

MG32F04A064

Arm® Cortex® M0+ Core

User Guide

Version 0.01

Date 2026/09/09

Contents

Contents	3
Figures.....	13
Tables	16
1. System and Memory Architecture	18
1.1 System architecture overview.....	18
1.1.1 System bus.....	18
1.1.2 BUS matrix	19
1.2 Memory introduction.....	19
1.2.1 Memory map and register addressing	19
1.2.2 Embedded SRAM	20
1.2.3 FLASH memory overview	20
2 Embedded FLASH.....	21
2.1 Overview	21
2.2 Flash Composition and Description	21
2.2.1 Flash composition	21
2.2.2 Option Byte Description	22
2.3 Flash Operations and Processes.....	23
2.3.1 Read operation of the Flash memory	23
2.3.2 Flash Programming Mode and Operating Procedures.....	24
2.3.3 Disable and enable the operation restriction on Flash block.....	25
2.3.4 Disable and enable the operation restriction on option byte block.....	25
2.3.5 Main Flash block erasing	27
2.3.6 Main Flash block programming.....	29
2.3.7 Option byte block erasing	30
2.3.8 Option byte block programming.....	31
2.3.9 Flash memory protection	32
2.3.9.1 Main Flash block read protection	33
2.3.9.2 Main Flash block write protection	34
2.4 Register	34
2.4.1 Register overview.....	34
2.4.2 FLASH_ACR Flash Access Control Register	35
2.4.3 FLASH_KEYR FPEC Key Register	35
2.4.4 FLASH_OPTKEYR Flash OPTKEY Register	36
2.4.5 FLASH_SR Flash Status Register	36
2.4.6 FLASH_CR Flash Control Register	37
2.4.7 FLASH_AR Flash Address Register	37
2.4.8 FLASH_OBR Option Byte Register	38
2.4.9 FLASH_WRPWR Write Protection Register	38
3 CRC Cyclic Redundancy Check Calculation Unit	40
3.1 Introduction.....	40
3.2 Key Features.....	40
3.3 Function Description	40
3.3.1 Functional Block Diagram	40
3.3.2 Functional Overview.....	40
3.3.3 Usage	40
3.3.3.1 CRC Calculation Operation Steps.....	40
3.4 Register	41
3.4.1 Register Overview.....	41
3.4.2 CRC_DR CRC Data Register	41

	3.4.3	CRC_IDR CRC Independent Data Register	41
	3.4.4	CRC_CR CRC Control Register	42
4		Chip-specific Configuration.....	43
	4.1	Interrupt vector table	43
	4.2	External interrupt mapping	44
	4.3	ADC Channel Assignment	44
	4.4	System Module Hardware Interconnect.....	45
5		PWR Power Controller	48
	5.1	Power supply system	48
	5.1.1	VDD domain	48
	5.1.2	VDD_Core domain	48
	5.2	Power manager	49
	5.2.1	Power on reset and power down reset	49
	5.2.2	Programmable voltage detector	49
	5.3	Power consumption control.....	50
	5.3.1	Power consumption control overview	50
	5.3.2	Reducing system clock under run mode.....	51
	5.3.3	External clock control	51
	5.3.4	Sleep mode	51
	5.3.4.1	Entering Sleep mode	51
	5.3.5	Stop Mode	52
	5.3.5.1	Enter stop mode	52
	5.3.5.2	Exit stop mode	53
	5.3.6	DeepStop Mode	53
	5.3.6.1	Entering Deep Stop mode	53
	5.3.6.2	Exiting Deep Stop mode.....	54
	5.4	Power control register	54
	5.4.1	PWR_CR Power Control Register	54
	5.4.2	PWR_CSR Power Control/Status Register	55
6		RCC Clock and Reset	56
	6.1	Reset unit	56
	6.1.1	Overview	56
	6.1.2	Functional block diagram	56
	6.1.3	Main characteristics	56
	6.1.4	Functional description	56
	6.1.4.1	POR reset	56
	6.1.4.2	System reset.....	56
	6.2	Clock unit.....	57
	6.2.1	Overview	57
	6.2.2	Functional block diagram	58
	6.2.3	Main characteristics	58
	6.2.4	Functional description	58
	6.2.4.1	External high-speed clock (HSE)	58
	6.2.4.2	High-speed internal clock (HSI).....	59
	6.2.4.3	Low-speed internal clock (LSI)	59
	6.2.4.4	Interrupt	59
	6.2.4.5	System clock selection (SWS)	60
	6.2.4.6	System clock frequency switch	60
	6.2.4.7	Peripheral reset	60
	6.2.4.8	Microcontroller clock output (MCO).....	60

6.2.4.9	Independent watchdog clock.....	60
6.3	Register description	61
6.3.1	Register overview.....	61
6.3.2	RCC_CR Clock Control Register	61
6.3.3	RCC_CFGR Clock Configuration Register	62
6.3.4	RCC_CIR Clock Interrupt Register	63
6.3.5	RCC_APB1RSTR APB1 Peripheral Reset Register	64
6.3.6	RCC_AHBENR AHB Peripheral Clock Enable Register	65
6.3.7	RCC_APB1ENR APB1 Peripheral Clock Enable Register	65
6.3.8	RCC_CSR Control Status Register	67
6.3.9	RCC_AHBRSTR AHB Peripheral Reset Register	68
6.3.10	RCC_SYSCFG System Configuration Register	68
7	GPIO General-Purpose I/Os.....	70
7.1	Overview	70
7.2	Main characteristics	70
7.3	Functional description	70
7.3.1	Functional block diagram	70
7.3.2	GPIO port configuration	71
7.3.3	Alternate function	71
7.3.4	GPIO locking mechanism	72
7.3.5	Input configuration.....	72
7.3.6	Output configuration.....	73
7.3.7	Alternate function configuration	73
7.3.8	Analog input configuration	74
7.3.9	External clock used as alternate GPIO port	74
7.3.10	SWD alternate function remapping.....	75
7.3.11	NRST alternate function remapping	75
7.4	Register	75
7.4.1	Overview of registers	75
7.4.2	GPIOx_CRL Port Configuration Register Low.....	75
7.4.3	GPIOx_CRH Port Configuration Register High	76
7.4.4	GPIOx_IDR Port Input Data Register	76
7.4.5	GPIOx_ODR Port Output Data Register.....	77
7.4.6	GPIOx_BSRR Port Bit Set/Reset Register	77
7.4.7	GPIOx_BRR Port Bit Reset Register	77
7.4.8	GPIOx_LCKR Port Configuration Lock Register	78
7.4.9	GPIOx_DCR Port Output Open Drain Control Register	78
7.4.10	GPIOx_AFRH Port Alternate Function Register Low.....	78
7.4.11	GPIOx_AFRH Port Alternate Function Register High	79
8	EXTI Interrupt and Event	80
8.1	Overview	80
8.2	Main characteristics	80
8.3	Function description	80
8.3.1	Function block diagram.....	80
8.3.2	Interrupt and anomaly vector	80
8.3.3	Wake-up event management.....	81
8.3.4	Interrupt function description.....	82
8.3.5	Hardware interrupt output	82
8.3.6	Hardware event output.....	82
8.3.7	Software interrupt and event output.....	82

8.3.8	External interrupt mapping	82
8.4	Register	83
8.4.1	Register overview.....	83
8.4.2	EXTI_IMR Interrupt Mask Register	83
8.4.3	EXTI_EMR Event Mask Register	83
8.4.4	EXTI_RTSM Rising Edge Trigger Selection Register	84
8.4.5	EXTI_FTSR Falling Edge Trigger Selection Register	84
8.4.6	EXTI_SWIER Software Interrupt Event Register	84
8.4.7	EXTI_PR Software Interrupt Event Pending Register	85
9	ADC Analog-to-Digital Converter.....	86
9.1	ADC introduction	86
9.2	ADC main features.....	86
9.3	ADC system block diagram.....	87
9.4	ADC Functional Description.....	87
9.4.1	ADC on-off control.....	87
9.4.2	Channel Selection	88
9.5	Arbitrary channel operation mode.....	88
9.5.1	Single Conversion Mode.....	88
9.5.2	One-cycle scan mode	89
9.5.3	Continuous Scan Mode.....	89
9.6	Data Alignment.....	90
9.7	Programmable Resolution	91
9.8	Programmable Sampling Time.....	91
9.9	Conversion on external trigger	91
9.10	Internal Reference Voltage	91
9.11	Monitoring ADC Conversion Results in Window Comparator Mode	91
9.12	ADC register description	92
9.12.1	Register Overview	92
9.12.2	A/D data register (ADC_ADDDATA)	92
9.12.3	A/D configuration register (ADC_ADCCFG)	93
9.12.4	ADC_ADCR Control Register	94
9.12.5	A/D window comparison register (ADC_ADCMPR)	95
9.12.6	ADC_ADSTA Status Register	96
9.12.7	A/D data register (ADC_ADDR0 ~ 12)	96
9.12.8	A/D Extended State Register (ADC_ADSTA_EXT)	97
9.12.9	A/D arbitrary channel selection register (ADC_CHANY0)	97
9.12.10	A/D arbitrary channel selection register 1 (ADC_CHANY1)	98
9.12.11	A/D arbitrary channel configuration register (ADC_ANY_CFG).....	98
9.12.12	A/D arbitrary channel control register (ADC_ANY_CR)	99
10	COMP Comparator	100
10.1	Overview	100
10.2	Function block diagram.....	100
10.3	Main characteristics	100
10.4	Functional description	100
10.4.1	COMP clocks and reset.....	100
10.4.2	COMP switch	101
10.4.3	COMP inputs and outputs.....	101
10.4.4	COMP channel selection	101
10.4.5	Interrupt and wakeup	101
10.4.6	Comparator LOCK mechanism	102

10.4.7	Hysteresis voltage	102
10.5	Register	102
10.5.1	COMP control and status register (COMPx_CSR)(x=1)	102
10.5.2	COMP comparator reference voltage register (COMP_CRV).....	104
10.5.3	COMP polling register (COMPx_POLL) (x=1).....	104
11	TIM1 Advanced-Control Timer.....	106
11.1	Introduction.....	106
11.2	Main features	107
11.3	Functional description	108
11.3.1	Clock	108
11.3.1.1	Clock selection	108
11.3.1.2	Time base unit	109
11.3.1.3	Counter modes	110
11.3.2	Repeat counter	113
11.3.3	Output compare	116
11.3.3.1	Force output.....	116
11.3.3.2	Output compare.....	116
11.3.3.3	PWM output.....	117
11.3.3.4	Complementary output and dead-time insertion	126
11.3.3.5	Break function.....	127
11.3.3.6	One-pulse output.....	130
11.3.4	Slave mode	131
11.3.4.1	Reset mode	131
11.3.4.2	Gate mode	132
11.3.4.3	Trigger mode	132
11.3.5	Timer synchronization.....	133
11.3.6	Debug mode	135
11.3.7	Interrupt.....	135
11.3.8	Software Output Settings.....	135
11.4	Register	136
11.4.1	TIM1_CR1 Control Register 1	136
11.4.2	TIM1_CR2 Control Register 2	137
11.4.3	TIM1_SMCR Slave Mode Control Register.....	138
11.4.4	TIM1_DIER Interrupt Enable Register.....	140
11.4.5	TIM1_SR Status Register	140
11.4.6	TIM1_EGR Event Generation Register	141
11.4.7	TIM1_CCMR1 Compare Mode Register 1	142
11.4.8	TIM1_CCMR2 Compare Mode Register 2	143
11.4.9	TIM1_CCER Compare Enable Register.....	144
11.4.10	TIM1_CNT Counter.....	145
11.4.11	TIM1_PSC Prescaler	145
11.4.12	TIM1_ARR Auto Reload Register.....	146
11.4.13	TIM1_RCR Repeat Count Register	146
11.4.14	TIM1_CCR1 Compare Register 1	146
11.4.15	TIM1_CCR2 Compare Register 2	147
11.4.16	TIM1_CCR3 Compare Register 3	147
11.4.17	TIM1_CCR4 Compare Register 4	147
11.4.18	TIM1_BDTR Break and Dead-Time Register.....	147
11.4.19	TIM1_CCMR3 Compare Mode Register 3	149
11.4.20	TIM1_CCR5 Compare Register 5	149

11.4.21	TIM1_PDER PWM Phase Shift Enable Register	150
11.4.22	TIM1_CCRxFALL PWM Phase Shift Count Down Compare Register.....	151
11.4.23	TIM1_BKINF Break Input Filter Register	151
11.4.24	TIM1_DOCTR Debug Mode Output Control Register	152
11.4.25	TIM1_SOER Software Output Global Enable Register.....	152
11.4.26	TIM1_SOCR Software Output Configuration Register	153
11.4.27	TIM1_BKSR Break status register.....	153
12	TIM3 16-Bit General-purpose Timer	155
12.1	Overview	155
12.2	Function block diagram	155
12.3	Main characteristics	155
12.4	Function description	156
12.4.1	Clock	156
12.4.1.1	Clock selection	156
12.4.1.2	Time base unit	158
12.4.1.3	Counter modes	158
12.4.2	Input capture	162
12.4.2.1	Input capture.....	162
12.4.2.2	PWM capture	163
12.4.3	Output compare	164
12.4.3.1	Force output.....	164
12.4.3.2	Output compare.....	164
12.4.3.3	PWM output.....	165
12.4.3.4	External event clear OCxREF	169
12.4.3.5	One-pulse output	169
12.4.4	Slave mode	170
12.4.4.1	Encoder interface	170
12.4.4.2	Reset mode	172
12.4.4.3	Gate mode	172
12.4.4.4	Trigger mode	173
12.4.4.5	External clock mode 2+ slave mode	173
12.4.5	Timer synchronization.....	174
12.4.6	Timer XOR function	174
12.4.7	Debug mode	174
12.4.8	Interrupt.....	175
12.5	Register	175
12.5.1	TIM3_CR1 Control Register 1	175
12.5.2	TIM3_CR2 Control Register 2	176
12.5.3	TIM3_SMCR Slave Mode Control Register.....	177
12.5.4	TIM3_DIER Interrupt Enable Register.....	179
12.5.5	TIM3_SR Status Register	180
12.5.6	TIM3_EGR Event Generation Register	181
12.5.7	TIM3_CCMR1 Capture/Compare Mode Register 1	182
12.5.8	TIM3_CCMR2 Capture/Compare Mode Register 2	184
12.5.9	TIM3_CCER Capture/Compare Enable Register.....	187
12.5.10	TIM3_CNT Counter.....	188
12.5.11	TIM3_PSC Prescaler	188
12.5.12	TIM3_ARR Auto Reload Register.....	188
12.5.13	TIM3_CCR1 Capture/Compare Register 1	188
12.5.14	TIM3_CCR2 Capture/Compare Register 2	189

12.5.15	TIM3_CCR3 Capture/Compare Register 3	189
12.5.16	TIM3_CCR4 Capture/Compare Register 4	189
12.5.17	TIM3_OR Input Option Register	189
13	TIM14 Basic Timer	191
13.1	Overview	191
13.2	Function block diagram	191
13.3	Main characteristics	191
13.4	Function description	191
13.4.1	Clock	191
13.4.1.1	Clock selection	191
13.4.1.2	Time base unit	191
13.4.1.3	Counter modes	192
13.4.2	Input capture	193
13.4.2.1	Input capture	193
13.4.3	Output compare	194
13.4.3.1	Force output	195
13.4.3.2	Output compare	195
13.4.3.3	PWM output	196
13.4.3.4	One-pulse output	197
13.4.4	Debug mode	197
13.4.5	Interrupt	197
13.5	Register	198
13.5.1	TIM14_CR1 Control Register 1	198
13.5.2	TIM14_DIER Interrupt Enable Register	199
13.5.3	TIM14_SR Status Register	199
13.5.4	TIM14_EGR Event Generation Register	200
13.5.5	TIM14_CCMR1 Capture/Compare Mode Register 1	200
13.5.6	TIM14_CCER Capture/Compare Enable Register	202
13.5.7	TIM14_CNT Counter	203
13.5.8	TIM14_PSC Prescaler	203
13.5.9	TIM14_ARR Auto Reload Register	203
13.5.10	TIM14_CCR1 Capture/Compare Register 1	204
13.5.11	TIM14_BDTR Break and Dead-Time Register	204
14	IWDG Independent watchdog	205
14.1	Introduction	205
14.2	Main performance of IWDG	205
14.3	IWDG Function Description	205
14.3.1	Hardware Watchdog	206
14.3.2	Register Access Protection	206
14.3.3	Debug mode	206
14.4	Register	206
14.4.1	Overview of registers	206
14.4.2	IWDG_KR Key Register	206
14.4.3	IWDG_PR Prescaler Register	207
14.4.4	IWDG_RLR Reload Register	207
14.4.5	IWDG_SR Status Register	207
14.4.6	IWDG_CR Control Register	208
14.4.7	IWDG_IGEN Interrupt Generate Register	208
14.4.8	IWDG_CNT Counter Register	209
15	SPI Serial Peripheral Interface	210

