channel configured to output (CCxS=00

		in the TIM6_CCMRx register). Note: As long as the counter is enabled (CEN=1), you cannot switch from edge alignment mode to center alignment mode. Note: This bit is configured before the TIM6_EN register is configured
4	DIR	Counter counting direction selection: 0: indicates that the counter is incremented 1: indicates the counter decrement count Note: When the timer is configured in center alignment mode or encoder mode, this bit is read-only.
3	OPM	Single pulse mode selection: 0: The counter does not stop counting when an update event occurs 1: The counter stops counting when the next update event occurs (clears the CEN bit to zero)
2	URS	Update request source: This bit is set to 1 and cleared by the software to select the UEV event source. 0: When enabled, all of the following events generate update interrupts or DMA requests. Such incidents include: - The counter overflows or underflows - Set UG to position 1 - Via update events generated from the schema controller 1: When enabled, only counter overflows/underflows generate update interrupts or DMA requests.
1	UDIS	Update prohibited: This bit is set to 1 and cleared by the software to enable/disable UEV event generation. 0: UEV is enabled. Update (UEV) events can be generated by one of the following events: - The counter overflows or underflows - Set UG to position 1 - The value of the shadow register is then updated through the update event generated from the pattern controller. 1: Prohibit UEV. No update event is generated, and the values of each shadow register (ARR, PSC, and CCRx) remain unchanged. However, if the UG position 1, or hardware reset is received from the slave mode controller, the counter and pre-divider are reinitialized.
0	CEN	Counter enable: 0: disables the counter 1: Enables the counter Note: External clock, gated mode and encoder mode can be used only if CEN position 1 is set by software in advance, while trigger mode can be set by hardware automatically.

13.4.6. TIM6 Control register 2(TIM6_CR2)

Address offset: 0x04

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19:16
Res.								MMS2[3:0]				Res.
								RW				

15:8	7	6	5	4	3:0
Res.	TI1S	MMS[2:0]			Res.
	RW	RW			

31:24	-	Reserved
23:20	MMS2[3:0]	Main mode selection 2: These bits select the information that will be sent to the ADC for synchronization (TRGO2); The combination of these bits is as follows: 0000: Reset -- UG bit in TIM6_EGR register used as trigger output (TRGO2) 0001: Enable -- The counter enable signal CNT_EN is used as the trigger output (TRGO2). 0010: Update -- Select the update event as the trigger output (TRGO2). 0011: Comparison pulse - When the CC1IF flag is set to 1 (even if it is already high), the trigger output (TRGO2) sends a positive pulse whenever a capture or comparison match occurs. 0100: Comparison -- OC1REF signal used as trigger output (TRGO2) 0101: Comparison -- OC2REF signal used as trigger output (TRGO2) 0110: Comparison -- OC3REF signal used as trigger output (TRGO2) 0111: Comparison -- OC4REF signal used as trigger output (TRGO2) 1000: Compare pulse - when OC4REF rises or falls, a pulse is generated on TRGO2 Other: Reserved Note: The ADC clock must be enabled before events can be received from the master timer. The ADC clock must not be changed in real time when the trigger signal is received from the master timer. Note: This bit is configured before the TIM6_EN register is configured
19:8	-	Reserved
7	TI1S	TI1 Selection: 0: Pin TIM6_CH1 is connected to the TI1 input 1: TIM6_CH1, CH2, and CH3 pins connected to TI1 input (XOR combination)
6:4	MMS[2:0]	Master mode selection: These bits select the information to be sent to the slave timer in master mode for synchronization (TRGO). The combination of these bits is as follows:

		000: Reset, UG bit in TIM6_EGR register used as trigger output (TRGO). 001: Enable, the counter enable signal CNT_EN is used as a trigger output (TRGO). The trigger output can be used to start multiple timers at the same time, or to control enabling the slave timer over a period of time. In gated mode, the counter enable signal is the logic and generation of the CEN control bit and trigger input signal. There is a delay in TRGO when the counter enable signal is controlled by the trigger input. 010: Update, select Update event as trigger output (TRGO). 011: Compare pulses. Once an input capture or compare match event occurs, the trigger output sends a positive pulse (TRGO) when the CC1IF flag is set to 1 (even if it is high). 100: Comparison, OC1REF signal used as trigger output (TRGO) 101: Comparison, OC2REF signal used as trigger output (TRGO) 110: Comparison, OC3REF signal used as trigger output (TRGO) 111: Comparison, OC4REF signal used as trigger output (TRGO) Note: The clock of the slave timer or ADC must be enabled before events can be received from the master timer. When the trigger signal is received from the master timer, the clock of the slave timer or ADC must not be changed in real time. Note: This bit is configured before the TIM6_EN register is configured
3:0	-	Reserved

13.4.7. Slave mode control register (TIM6_SMCR)

Address offset: 0x08

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19:17	16
Res.										TS[4:3]		Res.	SMS[3]
										RW			RW

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETP	ECE	ETPS		ETF				MSM	TS[2:0]			Res.	SMS[2:0]		
RW	RW	RW		RW				RW	RW				RW		

31:22	-	Reserved
21:20	TS[4:3]	Trigger selection: This bit is the bit[4:3] of TS[4:0], note see TS[4:0]
19:17	-	Reserved
16	SMS[3]	Select from mode: This bit is the bit[3] of SMS[3:0], note see SMS[2:0]
15	ETP	External trigger polarity: This bit selects whether the ETR or the inverse of the ETR is used to trigger the

		operation 0: ETR is not reversed, high or rising edge is effective. 1: ETR reverse phase, low level or falling edge effective.
14	ECE	External clock enabled: This bit enables external clock mode 2. 0: disables external clock mode 2. 1: indicates that external clock mode 2 is enabled. The counter clock is provided by any effective edge of the ETRF signal. Note: 1. Setting ECE position 1 has the same effect as selecting external clock mode 1 and connecting TRGI to ETRF (SMS= 111 and TS=00111). 2, External clock mode 2 can be used with the following slave modes: reset mode, gate mode, and trigger mode. In this case, the TRGI must not connect to the ETRF (the TS bit must not be 00111). 3. External clock mode 1 and external clock mode 2 cannot be used simultaneously.
13:12	ETPS	External trigger predivider: The ETRP frequency of the external trigger signal must not exceed 1/4 of the TIMx_CLK frequency. The ETRP frequency can be reduced by enabling the pre-divider. This approach is useful when entering a fast external clock. 00: The pre-divider is off 01:2 Frequency Dividing ETRP frequency 10:4 FRC ETRP frequency 11:8 Divided ETRP frequency Note: This bit is configured before the TIM6_EN register is configured
11:8	ETF	External trigger filter: This bitfield defines the sampling frequency of ETRP signals and the bandwidth of digital filters applicable to ETRP. The digital filter consists of an event counter, where each N consecutive events is considered a valid output edge: 0000: No filter, sampling at f_{DTS} frequency 0001: $f_{SAMPLING}$=f_{TIM6_CLK}, N=2 0010: $f_{SAMPLING}$=f_{TIM6_CLK}, N=4 0011: $f_{SAMPLING}$=f_{TIM6_CLK}, N=8 0100: $f_{SAMPLING}$=f_{DTS}/2, N=6 0101: $f_{SAMPLING}$=f_{DTS}/2, N=8 0110: $f_{SAMPLING}$=f_{DTS}/4, N=6 0111: $f_{SAMPLING}$=f_{DTS}/4, N=8 1000: $f_{SAMPLING}$=f_{DTS}/8, N=6 1001: $f_{SAMPLING}$=f_{DTS}/8, N=8 1010: $f_{SAMPLING}$=f_{DTS}/16, N=5 1011: $f_{SAMPLING}$=f_{DTS}/16, N=6 1100: $f_{SAMPLING}$=f_{DTS}/16, N=8 1101: $f_{SAMPLING}$=f_{DTS}/32, N=5 1110: $f_{SAMPLING}$=f_{DTS}/32, N=6

		1111: $f_{SAMPLING}=f_{DTS}/32$, N=8 Note: This bit is configured before the TIM6_EN register is configured
7	MSM	Master/slave mode: 0: no operation is performed. 1: The trigger input event (TRGI) action of the current timer is delayed so that the current timer is perfectly synchronized with its slave timer (via TRGO). This setting is useful when multiple timers are synchronized by a single external event. Note: This bit is configured before the TIM6_EN register is configured
6:4	TS[2:0]	Trigger selection: This bit field selects the trigger input that will be used to synchronize the counter. The bit of TS [4:3] is located at the bit of TIM6_SMCR [21:20] 00000: internal trigger 0 (ITR0)-------->TIM0_TRGO 00001/00010/00011: Reserved 00100: TI1 Edge Detector (TI1F_ED) 00101: Filtered timer input 1 (TI1FP1) 00110: Filtered timer input 2 (TI2FP2) 00111: External Trigger Input (ETRF) Other values: reserved Note: These bits can only be changed when not in use (for example, when SMS=000) to avoid erroneous edge detection during conversion. Note: This bit is configured before the TIM6_EN register is configured
3	-	Reserved
2:0	SMS[2:0]	Select from mode: When selecting an external signal, the effective edge of the trigger signal (TRGI) is related to the polarity selected on the external input (see Input Control Register and Control Register instructions). The SMS bit[3] is located at the bit of TIMx_SMCR [16] 0000: Disable slave mode. If CEN= "1", the pre-divider clock is provided directly by the internal clock. 0001: Encoder mode 1, the counter increments/decays the count along the edge of TI1FP1 according to the TI2FP2 level. 0010: Encoder mode 2, the counter increments/decays the count along the edge of TI2FP2 according to the TI1FP1 level. 0011: Encoder mode 3, the counter counts at the edge of TI1FP1 and TI2FP2, the direction of the count depends on the level of the other input. 0100: Reset mode reinitializes the counter and generates a register update event when the selected trigger input (TRGI) rise edge appears. 0101: Gated mode, trigger input (TRGI) enables counter clock for high power hours. As soon as the trigger input becomes low, the counter immediately stops counting (but no longer bits). The start and stop of the counter are controlled. 0110: Trigger mode, when the trigger signal TRGI appears rising edge to start the counter (but no longer bit). Only the startup of the counter is controlled.

		0111: External clock mode 1, the counter clock is provided by the rising edge of the selected trigger signal (TRGI). 1000: Combined reset + trigger mode, reinitializes the counter when the selected trigger input (TRGI) rise edge appears, generates a register update event and starts the counter. Codes of more than 1000: reserved. Note: If TI1F_ED is selected as the trigger input (TS=00100), gating mode cannot be used. In fact, each time TI1F is converted, TI1F_ED outputs a pulse, and the gated mode checks the level of the trigger signal. Note: The clock of the slave peripheral (timer, ADC, etc.) that receives the TRGO or TRGO2 signal must be enabled before the event can be received from the master timer; And when the trigger signal is received from the master timer, the clock frequency (pre-divider) must not be changed in real time. Note: This bit is configured before the TIM6_EN register is configured

13.4.8. TIM6 DMA/ Interrupt Enable Register (TIM6_DIER)

Address offset: 0x0C

Reset value: 0x0000 0000

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

15	14	13	12	11	10	9	8
Res.	TDE	Res.	CC4DE	CC3DE	CC2DE	CC1DE	UDE
	RW		RW	RW	RW	RW	RW

7	6	5	4	3	2	1	0
Res.	TIE	Res.	CC4IE	CC3IE	CC2IE	CC1IE	UIE
	RW		RW	RW	RW	RW	RW

31:15	-	Reserved
14	TDE	Trigger DMA request enable: 0: disables DMA request triggering 1: Enables DMA requests to be triggered
13	-	Reserved
12	CC4DE	Capture/compare 4 DMA request enable: 0: disables CC4 DMA requests 1: Enables the CC4 DMA request
11	CC3DE	Capture/compare 3 DMA request enable: 0: disables CC3 DMA requests 1: Enables the CC3 DMA request
10	CC2DE	Capture/compare 2 DMA request enable:

		0: disables CC2 DMA requests 1: Enables the CC2 DMA request
9	CC1DE	Capture/compare 1 DMA request enable: 0: disables CC1 DMA requests 1: Enables the CC1 DMA request
8	UDE	Update DMA request enable: 0: disables updating DMA requests 1: Enables the DMA request to be updated
7	-	Reserved
6	TIE	Trigger interrupt enable: 0: disables interrupt triggering 1: Enables an interrupt to be triggered
5	-	Reserved
4	CC4IE	Capture/compare 4 Interrupt enable: 0: disables CC4 interruption 1: disables CC4 interrupt
3	CC3IE	Capture/compare 3 Interrupt Enable: 0: disables CC3 interrupt 1: disables CC3 interrupt
2	CC2IE	Capture/compare 2 Interrupt Enable: 0: disables CC2 interrupt 1: disables CC2 interrupt
1	CC1IE	Capture/compare 1 Interrupt Enable: 0: disables CC1 interruption 1: disables CC1 interruption
0	UIE	Update interrupt enable: 0: disables update interruption 1: The update interrupt is enabled

13.4.9. TIM6 Status Register (TIM6_SR)

Address offset: 0x10

Reset value: 0x0000 0000

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

15:13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	CC4OF	CC3OF	CC2OF	CC1OF	Res.		TIF	Res.	CC4IF	CC3IF	CC2IF	CC1IF	UIF
	RC_W0	RC_W0	RC_W0	RC_W0			RC_W0		RC_W0	RC_W0	RC_W0	RC_W0	RC_W0

31:19	-	Reserved

18	RDCNTF	TIM6_CNT register read flag bit: 1: The software is written 1, the counter count value latch is enabled, the latch counter count value 0: This bit will be cleared by hardware after the counter value latch is completed. This bit can also be cleared by software writing 0.
17:13	-	Reserved
12	CC4OF	Capture/compare 4 Repeat capture flag: See CC1OF instructions
11	CC3OF	Capture/Compare 3 Repeat capture flag: See CC1OF instructions
10	CC2OF	Capture/Compare 2 Repeat capture flag: See CC1OF instructions
9	CC1OF	Capture/compare 1 Repeat capture flag: This flag bit is set to 1 by the hardware only if the corresponding channel is configured in input capture mode. This bit can be cleared by writing "0" to the software. 0: No duplicate capture is detected. 1: The counter value is captured in the TIMx_CCR1 register and the CC1IF flag is set to 1.
8: 7	-	Reserved
6	TIF	Trigger interrupt flag: In all modes except gated mode, the flag is set to 1 by the hardware when a valid edge is detected on the TRGI input after the mode controller is enabled. When gating mode is selected, the flag is set to 1 when the counter is started or stopped. But it needs to be cleared by software. 0: No trigger event occurs. 1: triggers the interrupt suspension.
5	-	Reserved
4	CC4IF	Capture/compare 4 Interrupt flags: See the CC1IF description
3	CC3IF	Capture/compare 3 Interrupt flags: See the CC1IF description
2	CC2IF	Capture/compare 2 Interrupt flags: See the CC1IF description
1	CC1IF	Capture/compare 1 Interrupt flag: **If channel CC1 is configured to output:** This flag is set to 1 by hardware when the counter matches the comparison value, except in center alignment mode (see CMS bit description in the TIM6_CR1 register). But it needs to be cleared by software. 0: does not match. 1: The value of the TIM6_CNT counter matches that of the TIM6_CCR1 register. When the value of TIM6_CCR1 is greater than the value of TIM6_ARR, the

		CC1IF bit becomes high when the counter overflows (in increasing and decreasing count modes) or underflows (in decreasing count modes). **If channel CC1 is configured as input:** this bit will be set to 1 by the hardware when a capture event occurs. Clear this bit to zero by software or reading the TIM6_CCR1 register. 0: No input capture event occurred 1: The counter value has been captured in the TIM6_CCR1 register (an edge matching the selected polarity has been detected on IC1)
0	UIF	Update interrupt flag: This bit is set to 1 by hardware when an update event occurs. But it needs to be cleared by software. 0: No update has occurred. 1: The update is suspended. This bit is set to 1 by the hardware when updating the register in the following cases: - UDIS=0 in the TIM6_CR1 register, and repeat when the counter value overflows or underflows (update when repeat counter =0). - URS = 0 and UDIS = 0 in the TIM6_CR1 register, and CNT is re-initialized by software using UG bits in the TIM6_EGR register. - URS=0 and UDIS=0 in the TIM6_CR1 register, and CNT is reinitialized by the trigger event.

13.4.10. TIM6 Event generation register (TIM6_EGR)

Address offset: 0x14

Reset value: 0x0000 0000

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

15:7	6	5	4	3	2	1	0
Res.	TG	Res.	CC4G	CC3G	CC2G	CC1G	UG
	RW		RW	RW	RW	RW	RW

31:7	-	Reserved
6	TG	Trigger generation: This bit is set to 1 by the software to generate events and automatically cleared by the hardware. 0: no operation is performed 1: The TIF flag in the TIM6_SR register is set to 1. After this function is enabled, related interrupts or DMA transfer events can occur
5	-	Reserved
4	CC4G	Capture/compare 4 Generated: See the CC1G instructions

3	CC3G	Capture/compare 3 Generated: See the CC1G instructions
2	CC2G	Capture/compare 2 Generated: See the CC1G instructions
1	CC1G	Capture/compare 1 Generated: This bit is set to 1 by the software to generate events and automatically cleared by the hardware. 0: no operation is performed 1: Generate capture/compare events on channel 1: **If channel CC1 is configured as output:** When enabled, the CC1IF flag is set to 1 and the corresponding interrupt or DMA request is sent. **If channel CC1 is configured as input:** The TIM6_CCR1 register will capture the current value of the counter. When enabled, the CC1IF flag is set to 1 and the corresponding interrupt or DMA request is sent. If the CC1IF flag is already high, the CC1OF flag is set to 1.
0	UG	Update generation: This bit can be set to 1 by software and automatically cleared by hardware. 0: no operation is performed. 1: Reinitializes the counter and generates a register update event. Note that the pre-divider counter will also be cleared to zero (but the pre-divider ratio will not be affected). If you select center alignment mode or DIR=0 (incrementing count), the counter clears to zero; If DIR=1 (decrement count), the counter uses an automatically overloaded value (TIM6_ARR).

13.4.11. TIM6 Capture/Compare mode register1[multiplexing](TIM6_CCMR1)

Address offset: 0x18

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
Res.							OC2M[3]	Res.							OC1M[3]
							RW								RW

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1:0
IC2F[3:0]				IC2PSC[1:0]		CC2S[1:0]		IC1F[3:0]				IC1PSC[1:0]		CC1S[1:0]
OC2CE	OC2M[2:0]			OC2PE	OC2FE	CC2S[1:0]		OC1CE	OC1M[2:0]			OC1PE	OC1FE	CC1S[1:0]
RW				RW		RW		RW				RW		RW

Input capture mode:

31:16	-	Reserved
15:12	IC2F[3:0]	Input capture 2 filter: See IC1F[3:0] description

11:10	IC2PSC[1:0]	Input capture 2 pre-divider: See the description of OC1PSC[1:0]
9:8	CC2S[1:0]	Capture/Compare 2 Select: This bit field defines the direction of the channel (input/output) and the input used. 00: The CC2 channel is configured as output 01: The CC2 channel is configured as input, and IC2 is mapped to TI2 10: The CC2 channel is configured as input, and IC2 is mapped to TI1 11: The CC2 channel is configured as the input and IC2 is mapped to the TRC. This mode is only valid if the internal trigger input is selected via the TS bit (TIM6_SMCR register) Note: Data can be written to the CC2S bit only when the channel is closed (CC2E = "0" in TIM6_CCER). Note: This bit is configured before the TIM6_EN register is configured
7:4	IC1F[3:0]	Input capture 1 filter: This bit field defines the sampling frequency of the TI1 signal and the bandwidth of the digital filter applicable to TI1. The digital filter consists of an event counter, where each N consecutive events is considered a valid output edge: 0000: No filter, sampling at f_{DTS} frequency 0001: $f_{SAMPLING}=f_{TIM6\_CLK}$, N=2 0010: $f_{SAMPLING}=f_{TIM6\_CLK}$, N=4 0011: $f_{SAMPLING}=f_{TIM6\_CLK}$, N=8 0100: $f_{SAMPLING}=f_{DTS}/2$, N=6 0101: $f_{SAMPLING}=f_{DTS}/2$, N=8 0110: $f_{SAMPLING}=f_{DTS}/4$, N=6 0111: $f_{SAMPLING}=f_{DTS}/4$, N=8 1000: $f_{SAMPLING}=f_{DTS}/8$, N=6 1001: $f_{SAMPLING}=f_{DTS}/8$, N=8 1010: $f_{SAMPLING}=f_{DTS}/16$, N=5 1011: $f_{SAMPLING}=f_{DTS}/16$, N=6 1100: $f_{SAMPLING}=f_{DTS}/16$, N=8 1101: $f_{SAMPLING}=f_{DTS}/32$, N=5 1110: $f_{SAMPLING}=f_{DTS}/32$, N=6 1111: $f_{SAMPLING}=f_{DTS}/32$, N=8 Note: This bit is configured before the TIM6_EN register is configured
3:2	IC1PSC[1:0]	Input capture 1 predivider: This bit field defines the pre-division ratio of the CC1 input (IC1). As long as CC1E= "0" (TIMx_CCER register), the pre-divider is reset immediately. 00: No pre-divider, capture is performed on every edge detected on the capture input 01: A capture is performed for every 2 events 10: A capture is performed every 4 events 11: A capture is performed for every 8 events Note: This bit is configured before the TIM6_EN register is configured

1:0	CC1S[1:0]	Capture/Compare 1 Select: This bit field defines the direction of the channel (input/output) and the input used. 00: The CC1 channel is configured as output 01: The CC1 channel is configured as input, and IC1 is mapped to TI1 10: The CC1 channel is configured as input, and IC1 is mapped to TI2 11: The CC1 channel is configured as the input, and IC1 is mapped to the TRC. This mode is only valid if the internal trigger input is selected via the TS bit (TIM6_SMCR register) Note: Data can be written to the CC1S bit only when the channel is closed (CC1E = "0" in TIM6_CCER). Note: This bit is configured before the TIM6_EN register is configured

Output comparison mode:

31:25	-	Reserved
24	OC2M[3]	Output comparison 2 mode: This bit is the bit[3] of OC2M[3:0], note see OC2M[3:0]
23:17	-	Reserved
16	OC1M[3]	Output comparison 1 mode: This bit is the bit[3] of OC1M[3:0], see OC1M[3:0] for comments
15	OC2CE	Output comparison 2 Clear enable: See the OC1CE description.
14:12	OC2M[2:0]	Output compare 2 mode The bit of OC2M [3] is located at the bit of TIMx_CCMR1(output comparison) [24] See the description of OC1M[3:0].
11	OC2PE	Output comparison 2 Preload enable: For details, see OC1PE Description.
10	OC2FE	Output Comparison 2 Quick enable: See the OC1FE description.
9:8	CC2S[1:0]	Capture/Compare 2 Select: This bit field defines the direction of the channel (input/output) and the input used. 00: The CC2 channel is configured as output 01: The CC2 channel is configured as input, and IC2 is mapped to TI2 10: The CC2 channel is configured as input, and IC2 is mapped to TI1 11: The CC2 channel is configured as the input and IC2 is mapped to the TRC. This mode is only valid if the internal trigger input is selected via the TS bit (TIM6_SMCR register) Note: Data can be written to the CC2S bit only when the channel is closed (CC2E= "0" in TIM6_CCER).
7	OC1CE	Output comparison reset enable: 0: OC1REF is not affected by the ocref_clr_int signal 1: When high voltage is detected on the ocref_clr_int signal (ETRF input), the OC1REF clears immediately
6:4	OC1M[2:0]	Output comparison 1 mode:

		The bit of OC1M [3] is located at the bit of TIM6_CCMR1(output comparison) [16] These bits define the behavior of OC1REF, the output reference signal for OC1 and OC1N. OC1REF is active at a high level, while the active levels of OC1 and OC1N depend on the CC1P bit and CC1NP bit. 0000: Freeze, comparing the output comparison register TIM6_CCR with the counter TIM6_CNT does not affect the output. (This pattern is used to generate the time base) 0001: Set channel 1 to the output active level when matched. When the counter TIM6_CNT matches the capture/comparison register 1(TIM6_CCR1), the OC1REF signal is forced to a high level. 0010: Set channel 1 to output invalid level when matched. When the counter TIM6_CNT matches the capture/comparison register 1(TIM6_CCR1), the OC1REF signal is forced to a low level. 0011: The OC1REF is flipped when TIM6_CNT=TIM6_CCR1. 0100: forcibly changes to invalid level, OC1REF forcibly changes to low level. 0101: forcibly changes to active level, OC1REF forcibly changes to high level. 0110: PWM mode 1, in incremental counting mode, as long as TIM6_CNT<TIM6_CCR1, channel 1 is valid state, otherwise it is invalid state. In decrement mode, as long as TIM6_CNT>TIM6_CCR1, channel 1 is invalid (OC1REF= "0"), otherwise it is valid (OC1REF= "1"). 0111: PWM mode 2, in incremental counting mode, as long as TIM6_CNT<TIM6_CCR1, channel 1 is invalid state, otherwise it is valid state. In decrement mode, as long as TIM6_CNT>TIM6_CCR1, channel 1 is valid, otherwise it is invalid. 1000: Can trigger OPM mode 1 again, in incremental count mode, the channel is in a valid state until a trigger event is detected (on the TRGI signal). The comparison is then made in PWM mode 1, and the channel becomes active again at the next update. In decrement count mode, the channel is in an invalid state until a trigger event is detected (on the TRGI signal). The comparison is then made in PWM mode 1, and the channel becomes invalid again at the next update. 1001: OPM mode 2 can be triggered again, in incremental count mode, the channel is invalid until a trigger event is detected (on the TRGI signal). Then, when compared in PWM mode 2, the channel becomes invalid again at the next update. In decreasing count mode, the channel is in a valid state until a trigger event is detected (on the TRGI signal). The comparison is then made in PWM mode 1, and the channel becomes active again at the next update. 1100: Combined PWM mode 1, OC1REF behaves the same as in PWM mode 1. OC1REFC is the logical or operational result of OC1REF and OC2REF. 1101: Combined PWM mode 2, OC1REF behaves the same as in PWM mode 2. OC1REFC is the logical and operational result of OC1REF and OC2REF. 1110: Asymmetric PWM mode 1, OC1REF behaves the same as in PWM mode 1. When the counter is incremented, OC1REFC outputs OC1REF; When the counter decays the count, OC1REFC outputs OC2REF.

		1111: Asymmetric PWM mode 2, OC1REF behaves the same as in PWM mode 2. When the counter is incremented, OC1REFC outputs OC1REF; When the counter decays the count, OC1REFC outputs OC2REF. Other: Reserved Note: In PWM mode, only when the comparison result changes or the output comparison mode is switched by "freeze" mode to"PWM"mode, OCREF level will change.
3	OC1PE	Output comparison 1 Preloaded Enable: 0: disables the pre-loaded register associated with TIM6_CCR1. Data can be written to TIM6_CCR1 at any time, and the new value will be used immediately after writing. 1: Enables the pre-loaded register associated with TIM6_CCR1. Read/write access to the preloaded register. The TIM6_CCR1 preloaded value is loaded into the current register each time an update event is generated. Note: 1, as long as the LOCK (LOCK bit in TIM6_BDTR register) level 3 is programmed and CC1S= "00" (channel configuration as output), these bits cannot be modified. 2, only in monopulse mode can PWM mode be used without verifying the pre-loaded register (OPM position 1 in the TIM6_CR1 register). In other cases, the behavior is not guaranteed.
2	OC1FE	Output Comparison 1 Quick enable: This bit is used to speed up the impact of triggered input events on CC output. 0: CC1 will work normally according to the counter and CCR1 value even if triggered to turn on. 1: The occurrence of an effective edge on the trigger input is equivalent to a comparative match on the CC1 output. Subsequently, OC is set to the comparison level regardless of the comparison result. The OC1FE works only when the channel is configured in PWM1 or PWM2 mode.
1:0	CC1S[1:0]	Capture/Compare 1 Select: This bit field defines the direction of the channel (input/output) and the input used. 00: The CC1 channel is configured as output 01: The CC1 channel is configured as input, and IC1 is mapped to TI1 10: The CC1 channel is configured as input, and IC1 is mapped to TI2 11: The CC1 channel is configured as the input, and IC1 is mapped to the TRC. This mode is only valid if the internal trigger input is selected via the TS bit (TIM6_SMCR register) Note: Data can be written to the CC1S bit only when the channel is closed (CC1E = "0" in TIM6_CCER).

13.4.12. TIM6 Capture/Compare moderegister2[multiplexing] (TIM6_CCMR2)

Address offset: 0x1C

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
Res.							OC4M[3]	Res.							OC3M[3]
							RW								RW

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IC4F[3:0]				IC4PSC[1:0]		CC4S		IC3F[3:0]				IC3PSC[1:0]		CC3S[1:0]	
OC4CE	OC4M[2:0]			OC4PE	OC4FE	CC4S		OC3CE	OC3M[2:0]			OC3PE	OC3FE	CC3S[1:0]	
RW				RW		RW		RW				RW		RW	

Enter the capture mode:

31:16	-	Reserved
15:12	IC4F[3:0]	Input capture 4 filter: See IC1F[3:0] description
11:10	IC4PSC[1:0]	Input capture 4 Predivider: See IC1PSC[1:0] for details
9:8	CC4S	Capture/Compare 4 Select: This bit field defines the direction of the channel (input/output) and the input used. 00: The CC4 channel is configured as output 01: The CC4 channel is configured as input, and IC4 is mapped to TI4 10: The CC4 channel is configured as input, and IC4 is mapped to TI3 11: The CC4 channel is configured as input, and IC4 is mapped to the TRC. This mode is only valid if the internal trigger input is selected via the TS bit (TIM6_SMCR register) Note: Data can be written to the CC4S bit only when the channel is closed (CC4E = "0" in TIM6_CCER). Note: This bit is configured before the TIM6_EN register is configured
7:4	IC3F[3:0]	Input capture 3 filters: See IC1F[3:0] description.
3:2	IC3PSC[1:0]	Input capture 3 pre-divider: See IC1PSC[1:0] for details.
1:0	CC3S[1:0]	Capture/compare 3 Select: This bit field defines the direction of the channel (input/output) and the input used. 00: The CC3 channel is configured as output 01: The CC3 channel is configured as input, and IC3 is mapped to TI3 10: The CC3 channel is configured as input, and IC3 is mapped to TI4 11: The CC3 channel is configured as the input and IC3 is mapped to the TRC. This mode is only valid if the internal trigger input is selected via the TS bit (TIM6_SMCR register) Note: Data can be written to the CC3S bit only when the channel is closed (CC3E = "0" in TIM6_CCER).

		Note: This bit is configured before the TIM6_EN register is configured

Output comparison mode:

31:25	-	Reserved
24	OC4M	Output comparison 4 mode: This bit is the bit[3] of OC4M[3:0], note see OC4M[3:0]
23:17	-	Reserved
16	OC3M	Output comparison 3 mode This bit is the bit[3] of OC3M[3:0]. See OC3M[3:0] for comments.
15	OC4CE	Output comparison 4 Clear enable: See the OC1CE description.
14:12	OC4M[2:0]	Output comparison 4 mode: The bit of the OC4M [3] is located at the bit of TIMx_CCMR2(output comparison) [24] See the OC4M description.
11	OC4PE	Output comparison 4 Preloaded Enable: For details, see OC1PE Description.
10	OC4FE	Output Comparison 4 Quick enable: See the OC1FE description.
9:8	CC4S	Capture/Compare 4 Select: This bit field defines the direction of the channel (input/output) and the input used. 00: The CC4 channel is configured as output 01: The CC4 channel is configured as input, and IC4 is mapped to TI4 10: The CC4 channel is configured as input, and IC4 is mapped to TI3 11: The CC4 channel is configured as input, and IC4 is mapped to the TRC. This mode is only valid if the internal trigger input is selected via the TS bit (TIM6_SMCR register) Note: Data can be written to the CC4S bit only when the channel is closed (CC4E= "0" in TIM6_CCER).
7	OC3CE	Output comparison 3 Clear Zero enable: See the OC1CE description.
6:4	OC3M[2:0]	Output comparison 3 mode: The OC3M bit[3] is located in the bit of TIMx_CCMR2(output comparison) [16] See OC1M Description.
3	OC3PE	Output comparison 3 Preload Enable: For details, see OC1PE Description
2	OC3FE	Output Comparison 3 Quick Enable: See the OC1FE description.
1:0	CC3S[1:0]	Capture/compare 3 Select: This bit field defines the direction of the channel (input/output) and the input used.

		00: The CC3 channel is configured as output 01: The CC3 channel is configured as input, and IC3 is mapped to TI3 10: The CC3 channel is configured as input, and IC3 is mapped to TI4 11: The CC3 channel is configured as the input and IC3 is mapped to the TRC. This mode is only valid if the internal trigger input is selected via the TS bit (TIM6_SMCR register) Note: Data can be written to the CC3S bit only when the channel is closed (CC3E = "0" in TIM6_CCER).