15.1	Function block diagram.....	210
15.2	Introduction.....	210
15.3	SPI function description	210
15.3.1	Overview	210
15.3.1.1	Phase bit and polarity of clock signal	211
15.3.1.2	High speed transmission	213
15.3.1.3	Data frame format.....	213
15.3.2	Main characteristics of SPI	213
15.3.3	SPI slave mode.....	214
15.3.3.1	Configuration steps.....	214
15.3.3.2	Data transmission.....	214
15.3.3.3	Data receipt	215
15.3.4	SPI master mode	215
15.3.4.1	Configuration steps.....	215
15.3.4.2	Data transmission.....	215
15.3.4.3	Data receipt	215
15.3.5	Baud rate setting.....	216
15.3.6	Interrupt.....	216
15.3.6.1	Status flag.....	216
15.4	Register description	217
15.4.1	SPI_TXREG Transmission Data Register	217
15.4.2	SPI_RXREG Receipt Data Register.....	217
15.4.3	SPI_CSTAT Current Status Register	218
15.4.4	SPI_INTSTAT Interrupt Status Register	218
15.4.5	SPI_INTEN Interrupt Enable Register	219
15.4.6	SPI_INTCLR Interrupt Clear Register	220
15.4.7	SPI_GCTL Global Control Register	220
15.4.8	SPI_CCTL Universal Control Register	222
15.4.9	SPI_SPBRG Baud Rate Generator	223
15.4.10	SPI_RXDNR Receipt Data Number Register.....	223
15.4.11	SPI_NSSR Slave Chip Selection Register	223
15.4.12	SPI_EXTCTL Data Length Control Register	224
16	I2C Inter-Integrated Circuit Interface	225
16.1	Introduction.....	225
16.2	Main characteristics	225
16.3	Functional description	226
16.3.1	Functional block diagram.....	226
16.3.2	Pin definitions.....	226
16.3.3	I2C protocol.....	226
16.3.3.1	Start and STOP conditions	226
16.3.3.2	Addressing protocol.....	227
16.3.3.3	Transmitting and receiving protocol	227
16.3.3.4	Arbitration	229
16.3.3.5	Clock synchronization.....	230
16.3.3.6	SCL configuration	230
16.3.4	Operating mode	231
16.3.4.1	Spike filtering	231
16.3.4.2	Address matching.....	231
16.3.5	Interrupt.....	231
16.3.5.1	Master transmitter mode	231

16.3.5.2	Master receiver mode.....	234
16.3.5.3	Slave transmitter mode	235
16.3.5.4	Slave receiver mode.....	236
16.3.5.5	Other modes.....	239
16.4	Register	239
16.4.1	Overview of registers	239
16.4.2	I2C_DAT Data Register	239
16.4.3	I2C_ADR Address Register	240
16.4.4	I2C_CON Control Register	240
16.4.5	I2C_STA Status Register	241
16.4.6	I2C_DIV Frequency Divider Register	241
17	USART Universal Synchronous Asynchronous Receiver Transmitter	242
17.1	Introduction.....	242
17.2	USART features	242
17.3	USART functional description	243
17.3.1	Functional block diagram.....	243
17.3.2	Signal description	243
17.3.3	Functional description.....	243
17.3.4	Character description.....	244
17.3.5	Baud rate generator.....	246
17.3.6	Sampling	246
17.3.7	Parity check control	246
17.3.8	Transmitter	246
17.3.8.1	Character transmission	247
17.3.8.2	Send break frame	247
17.3.8.3	Transmission configuration procedure	247
17.3.9	Receiver	247
17.3.9.1	Receive break frame	248
17.3.9.2	Receive idle character	248
17.3.9.3	Reception configuration procedure	248
17.3.10	Synchronous mode.....	248
17.3.10.1	Clock description	248
17.3.10.2	Clock synchronization description.....	249
17.3.11	Single-wire half-duplex communication	249
17.3.12	Interrupts.....	249
17.4	Register	250
17.4.1	Overview of registers	250
17.4.2	USART_SR status register	250
17.4.3	USART_DR data register	251
17.4.4	USART_BRR baud rate register.....	252
17.4.5	USART_CR1 control register 1	252
17.4.6	USART_CR2 control register 2	254
17.4.7	USART_CR3 control register 3	254
18	SYSCFG System Controller	256
18.1	Introduction.....	256
18.2	GPIOA 7-segment display driver	256
18.3	Internal Reference Voltage	257
18.4	Register	257
18.4.1	Register overview	257
18.4.2	SYSCFG_CFGR Configuration Register.....	257

18.4.3	SYSCFG_EXTICR1 External Interrupt Configuration Register 1.....	258
18.4.4	SYSCFG_EXTICR2 External Interrupt Configuration Register 2.....	258
18.4.5	SYSCFG_EXTICR3 External Interrupt Configuration Register 3.....	258
18.4.6	SYSCFG_EXTICR4 External Interrupt Configuration Register 4.....	259
18.4.7	SYSCFG_SENSORCR Analog Sensor Configuration Register.....	259
18.4.8	SYSCFG_LEDIOCR Constant-Current Source IO Configuration Register	259
18.4.9	SYSCFG_HCIOCR Sink Current I/O Configuration Register	260
19	Device Electronic Signature	261
19.1	Overview	261
19.2	Register description	261
19.2.1	UID1 Unique Identifier	261
19.2.2	UID2 Unique Identifier	262
19.2.3	UID3 Unique Identifier	262
20	DBG Debug Support.....	263
20.1	Introduction.....	263
20.2	Function descriptions	263
20.2.1	Function block diagram.....	263
20.2.2	SWD internal pull up and down	263
20.2.3	SWD debug port	264
20.3	ID code	264
20.3.1	Microcontroller device ID code	264
20.3.2	Cortex JEDEC-106 ID code.....	264
20.4	SW debug port	264
20.4.1	SW protocol introduction.....	264
20.4.2	SW protocol series.....	264
20.4.3	SW-DP status unit (Reset, Idle states, ID code)	265
20.4.4	DP and AP read/write access	265
20.4.5	SW-DP register	266
20.4.6	SW-AP register	266
20.5	MCU debug module (DBGMCU).....	266
20.5.1	Debug support at low power mode.....	266
20.5.2	Support timer, watchdog	267
20.6	Register	267
20.6.1	Register overview	267
20.6.2	DBG_IDCODE ID Encode Register.....	267
20.6.3	DBG_CR Control Register.....	267
21	History Records	269

Figures

Figure 1-1 System architecture block diagram	18
Figure 3-1 CRC Functional Block Diagram	40
Figure 5-1 Power supply block diagram	48
Figure 5-2 Power-on reset and power-down reset waveform	49
Figure 5-3 PVD threshold waveform	50
Figure 6-1 Reset functional block diagram	56
Figure 6-2 Clock tree	58
Figure 6-3 External high-speed input speed.....	58
Figure 7-1 Standard I/O port.....	70
Figure 7-2 Floating/pull-up/pull-down input configuration	72
Figure 7-3 Output configuration	73
Figure 7-4 Alternate function configuration.....	74
Figure 7-5 Analog input.....	74
Figure 8-1 EXTI structure block diagram.....	80
Figure 9-1 ADC system block diagram	87
Figure 9-2 ADC diagram	87
Figure 9-3 Channel conversion timing diagram in single conversion mode.....	88
Figure 9-4 Channel conversion timing diagram in One-cycle conversion mode.....	89
Figure 9-5 Channel conversion timing diagram in Continuous scan mode.....	90
Figure 9-6 Timing diagram with dynamically updated configuration, in Continuous scan mode	90
Figure 9-7 Data alignment mode	90
Figure 10-1 Comparator block diagrams	100
Figure 10-2 Comparator hysteresis	102
Figure 11-1 Advanced-control timer block diagram	106
Figure 11-2 Clock selection.....	108
Figure 11-3 Control circuit in external clock mode 1	109
Figure 11-4 Auto preload.....	109
Figure 11-5 Up counting mode (UDIS=0)	110
Figure 11-6 Up counting mode (UDIS =1, disable update event)	111
Figure 11-7 Down counting mode (UDIS=0).....	111
Figure 11-8 Down counting mode (UDIS=1 disables update event)	111
Figure 11-9 Central count mode (UDIS=0)	112
Figure 11-10 Central count mode (UDIS=1 disables update event)	112
Figure 11-11 Repeat count time sequence in the central alignment mode	113
Figure 11-12 Up count time sequence in the edge alignment mode.....	114
Figure 11-13 Down count time sequence in the edge alignment mode	114
Figure 11-14 Output compare block diagram	116
Figure 11-15 Output compare mode, OC1 signals toggle when match	117
Figure 11-16 Waveform in PWM mode 1 during edge alignment up count.....	118
Figure 11-17 Waveform in PWM mode 1 during edge alignment down count.....	119
Figure 11-18 Waveform in central alignment PWM mode 1.....	119
Figure 11-19 Move phase function	120
Figure 11-20 Triggering of CCRxFALL incrementing count in phase-shift mode	121
Figure 11-21	122
Figure 11-22	123
Figure 11-23	124
Figure 11-24 Example of six step PWM with COM (OSSR = 1).....	125
Figure 11-25 CCx_SETTRGO in edge aligned incremental counting mode output example.....	125

Figure 11-26 CCx_SETTRGO in central aligned incremental counting mode output example	126
Figure 11-27 Dead-time insertion	127
Figure 11-28 Response break output (OISx=0 , OISxN=0)	129
Figure 11-29 Response break output (OISx=0, OISxN=1).....	129
Figure 11-30 Response break output (OISx=1,OISxN=0)	130
Figure 11-31 Response break output (OISx=1, OISxN=1).....	130
Figure 11-32 one-pulse mode	130
Figure 11-33 Control time sequence in reset mode	131
Figure 11-34 Control time sequence in gate mode	132
Figure 11-35 Control time sequence in trigger mode	132
Figure 11-36 Timer connectivity	133
Figure 11-37 Use master timer as slave timer prescaler.....	133
Figure 11-38 Use master timer to enable slave timer.....	134
Figure 11-39 Use master timer update event to start slave timer	134
Figure 11-40 master timer ETR start master and slave timer in step.....	135
Figure 12-1 TIM3 block diagram.....	155
Figure 12-2 Clock selection	156
Figure 12-3 Control circuit in external clock mode 1	157
Figure 12-4 Control circuit in external clock mode 2	157
Figure 12-5 Auto preload	158
Figure 12-6 Up counting mode (UDIS=0)	159
Figure 12-7 Up counting mode (UDIS=1 disables update event)	159
Figure 12-8 Down counting mode (UDIS=0).....	160
Figure 12-9 Down counting mode (UDIS=1 disables update event).....	160
Figure 12-10 Central count mode (UDIS=0)	161
Figure 12-11 Central count mode (UDIS=1 disables update event)	161
Figure 12-12 TIM3 input capture block diagram.....	162
Figure 12-13 PWM input mode sequence.....	163
Figure 12-14 Output compare block diagram.....	164
Figure 12-15 Output compare mode, OC1 signals toggle when match.....	165
Figure 12-16 Waveform in PWM mode 1 during edge alignment up count	166
Figure 12-17 Waveform in PWM mode 1 during edge alignment down count.....	166
Figure 12-18 Waveform in central alignment PWM mode 1.....	167
Figure 12-19 example of CCx_SETTRGO output with edge aligned up counting mode.....	168
Figure 12-20 example of CCx_SETTRGO output with central alignment mode.....	168
Figure 12-21 The external event clear OCxREF	169
Figure 12-22 one-pulse mode.....	170
Figure 12-23 Counter time sequence in the encoder code	171
Figure 12-24 IC1FP1 polarity inverted sequence of encoder interface mode	171
Figure 12-25 Control time sequence of reset mode	172
Figure 12-26 Control time sequence of gate mode	172
Figure 12-27 Control time sequence of trigger mode.....	173
Figure 12-28 Control time sequence of external clock mode 2 +slave mode (trigger mode)	173
Figure 12-29 (TI1 XOR input) input capture waveform	174
Figure 13-1 TIM14 block diagram.....	191
Figure 13-2 Auto preload	192

Figure 13-3 Up counting mode (UDIS=0)	192
Figure 13-4 Up counting mode (UDIS =1, disable update event)	192
Figure 14-1 Independent Watchdog Block Diagram	205
Figure 15-1 SPI function block diagram	210
Figure 15-2 Single master and slave application	211
Figure 15-3 Data clock sequence	213
Figure 16-1 I2C functional block diagram.....	226
Figure 16-2 Start and STOP conditions.....	226
Figure 16-3 7-bit address format	227
Figure 16-4 Master-transmitter protocol	228
Figure 16-5 Master-receiver protocol	228
Figure 16-6 Master-transmitter and receiver protocol with a RESTART (SR)	229
Figure 16-7 Dual master arbitration	229
Figure 16-8 Clock synchronization (schematic diagram)	230
Figure 16-9 Clock synchronization (timing diagram)	230
Figure 17-1 USART functional block diagram	243
Figure 17-2 UART data frame type diagram	244
Figure 18-1 Schematic diagram of drive waveforms for a 3-digit 7-segment display	256
Figure 20-1 Debug function block diagram.....	263

Tables

Table 1-1 Memory mapping	19
Table 2-1 Flash storage space.....	21
Table 2-2 Information block.....	21
Table 2-3 Data memory block	22
Table 2-4 Option byte organizaiont.....	22
Table 2-5 USER bit meaning	22
Table 2-6 Relationships between Latency settingsit.....	24
Table 2-7 Programming method	24
Table 2-8 Changes of status in the protection setting	26
Table 2-9 Flash read protection status	33
Table 2-10 Flash disable read protection status.....	33
Table 2-11 Write protection area.....	34
Table 2-12 FLASH register overview	34
Table 3-1 CRC Register Overview.....	41
Table 4-1 Abnormal vector	43
Table 4-2 Interrupt vector	43
Table 4-3 EXTI trigger source	44
Table 4-4 ADC Channel Assignment	44
Table 4-5 Interconnection of ADC trigger source.....	45
Table 4-6 Comparator Output Direction Selection.....	46
Table 4-7 Interconnection of TIM ITR Event.....	46
Table 4-8 Interconnection of TIM ETR Event.....	47
Table 4-9 Interconnection of Input Capture Event.....	47
Table 4-10 Interconnection of TIM OCREFLR Event.....	47
Table 4-11 Interconnection of TIM Break Event.....	47
Table 5-1 Low power modes.....	50
Table 5-2 SLEEPNOW mode.....	52
Table 5-3 SLEEPONEXIT mode	52
Table 5-4 Stop mode.....	53
Table 5-5 Deep Stop mode	54
Table 5-6 Power control register overview.....	54
Table 6-1 RCC global interrupt	59
Table 6-2 Correlation of MCO and clock source.....	60
Table 6-3 RCC register overview	61
Table 7-1 Port bit configuration table (take port0 as an example)	71
Table 7-2 SWD alternate function remapping.....	75
Table 7-3 Overview of GPIO registers	75
Table 8-1 Abnormal vector	80
Table 8-2 Interrupt vector	81
Table 8-3 EXTI trigger source	82
Table 8-4 EXTI register overview.....	83
Table 9-1 ADC register overview	92
Table 10-1 Overview of COMP registers	102
Table 11-1 Dead-time calculation.....	127
Table 11-2 In case of MOE=1, OSSR=0/1, OSSR=0:.....	128
Table 11-3 In case of MOE=1, OSSR=0/1, OSSR=1.....	128
Table 11-4 In case of MOE=0, OSSR=0, OSSR=0/1.....	128

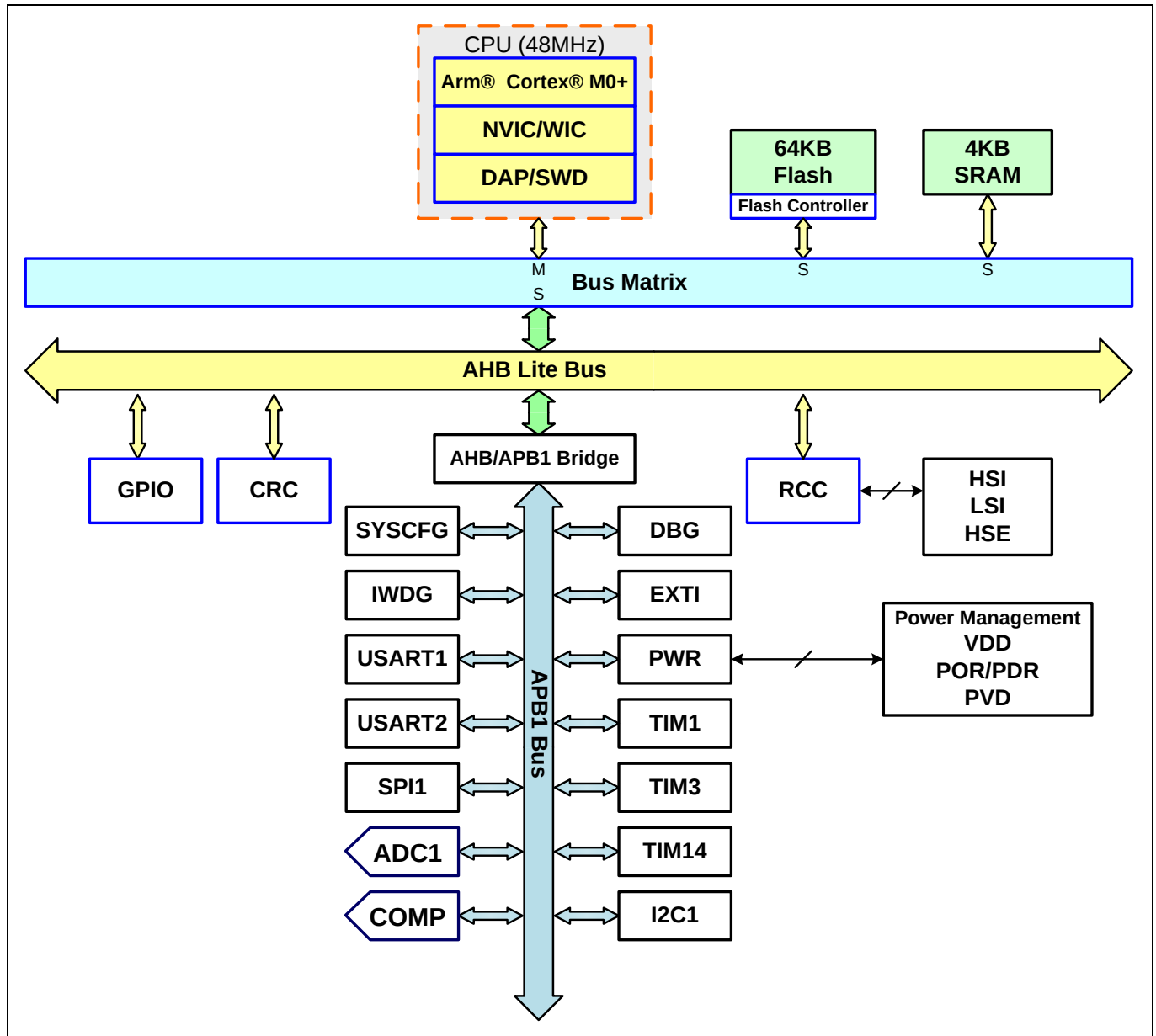
Table 11-5 In case of MOE=0, OSS1=1, OSSR=0/1	128
Table 11-6 List of interrupt events.....	135
Table 11-7 TIM1 register overview	136
Table 12-1 Digital filter width and ICxF correlation	162
Table 12-2 Relations of count direction and encoder signal.....	171
Table 12-3 List of interrupt events.....	175
Table 12-4 TIM3 register overview.....	175
Table 12-5 ICx polarity/level selection	188
Table 13-1 Digital filter width and IC1F correlation.....	193
Table 13-2 List of interrupt events.....	197
Table 13-3 TIM14 register overview.....	198
Table 13-4 IC1 polarity/level selection table	203
Table 14-1 IWDG timeout (For example, the LSI clock frequency is 40 KHz)	205
Table 14-2 Overview of IWDG registers	206
Table 15-1 Baud rate formula	216
Table 15-2 SPI status.....	216
Table 15-3 SPI register overview	217
Table 16-1 Pin definitions.....	226
Table 16-2 First byte for I2C	227
Table 16-4 I2C Master Receiver Mode Interrupt Status	234
Table 16-5 I2C Slave Transmitter Mode Interrupt Status	235
Table 16-6 I2C Slave Receiver Mode Interrupt Status	237
Table 16-7.Miscellaneous Status	239
Table 16-8 Overview of I2C registers	239
Table 17-1 UART interrupt requests	250
Table 17-2 Overview of USART registers.....	250
Table 18-1 SYSCFG register overview.....	257
Table 19-1 Device electronic signature register overview	261
Table 20-1 SWD debug port pin.....	264
Table 20-2 ID code.....	264
Table 20-3 8bit request packet.....	265
Table 20-4 3bit response package.....	265
Table 20-5 33bit data packet.....	265
Table 20-6 SW-DP register	266
Table 20-7 DBG register overview	267
Table 21-1 History records.....	269

1. System and Memory Architecture

1.1 System architecture overview

MG32F04A064 product is a 32-bit microcontroller product based on Arm®Cortex®-M0+ processor. It features high performance and low power consumption. It adopts the matrix bus structure, which includes one AHB master: CPU and the following slaves: SRAM, Flash memory, AHB bus (including bus bridge from AHB to APB), and a variety of devices connected to APB bus.

Figure 1-1 System architecture block diagram



1.1.1 System bus

The system bus works to achieve data transfer by connecting the CPU core and bus matrix to build the path between CPU and bus matrix. The CPU can work as the host drive bus, enabling the read and write operations to slaves such as APB peripheral, SRAM, Flash, etc.

In addition, the CPU can quickly access GPIO through a dedicated IOP bus.

1.1.2 BUS matrix

The bus matrix consists of an AHB interconnected matrix, an AHB bus and a bridged APB bus. Peripherals (RCC, GPIO and CRC) of the AHB bus are connected to the system bus via the AHB interconnected matrix. The connections between the APB and AHB buses exchange data via the AHB1APB bridge. When the APB registers are accessed by 8-bit, 16-bit, the APB is automatically widened to 32-bit, and similarly, the AHB1APB bridge has an automatic widening function.

1.2 Memory introduction

Program memory, data memory, registers, and I/O interfaces are located in memory address space (linear 4GB address space) with different address range. 4GB address space is divided into 8 blocks, with 512MB per block. The space allocated to the on-chip memory and peripherals is fixed, and not subject to change. The remaining address space is reserved, and can be defined and used by the chip vendor.

1.2.1 Memory map and register addressing

For the memory map, please refer to the memory map diagram in the corresponding chapter of each peripheral.

Table 1-1 Memory mapping

Bus	Address range	Size	Peripheral
Flash	0x0000 0000 - 0x0000 FFFF	64 KB	Could be mapped as main memory, system memory or SRAM, based on SYSCFG registers configuration
	0x0001 0000 - 0x07FF FFFF	~127 MB	Reserved
	0x0800 0000 - 0x0800 FFFF	64 KB	Main memory
	0x0801 0000 - 0x1FFD FFFF	~383 MB	Reserved
	0x1FFE 0000 - 0x1FFE 03FF	1 KB	Data memory
	0x1FFE 0400 - 0x1FFF F3FF	124 KB	Reserved System memory
	0x1FFE F400 - 0x1FFF F7FF	1 KB	System memory
	0x1FFF F800 - 0x1FFF F9FF	0.5KB	Option byte
	0x1FFF FA00 - 0x1FFF FFFF	1.5KB	Reserved
SRAM	0x2000 0000 - 0x2000 0FFF	4 KB	SRAM
	0x2000 1000 - 0x2FFF FFFF	~255 MB	Reserved
APB1	0x4000 0000 - 0x4000 03FF	1KB	Reserved
	0x4000 0400 - 0x4000 07FF	1KB	TIM3
	0x4000 0800 - 0x4000 2FFF	11KB	Reserved
	0x4000 3000 - 0x4000 33FF	1KB	IWDG
	0x4000 3400 - 0x4000 43FF	4KB	Reserved
	0x4000 4400 - 0x4000 47FF	1KB	USART2
	0x4000 4800 - 0x4000 53FF	3KB	Reserved
	0x4000 5400 - 0x4000 57FF	1KB	I2C1
	0x4000 5800 - 0x4000 6FFF	6KB	Reserved
	0x4000 7000 - 0x4000 73FF	1KB	PWR
	0x4000 7400 - 0x4000 FFFF	35KB	Reserved
	0x4001 0000 - 0x4001 03FF	1KB	SYSCFG
	0x4001 0400 - 0x4001 07FF	1KB	EXTI
	0x4001 0800 - 0x4001 23FF	7KB	Reserved
	0x4001 2400 - 0x4001 27FF	1KB	ADC1
	0x4001 2800 - 0x4001 2BFF	1KB	Reserved
	0x4001 2C00 - 0x4001 2FFF	1KB	TIM1
	0x4001 3000 - 0x4001 33FF	1KB	SPI1
	0x4001 3400 - 0x4001 37FF	1KB	DBG

Bus	Address range	Size	Peripheral
	0x4001 3800 – 0x4001 3BFF	1KB	USART1
	0x4001 3C00 – 0x4001 3FFF	1KB	Reserved
	0x4001 4000 – 0x4001 43FF	1KB	TIM14
	0x4001 4400 – 0x4001 47FF	1KB	COMP
	0x4001 4800 – 0x4001 FFFF	46KB	Reserved
AHB	0x4002 0000 – 0x4002 0FFF	4KB	Reserved
	0x4002 1000 – 0x4002 13FF	1KB	RCC
	0x4002 1400 – 0x4002 1FFF	3KB	Reserved
	0x4002 2000 – 0x4002 23FF	1KB	Flash Interface
	0x4002 2400 – 0x4002 2FFF	3KB	Reserved
	0x4002 3000 – 0x4002 33FF	1KB	CRC
	0x4002 3400 – 0x4002 FFFF	51KB	Reserved
IOP	0x5000 0000 – 0x5000 03FF	1KB	PORT A
	0x5000 0400 – 0x5000 07FF	1KB	PORT B

1.2.2 Embedded SRAM

It features up to 4K bytes of a built-in static SRAM. It can be accessed in byte (8 bits), half-word (16 bits) or word (32 bits). The starting address of SRAM is 0x2000 0000.

SRAM can be accessed by CPU with the fastest system clock and without any waiting.

1.2.3 FLASH memory overview

The Flash memory is composed of three storage areas:

- The main Flash memory block includes the application data and user data area.
- The information block includes option bytes and system memory:
- Option bytes - Containing hardware and user storage protection configuration options.
- System memory - Containing system information.
- The data Flash memory block: Used to store non-volatile data and additional user program data.

The Flash memory interface based on the AHB protocol, executes commands and data access. The prefetch buffer function of the Flash interface can speed up the code execution of CPU.

2 Embedded FLASH Overview

- The embedded flash memory supports up to 64 KB of on-chip Main Flash. The flash controller supports read operations, page erase and whole-chip erase. These can be write into the Flash memory by 32-bit (word) programming, with an endurance of 100,000 write/erase cycles. When reading data, the Flash memory controller supports the data interface with the prefetch buffer to support the MCU running at a higher main frequency.

2.2 Flash Composition and Description

2.2.1 Flash composition

- The Flash memory is organized as 32-bit wide memory cells that can be used for storing both code and data.
- The main Flash memory block is divided into 64 pages (1K bytes per page) or 16 write protection blocks (4K bytes per block).
- The main main Flash memory block can be erased by page (per 1Kbyte).
- The write protection is set by taking the unit of 4 pages (4Kbytes) as one write protection block.
- The entire on-chip Flash is composed of three parts: Main storage block, information storage block and data storage block.

The main storage block is used to store user code and data, the user code can erase, program and read the main memory. Each 1Kbyte is called one page in the main storage block, which can perform the smallest unit of erasing. In addition, write protection is allocated in the unit of 1 write protection area (4Kbytes, 4 pages=1 write protection block), as shown in Table 2-1.

Table 2-1 Flash storage space

Module	Block Name	Page Name	Address	Size(byte)
Main memory block	Write-protected area 0	Page 0	0x0800 0000 - 0x0800 03FF	1K
		Page 1	0x0800 0400 - 0x0800 07FF	1K
		Page 2	0x0800 0800 - 0x0800 0BFF	1K
		Page 3	0x0800 0C00 - 0x0800 0FFF	1K
	...	...	...	...
	Write-protected area 3	Page 12	0x0800 3000 - 0x0800 33FF	1K
		Page 13	0x0800 3400 - 0x0800 37FF	1K
		Page 14	0x0800 3800 - 0x0800 3BFF	1K
		Page 15	0x0800 3C00 - 0x0800 3FFF	1K
	...	...	...	...
	Write-protected area 15	Page 60	0x0800 F000 - 0x0800 F3FF	1K
		Page 61	0x0800 F400 - 0x0800 F7FF	1K
		Page 62	0x0800 F800 - 0x0800 FBFF	1K
		Page 63	0x0800 FC00 - 0x0800 FFFF	1K

In the information storage block, except for the "System Memory ISP" area that is factory locked and cannot be written by the user, other areas can be read and written by the user under certain conditions. The information memory can be divided into two parts: system memory and option bytes. The user can erase, program and read the information storage block through a specified procedure.

Table 2-2 Information block

Module	Name	Address	Size(byte)
Information block	System memory	0x1FFF F400 - 0x1FFF F7FF	1K

Module	Name	Address	Size(byte)
	Option bytes	0x1FFF F800 - 0x1FFF F9FF	0.5K

The data storage block includes a data storage area for holding non-volatile data and additional user programs. Erase, program, and read operations for this data memory area are identical to those for the option byte area (except that there is no bit-inversion operation). The data memory area is not restricted by read or write protection mechanisms.

Table 2-3 Data memory block

Module	Name	Address	Size(byte)
Data memory block	Data memory	0x1FFE 0000 - 0x1FFE 03FF	1K

2.2.2 Option Byte Description

In the option byte page, the content mainly includes write protection enable, hardware watchdog enable, etc. The Flash controller can disable the write function of the main memory by setting the value in the option byte to prevent illegal write and it can also enable the hardware watchdog. Relevant information is stored in the option byte. The content in the option bytes, after modification, will only be valid after reset or power-on again. Each time after system reset, the option byte will reload the data of optional byte information block and make corresponding judgments and status changes. These sates are stored in the option byte registers (FLASH_OBR and FLASH_WRP). Each selection bit in the information block has the corresponding inverse code bit, which is used to verify whether the selected bit is correct when loading the selected bit. If difference is detected during the loading process, an option byte error flag (OPTERR) will be produced. When both the option byte and its complement are 0xFF (the erased state), the aforementioned verification function is disabled.