13.4.13. TIM6 Capture/Compare Enable Register (TIM6_CCER)

Address offset: 0x20

Reset value: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CC4NP	Res.	CC4P	CC4E	CC3NP	Res.	CC3P	CC3E	CC2NP	Res.	CC2P	CC2E	CC1NP	Res.	CC1P	CC1E
RW		RW	RW	RW		RW	RW	RW		RW	RW	RW		RW	RW

31:16	-	Reserved
15	CC4NP	Capture/compare 4 Complementary output polarity: See CC1NP instructions
14	-	Reserved
13	CC4P	Capture/compare 4 Output polarity: See CC1P instructions
12	CC4E	Capture/compare 4 Output Enable: See CC1E instructions
11	CC3NP	Capture/compare 3 Complementary output polarity: See CC1NP instructions
10	-	Reserved
9	CC3P	Capture/compare 3 Output polarity: See CC1P instructions
8	CC3E	Capture/compare 3 Output Enable: See CC1E instructions
7	CC2NP	Capture/compare 2 Complementary output polarity: See CC1NP instructions
6	-	Reserved
5	CC2P	Capture/compare 2 Output polarity: See CC1P instructions
4	CC2E	Capture/compare 2 Output Enable: See CC1E instructions

3	CC1NP	Capture/compare 1 Complementary output polarity: **The CC1 channel is configured as output:** 0: indicates that the OC1N high level is valid 1: indicates that the OC1N low level is valid **The CC1 channel is configured as input:** This bit is used in conjunction with CC1P to define the polarity of TI1FP1 and TI2FP1. See CC1P instructions
2	-	Reserved
1	CC1P	Capture/compare 1 Output polarity: **The CC1 channel is configured as output:** 0: indicates that the OC1 level is high 1: indicates that the OC1 level is low **The CC1 channel is configured as an input:** the CC1NP/CC1P bit selects the effective polarity of TI1FP1 and TI2FP1 for trigger or capture operations. 00: Non-inverting/rising edge trigger. The circuit acts on the rising edge of TIxFP1 (performs capture or trigger operations in reset mode, external clock mode, or trigger mode), and TIxFP1 is not inverted (performs trigger operations in gated mode or encoder mode). 01: Reverse phase/falling edge trigger. The circuit acts on the falling edge of TIxFP1 (performs capture or trigger operations in reset mode, external clock mode, or trigger mode), and TIxFP1 inverts (performs trigger operations in gated mode or encoder mode). 10: Reserved. This configuration is not used. 11: Non-inverting/both rising and falling edges are triggered. The circuit acts on the rising and falling edges ofthe TIxFP1 (performs capture or trigger operations in reset mode, external clock mode, or trigger mode), and the TIxFP1 is not inverted (performs touch in gated mode) Send operation). This configuration cannot be used in encoder mode.
0	CC1E	Capture/compare 1 Output Enable: **The CC1 channel is configured as output:** 0: indicates that OC1 is disabled. 1: On, the OC1 signal is output to the corresponding output pin. **The CC1 channel is configured as input:** This bit determines whether the counter value can actually be captured into the input capture/comparison register 1(TIM6_CCR1). 0: The capture is prohibited. 1: Enables capture.

13.4.14. TIM6 Counter (TIM6_CNT)

Address offset: 0x24

Reset value: 0x0000 0000

31	30:16
UIFCPY	Res.
R	

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

31	UIFCPY	UIF copy: This bit is a read-only copy of the UIF bit in the TIM6_ISR register. If the UIFREMAP bit in TIM6_CR1 is reset, bit 31 is retained and read as 0.
30:16	-	Reserved
15:0	CNT[15:0]	To count the value, do not write to the register during the counter count: When the counter has not started counting, the initial value of the counter can be modified by writing the TIM6_CNT register. Reading process: 1, TIM6_SR register bit[18] write 1, this time will latch the counter count value; 2, read TIM6_SR register, loop check bit[18], when bit[18] is 0, it means that the counter count value latch is completed; 3, read the TIM6_CNT register, and the value read at this time is the value that the current read process wants.

13.4.15. TIM6 Pre-divider (TIM6_PSC)

Address offset: 0x28

Reset value: 0x0000 0000

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

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

31:16	-	Reserved
15:0	PSC[15:0]	Pre-divider value: The counter clock frequency (CK_CNT) is equal to f_{CK_PSC}/(PSC[15:0]+1). PS contains the value to be loaded into the active predivider register each time an update event occurs, including when the counter is cleared by the UG bit in the TIM6_EGR register, or when the controller is triggered to clear when configured in Reset mode.

13.4.16. TIM6 Pre-divider (TIM6_PSC)

Address offset: 0x2C
Reset value: 0x0000 0000

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

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

31:16	-	Reserved
15:0	ARR[15:0]	Automatic overload values: ARR is the value to be loaded into the actual automatic reload register. The counter does not work when the automatic overload value is empty.

13.4.17. TIM6 Capture/compare register 1(TIM6_CCR1)

Address offset: 0x34
Reset value: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR1[15:0]															
RW															

31:16	-	Reserved
15:0	CCR1[15:0]	Capture/compare 1 values: **If channel CC1 is configured to output:** CCR1 is the value to be loaded into the valid capture/compare 1 register (the preloaded value). If the OC1PE bit in the TIM6_CCMR1 register is not used to preload the function, the value takes effect immediately. Otherwise it only takes effect when an update event occurs (copy to the actual active capture/compare register 1). The actual capture/compare register contains the value to be compared to the counter TIM6_CN and signaled on the OC1 output. **If channel CC1 is configured as an input:** CR1 captures the counter value for the previous input when the 1 event (IC1) occurred. The TIM6_CCR1 register can only be read and cannot be programmed.

13.4.18. TIM6 Capture/Compare register 2(TIM6_CCR2)

Address offset: 0x38

Reset value: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR2[15:0]															
RW															

31:16	-	Reserved
15:0	CCR2[15:0]	Capture/compare 2 values: **If the channel CC2 is configured to output:** CCR2 is the value to be loaded into the valid capture/compare 2 register (the preloaded value). If the OC2PE bit in the TIM6_CCMR1 register is not used to preload the function, the value takes effect immediately. Otherwise it only takes effect when an update event occurs (copy to the actual active capture/compare register 2). The effective capture/compare register contains the value to be compared with the counter TIM6_CNT and signaled on the OC2 output. **If channel CC2 is configured as an input:** CCR2 captures the counter value for the previous input when the 2 event (IC2) occurred. The TIM6_CCR2 register can only be read and cannot be programmed.

13.4.19. TIM6 Capture/Compare register 3(TIM6_CCR3)

Address offset: 0x3C

Reset value: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR3[15:0]															
RW															

31:16	-	Reserved
15:0	CCR3[15:0]	Capture/compare value 3: **If the channel CC3 is configured to output:** CCR3 is the value to be loaded into the valid capture/compare 3 register (the

		preloaded value). If the OC3PE bit in the TIM6_CCMR2 register is not used to preload the function, the value takes effect immediately. Otherwise it only takes effect when an update event occurs (copy to a valid capture/compare register 3). The effective capture/compare register contains the value to be compared with the counter TIM6_CNT and signaled on the OC3 output. **If channel CC3 is configured as an input:** CCR3 captures the counter value for the previous input when the 3 event (IC3) occurred. The TIM6_CCR3 register can only be read and cannot be programmed.

13.4.20. TIM6 Capture/Compare Register 4(TIM6_CCR4)

Address offset: 0x40

Reset value: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR4[15:0]															
RW															

31:16	-	Reserved
15:0	CCR4[15:0]	Capture/Compare 4: **If the channel CC4 is configured to output:** CCR4 is the value to be loaded into the valid capture/compare 4 register (the preloaded value). If the OC4PE bit in the TIM6_CCMR2 register is not used to preload the function, the value takes effect immediately. Otherwise it only takes effect when an update event occurs (copy to a valid capture/compare register 4). The effective capture/compare register contains the value to be compared with the counter TIM6_CNT and signaled on the OC4 output. **If channel CC4 is configured as an input:** CCR4 captures the counter value for the previous input when the 4 event (IC4) occurred. The TIM6_CCR4 register can only be read and cannot be programmed.

13.4.21. TIM6 DMA Control Register (TIM6_DCR)

Address offset: 0x48

Reset value: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.			DBL[4:0]					Res.			DBA[4:0]				
			RW								RW				

31:13	-	Reserved
12:8	DBL[4:0]	DMA continuous transfer length: This 5-bit vector defines the DMA transfer length (when a read or write access is made to the TIM6_DMAR address, the timer makes one successive transfer), that is, the number of transfers. It can be sent as half a word or byte (see example below). 00000:1 transfer 00001: Two transfers 00010:3 transfers ... 10001:18 transfers Example: Take the following transfer: DBL = 7 bytes and DBA = TIM6_CR1. - If DBL = 7 bytes and DBA = TIM6_CR1 indicates the address of the bytes to be transmitted, the address to be transmitted should be given by the following formula: (TIM6_CR1 address) +DBA+ (DMA index), where DMA index = DBL In this case, add 7 bytes to (TIM6_CR1 address) + DBA to get the source/destination address of the data to be copied. In this case, data is sent to seven registers starting at the following address: (TIM6_CR1 address) + DBA Depending on the configuration of the DMA data size, several things can happen: - If the DMA data size is configured by half word, 16 bits of data will be transferred to each of the seven registers. If the DMA data size is configured in bytes, data will also be sent to seven registers: the first register contains the first MSB byte, the second contains the first LSB byte, and so on. Therefore, when using the transfer timer, you must also specify the size of the data to be transferred by DMA.
7:5	-	Reserved
4:0	DBA[4:0]	DMA base address: The 5 bits define the base address of the DMA transfer (when read/write access

		is done via the TIM6_DMAR address). DBA is defined as the offset calculated from the TIM6_CR1 register address. Example: 00000: TIM6_CR1 00001: TIM6_CR2 00010: TIM6_SMCR ...

13.4.22. TIM6 Full Transmission Address (TIM6_DMAR)

Address offset: 0x4C

Reset value: 0x0000 0000

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

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

31:0	DMAB[31:0]	DMA continuous transfer register: Reading or writing to the DMAR register accesses the register at the following address: (TIM6_CR1 address) + (DBA+DMA index)x4 Where the TIM6_CR1 address is the address of control register 1, the DBA is the DMA base address configured in the TIM6_DCR register, the DMA index is automatically controlled by the DMA transfer, and its range is between 0 and DBL (DBL configured in the TIM6_DCR register).

13.4.23. TIM6 Input/Output Control register (TIM6_DIR)

Address offset: 0x68

Reset value: 0x0000 0000

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

15:4	3	2	1	0
Res.	TIM6_CH4_DIR	TIM6_CH3_DIR	TIM6_CH2_DIR	TIM6_CH1_DIR
	RW	RW	RW	RW

31:4	-	Reserved
3	TIM6_CH4_DIR	TIM6_CH4 direction control: 0: output 1: input

2	TIM6_CH3_DIR	TIM6_CH3 direction control: 0: output 1: input
1	TIM6_CH2_DIR	TIM6_CH2 direction control: 0: output 1: input
0	TIM6_CH1_DIR	TIM6_CH1 direction control: 0: output 1: input

13.4.24. TIM6 Enable register (TIM6_EN)

Address offset: 0x6C

Reset value: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.															TIM6_EN
															RW

31:1	-	Reserved
0	TIM6_EN	TIM6 module work enable: 1: The module works. 0: The module stops working

Chapter 14 Analog to digital conversion module(ADC)

14.1. ADC Overview

An analog-to-digital converter (ADC) converts a continuous analog signal into a discrete digital signal. ADC can perform analog-to-digital conversion on any channel selected by the software, and can convert a single channel or a series of channels. The conversion result of the ADC is stored in the ADC_RDATA register.

14.2. ADC functional features

- The resolution of the data conversion is optional: 12 bit /8 bit linear successive approximation ADC
- The digital module has 18+2 analog input channels (18 IO, 1 internal channel reference voltage VREF channel, 1 temperature channel TS)
- Single channel single conversion mode (hardware average conversion mode with multiple samples)
- Continuous multichannel DMA polling mode
- Sampling time and conversion speed can be configured
- Hardware event triggering (TIM's TRGO signal) and software register triggering are supported
- Support data overflow interrupt, data management after overflow (DMA mode)
- Include stop register
- Hardware average function
- Reference voltage: VCC/2V/4V

14.3. ADC Function Description

14.3.1. ADC structure block diagram

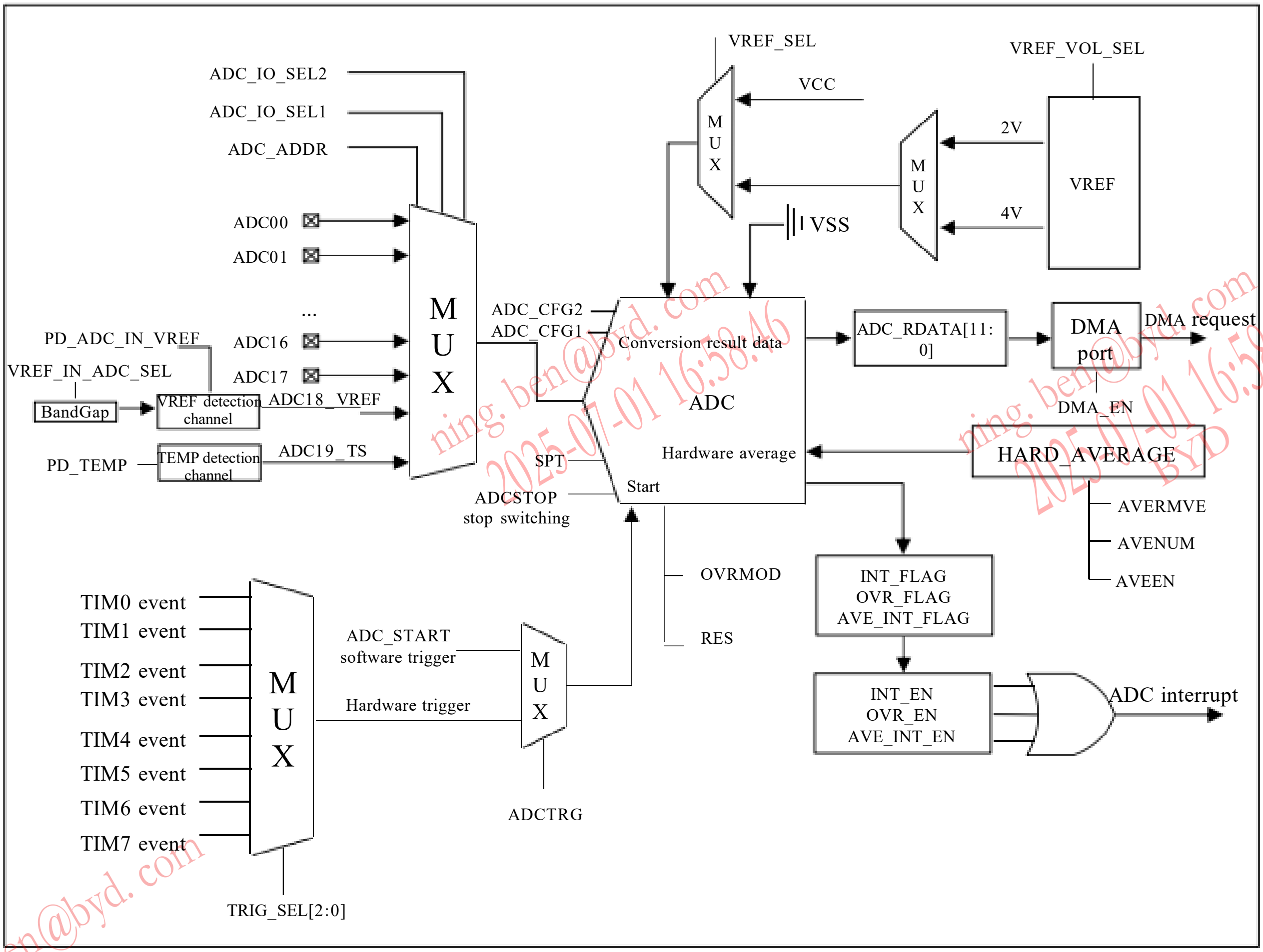


Figure 14.1 ADC structure block diagram

14.3.2. ADC clock frequency division

The ADC clock source is 48M and can be divided by the ADCK bit of the ADC_CFG1 register.

ADCK	ADC working clock(MHz)
0	24
1	16
2	12
3	6
4	4.8
5	4
6	2
7	1

Considering the low power consumption requirement, it is necessary to select the appropriate bias current under different ADCK

I_SEL_0[5:1]	Comparator bias current(μA)	ADC working clock(MHz)
00000	2	1、2、3、4.8、6
00001	4	1、2、3、4.8、6、12
00010	5	1、2、3、4.8、6、12
00011/00100	7	1、2、3、4.8、6、12、16
00101	9	1、2、3、4.8、6、12、16
00110	10	1、2、3、4.8、6、12、16
00111	12	1、2、3、4.8、6、12、16
11000	14	1、2、3、4.8、6、12、16、24
11001	16	1、2、3、4.8、6、12、16、24
11010	17	1、2、3、4.8、6、12、16、24
11011/11100	19	1、2、3、4.8、6、12、16、24
11101	21	1、2、3、4.8、6、12、16、24
11110	22	1、2、3、4.8、6、12、16、24
11111	24	1、2、3、4.8、6、12、16、24

I_SEL_ 1[7:6]	BUFFER bias current size(μA)	Minimum sampling time(ns)
00	2	500
01	3	400
10	4	300
11	5	300

14.3.3. ADC conversion time

The ADC clock source is 48M, and the converted clock can be selected by configuring register (ADC_CFG1)ADCK[2:0] bit.
There are 8 options:
ADCK: 24MHz, 16MHz, 12MHz, 6MHz, 4.8MHz, 3MHz, 2MHz, 1MHz
As shown in the table, ADC conversion time formula:

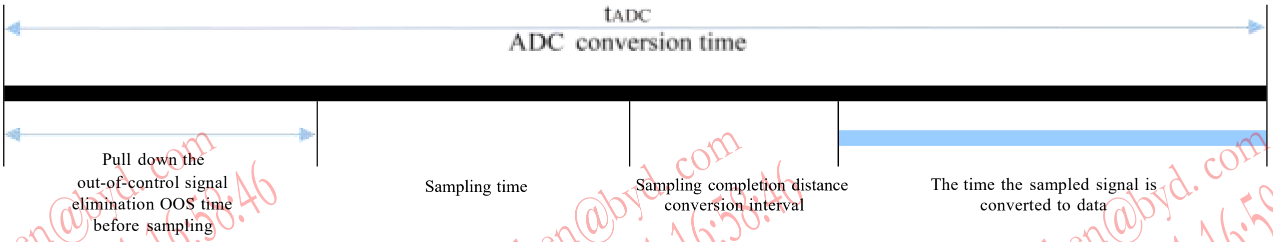

tADC
ADC conversion time
Pull down the out-of-control signal elimination OOS time before sampling
Sampling time
Sampling completion distance conversion interval
The time the sampled signal is converted to data

Figure 14.2 ADC conversion time diagram

Formula	Remark
$t_{ADC} = t_1 + t_2 + t_3 + t_4$	ADC conversion time
t_1 = ADCOOS* t_{ADCK}	Pull down the out-of-control signal elimination OOS time before sampling ADCOOS: Pull down the offset control signal selector register before sampling
t_2 = (ADC_SPT+1) * t_{ADCK}	Sampling time ADC_SPT: ADC sampling time configuration register (When ADC_SPT is configured with 0/1/2, t3 =3*tADCK)
t_3 = 3 * t_{ADCK}	Interval of sampling completion distance conversion

t_4 = (2*1 + SHIFT_CNT-DLY_CNT) * t_{ADCK}	The time the sampled signal is converted to data
	When RES = 0, SHIFT_CNT = 12;
	When RES = 1, SHIFT_CNT = 8;
	When ADCDLY = 0, DLY_CNT=1;
	When ADCDLY = 1, DLY_CNT=0;
	Note: SHIFT_CNT represents the number of resolution digits
	DLY_CNT Indicates the number of delay times

14.3.4. Channel selection

There are 20 input channels:

- 18 analog inputs from GPIO pins
- 2 internal analog inputs (temperature sensor TS, internal reference voltage VREF)

A single channel can be converted, or a series of channels can be converted:

- Single channel selection - Single/hardware average mode

The register (ADC_IO_SEL1/2) is used to enable or disable the ADC control function of the analog input pin, and each channel has a special selection bit. The ADC channel address selection register (ADC_ADDR) is used to control the address selection of the current scan channel.

For the 18-channel GPIO analog input, configure ADC_ADDR to enable the corresponding channel address after the channel is selected by ADC_IO_SEL1. For the two internal analog inputs (temperature sensor and internal reference voltage), you only need to set the channel address ADC_ADDR. ADC_IO_SEL2 is not required.

- Multi-channel selection (DMA mode multi-channel rotational scanning)

Each of the 20 channels is controlled by the ADC_IO_SEL1/2 register. Configure the ADC_IO_SEL1/2 register to determine the channel to be converted. Each ADC_IO_SEL1/2 bit enables one channel. The ADC_ADDR register changes with the scan channel, and a scan is completed quickly. If the address register is read during the scan process, it may be read incorrectly due to insufficient time. The channel scanning sequence is from the low to high level of the ADC_IO_SELx register. The channels selected by ADC_IO_SEL1 are scanned first, and then the channels selected by ADC_IO_SEL2 are scanned. A maximum of20 channels can be scanned at a time.

14.3.5. Trigger mode

ADC controllers can be triggered in two ways: software register trigger and hardware event trigger. Use ADCTRG to select the conversion trigger type. ADCTRG = 0 Select software register trigger, ADCTRG = 1 Select hardware event trigger. Select the hardware trigger source by configuring the ADC_TRIG_SEL register.

Trigger source	ADC_TRIG_SEL[3:0]
TIM0 event	000
TIM1 event	001
TIM2 event	010
TIM3 event	011
TIM4 event	100
TIM5 event	101
TIM6 event	110
TIM7 event	111

14.3.6. Start and stop

When selecting software register trigger, you must configure ADC_START=0 and then ADC_START=1 to enable conversion. When the hardware event trigger is selected, the hardware sets ADC_START to 1 to start the conversion after the hardware event is triggered. The software does not write ADC_START.

In single mode, the hardware is cleared as soon as the conversion is finished. In hardware averaging mode, the hardware is cleared when the average result of multiple channel conversions ends. In DMA continuous mode, the hardware clears ADC_START when all the configured channels are converted.

Setting the ADCSTOP bit to 1 by the software aborts any ongoing conversion and drops the conversion result (the ADC_RDATA register is not updated to the current conversion result), and the scan sequence is aborted and reset (restarting the ADC will restart the new sequence); This action resets the ADC, which is idle and ready for a new operation. When this process is complete, both the ADCSTOP bit and the ADC_START bit are zeroed out by the hardware, and the software must wait for ADC_START=0 before it can begin a new conversion.

When the software register is selected to trigger ADC startup, ADC_START =0 or ADCSTOP = 1 can be configured to stop ADC conversion. When hardware events are selected to trigger ADC startup, ADCSTOP = 1 is configured to stop ADC conversion. The ADC_START bit is invalid.

14.3.7. Temperature sensor and internal reference channel

The temperature sensor is used to measure the junction temperature of the chip itself. The TS channel inside the temperature sensor connected to the ADC is used to convert the sensor output voltage to a digital value.

Before measuring temperature, the ADC must be calibrated to support a temperature range of -40℃ to 105℃.

The output voltage of temperature sensor V_{TS} is calculated by formula (1) :

$$V_{TS} = V_{ref}/FS * ADC_{TS} \qquad (1)$$

V_{ref}: ADC reference voltage.

ADC_{TS}: ADC temperature value, obtained from the scan result register ADC_RDATA.

FS: indicates the maximum value of the ADC output. 12-bit resolution, FS 4095. 8-bit resolution, FS is 255.

The actual temperature is calculated by formula (2)

$$\text{Temperarute (℃)} = (V_{TS} - V_{27})/\text{Avg\_slope} + 27℃ \qquad (2)$$

Note: The linearity of the temperature sensor is ±10℃.

The voltage calibration value $V_{27℃}$(unit mV) corresponding to 27℃ is stored in the information block (NVR2) storage area. The NVR2 can read the calibration value only after it is unlocked. Users cannot change the calibration value. For details about how to unlock the NVR2, see "NVR2 Reading".

Address [0x203CC] = $V_{27℃}$ calibration value higher eight bits;

Address [0x203CD] = $V_{27℃}$ calibration value lower eight bits;

V_{27}: Voltage of the temperature sensor at 27℃.

Avg_slope: average temperature slope, unit mV/℃, reference temperature sensor characteristics.

The internal channel voltage ADC18_VREF can be selected by the VREF_IN_ADC_SEL bit of the ANA_CFG register, and the calibration value needs to be obtained by reading the corresponding address of NVR2 (for details, refer to thechip calibration value stored in the NVR2 (information block) storage area).

If the PD_TEMP bit is set to 0, the ADC temperature detection channel works properly. If the PD_ADC_IN_VREF bit is set to 0, the ADC VREF detection channel works properly.

14.3.8. Reference voltage

The BF7707AMXX series has three reference voltage sources: VCC/2V/4V

When selecting VCC as the ADC reference voltage:

When the power supply voltage fluctuates or drops, the VCC voltage value can be inversely calculated by formula(1) :

ADCINNER_Data/VREF_IN_ADC_SEL = 4096/VCC (1)

Note:

- ADCINNER_Data: ADC internal channel data;
- VREF_IN_ADC_SEL: needs to read the chip calibration value;

Vin voltage can be inversely calculated by formula (2) :

Vin_Data/Vin = 4096/VCC (2)

Note:

- Vin_Data: ADC input channel data;
- Vin: Input voltage;

The Vin voltage value can be obtained by formula (3) :

Vin = (Vin_Data/ADCINNER_Data) * VREF_IN_ADC_SEL (3)

VREF_IN_ADC_SEL needs to read the chip calibration value, obtain the internal channel data first, and then obtain the input voltage Vin_Data data, and the interval between two data acquisition is as short as possible.

Select ADC_VREF_VOL_SEL 2V/4V reference voltage (frequency must be ≤ 3MHz) :

The Vin voltage value can be reversed by formula (4).

Vin_Data/Vin=4096/ADC_VREF_VOL_SEL (4)

Note:

- Vin_Data: ADC input channel data;
- Vin: Input voltage (0~ ADC_VREF_VOL_SEL).
- ADC_VREF_VOL_SEL: need to read the chip calibration value.

Note: When the 4V reference voltage is selected, the VCC is greater than 4.5V.

Chip calibration values are stored in the NVR2 (information block) storage area. NVR2 needs to be unlocked to read the calibration value, and users cannot change it. For details about how to unlock the value, see the reference section of "NVR2 Reading" :

Address [0x203C0] : The ADC internal channel input voltage calibration value is eight bits higher;

Address [0x203C1] : The ADC internal channel input voltage calibration value is eight bits

lower;
Read the calibration value of chip information address ADC internal channel input voltage 1.3801V;

Address [0x203C2] : The ADC internal channel input voltage calibration value is eight bits higher;
Address [0x203C3] : The ADC internal channel input voltage calibration value is eight bits lower;
Read the calibration value of chip information address ADC internal channel input voltage 2.2832V;

Address [0x203C4] : The ADC internal channel input voltage calibration value is eight bits higher;
Address [0x203C5] : The ADC internal channel input voltage calibration value is eight bits lower;
Read the calibration value of the chip information address ADC internal channel input voltage 3.1517V;

Address [0x203C6] : The ADC internal channel input voltage calibration value is eight bits higher;
Address [0x203C7] : The ADC internal channel input voltage calibration value is eight bits lower;
Read the calibration value of chip information address ADC internal channel input voltage 4.135V;

Address [0x203C8] : The ADC internal channel input voltage calibration value is eight bits higher;
Address [0x203C9] : The ADC internal channel input voltage calibration value is eight bits lower;
Read the calibration value of chip information address ADC_VREF 2V;

Address [0x203CA] : The ADC internal channel input voltage calibration value is eight bits higher;
Address [0x203CB] : The ADC internal channel input voltage calibration value is eight bits lower;
Read the calibration value of chip information address ADC_VREF 4V;

14.3.9. Conversion mode

The ADC conversion resolution can be selected from 12 or 8 bits, depending on the RES bit. You can select three conversion modes by configuring the DMA_EN bit of the ADC_CFG3 register and the AVEEN bit of the ADC_AVE register.

Non-dma mode single conversion (ADC_CFG3.DMA_EN=0, AVEEN=0) :

After the ADC is triggered, the ADC scan is started. After one scan is finished, the hardware automatically shuts down the ADC scan. To start the next scan, the software/hardware needs to trigger again. When the selected channel data conversion and sampling is complete, a single conversion end interrupt is generated (if INT_EN is enabled), and the data is latched to the ADC_RDATA register. If ADC_START=0 (triggered by the software register) or ADCSTOP= 1 is set during scanning, the scanning stops immediately and related signals inside the module are reset.

Non-dma mode hardware averaging conversion (ADC_CFG3.DMA_EN=0, AVEEN=1) :

ADC starts scanning after it is triggered, and the module starts to convert the selected channel for N times according to registers AVEEN, AVENUM and AVERMV and averages out the results, resulting in output hardware average interrupt (if AVE_INT_EN is enabled). At the same time, the data is locked to the ADC_RDATA register. The hardware automatically turns off the ADC scan, and to start the next scan, the software/hardware needs to trigger again. If ADC_START=0 (triggered by the software register) or ADCSTOP= 1 is set during scanning, the scanning stops immediately and related signals inside the module are reset.

DMA mode continuous conversion (ADC_CFG3.DMA_EN=1, AVEEN=0) :

When selecting continuous conversion, configure ADC_CFG3.DMA_EN = 1, and configure ADC_IO_SEL1/2 to determine the channel to be converted. After the ADC is triggered, the scan begins, polling the register ADC_IO_SELx until the configured channel scan is complete. A DMA request is generated at the end of each channel conversion, and the request signal is pulled down until the DMA reads the ADC_RDATA register. If the data is not read in time by DMA, an overflow interrupt is generated (if OVR_EN is enabled) and the ADC can continue the conversion. When the received data length set by DMA is reached, the DMA transmission ends and a DMA interrupt is generated, which is generated in the DMA module.

If the DMA mode of ADC is disabled during conversion, the single conversion mode is enabled (ADC_CFG3.DMA_EN=0). If the IO port corresponding to the channel address ADC_ADDR is enabled as the ADC port, the channel stops after scanning and the internal related signals are reset. If the IO port corresponding to the channel address is not enabled as the ADC port, the scanning stops immediately. To rescan the configured channels, enable ADC_CFG3.DMA_EN and ADC_START again.

14.3.10. DMA overflow

In a continuous conversion in DMA mode, if the converted data is not read by DMA in time, the overflow flag indicates the data overflow event before the new conversion generates the data. The overflow interrupt can be generated by placing the OVR_EN position in the ADC_INT_CFG register at 1. If an overflow occurs, the ADC remains active and continues the conversion, unless software stops and resets the scan sequence by setting the ADCSTOP position 1 in the ADC_CFG3 register. The overflow flag can be cleared by writing a 1 to OVR_CLR by the software.

The OVRMOD bit in the ADC_CFG3 register can be configured to choose whether data is retained or overwritten in the event of an overflow event. When OVRMOD=0, the overflow event will retain the original data of the ADC_RDATA register and discard the new conversion result. If the overflow flag OVR_FLAG is kept as 1, the conversion can continue, but the resulting data will be discarded. When OVRMOD = 1, the ADC_RDATA register will be overwritten by the new conversion result, and the previously unread data will be lost. If the OVR_FLAG remains 1, the conversion can continue, and the ADC_RDATA register always contains the latest converted data.

14.3.11. Interrupt

There are three interrupt sources: single single channel conversion completed interrupt, DMA mode overflow interrupt, single channel hardware average mode conversion completed interrupt. And all three modes have interrupt enablement, as described in register ADC_INT_CFG.

14.3.12. Single channel hardware average

Single-channel hardware averaging means that the selected channel is sampled continuously for N times, and the obtained values are averaged each time, and the maximum and minimum values are removed and then averaged. The hardware average number of samples is determined by the register ADC_AVE.

AVEEN = 1

AVENUM[1:0]	Average frequency	AVERMV	Sampling frequency
00	4	0	4
		1	6
01	8	0	8
		1	10
1x	16	0	16
		1	18

14.4. ADC register

Base address: 0x5000 0000

Offset address	Register	Description
0x04	RCU_EN	Peripheral module clock control register
0x10	ANA_CFG	Analog module switch register
0x54	ADC_TRIG_SEL	The ADC hardware triggers the source selection register

Base address:0x5007 0000

Offset address	Register	Description
0x00	ADC_START	ADC scans the open register
0x04	ADC_ADDR	ADC channel address selection register
0x08	ADC_SPT	ADC sampling time configuration register
0x0C	ADC_CFG1	ADC configuration register 1
0x10	ADC_CFG2	ADC configuration register 2
0x14	ADC_RDATA	ADC scan result register
0x18	ADC_INT_CFG	ADC interrupt control register
0x1C	ADC_IO_SEL1	ADC function selection
0x20	ADC_IO_SEL2	ADC function selection
0x24	ADC_CFG3	ADC configuration register 3
0x28	ADC_AVE	ADC hardware average configuration register

14.4.1. Peripheral module clock control register (RCU_EN)

This register is the register that sets whether to allow or disallow the provision of clocks to peripheral modules

Address offset: 0x04

Reset value: 0x0000 0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.				DIV_ CLKEN	GPIOD_ CLKEN	GPIOC_ CLKEN	GPIOB_ CLKEN	GPIOA_ CLKEN	DMA_ CLKEN	CRC_ CLKEN	ADC_ CLKEN	WDT_ CLKEN	TIM7_ CLKEN	TIM6_ CLKEN	TIM5_ CLKEN
				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
TIM4_ CLKEN	TIM3_ CLKEN	TIM2_ CLKEN	TIM1_ CLKEN	TIM0_ CLKEN	PWM1_ CLKEN	PWM0_ CLKEN	IIC_ CLKEN	UART4_ CLKEN	UART3_ CLKEN	UART2_ CLKEN	UART1_ CLKEN	UART0_ CLKEN	SPI1_ CLKEN	SPI0_ CLKEN	Res.
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	

26	GPIOD_CLKEN	GPIOD module working enable: 1: Work 0: Disable. Default is 0
25	GPIOC_CLKEN	GPIOC module working enable:

		1: Work 0: Disable. Default is 0
24	GPIOB_CLKEN	GPIOB module working enable: 1: Work 0: Disable. Default is 0
23	GPIOA_CLKEN	GPIOA module working enable:: 1: Work 0: Disable. Default is 0
22	DMA_CLKEN	DMA(include DMA_SRAM/DMAMUX) module working enable: 1: Work 0: Disable. Default is 0
20	ADC_CLKEN	ADC module working enable: 1: Work 0: Disable. Default is 0
18	TIM7_CLKEN	TIM7 module working enable: 1: Work 0: Disable. Default is 0
17	TIM6_CLKEN	TIM6 module working enable: 1: Work 0: Disable. Default is 0
16	TIM5_CLKEN	TIM5 module working enable: 1: Work 0: Disable. Default is 0
15	TIM4_CLKEN	TIM4 module working enable: 1: Work 0: Disable. Default is 0
14	TIM3_CLKEN	TIM3 module working enable: 1: Work 0: Disable. Default is 0
13	TIM2_CLKEN	TIM2 module working enable: 1: Work 0: Disable. Default is 0
12	TIM1_CLKEN	TIM1 module working enable: 1: Work 0: Disable. Default is 0
11	TIM0_CLKEN	TIM0 module working enable: 1: Work 0: Disable. Default is 0

14.4.2. Analog module switch register (ANA_CFG)

Address offset: 0x10

Reset value: 0x0000 1850

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

15:13	12	11	10	9	8:7	6	5	4	3 2	1:0
Res.	PD_TEMP	PD_ADC	VREF_SEL	VREF_VOL_SEL	VREF_IN_ADC_SEL	PD_ADC_IN_VREF	XTAL_HFR_SEL	XTAL_SEL	XTAL_BYP_SEL	XTAL_EN_SEL
	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

31:13	-	Reserved
12	PD_TEMP	Analog ADC TEMP detects channel shutdown control register: 0: indicates that the module works properly 1: The module does not work
11	PD_ADC	Analog ADC off control register: 0: The ADC module works properly 1: The ADC module does not work
10	VREF_SEL	Select the source of the reference voltage: 0: Select VCC as the reference voltage 1: Select the output of the ADC_VREF module as the reference voltage
9	VREF_VOL_SEL	ADC_VREF output mode control signal, ADCK frequency must be ≤ 3MHz: 0:2V is used as the ADC reference voltage 1:4V as a reference voltage (VCC greater than 4.5V)
8:7	VREF_IN_ADC_SEL	Voltage input to ADC18 ADC18_VREF: 00:1.3801V 01:2.2832V 10:3.1517V 11:4.135V Default 00
6	PD_ADC_IN_VREF	Analog ADC VREF detect channel shutdown control register: 0: indicates that the module works properly 1: The module does not work

14.4.3. ADC hardware trigger source selection register (ADC_TRIG_SEL)

Address offset: 0x54

Reset value: 0x0000 0000

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

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

31:3	-	Reserved
2:0	TRIG_SEL	ADC hardware trigger source selection: 000: indicates the TIM0 event 001: TIM1 event 010: TIM2 event 011: TIM3 event 100: TIM4 event 101: TIM5 event 110: TIM6 event 111: TIM7 event

14.4.4. ADC Scan Open register (ADC_START)

Address offset: 0x00
Reset value: 0x0000 0000

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

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

31:1	-	Reserved
0	START	ADC scan open register: The value of START is set from 0 to 1, and the ADC starts scanning. After the scanning is complete, the START hardware automatically sets the value to 0

14.4.5. ADC channel address selection register (ADC_ADDR)

Address offset: 0x04
Reset value: 0x0000 0000

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

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

31:5	-	Reserved
4:0	ADDR	ADC channel address selection register: Single scan mode (non-DMA mode) :

		00000: Scan ADC00 00001: scan ADC01 00010: scans ADC02 00011: scans ADC03 00100: scans ADC04 00101: scans ADC05 ... 10000: scans ADC16 10001: scans ADC17 10010: ADC18_VREF input channel 10011: TS input channel for the ADC19_TS internal temperature sensor Other: Reserved In continuous mode (DMA mode), there is no need to write configuration, the address changes with the scan channel, and the address register may be read incorrectly due to insufficient time.

14.4.6. ADC Sampling Time Configuration Register (ADC_SPT)

Address offset: 0x08

Reset value: 0x0000 0000

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

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

31:10	-	Reserved
9:0	SPT	ADC Sampling time configuration register, sampling time: t_2 SPT<=2: $t_2 = 3* t_{ADCK}$ SPT>=3: $t_2 = (SPT+1)*t_{ADCK}$

14.4.7. ADC configuration register 1 (ADC_CFG1)

Address offset: 0x0C

Reset value: 0x0000 0000

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

15:13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	ADCOOS	ADCDLY	Res.							RES	ADCK		
	RW	RW								RW	RW		

31:12	-	Reserved
12	ADCOOS	ADC pre-sampling pull down offset control signal elimination OOS selection configuration:

		0: does not lower 1: Pull down 1(t_{ADCK}) Set this parameter to 1 if the first data sampled is unavailable; otherwise, set this parameter to 0
11	ADCDLY	Delay configuration after ADC converts the highest bit data in the timing: 0: no delay 1: Delay 1(t_{ADCK}) After the sample data is obtained after the conversion of the highest bit, whether to delay 1 t_{ADCK} to convert the sample data of the next bit
10:4	-	Reserved
3	RES	Data resolution: 0:12 bit resolution 1:8-bit resolution
2:0	ADCK	ADC module clock 48MHz, clock frequency division t_{ADCK}: 000:24M 001:16M 010:12M 011:6M 100:4.8M 101:3M 110:2M 111:1M When selecting the internal reference voltage ADC_VREF, the frequency must be less than or equal to 3MHz

14.4.8. ADC configuration Register 2 (ADC_CFG2)

Address offset: 0x10

Reset value: 0x0000 01CE

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

15:9	8	7	6	5	4	3	2	1	0
Res.	CTRL_SEL	I_SEL1		I_SEL0					Res.
	RW	RW		RW					

31:9	-	Reserved
8	CTRL_SEL	Comparator offset cancellation selection signal: 0: Sample the comparator before eliminating the offset 1: Sampling and comparator offset elimination are carried out at the same time, and offset elimination is completed together. The default value is 1
7:6	I_SEL_1	BUFFER bias current selection: 00:2 uA 01:3 uA 10:4 uA 11:5 uA The higher the bias current, the faster the comparator speed. Recommended 4uA For details about the configuration requirements, see ADC Clock Frequency Division

5:1	I_SEL_0	Comparator bias current: 00000:2uA 00001:4uA 00010:5uA 00011/00100:7uA 00101:9uA 00110:10uA 00111:12uA 11000:14uA 11001:16uA 11010:17uA 11011/11100:19uA 11101:21uA 11110:22uA 11111:24uA Other: Reserved The higher the bias current, the faster the comparator speed. 14uA recommended For details about the configuration requirements, see ADC Clock Frequency Division
0	-	Reserved

14.4.9. ADC Scan Result register (ADC_RDATA)

Address offset: 0x14

Reset value: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.				RDATA[11:0]											
				R											

31:12	-	Reserved
11:0	RDATA[11:0]	ADC scan result register

14.4.10. ADC interrupt configuration register (ADC_INT_CFG)

Address offset: 0x18

Reset value: 0x0000 0000

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

15:11	10	9	8	7	6	5	4	3	2	1	0
Res.	AVE_ INT_CLR	AVE_ INT_FLAG	AVE_ INT_EN	OVR_ CLR	OVR_ FLAG	OVR_ EN	TRIG_ CLR	TRIG_ FLAG	INT_ CLR	INT_ FLAG	INT_ EN
	W	R	RW	W	R	RW	W	R	W	R	RW

31:11	-	Reserved
10	AVE_INT_CLR	Single-channel hardware averaging when the round average ends

		interrupt status flag clear register, write 1 clear AVE_INT_FLAG, write only
9	AVE_INT_FLAG	Single-channel hardware averaging, a round average ends interrupt status marker register: 1: The interrupt is valid 0: The interrupt is invalid
8	AVE_INT_EN	Single channel hardware Mean interrupt status enable register: 1: indicates that the interrupt function is enabled 0: indicates that interrupt is disabled
7	OVR_CLR	To clear the ADC DMA mode overflow flag, write 1 to clear the OVR_FLAG
6	OVR_FLAG	ADC DMA Mode Overflow flag: 1: overflow 0: No overflow or overflow cleared 0
5	OVR_EN	ADC DMA Mode overflow interrupt enable: 1: Enable overflow interrupt 0: Disable overflow interrupt
4	TRIG_CLR	The event triggered flag is cleared Write 1 to clear zero
3	TRIG_FLAG	Event trigger flag bit: 1: indicates that a hardware event is triggered 0: indicates the software register trigger flag
2	INT_CLR	Single single-channel transition End interrupt status flag clear register: Write 1 to clear INT_FLAG
1	INT_FLAG	Single single-channel transition end interrupt status flag register: 1: The interrupt is valid 0: The interrupt is invalid In single-channel hardware averaging, this bit is not set
0	INT_EN	Single-channel conversion end interrupt enable register: 1: indicates that the interrupt function is enabled 0: interrupt disable (for polling mode)

14.4.11. ADC Select Enable Register 1(ADC_IO_SEL1)

Address offset: 0x1C

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.														ADC17	ADC16
														RW	RW

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADC15	ADC14	ADC13	ADC12	ADC11	ADC10	ADC09	ADC08	ADC07	ADC06	ADC05	ADC04	ADC03	ADC02	ADC01	ADC00

RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

31:18	-	Reserved
17:0	ADCxx	ADC features for each GPIO channel: 1: Select the ADC function 0: The ADC function is not selected Bit[0] to Bit[17] corresponds to ADC00 to ADC17

14.4.12. ADC Select Enable Register 2(ADC_IO_SEL2)

Address offset: 0x20

Reset value: 0x0000 0000

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

15:2	1	0
Res.	VIN_TS	VIN_VREF
	RW	RW

31:2	-	Reserved
1	VIN_TS	Select the TS input channel of the internal temperature sensor. This parameter is configured only in continuous scanning mode (DMA mode). This parameter is invalid in non-DMA mode and no value is required 1: Select the ADC function 0: The ADC function is not selected
0	VIN_VREF	Internal reference voltage VREF input channel selection, only in continuous scanning (DMA mode) configuration, non-DMA mode configuration is invalid, no need to assign value 1: Select the ADC function 0: The ADC function is not selected

14.4.13. ADC configuration Register 3(ADC_CFG3)

Address offset: 0x24

Reset value: 0x0000 0000

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

15:4	3	2	1	0
Res.	ADCSTOP	OVRMOD	ADCTRG	DMA_EN
	RW	RW	RW	RW

31:4	-	Reserved
3	ADCSTOP	ADC stop conversion command: 0: The stop conversion command is not executed 1: Write 1 to stop the ADC
2	OVRMOD	ADC Overflow management mode: 0: Overflow is detected and the ADC_RDATA register retains the original data 1: When overflow is detected, the ADC_RDATA register will be overwritten by the previous conversion data
1	ADCTRG	Conversion trigger selection: There are two triggers to choose from: software register trigger and hardware event trigger. 0: disables hardware event triggering. Select software register triggering 1: Select a hardware event (TIM event) to trigger
0	DMA_EN	ADC DMA mode enabled: 1: indicates that DMA is enabled 0: indicates that DMA is disabled

14.4.14. ADC Single Channel Hardware Average Configuration Register (ADC_AVE)

Address offset: 0x28

Reset value: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.												AVEEN	AVENUM		AVERMV
												RW	RW		RW

31:4	-	Reserved
3	AVEEN	Single-channel hardware average function enabled: 1: indicates that single-channel hardware is enabled on average 0: indicates that the average single-channel hardware is disabled
2:1	AVENUM	Single channel hardware average time sampling number selection: 00:4 samples 01:8 samples 1x: 16 samples
0	AVERMV	Average feature selection: 1: Remove the maximum and minimum values and then average them 0: Average without removing the maximum and minimum values

14.5. ADC Configuration Process

Single conversion mode configuration process:

1. Enable the peripheral clock.
2. Configure the GPIO simulation function. For details, see " Configuring GPIO Simulation Function " .
3. Set ADC_START and ADCTRG to zero.
4. Configure ADC_IO_SEL and ADC_ADDR to determine the channel to be converted.
5. Configure the ADC_CFG1/ADC_CFG2 register, various parameters of ADC, sampling time, delay time after high data, resolution, clock frequency division, reference voltage and other parameters.
6. Select the hardware trigger source (if hardware event trigger is selected) ADC_TRIG_SEL.
7. Clear the corresponding interrupt flag bit.
8. Enable ADC single conversion end interrupt Enable INT_EN.
9. Enable the ADC module.
10. If software register trigger is selected, set ADC_START = 1. If hardware event triggering is selected, set ADCTRG = 1 to start. The module waits for the hardware event triggering source signal to start scanning.
11. Get the interrupt flag in the interrupt service function and clear it, and read the data register ADC_RDATA to obtain the converted data when the interrupt is generated.

Hardware averaging configuration process:

1. Enable the peripheral clock.
2. Configure the GPIO simulation function. For details, see " Configuring GPIO Simulation Function " .
3. Set ADC_START and ADCTRG to zero.
4. Configure ADC_IO_SEL and ADC_ADDR to determine the channel to be converted.
5. Configure the ADC_CFG1/ADC_CFG2 register, and configure various ADC parameters, sampling time, delay time after the highest bit data, resolution, clock frequency division, reference voltage, etc.;
6. Select the hardware trigger source (if hardware event trigger is selected) ADC_TRIG_SEL.
7. Configure the ADC_AVE register to enable hardware averaging, select sampling times for hardware averaging, and determine whether to remove the maximum and minimum values and average again.
8. Clear the corresponding interrupt flag bit.
9. Enable ADC hardware average interrupt Enable AVE_INT_EN.
10. Enable the ADC module.
11. If software register trigger is selected, set ADC_START = 1. If hardware event triggering is selected, set ADCTRG = 1 to start. The module waits for the hardware event triggering source signal to start scanning.
12. Get the hardware average interrupt flag in the interrupt service function and clear it. After the interrupt is generated, read the data register ADC_RDATA to get the converted data.

DMA mode continuous conversion configuration process:

1. Enable the clock ofthe ADC and DMA modules.
2. Configure the GPIO simulation function. For details, see " Configuring GPIO Simulation Function " .
3. Set ADC_START and ADCTRG to zero.
4. Configure ADC_IO_SEL to determine the channel to be converted.
5. Configure the ADC_CFG1/ADC_CFG2 register, and configure various ADC parameters, sampling time, delay time after the highest bit data, resolution, clock frequency division, reference voltage, etc.
6. Select the hardware trigger source (if hardware event trigger is selected) ADC_TRIG_SEL.
7. Configure the OVRMOD bit of the ADC_CFG register and select whether to retain data or overwrite data when an overflow event occurs.
8. Configure the CTRL_BASE_PTR of the master data structure to determine the master data address (DMA_SRAM_BASE).
9. Configure CHNL_EN_SET and Rice vinegar 01CHNL_INT_EN To enable and disable DMA transfer channel x.
10. Configure DMAMUX_SEL to select the ADC request source mapped to DMA channel x.
11. Set CHNLx_SRC_END_PTR to the ADC_RDATA address (source address).
12. Configure CHNLx_DST_END_PTR as the SRAM address (destination address).
13. Configure CHNLx_CONTROL to determine the data size, data quantity, and transmission mode.
14. Clear the corresponding interrupt flag bit.
15. Configure ADC_INT_CFG to enable ADC overflow OVR_EN.
16. Configure DMA_EN for ADC_CFG3 to enable DMA for the ADC.
17. Enable the ADC module and DMA module.
18. Select software register trigger and configure ADC_START = 1; If hardware event triggering is selected, set ADCTRG = 1 to start. The module waits for the hardware event triggering source signal to start scanning.
19. After the transmission reaches the received data length set by DMA, the DMA transmission ends and a DMA interrupt is generated.
20. A DMA request will be generated after the end of each channel scan. If DMA fails to read the data in the RDATA register in time, ADC overflow interrupt will be generated;

    Depending on the configuration of the OVRMOD bits, the ADC overwrites the RDATA or drops the new data, and the conversion of the ADC continues unless the software sets the ADCSTOP to 1.

14.6. Note

1. **When selecting software register trigger, configure ADC_START=0 and then ADC_START=1 to enable conversion.**
2. The conversion ends. Clear hardware ADC_START. ADC_START cannot be configured during scanning.
3. To stop the conversion while working, write 1 to the ADCSTOP register. Do not modify the configuration in the process of work, modify the configuration only after the (single/hardware average /DMA mode) conversion is completed.
4. If the conversion between DMA mode and hardware average mode is complete, you need to switch to other modes. Enable AVE_EN (complete software programming, no hardware pull down logic) for DMA_EN/ hardware average of ADC_CFG3, and set the configuration value to 0.
5. The DMA mode and hardware average mode of ADC have no hardware circumvention logic, and it is forbidden to set the DMA_EN bit and AVE_EN bit to 1 at the same time.
6. The software needs to circumvent the output interrupt function by following the table (register ADC_INT_CFG).

Precautions for enabling three types of interrupt in different working modes

	INT_EN	AVE_INT_EN	OVR_EN
Single conversion mode	With a 1 or a 0	Must match 0	Must match 0
Hardware average mode	Must match 0	With a 1 or a 0	Must match 0
DMA mode	Must match 0	Must match 0	With a 1 or a 0

Chapter 15 Divider (DIV)

15.1. DIV Overview

The DIV module's main function is a 32-bit divider. The module includes signed division and unsigned division operations, and the dividend and divisor are 32 bits. The 16 clock cycle division operations are completed, producing the interrupt flag, the QUOTIENT is placed in the Quotient register, and the REMAINDER is placed in the Remainder register. If division by 0 is detected, the operation is terminated in advance, and the division by 0 flag bit DIV0_FLAG is set to 1.

Note: In signed division, the remainder is divided into a positive remainder and a negative remainder. If the dividend is positive, the remainder is positive, and if the dividend is negative, the remainder is negative.
Example: 100÷3 = 33... 1, 100 ÷(-3) = (-33)... 1, (-100)÷3=(-33)... (-1), (-100)÷(-3)=33... (-1)

15.2. DIV register

Base address: 0x5000 0000

Offset address	Register	Description
0x04	RCU_EN	Peripheral module clock control register

Base address: 0x5005 0300

Offset address	Register	Description
0x00	DIV_CFG	DIV configuration register
0x04	DIV_STATE	DIV interrupt status register
0x08	DIVIDEND	Dividend register
0x0C	DIVISOR	Divisor register
0x10	QUOTIENT	Quotient register
0x14	REMAINDER	Remainder register

15.2.1. Peripheral module clock control register (RCU_EN)

Address offset: 0x04

Reset value: 0x0000 0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.				DIV_ CLKEN	GPIOD_ CLKEN	GPIOC_ CLKEN	GPIOB_ CLKEN	GPIOA_ CLKEN	DMA_ CLKEN	CRC_ CLKEN	ADC_ CLKEN	WDT_ CLKEN	TIM7_ CLKEN	TIM6_ CLKEN	TIM5_ CLKEN
				RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

27	DIV_CLKEN	DIV module work enable: 1: Work 0: Disable. Default is 0

15.2.2. DIV Control Signal register (DIV_CFG)

Address offset: 0x00

Reset value: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.														SIGN	INT_EN
														RW	RW

31:2	-	Reserved
1	SIGN	Division unsigned select register: 0: signed division 1: unsigned division
0	INT_EN	DIV interrupt enable register: 0: off 1: on

15.2.3. DIV Interrupt status register (DIV_INT_STATE)

Address offset: 0x04

Reset value: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.														DIV0_FLAG	INT_STATE
														RC_W1	RC_W1

31:2	-	Reserved
1	DIV0_FLAG	Divide by 0 flag: 1: The operation ends prematurely when division by 0 is detected 0: Division by 0 is not detected Software write 1 clear 0
0	INT_STATE	Interrupt status bit: 1: The division operation is complete, and the interrupt flag is not cleared 0: indicates that there is no division operation or the division operation is not complete Write 1 to clear 0

15.2.4. DIV DIVIDEND register (DIVIDEND)

Address offset: 0x08

Reset value: 0x0000 0000

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

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

31:0	DIVIDEND	Store the dividend

15.2.5. DIV DIVISOR register

Address offset: 0x0C

Reset value: 0x0000 0000

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

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

31:0	DIVISOR	Memory divisor

15.2.6. DIV QUOTIENT register (QUOTIENT)

Address offset: 0x10

Reset value: 0x0000 0000

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

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

31:0	QUOTIENT	Depositor

15.2.7. DIV REMAINDER register

Address offset: 0x14

Reset value: 0x0000 0000

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

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

31:0	REMAINDER	Memory remainder

15.3. DIV configuration process

1. Enable the clock function for the DIV module.
2. Configure the DIV_CFG register to configure SIGN and INT_EN. SIGN indicates the symbol of the division operation. 0 indicates signed division and 1 indicates unsigned division. INT_EN indicates that interrupt is enabled, 0 indicates off, and 1 indicates on.
3. Configure the DIVIDEND register dividend.
4. Configure the DIVISOR register. After the configuration, the DIV module is automatically enabled.
5. Check the DIVISOR flag DIV0_FLAG in the interrupt function. If it is 1, it means that the divisor is detected, the operation is not carried out, and the operation is finished in advance. Check the interrupt status bit INT_STATE. 1 indicates that division is complete, 0 indicates that no division operation is performed or division operation is not complete, and 1 is written to the register to clear zero.

15.4. Note

1. Do not write any register during the division operation, otherwise unforeseen errors will occur.
2. Only when the interrupt flag is detected is the result of the subdivision. Otherwise, the last result is read.

Chapter 16 Serial peripheral interface (SPI)

16.1. SPI overview

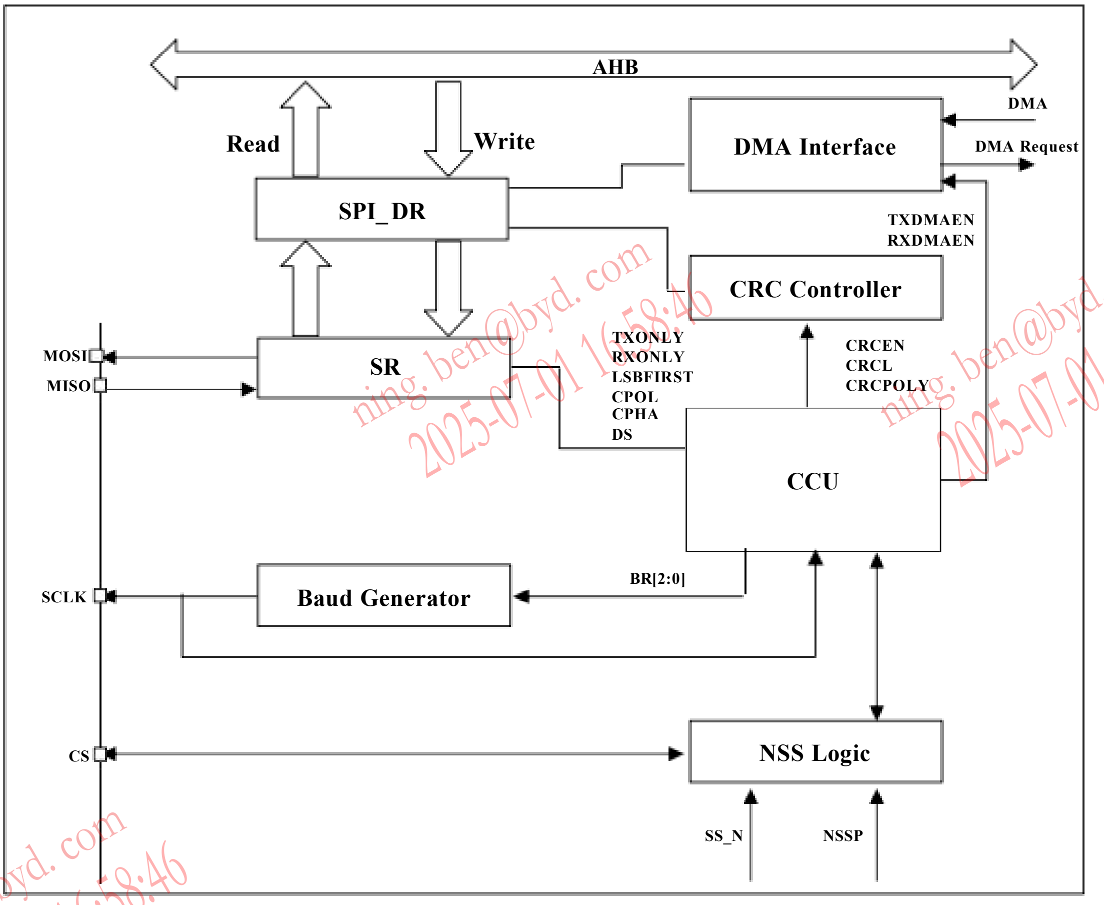
The SPI of the BF7707AMXX supports half/full duplex, synchronous, serial communication between the MCU and internal modules or external devices. BF7707AMXX contains two modules SPI0 and SPI1.

16.2. SPI functional characteristics

- Supports master or slave mode.
- Four communication formats are available.
- Support full duplex, half duplex receiving, half duplex sending mode.
- 4 to 16/32bit data size selection (32bit is a whole, not 8bit*4), data MSB/LSB transmission can be configured.
- The maximum communication frequency of SPI host is 8M, and the maximum communication frequency of SPI slave is 4M.
- Support 4/6/8/12/24/48/96/192 baud pre-division.
- High speed port enable control (frequency over 1M).
- The host chip selection signal is controlled by SFR or by hardware automatic control (NSSP pulse mode).
- It has four 32-bit buffers (two each for receiving/sending), the amount of which is Optional hardware CRC check, CRC check code length is optional 8/16 bits, programmable CRC polynomial, with CRC error flag and interrupt, the calculated CRC check code can be read, and CRC frames can be automatically sent after all data is sent.
- CRC and receive mode only, configure the amount of data to be transmitted.
- With send empty and receive full status flags, optional polling/interrupt /DMA mode to process data.
- BUSY flag bit.
- Error events such as Buffer overflow, CRC check errors have flag bits and error interrupts.

16.3. SPI functional description

16.3.1. SPI structure diagram



Figures 16.1 SPI structure diagram

SPI connects to external devices with 4 pins:

- MOSI: In master mode, data is sent, and in slave mode, data is received.
- MISO: In master mode, data is received, and in slave mode, data is sent.
- SCLK: Clock input/output pin, used to output clock from host or input clock from slave.
- CS: Select signal from the chip, output from the host, input from the slave.

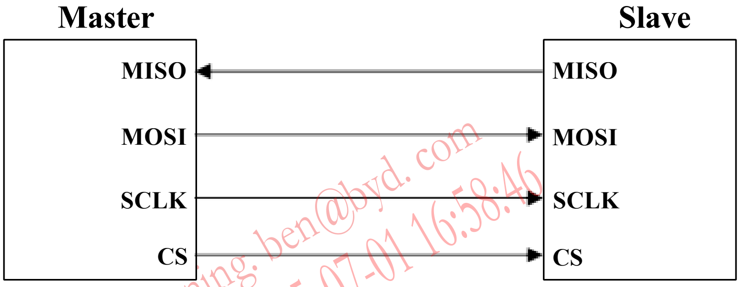
16.3.2. SPI communication format

The communication format is configured by the CPOL,CPHA,LSBFIRST bits ofthe SPI_CR1 register

- CPOL bit: Controls the idle clock level. CPOL = 0 indicates that the idle clock level is low, and CPOL = 1 indicates that the idle clock level is high.
- CPHA bit: Select the sampling clock edge. CPHA = 0 indicates that the first clock edge is sampled, and CPHA = 1 indicates that the second clock edge is sampled.
- LSBFIRST: Select the order in which data is sent (LSB or MSB).

16.3.3. SPI working mode

Set the TXONLY bit and RXONLY of the SPI_CR1 register to 0 at the same time, and configure the SPI to be in full duplex mode, sending and receiving data simultaneously. Set TXONLY position to 1 and RXONLY to 0, and SPI works in send only mode. Set TXONLY position to 0 and RXONLY to 1, and SPI works in receive-only mode. TXONLY and RXONLY cannot both be set to 1..



Figures 16.2 connection mode of master and salve

16.3.4. SPI master clock and CS

Clock:

- When the SPI is configured to host receive only mode, as long as the slice selection signal SS_N is configured to 0, the clock will be automatically generated and continuously output until the required amount of data is received (through the SPI_NUM register configuration), after which the SPI will stop generating clock, and the RXONLY enable will be automatically pulled down by the hardware. Amount of data must be configured in receiving-only mode (SPI_NUM register).
- When the SPI is configured as the host not only receiving mode, the SPI automatically detects whether the current data needs to be sent after the slice selection is lowered. If there is data, the clock is generated and the data is sent until the end of a frame of data transmission to determine whether there is still data to be sent. If there is data, the SPI continuously sends data, otherwise the clock stops until the bus next writes data.

When the data transmission conditions are met, the pre-divider starts to work and generates the clock for HCLK according to the baud rate configured by BR. When CRC is enabled, the hardware automatically transmits CRC frames after the last frame of data has been transferred.

CS NSS:

When the SPI is configured as a master, CS NSS is generated within the SPI and output to IO. There are two ways to control the master's CS NSS :

- SFR control: By configuring the SS_N bit control CS ofthe SPI_CR1 register, the NSSP bit must be kept clear in this mode.
- Hardware control (NSSP mode) : The NSS pulse mode can be enabled by setting the NSSP bit of the SPI_CR1 register. NSSP mode only works when CPHA=0. In this mode, CS is

controlled automatically by the hardware, which only keeps the low level during the data transmission, and keeps the high level at other times. In the process of continuous data transmission, the NSS signal will insert a high level of a clock cycle between the two data, which is equivalent to "disconnecting" the data pair with an NSS pulse. Note that CRC checksum receive only mode is not available when using this pulse mode.

16.3.5. SPI sending control

The SPI module contains two 32Bit TXBUFs for storing data written by the bus to be sent. When SPI is disabled, TXBUF is reset.

TXBUF_NUM[1:0] Indicates how much data is contained in the current TXBUF, which can be read out through the status register SPI_SR. When TXBUF_NUM =2, if the bus writes data, overflow occurs, the data is not written to the TXBUF, and the OVR flag is raised and an interrupt is generated (when the error interrupt enable is turned on). When TXBUF_NUM = 0, the SPI stops communicating after sending the current data frame until the bus writes the next data.

The TXE bit marks the data status ofthe TXBUF. If it is set to 1, it indicates that the TXBUF is not full and the bus can write data. You can choose the following three ways to process the sent data:

- Polling status register: Read the SPI_SR register and determine the status ofthe TXBUF through the TXE bit or TXBUF_NUM[1:0] bit. When the TXBUF is not full, data is written. This process is the fastest.
- TXE interrupt: Enable the TXEIE interrupt function. When the TXBUF capacity is not full, an SPI interrupt is generated. Read SPI_SR register in interrupt service function to judge TXBUF state and write data. If the system responds slowly to interrupts, you are advised to fill the TXBUF after the SPI is enabled and then lower the SS_N.
- TXDMA: Configure the DMA sending channel and data, and enable TXDMAEN. After both SPI and TXDMAEN are enabled, TXDMA requests are generated immediately. The DMA writes data to be sent successively through the bus. After a TXDMA transfer is complete, you must disable the TXDMAEN function and enable it the next time. Otherwise, TXDMA requests remain high.

When DMA mode is used for transmission, the HSPEED_EN register needs to be configured to enable the corresponding IO port to be in high-speed mode.

In slave transmission mode, if continuous communication is desired, the next data must be written to TXBUF before the current data frame is sent, otherwise the transmission will fail.

If CRC is enabled, SPI will perform CRC calculation on the data output at the TXD end according to the configured CRC length and CRC polynomial along each clock sampling line until the data volume configured in the SPI_NUM register is transmitted. After that, SPI will stop CRC calculation and automatically send the calculated CRC frame attached to the data frame. The size end format ofCRC frame is consistent with that of data frame. CRC frame length is determined by CRCL bit. If DS and CRCL are inconsistent, hardware can automatically correct DS to the corresponding value during CRC frame transmission.