The organization of option bytes within the option byte block is shown in the table below (the values in bits 15–8 are the bitwise complements of the option bytes in bits 7–0; bits 31–16 are reserved and do not undergo complement processing):

Table 2-4 Option byte organizaiont

Address	[15: 8]	[7: 0]	Default value
0x1FFF F800	nRDP	RDP	0x5AA5
0x1FFF F804	nUSER	USER	0x00FF
0x1FFF F808	nData0	Data0	0x00FF
0x1FFF F80C	nData1	Data1	0x00FF
0x1FFF F810	nWRP0	WRP0	0x00FF
0x1FFF F814	nWRP1	WRP1	0x00FF

Table 2-5 USER bit meaning

	Bit	Field	Type	Default	Description	FLASH_OBR
RDP	7: 0	RDP	rw	0xA5	0xA5 Read protection option byte. The read protection helps the user protect the code stored in Flash memory.	FLASH_OBR. Bit1
nRDP	15: 8	nRDP	rw	0x5A	0x5A	
User Byte	0	WDG_SW	rw	0x01	0: Hardware watchdog 1: Software watchdog	FLASH_OBR.Bit2

	Bit	Field	Type	Default	Description	FLASH_OBR
	1	nRST_STOP	rw	0x01	0: Reset is generated when entering STOP mode 1: No reset when entering STOP mode	FLASH_OBR. Bit3
	2	OBR_PVDE	rw	0x01	0: Enable PVD 1: Disable PVD	FLASH_OBR.Bit26
	3	Reserved	rw	0x01	Reserved as 0x01	Reserved
	5: 4	OBR_PLS	rw	0x03	PVD monitoring level selection bits [1:0]	FLASH_OBR.Bit28~27
	7: 6	Reserved	rw	0x03	Reserved as 0x03	Reserved
DATA0 Byte	0	DATA0.Bit0	rw	0x01	User-defined	FLASH_OBR. Bit10
	1	DATA0.Bit1	rw	0x01	User-defined	FLASH_OBR. Bit11
	2	DATA0.Bit2	rw	0x01	User-defined	FLASH_OBR. Bit12
	3	DATA0.Bit3	rw	0x01	User-defined	FLASH_OBR. Bit13
	4	DATA0.Bit4	rw	0x01	User-defined	FLASH_OBR. Bit14
	5	DATA0.Bit5	rw	0x01	User-defined	FLASH_OBR. Bit15
	6	DATA0.Bit6	rw	0x01	User-defined	FLASH_OBR. Bit16
	7	DATA0.Bit7	rw	0x01	User-defined	FLASH_OBR. Bit17
DATA1 Byte	0	DATA1.Bit0	rw	0x01	User-defined	FLASH_OBR. Bit18
	1	DATA1.Bit1	rw	0x01	User-defined	FLASH_OBR. Bit19
	2	DATA1.Bit2	rw	0x01	User-defined	FLASH_OBR. Bit20
	3	DATA1.Bit3	rw	0x01	User-defined	FLASH_OBR. Bit21
	4	DATA1.Bit4	rw	0x01	User-defined	FLASH_OBR. Bit22
	5	DATA1.Bit5	rw	0x01	User-defined	FLASH_OBR. Bit23
	6	DATA1.Bit6	rw	0x01	User-defined	FLASH_OBR. Bit24
	7	DATA1.Bit7	rw	0x01	User-defined	FLASH_OBR. Bit25

Note: In the write-protection value, one bit corresponds to four pages, namely 4,096 bytes.

Note: When programming the first eight word addresses of the option byte area, the hardware automatically sets the upper 8 bits of the word's lower 16 bits to the bitwise inverse of the lower 8 bits, ensuring the written option byte value is always valid. The hardware does not perform this automatic inversion operation when programming other word addresses in the option byte area.

2.3 Flash Operations and Processes

Note: For programming and erase operations, the correct access latency (LATENCY) must also be configured. The principles for configuring LATENCY are the same as those for read operations; please refer to the subsequent instructions.

Note: The system should avoid power loss during programming or erasing operations.

2.3.1 Read operation of the Flash memory

User code and data are stored in the memory block and the Flash memory controller can read data or instructions in 8bit/16-bit/32bit. The main Flash module is uniformly addressed like ordinary peripherals. Based on the read and write protection requirements, any read or write operations on the content of the main memory block shall undergo a specific judgment process to prevent illegal read and write.

The flash memory executes instruction fetching and data fetching via the AHB bus according to the method set in the Flash access control register (FLASH_ACR). Based on AHB clock, set the corresponding access latency and enable the prefetch value buffer. By doing so, the instruction fetch speed of CPU can be

increased, thereby raising the running speed of CPU. The access latency if less than or equal to 24MHz in SYSCLK, can be set as 0. For an increment of 24MHz, an additional latency is required.

After power-on reset, the Flash memory controller sets the prefetch buffer to be open by default. To enable or disable the instruction prefetch buffer function via register configuration, the HCLK frequency must first be set to 24 MHz or lower and the LATENCY set to 0; once the configuration takes effect, the AHB clock (HCLK) is no longer required to remain at or below 24 MHz.

In order to protect the correct reading of Flash, the speed ratio of the prefetch instruction controller must be specified in the LATENCY[2:0] in the Flash access control register. The value is equal to the number of waiting cycles inserted from each Flash access to the next access. After reset, this value is zero by default, that is, no waiting cycle is inserted and the corresponding system clock is also reset as the built-in clock HSI=8MHz. If the system clock is to be modified after reset, the safe value of LATENCY[2:0] shall be configured first. Set the corresponding access latency.

Table 2-6 Relationships between Latency settingsit

HCLK	Latency
0MHz < HCLK <=24MHz	0
24MHz < HCLK <= 48MHz	1
48MHz < HCLK <= 72MHz	2

2.3.2 Flash Programming Mode and Operating Procedures

The embedded flash memory supports the following three programming methods.

Table 2-7 Programming method

Programming Method	Programing Instruction
In-circuit programming (ICP)	ICP means that the user code is burned into MCU by a specific programmer using SWD interface and changing Flash content.
In-application programming (IAP)	Different from ICP mode, IAP (in application programming) can use any MCU supported communication interface (UART, I2C, SPI, etc) to download program or data. IAP allows the user to rewrite the application during program operation, under the precondition that some programs must be preburned by ICP mode.

The burning and erasing can be completed within the working voltage range of the whole product. When writing or erasing the Flash space, the internal oscillator (HSI) must be turned on and the AHB clock must be greater than or equal to 8MHz.

As long as CPU does not access Flash space, the ongoing Flash write operation will not hamper the running of CPU (For example, running in RAM). When writing or erasing the Flash, the read access to the Flash will always encounter a bus pause, and it will not continue until the writing or erasing operation is completed. Therefore, it is not allowed to fetch instruction and access data when the Flash is written or erased.

The programming of Flash is composed of a series of actions, mainly including:

- Unlocking and Protecting Flash Operations
- Flash Erase (Page erasing and whole-chip erase)
- Flash memory Programming. (32-bit whole Word Programming)
- Unlocking and protecting the operations of each sector in the information block (such as option bytes)
- Erase each sector in the information block (such as option bytes)

- Program each sector in the information block (such as option bytes) (32-bit whole Word Programming)

2.3.3 Disable and enable the operation restriction on Flash block

The embedded Flash memory is in a protected status after reset, which can avoid accidental operations that damage the Flash storage space, such as page erase, whole chip erase and value writing. FLASH_CR register is locked after reset, and the LCOK bit of FLASH_CR is set as 1 by the controller module. Only by writing 0x45670123 and 0xCDEF89AB respectively to the FLASH_KEYR register for unlocking and setting the LCOK bit of FLASH_CR as 0, can FLASH_CR be accessed. Otherwise, the FLASH_CR register is not allowed to be rewritten.

Set the LOCK bit of FLASH_CR LOCK as 1 by software to lock again, making the Flash memory in the protected status.

Disable protection operation code:

```
#define FLASH_KEY1      ((unsigned int)0x45670123)
#define FLASH_KEY2      ((unsigned int)0xCDEF89AB)
void FLASH_Unlock()
{
    FLASH->KEYR = ((unsigned int)0x45670123);
    FLASH->KEYR = ((unsigned int)0xCDEF89AB);
}
```

Operations that do not conform to the above sequence and writing the wrong value will lock FLASH_CR and trigger a bus error till the next reset.

Enable protection operation code:

```
#define FLASH_CR_LOCK_Pos      (7)
#define FLASH_CR_LOCK          (0x01U << FLASH_CR_LOCK_Pos)
void FLASH_Lock(void)
{
    FLASH->CR |= FLASH_CR_LOCK;
}
```

2.3.4 Disable and enable the operation restriction on option byte block

After the Flash memory controller is reset, its option byte block is write-protected by default, and readable at any time. In order to avoid destructive operations such as block erasing and writing value on the option byte area and read protection setting area, after reset, FLASH_CR register enters the locked status and the LOCK bit of FLASH_CR is set as 1 by the controller module, while OPTWRE bit is cleared as 0 by the controller module. Therefore, it is necessary to write 0x45670123 and 0xCDEF89AB to the FLASH_KEYR register to unlock the FLASH. After the LOCK bit of FLASH_CR is 0, the option byte block are unlocked. Writing 0x45670123 and 0xCDEF89AB to FLASH_OPT_KEYR register, the hardware will set the OPTWRE bit of FLASH_CR register as 1. By doing so, block erasing and 32-bit word programming can be performed on the option byte area. The OPTWRE bit of the FLASH_CR register can be set as 0, thereby disabling block erasing or 32-bit word programming on the option byte area.

Table 2-8 Changes of status in the protection setting

Set and status	Main Flash block	Information block	Description
Power on and reset The status of Flash controller is FLASH_CR.LOCK=1 FLASH_CR.OPTWRE=0	Protection	Protection	Enable operation protection of main Flash block Enable operation protection of option byte block
Set FLASH_KEYR=0x45670123 FLASH_KEYR=0xCDEF89AB The status of Flash controller is FLASH_CR.LOCK=0 FLASH_CR.OPTWRE=0	Disable protection	Protection	Disable operation protection of main Flash block, execute whole chip erasing, page erasing, 32-bit word programming of main Flash block. Keep enabling operation protection of option byte block. It disables block erasing and 32-bit word programming of option byte area.
FLASH_KEYR=0x45670123 FLASH_KEYR=0xCDEF89AB FLASH_OTPKEYR=0x45670123 FLASH_OTPKEYR=0xCDEF89AB The status of Flash controller turns FLASH_CR.LOCK=0 FLASH_CR.OPTWRE=1	Disable protection	Disable protection	Disable operation protection of main Flash block, execute whole chip erasing, page erasing, 32-bit word programming of main Flash block. Disable operation protection of option byte block. It enables block erasing and 32-bit word programming of option byte area.
Set FLASH_CR.OPTWRE=0 Keep FLASH_CR.LOCK=0	Disable protection	Enable protection	It disables operation protection of main Flash block, and enables operation protection on option byte block.
Set FLASH_CR.OPTWRE=0 Set FLASH_CR.LOCK=1	Enable protection	Enable protection	Enable operation protection of main Flash block as well as option byte block.

Disable protection operation code:

```
#define FLASH_KEY1      ((unsigned int)0x45670123)
#define FLASH_KEY2      ((unsigned int)0xCDEF89AB)
void FLASH_Unlock(void)
{
    FLASH->KEYR = ((unsigned int)0x45670123);
    FLASH->KEYR = ((unsigned int)0xCDEF89AB);
}
```

Enable protection operation code:

```
#define FLASH_CR_LOCK_Pos      (7)
#define FLASH_CR_LOCK      (0x01U << FLASH_CR_LOCK_Pos)
void FLASH_Lock(void)
{
    FLASH->CR |= FLASH_CR_LOCK;
}
```

Disable operation code of option byte as well as read protection setting area:

```
#define FLASH_KEY1      ((unsigned int)0x45670123)
#define FLASH_KEY2      ((unsigned int)0xCDEF89AB)
```



```
void FLASH_OPT_Unlock (void)
{
    FLASH->OPTKEYR = FLASH_KEY1;
    FLASH->OPTKEYR = FLASH_KEY2;
}
```

Enable operation code of option byte as well as read protection setting area:

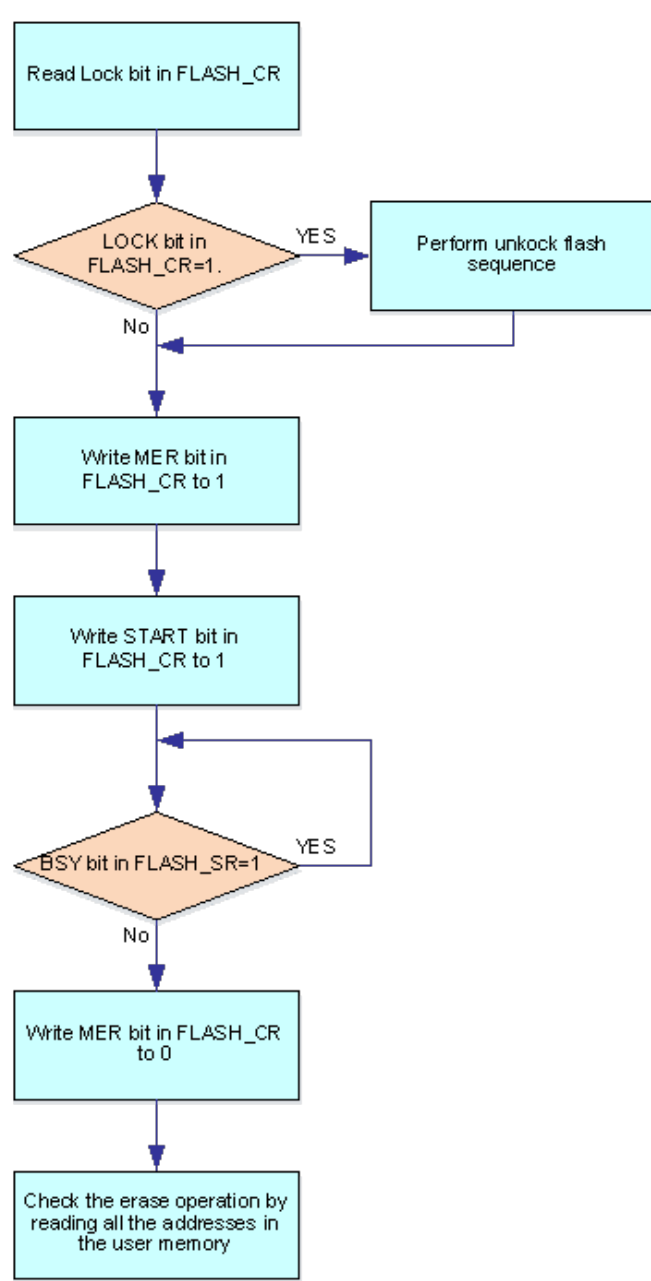
```
#define FLASH_CR_OPTWRE_Pos      (9)
#define FLASH_CR_OPTWRE        (0x01U << FLASH_CR_OPTWRE_Pos)
void FLASH_OPT_Lock(void)
{
    FLASH->CR &= ~FLASH_CR_OPTWRE;
}
```

2.3.5 Main Flash block erasing

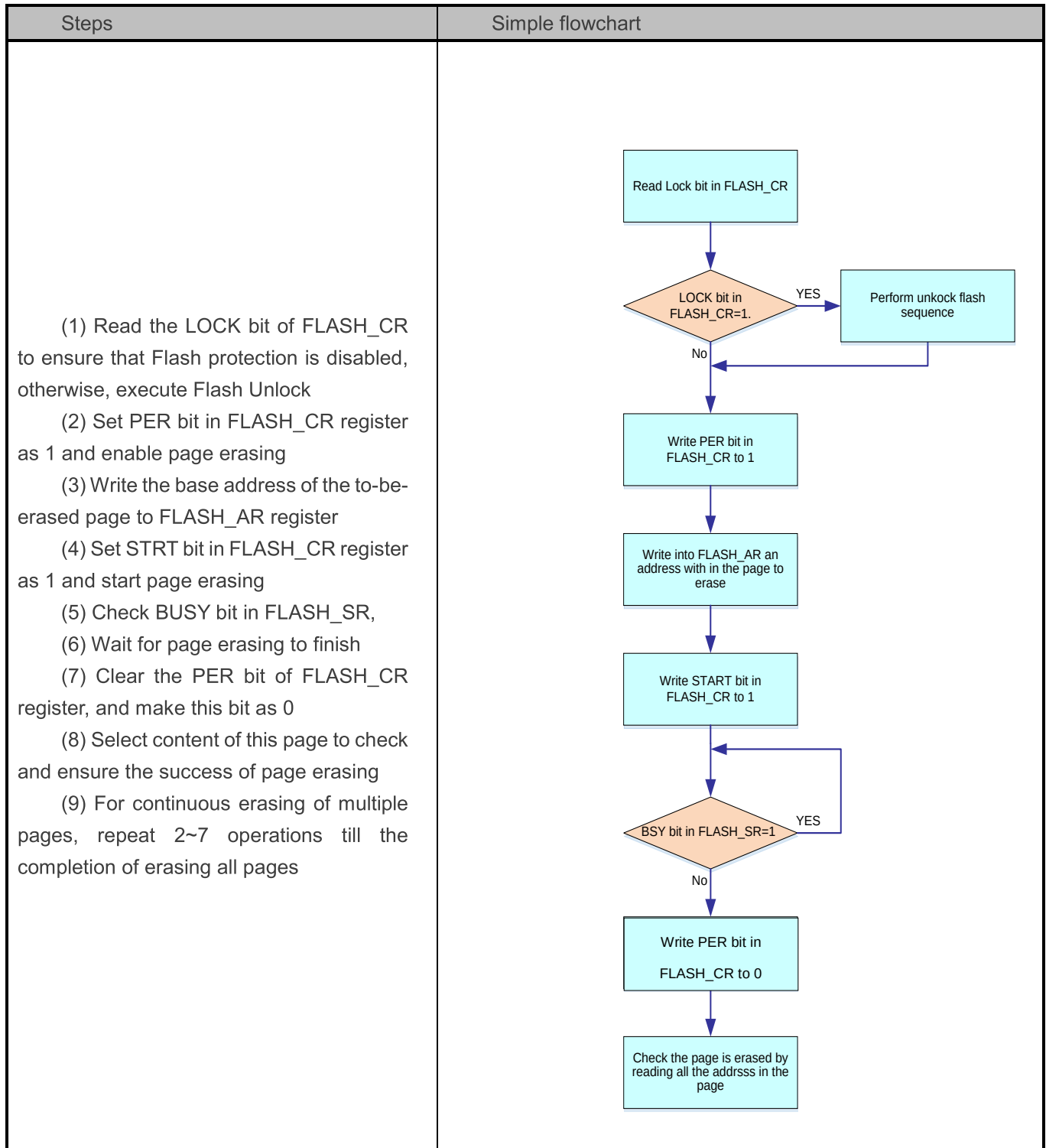
The Flash memory controller supports the whole chip erasing of main Flash block and the erasing of pages inside the main Flash memory in the unit of page.