When CRC is enabled, slice selection NSS must be kept low throughout the transmission, so the NSSP mode is incompatible with CRC. In CRC mode, after sending a batch of data (as configured by SPI_NUM register), the CRC function is cleared by the hardware. The CRC function needs to be re-enabled the next time. The calculated CRC check code will be stored in the SPI_TXCRCR register after transmission.

Note: In CRC mode, DS can only be configured as 8/16/32Bit multiples of 8, and CRC length can only be configured as 8 bit or 16Bit.

16.3.6. SPI receive control

The SPI samples the data input to the RXD port at each clock sampling edge and saves it. The SPI module contains two 32-bit RXBUFs for storing data received by the SPI. When SPI is disabled, RXBUF is reset.

RXBUF_NUM[1:0] Indicates how much data is contained in the current RXBUF, which can be read out through the status register. When RXBUF_NUM =2, if new data is received, overflow occurs, data is not stored in RXBUF (lost), and the OVR flag is raised and an interrupt is emitted (error interrupt enabled). When RXBUF_NUM =0, the bus reads the SPI_DR register and returns an invalid data (the data from the previous read), which does not affect RXBUF and its status bits. Therefore, the SPI_SR register must be read before reading the SPI_DR register to ensure that there is data in RXBUF, otherwise the read data is invalid.

The RXNE bit marks the data status of RXBUF. When set to 1, it indicates that there is unread data in RXBUF. The received data can be processed in the following three ways:

- Polling status register: Read the SPI_SR register and judge the status of RXBUF by RXNE bit or RXBUF_NUM[1:0] bit. When RXBUF has data, the bus reads the data away. This process is the fastest.
- RXNE interrupt: Enable RXNEIE interrupt. SPI interrupt is generated each time a packet of data is received. The SPI_SR register is read in the interrupt service function to determine the RXBUF status and read the data.
- RXDMA: Configure the DMA receiving channel and enable RXDMAEN. Each time a packet of data is received, an RXDMA request is generated, and the data is read sequentially by DMA across the bus. After an RXDMA transfer is complete, you must turn off the RXDMAEN enable and turn it on the next time you use it.

When DMA mode is used for transmission, the HSPEED_EN register needs to be configured to enable the corresponding IO port to be in high-speed mode.

In receiver-only mode, the amount of transmitted data needs to be written to the SPI_NUM register (no CRC frame is required). In receiver-only mode, the NSS must be kept low throughout the transmission process, so the NSSP mode and receiver-only mode are incompatible. After receiving a batch of data (as configured by the SPI_NUM register), RXONLY is disabled by the hardware and RXONLY needs to be turned back on the next time you use it.

If CRC is enabled, SPI performs CRC calculations on the data entered on the RXD side at each clock sampling edge based on the configured CRC length and CRC polynomial until the data volume configured on the SPI_NUM register is transferred, after which the CRC calculations are stopped and another CRC frame is received. If the DS (8/16/32Bit) is inconsistent with the CRC length (8/16Bit), the hardware can automatically correct the DS to the corresponding value during CRC frame transfer.

When CRC is enabled, the NSS must be kept low throughout the transmission, so the NSSP mode and CRC are not compatible. In CRC mode, after receiving a batch of data (as configured by the SPI_NUM register), CRC is disabled by the hardware and CRC needs to be enabled again for the next use. The received CRC check code will be stored in RXBUF, and the calculated CRC check code will be stored in the SPI_RXCRCR register after transmission.

Note: CRC mode only allows DS to be configured as 8/16/32Bit multiples of these 8s, and CRC length can only be configured as 8Bit or 16Bit.

16.3.7. Status flags and interrupts

Interrupt enable:

- **ERRIE**: The error interrupt function was enabled. Output interrupts when Buffer overflow or CRC check error events occur.
- **TXEIE**: Send air break. If the TXBUF is not full, the TXBUF can be outputted.
- **RXNEIE**: Receive interrupt. RXBUF can output interrupts when received data is inside.

Status flag:

- **OVR:** Buffer overflow flag. When TXBUF is full, the bus writes data or RXBUF receives new data, the position is 1, and the software writes 0 to clear the zero. Output error interrupt.
- **CRCERR:** CRC check error flag. When the received CRC value does not match the SPI calculated value, the software writes 0 at this position 1 to clear the value. Output error interrupt.
- **BUSY:** SPI busy flag. When the SPI is currently busy communicating, this bit is pulled up, and when the transmission is complete, the hardware is pulled down. In host mode, keep BUSY high only when the clock is being output, and keep it low for the rest ofthe time. In slave mode, the BUSY signal remains high as long as the input chip is selected as low. This bit is used by the software to determine whether the transfer is complete.
- **TXBUF_NUM[1:0]:** indicates the amount of data in TXBUF.
- **RXBUF_NUM[1:0]:** indicates the amount of data in RXBUF.
- **TXE:** Sends a dissatisfaction flag. When TXBUF is not full, this position is 1.
- **RXNE:** Receives the not empty flag. When RXBUF has received data, this position is 1.

16.4. SPI register

Base address: 0x5000 0000

Address offset	Register	Description
0x04	RCU_EN	Peripheral module clock control register
0x50	HSPEED_EN	SPI high-speed mode enable register

SPI0 base address: 0x5002 0000

SPI1 base address: 0x5002 0100

Address offset	Register	Description
0x00	SPI_CR1	SPI control register 1
0x04	SPI_CR2	SPI control register 2
0x08	SPI_SR	SPI status register
0x0C	SPI_DR	SPI data register
0x10	SPI_NUM	SPI high-speed mode data buffer address number
0x14	SPI_CRCPR	CRC polynomial register
0x18	SPI_RXCRCR	SPI receives CRC registers
0x1C	SPI_TXCRCR	SPI sends CRC registers

16.4.1. Peripheral module clock control register (RCU_EN)

Address offset: 0x04

Reset value: 0x0000 0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.				DIV_CLKEN	GPIOD_CLKEN	GPIOC_CLKEN	GPIOB_CLKEN	GPIOA_CLKEN	DMA_CLKEN	CRC_CLKEN	ADC_CLKEN	WDT_CLKEN	TIM7_CLKEN	TIM6_CLKEN	TIM5_CLKEN
				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
TIM4_CLKEN	TIM3_CLKEN	TIM2_CLKEN	TIM1_CLKEN	TIM0_CLKEN	PWM1_CLKEN	PWM0_CLKEN	IIC_CLKEN	UART4_CLKEN	UART3_CLKEN	UART2_CLKEN	UART1_CLKEN	UART0_CLKEN	SPI1_CLKEN	SPI0_CLKEN	Res.
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	

25	GPIOC_CLKEN	GPIOC module operation enable: 1: Work 0: Off, the default is 0
24	GPIOB_CLKEN	GPIOB module operation enable: 1: Work 0: Off, the default is 0
23	GPIOA_CLKEN	GPIOA module operation enable: 1: Work 0: Off, the default is 0
22	DMA_CLKEN	DMA(including DMA SRAM/DMA SFR/DMAMUX) module was enabled: 1: Work

		0: Off, the default is 0
21	CRC_CLKEN	CRC module operation enable: 1: Work 0: Off, the default is 0
2	SPI1_CLKEN	SPI1 module operation enable: 1: Work 0: Off, the default is 0
1	SPI0_CLKEN	SPI0 module operation enable: 1: Work 0: Off, the default is 0

16.4.2. SPI high-speed mode enable register(HSPEED_EN)

Address offset: 0x50

Reset value: 0x0000 0000

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

15:9	8	7	6	5	4	3	2	1	0
Res.	HSSEL8	HSSEL7	HSSEL6	HSSEL5	HSSEL4	HSSEL3	HSSEL2	HSSEL1	HSSEL0
	RW	RW	RW	RW	RW	RW	RW	RW	RW

31:9	-	Reserved
8	HSSEL8	The high-speed mode is enabled for the SPI communication port PA15: 1: high-speed mode 0: common mode
7	HSSEL7	The high-speed mode is enabled for the SPI communication port PB4: 1: high-speed mode 0: common mode
6	HSSEL6	The high-speed mode is enabled for the SPI communication port PB5: 1: high-speed mode 0: common mode
5	HSSEL5	The high-speed mode is enabled for the SPI communication port PB10: 1: high-speed mode 0: common mode
4	HSSEL4	The high-speed mode is enabled for the SPI communication port PB11: 1: high-speed mode 0: common mode
3	HSSEL3	The high-speed mode is enabled for the SPI communication port PB12: 1: high-speed mode 0: common mode
2	HSSEL2	The high-speed mode is enabled for the SPI communication port PC3: 1: high-speed mode 0: common mode
1	HSSEL1	The high-speed mode is enabled for the SPI communication port PC4: 1: high-speed mode 0: common mode
0	HSSEL0	The high-speed mode is enabled for the SPI communication port PC5: 1: high-speed mode 0: common mode

Note: When the communication frequency is above 1M, high-speed mode must be enabled.

16.4.3. SPI control Register 1(SPI_CR1)

Address offset: 0x00

Reset value: 0x0000 0400

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

15:12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	SPE	SS_N	NSSP	TXONLY	RXONLY	Res.	CRCEN	CRCL	MSTR	LSBFIRST	CPOL	CPHA
	RW	RW	RW	RW	RW		RW	RW	RW	RW	RW	RW

31:12	-	Reserved
11	SPE	SPI enable: 0: Disable SPI 1: Enable SPI
10	SS_N	CS: 0: indicates that the SPI communicates normally 1: indicates that the SPI stops communication Note: This bit is not valid in NSSP mode
9	NSSP	NSS pulse mode: This bit is used by the SPI host to generate NSS pulses. When transmitted continuously, this bit enables SPI to generate NSS pulses between two consecutive data sets. When this bit is valid, the NSS is pulled down only during data transmission, remains high when idle, and the SS_N bit is invalid. This mode works only when CPHA=0 is configured. 0: disables the NSS pulse mode 1: Enables the NSS pulse mode This bit cannot be changed during communication. Note: CRC cannot be used in NSSP mode, and cannot be used in receiving-only mode (pulling up the slice will cause the counter to be cleared and cannot count)
8	TXONLY	Send mode only enable: 0: disables the send only mode 1: enables the send only mode This bit cannot be changed during communication. Note: If TXONLY and RXONLY are set to 0 at the same time, SPI works in full duplex and cannot be set to 1 at the same time.
7	RXONLY	Receive only mode enable: 0: disables the receive only mode 1: Enable the receive only mode When the host is enabled to receive only mode, the clock will output continuously, and the amount of transmitted data needs to be written to the SPI_NUM register to control the clock stop.

		This bit cannot be changed during communication. Note: If TXONLY and RXONLY are set to 0 at the same time, SPI works in full duplex and cannot be set to 1 at the same time.
6	-	Reserved
5	CRCEN	Hardware CRC computing Enable: 0: disables CRC calculation 1: Enables CRC computing When CRC is enabled, the transmitted data must be written to the SPI_NUM register for the CRC to work properly. This bit cannot be changed during communication. Note: CRC can only be used if DS=8/16/32Bit
4	CRCL	CRC Length: 0:8-bit CRC length 1:16 bit CRC length This bit cannot be changed during communication. Note: CRC length is independent of DS=8/16/32Bit width
3	MSTR	Master/slave mode selection: 0: Slave 1: Master This bit cannot be changed during communication.
2	LSBFIRST	Data size end: 0: The MSB is in front when sending/receiving data 1: LSB is in front when sending/receiving data This bit cannot be changed during communication.
1	CPOL	Clock polarity: 0: In idle state, SCLK is 0 1: In idle state, SCLK is 1 This bit cannot be changed during communication.
0	CPHA	Clock phase: 0: Data is sampled from the first clock edge 1: Data is sampled from the second clock edge This bit cannot be changed during communication.

16.4.4. SPI control register 2(SPI_CR2)

Address offset: 0x04

Reset value: 0x0000 0007

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

15:12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	TXDMAEN	RXDMAEN	ERRIE	TXEIE	RXNEIE	BR			DS			

	RW	RW	RW	RW	RW	RW	RW

31:12	-	Reserved
11	TXDMAEN	Send DMA enable: 0: disables DMA transmission 1: indicates that DMA sending is enabled
10	RXDMAEN	Receive DMA Enable: 0: disables DMA receiving 1: indicates that DMA receiving is enabled
9	ERRIE	Error interrupt enable: This bit is used to control the generation of interrupts when an error condition (CRCERR, OVR) occurs. 0: disables error interrupt 1: Error interrupt is enabled
8	TXEIE	Send air break enable: 0: disables TXE interruption 1: indicates that TXE interruption is enabled
7	RXNEIE	Receive full interrupt enable: 0: disables RXNE interruption 1: RXNE interrupt is enabled
6:4	BR	Baud rate predivision bit: 000: f_{HCLK}/4 001: f_{HCLK}/6 010: f_{HCLK}/8 011: f_{HCLK}/12 100: f_{HCLK}/24 101: f_{HCLK}/48 110: f_{HCLK}/96 111: f_{HCLK}/192 This bit cannot be changed during communication. Note: The SPI host supports a maximum communication frequency of 8M. When the system clock HCLK frequency is 48M, configure f_{HCLK}/4, and the communication frequency will exceed the supported frequency.
3:0	DS	Data Size transmitted by SPI: 0000/0001: Reserved 0010:32 bits 0011:4 bits 0100:5 bits 0101:6 bits 0110:7 bits 0111:8 bits 1000:9 bits 1001:10 bits 1010:11 bits 1011:12 bits 11100:13 bits 1101:14 bits 1110:15 bits 1111:16 bits If the software writes one of the reserved values. The data bit width is forced to be 8 bits This bit cannot be changed during communication.

	Note: 32Bit is a whole, not 8Bit*4

16.4.5. SPI status register(SPI_SR)

Address offset: 0x08

Reset value: 0x0000 0002

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

15:9	8	7	6	5	4	3	2	1	0
Res.	OVR	CRCERR	BUSY	TXBUF_NUM		RXBUF_NUM		TXE	RXNE
	RC_W0	RC_W0	R	R		R		R	R

31:9	-	Reserved
8	OVR	Overflow flag: 0: No overflow occurs 1: Overflow occurs This flag is set to 1 by hardware and cleared by 0 written by software. Note: This flag is set when TXBUF is full and when RXBUF is full and data is received.
7	CRCERR	CRC error flag: 0: The received CRC value matches the SPI calculated value 1: The received CRC value does not match the SPI calculated value This flag is set to 1 by hardware and cleared by 0 written by software.
6	BUSY	SPI busy flag: 0: The SPI is not busy 1: SPI is busy communicating There is a wait time of several HCLK cycles from write to send data to BUSY pull
5:4	TXBUF_NUM	Amount of data in TXBUF: 00: There is no data in TXBUF 01: indicates that the TXBUF contains one data 10: indicates that two data sets exist in the TXBUF 11: Reservations
3:2	RXBUF_NUM	The amount of data in RXBUF: 00: There is no data in RXBUF 01: There is one data in RXBUF 10: There are two data sets in RXBUF 11: Reservations
1	TXE	TXBUF Empty flag: 0: TXBUF No free space 1: TXBUF has free space

0	RXNE	RXBUF non-empty flag: 0: RXBUF has no data 1: RXBUF has data

16.4.6. SPI data register(SPI_DR)

Address offset: 0x0C

Reset value: 0x0000 0000

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

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

31:0	DR[31:0]	Data register: When this register is read, the data in RXBUF is read out; When writing to this register, data is written to TXBUF.

16.4.7. SPI data volume register(SPI_NUM)

Address offset: 0x10

Reset value: 0x0000 03FF

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.						SPI_NUM[9:0]									
						RW									

31:10	-	Reserved
9:0	SPI_NUM[9:0]	Amount of data transmitted by SPI at a time =SPI_NUM[9:0]+1: Used only in CRC mode or receive only mode This bit cannot be changed during communication. Note: CRC frames are not included

16.4.8. CRC polynomial register(SPI_CRCPR)

Address offset: 0x14

Reset value: 0x0000 0007

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

Res.

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

31:16	-	Reserved
15:0	CRCPOLY[15:0]	CRC polynomial register: Polynomials for CRC calculations This bit cannot be changed during communication

16.4.9. SPI receives CRC registers(SPI_RXCRCR)

Address offset: 0x18

Reset value: 0x0000 0000

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

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

31:16	-	Reserved
15:0	RXCRC[15:0]	Calculated RXCRC value

16.4.10. SPI sends CRC registers(SPI_TXCRCR)

Address offset: 0x1C

Reset value: 0x0000 0000

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

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

31:16	-	Reserved
15:0	TXCRC[15:0]	Calculated TXCRC value

16.5. SPI configuration process

Master full duplex communication (interrupt data processing)

1. The peripheral clock was enabled.
2. The GPIO multiplexing function needs to be configured. For details, see " GPIO multiplexing function configuration " .
3. Configuring SPE=1.
4. Set MSTR to 1.
5. Configure TXONLY =0 and RXONLY=0.
6. Configure the CPOL,CPHA, and LSBFIRST bits.
7. Configuring ERRIE=1.
8. Configuring RXNEIE=1.
9. Configure the BR and DS.
10. Prewrites two strokes to send data to SPI_DR.
11. Lower SS_N to start communication.
12. Read the SPI_SR register in the interrupt function, if the error flag is set, the communication error occurs, and the communication needs to be restarted; IfRXNE=1 or RXBUF_NUM[1:0] is not zero, it indicates that there is data to be read in RXBUF, and the SPI_DR register needs to be read until RXBUF is empty. If TXE=1 or TXBUF_NUM[1:0] is less than 2, the TXBUF is not full. Data needs to be written and sent to the SPI_DR register until TXBUF is full.
13. Check again that the error flag bit of the SPI_SR register is not raised. If it is raised, the communication fails and needs to be re-initiated.
14. Pull up SS_N.
15. Disable the SPE function.

The host only sends mode NSSP pulse mode DMA processing data

1. The peripheral clock was enabled.
2. The GPIO multiplexing function needs to be configured. For details, see " GPIO multiplexing function configuration " .
3. Configure the DMA sending channel and data, and enable the DMA channel interrupt function.
4. Configuring SPE=1.
5. Configuring NSSP=1.
6. Configuring MSTR=1.
7. Configuring TXONLY =1.
8. Configure the CPOL,CPHA, and LSBFIRST bits.
9. Configuring ERRIE=1.
10. Configure the BR and DS.
11. Configuring TXDMAEN=1.
12. Wait for the DMA channel to complete all data transmission and interrupt, poll the SPI_SR register in the interrupt function, if the error flag is set, the communication error occurs,

and the communication needs to be re-initiated; If BUSY=0, data transfer is complete.

13. Configuring TXDMAEN=0.
14. Check again that the error flag bit of the SPI_SR register is not raised. If it is raised, the communication fails and needs to be re-initiated.
15. Configuring NSSP=0.
16. Disable the SPE function.

16.5.1. Master receives only the mode CRC check

1. The peripheral clock was enabled.
2. The GPIO multiplexing function needs to be configured. For details, see " GPIO multiplexing function configuration " .
3. Configuring SPE=1.
4. Configuring RXONLY=1.
5. Configuring CRCEN=1.
6. Configure CRCL.
7. Configuring MSTR=1.
8. Configure the CPOL,CPHA, and LSBFIRST bits.
9. Configuring ERRIE=1.
10. Configuring RXNEIE=1.
11. Configure BR.
12. Configure DS(Can only be matched in multiples of 8).
13. Configure the SPI_NUM register to write the amount of data to be transferred.
14. Configure the SPI CRCPR register.
15. Lower SS_N to start communication.
16. Read the SPI_SR register in the interrupt function, if the error flag is set, the communication error occurs, and the communication needs to be restarted; IfRXNE=1 or RXBUF_NUM[1:0] is not zero, RXBUF has data to be read, and the SPI_DR register needs to be read.
17. Read the SPI_SR register in the interrupt function. If BUSY is equal to 0, it indicates that data transmission is complete. Read all the data in RXBUF, and note that the last data is the CRC frame received; If the CRCERR position is 1, the CRC check error occurs and the communication needs to be initiated again. Reading SPI_RXCRCR can read the CRC check code calculated by SPI.
18. To initiate a receive only or CRC transmission again, you need to enable RXONLY or CRC again.
19. Check again that the error flag bit of the SPI_SR register is not raised. If it is raised, the communication fails and needs to be re-initiated.
20. Pull up SS_N.
21. Disable the SPE function.

16.5.2. Slave full duplex communication DMA processing data

1. The peripheral clock was enabled.
2. The GPIO multiplexing function needs to be configured. For details, see " GPIO multiplexing function configuration " .
3. Configure the HSPEEND_EN register to enable the corresponding I/O port to high-speed mode.
4. Set the DMA send and receive channels, and enable the DMA channel interrupt function.
5. Configuring SPE=1.
6. Configuring MSTR=0.
7. Configuring TXONLY =0, RXONLY=0.
8. Configure the CPOL,CPHA, and LSBFIRST bits.
9. Configuring ERRIE=1.
10. Configure the BR and DS.
11. Configuring TXDMAEN=1, RXDMAEN=1.
12. Wait for the host to pull down the chip selection and start communication.
13. Polling the SPI_SR register in the DMA interrupt function until the BUSY signal is pulled down (in slave mode, the BUSY signal is pulled down only after the slice is pulled up) indicates that the transmission is complete. If the error flag is set, the communication has failed and needs to be re-initiated.
14. Configuring TXDMAEN=0, RXDMAEN=0.
15. Disable the SPE function.

16.5.3. Master full duplex communication +CRC+ polling processing data

1. The peripheral clock was enabled.
2. The GPIO multiplexing function needs to be configured. For details, see " GPIO multiplexing function configuration " .
3. Configuring SPE=1.
4. Configure CRCPOLY.
5. Configure the CRCEN and CRCL bits.
6. Configuring MSTR=1.
7. Configure the CPOL,CPHA, and LSBFIRST bits.
8. Configuring ERRIE=1.
9. Configure the BR and DS.
10. Configuring SPI_NUM.
11. Lower SS_N to start communication.
12. The software polls the SPI_SR register. IfRXNE=1, the read SPI_DR register will read the received data; if TXE=1, the sent data will be written to SPI_DR (read before write).
13. After receiving the last data in the above way, poll the SPI_SR register until the BUSY signal is pulled down, indicating that the transmission is complete, at which time RXNE should be equal to 1 (CRC frame received), read the SPI_DR register to read away the CRC frame, if the error flag is set, it indicates that there is an error in communication and

the communication needs to be re-initiated.

14. To initiate CRC transmission again, enable CRC again.
15. Pull up SS_N to stop the communication.
16. Disable the SPE function.

16.5.4. Master full duplex communication +CRC+ interrupt processing data

1. The peripheral clock was enabled.
2. he GPIO multiplexing function needs to be configured. For details, see " GPIO multiplexing function configuration " .
3. Configuring SPE=1.
4. Configure CRCPOLY.
5. Configure the CRCEN and CRCL bits.
6. Configuring MSTR=1.
7. Configure the CPOL,CPHA, and LSBFIRST bits.
8. Configuring ERRIE=1.
9. Configuring RXNEIE=1 (only receive interrupts to avoid too many interrupts).
10. Configure the BR and DS.
11. Configuring SPI_NUM.
12. Lower SS_N to start communication.
13. The SPI_SR register is read in the interrupt function. IfRXNE=1, the SPI_DR register is read to read the received data; if TXE=1, the data is sent to SPI_DR. (Read before you write).
14. After receiving the last data in the above way, read the SPI_SR register in the next interrupt, the BUSY signal should be low, indicating that the transmission is complete, and RXNE should be equal to 1 (CRC frame received) at this time, read the SPI_DR register to read away the CRC frame, if the error flag is set, it indicates that there is an error in the communication and the communication needs to be re-initiated.
15. To initiate CRC transmission again, enable CRC again.
16. Pull up SS_N to stop the communication.
17. Disable the SPE function.

16.6. Note

1. The host clock is generated continuously only in receive mode. The amount of data to be transmitted must be configured. Otherwise, the clock cannot be stopped.
2. When CRC is enabled, you must set the amount of data to be transmitted (excluding CRC frames); otherwise, CRC verification fails.
3. Read the SPI_SR register first, ensure that RXBUF has data, and then read SPI_DR, otherwise the read data is invalid.
4. In CRC check mode or receive only mode, after a batch of data (configured by SPI_NUM) is transmitted, CRC or RXONLY is disabled by the hardware. The software needs to enable CRC or RXONLY again.
5. After each DMA channel data is processed, you must disable the DMA function inside the SPI and enable it the next time you use it.
6. It is recommended that SPI_NUM be configured in slave receiver-only mode,keep the master and slave machines unified (RXONLY is disabled when SPI_NUM+1 data is received in slave mode).
7. If the master and slave machines need to receive a large amount of data (exceeding the limit set by SPI_NUM), they can only use the full duplex mode or enable RXONLY once for every SPI_NUM+1 data.
8. The send air break is generated at the second BIT of each frame data, and the send CRC frame will not cause the send air break. A full receive interrupt is generated after each frame of data is received. A full receive interrupt is generated when CRC frames are received.
9. When a communication error occurs, you are advised to re-enable the SPI before using the SPI.
10. When the I/O port is used as the chip selection port, for example, PA0 is set to the common output mode instead of the multiplexing mode. When the communication starts, PA0 is set to low and SS_N=0. To end the communication, run SS_N=1 and set PA0 to a higher value.
11. In half duplex dual BUFFER mode: the host only receives, the slave only sends, the communication frequency is 4M, the DMA mode needs to be used, and the interrupt mode cannot be used.

Chapter 17 Universal asynchronous transceiver (UART)

17.1. UART overview

There are 5 UART modules in the BF7707AMXX series. Support in idle mode 0, UART receiving interrupt can wake up the system to work (interrupt enable); Support in idle mode 1, sleep wake interrupt can wake the system to work (sleep wake enable and receive enable); Supports multiple I/O port mapping.

17.2. UART functional characteristics

- Support full-duplex, half-duplex serial communication.
- With independent double-buffered receiver, single-buffered transmitter,continuous. receiving function can be realized.
- Programmable baud rate (13-bit analog-to-digital frequency divider).
- Interrupt-driven or polling operation:
  - Delivery completed.
  - Receive full.
  - Receive Idle.
  - Receive overflow, parity check error, frame error.
- Support hardware parity verification generation and inspection.
- Programmable 8-bit or 9-bit character length.
- Choose 1 or 2 STOP bit.
- Support multi-processor mode.
- Optional digital filter for receiving port.
- Optional transmitter output polarity and receiver input polarity.
- Supports UARTx remapping and TXD/RXD configuration.
- Send/receive can be enabled separately.
- Support wake-up in standby mode.
- Support DMA mode.

17.3. UART functional description

17.3.1. UART structure diagram

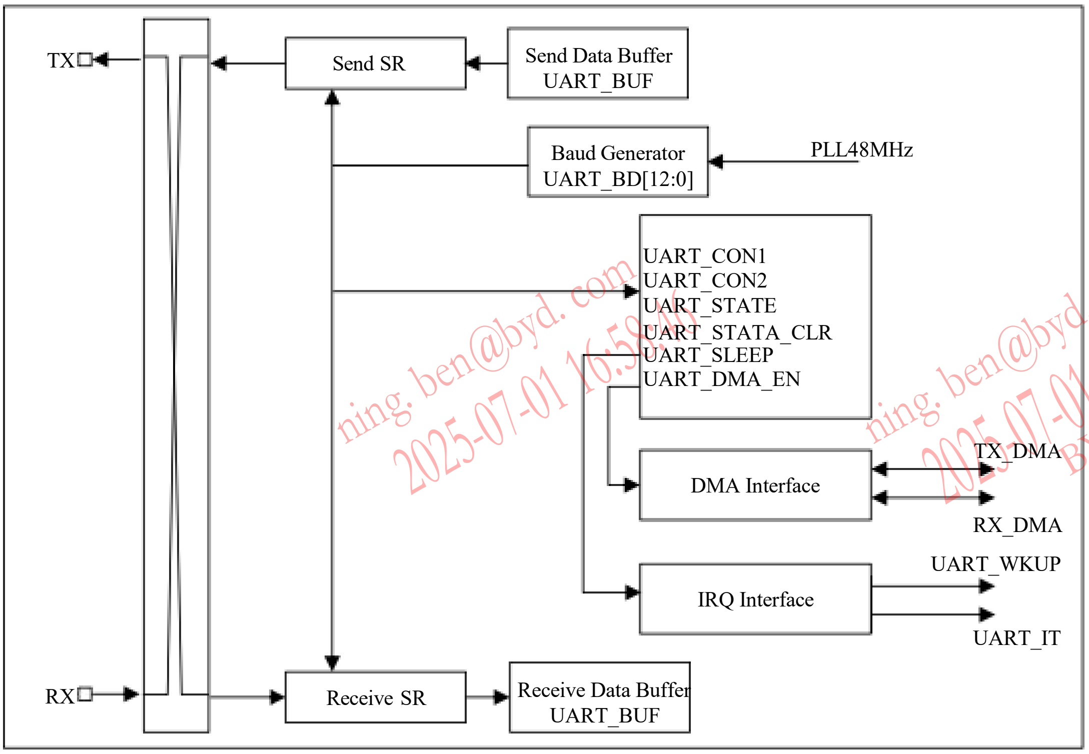


Figure 17.1 UART structure diagram

17.3.2. Baud rate generation

Baud rate generation modulus Baud_Mod = UART_BD [12:0]}.

Baud rate calculation formula: When Baud_Mod=0, the baud rate clock is not generated, when Baud_Mod= 1~8191, the baud rate = BUSCLK/ (16xBaud_Mod). BUSCLK uses the frequency division clock of the system clock source and is fixed at 48M.

Every time the baud rate register is configured, the internal counter will be cleared to regenerate the baud rate signal.

Communication requires the transmitter and receiver to use the same baud rate.

The permissible baud rate deviation for communication is not more than 4.5%.

17.3.3. Transmitter function

The transmitter is enabled by setting the TE bit in the UART_CON1 register. The entire sending process must be carried out when the module is enabled.

Sending data flow: Start sending by writing the UART_BUF value, set the sending interrupt after sending the stop bit, and clear the interrupt flag by software, and wait for the next write.

The idle state of the transmitter output pin (TXD) defaults to a logic high state.If TXDINV = 1, the transmitter output is reversed.

By writing data into the data register (UART_BUF), the data will be directly saved to the sending data buffer and the sending process will be started. In the subsequent complete sending process,do not write to the data register UART_BUF,until the sending is completed after the stop bit, the transmission interrupt flag is set, and UART_BUF is written again to restart a new transmission.

The central element of the serial port transmitter is the transmit shift register with a length of 10/11/12 bits (depending on the setting in the DATA_M control bit). Assuming DATA_M=0, select the normal 8-bit data mode. In 8-bit data mode, there are 1 start bit, 8 data bits, and 1/2 stop bits in the shift register.

Both sending and receiving are in little-endian mode (LSB first).

DMA mode: When the UART uses DMA for sending, the DMA send enable signal pull-up (DMA_TE) is configured and the send data register is empty, a DMA send request is generated. In order to work properly, after the transfer is complete, you need to turn off the DMA send enable and clear the relevant UART status flag.

17.3.4. Receiver function

By configuring RXDINV= 1, the receiver input is reversed.The receiver is enabled by setting the RE bit in the UART_CON1 register. The entire receiving process must be carried out when the module is enabled.(UART_EN = 1).

Receiving data flow: When the receiving enable is valid, the data is received at any time, the receiving interrupt is set after the stop bit is received, and the software clears the interrupt flag.

The currently received data will have a detection mechanism, which can detect three types of errors: receiving overflow, frame error, and parity error, all of which require software to clear the flag. It is recommended that after detecting the receiving interrupt, read the status flag, read the data BUF, and finally clear all the received data status flags (UART_STATE[3:0]).

The data character consists of a logic 0 start bit, 8 (or 9) data bits (LSB first) and a logic 1 stop bit (1bit). After receiving the stop bit into the receive shifter, if the receive data register is not full (RXFUF = 0), the data characters are transferred to the receive data register, and the receive data register is full status flag (RXFUF = 1) is set. If you have set the receive data register full (RXFUF) at this time, set the receive overflow flag (RXOVF), and the new data will be lost. Because the receiver is double-buffered, the program has a full character time for reading after setting the receive data register full (RXFUF) and before reading the data in the receive data buffer to avoid receiver overflow. When the program detects that the receive data register is full (RXFUF = 1), it obtains data from the receive data register by reading UART_BUF.

DMA mode: When the UART uses DMA to receive data, a DMA request is automatically generated after the UART successfully receives data. Receive full mark RXFUF is automatically zeroed out every time DMA reads data from the data register. Note that for subsequent communication, after the transfer using DMA is complete, DMA receive enable needs to be turned off and UART related status flags cleared.

Idle Receiving Idle: After receiving 1 frame of data, UART indicates the period for which RXD remains high. If the holding period exceeds 10/11/12 (depending on DATA_M and STOP) high, the idle receiving idle flag is raised to generate idle receiving interrupt (IDLEIE = 1 is enabled for idle receiving interrupt). The IDLE flag bit is cleared by writing IDLEIFCL = 1 in the software.