The whole chip erasing function will initialize all the contents of the main Flash memory block and set all values as 1. However, the information block will not be affected by this instruction.

Main Flash memory Erase procedure:

Steps	Simple flowchart
(1) Read the LOCK bit of FLASH_CR to ensure that Flash protection is disabled, (2) Otherwise execute Flash Unlock (3) Set MER bit in the FLASH_CR register as 1 and enable the whole chip erasing of main Flash (4) Set STRT bit in the FLASH_CR register as 1 (5) Start the whole chip erase of the main Flash (6) Check BUSY bit in FLASH_SR (7) Wait for completion of whole erase of main Flash (8) Clear the MER bit of FLASH_CR register, and make this bit as 0 (9) Optionally conduct Blank check to ensure erasing success	 <pre> graph TD A[Read Lock bit in FLASH_CR] --> B{LOCK bit in FLASH_CR=1} B -- YES --> C[Perform unlock flash sequence] C --> D[Write MER bit in FLASH_CR to 1] B -- No --> D D --> E[Write START bit in FLASH_CR to 1] E --> F{BSY bit in FLASH_SR=1} F -- YES --> F F -- No --> G[Write MER bit in FLASH_CR to 0] G --> H[Check the erase operation by reading all the addresses in the user memory] </pre> The flowchart illustrates the sequence for erasing the main Flash. It begins with reading the LOCK bit in the FLASH_CR register. If the LOCK bit is 1, a flash unlock sequence is performed. Following this, the MER bit in FLASH_CR is set to 1, and the START bit is also set to 1. A loop then checks the BSY bit in the FLASH_SR register. If BSY is 1, the process waits. Once BSY becomes 0, the MER bit in FLASH_CR is cleared to 0. Finally, the erase operation is verified by reading all addresses in the user memory.

Flash memory Page Erase procedure:



2.3.6 Main Flash block programming

The main Flash memory supports 32-bit whole word programming to change the content of main storage Flash memory block. If it is programmed with a length of 8-bit byte or 16-bit half-word, a hardware error interrupt is caused. At the same time, this programming operation was cancelled. When the PG bit of FLASH_CR is 1, a word (32 bits) can be directly written to the corresponding address, namely one programming operation.

The main Flash memory controller will pre-read whether the word to be programmed is whole 1 (whether it is 0xFFFFFFFF); if not, the programming will be automatically canceled and a programming error warning will be displayed on the PGERR bit of FLASH_SR register.

If the write protection bit in FLASH_WRRP of the corresponding write protection block of the address to be programmed is valid, there will be no programming action; but a programming error warning is generated. It will also produce programming error alarm. After programming, EOP bit in FLASH_SR register will produce the prompt.

Note: When the CPU enters the power saving mode, an error will be produced to the Flash memory operation via SWD interface. It is necessary to avoid running interrupt programs while perform erasing or programming operations.

Specific steps for the register setting of main Flash block programming:

Steps	Simple flowchart
(1) Read the LOCK bit of FLASH_CR to ensure that Flash protection is disabled, otherwise, execute Flash Unlock (2) Set PG bit in FLASH_CR register as 1 and enable 32-bit word programming (3) Write data to the target address in a unit of 32-bit word, and align the target address with 32-bit word (4) Check BUSY bit in FLASH_SR (5) Wait for 32-bit word programming to finish (6) Clear the PG bit of FLASH_CR register, and make this bit as 0 (7) Select to read target address data to ensure the success of 32-bit word programming (8) For more continuous 32-bit word programming, repeat 2~6 operations, till the completion of 32-bit word programming of all target addresses	Programming procedures <pre> graph TD A[Read Lock bit in FLASH_CR] --> B{LOCK bit in FLASH_CR=1} B -- YES --> C[Perform unlock flash sequence] C --> D[Write PG bit in FLASH_CR to 1] B -- No --> D D --> E[Perform whole-word write at the desired address] E --> F{BSY bit in FLASH_SR=1} F -- YES --> E F -- No --> G[Write PG bit in FLASH_CR to 0] G --> H[Check the programmed value by reading the programmed address] </pre>

2.3.7 Option byte block erasing

Note: If OPTER is high while MER or PER is also high, the option byte area will not be erased.

Specific steps for the register setting of option byte block erasing:

Steps	Simple flowchart
(1) Read the LOCK bit of FLASH_CR to ensure that Flash protection is disabled; otherwise, execute Flash Unlock (2) Read the OPTWRE bit of FLASH_CR to ensure that option block protection is disabled. Otherwise, perform the option block unlock operation. (3) Write block base address to be erased to FLASH_AR register (4) Set OPTER bit in FLASH_CR register as 1 and enable option byte block erasing (5) Set STRT bit in FLASH_CR register as 1 (6) Check BUSY bit in FLASH_SR (7) Wait for block erasing to finish (8) Clear the OPTER bit of FLASH_CR register, and set the value as 0 (9) Select to check the block content to ensure the success of the block erasing (10) For continuous erasing of multiple blocks, repeat 3~8 operations, till the completion of erasing all option blocks	<pre> graph TD A[Read LOCK bit in FLASH_CR] --> B{LOCK bit in FLASH_CR=1} B -- Yes --> C[Perform unlock sequence] B -- No --> D{OPTWRE bit in FLASH_CR=0} C --> D D -- Yes --> E[Execute the OPT unlock sequence.] D -- No --> F[Write the base address of the Flash option byte area to FLASH_AR.] F --> G[Set OPTER bit in FLASH_CR to 1] G --> H[Set STRT bit in FLASH_CR to 1] H --> I{BSY bit in FLASH_SR=1} I -- Yes --> H I -- No --> J[Write OPTER bit in FLASH_CR to 0] J --> K[Verify the flash option erase operation by reading all addresses in the flash option area] </pre>

2.3.8 Option byte block programming

The programming of the option byte block is different from that of the main Flash memory block address, because the written value after reset is loaded to the configuration option, which requires more stringent protection. After the access restriction on the Flash memory controller is removed, the access restriction on the option byte block and read protection area should also be disabled. After this operation is completed, the OPTWRE bit in the FLASH_CR register will be set as 1 before the follow-up programming is allowed.

The valid data of option byte is the low 8 bits. During programming, the software sets the upper 8 bits of the lower 16-bit word to the bitwise inverse of the lower 8 bits to ensure the value written to the option byte is always correct; the upper 16 bits of the 32-bit word are reserved and require no processing.

The change of optional byte is only valid after the system is powered-on and reset.

Option byte block Programming procedure:

Steps	Simple flowchart
(1) Read the LOCK bit of FLASH_CR to ensure that the Flash protection is disabled. Otherwise, execute Flash Unlock (2) Unlock Read the OPTWRE bit of FLASH_CR to ensure that the option block protection is disabled. Otherwise, execute option block Unlock (3) Set OPTPG bit of FLASH_CR register as 1 and enable option byte block 32-bit word programming (4) Write data (32-bit word) to target address and align target address by 32-bit word (5) Check BUSY bit of FLASH_SR (6) Wait for the completion of 32-bit word programming (7) Clear the OPTPG bit of FLASH_CR register, and set the value as 0 (8) Select to read target address data and ensure the success of 32-bit word programming (10) For continuous programming of multiple 32-bit words, repeat 3~7 operations, till the completion of 32-bit word programming of all target addresses	<pre> graph TD A[Read LOCK bit in FLASH_CR] --> B{LOCK bit in FLASH_CR=1} B -- Yes --> C[Perform unlock sequence] B -- No --> D{OPTWRE bit in FLASH_CR=0} C --> D D -- Yes --> E[Execute the OPT unlock sequence.] D -- No --> F[Write the base address of the Flash option byte area to FLASH_AR.] E --> F F --> G[Set OPTER bit in FLASH_CR to 1] G --> H[Set STRT bit in FLASH_CR to 1] H --> I{BSY bit in FLASH_SR=1} I -- Yes --> H I -- No --> J[Write OPTER bit in FLASH_CR to 0] J --> K[Verify the flash option erase operation by reading all addresses in the flash option area] </pre>

2.3.9 Flash memory protection

The main Flash block enable read protection can prevent the codes of the main Flash block from being read by unreliable codes and the accidental erasing and programming of main Flash memory block during program

runaway. The read protection enable operation works for the whole main memory block while the minimum unit of enabling write protection is a write protection block (namely 4 pages).

2.3.9.1 Main Flash block read protection

Enabling or disabling read protection for the main flash memory block is achieved by configuring the RDP setting in the option bytes; the change takes effect after a system reset, which loads the new RDP configuration.

Enable read protection

According to the operation mode of 32-bit word programming in the option byte area, write in sequence the word of RDP to the corresponding address.

- Set the address value of FLASH_AR as 0x1FFFF800, and execute erasing of the option block.
- According to the operation mode of 32-bit word programming in the option byte area, write in sequence the word of 0x0000807F to the corresponding address.
- The system is reset to reload the option bytes, at which point read protection is enabled.

When RDP word includes the following values, the main Flash block is placed in the protected status after being re-powered on and reset.

Table 2-9 Flash read protection status

Enable read protection	Read protection status
Erase 0x1FFFF800 option block Write 0x0000807F 32-bit word to the corresponding address 0x1FFFF800 Enable read protection after being re-powered on and reset	Protection

When the read protection RDP is written within the corresponding value:

1. Only read operations to the main Flash memory from user code (start from main Flash memory in non-debug mode) are allowed.
2. After read protection, operations to the Flash are disabled in the debug mode (SRAM boot and debug mode).
3. MCU can be programmed by the code executed in the main Flash memory (to implement functions such as IAP or data storage), however, write or page erase operations on the main flash memory block (with the exception of a mass erase) are not permitted in debug mode or after booting from internal SRAM.
4. All functions of loading and executing code to the built-in SRAM through SWD remain valid or it can also boot from the built-in SRAM. This function can be used to disable read protection.
5. Access to the main Flash memory by executing code from the built-in SRAM or access to the Flash via SWD (Serial Wire Debug) will all be disabled.

Disable read protection

Disable read protection from built-in SRAM or ICP mode:

1. Set FLASH_AR address value as 0x1FFFF800 and execute the operation block erase.
2. According to the operation mode of 32-bit word programming in the option byte area, write in sequence 32-bit words of 0x00005AA5 to the corresponding address.
3. Apply a POR reset to reload the option bytes, to disable the read protection.

Table 2-10 Flash disable read protection status

Disable read protection status	Read protection status
Erase 0x1FFFF800 option block Write 0x00005AA5 32-bit word to the corresponding address 0x1FFFF800 Erase 0x08000000 main Flash whole chip	Disable read protection

Disable read protection status	Read protection status
Power on and reset, disable read protection	

Note: If the corresponding address value of the option byte block is not 0xFFFFFFFF, execute the erasing of option byte block first, which will not lead to an automatic erasing of the whole chip or change the read protection status.

2.3.9.2 Main Flash block write protection

The write protection is controlled in a unit of sector (4 pages). The WRP bit in the option bytes should be configured, and then the system reset will load the new option bytes to enable the protection. If a write or an erase operation is performed on a protected sector. The WRPRERR flag bit is set on the FLASH_SR.

Table 2-11 Write protection area

Address	[15: 8]	[7: 0]	Default value	Remark
0x1FFF F810	nWRP0	WRP0	0x00FF	
0x1FFF F814	nWRP1	WRP1	0x00FF	

Disable write protection

There are two scenarios for disabling the write protection:

1. Scenario 1: Disable write and read protection:

a. Use the OPTER bit in the Flash control register (FLASH_CR) to erase the whole option byte block; write the word of 0x00005AA5 to the corresponding address 0x1FFFF800; and trigger a whole-chip erasing on the main Flash block of 0x08000000;

b. Reset the system, and reload the option byte (including the new WRP byte); write protection is disabled. Use of this method will disable the write protection of the whole chip main Flash module and erase the whole chip of main Flash block.

2. Scenario 2: Disable write protection and keep the read protection valid. This scenario is commonly seen in the user's startup program:

a. Use OPTER bit of Flash control register (FLASH_CR) to only erase the whole option byte block;

b. Reset the system and reload option byte (including the new WRP byte); write protection is disabled.

Use of this method will disable the write protection of the whole main Flash module and keep the read protection valid.

2.4 Register

2.4.1 Register overview

Table 2-12 FLASH register overview

Offset	Acronym	Register Name	Reset
0x00	FLASH_ACR	Flash access control register	0x00000038
0x04	FLASH_KEYR	FPEC key register	0x00000000
0x08	FLASH_OPTKEYR	Flash OPTKEY register	0x00000000
0x0C	FLASH_SR	Flash status register	0x00000000
0x10	FLASH_CR	Flash control register	0x00000080
0x14	FLASH_AR	Flash address register	0x00000000
0x1C	FLASH_OBR	Option byte register	0x1BFFFC0E
0x20	FLASH_WRPR	Write protection register	0x0000FFFF

Note: Flash register only supports 32-bit access

2.4.2 FLASH_ACR Flash Access Control Register

Offset address: 0x00

Reset value: 0x0000 0038

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.										PRFTBS	PRFTBE	Res.	LATENCY		
										r	rw		rw		

Bit	Field	Description
31 : 6	Res.	Reserved, must retain the reset value
5	PRFTBS	Prefetch buffer status 0: Prefetch buffer close. 1: Prefetch buffer open.
4	PRFTBE	Prefetch buffer enable 0: Close prefetch buffer. 1: Open prefetch buffer. Note 1: Only when LATENCY is set to 0, can the prefetch buffer be controlled through this bit.
3	Res.	Reserved, must retain the reset value
2 : 0	LATENCY	Latency These bits indicate the ratio of HCLK (AHB clock) period to Flash memory access time. 000: zero wait state, $0 < \text{HCLK} \leq 24\text{MHz}$ 001: one wait state, $24\text{MHz} < \text{HCLK} \leq 48\text{MHz}$ 010: two wait state, $48\text{MHz} < \text{HCLK} \leq 72\text{MHz}$

2.4.3 FLASH_KEYR FPEC Key Register

Offset address: 0x04

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FKEYR															
w															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FKEYR															
w															

Bit	Field	Description
31 : 0	FKEYR	FPEC Flash key These bits are used to input FPEC unlock key.

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

2.4.4 FLASH_OPTKEYR Flash OPTKEY Register

Offset address: 0x08

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OPTKEYR															
W															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OPTKEYR															
W															

Bit	Field	Description
31 : 0	OPTKEYR	Option byte key These bits are used as the input option byte key to disable OPTWRE.

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

2.4.5 FLASH_SR Flash Status Register

Offset 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
Res.										EOP	WRPRERR	Res.	PGERR	Res.	BSY
										rw	rw		rw		r

Bit	Field	Description
31:6	Reserved	Reserved, must retain the reset value
5	EOP	End of operation When the Flash memory operation (programming / erasing) is completed, the hardware sets this bit as "1" and write "1" to clear the bit status. Note: EOP status is set at the end of each successful program or erase operation.
4	WRPRERR	Write protection error When an attempt is made to program the write-protected Flash memory address, the hardware sets this bit as "1" and write "1" to clear the bit status.
3	Res.	Reserved, must retain the reset value
2	PGERR	Programming error When an attempt is made to program an address whose content is not "0x FFFF", the hardware sets this bit as "1" and write "1" to clear the bit status Note: Prior to programming, clear the STRT bit of FLASH_CR register. Note: When the PGERR bit is 1, the CPU cannot be cleared and cannot be written again. It can only be reset.
1	Res.	Reserved, must retain the reset value
0	BSY	Busy This bit indicates that the operation of Flash memory is in progress. Before Flash operation starts, this bit is set as "1". After the operation or in case of an error, this bit is cleared as "0".