17.3.5. Receiver sampling method

The receiver uses a 16 times baud rate clock for sampling. The receiver searches for the falling edge on the RXD serial data input pin by extracting logic level samples at 16 times the baud rate. The falling edge is defined as logic 0 samples after 3 consecutive logic 1 samples. The 16 times baud rate clock is used to divide the bit time into 16 segments, labeled RT1 to RT16 respectively.

The receiver then samples at each bit time of RT8, RT9 and RT10, including the start bit and stop bit, to determine the logic level of the bit. The logic level is the logic level of the vast majority of samples taken during the bit time. When the falling edge is positioned, the logic level is 0 to ensure that this is the real start bit, not noise. If at least two of these three samples are 0, the receiver assumes that it is synchronized with the receiver character and starts Shift receives the following data, if the above conditions are not met, exit the state machine and return to the state of waiting for the falling edge.

The falling edge detection logic keeps looking for a falling edge. If an edge is detected, the sample clock resynchronizes the bit time. In this way, when noise or baud rate is not matched, the reliability of the receiver can be improved.

17.3.6. Multiprocessor mode

Multiprocessor mode, only works in 9-bit mode. When the 9th bit of data received is 1, the receive interrupt is set, otherwise it is not set. The function of this mechanism is to use hardware detection to eliminate the software overhead of processing unimportant information characters. Allows the receiver to ignore characters in messages used for different receivers.

In this application system, all receivers estimate the address character of each message (the 9th bit = 1). Once it is determined that the information is intended for different receivers, subsequent data characters (the 9th bit = 0) will not be received.

Configuration process: Configure receiving enable, configure multi-processor mode, receive address data (the 9th bit = 1), receive and generate an interrupt, the application confirms whether the address matches, and if it matches, the configuration closes the multi-processor mode, and all subsequent data (the 9th bit = 0) can be received and interrupted, until the next time the address data is received, and the address does not match, the multi-processor mode is turned on, and all subsequent data will not be received until the next address data, and loop in turn application.

17.3.7. Standby mode wake-up function

In idle mode 0, the UART clock is turned on, the module supports normal reception, and the receive interrupt can interrupt the wake-up core work, provided that the receive interrupt configuration is enabled.

In idle mode 1, the UART clock is turned off, the module does not support normal reception, but supports wake-up interrupt wake-up core work, provided that wake-up enable, receive enable and module enable are configured. When the receiving port inputs a low level, the wake-up process starts, and the UART interrupt processing function (wake-up status flag bit) is executed after the system wakes up.

17.4. UART Registers

Base address:0x5000 0000

Address offset	Register	Description
0x04	RCU_EN	Peripheral module clock control register

Base address: UART0: 0x5003 0000; UART1: 0x5003 0100;
UART2: 0x5003 0200; UART3: 0x5003 0300; UART4: 0x5003 0400;

Address offset	Register	Description
0x00	UART_BD	Baud rate control register
0x04	UART_CON1	Mode control register 1
0x08	UART_CON2	Mode control register 2
0x0C	UART_STATE	Status flag register
0x10	UART_BUF	Data register
0x14	UART_STATE_CLR	Status flag clear register
0x18	UART_SLEEP	Wake up control register
0x1C	UART_DMA_EN	Receive/Send Channel DMA Enable

17.4.1. Peripheral module clock control register (RCU_EN)

Address offset: 0x04

Reset value: 0x0000 0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.				DIV_ CLKEN	GPIOD_ CLKEN	GPIOC_ CLKEN	GPIOB_ CLKEN	GPIOA_ CLKEN	DMA_ CLKEN	CRC_ CLKEN	ADC_ CLKEN	WDT_ CLKEN	TIM7_ CLKEN	TIM6_ CLKEN	TIM5_ CLKEN
				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
TIM4_ CLKEN	TIM3_ CLKEN	TIM2_ CLKEN	TIM1_ CLKEN	TIM0_ CLKEN	PWM1_ CLKEN	PWM0_ CLKEN	IIC_ CLKEN	UART4_ CLKEN	UART3_ CLKEN	UART2_ CLKEN	UART1_ CLKEN	UART0_ CLKEN	SPI1_ CLKEN	SPI0_ CLKEN	Res.
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	

26	GPIOD_CLKEN	GPIOD module operation enable 1: Work 0: Off, the default is 0
25	GPIOC_CLKEN	GPIOC module operation enable 1: Work 0: Off, the default is 0
24	GPIOB_CLKEN	GPIOB module operation enable 1: Work 0: Off, the default is 0
23	GPIOA_CLKEN	GPIOA module operation enable 1: Work 0: Off, the default is 0

22	DMA_CLKEN	DMA(Includes DMA SRAM/DMAMUX) module operation enable 1: Work 0: Off, the default is 0
7	UART4_CLKEN	UART4 module operation enable 1: Work 0: Off, the default is 0
6	UART3_CLKEN	UART3 module operation enable 1: Work 0: Off, the default is 0
5	UART2_CLKEN	UART2 module operation enable 1: Work 0: Off, the default is 0
4	UART1_CLKEN	UART1 module operation enable 1: Work 0: Off, the default is 0
3	UART0_CLKEN	UART0 module operation enable 1: Work 0: Off, the default is 0

17.4.2. Baud rate control register (UART_BD)

Address offset: 0x00

Reset value: 0x0000 0000

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

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

31:13	-	Reserved
12:0	BD	Baud rate modulus divisor register

17.4.3. Mode control register 1 (UART_CON1)

Address offset: 0x04

Reset value: 0x0000 0000

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

15:8	7	6	5	4	3	2	1	0
Res.	UART_EN	TE	RE	MULTI_M	STOP	DATA_M	PCE	PS

	RW	RW	RW	RW	RW	RW	RW	RW

31:8	-	Reserved
7	UART_EN	Module enable 1: Module enable 0: The module is off
6	TE	Transmitter enable 1: Transmitter is enabled 0: Transmitter is off
5	RE	Receiver enable 1: The receiver is turned on 0: The receiver is off
4	MULTI_M	Multi-processor communication mode 1: Mode enable 0: Mode disabled
3	STOP	STOP bit width selection 1: 2 bits 0: 1 bit
2	DATA_M	Data mode selection 1: 9-bit mode 0: 8-bit mode
1	PCE	Parity check enabled 1: Parity check enabled 0: Parity check is disabled Note: Applicable to 9-bit mode only, the 9th bit sent by the transmitter is the bit generated by the calculation of the 8-bit data, and the value obtained by the receiver verifies the first 8-bit data is compared with the 9th bit.
0	PS	Parity selection 1: Odd parity 0: Even parity

17.4.4. Mode control register 2 (UART_CON2)

Address offset: 0x08

Reset value: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	4	2	1	0
Res.										TXDINV	RXDINV	IDLEIE	TXIEN	RXIEN	DFILEN
										RW	RW	RW	RW	RW	RW

31:6	-	Reserved
5	TXDINV	Data inverting on TXD: 1: The data at the sending end is reversed, and all the output signals are reversed 0: data on the sending end is not reversed
4	RXDINV	RXD end inverting: 1: The received data is reversed, and all the input signals are reversed 0: The received data is not reversed
3	IDLEIE	Receive idle interrupt enable: 1: indicates that the interrupt function is enabled 0: interrupt disable (for polling mode)
2	TXIEN	Transmit interrupt enable 1: Interrupt enable 0: Interrupt disabled (used in polling mode)
1	RXIEN	Receive interrupt enable 1: Interrupt enable 0: Interrupt disabled (used in polling mode)
0	DFILEN	Filtering is enabled, the input signal of the receiving port can be configured to select the filter function 1: Enable 0: Disable

17.4.5. Status flag register (UART_STATE)

Address offset: 0x0C

Reset value: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.										IDLEIF	TXEMF	RXFUF	RXOVF	FEF	PEF
										R	R	R	R	R	R

31:6	-	Reserved
5	IDLEIF	Receive idle flag: 1: The receiving data cable is idle 0: The receiving data cable is not idle
4	TXEMF	Send interrupt flag 1: The sending buffer is empty 0: The sending buffer is full
3	RXFUF	Receive interrupt flag 1: The receive buffer is full 0: The receive buffer is empty
2	RXOVF	Receive overflow flag

		1: Receive overflow (new data is lost) 0: No overflow
1	FEF	Frame error flag 1: Frame error detected 0: No frame error detected
0	PEF	Parity error flag 1: Receiver parity error 0: Receiver parity check is correct

17.4.6. Data register (UART_BUF)

Address offset: 0x10

Reset value: 0x0000 00FF

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

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

31:9	-	Reserved
8:0	BUF	Data register Read returns the contents ofthe read-only receive data buffer, write into the write-only transmit data buffer

17.4.7. Status flag clear register (UART_STATE_CLR)

Address offset: 0x14
Reset value: 0x0000 0000

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

15:6	5	4	3	2	1	0
Res.	IDLEIFCL	TXEMFCL	RXFUFCL	RXOVFCL	FEFCL	PEFCL
	W	W	W	W	W	W

31:6	-	Reserved
5	IDLEIFCL	Receive idle flag clear and write 1 clear IDLEIF
4	TXEMFCL	Send interrupt flag to clear, write 1 to clear TXEMF
3	RXFUFCL	Receive interrupt flag is cleared, write 1 to clear RXFUF
2	RXOVFCL	Receive overflow flag to clear, write 1 to clear RXOVF
1	FEFCL	Frame error flag is cleared, write 1 to clear FEF
0	PEFCL	Clear parity error flag, write 1 to clear PEF

17.4.8. Wake-up control register (UART_SLEEP)

Address offset: 0x18
Reset value: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.													WASFCL	WASF	WAKEN
													W	R	RW

31:3	-	Reserved
2	WASFCL	Idle mode 1 wake-up state clear flag Write 1 to clear zero, write only
1	WASF	Idle mode 1 wake-up status flag, read only
0	WAEN	Idle mode 1 wake-up enable 1: Enable 0: Disable

17.4.9. Receive/Send Channel DMA Enable (UART DMA EN)

Address offset: 0x1C

Reset value: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.														DMA_RE	DMA_TE
														RW	RW

31:2	-	Reserved
1	DMA_RE	Receive channel DMA enable: 1: Enables the UART to store the received data from the UART_BUF directly into the defined SRAM space via DMA 0: Indicates that data is disabled and receives data through the normal UART receiving process
0	DMA_TE	Send channel DMA Enable: 1: Enables the UART to read data directly from the defined SRAM space to the UART_BUF through DMA 0: Disables the UART to send data through the normal UART sending process

17.5. UART configuration process

Common sending and receiving configuration process:

1. Turn on the peripheral clock.
2. GPIO UART multiplexing function selection needs to be configured.,for details see ″ GPIO multiplexing function configuration " .
3. Configure module enable, send enable, receive enable, mode selection: UARTx_CON1;
4. Configure the baud rate and turn on the interrupt enable: UARTx_BD, UARTx_CON2;
5. Write UARTx_BUF to start sending data. After detecting the sending interrupt, clear the sending interrupt flag TXEMF. Once the sending process is completed, wait for the next write to UARTx_BUF to start the sending process (it is not allowed to configure the next data during the sending process, including UART_BUF).
6. Detect the receiving interrupt, first read the receiving status UARTx_STATE, then read R8 and UARTx_BUF, and finally clear the receiving status flag, once the receiving process is completed, wait for the next receiving interrupt.
7. If the configuration interrupt is not enabled, the program polls to perform the UART function, and also reads the status flag first, then reads UARTx_BUF, and finally clears the receiving status flag.

DMA communication configuration process:

1. Turn on the peripheral clock.
2. GPIO UART multiplexing function selection needs to be configured.,for details see ″ GPIO multiplexing function configuration " .
3. Disabling UART interrupt was configured.
4. Enable UART EN on the UART module.
5. Configure the UART to enable RE for receiving messages or TE for sending messages.
6. Configure the DMA master data structure base address CTRL_BASE_PTR and DMA_EN of DMA_CFG.
7. Configure the DMA send channel Enable CHNL_EN_SET Channel interrupt Enable CHNL_INT_EN.
8. Configure the DMA source address CHNLx SRC END PTR,destination address CHNLx_DST_END_PTR CHNLx_CONTROL Determines the size, quantity, and type of the data to be transmitted (Transfer data size: 8-bit data mode can choose bytes and half words, 9-bit data mode can choose half words; R_power must remain 0 and cannot be configured).
9. Enable UART DMA.

17.6. Note

1. When clearing the interrupt flag bit, note that only the clear bit (UARTx_STATE_CLR) corresponding to the flag bit to be cleared needs to be set to 1, and other bits need to be written to 0, which can prevent false clearing;
2. It is not recommended to modify the configuration during communication: UARTx_BD and UARTx_CON1[3:0] (STOP /DATA_M/PCE/PS), otherwise the current frame communication will be invalid.
3. For continuous DMA communication, it is recommended that you turn off the DMA receive/Send enable and then turn it back on each DMA interrupt after processing the interrupt flag. If the DMA_TE is not turned off and turned on again after the completion of a DMA communication, the first data cannot be sent out normally during the next DMA communication.
4. After receiving the idle interrupt and entering the interrupt handling function, UART needs to clear the idle flag first to prevent it from entering the interrupt handling function normally after receiving or sending interrupt due to the uncleared flag.

Chapter 18 Internal integrated circuit (IIC)

18.1. IIC overview

The IIC bus interface handles communication between the microcontroller and the serial IIC bus, providing multi-master mode capabilities to control all IIC bus-specific sequences, protocols, arbitrations, and timing.

This circuit is based on interrupt service subroutine to complete the communication. After each byte is sent or received by the IIC, an interrupt signal is generated, which tells the CPU to process it accordingly, so as to realize the sending or receiving of data. When the processing time of interrupt service subroutine is too long, the data cannot be processed normally, and the low level time must be extended to achieve the purpose of normal communication. And when the communication is complete, pull down the clock line to determine the subsequent communication status.

18.2. IIC functional characteristics

- Slave mode and master mode.
- Multi-master arbitration mode is optional.
- Two serial interfaces: Serial data line SDA and serial clock line SCL.
- Compliant with the standard communication protocol of philips.
- Support standard mode (up to 100kHz), fast mode (up to 400kHz), super fast mode (up to 1MHz).
- Timing time parameters can be configured.
- Support 7-bit address addressing.
- Multiple 7-bit slave addresses (2 slave device address registers, 1 with configurable mask bit segments).
- Supports the reply mode of all 7-bit addresses.
- Support broadcast call.
- The slave mode supports NACK/ACK control.
- Easy to use interrupt event management.
- Optional clock low extension.
- Slave mode supports overflow detection in unextended clock mode.
- Main mode extends clock low when data is not ready and not received.
- The master mode supports the function of detecting low level extension of the slave mode clock.
- Support Register shutdown enables reset of internal state machine and related timing.
- Wake up from system standby mode when the address matches.
- Supports DMA read and write requests.
- Independent clock :iic_clk(24MHz), so that the communication speed is not affected by changes in the system clock frequency.

Note: When the slave communication rate reaches 1MHz or when the slave needs to process other transactions, the slave can extend the low level time of the clock line by pulling down the clock line.

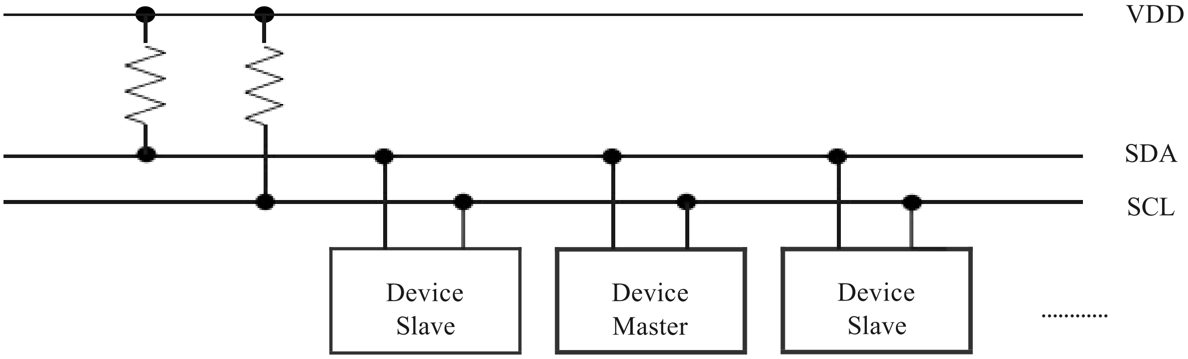


Figure 18.1 IIC master/slave computer connection diagram

The host and slave are connected by the SCL(serial clock) line and SDA(serial data) line, and the SCL and SDA must be connected to the pull resistance (4.7k~10k is recommended).

18.3. IIC functional description

18.3.1. IIC structure block diagram

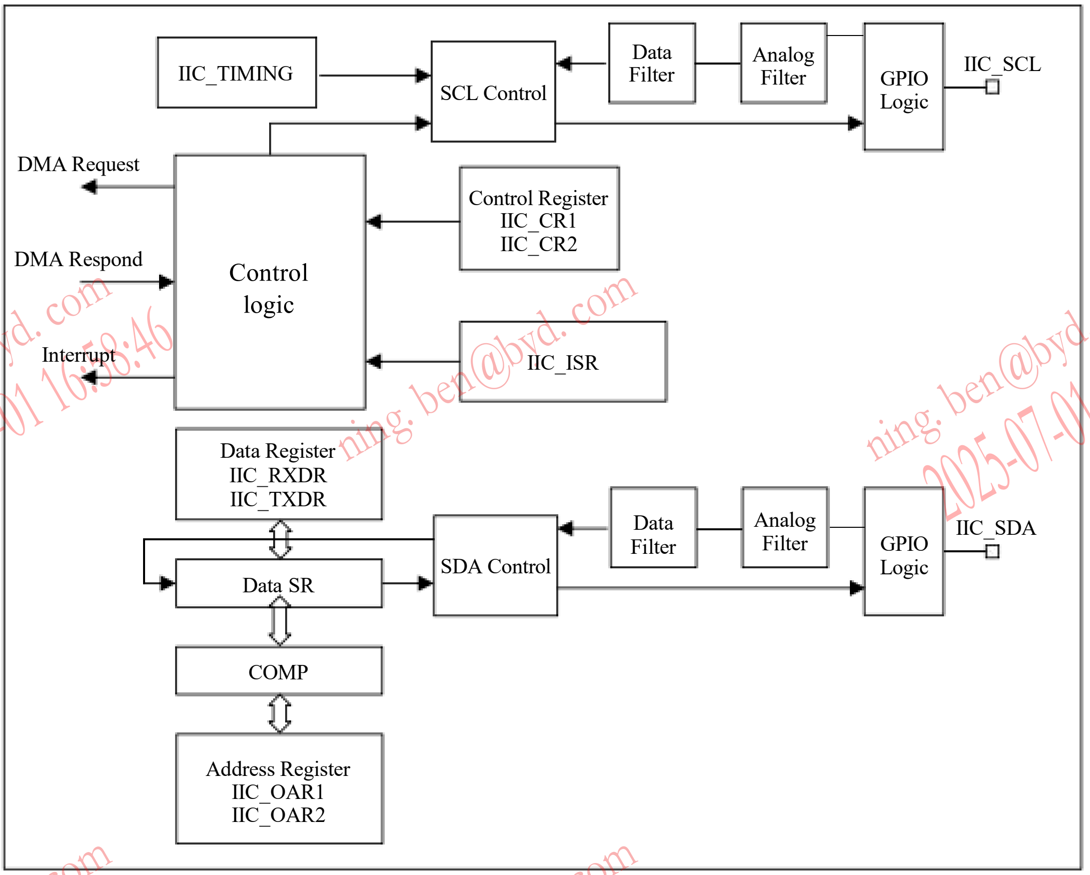


Figure 18.2 IIC structure block diagram

18.3.2. Supports 7-bit addressing and broadcast calls

The BF7707AMXX IIC only supports 7-bit addressing and supports broadcast calls. When GCEN=1, the slave replies to address 00000000B.

After generating the initial signal, the host sends the address code of the slave to be addressed, and communicates ifit receives a matching reply signal from the slave. If no reply is received from the incoming server, the host generates a stop signal and returns to the bus idle state. The slave machine performs authentication by comparing the contents ofthe address code with the slave machine address after receiving it. Communication can be performed only if the addresses match. The slave address is configured by registers IIC_OAR1 and IIC_OAR2.

Note: After the correct address of the slave, there must be read and write data operations, otherwise the communication will be affected.

18.3.3. Slave device support NACK/ACK control mode

In slave mode, byte counters are disabled by default. The NACK/ACK control function can be used when the slave wants to control whether to reply to received data bytes. Enable the NACK/ACK control mode by setting the SBC (slave byte control) position 1 in the IIC_CR1 register and the RELOAD (reload mode) position 1 in the IIC_CR2 register. In addition, the NACK/ACK control mode must be used when extending the clock line is allowed.

In this mode, after receiving the number of programmed bytes of data in the NBYTES, extend the low level time between the 8th and 9th pulses ofthe SCL signal. The TCR flag is set to 1, and an interrupt is generated when TCIE is set to 1. The answer is then determined by configuring the NACK bit in the IIC_CR2 register.

When this mode is not used, the slave sends an ACK all the time.

18.3.4. Extend the clock low level

The slave has the option to extend the clock low level. Controlled by the NOSTRETCH bit of register IIC_CR1, the default NOSTRETCH = 0, can extend the clock low level. The IIC extends the clock low level from the opportunity in the following situations.

- When ADDR flag is set to 1: The received address matches one of the enabled slave addresses, extending the low level between the 8th falling edge of the clock and the 9th rising edge of the clock. The clock line is released when the ADDRCF position 1 is marked by clear ADDR through software. This extended clock low function is not valid in DMA.
- Data that is not ready to be sent before the first clock is extended low at the low level between the falling edge of the ninth clock and the rising edge of the first clock. Until IIC_TXDR is written.
- If the data of the previous stroke is not read before the 8th clock falling edge, the low level is extended between the 8th clock falling edge and the 9th clock rising edge. Until IIC_RXDR is read.
- In the slave device NACK/ACK control mode, when the last data transfer is completed, the low level is extended between the 8th clock falling edge and the 9th clock rising edge. Until the IIC_CR2 register is written and the NBYTES are not zero (the NACK bit is recommended).

The master has the function of extending the clock low by default. The host will extend the clock low level under the following conditions.

- Data that is not ready to be sent before the first clock is extended low at the low level between the falling edge of the ninth clock and the rising edge of the first clock until IIC_TXDR is written.
- If the data of the previous stroke is not read before the 8th clock falling edge, the low level between the 8th clock falling edge and the 9th clock rising edge is extended until the IIC_RXDR is read.
- In non-reload mode (RELOAD = 0) and software end mode (AUTOEND = 0), when the

last data transfer is completed, the low level is extended between the 9th falling edge of the clock and the 1st rising edge of the clock until the start bit or stop position 1.

- In RELOAD mode (RELOAD = 1), when the last data transfer is completed, the low level between the ninth falling edge of the clock and the first rising edge of the clock is extended low level until the IIC_CR2 register is written and NBYTES are not zero (writing NBYTES to a non-zero value is recommended).

18.3.5. Without extend the clock low level detection overflow exception

When NOSTRETCH = 1 in the IIC_CR1 register, the IIC slave device does not extend the SCL signal.

- When the ADDR flag is set to 1, the SCL clock is not extended.
- When sending, data must be written to the IIC_TXDR register before the first SCL pulse corresponding to the sent data appears. Otherwise, underflow occurs, the OVR flag in the IIC_ISR register is set to 1, and an interrupt is generated if the ERRIE position in the IIC_CR1 register is 1. When the first data transmission begins and the STOPF bit is still set to 1 (not cleared), the OVR flag is also set to 1. Therefore, if the STOPF flag of the previous transfer is cleared only after the first data to be sent in the next transfer is written, the OVR state will appear, even for the first data to be sent.
- When receiving, data must be read from the IIC_RXDR register before the 9th SCL pulse (ACK pulse) of the next data byte appears. Otherwise, an overflow occurs, the OVR flag in the IIC_ISR register is set to 1, and an interrupt is generated if the ERRIE position in the IIC_CR1 register is 1.

18.3.6. Detect the slave clock low level extension function

When the host just sends out SCL_OUT clock high power level, the input SCL_IN status is detected. If the input SCL_IN is low, it indicates that the slave clock has low level extension phenomenon. The host clock remains high until the slave clock low level is extended, and the host continues to lower the high level after the slave clock is extended.

18.3.7. IIC timing

BF7707AMXX In order to meet the requirements ofIIC communication, the following time parameters can be configured, see the description of register IIC_TIMING.

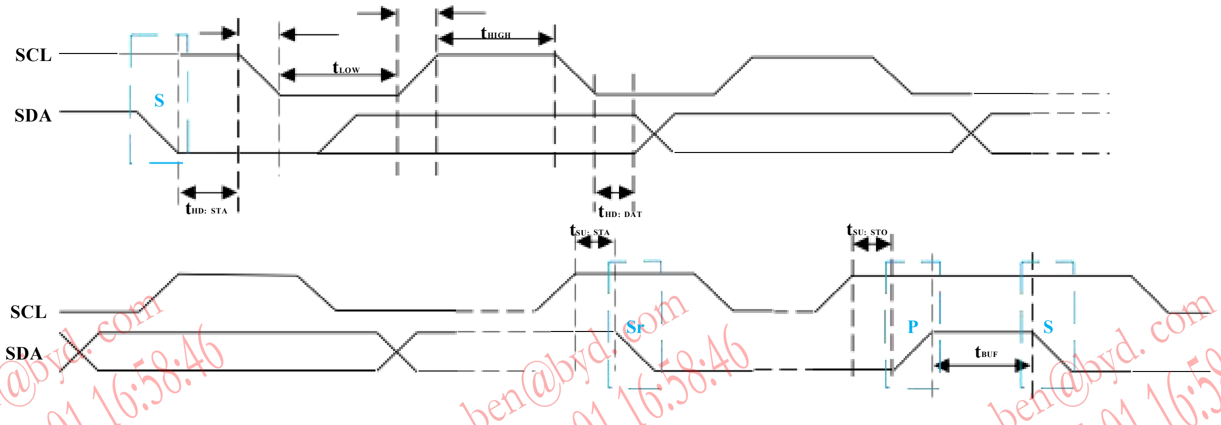


Figure 18.3 IIC master sequence diagram

S: The starting signal, SCL and SDA lines are high at the same time, and the signal from high to low appears on the SDA line.

Sr: When there is no stop signal between the two start signals, a repeat start signal is generated.

P: Stop signal, when the SCL line is high, the SDA line appears from low to high signal.

t_{HIGH}: High level of SCL clock, configured by SCLH;

t_{LOW}: Low level of SCL clock, configured by SCLL;

$t_{HD:DAT}$: SDA data retention time, which is configured by SDADEL.

$t_{HD: STA}$: The retention time of the initial condition, configured by SCLH;

$t_{SU: STA}$: Repeat the establishment time of the initial condition, configured by SCLL;

$t_{SU: STO}$: The establishment time of the stop condition is configured by SCLH.

t_{BUF}: The bus idle time between stop and start conditions, configured by SCLL.

- Without extending the clock low:

  The response signal holding time is fixed as: 3*hclk + 4*iic_clk.

  The creation time of the highest bit of data is: t_{LOW} – (3*hclk + 4*iic_clk).

  Limit case: When the IIC communication frequency is 1M and the system clock is 12M, 3*hclk + 4*iic_clk=416.3ns, the SCLL configuration can be appropriately enlarged and the SCLH configuration reduced.
- Extend the clock low:

  t_{LOW} must be greater than 3*hclk + 2*iic_clk. In the mode where sending stops and starts again, the START bit must be written after the STOPF flag is detected. Otherwise, the stop bit may fail to be sent successfully.So the t_{LOW} minimum is 3*hclk + 4*iic_clk + $t_{SU:SDA}$.
- t_{HIGH} is at least 3*iic_clk.
- t_{HD}:DAT: SDA data retention time = (SDADEL+1)*(PRESC+1)* iic_clk.

  Limit case: Both SDADEL and PRESC are configured to 0, and the minimum data retention time is 1 iic_clk (41.6ns). SDADEL cannot be set to zero.

18.3.8. Slave interrupt event management

The IIC of BF7707BMXX has multiple interrupt management events. It takes effect only when the corresponding enable configuration is enabled. For details, see the IIC_ISR register.

- The IIC of BF7707BMXX has multiple interrupt management events. It takes effect only when the corresponding enable configuration is enabled. For details, see the IIC_ISR register.
- Send data register for air break INT_TXIS: The first clock is not ready when the pen to send data is generated after the 9th SCL drop edge and the send data register is empty (TXE=1). Non-overloaded mode does not generate this interrupt after the last data is sent.
- Transfer completion interrupt INT_TC: Wait for transfer completion Overload (TCR). In overload mode and byte control, after receiving the set NBYTES, it is generated at the 8th SCL falling edge.
- The receive data register is not in the air. INT_RXNE: The last received data was not read until the eighth clock down edge. Generated at the 8th SCL falling edge when the received data register is full (RXNE=1).
- INT_NACK: A negative response was received, generated at the 9th SCL falling edge.
- Stop bit check interrupt INT_STOP: The host has sent the stop bit. This interrupt occurs when a stop signal is detected during transmission.
- Error interrupt INT_ERRIR: Arbitration loss (ARLO)/ Bus error detection (BERR)/ overflow/underflow. The interrupt occurs when these errors occur during transmission.

	The 8th SCL drops after the edge	The 9th SCL drops after the edge	1-9 Number of interrupted CPU in the process
Receiving address			
Send data after receiving the address	INT_ADDR	INT_TXIS	2
Receive the data after receiving the address	INT_ADDR	-	1
Send			
During sending	-	INT_TXIS	1
Receive NACK	-	INT_NACK	1
Receive			
On receiving	INT_RXNE	-	1
Byte control mode Last data + no delay clock line	INT_RXNE+INT_TC	-	1
Byte control mode Last bit of data + extended clock line	INT_RXNE+INT_TC	-	2

18.3.9. Master interrupt event Management

To ensure proper communication, the IIC host mode has multiple interrupt management events. The corresponding enable configuration takes effect. For details, see the IIC_ISR register and IIC timing description.

- The send data register is empty INT_TXIS: The first clock is not ready when the pen is to send the data. Occurs when the send data register is empty (TXE=1) after the 9th SCL falling edge. Non-overloaded mode does not generate this interrupt after the last data is sent.
- Transfer completion interrupt INT_TC: Transfer Completion (TC) or Transfer Completion Wait Reload (TCR). Generates the 9th SCL falling edge of the last data completion in non-overloaded mode or the 255 data completion in overloaded mode.
- The receive data register is not empty. INT_RXNE: The last received data has not been read until the eighth clock down edge. Generated at the 8th SCL falling edge when the received data register is full (RXNE = 1).
- INT_NACK: A negative response was received. Generated at the 9th SCL falling edge.
- Stop bit check interrupt INT_STOP: The host has sent the stop bit.
- Error interrupt INT_ERRIR: Arbitration loss (ARLO) or Bus error detection (BERR). This interrupt occurs when a quorum loss or bus error occurs during transmission.

Parameter	The 8th SCL drops after the edge	The 9th SCL drops after the edge	1-9 Number of interrupted CPU in the process
Send			
During sending	-	INT_TXIS	1
Send overloaded 255th data + Does not extend the clock low level	-	INT_TXIS + INT_TC	1
Send overloaded 255th data + Extend the clock low	-	INT_TXIS + INT_TC	2
Send non-reload last tranche of data + software end	-	INT_TC	1
Send non-overloaded last tranche of data + automatically end	-	-	0
Receive			
On receiving	INT_RXNE	-	1
Receives overloaded 255th data	INT_RXNE	INT_TC	2
Receive non-overloaded last data + end of software	-	INT_TC	1
Receive non-overloaded last data + automatically end	-	-	0
Other situations			
Receive NACK, NACKEND = 1	-	INT_NACK	1

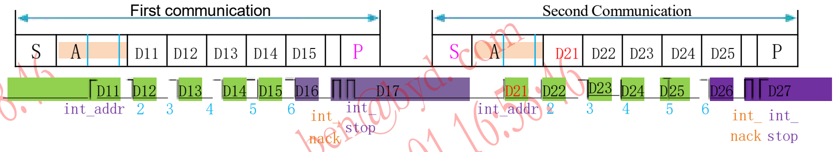
NACK is received during sending, and NACKEND = 0	-	INT_NACK + INT_TXIS	1
Send overloaded 255th data received by NACK+ without extending the clock low level	-	INT_NACK + INT_TXIS + INT_TC	1
Send overloaded 255th data received by NACK+ extended clock low	-	INT_NACK + INT_TXIS + INT_TC	2
Send last data received NACK+NACKEND = 0+ software end	-	INT_NACK + INT_TXIS + INT_TC	1
Send the last data received by NACK+NACKEND = 0+ automatically ends	-	INT_NACK	1

18.3.10. Slave interrupt diagram

18.3.10.1. Schematic diagram of the interrupt signal during transmission

1/2/3...: int_txis
S: Start
P: Stop
A: address
D: Data

Start after stop



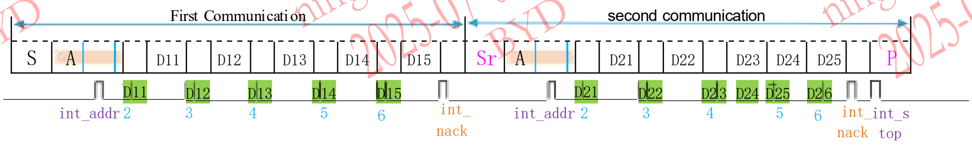
After receiving the address, enter the address interrupt. In the address interrupt, determine the write and read flag DIR and the matching address ADDCODE[7:0]. If DIR=1, the slave machine sends data, the first data can be prepared in the address interrupt, or in advance. Finally write ADDRCF to release extended clock low level.

In the second interrupt int_txis, prepare the second data to be sent, and so on. When the host sends an NACK, it detects the NACKF flag and writes NACKCF to clear. At the same time, int_nack is generated. In this case, int_txis is not generated regardless of whether the data is empty.

When the host sends a stop signal, the STOPF flag is detected, and STOPCF is written to clear. int_stop is generated.

The D16 written in the sixth interrupt int_txis can be used as the first data for the next communication. You can also update the first data in subsequent interrupts by setting TXE to 1. D16, D17 are invalid data, and the first data is D21.

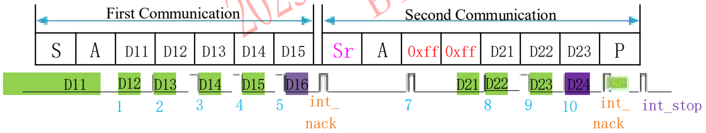
Extend the clock low level



In the first communication phase, the first data is not prepared, the clock low level is extended, the first data D11 is prepared in the first int_txis interrupt, and then the D11 is issued, while the second int_txis interrupt is generated for preparing the second data. After the first data is sent, the second data is not ready, the clock low level is extended again, and after the second data is written, D12 is issued, and a third int_txis interrupt is generated. And so on. In the fifth int_txis interrupt, prepare the last data, D15.

In the second communication phase, the first data was not prepared in advance during initialization. Before D23 is sent, the third data D24 is prepared in the 11th interrupt, and then D24 is sent without extending the clock low level.

Without extend the clock low level overflow



If the clock low extension function is disabled, address interrupt is disabled.

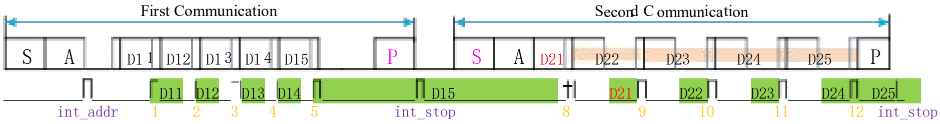
First communication: Prepare the first data in advance, and each interrupt can prepare the next data to be sent in advance to achieve normal communication.

Second communication: The first batch of data is not prepared in advance, and overflow occurs. The first batch of data is all 1. The second batch of data is all 1 and no int_txis interruption is generated after the first batch of data is sent.

18.3.10.2. Diagram of interruption during data reception

1/2/3……: int_rxne

Without extend + Extend the clock low response

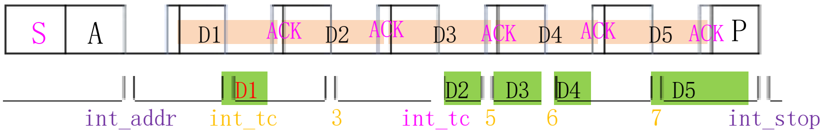


After receiving the address, enter the address interrupt. In the address interrupt, determine the write and read flag DIR and the matching address ADDCODE[7:0]. If DIR=0, data is received from the slave machine. Finally write ADDRCF to release extended clock low level.

The first communication receives five packets of data. After receiving eight bits of data, int_rxne is interrupted. Read D11 in the first int_rxne, D12 in the second interrupt, and so on. Read the last data D15 after the fifth interrupt and before the first data is sent in the next communication. Generates int_stop when a stop signal is detected.

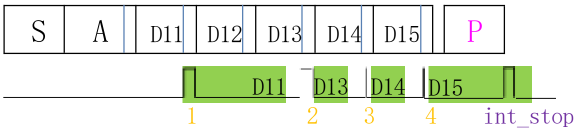
The second communication receives 5 packets of data. After entering the eighth interrupt int_rxne, D21 is not read when the D22 reception is complete, and the clock low level needs to be extended until D21 is read. A ninth interrupt, int_rxne, is then generated, reading D22. After the 12th interrupt int_rxne, the last data D25 is read.

Byte control mode



The slave controls the ACK bit after receiving each data stroke, and the eighth clock falling edge of each data stroke produces int_tc. The eighth clock falling edge of the first data generates both int_rxne and int_tc, extending the clock low level and releasing the clock after writing the NACK signal. The second data, D2, is not read before receiving D3, extending the clock low level to read D3. After the eighth clock fall edge of the third data, int_tc and int_rxne have two different interrupts, and there are two extended clock low levels, writing the NACK signal and reading D3 are completed before the clock is released.

Without extend the clock low level overflow



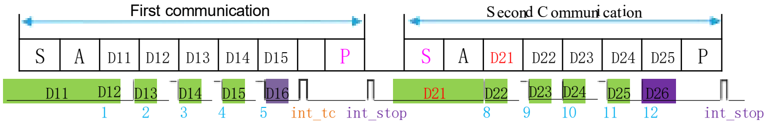
If the first data D11 is not read away before the second data D12 is received, D12 is lost, and int_rxne is not generated after the eighth clock falling edge of D12. Read D11 before receiving the third data D13, D13 will not be lost.

18.3.11. Master interrupt diagram

18.3.11.1. Schematic diagram of the interrupt signal during transmission

1/2/3...: int_txis
S: Start
P: Stop
A: Address
D: Data

Start after stop



The first communication sends 5 batches of data, and the software ends. The second communication sends 5 packets of data and automatically ends.

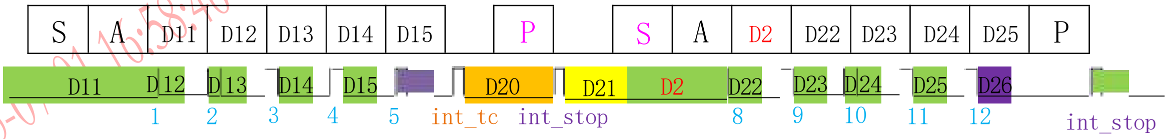
For the first communication, the first data is prepared in advance at the time of initialization, that is, D11 is prepared before the address response is complete. Send D11 immediately after receiving the address reply, prepare the second data D12 in the first int_txis interrupt, and so on. Prepare the final data D15 in the fourth int_txis interrupt.

If data D16 is written in the fifth int_txis interrupt, the hardware sets TXE to 1 after D15 is sent, and D16 is lost as invalid data.

When the last D15 is sent, int_txis is not generated, and int_tc is interrupted, waiting for the write STOP. After issuing the stop bit, int_stop is generated.

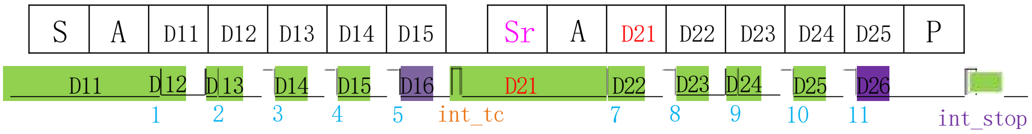
Write START to restart the communication. The first data D21 of the second communication can be prepared in advance before int_tc is interrupted until the address answer is complete. Similarly, D26 is invalid data. After sending the last data D25, the stop bit is automatically issued without any interruption. After the stop is issued, int_stop is generated.

Start and change the first data after stopping



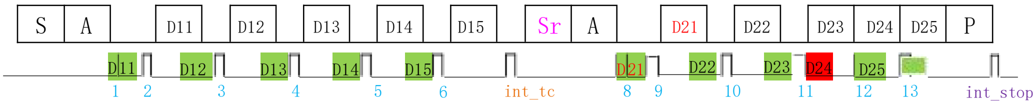
The first data of the second communication can be prepared in advance before int_tc is interrupted until the address answer is complete. The first data can be updated by setting TXE to 1 through software. The last data written before the address answer is complete, D2, is the first data of the second communication. Similarly, D26 is invalid data.

Restart



Software end mode: After the first communication ends, int_tc is interrupted, and START is written to restart the communication.

Extend the clock low level

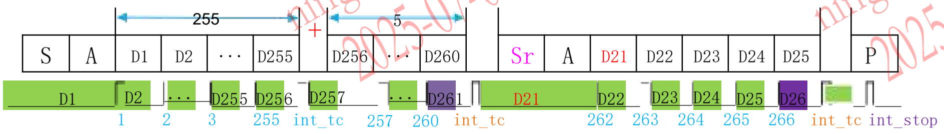


Five data transactions are sent per communication.

In the first communication phase, the first data is not prepared in advance during initialization, the clock low level is extended, the first data D11 is prepared in the first int_txis interrupt, and then D11 is issued, while the second int_txis interrupt is generated for preparing the second data. After the first data is sent, the second data is not ready, the clock low level is extended again, and after the second data is written, D12 is issued, and a third int_txis interrupt is generated. And so on. In the fifth int_txis interrupt, prepare the last data, D15.

In the second communication phase, the first data was not prepared in advance during initialization. Before D23 is sent, the third data D24 is prepared in the 11th interrupt, and then D24 is sent without extending the clock low level.

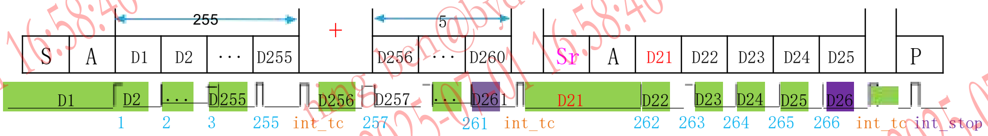
Send data >255



The first communication needs to send 260 data, 255+5 respectively. First select the overload mode, send 255 data, generate int_tc, extend the clock low level, select the non-overload mode, and write the number of to be sent 5.

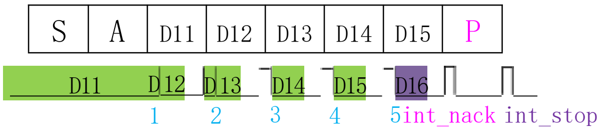
int_txis is generated when int_tc is generated, but in reality the CPU only enters an interrupt once.

Send Data >255 extend the clock line



The 255th int_txis interrupt is not ready for the 256th data. After 255 data is sent, only int_tc is generated, and no int_txis is generated. The low level is extended until the 256th data is written, and D256 is sent, and the 257th interrupt int_txis is generated.

Receive NACK



Need to send ten data.

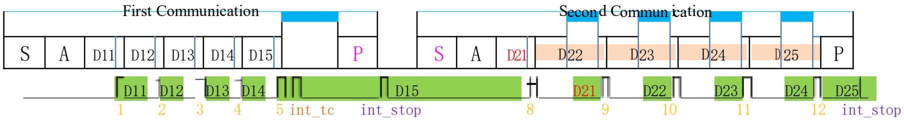
int_nack is generated when the NACK is received after D15 is sent. When NACKEND=1, the stop bit is automatically sent.

If NACKEND=0, continue to send the remaining 5 data.

18.3.11.2. Diagram of interruption during data reception

1/2/3...: int_rxne

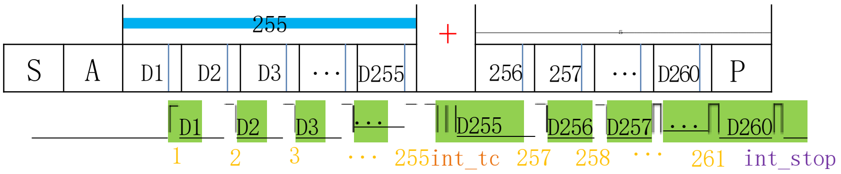
Without extend + extend the clock line



Five packets of data are received during the first communication. After 8-bit data is sent, int_rxne is interrupted. Read D11 in the first int_rxne, D12 in the second interrupt, and so on. Read the last data D15 after the fifth interrupt and before the first data is sent in the next communication. int_tc is generated after the last data reply bit, waiting for STOP or START to be written.

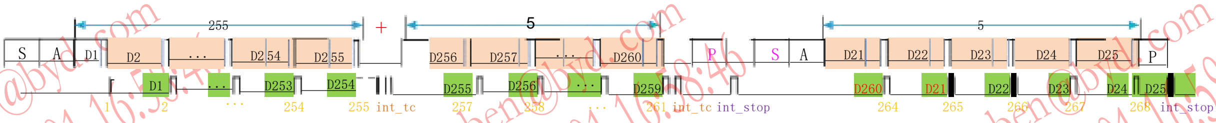
The second communication receives 5 packets of data. After entering the eighth interrupt int_rxne, D21 is not read when the D22 reception is complete, and the clock low level needs to be extended until D21 is read. A ninth interrupt, int_rxne, is then generated, reading D22. After the 12th interrupt int_rxne, the last data D25 is read. A NACK is sent after the last data is received, followed by a stop bit.

Receiving data >255 does not extend


255
+
S
A
D1
D2
D3
···
D255
256
257
···
D260
P
D1 D2 D3 ··· D255 D256 D257 ··· D260
1 2 3 ··· 255 int_tc 257 258 ··· 261 int_stop

Receive 260 data, divided into 255+5, automatically send stop bits.

Received data >255 extended


255
+
5
5
S
A
D1
D2
···
D254
D255
D256
D257
···
D260
P
S
A
D21
D22
D23
D24
D25
P
D1 ··· D253 D254 D255 D256 ··· D259
D260 D21 D22 D23 D24 D25
1 2 ··· 254 255 int_tc 257 258 ··· 261 int_tc int_stop
264 265 266 267 268 int_stop

Receive 260 data, divided into 255+5, the last data response bit generates int_tc, extend the clock low level, wait for write stop, send stop bit.

The last data D260 of the previous communication has not been read before the first data D21 of the second communication is received, and the clock line is extended until the D260 is read. The 264th interrupt, int_rxne, is then generated, reading D21.

18.3.12. Wake up from system standby mode when the address matches

The address matching interrupt INT_ADDR is generated by HCLK in normal mode and by the SCL clock when the system is in standby mode and the WUPEN position is 1 in the IIC_CR1 register. When addressed, the IIC is able to wake the MCU from system standby mode.

In the case of address matching, the MCU wake up time, the IIC extends the SCL to keep it at a low level. When the software clears the ADDR flag, the extension is released and the transfer proceeds normally. Therefore, this wake-up function must be ensured when the clock low extension function is enabled.

18.3.13. Arbitration mode is optional

Arbitration loss is detected when a high level is sent on the SDA line but a low level is sampled at the rising edge of the SCL. The slave detects quorum loss during the data phase and data answer phase, and the loss of quorum releases SCL and SDA.

When the bus SCL rises the edge, the SDA_OUT output of the host is different from the SDA_IN input of the bus SDA, and the host loses arbitration. The host is always detecting whether the arbitration is lost, and the loss of the arbitration will cause an error interrupt. If the quorum reset is enabled, the internal state machine will reset.

18.3.14. Support hardware and software reset

A software reset can be performed by zeroing the PE bit in the IIC_CR1 register. In this case, the IIC wire SCL and SDA are released, the internal state machine is reset, and the communication control bit and the status bit are restored to their reset values. The configuration register is not affected. The affected register bits are listed below:

1. IIC CR2 registers: START and STOP and NACK.
2. IIC_ISR registers: BUSY、TXE、TXIS、RXNE、ADDR、NACKF、TCR、TC、STOPF、BERR、ARLO and OVR.

Receiving a start signal, address mismatch, and quorum loss hardware automatically resets the internal timing (where the start signal reset does not include the state machine), and registers are not affected.

When quorum loss and bus errors occur, the internal state machine can be reset by placing ERR_RST_EN position 1 in the IIC_CR2 register, releasing SCL and SDA without affecting the register.

A system reset resets the entire IIC module.

18.3.15. Supports DMA read and write requests

Support DMA write request when the send data register is empty; When the receiving data register is full, a DMA read request is issued. This function requires that the function of extending the clock low level be enabled.

18.4. IIC register

Address base: 0x5000 0000

Address offset	Register	Description
0x04	RCU_EN	Peripheral module clock control register
0x38	IIC_IO_CTRL	IIC control register

Address base: 0x5004 0000

Address offset	Register	Description
0x00	IIC_CR1	IIC control register 1
0x04	IIC_CR2	IIC control register 2
0x08	IIC_OAR1	IIC itself address 1 register
0x0C	IIC_OAR2	IIC itself address 2 register
0x10	IIC_TIMINGR	IIC timing register
0x14	IIC_ISR	IIC Interrupt and status register
0x18	IIC_ICR	IIC interrupts clear register
0x1C	IIC_RXDR	IIC Receive data register
0x20	IIC_TXDR	IIC send data register

18.4.1. Peripheral module clock control register (RCU_EN)

Address offset: 0x04

Reset value: 0x0000 0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.				DIV_CLKEN	GPIOD_CLKEN	GPIOC_CLKEN	GPIOB_CLKEN	GPIOA_CLKEN	DMA_CLKEN	CRC_CLKEN	ADC_CLKEN	WDT_CLKEN	TIM7_CLKEN	TIM6_CLKEN	TIM5_CLKEN
				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
TIM4_CLKEN	TIM3_CLKEN	TIM2_CLKEN	TIM1_CLKEN	TIM0_CLKEN	PWM1_CLKEN	PWM0_CLKEN	IIC_CLKEN	UART4_CLKEN	UART3_CLKEN	UART2_CLKEN	UART1_CLKEN	UART0_CLKEN	SPI1_CLKEN	SPI0_CLKEN	Res.
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	

26	GPIOD_CLKEN	GPIOD module operation enable: 1: Work 0: Off, the default is 0
25	GPIOC_CLKEN	GPIOC module operation enable: 1: Work 0: Off, the default is 0
24	GPIOB_CLKEN	GPIOB module operation enable: 1: Work 0: Off, the default is 0
23	GPIOA_CLKEN	GPIOA module operation enable: 1: Work

		0: Off, the default is 0
22	DMA_CLKEN	DMA module was enabled: 1: Work 0: Off, the default is 0
8	IIC_CLKEN	IIC module was enabled: 1: Work 0: Off, the default is 0

18.4.2. IIC control register(IIC_IO_CTRL)

Address offset: 0x38

Reset value: 0x0000 0001

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.														DFIL_SEL	AFIL_SEL
														RW	RW

31:2	-	Reserved
1	DFIL_SEL	IIC function digital filtering enabled: 1: Enables 0: disables. Default is 0
0	AFIL_SEL	IIC function analog filtering enabled: 1: Enables 0: disables. Default is 1

18.4.3. IIC control register 1 (IIC_CR1)

Address offset: 0x00

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.												GCEN	WUPEN	NOSTRETCH	SBC
												RW	RW	RW	RW

15	14	13	12:8	7	6	5	4	3	2	1	0
RXDMAEN	TXDMAEN	IIC_SR	Res.	ERRIE	TCIE	STOPIE	NACKIE	ADDRIE	RXIE	TXIE	PE
RW	RW	RW		RW	RW	RW	RW	RW	RW	RW	RW

31:20	-	Reserved
19	GCEN	Broadcast call enable: 0: disables broadcast addresses. No answer to address 00000000B

		1: enables the broadcast address. Answer to address 00000000B
18	WUPEN	Wake up from standby mode Enable: 0: cannot wake up from the standby mode 1: Enable wake up from standby mode Note: WUPEN can set 1 only if the digital filter is invalid
17	NOSTRETCH	Clock extension prohibited: This bit is used to disable clock extension in slave mode. It must remain clear in master mode. 0: Clock extension is enabled 1: Disables clock extension This bit can only be programmed when the IIC prohibits it (PE=0)
16	SBC	Byte control in slave device mode: This bit is used to enable hardware byte control in slave device mode 0: disables byte control in secondary device mode 1: Enables byte control in secondary device mode
15	RXDMAEN	DMA Receive request Enable: 0: disables DMA from receiving requests 1: Enables DMA to receive requests
14	TXDMAEN	DMA send request Enable: 0: disables DMA from sending requests 1: enables DMA to send requests
13	IIC_SR	Conversion rate control position: 1: Conversion rate control is turned on to fit the standard speed mode 0: Conversion rate control is turned off to accommodate fast speed mode
12:8	-	Reserved
7	ERRIE	Error interrupt enable: 0: disables error detection interruption 1: Disables the error detection interrupt Any of the following errors will generate an interrupt: arbitration loss (ARLO), bus error detection (BERR), overflow/Underflow (OVR)(slave)
6	TCIE	Transfer completion interrupt enable: 0: disables transmission completion interruption 1: enables the transmission completion interrupt An interrupt is generated in either of the following situations: Transfer Complete (TC), Transfer Complete Wait Reload (TCR)
5	STOPIE	Stop bit detection interrupt enable: 0: disables stop bit detection (STOPF) 1: indicates that STOPF is enabled
4	NACKIE	Interrupt to receive a negative reply enable: 0: disables receiving a negative reply (NACKF) interrupt 1: indicates that a negative reply (NACKF) interrupt is received

3	ADDRIE	Address matching interrupt enable (only from mode) : 0: Disables the interruption of address matching (ADDR) 1: Interrupted address matching (ADDR) is enabled
2	RXIE	RX interrupt enable: 0: Disable receiving (RXNE) interrupt 1: Enables the RXNE interrupt
1	TXIE	TX interrupt Enable: 0: disables TXIS interruption 1: enables TXIS to be interrupted
0	PE	IIC peripheral enable: 0: disables peripherals 1: Enables peripherals Note: When PE=0, the SCL and SDA wires ofthe IIC are released. The internal state machine and status bits are restored to reset values

18.4.4. IIC control register 1(IIC_CR2)

Address offset: 0x04

Reset value: 0x1000 0000

31:29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	NACKEND	ERR_RST_EN	Res.	AUTOEND	RELOAD	NBYTES							
	RW	RW		RW	RW	RW							

15	14	13	12:11	10	9:8	7	6	5	4	3	2	1	0
NACK	STOP	START	Res.	RD_WRN	Res.	SADD[7:1]							Res.
RS	RS	RS		RW		RW							

31:29	-	Reserved
28	NACKEND	Receive NACK Auto End mode (main mode) : This bit is set to 1 and cleared to zero by the software 0: The transmission continues after receiving the NACK 1: Automatically sends the stop bit after receiving the NACK
27	ERR_RST_EN	Error reset enable: This bit is set to 1 and cleared to zero by the software 0: Arbitration is lost, bus error occurs and transmission continues (main mode) Transfer after arbitration loss (slave mode) 1: Reset sequence after arbitration is lost and bus occurs. Includes releasing I2C SCL and SDA lines and resetting internal state machine (main mode)
26	-	Reserved
25	AUTOEND	Auto End mode (main mode) : Set 1 and clear zero by software 0: Software end mode: When the NBYTES data transmission is complete, the

		TC flag will be set to 1, and the low level of SCL will be extended until the corresponding software operation is completed 1: Auto end mode: When the NBYTES data transmission is complete, the stop bit is automatically sent. Note: This bit does not work in slave mode, or when RELOAD is set to 1.
24	RELOAD	NBYTES overload mode: Set 1 and clear zero by software. 0: The transmission completes after the transmission of NBYTES data (followed by the stop bit or repeated start bit) 1: Transfer of NBYTES data is not completed (NBYTES will be overridden) When the NBYTES data transfer is complete, the TCR flag is set to 1, and the low level of the SCL is extended until the corresponding software operation is complete.
23:16	NBYTES	Number of bytes: Set the number of bytes to be sent/received here. In slave mode, when SBC=0, this field is irrelevant. Note: This bit cannot be changed when the START position is 1.
15	NACK	NACK generation (from mode) : Set to 1 by software and cleared by hardware when sending Nacks, receiving stop bits or matching addresses, or PE=0. 0: sends an ACK after the currently received byte 1: sends NACK after the currently received byte Note: Writing a "0" to this bit does not work. Note: This bit is used only in slave mode; In master receiver mode, no matter what the value of the NACK is, the NACK is automatically generated after the last byte (followed by a stop bit or a repeat start bit). When an overflow occurs from the receiver NOSTRETCH mode, the NACK is automatically generated regardless of the value of the NACK bit. Write 1 does not send NACK when slave mode byte control is disabled (SBC=0).
14	STOP	Stop bit generation (main mode) : Set to 1 by the software, and can be cleared by the hardware when the stop l stop is detected or when PE=0. In main mode: 0: No stop bit is generated 1: Generates a stop bit after the current byte is transferred. Note: Writing a "0" to this bit does not work.
13	START	Start bit generation: Set to 1 by the software and cleared by the hardware after sending the start bit (followed by an address sequence), in case of arbitration loss, in case of timeout error, or when PE=0. It can also be zeroed out by software by writing " 1" to the

		ADDRCF bit in the I2C_ICR1 register. 0: The start bit is not generated 1: Generate a repeat start/start bit If the IIC is already in master mode and AUTOEND=0, setting the position 1 will generate a repeat start bit after the NBYTES transfer ends and RELOAD=0. Note: Writing a "0" to this bit does not work. START position 1 even if the bus is busy or the IIC is in slave mode. When RELOAD is set to 1, this bit does not work.
12:11	-	Reserved
10	RD_WRN	Transmission direction (main mode) : 0: The main device requests write transmission 1: The main device requests read transmission Note: This bit cannot be changed when the START position is 1.
9:8	-	Reserved
7:1	SADD[7:1]	Slave address bit [7:1] (master mode) : In 7-bit addressing mode: These bits should be written to the 7-bit slave address to be sent Note: This bit cannot be changed when the START position is 1.
0	-	Reserved

18.4.5. IIC itself address 1 register(IIC_OAR1)

Address offset: 0x08

Reset value: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OA1EN	Res.							OA1[6:0]							Res.
RW								RW							

31:16	-	Reserved
15	OA1EN	Device address 1 enable: 0: disables the device's own address 1 and does not answer the received secondary address OA1 1: Enables the device to respond to the received secondary address OA1
14:8	-	Reserved
7:1	OA1[6:0]	Interface address: 7-bit addressing mode: 7-bit address
0	-	Reserved

18.4.6. IIC itself address 2 register(IIC_OAR2)

Address offset: 0x0C

Reset value: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OA2EN	Res.				OA2MSK			OA2[7:1]							Res.
RW					RW			RW							

31:16	-	Reserved
15	OA2EN	Device address 2 enable: 0: disables the device's own address 2 and does not answer the received secondary address OA2 1: enables address 2 of the device to respond to the received secondary address OA2
14:11	-	Reserved
10:8	OA2MSK	Device address 2 Mask bit 000: no mask 001: OA2[1] is masked and is an irrelevant bit. Compare only OA2[7:2] 010: OA2[2:1] is masked and is an irrelevant bit. Compare OA2 only [7:3] 011: OA2[3:1] is masked and is an irrelevant bit. Compare OA2 only [7:4] 100: OA2[4:1] is masked and is an irrelevant bit. Compare OA2 only [7:5] 101: OA2[5:1] is masked and is an irrelevant bit. Compare OA2 only [7:6] 110: OA2[6:1] is masked as an irrelevant bit. Compare only OA2[7] 111: OA2[7:1] is masked and is an irrelevant bit. All 7-bit addresses are answered without comparison. Note: These bits can only be written when OA2EN = 0.
7:1	OA2[7:1]	Interface address: 7-bit addressing mode: 7-bit address
0	-	Reserved

18.4.7. IIC timing register(IIC_TIMINGR)

Address offset: 0x10

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PRESC				Res.				SCLDEL				SDADEL			
RW								RW				RW			

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

SCLH	SCLL
RW	RW

31:28	PRESC	Timing predivision factor (master mode) : This field is used to pre-divide IIC_CLK to generate clock cycles t_{PRESC} for data build and hold counters and SCL high and low counters. t_{PRESC}=(PRESC+1)*t_{iic_clk}
27:24	-	Reserved
23:20	SCLDEL	Slave mode data highest bit and the NACK controlled creation time (slave mode) : This field is used to generate a delayed t_{SCLDEL} between the SDA edge and the SCL rising edge. If NOSTRETCH=0, the low level time of the SCL line is extended during t_{SCLDEL} when the highest bit of the data is sent or when the control NACK mode is sent to the reply bit. t_{SCLDEL}=(SCLDEL+1)*t_{PRESC}
19:16	SDADEL	Data retention time (master mode) : This field is used to generate a delayed t_{SDADEL} between the SCL falling edge and the SDA edge. t_{SDADEL}=(SDADEL+1)*t_{PRESC} Note :SDADEL is used to generate $t_{HD:DAT}$ timing.
15:8	SCLH	SCL High Level Period (Master mode): In master mode, this field is used to generate SCL high level period. t_{SCLH}=(SCLH+1)*t_{PRESC} Note :SCLH is also used to generate $t_{SU:STO}$ and t_{HD}:STA timings
7:0	SCLL	SCL Low level cycle (master mode) : In master mode, this field is used to generate SCL low level cycles t_{SCLL}=(SCLL+1)*t_{PRESC} Note: SCLL is also used to generate t_{BUF} and $t_{SU:STA}$ timings. SCLL needs to ensure that data retention time is met at the same time.

Note: This register must be configured when the IIC disables (PE=0).

18.4.8. IIC interrupt status register(IIC_ISR)

Address offset: 0x14

Reset value: 0x0000 0001

31:24	23	22	21	20	19	18	17	16
Res.	ADDCODE							DIR
	R							R

15	14:11	10	9	8	7	6	5	4	3	2	1	0
BUSY	Res.	OVR	ARLO	BERR	TCR	TC	STOPF	NACKF	ADDR	RXNE	TXIS	TXE
R		R	R	R	R	R	R	R	R	R	RS	RS

31:24	-	Reserved
23:17	ADDCODE	Address matching code (slave mode) : When an address match event occurs (ADDR= 1), these bits are updated to the received address.
16	DIR	Transmission direction (slave mode) : This flag is updated when an address match event occurs (ADDR= 1). 0: Write transfer, from the device into receiver mode 1: Read the transmission, from the device into the transmitter mode
15	BUSY	Bus busy: This flag is used to indicate that communication is taking place on the bus. When the start bit is detected, the bit is set to 1 by the hardware. When a stop bit or PE=0 is detected, this bit is cleared by the hardware.
14:11	-	Reserved
10	OVR	Upflow/Underflow (slave mode) : When in slave mode and NOSTRETCH = 1, if an overflow/underflow error occurs, this bit is set to 1 by the hardware. This flag is cleared by the software by setting OVRCF to 1. Note: When PE=0, this bit is cleared by the hardware.
9	ARLO	President lost: When a loss ofthe president occurs, the flag is set to 1 by the hardware. The flag is cleared by software by setting ARLOCF to 1. Note: When PE=0, this bit is cleared by the hardware.
8	BERR	Bus error: When a misaligned start bit or stop bit is detected and a peripheral is also involved in transmission, the flag is set to 1 by the hardware. In the address phase of slave mode, the flag is not set to 1. This flag is cleared by software by setting BERRCF to 1. Note: When PE=0, this bit is cleared by the hardware.
7	TCR	Transfer complete waiting for heavy loading: When RELOAD= 1 and the NBYTES data transfer is complete, the flag is set to 1 by the hardware. When NBYTES writes a non-zero value, the flag is cleared by the software. Note: When PE=0, this bit is cleared by the hardware. This flag is used for master mode alone and for slave mode alone at SBC position 1.
6	TC	Transfer complete (master mode) : When RELOAD=0, AUTOEND =0 and the NBYTES data transfer is complete, the flag is set to 1 by the hardware. When the START bit or STOP position is 1, the flag is cleared by the software. Note: When PE=0, this bit is cleared by the hardware.
5	STOPF	Stop bit detection flag: When a stop bit is detected on the bus and the peripheral is also involved in this transmission, the flag is set to 1 by the hardware:

		--Peripheral is used as the main device, and the position bit is provided that the peripheral has issued a stop bit. --Peripheral as a slave device, the premise of the position bit is that the address object of the transmission is the peripheral. The flag is cleared by the software by setting STOPCF to 1. Note: When PE=0, this bit is cleared by the hardware.
4	NACKF	Negative response flag received: When a NACK is received after transferring bytes, the flag is set to 1 by the hardware. This flag is cleared by software by setting NACKCF to 1. Note: When PE=0, this bit is cleared by the hardware.
3	ADDR	Address matching (slave mode) : When the received address matches one of the enabled slave device addresses, the flag is set to 1 by the hardware. This flag is cleared by the software by setting the ADDRCF to 1. Note: When PE=0, this bit is cleared by the hardware.
2	RXNE	The receiving data register is not empty (receiver) : This flag is set to 1 by the hardware when the received data has been copied to the IIC_RXDR register and is ready to be read. When IIC_RXDR is read, this bit is cleared. Note: When PE=0, this bit is cleared by the hardware.
1	TXIS	Send interrupt status (sender) : When the IIC_TXDR register is empty, the bit is set to 1 by the hardware. The data to be sent must be written to the IIC_TXDR register. When the next data to be sent is written to the IIC_TXDR register, the bit is cleared to zero. This bit can only be written to " 1" by software at NOSTRETCH = 1 to generate TXIS events (interrupt at TXIE = 1, DMA request at TXDMAEN= 1). Note: When PE=0, this bit is cleared by the hardware.
0	TXE	The send data register is empty (sender) : When the IIC_TXDR register is empty, the flag is set to 1 by the hardware. When the next data to be sent is written to the IIC_TXDR register, the bit is cleared to zero. This bit can be written " 1" by the software to refresh the send data register IIC_TXDR. Note: When PE=0, this bit is set to 1 by the hardware.