2.4.6 FLASH_CR Flash Control Register

Offset address: 0x10

Reset value: 0x0000 0080

31	30	29	28	28	27	26	25	24	23	22	21	20	19	18	17
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.			EOPIE	Res.	ERRIE	OPTWRE	Res.	LOCK	STRT	OPTER	OPTPG	Res.	MER	PER	PG
			rw		rw	rw		rw	rw	rw	rw		rw	rw	rw

Bit	Field	Description
31:10	Res.	Reserved, must retain the reset value
12	EOPIE	End of operation interrupt enable This bit enables the interrupt generation when the EOP bit in the FLASH_SR register goes to '1'. 0: interrupt generation disabled 1: interrupt generation enabled
11	Res.	Reserved, must retain the reset value
10	ERRIE	Error interrupt enable This bit enables the interrupt generation on an FPEC error (when PGERR/WRPRTERR are set to '1' in the FLASH_SR register). 0: interrupt generation disabled 1: interrupt generation enabled
9	OPTWRE	Option byte write enable When this bit is "1", the option byte is allowed to be program. When the correct key sequence is written in the FLASH_OPTKEYR register, the bit is set as "1". This bit can be cleared by writing 0.
8	Res.	Reserved, must retain the reset value
7	LOCK	Lock Only write "1". When the bit is "1", it indicates that FPEC and FLASH_CR are locked. When the correct unlocking sequence is detected, the hardware automatically clears this bit as "0". After an unsuccessful unlocking operation, this bit cannot be changed before the next system reset.
6	STRT	Start An erasing is triggered when this bit is "1". The bit can only be set to "1" by software and automatically be cleared to "0" when BSY turns into "1".
5	OPTER	Option byte erase Erase option byte.
4	OPTPG	Option byte programming Program option byte.
3	Res.	Reserved, must retain the reset value
2	MER	Mass erase Choose to erase all user pages.
1	PER	Page erase Choose to erase page.
0	PG	Programming Choose to conduct programming.

2.4.7 FLASH_AR Flash Address Register

Offset address: 0x14

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FAR															
w															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FAR															
w															

Bit	Field	Description
31 : 0	FAR	Flash Address Select the page to be erased when performing page erasing. Note: When the BSY bit in FLASH_SR is “1”, the register cannot be written.

The address is modified by the hardware to the currently used one. During the operation of page erasing, the register is modified to specify the page to be erased.

2.4.8 FLASH_OBR Option Byte Register

Offset address: 0x1C

Reset value: 0x1BFF FC0E

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.			OBR_PLS		OBR_PVDE	Data1								Data0	
r			r		r	r								r	
Data0						Res.						nRST_STOP	WDG_SW	RDPRT	OPTERR
r												r	r	r	r

Bit	Field	Description
31 : 29	Res.	Reserved, must retain the reset value
28: 27	OBR_PLS	PVD monitoring level selection bits [1:0]
26	OBR_PVDE	0: PVD disable 1: PVD enable
25 : 18	Data1	Data1
17 : 10	Data0	Data0
9: 4	Res.	Res.
3	nRST_STOP	Reset event in stop mode 0: Reset in stop mode 1: No reset in stop mode
2	WDG_SW	Select watchdog event 0: Hardware watchdog 1: Software watchdog
1	RDPRT	Read protection level status When setting to “1”, it indicates that the Flash memory is read protected. Note: The bit is read-only.
0	OPTERR	Option byte error When this bit is “1”, it indicates that the option byte does not match its inverse code. Note: The bit is read-only.

2.4.9 FLASH_WRP Write Protection Register

Offset address: 0x20

Reset value: 0x0000 FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
WRP1								WRP0							
r								r							

Bit	Field	Description
31:16	Reserved	Reserved, must retain the reset value

Bit	Field	Description
15:0	WRP	Write protection The register includes the write protection option byte loaded by OBL. 0: Write protection enabled 1: Write protection disabled

3 CRC Cyclic Redundancy Check Calculation Unit Introduction

The CRC calculation unit utilizes a fixed polynomial to compute the CRC checksum value of 32-bit data, used to verify the integrity of data transmission or storage.

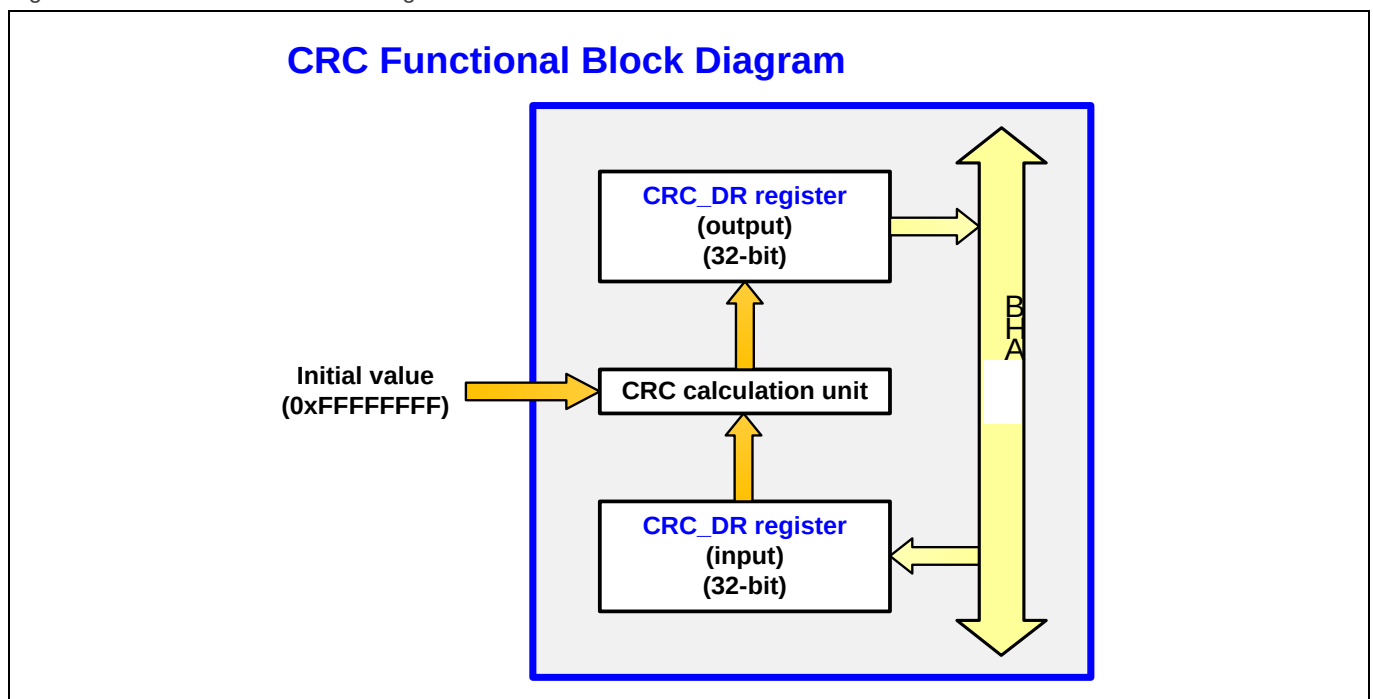
3.2 Key Features

- Supports CRC-32/MPEG-2 (Ethernet) polynomial: $0x4C11DB7$
 $X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$
- Supports 32-bit wide data register for input/output
- Hardware computation time is 3 HCLK cycles
- 8-bit independent data register for storing temporary data

3.3 Function Description

3.3.1 Functional Block Diagram

Figure 3-1 CRC Functional Block Diagram



3.3.2 Functional Overview

- The CRC calculation unit contains one 32-bit data register:
- When written to, it acts as an input register, accepting new data for CRC calculation.
- When read from, it returns the result of the previous CRC calculation.
- Each write to the data register combines the previous CRC result with the new computation result (CRC calculation is done on the entire 32-bit word, not byte by byte).
- During CRC calculation, write operations are paused, allowing for consecutive writes or continuous write-read operations on register CRC_DR.
- The data register CRC_DR can be reset to 0xFFFFFFFF by setting the RST bit in register CRC_CR. This operation does not affect the data inside register CRC_IDR.

3.3.3 Usage

3.3.3.1 CRC Calculation Operation Steps

- Enable the clock to the CRC module.
- Reset the CRC module.

- Restore the CRC to its initial state by configuring the RST bit in the CRC control register (CRC_CR).
- Write data sequentially into the CRC data register (CRC_DR).
- Read the CRC data register (CRC_DR) to obtain the CRC calculation result.

3.4 Register

3.4.1 Register Overview

Table 3-1 CRC Register Overview

Offset	Acronym	Register Name	Reset
0x00	CRC_DR	CRC Data Register	0xFFFFFFFF
0x04	CRC_IDR	CRC Independent Data Register	0x00000000
0x08	CRC_CR	CRC Control Register	0x00000000

3.4.2 CRC_DR CRC Data Register

Offset Address: 0x00

Reset Value: 0xFFFF FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DR															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DR															
rw															

Bit	Field	Description
31 : 0	DR	Data Register When written to, acts as an input register, performing CRC calculation on the written data and the previous result. When read from, returns the result of the CRC calculation.

3.4.3 CRC_IDR CRC Independent Data Register

Offset Address: 0x04

Reset Value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								IDR							
								rw							

Bit	Field	Description
31 : 8	Reserved	Reserved, must retain the reset value.
7 : 0	IDR	General-purpose 8-bit Data Register Can be used as a temporary storage location for one byte. This register is not affected by the RST bit in the CRC control register (CRC_CR).

3.4.4 CRC_CR CRC Control Register

Offset Address: 0x08

Reset Value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved														RST	
														W	

Bit	Field	Description
31 : 1	Reserved	Reserved, must be kept at reset value
0	RST	CRC Reset CRC Data Register (CRC_DR) is reset to 0xFFFFFFFF. This bit can only be written as "1," and hardware automatically clears it to "0".

4 Chip-specific ConfigurationInterrupt vector table

Under Handler mode, Cortex®-M0+ processor and Nested Vectored Interrupt Controller (NVIC) can process all anomalies based on priority. In case of anomaly, the system will pop the current processing work stack after interrupt service. Simultaneous processing of vector and current work status improves the interrupt efficiency. The table below presents the anomaly type and interrupt vector.

Table 4-1 Abnormal vector

Position	Priority	Priority type	Name	Description	Address
-	-	-	-	Reserved	0x0000 0000
	-3	Fixed	Reset	Reset	0x0000 0004
	-2	Fixed	NMI	Non-maskable interrupt	0x0000 0008
	-1	Fixed	Hard Fault	All types of faults	0x0000 000C

Table 4-2 Interrupt vector

Position	Priority	Priority type	Name	Description	Address
	3	Settable	SVCall	System service calls via the SWI instruction	0x0000 002C
				Reserved	0x0000 0030
				Reserved	0x0000 0034
	5	Settable	PendSV	Pending system service	0x0000 0038
	6	Settable	SysTick	System Tick Timer	0x0000 003C
0	7	Settable	IWDG	Independent watchdog interrupt connected to EXTI17	0x0000 0040
1	8	Settable	PVD	Power Voltage Detection (PVD) connected to EXTI16	0x0000 0044
2	9	Settable	Reserved	Reserved	0x0000 0048
3	10	Settable	Flash	Flash global interrupt	0x0000 004C
4	11	Settable	RCC	RCC global interrupt	0x0000 0050
5	12	Settable	EXTI0_1	EXTI Line [1:0] interrupt	0x0000 0054
6	13	Settable	EXTI2_3	EXTI Line [3:2] interrupt	0x0000 0058
7	14	Settable	EXTI4_15	EXTI Line [15:4] interrupt	0x0000 005C
8	15	Settable	Reserved	Reserved	0x0000 0060
9	16	Settable	Reserved	Reserved	0x0000 0064
10	17	Settable	Reserved	Reserved	0x0000 0068
11	18	Settable	Reserved	Reserved	0x0000 006C
12	19	Settable	ADC1_COMP	ADC1 global interrupt Comparator global interrupt	0x0000 0070
13	20	Settable	TIM1_BRK_UP_TRG_COM	TIM1 brake refresh trigger COM interrupt	0x0000 0074
14	21	Settable	TIM1_CC	TIM1 Capture Compare Interrupt	0x0000 0078
15	22	Settable	Reserved	Reserved	0x0000 007C
16	23	Settable	TIM3	TIM3 global interrupt	0x0000 0080
17	24	Settable	Reserved	Reserved	0x0000 0084
18	25	Settable	Reserved	Reserved	0x0000 0088
19	26	Settable	TIM14	TIM14 global interrupt	0x0000 008C

Position	Priority	Priority type	Name	Description	Address
20	27	Settable	Reserved	Reserved	0x0000 0090
21	28	Settable	Reserved	Reserved	0x0000 0094
22	29	Settable	Reserved	Reserved	0x0000 0098
23	30	Settable	I2C1	I2C1 global interrupt	0x0000 009C
24	31	Settable	Reserved	Reserved	0x0000 00A0
25	32	Settable	SPI1	SPI1 global interrupt	0x0000 00A4
26	33	Settable	Reserved	Reserved	0x0000 00A8
27	34	Settable	USART1	USART1 global interrupt	0x0000 00AC
28	35	Settable	USART2	USART2 global interrupt	0x0000 00B0
29	36	Settable	Reserved	Reserved	0x0000 00B4
30	37	Settable	Reserved	Reserved	0x0000 00B8
31	38	Settable	Reserved	Reserved	0x0000 00BC

4.2 External interrupt mapping

The 16 external interrupt/event mapping relationships corresponding to the GPIOs are shown in the following table:

Table 4-3 EXTI trigger source

External interrupt line	I/O mapping	Control Bit
EXTI0	PX0 (X=A, B)	SYSCFG_EXTICR1 register EXTI0
EXTI1	PX1 (X=A, B)	SYSCFG_EXTICR1 register EXTI1
EXTI2	PX2 (X=A, B)	SYSCFG_EXTICR1 register EXTI2
EXTI3	PX3 (X=A, B)	SYSCFG_EXTICR1 register EXTI3
EXTI4	PX4 (X=A, B)	SYSCFG_EXTICR2 register EXTI4
EXTI5	PX5 (X=A, B)	SYSCFG_EXTICR2 register EXTI5
EXTI6	PX6 (X=A)	SYSCFG_EXTICR2 register EXTI6
EXTI7	PX7 (X=A)	SYSCFG_EXTICR2 register EXTI7
EXTI8	PX8 (X=A)	SYSCFG_EXTICR3 register EXTI8
EXTI9	PX9 (X=A)	SYSCFG_EXTICR3 register EXTI9
EXTI10	PX10 (X=A)	SYSCFG_EXTICR3 register EXTI10
EXTI11	PX11 (X=A)	SYSCFG_EXTICR3 register EXTI11
EXTI12	PX12 (X=A)	SYSCFG_EXTICR4 register EXTI12
EXTI13	PX13 (X=A)	SYSCFG_EXTICR4 register EXTI13
EXTI14	PX14 (X=A)	SYSCFG_EXTICR4 register EXTI14
EXTI15	PX15 (X=A)	SYSCFG_EXTICR4 register EXTI15

Other external interrupt/event controller connection:

- EXTI line 16 connected to PVD output.
- EXTI line 17 connected to IWDG output.
- EXTI line 19 connected to COMP1 output.

4.3 ADC Channel Assignment

Table 4-4 ADC Channel Assignment

Conversion input channel	ADC1 channel assignment
Channel 0	ADC_IN0

Conversion input channel	ADC1 channel assignment
Channel 1	ADC_IN1
Channel 2	ADC_IN2
Channel 3	ADC_IN3
Channel 4	ADC_IN4
Channel 5	ADC_IN5
Channel 6	ADC_IN6
Channel 7	ADC_IN7
Channel 8	ADC_IN8
Channel 9	ADC_IN9
Channel 10	ADC_IN10
Channel 11	ADC_IN11
Channel 12	Built-in T-Sensor / Built-in V-Sensor

ADC channel 12 can be switched by configuring SYSCFG_SENSORCR.VS_EN and SYSCFG_SENSORCR.TS_EN.

4.4 System Module Hardware Interconnect

Table 4-5 Interconnection of ADC trigger source

Source	Register setting	Target (ADC trigger input)
TIM1_CC1	ADC1_ADCR.TRGSEL=0x0	TRG0
TIM1_CC2	ADC1_ADCR.TRGSEL=0x1	TRG1
TIM1_CC3	ADC1_ADCR.TRGSEL=0x2	TRG2
Reserved	ADC1_ADCR.TRGSEL=0x3	TRG3
TIM3_TRGO	ADC1_ADCR.TRGSEL=0x4	TRG4
TIM1_CC4 XOR TIM1_CC5	ADC1_ADCR.TRGSEL=0x5	TRG5
TIM3_CC1	ADC1_ADCR.TRGSEL=0x6	TRG6
EXTI Line [11]	ADC1_ADCR.TRGSEL=0x7	TRG7
TIM1_TRGO	ADC1_ADCR.TRGSEL=0x8	TRG8
Reserved	ADC1_ADCR.TRGSEL=0x9	TRG9
Reserved	ADC1_ADCR.TRGSEL=0xA	TRG10
Reserved	ADC1_ADCR.TRGSEL=0xB	TRG11
TIM3_CC4	ADC1_ADCR.TRGSEL=0xC	TRG12
Reserved	ADC1_ADCR.TRGSEL=0xD	TRG13
Reserved	ADC1_ADCR.TRGSEL=0xE	TRG14
EXTI Line [15]	ADC1_ADCR.TRGSEL=0xF	TRG15
TIM1_CC4	ADC1_ADCR.TRGSEL=0x10	TRG16
TIM1_CC5	ADC1_ADCR.TRGSEL=0x11	TRG17

For TIMx_CCy (the compare pulse of TIMx), additional interconnect processing is performed when triggering the ADC:

- When TIMx_CR1.CMS is set to 0x3 (Center-aligned mode 3), any compare pulse toggles the corresponding ADC trigger input (TRG) once.
- When TIMx_CR1 is in other counting modes, a compare event generated during up-counting clears the corresponding ADC trigger input (TRG), whereas a compare event generated during down-counting sets the corresponding ADC trigger input (TRG).

For TIMx_TRGO (the TRGO output of TIMx), additional interconnect processing is performed when triggering the ADC:

- When TIMx_CR2.MMS is set to 0x0 (Reset mode), the corresponding ADC trigger input (TRG) toggles each time TIMx_SR.UG is set.
- When TIMx_CR2.MMS is set to 0x1, the corresponding ADC trigger input (TRG) is equivalent to TIMx_CR1.CEN.
- When TIMx_CR2.MMS is set to 0x2 (Update mode), any update event toggles the corresponding ADC trigger input (TRG).
- When TIMx_CR2.MMS=0x3 is selected for the Channel 1 compare pulse, the handling of the interconnection between the ADC trigger input (TRG) and TIMx_CCy is the same.
- When TIMx_CR2.MMS=0x4 selects OC1REF, the corresponding ADC trigger input (TRG) is equivalent to the TIMx OC1REF signal.
- When TIMx_CR2.MMS = 0x5 selects OC2REF, the corresponding ADC trigger input (TRG) is equivalent to the TIMx OC1REF.
- When TIMx_CR2.MMS=0x6 selects OC3REF, the corresponding ADC trigger input (TRG) is equivalent to the TIMx OC1REF.
- When TIMx_CR2.MMS = 0x7, OC4REF is selected; the corresponding ADC trigger input (TRG) is triggered by TIMx's OC1REF.

Table 4-6 Comparator Output Direction Selection

Source	Register setting	Output direction	Target
COMPx_OUT (x=1)	COMPx_CSR:OUT_SEL=0x0 (x=1)	Direction 0	Reserved
	COMPx_CSR:OUT_SEL=0x1 (x=1)	Direction 1	TIM3 Input Capture 2
	COMPx_CSR:OUT_SEL=0x2 (x=1)	Direction 2	TIM1 Brake Input
	COMPx_CSR:OUT_SEL=0x3 (x=1)	Direction 3	Reserved
	COMPx_CSR:OUT_SEL=0x4 (x=1)	Direction 4	Reserved
	COMPx_CSR:OUT_SEL=0x5 (x=1)	Direction 5	Reserved
	COMPx_CSR:OUT_SEL=0x6 (x=1)	Direction 6	TIM1 Ocrefclear input
	COMPx_CSR:OUT_SEL=0x7 (x=1)	Direction 7	Reserved
	COMPx_CSR:OUT_SEL=0x8 (x=1)	Direction 8	Reserved
	COMPx_CSR:OUT_SEL=0x9 (x=1)	Direction 9	Reserved
	COMPx_CSR:OUT_SEL=0xA (x=1)	Direction 10	TIM3 Input Capture 1
	COMPx_CSR:OUT_SEL=0xB (x=1)	Direction 11	TIM3 Ocrefclear input
	COMPx_CSR:OUT_SEL=0xC (x=1)	Direction 12	Reserved
	COMPx_CSR:OUT_SEL=0xD (x=1)	Direction 13	Reserved
	COMPx_CSR:OUT_SEL=0xE (x=1)	Direction 14	Reserved
	COMPx_CSR:OUT_SEL=0xF (x=1)	Direction 15	Reserved

Table 4-7 Interconnection of TIM ITR Event

	TIM1	TIM3
ITR0	Reserved	TIM1
ITR1	Reserved	Reserved
ITR2	TIM3	Reserved
ITR3	Reserved	Reserved

Note: The TIMx_SMCR.TS bits in the TIMx slave mode control register control the selection of the internal trigger source (where x = 1, 3).

Table 4-8 Interconnection of TIM ETR Event

Source	Register selection	Target
ETR GPIO input	TIMx_OR.ETR_RMP=00b (x=3)	TIMx_ETR (x=3)
LSI_CLK	TIMx_OR.ETR_RMP=01b (x=3)	
Reserved	TIMx_OR.ETR_RMP=10b (x=3)	
HSE_CLK_DIV_128	TIMx_OR.ETR_RMP=11b (x=3)	

Table 4-9 Interconnection of Input Capture Event

Source	Register setting	Target
COMP1_OUT	COMP1_CSR.OUT_SEL=0xA	TIM3_CH1_IN
COMP1_OUT	COMP1_CSR.OUT_SEL=0x1	TIM3_CH2_IN
CH4 GPIO input	TIM3_OR.TI4_RMP=0x0	TIM3_CH4_IN
LSI_CLK	TIM3_OR.TI4_RMP=0x1	TIM3_CH4_IN
HSE_CLK_DIV_128	TIM3_OR.TI4_RMP=0x3	TIM3_CH4_IN

Table 4-10 Interconnection of TIM OCREFCLR Event

Source	Register setting	Target
COMP1_OUT	COMP1_CSR.OUT_SEL=0x6	TIM1_OCREFCLR
COMP1_OUT	COMP1_CSR.OUT_SEL=0xB	TIM3_OCREFCLR

Table 4-11 Interconnection of TIM Break Event

Source	Register setting	Target
COMP1_OUT	COMP1_CSR.OUT_SEL=0x2	TIM1_COMPBKIN
IO	TIM1_BKINF.IOBKIN_SEL=0x01	TIM1_IOBKIN

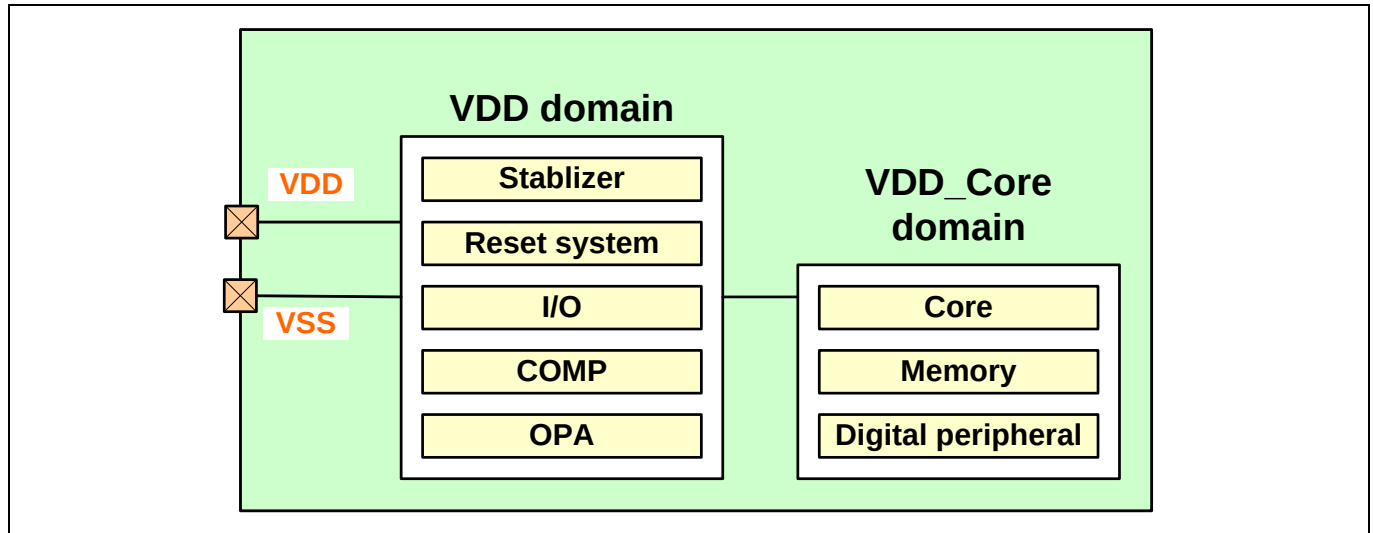
5 PWR Power Controller

Power controller (PWR) mainly involves features such as chip's power supply, power manager and low-power mode.

5.1 Power supply system

The chip is powered by four types of power supply below:

Figure 5-1 Power supply block diagram



- Power supplied via VDD and VSS is used for the operation of the analog section, digital section, and I/O pins.
- In the power supply system, external 10μF and 100nF capacitors are required at the corresponding power pins, and they should be placed as close to the pins as possible.

5.1.1 VDD domain

The VDD domain primarily supplies power to the analog sections of the PMU (LDO and power detection), the analog components of the reset system, the comparator (COMP), and the ADC, as well as the I/O; it remains operational after power-up.

5.1.2 VDD_Core domain

VDD_Core domain mainly provides power for the core, memory and peripherals of the chip, and keeps open after power-on by default. When entering the low-power standby mode, the chip hardware will choose the low-power mode for the voltage regulator and it automatically switches the regulator to its normal power mode after wake up. It has the following several working states:

Run Mode: VDD_Core domain runs in a normal power consumption mode, and memory and peripherals normally work.

Sleep mode: VDD_Core domain runs in a normal power consumption mode. CPU enters the sleep mode, while memory and peripherals work in a normal power consumption mode.

Stop mode: VDD_Core domain works in a low power consumption mode, and only keeps the content of register and RAM.

Deep stop mode: VDD_Core domain operates in the lowest power consumption mode, retaining only the contents of registers and RAM.

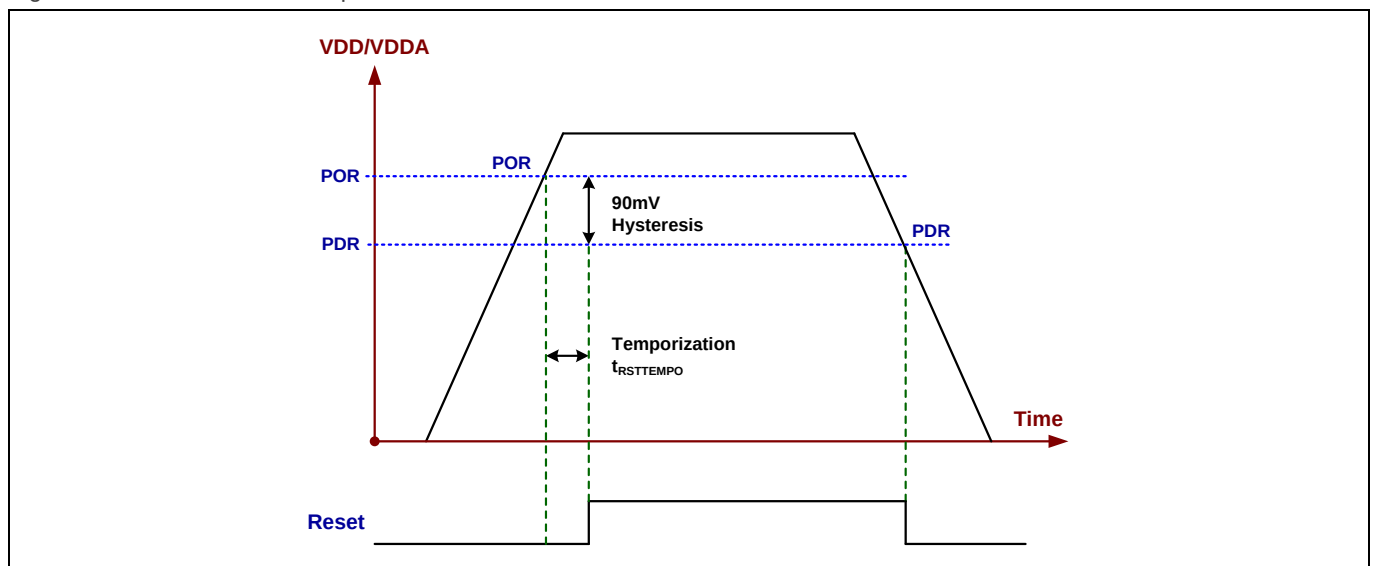
5.2 Power manager

5.2.1 Power on reset and power down reset

The chip has a complete power on reset (POR) and power down reset (PDR) circuit. When the power voltage reaches the minimum working voltage, the system works normally. When the power voltage is lower than the minimum working voltage, the system doesn't work. When the chip is powered on or powered down, the power voltage reaches the minimum working voltage of the chip during the power-on operation and the chip will produce a power-on reset. Similarly, during the chip power-down operation, the chip will produce a power-down reset when the voltage is lower than the minimum working voltage.

When VDD is lower than the specified limit voltage POR/PDR, the system remains reset, and NRST reset pin is in low level. Please refer to the electric characteristics section of the Data Manual for details on the reset duration ($t_{RSTTEMPO}$) power-on reset and power-down reset.

Figure 5-2 Power-on reset and power-down reset waveform



5.2.2 Programmable voltage detector

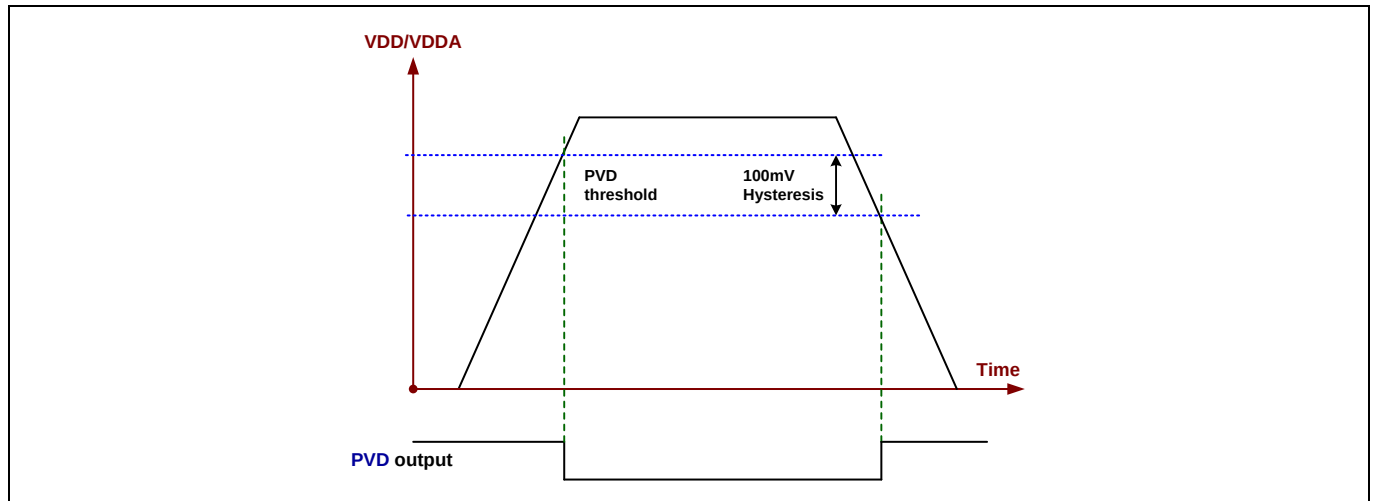
The Programmable Voltage Detector (PVD) can be used to monitor the power voltage of the chip. When the power voltage drops below a specified threshold, it also produces an interrupt, and the software can conduct an emergency treatment. When the power voltage returns to a specified threshold higher than the original one, it also produces an interrupt. The power supply is resumed after the software processing. There is a fixed difference between the power-down threshold and power-rise threshold, and this is PVD lag voltage. This difference can be detected through the listed PVD threshold data. The purpose of introducing the difference is to prevent the frequent interrupt caused by voltage fluctuation around the threshold.