18.4.9. IIC Interrupt Clear Register (IIC ICR)

Address offset: 0x18

Reset value: 0x0000 0000

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

15:11	10	9	8	7:6	5	4	3	2:0
Res.	OVRCF	ARLOCF	BERRCF	Res.	STOPCF	NACKCF	ADDRCF	Res.
	W	W	W		W	W	W	

31:11	-	Reserved
10	OVRCF	Upflow/Underflow flag clear zero: Writing 1 to this bit clears the OVR flag in the IIC_ISR register.
9	ARLOCF	Arbitration lost flag clearing: Writing 1 to this bit clears the ARLO flag in the I2C_ISR register.
8	BERRCF	Bus error flag cleared: Writing 1 to this bit clears the BERR flag in the IIC_ISR register.
7:6	-	Reserved
5	STOPCF	Stop bit detection flag clear zero: Writing 1 to this bit clears the STOPF flag in the IIC_ISR register to zero.
4	NACKCF	Negative answer flag clear to zero: Writing 1 to this bit clears the NACKF flag in the IIC_ISR register to zero.
3	ADDRCF	Address match flag clear: Writing 1 to this bit clears the ADDR flag in the IIC_ISR register. When a 1 is written to this bit, the START bit in the IIC_CR2 register is also cleared to zero.
2:0	-	Reserved

18.4.10. IIC Received Data Register (IIC RXDR)

Address offset: 0x1C

Reset value: 0x0000 0000

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

15	14	13	12	11	10	9	8	7:0
Res.								RXDATA
								R

31:8	-	Reserved
7:0	RXDATA	8-bit received data: Bytes of data received from the IIC bus

18.4.11. IIC Transmit Data Register (IIC TXDR)

Address offset: 0x20

Reset value: 0x0000 0000

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

15	14	13	12	11	10	9	8	7:0
Res.								TXDATA
								RW

31:8	-	Reserved
7:0	TXDATA	8-bit send data: Bytes of data to be sent to the IIC bus Note: These bits can only be written when TXE= 1.

18.5. IIC configuration process

18.5.1. Process for configuring the slave IIC

18.5.1.1. Common mode configuration

1. The peripheral clock was enabled.
2. The GPIO multiplexing function needs to be configured. For details, see " GPIO multiplexing function configuration " .
3. Clear PE (peripheral enable) in IIC_CR1 to zero.
4. Enable the IIC_OAR1 and IIC_OAR2 address registers as required.
5. The data creation time SCLDEL is configured when the clock line function is extended.
6. Enable the appropriate interrupt as needed:

   When the extended clock line function is enabled, all interrupts can be enabled. If the polling function is used, the address matching interrupt function cannot be enabled. Otherwise, it must be enabled.

   If the clock line extension function is disabled, you can enable the interrupt function except the address interrupt function.
7. Configure the IIC_CR1 and IIC_CR2 registers as required, and prepare the first data in advance.
8. Set PE (peripheral enable) in IIC_CR1 to 1.

18.5.1.2. Configure the NACK/ACK control mode

1. The peripheral clock was enabled.
2. The GPIO multiplexing function needs to be configured. For details, see " GPIO multiplexing function configuration " .
3. Clear PE from IIC CR1.
4. Enable the IIC_OAR1 and IIC_OAR2 address registers as required.
5. The data creation time SCLDEL is configured when the clock line function is extended.
6. Enable the appropriate interrupt as needed:
   a) When the extended clock line function is enabled, all interrupts can be enabled. If the polling function is used, the address matching interrupt function cannot be enabled. Otherwise, it must be enabled.
   b) If the clock line extension function is disabled, you can enable the interrupt function except the address interrupt function.
7. Configure the IIC_CR1 and IIC_CR2 registers as required. You must configure NOSTRETCH = 0, SBC=1, and NBYTES! =0, RELOAD = 1. When the NBYTES data transmission is completed, the low level will be extended, the TCR flag needs to be detected, the NACK bit is written and the NBYTES is not zero to release the clock line (in fact, as long as the IIC_CR2 register is written and the NBYTES is not zero, the clock line can be released), and the first data can be prepared in advance.

8. Set PE (peripheral enable) in IIC_CR1 to 1.

18.5.1.3. DMA mode configuration

1. The peripheral clock was enabled.
2. The GPIO multiplexing function needs to be configured. For details, see " GPIO multiplexing function configuration " .
3. Configure CHNL_EN_SET to select channel x enablement, configure DMAMUX_SELx, and select the IIC_RX or IIC_TX request source.
4. Configure CHNL INT EN to enable the channel x interrupt function for DMA.
5. Configure CTRL_BASE_PTR to point to the channel data address (0x5008 0000).
6. DMA read request:
   a) Configure CHNLx_SRC_END_PTR as the IIC_RXDR(read) address (source address).
   b) Configure the CHNLx DST END PTR as the SRAM address (destination address).
7. When DMA writes a request:
   a) Configure the CHNLx SRC END PTR as the SRAM address (source address).
   b) Set the CHNLx DST END PTR to IIC TXDR(Write)(Destination Address).
8. Configure CHNLx_CONTROL to determine the DMA working mode, transmission length, source data bit width, source address address increment, destination data bit width, and destination data address increment. (See DMA register list for details).
9. Configure RXDMAEN (DMA receive request enabled), TXDMAEN (DMA receive request enabled), and NOSTRATCH = 0 (clock extension enabled) in IIC_CR1.
10. IIC related configurations are the same as in normal mode except for the interrupt enable. TXIE, RXIE, TCIE, and NACKIE do not need to be enabled. If the STOPF flag is cleared in polling mode, you can disable the STOPIE command. Otherwise, you need to enable the STOPIE forward interrupt clear flag.

The send data register is empty, and the IIC sends a DMA write request when the DMA has processed the last DMA write request.When the received data is full, and the DMA has processed the last DMA read request, the IIC needs to detect the ADDR flag and clear it to zero when sending the DMA read request. In DMA mode, the function of extending the clock line must be used, otherwise the first data is not ready in time when the slave machine sends data.

18.5.2. Process for configuring the master IIC

18.5.2.1. Non DMA mode configuration

Perform the following steps to configure the master mode:

1. The peripheral clock was enabled.
2. The GPIO multiplexing function needs to be configured. For details, see " GPIO multiplexing function configuration " .
3. Clear PE(peripheral enable) in IIC_CR1 to zero.
4. Configure digital filter DFIL SEL and analog filter AFIL SEL.
5. Configure SDADEL[3:0] SCLH[7:0] and SCLL[7:0] in IIC_TIMINGR.
6. Set PE bit (peripheral enabled) in IIC CR1 to 1.
7. To be sent from address: SADD[7:1]; Transmission direction: RD_WRN; Number of bytes to be transmitted: NBYTES[7:0]; And other enable control bits.
   a) NBYTES[7:0] has a maximum size of 255, so if you want to transfer data greater than 255, you must use a combination of overloaded and non-overloaded modes. When the data to be transferred is less than or equal to 255, only the non-overloaded mode can be used.
   b) If the number of bytes to be transmitted is less than or equal to 255, RELOAD=0 must be configured. Configure AUTOEND=1 to automatically generate the stop bit after the transmission ends. Configure AUTOEND=0 to detect the TC flag after the end of the transfer, so that you can configure restart or STOP. (You can reconfigure the above configuration before restarting the configuration.) When the number of bytes to be transferred is equal to 255, NBYTES[7:0] must be filled with 0xFF during initialization.
   c) When the number of bytes to be transferred is greater than 255, RELOAD=1 must be configured, and NBYTES[7:0] must be filled with 0xFF during initialization. After 255 data is transmitted, TCR is detected, and if the remaining number of data to be transmitted is still greater than 255, NBYTES[7:0] must be filled with 0xFF again. Until the remaining number of transfers is less than or equal to 255, and TCR is detected, configure RELOAD=0, and configure AUTOEND and NBYTES[7:0].
8. Detects that the BUSY flag is 0 and puts the START position 1 in the IIC_CR2 register. When the START position is 1, you are not allowed to change all the bits in step 7. TXDATA[7:0] can be configured in step 7 or after START position 1.
9. When the bus is detected to be idle, it automatically sends the start bit after a tBUF delay, followed by the slave device address.

18.5.2.2. DMA mode configuration

1. The peripheral clock was enabled.
2. The GPIO multiplexing function needs to be configured. For details, see " GPIO multiplexing function configuration " .

3. Configure CHNL_EN_SET to enable channel x, configure DMAMUX_SELx, and select the IIC_RX or IIC_TX request source.
4. Configure CHNL INT EN to enable the channel x interrupt function for DMA.
5. Configure CTRL_BASE_PTR to point to the channel data address (0x5008_0000).
6. DMA read request:

   Configure CHNLx_SRC_END_PTR as the IIC_RXDR(read) address (source address).

   Configure the CHNLx DST END PTR as the SRAM address (destination address).
7. MA writes a request:

   Configure CHNLx_SRC_END_PTR as the SRAM address (source address).

   Set CHNLx_DST_END_PTR to IIC_TXDR(write)(destination address).
8. Configure CHNLx_CONTROL to determine the DMA working mode, transmission length, source data bit width, source address address increment, destination data bit width, and destination address address increment. (See DMA register list for details)
9. Configure RXDMAEN(DMA Receive request Enable) in IIC_CR1 TXDMAEN(DMA Receive request Enable).
10. Except for the interrupt function, IIC related configurations are the same as those in common modes. TXIE RXIE TCIE NACKIE and STOPIE do not need to be enabled.

The send data is empty, and the DMA write request is issued when the DMA has processed the last DMA write request. A DMA read request is issued when the received data is full and the DMA has processed the last DMA read request.

In the end mode of non-overloaded software, it is necessary to detect the TC flag and release the clock line at the start bit or stop position 1 for subsequent transmission.

In heavy load mode, the TCR flag must be detected, and the clock line must be written to the IIC_CR2 register when the NBYTES value is not zero (it is recommended that the NBYTES value be non-zero) to release the clock line for subsequent transmission.

18.6. Note

1. During initialization, check whether the BUSY flag is 1 before the START position 1. If the busy flag is 1, it indicates that the IIC bus is not idle. If the START position 1 is set, the start bit may not be sent successfully.
2. When RELOAD=1 is configured, that is, NBYTES[7:0] must be filled with 0xFF and 255 data must be sent. Otherwise, the clock is extended to wait for new data to be written until 255 data are sent. Set the TCR flag to 1, and then perform subsequent operations.
3. If the sending mode stops and starts again, write the START bit when the STOPF flag is detected. Otherwise, the stop bit may not be sent successfully.
4. If NBYTES[7:0] is filled with 0, the stop bit is automatically sent after the address is sent.
5. Slave mode The last configuration of the operation to release the clock line in all extended clock low situations.
6. The time from the fall edge of the ninth clock to the highest bit ofthe data is a fixed 4*t_{HCLK}, so when configuring the system clock, it should meet 4*t_{HCLK}<t_{SCLL}+ (analog filtering/digital filtering).

Chapter 19 Pulse width modulation module (PWM)

19.1. PWM overview

The BF7707AMXX contains two separate PWM modules (PWM0 and PWM1). PWM module clock Optional clock source PLL48M/LIRC 32K/XTAL, and then clock frequency division.

19.2. PWM functional characteristics

- 16-bit counter with automatic overload counting.
- Support idle mode 0 interrupt wake up (interrupt enable).
  The clock source and clock crossover of the PWM module are configured by PWM_CLK_SEL: PLL48M/LIRC 32K/XTAL (Passive crystal oscillator (32768Hz/4MHz/8MHz) and active crystal oscillator (1MHz~48MHz)), 2^PWM_DIV_CFG[3:0](2^0~2^15) clock frequency division optional.
- PWM0 supports one channel and three mapping interfaces, PWM0A, PWM0B, or PWM0C.
- PWM1 supports one channel and three mapping interfaces, PWM1A, PWM1B, or PWM1C.
- Configure the output PWM waveform.
  PWM waveform formula:
  Output cycle: T_{pwm_data} = (PWM_H+PWM_L)*T_{pwm_clk}
  Output duty cycle: D_{pwm_data} = PWM_H/(PWM_L+PWM_H)
- Supported timer mode.
  PWM timing period formula:
  T_{timer} = (PWM_H + PWM_L)*T_{pwm_clk} (Note: neither PWM_H nor PWM_L can be zero.)
- All I/O port output can be configured independently.

19.3. PWM functional description

An interrupt flag is set when the count overflows, and an interrupt response is triggered if the interrupt is configured to enable it.

The PWM0/1 pulse width modulation module can be configured with registers for both high and low level times. Once the configuration value is updated after the count starts, it is updated after a full cycle, with caching processing in between.

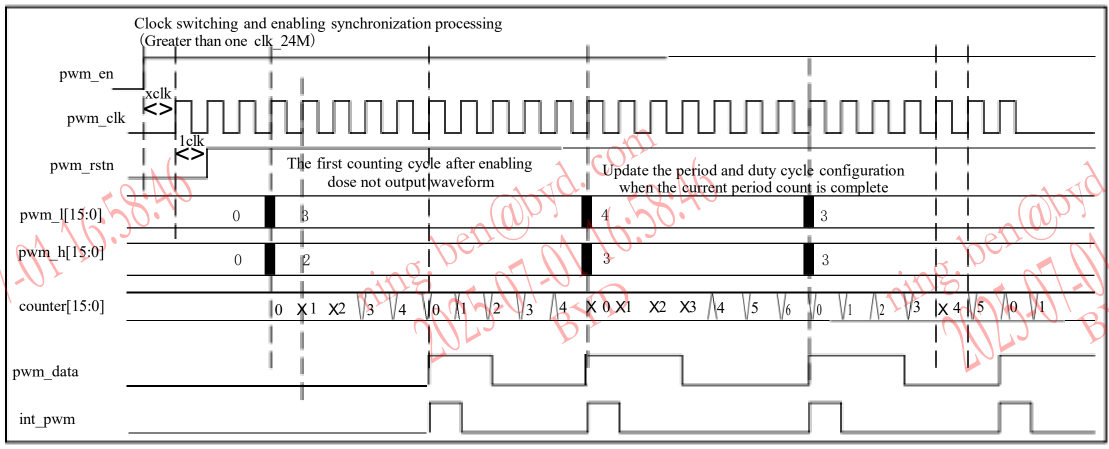

Clock switching and enabling synchronization processing
(Greater than one clk_24M)
pwm_en
xclk
pwm_clk
1clk
pwm_rstn
The first counting cycle after enabling
dose not output waveform
Update the period and duty cycle configuration
when the current period count is complete
pwm_l[15:0]
0
3
4
3
pwm_h[15:0]
0
2
3
3
counter[15:0]
0 1 2 3 4 0 1 2 3 4 0 1 2 3 4 5 6 0 1 2 3 4 5 0 1
pwm_data
int_pwm

Figure 19.1 PWM0/1 Timing diagram

The PWM normal output pulse waveform needs to be multiplexed into PWM function by configuring the Px_MODR and Px_AFSEL0/1 registers of GPIO. When the PWM multiplexing function of GPIO is disabled, PWM can be used as a timer.

19.4. PWM register

Base address: 0x5000 0000

Address offset	Register	Description
0x04	RCU_EN	Peripheral module clock control register
0x48	CLK_SEL_RDY	Clock switch completion flag register
0x4C	PWM_CLK_SEL	PWM0/1 clock configuration register

Base address: PWM0: 0x5005 0100 PWM1: 0x5005 0200

PWM0/1 register has the same function, x=0/1

Address offset	Register	Description
0x00	PWM_TIME	PWM0/1 level control register
0x04	PWM_CFG	PWM0/1 enables the register
0x08	PWM_INT_CFG	PWM0/1 interrupt control register

19.4.1. Peripheral module clock control register (RCU_EN)

Address offset: 0x04

Reset value: 0x0000 0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.				DIV_ CLKEN	GPIOD_ CLKEN	GPIOC_ CLKEN	GPIOB_ CLKEN	GPIOA_ CLKEN	DMA_ CLKEN	CRC_ CLKEN	ADC_ CLKEN	WDT_ CLKEN	TIM7_ CLKEN	TIM6_ CLKEN	TIM5_ CLKEN
				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
TIM4_ CLKEN	TIM3_ CLKEN	TIM2_ CLKEN	TIM1_ CLKEN	TIM0_ CLKEN	PWM1_ CLKEN	PWM0_ CLKEN	IIC_ CLKEN	UART4_ CLKEN	UART3_ CLKEN	UART2_ CLKEN	UART1_ CLKEN	UART0_ CLKEN	SPI1_ CLKEN	SPI0_ CLKEN	Res.
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	

26	GPIOD_CLKEN	GPIOD module operation enable 1: Work 0: Off, the default is 0
25	GPIOC_CLKEN	GPIOC module operation enable 1: Work 0: Off, the default is 0
24	GPIOB_CLKEN	GPIOB module operation enable 1: Work 0: Off, the default is 0
23	GPIOA_CLKEN	GPIOA module operation enable 1: Work 0: Off, the default is 0
10	PWM1_CLKEN	PWM1 module operation enable 1: Work 0: Off, the default is 0

9	PWM0_CLKEN	PWM0 module operation enable 1: Work 0: Off, the default is 0

19.4.2. Analog module switch register (ANA_CFG)

Address offset: 0x10

Reset value: 0x0000 1850

15:13	12	11	10	9	8:7	6	5	4	3 2	1:0
Res.	PD_TEMP	PD_ADC	VREF_SEL	VREF_VOL_SEL	VREF_IN_ADC_SEL	PD_ADC_IN_VREF	XTAL_HFR_SEL	XTAL_SEL	XTAL_BYP_SEL	XTAL_EN_SEL
	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

5	XTAL_HFR_SEL	Analog high frequency crystal oscillator circuit current configuration selection: 0:175µA (4MHz crystal oscillator recommended) 1:300µA (8MHz crystal oscillator recommended)
4	XTAL_SEL	When a passive crystal oscillator circuit is available 0: Select 32 KB 1: Select 4M/8M. Default value 1
3:2	XTAL_BYP_SEL	Active crystal oscillator path selection: 00: Channel closed, do not use active crystal oscillator 01: Select channel A (PA5) for input 10: Select channel B (PA2) for input 11: Reserved Note: Select any active crystal path, and the passive crystal vibration circuit is turned off at the same time. The default value is 00.
1:0	XTAL_EN_SEL	Passive crystal path selection: 00: The passive crystal vibration circuit is turned off and no passive crystal vibration is used 01: Select path A (PA5, PA4) 10: Select channel B (PA2, PA1) for input 11: Reservations When XTAL_BYP_SEL[1:0] = 00 and XTAL_EN_SEL[1:0]! = 00, turn on the passive crystal vibration circuit, the default value is 00 Note: Passive crystal oscillator must be enabled when active crystal oscillator is not enabled

19.4.3. Clock switch completion flag register (CLK_SEL_RDY)

Address offset: 0x48

Reset value: 0x0000 03FF

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

15:10	9	8	7	6	5	4	3	2	1	0
Res.	TIM7RDY	TIM6RDY	TIM5RDY	TIM4RDY	TIM3RDY	TIM2RDY	TIM1RDY	TIM0RDY	PWM1RDY	PWM0RDY
	R	R	R	R	R	R	R	R	R	R

1	PWM1RDY	PWM1 Clock switchover complete sign: 0: The clock is being switched. 1: The clock switchover is complete
0	PWM0RDY	PWM0 Clock switchover complete sign: 0: The clock is being switched. 1: The clock switchover is complete

19.4.4. PWM0/1 Clock configuration register (PWM_CLK_SEL)

Address offset: 0x4C

Reset value: 0x0000 0000

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

15:12	11:10	9:8	7:4	3:0
Res.	PWM1SEL	PWM0SEL	PWM1DIV	PWM0DIV
	RW	RW	RW	RW

31:12	-	Reserved
11:10	PWM1SEL	PWM1 Clock source Selection: 00: Select the PLL48M clock 01: Select the LIRC 32K clock 1x: Select XTAL (passive oscillator (32768Hz/4MHz/8MHz) and active oscillator (1MHz~48MHz)
9:8	PWM0SEL	PWM0 Clock source Selection: 00: Select the PLL48M clock 01: Select the LIRC 32K clock 1x: Select XTAL (passive oscillator (32768Hz/4MHz/8MHz) and active oscillator (1MHz~48MHz)
7:4	PWM1DIV	PWM1 Clock Frequency division selection: Clock division selection register for PWM1 clock source selection 0~ 15:2 ^(0~15) Frequency division
3:0	PWM0DIV	PPWM0 Clock frequency division selection:

		Clock division selection register for PWM0 clock source selection 0~ 15:2 ^(0~15) Frequency division

19.4.5. PWM level control register(PWM_TIME)

Address offset: 0x00

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PWM_H[15:0]															
RW															

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

31:16	PWM_H[15:0]	PWM high level control register
15:0	PWM_L[15:0]	PWM low level control register

19.4.6. PWM module enable register(PWM_CFG)

Address offset: 0x04

Reset value: 0x0000 0000

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

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

31:1	-	Reserved
0	PWM_EN	PWM module enable: 1: PWM function is enabled 0: PWM function disabled

19.4.7. PWM interrupt control register(PWM_INT_CFG)

Address offset: 0x08

Reset value: 0x0000 0000

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

15:3	2	1	0

Res.	INT_CLR	INT_FLAG	INT_EN
	W	R	RW

31:3	-	Reserved
2	INT_CLR	Interrupt status flag clear register: Write 1 to clear INT_FLAG
1	INT_FLAG	Interrupt status flag register: 1: End of counting period 0: Count incomplete Note: System reset can clear the flag bit; Write 1 to clear the INT_CLR flag bit.
0	INT_EN	Interrupt enable register: 1: Interrupt enable 0: Interrupt disabled (used in polling mode)

19.5. PWM configuration process

1. The peripheral clock was enabled.
2. Configure the GPIO multiplexing function to the PWM function. For details, see " GPIO multiplexing function configuration " .
3. Select a peripheral clock source and configure clock frequency division.
4. Configuration register PWM_TIME.
5. Clears the PWM interrupt flag bit.
6. Configure the interrupt function for PWM interrupt.
7. Indicates the module that enables PWM.
8. Gets the interrupt flag in the interrupt service function and clears it.

Note: The recommended PWM frequency is shown in the following table.

PWM clock source(Hz)	Clock frequency division	Output frequency(Hz)	Rank
PLL48M	48M	750Hz~480K	64000~100
PLL48M	24M	375Hz~240K	64000~100
PLL48M	12M	192Hz~120K	62500~100
PLL48M	6M	96Hz~60K	62500~100
PLL48M	3M	48Hz~30K	62500~100
PLL48M	1.5M	24Hz~15K	62500~100
PLL48M	0.75M	12Hz~7.5K	62500~100
PLL48M	0.375M	6Hz~3.75K	62500~100
LIRC 32K	32K	1Hz~320Hz	32768~100
XTAL 32768Hz	32768Hz	1Hz~320Hz	32768~100
XTAL 4M	4M	64Hz~40K	62500~100
XTAL 8M	8M	125Hz~80K	64000~100

19.6. Note

1. Start working after PWM_EN is enabled (high effective), first turn on counter gated clock, module reset release, then start counting, the counter will automatically reload until configuration enable is off (PWM_EN=0).
2. The first count cycle of PWM0/1 does not output a waveform, but will output a low level for one cycle.
3. The I/O multiplexing function must be configured before the PWM function is enabled.
4. During PWM output, it is not allowed to change the clock frequency, otherwise the output will be wrong when changing.

Chapter 20 Low voltage detection (LVDT)

20.1. LVDT overview

BF7707AMXX series supports low voltage alarm function, which can effectively monitor the dynamic voltage changes.Voltage at the same time support 7 gear, respectively is: 2.7V/ 3.0V/ 3.3V / 3.6V/ 3.9V/ 4.2V/ 4.5V(step-down preset point, produce corresponding hysteresis 0.1 V booster interrupt).When the voltage monitoring is configured with the preceding threshold, the voltage drops to this threshold, triggering a low voltage interruption. The system can handle the low voltage interruption properly based on application requirements.

20.2. LVDT register

Base address: 0x5000 0000

Address offset	Register	Description
0x1C	LVDT_CTRL	LVDT control register
0x20	LVDT_STATE	LVDT status register

20.2.1. LVDT control register(LVDT_CTRL)

Address offset: 0x1C

Reset value: 0x0000 0004

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

15:8	7 6 5	4 3	2	1	0
Res.	VTH_SEL	DELAY_SEL	PD_LVDT	PO_INTEN	BO_INTEN
	RW	RW	RW	RW	RW

31:8	-	Reserved
7:5	VTH_SEL	LVDT threshold selection: 000: Reserved, do not use 001: 2.7V 010: 3.0V 011: 3.3V 100: 3.6V 101: 3.9V 110: 4.2V 111: 4.5V For the specific corresponding threshold voltage, see the table "Threshold and delay selection"
4:3	DELAY_SEL	LVDT its low voltage detection delay(power-down delay) configuration: 00: Power-down delay 1 01: Power-down delay 2 10: Power-down delay 3

		11: Power-down delay 4
2	PD_LVDT	LVDT control register: 1: Off 0: On, offby default
1	PO_INTEN	LVDT low voltage boost interrupt enable: 1: Enable 0: Disable
0	BO_INTEN	LVDT low voltage step-down interrupt enable: 1: Enable 0: Disable

20.2.2. LVDT status register(LVDT_STATE)

Address offset: 0x20

Reset value: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.														POF	BOF
														RC_W1	RC_W1

31:2	-	Reserved
1	POF	LVDT boost interrupt flag: 1: Effective 0: Invalid
0	BOF	LVDT step-down interrupt flag: 1: Effective 0: Invalid

20.3. Threshold and delay selection

Threshold selection (VTH_SEL)	Delay selection (DELAY_SEL)	LVDT			
		Power down threshold (V)	Recovery threshold (V)	Hysteresis (mV)	Delay (µs)
001	00	2.7	2.8	103.3	7
	01				14
	10				29
	11				57
010	00	3.0	3.1	93.0	8
	01				16
	10				32
	11				63
011	00	3.3	3.4	113.3	9
	01				17
	10				34
	11				68
100	00	3.6	3.7	83.3	9
	01				18
	10				36
	11				73
101	00	3.9	4.0	101.0	10
	01				19
	10				38
	11				77
110	00	4.2	4.3	89.0	10
	01				20
	10				40
	11				80
111	00	4.5	4.6	54.3	16
	01				21
	10				42
	11				83

Table 20.1 LVDT threshold and delay selection

Note:

1. When the current voltage is interrupted by power failure at the threshold voltage point, the voltage will recover with a certain voltage lag.
2. The system responds to the threshold voltage signal with a time delay. If the voltage response time is shorter than the delay time, no interrupt response is generated.

20.4. LVDT Configuration Process

1. Set the LVDT threshold voltage VTH_SEL and power-off delay DALAY_SEL.
2. Clear the LVDT interrupt flag.
3. Enable LVDT interrupt.
4. Enable the LVDT module.
5. Obtain the interrupt flag bit in the interrupt service function and clear it.

Chapter 21 Cyclic redundancy check module (CRC)

21.1. CRC overview

The cyclic redundancy check module uses a polynomial generator to generate a CRC code from an 8-bit/16-bit/32-bit data.

BF7707AMXX adopts look-up table method to realize CRC calculation.The CRC calculation unit has a single 32-bit read/write data register (CRC_DR), which is used to input new data (write access) and save the result of the previous CRC calculation (read access). Each write to the data register performs a CRC calculation on the previous CRC value and the new value.CRC calculations are done on the entire 32-bit data word or byte by byte, depending on the format in which the data is written (by word, right justified half word, and aligned byte access).

21.2. CRC functional characteristics

- Support 3 kinds ofCRC polynomials
  - Use CRC-32 by default: 0x04C11DB7
    Polynomial: $x^{32}+x^{26}+x^{23}+x^{22}+x^{16}+x^{12}+x^{11}+x^{10}+x^{8}+x^{7}+x^{5}+x^{4}+x^{2}+x+1$
  - CRC-16: 0x1021
    Polynomial: $x^{16}+x^{12}+x^{5}+1$
  - CRC-8: 0x07
    Polynomial: $x^{8}+x^{2}+x+1$
- Handling 8-bit, 16-bit, 32-bit data size
  - 32-bit data size, CRC calculation is completed within 4 HCLK clock cycles
  - 16-bit data size, CRC calculation is completed within 2 HCLK clock cycles
  - 8-bit data size, CRC calculation is completed within 1 HCLK clock cycle
- The input data can be reversed bit by byte, half word, and word
- The result can be reversed bit by bit
- The initial value and xor value can be configured

21.3. CRC function description

21.3.1. CRC structure block diagram

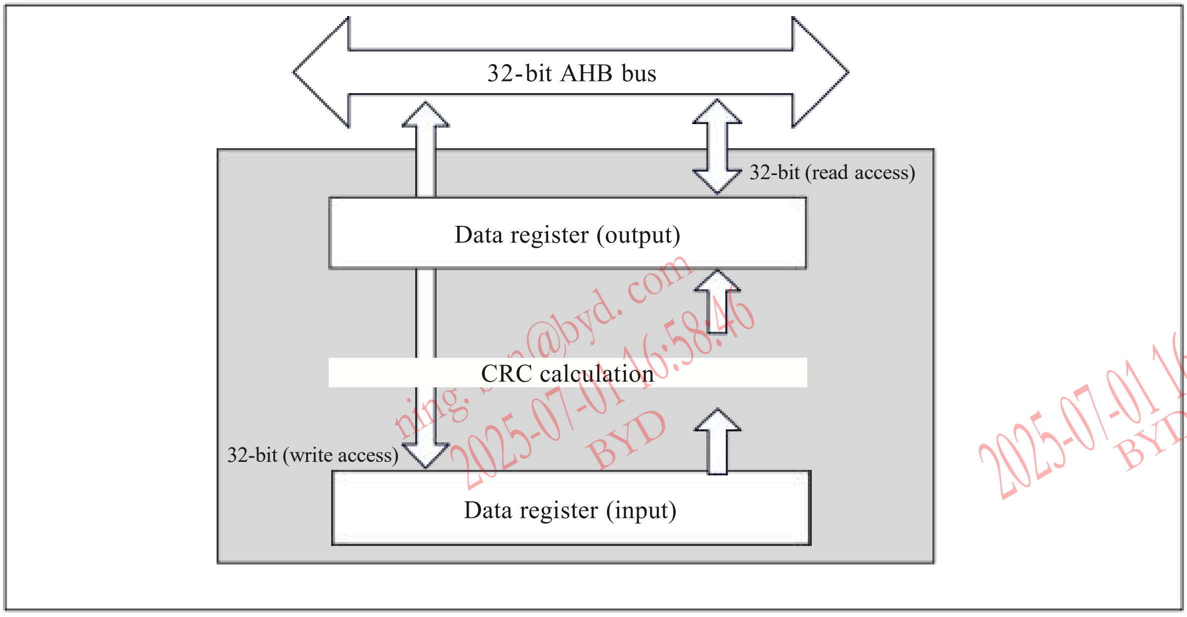


Figure 21.1 CRC structure block diagram

21.3.2. CRC data format conversion

Register operations can be performed in byte, half word and word formats. Only related to CRC_POL_SEL when reading.

Data format conversion description: Write data

	REV_IN=0	REV_IN= 1	REV_IN=2	REV_IN=3
8-bit write	Does not affect bit order	Byte-wise bit-reversal	Byte-wise bit-reversal	Byte-wise bit-reversal
16-bit write	Does not affect bit order	Byte-wise bit-reversal	Perform bit reversal by half word	Perform bit reversal by half word
32-bit write	Does not affect bit order	Byte-wise bit-reversal	Perform bit reversal by half word	Perform bit reversal by word

Data format conversion description: Read data

CRC-8		CRC-16		CRC-32	
REV_OUT=0	REV_OUT= 1	REV_OUT=0	REV_OUT= 1	REV_OUT=0	REV_OUT= 1
Does not affect bit order	Byte-wise bit-reversal	Does not affect bit order	Perform bit reversal by half word	Does not affect bit order	Perform bit reversal by word

21.4. CRC registers

Base address: 0x5000 0000

Address offset	Register	Description
0x04	RCU_EN	Peripheral module clock control register

Base address: 0x5000_ 1000

Address offset	Register	Description
0x00	CRC_DR	CRC data register
0x04	CRC_CR	CRC control register
0x08	CRC_POL_SEL	CRC polynomial selection
0x0C	CRC_INIT	CRC initial value
0x10	CRC_XOR	CRC xor value

21.4.1. Peripheral module clock control register (RCU_EN)

Address offset: 0x04

Reset value: 0x0000 0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.				DIV_ CLKEN	GPIOD_ CLKEN	GPIOC_ CLKEN	GPIOB_ CLKEN	GPIOA_ CLKEN	DMA_ CLKEN	CRC_ CLKEN	ADC_ CLKEN	WDT_ CLKEN	TIM7_ CLKEN	TIM6_ CLKEN	TIM5_ CLKEN
				RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

21	CRC_CLKEN	CRC module operation enable: 1: Work 0: Off the default is 0

21.4.2. CRC data register (CRC_DR)

Address offset: 0x00

Reset value: 0xFFFF FFFF

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

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

31:0	DR[31:0]	This register is used to write new data to the CRC cell: Reading this register can read the results of previous CRC calculations Note: If the data bit size is less than 32 bits, the low byte bits can be effectively read and written in byte, half word, or word format

21.4.3. CRC control register (CRC_CR)

Address offset: 0x04

Reset value: 0x0000 0000

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

15:5	4	3	2	1	0
Res.	CF	REV_OUT	REV_IN		RESET
	R	RW	RW		RS

31:5	-	Reserved
4	CF	CRC calculation flag bits: When the data register is written, the hardware pulls the flag bit higher, and the hardware clears the flag after the calculation is complete.When the flag is high, no new data can be written.
3	REV_OUT	Used to control the reversal of the output data bit order: 0: Does not affect the bit order 1: Bit-reversed output format
2:1	REV_IN	Used to control the reversal of the input data bit order: 00: Does not affect the bit order 01: Perform bit reversal by byte 10: Perform bit reversal by half word 11: Perform bit reversal by word: Note: If the input data is written in half-word format, the REV_IN format is configured to 11 Perform bit inversion by word, and the input data is performed by half-word inversion. If the input data is written in byte format, the REV_IN format is configured to 11 when bit inversion is performed by word and 10 when bit inversion is performed by half word, and the input data is performed by byte inversion.
0	RESET	Set by software, clear by hardware Used to reset the CRC module

21.4.4. CRC polynomial selection (CRC_POL_SEL)

Address offset: 0x08

Reset value: 0x0000 0000

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

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

	RW

31:2	-	Reserved
1:0	POL_SEL	For selecting polynomials: 00: 32-bit polynomial, 0x04C11DB7 01: 16-bit polynomial, 0x1021 10: 8-bit polynomial, 0x07 11: Reserved

21.4.5. CRC initial value (CRC_INIT)

Address offset: 0x0C
Reset value: 0xFFFF FFFF

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

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

31:0	INIT	CRC initial value When 8/16-bit polynomial is selected, the lower bits are valid Initial value cannot be configured during CRC calculation

21.4.6. CRC xor value (CRC_XOR)

Address offset: 0x10
Reset value: 0x0000 0000

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

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

31:0	XOR	CRC XOR value: When 8/16-bit polynomial is selected, the lower bits are valid XOR value cannot be configured during CRC calculation

21.5. CRC Configuration process

1. Configure and enable the CRC clock.
2. Configure the relevant registers (polynomial selection/initial value/XOR value/bit inversion).
3. Set the RESET bit to reset the compute unit.
4. After CF is read as 0, write the data register CRC_DR to calculate the CRC check result.
5. When CF is 0, the CRC check result can be read after one system clock period is delayed.

21.6. Note

1. The configuration process must be configured in sequence.
2. When dealing with 8-bit /16-bit data size, the data register CRC_DR should be converted to 8-bit /16-bit.
3. If the RESET bit ofthe CRC_CR register is set to 1, the CRC computing unit is reset.The RESET bit clears hardware and the pulse width is one system clock period.
4. The polynomial selection register, the XOR value register, and the bit reversal control register cannot be configured in real time during CRC calculation.

Chapter 22 Programming and debugging

22.1. SWD debug interface

The BF7707AMXX series uses two pins, as shown in the following table.PA6/PA7 is the default function port for SWD download and debugging.When SW communication is configured, the internal pull-up resistance of PA6 or PA7 is enabled by default.

Note: If PA6 or PA7 is configured as a common I/O port, the SWD function is affected. You are advised not to use PA6 or PA7 as common I/O or other functions.

Pin name	Description	Corresponding pin
SWCLK	Clock signal	PA6
SWDIO	Data input/output	PA7

22.2. SWD Register

PA base address: 0x5000 2000

Address offset	Register	Description
0x00	PA_MODR	PA port mode register
0x80	PA_AFSEL0	PA port multiplexing function register 0

22.2.1. PA port mode register (PA_MODR)

Address offset: 0x10

Reset value: 0x0000 A000

31 30	29 28	27 26	25 24	23 22	21 20	19 18	17 16
MODE15	MODE14	MODE13	MODE12	MODE11	MODE10	MODE9	MODE8
RW	RW	RW	RW	RW	RW	RW	RW

15 14	13 12	11 10	9 8	7 6	5 4	3 2	1 0
MODE7	MODE6	MODE5	MODE4	MODE3	MODE2	MODE1	MODE0
RW	RW	RW	RW	RW	RW	RW	RW

31:0	MODEy	Port mode register: The bit width [31:0] corresponds to 16 pins, and each pin features 2 bits to define the pattern. MODEy[1:0] : Port configuration I/O pin y (y = 0 to 15) These bits are written by software to configure the I/O mode. 00: indicates the input mode 01: General output mode 10: Reuse function mode

		11: Simulation mode Note: Locked after entering SW mode, the simulation mode configuration is invalid.

22.2.2. PA port multiplexing function register 0 (PA_AFSEL0)

Address offset: 0x80

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AFSEL7				AFSEL6				AFSEL5				AFSEL4			
RW				RW				RW				RW			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AFSEL3				AFSEL2				AFSEL1				AFSEL0			
RW				RW				RW				RW			

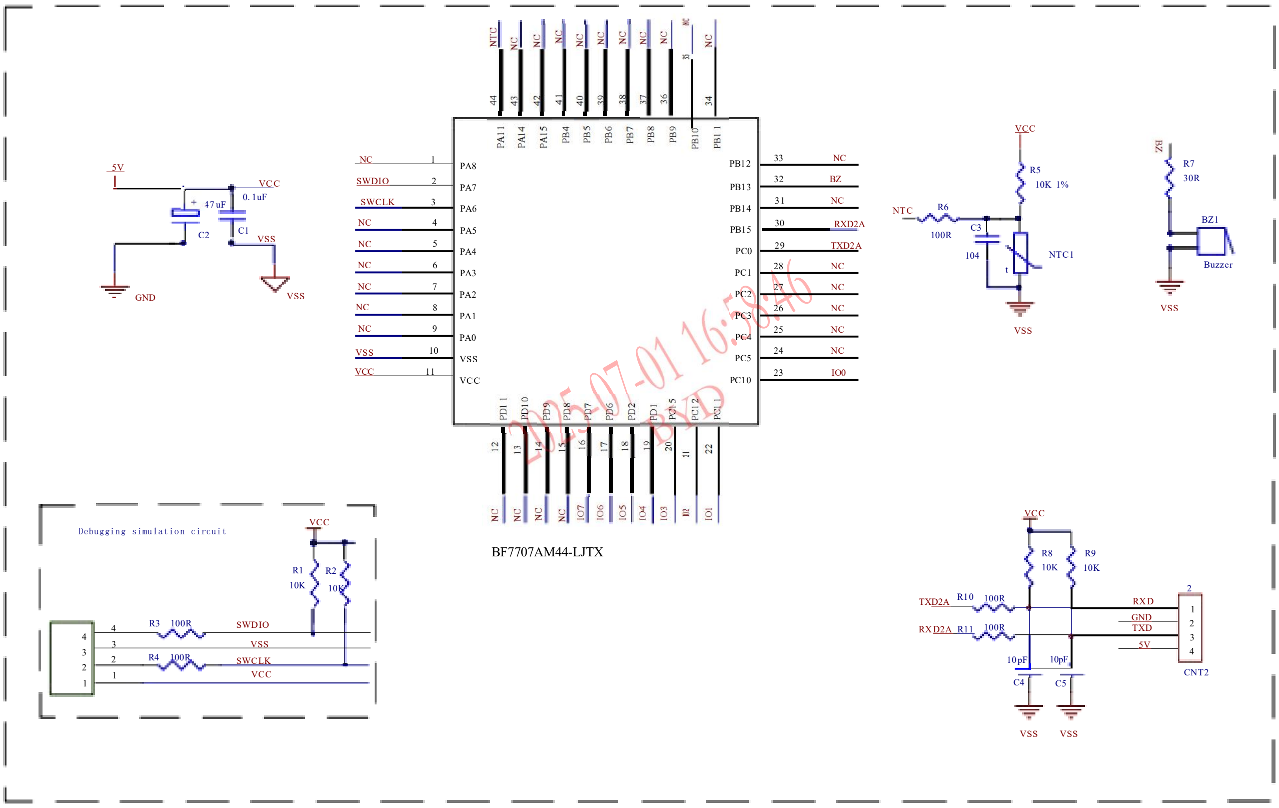
31:0	AFSELy	port multiplexing function register 0: The bit width [31:0] corresponds to 8 pins, and each pin function defines the pattern with 4 bits. AFSELy[3:0] : Reuse function selection of PA port pin y (y = 0~7) These bits are written by software to configure multiplexed function I/O. AFSELy choose: 0000: AF0 0001: AF1 0010: AF2 0011: AF3 Other: Reserved For the port multiplexing function, see "GPIO multiplexing function configuration".

22.3. PGC, PGD burning

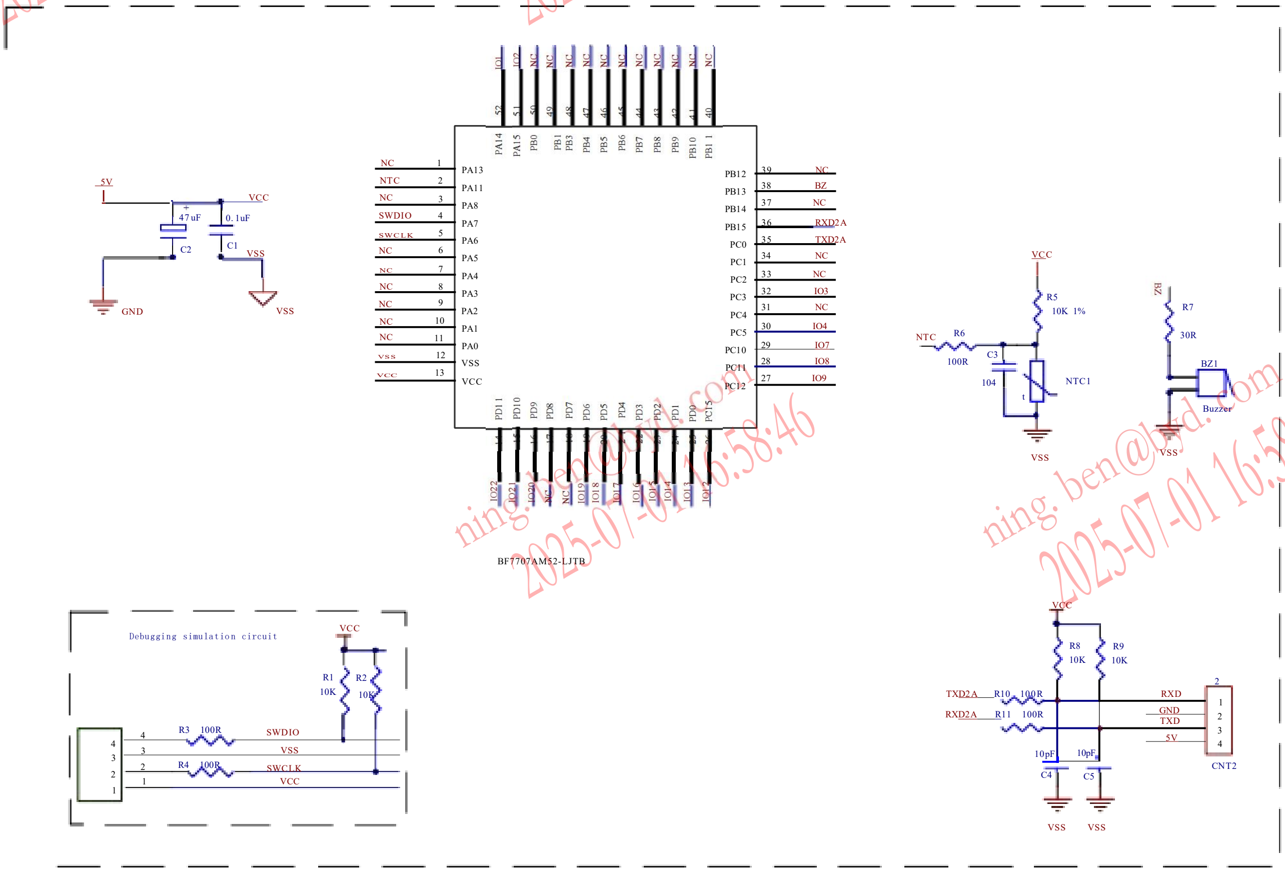
The BF7707AMXX supports PGCA (PGCB/PGCC), PGDA (PGDB/PGDC) programming, using our dedicated MP100 programming tool. For details, see "BYD MP100 user manual for mass production and burning tool".

Chapter 23 Reference application circuit

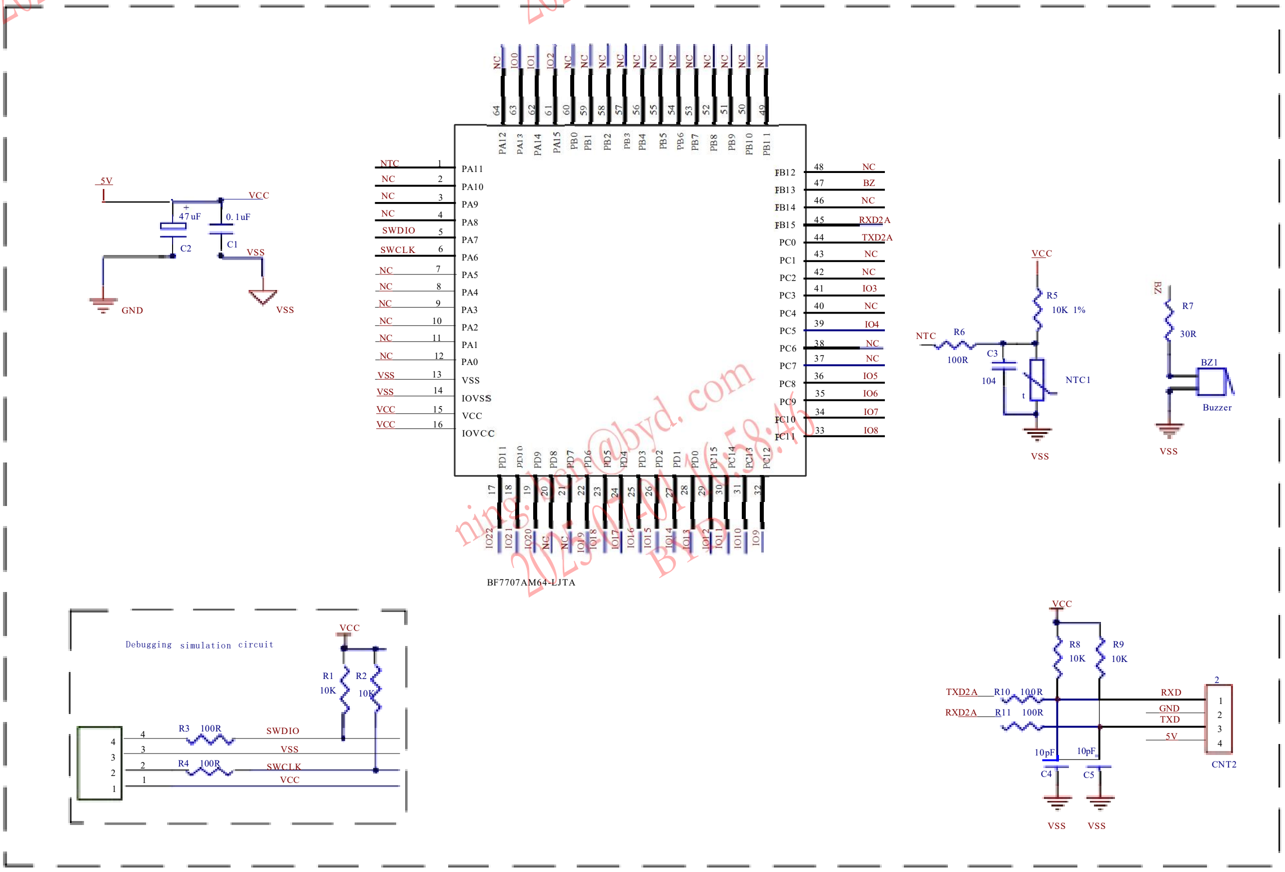
23.1. BF7707AM44-LJTX reference circuit



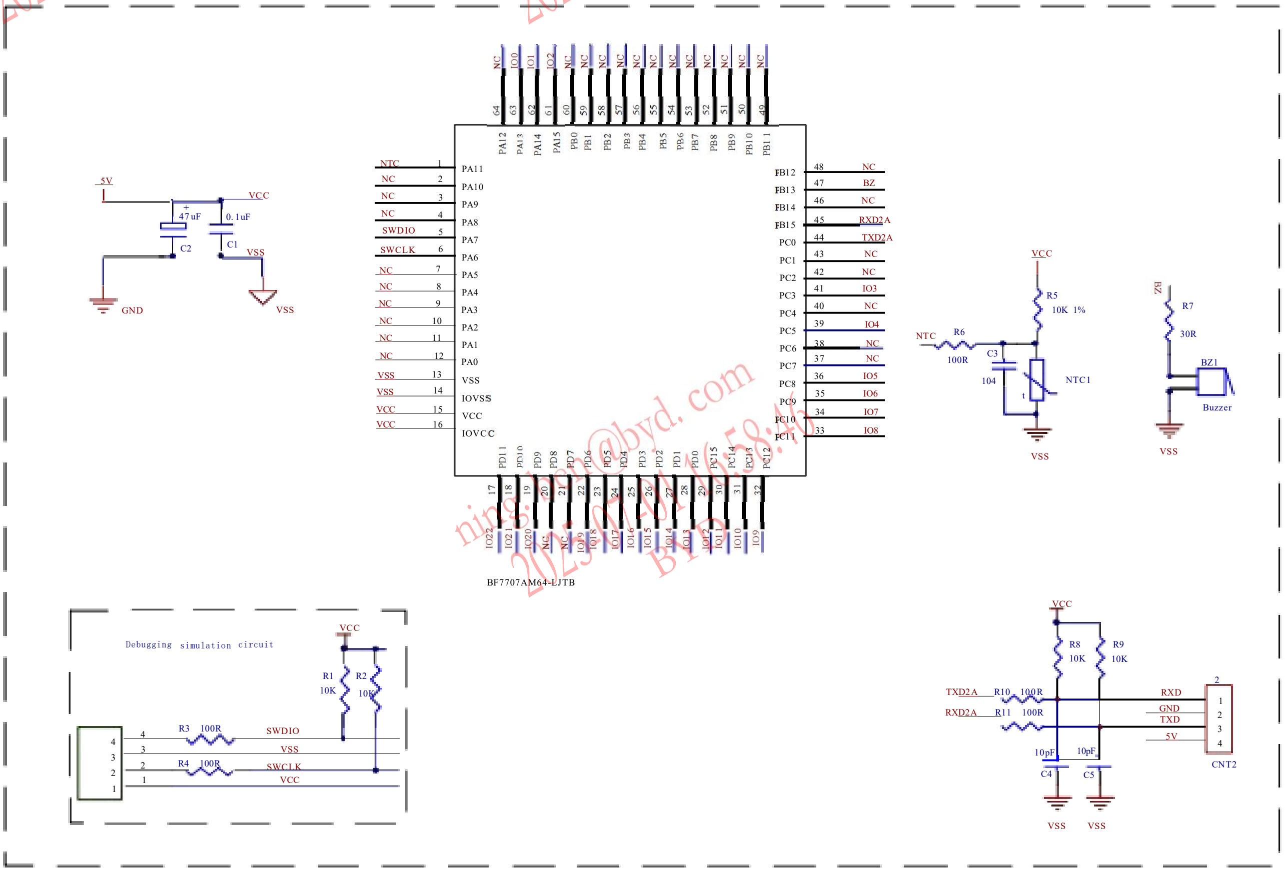
23.2. BF7707AM52-LJTB reference circuit



23.3. BF7707AM64-LJTA reference circuit



23.4. BF7707AM64-LJTB reference circuit



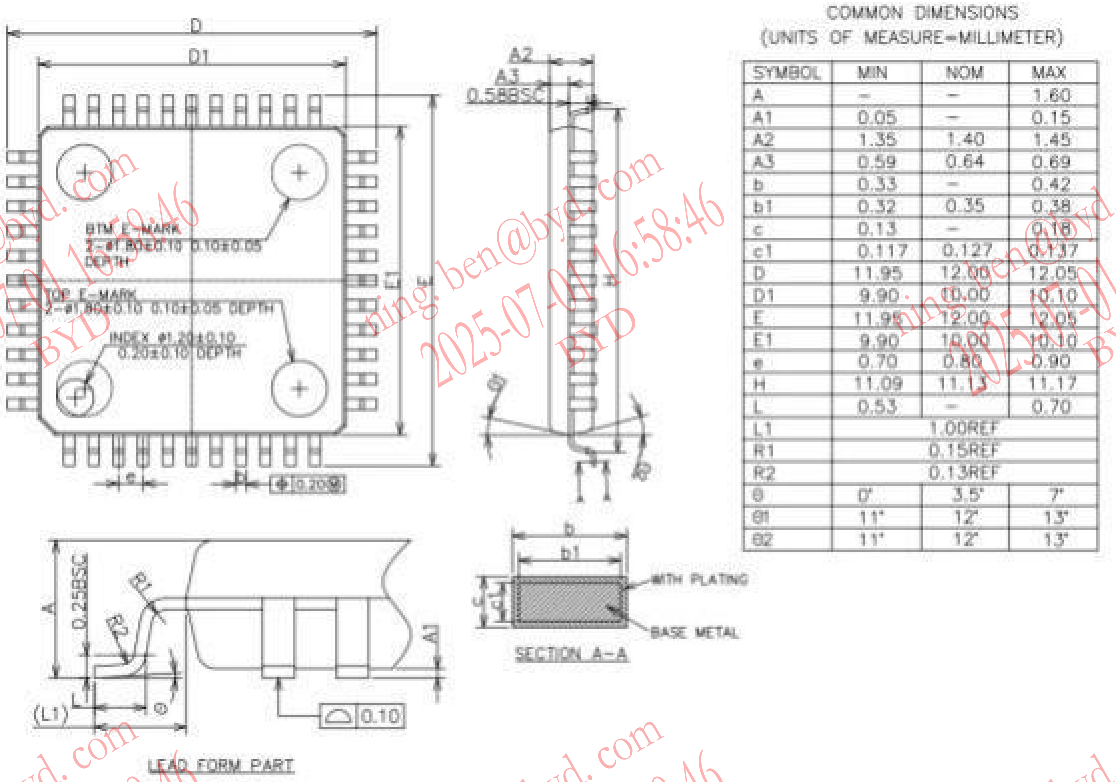
Note:

1. The above reference application circuit is for reference design only.
2. Debug the simulation circuit. If a pull-up resistor has been connected to the simulator or switch board, you do not need to connect a pull-up resistor.

Chapter 24 Package information

24.1. LQFP44

24.1.1. Package1（44-pin LQFP Package）（BF7707AM44-LJTX-CXXX）

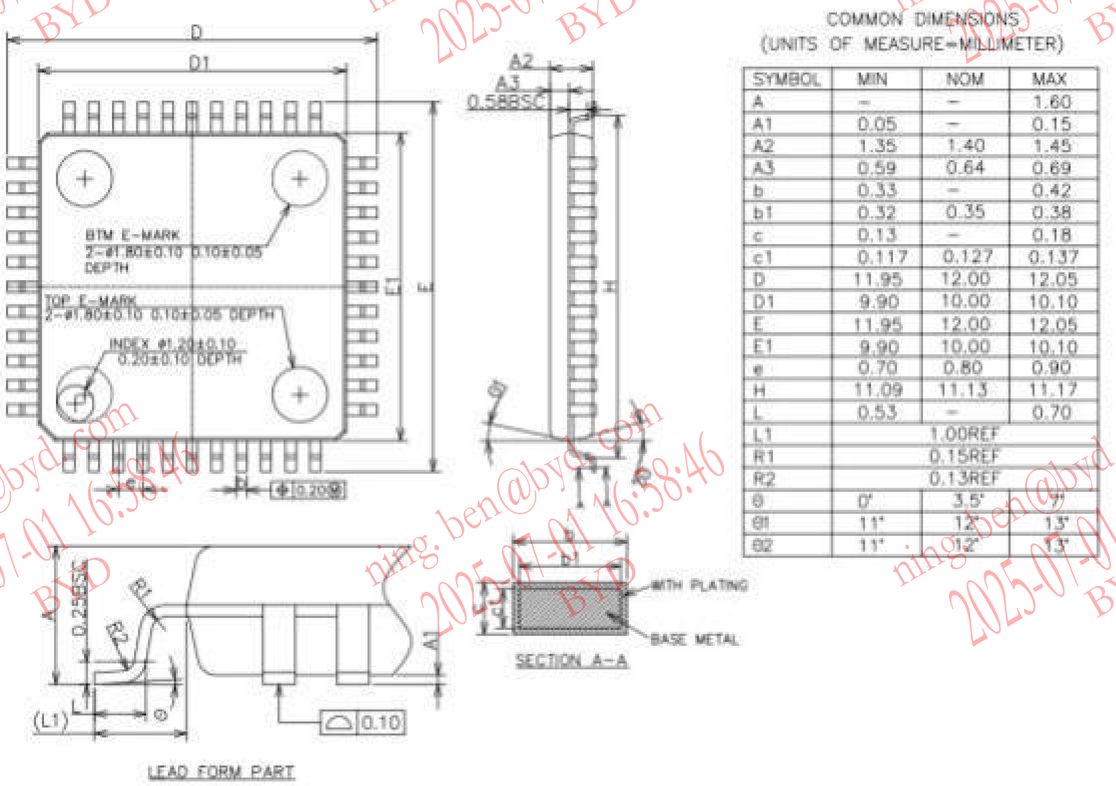


COMMON DIMENSIONS
(UNITS OF MEASURE=MILLIMETER)

SYMBOL	MIN	NOM	MAX
A	–	–	1.60
A1	0.05	–	0.15
A2	1.35	1.40	1.45
A3	0.59	0.64	0.69
b	0.33	–	0.42
b1	0.32	0.35	0.38
c	0.13	–	0.18
c1	0.117	0.127	0.137
D	11.95	12.00	12.05
D1	9.90	10.00	10.10
E	11.95	12.00	12.05
E1	9.90	10.00	10.10
e	0.70	0.80	0.90
H	11.09	11.13	11.17
L	0.53	–	0.70
L1	1.00REF		
R1	0.15REF		
R2	0.13REF		
θ	0°	3.5°	7°
θ1	11°	12°	13°
θ2	11°	12°	13°

BF7707AM44-LJTX-CXXX

24.1.2. Package2（44-pin LQFP Package）（BF7707AM44-LJTX-WXXX）

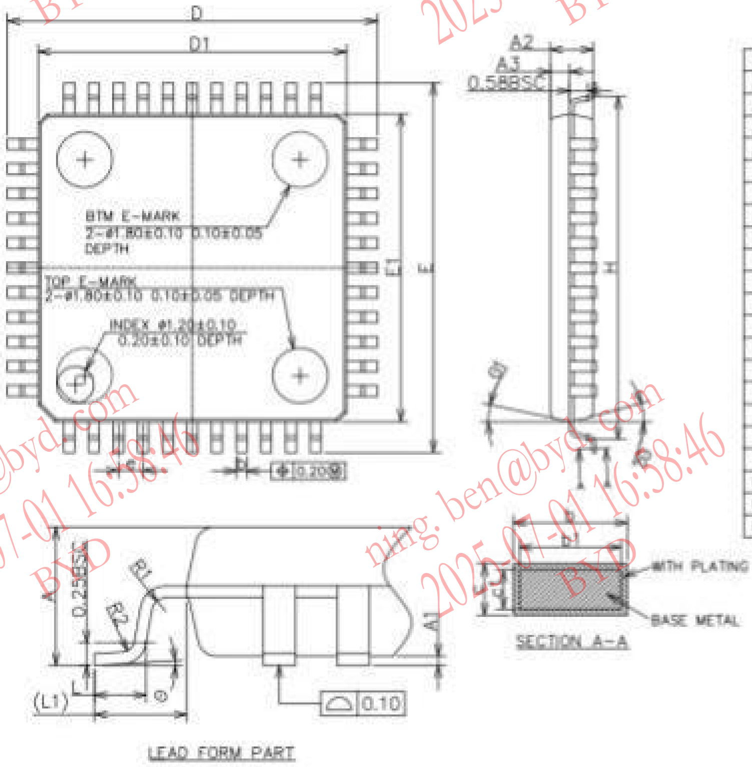


COMMON DIMENSIONS
(UNITS OF MEASURE=MILLIMETER)

SYMBOL	MIN	NOM	MAX
A	–	–	1.60
A1	0.05	–	0.15
A2	1.35	1.40	1.45
A3	0.59	0.64	0.69
b	0.33	–	0.42
b1	0.32	0.35	0.38
c	0.13	–	0.18
c1	0.117	0.127	0.137
D	11.95	12.00	12.05
D1	9.90	10.00	10.10
E	11.95	12.00	12.05
E1	9.90	10.00	10.10
e	0.70	0.80	0.90
H	11.09	11.13	11.17
L	0.53	–	0.70
L1	1.00REF		
R1	0.15REF		
R2	0.13REF		
θ	0°	3.5°	7°
θ1	11°	12°	13°
θ2	11°	12°	13°

BF7707AM44-LJTX-WXXX

24.1.3. Package3（44-pin LQFP Package）（BF7707AM44-LJTX-BXXX）



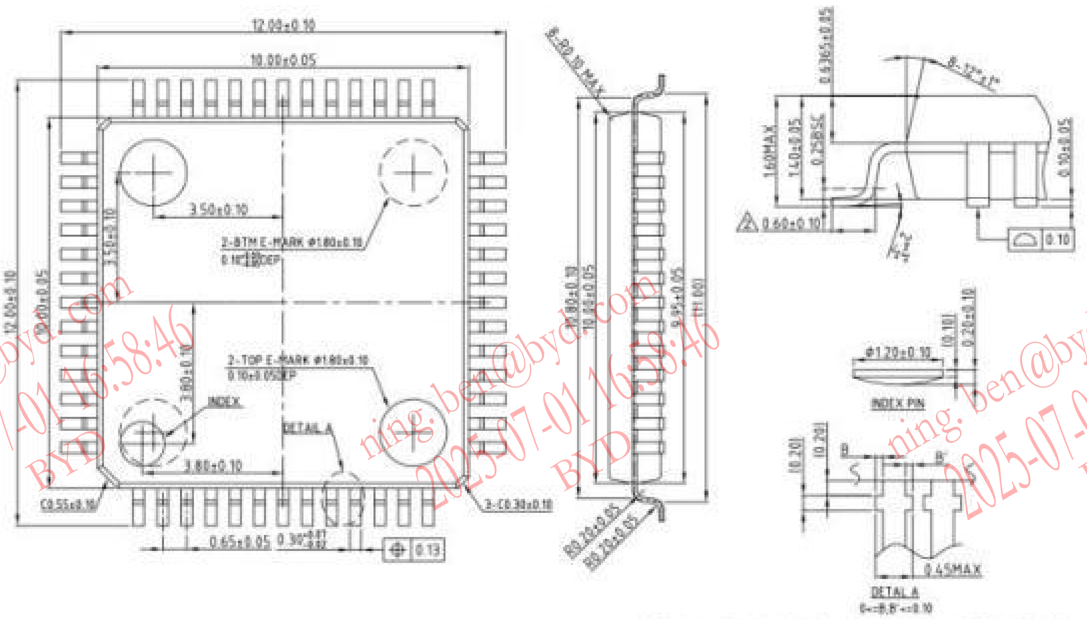
COMMON DIMENSIONS
(UNITS OF MEASURE=MILLIMETER)

SYMBOL	MIN	NOM	MAX
A	–	–	1.60
A1	0.05	–	0.15
A2	1.35	1.40	1.45
A3	0.59	0.64	0.69
b	0.33	–	0.42
b1	0.32	0.35	0.38
c	0.13	–	0.18
c1	0.117	0.127	0.137
D	11.95	12.00	12.05
D1	9.90	10.00	10.10
E	11.95	12.00	12.05
E1	9.90	10.00	10.10
e	0.70	0.80	0.90
H	11.09	11.13	11.17
L	0.53	–	0.70
L1	1.00REF		
R1	0.15REF		
R2	0.13REF		
θ	0°	3.5°	7°
θ1	11°	12°	13°
θ2	11°	12°	13°

BF7707AM44-LJTX-BXXX

24.2. LQFP52

24.2.1. Package1（52-pin LQFP Package）（BF7707AM52-LJTB-BXXX）



NOTES: 1. THE ROUGHNESS OF ALL PACKAGE SURFACES EXCEPT INDEX AND E-MARK: Rz 12±2um.
2.L/F THICKNESS AFTER PLATING 0.127+0.05.

BF7707AM52-LJTB-BXXX

24.3. LQFP64

24.3.1. Package1（64-pin LQFP Package）（BF7707AM64-LJTA-WXXX）

BF7707AM64-LJTA-WXXX