The user can set the threshold voltage of the PLS bit in the power control register (PWR_CR) through the software and compare this value with the power supply voltage of the chip for power monitoring.

PVD can be enabled by setting the PVDE bit in the PWR_CR. PVDO flag in the power control/status register (PWR_CSR) is to indicate whether VDD is higher or lower than PLS bit threshold voltage in the PWR_CR.

PVD interrupt corresponds to the external interrupt EXTI16. If the user has the EXTI 16 of external interrupt, the event will produce interrupt and enters interrupt service function. When VDD drops below the PVD threshold or VDD rises above the PVD threshold, PVD interrupt will produce (PVD reset also produces via software configuration) according to rising/falling edge of the EXTI 16 of external interrupt. The user may perform some corresponding operations during interrupt. For example, when the condition triggers and power down speed is slower than the execution time of interrupt processing procedure, the emergency close may be executed if the system needs to enter the special protection status. Save some important system data, and conduct corresponding protection operation on the peripherals.

Figure 5-3 PVD threshold waveform



5.3 Power consumption control

5.3.1 Power consumption control overview

In order to extend battery product life, when MCU doesn't work, MCU may enable low power mode to save power. When MCU needs to work, MCU may be woken by various wake-up sources, thereby minimizing the chip's current consumption.

The chip has three low power modes with different power consumption, wake-up time and wake-up sources. Upon need, the user may choose the optimal low power mode.

Three low power modes of the chip:

Sleep mode: CPU stops, and all chip peripherals, including CPU peripherals, such as NVIC, system clock (SysTick) still run.

Stop mode: VDD_Core domain works at lower power mode. It only keeps content of the register and RAM and CPU and all peripheral clocks stop.

Deep stop mode: VDD_Core domain works at the lowest power mode. It only keeps content of the register and RAM and CPU and all peripheral clocks stop.

Besides, under run mode, reduce power consumption by any mode below:

- Reduce system clock frequency: While meeting system demand, select lower speed time frequency or adopt the cyclic changeover of high speed clock and low speed clock to save power.
- Close unused peripheral clock of APB and AHB bus: The user only enables the required application clock signal. Other excessive clocks are closed.
- Select lower voltage power: The higher the power voltage, the higher the power consumption of the chip. Select the proper power voltage within the chip's safety voltage range.

Table 5-1 Low power modes

Mode	Entry mode	Wake-up mode	Influence on VDD_Core domain clock	Influence on VDD domain clock	Voltage regulator	Influence on data and register	Precaution
SLEEP Mode	WFI (Wait for Interrupt)	Any interrupt	CPU clock off, no influence on other clock and ADC clock	N/A	On	N/A	External clock remains, and the content of register and SRAM remains
	WFE (Wait for Event)	Wake-up event					

Mode	Entry mode	Wake-up mode	Influence on VDD_Core domain clock	Influence on VDD domain clock	Voltage regulator	Influence on data and register	Precaution
Stop Mode	Clear LPDS bit; set SLEEPDEEP bit; WFI or WFE;	Any external interrupt (setup in external interrupt register) or event, IWDG interrupt (no reset) wake-up	All VDD_Core domain clocks close	HSI oscillators close	On	Content of register and SRAM remains	Unused GPIO before entering low power should be equipped with analog input status
DeepStop Mode	Set LPDS bit; SLEEPDEEP bit; WFI or WFE;	Any external interrupt (setup in external interrupt register) or event, IWDG interrupt (no reset) wake-up			On	Content of register and SRAM remains	Unused GPIO before entering low power should be equipped with analog input status

5.3.2 Reducing system clock under run mode

While meeting use demand, select low speed time frequency or adopt the cyclic changeover of high speed clock and low speed clock to save power.

The system clock of the chip may be flexibly configured. The user may select the system clock of different sources or reduce the frequency of any system clock (SYSCLK, HCLK, PCLK) by configuring different clock dividers.

Prior to entering the sleep mode, reducing peripheral clock may effectively save power under the sleep mode.

5.3.3 External clock control

While the chip executes the program, close the peripheral clock to reduce power consumption.

In sleep mode, close peripheral clock before executing WFI or WFE instruction and effectively reduce current consumption of peripheral under sleep mode.

The peripheral is mainly on the AHB peripheral clock enable register (RCC_AHBENR) and APB peripheral clock enable register (RCC_APBENR) bus. The user can independently configure register peripheral control bit to turn off peripheral clock.

5.3.4 Sleep mode

5.3.4.1 Entering Sleep mode

Execute WFI (Wait for Interrupt)/WFE (Wait for Event) instruction, and request MCU to enter the Sleep mode. According to the value of SLEEPONEXIT bit inside the CPU system control register (SCB->SCR), there are two options available for selecting the sleep mode entry mechanism:

SLEEPNOW: If SLEEPONEXIT bit is cleared, when WFI or WFE is executed, the MCU immediately enters the sleep mode.

SLEEPONEXIT: If SLEEPONEXIT bit is asserted and the system exits from the interrupt processing program with the lowest priority, the MCU immediately enters the sleep mode.

Under the sleep mode, all I/O pins keep the status under the run mode.

Table 5-2 SLEEPNOW mode

SLEEP NOW mode	Description
Entry	Execute WFI (Wait for Interrupt) or WFE (Wait for Event) instructions under conditions below: SLEEPDEEP = 0 SLEEPONEXIT = 0
Exit	If executing WFI, enter Sleep Mode: interrupt (refer to interrupt vector table) If executing WFE, enter Sleep Mode: wake-up event (refer to wake-up event management)
Wake-up latency	Immediate wake-up

Table 5-3 SLEEPONEXIT mode

SLEEP ON EXIT mode	Description
Entry	Execute WFI (Wait for Interrupt) instructions under conditions below: SLEEPDEEP = 0 SLEEPONEXIT = 1
Exit	Interrupt (refer to interrupt vector table)
Wake-up latency	Immediate wake-up

5.3.5 Stop Mode

Stop mode is composed of CPU deep sleep mode and peripheral's clock control. In stop mode, CPU enters the deep sleep mode, and all clocks in the VDD_Core domain are stopped. Functions of HSI oscillators is disabled. Contents of SRAM and register are saved.

In stop mode, all I/O pins remain in the same state as they were in run mode.

5.3.5.1 Enter stop mode

The stop mode has two entry ways based on different wake-up methods by programming separate control bits:

- Stop mode to wait for external interrupt line WFI mode: Configure LPDS =0 of power control register (PWR_CR); SLEEPDEEP=1 of CPU system control register (SCR). When WFI is executed, the MCU immediately enters the stop mode.
- Stop mode to wait for external event WFE: Configure LPDS =0 of power control register (PWR_CR); SLEEPDEEP=1 of CPU system control register (SCR). When WFE is executed, the MCU immediately enters the stop mode.

Functions below can be selected when entering the stop mode:

- Independent watchdog (IWDG): Initiate the independent watchdog by the key register or hardware written into the independent watchdog. The independent watchdog may select interrupt or reset to wake up chip. When the interrupt wakes up the chip, MCU continues executing the procedure before entering low power. MCU executes reset when the reset is woken. The user may select to close LSI clock source, and thus close independent watchdog.
- Internal low speed oscillator (LSI oscillator): Set by the LSION bit of RCC control/status register (RCC_CSR).

If ADC is not closed before entering the stop mode, ADC continues to consume current. Close the peripheral by setting ADEN bit of the register ADC_ADCFG as 0. The analog input mode should be set for the unused GPIO. Or else, there is still current consumption.

5.3.5.2 Exit stop mode

- When one interrupt or wake-up event leads to exiting stop mode, the system clock source hardware is auto selected as the HSI oscillator. The user needs to make reconfiguration when other clock source is selected as the clock source.
- When the voltage regulator is in run mode and the system exits from the stop mode, there will be an extra startup delay.

Table 5-4 Stop mode

Stop mode	Description
Entry	Execute WFI (Wait for Interrupt) or WFE (Wait for Event) instructions under conditions below: Set SLEEPDEEP bit in the CPU System Control Register; Reset the LPDS bit of the Power Control Register (PWR_CR); System clock switches to LSI or HSI Note: In order to enter the stop mode, all external interrupt request bit (interrupt event pending register EXTI_PR) flags must be cleared. Or else, the entry procedure of stop mode will be jumped, and the procedure continues to run.
Exit	Execute WFI (Wait for Interrupt) instruction under the following conditions: Any external interrupt line is set to interrupt mode (the corresponding external interrupt vector must be enabled in NVIC), refer to interrupt vector table Wait for Event; Execute WFE (Wait for Event) instruction under the following conditions: Any external interrupt line is set to event mode, such as a watchdog interrupt;
Wake-up latency	LSI or HSI wake-up time and extra time produced by voltage regulator wake-up
Precaution	When entering the stop mode, the unused GPIO is set as analog input mode

5.3.6 DeepStop Mode

Deep Stop is a low power consumption mode combining the clock control and voltage regulator control mechanism of peripherals on the basis of CPU deep sleep mode. In the deep stop mode, all clocks in the VDD_Core domain are stopped, HSI functions are disabled, and SRAM and register contents are retained.

In the deep stop mode, all I/O pins remain in the state of run mode.

5.3.6.1 Entering Deep Stop mode

The stop mode has two entry ways based on different wake-up methods by programming separate control bits:

- Deep stop mode to wait for external interrupt line WFI: Configure LPDS = 1 of power control register (PWR_CR); SLEEPDEEP=1 of CPU system control register (SCR). When WFI is executed, the MCU immediately enters the deep stop mode.
- Deep stop mode to wait for external event WFE: Configure LPDS =1 of power control register (PWR_CR); SLEEPDEEP=1 of CPU system control register (SCR). When WFE is executed, the MCU immediately enters the deep stop mode.

In Deep Stop mode, the following features can be selected by setting individual control bits:

- Independent watchdog (IWDG): The IWDG is started by writing to its key register or by hardware option. The independent watchdog may select interrupt or reset to wake up chip. When the interrupt wakes up the chip, MCU continues executing the procedure before entering low power. MCU executes reset when the reset is woken. The user may select to close LSI clock source, and thus close independent watchdog.
- Internal oscillator (LSI oscillator): This is configured by the LSION bit in the control/status register (RCC_CSR).

If ADC is not closed before entering the deep stop mode, ADC continues to consume current. Close the peripheral by setting ADEN bit of the register ADC_ADCFG as 0. The analog input mode should be set for the unused GPIO. Or else, there is still current consumption.

5.3.6.2 Exiting Deep Stop mode

When the DeepStop mode is interrupted or the event is woken up or exited, the system clock serves as HSI oscillator. The user needs to make reconfiguration when other clock source is selected as the clock source.

When the voltage regulator is operation in low-power mode and the system exits from the deep stop mode, there will be an extra startup delay.

Table 5-5 Deep Stop mode

Deep Stop Mode	Description
Entry	Execute WFI (Wait for Interrupt) or WFE (Wait for Event) instructions under conditions below: Set SLEEPDEEP bit in the CPU System Control Register; Reset the LPDS bit of the Power Control Register (PWR_CR); System clock switches to LSI or HSI Note: In order to enter the deep stop mode, all external interrupt request bit (interrupt event pending register EXTI_PR) flags must be cleared. Or else, the entry procedure of deep stop mode will be jumped, and the procedure continues to run.
Exit	Execute WFI (Wait for Interrupt) instruction under the following conditions: Any external interrupt line is set to interrupt mode (the corresponding external interrupt vector must be enabled in NVIC), refer to interrupt vector table Wait for Event; Execute WFE (Wait for Event) instruction under the following conditions: Any external interrupt line is set to event mode, such as a watchdog interrupt;
Wake-up latency	LSI or HSI wake-up time and extra time produced by voltage regulator wake-up
Precaution	When entering the deep stop mode, the unused GPIO is set as analog input mode

5.4 Power control register

Table 5-6 Power control register overview

Offset	Acronym	Register Name	Reset
0x00	PWR_CR	Power control register	0x00000600
0x04	PWR_CSR	Power control status register	0x00000000

5.4.1 PWR_CR Power Control Register

Address offset: 0x00

Reset value: 0x00000600

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			PLS				Res				PVDE	Res			LPDS
			rw								rw				rw

Bit	Field	Description
31:13	Reserved	Reserved, always read as 0
12 : 9	PLS	PVD level selection These bits are used to select the voltage threshold of the power voltage detector. 0000 : 1.8V 0100 : 3.0V 1000 : 4.2V 0001 : 2.1V 0101 : 3.3V 1001 : 4.5V 0010 : 2.4V 0110 : 3.6V 1010 : 4.8V 0011 : 2.7V 0111 : 3.9V Others: Reserved Note: Refer to the electrical characteristics section in the datasheet for detailed information.
8	Reserved	Reserved, always read as 0
7:5	Reserved	Reserved, always read as 0

Bit	Field	Description
4	PVDE	Power voltage detector enable 1 = PVD enabled 0 = PVD disabled
3:1	Reserved	Reserved, always read as 0
0	LPDS	Low Power DeepStop: 1: When entering Stop Mode, the voltage regulator operates in low-power mode. 0: When entering Stop Mode, the voltage regulator operates in normal power mode. When entering Stop Mode, the current with LPDS = 1 is lower than the current with LPDS = 0. Refer to the corresponding datasheet of this chip for more details.

5.4.2 PWR_CSR Power Control/Status Register

Address offset: 0x04

Reset value: 0x00000000

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													PVDO	Res	
													r		

Bit	Field	Description
31:3	Reserved	Reserved, always read as 0
2	PVDO	PVD output This bit is valid when PVD is enabled by the PVDE bit. 1: VDD/VDDA is lower than the PVD threshold selected by PLS[3:0] 0: VDD/VDDA is higher than the PVD threshold selected by PLS[3:0].
1:0	Reserved	Reserved, always read as 0

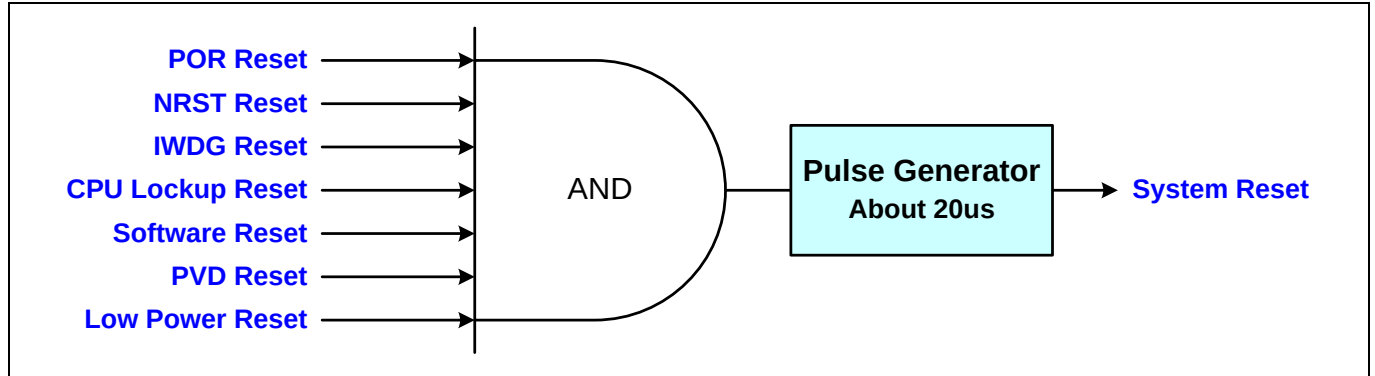
6 RCC Clock and ResetReset unit

6.1.1 Overview

There are two types of resets: power reset, system reset.

6.1.2 Functional block diagram

Figure 6-1 Reset functional block diagram



6.1.3 Main characteristics

- Reset event judgment: Judge through the reset flag bit of the control status register (RCC_CSR).
- Power on reset: Reset all registers.
- System reset: Except that the reset flag and internal low-speed oscillator enable flag in clock control register (RCC_CSR), the wake-up flag in power control register (PWR_CSR), and the DBG control register (DBG_CR) are not affected by system reset, other registers are reset by the system.

6.1.4 Functional description

6.1.4.1 POR reset

Power reset includes:

- Power-on Reset
- Power-down Reset

6.1.4.2 System reset

System reset includes:

- NRST Reset
- IWDG Reset
- Software Reset
- CPU Lockup Reset
- PVD Reset
- Low Power Reset

NRST Reset:

- When inputing the low level through NRST Pin, the NRST reset will produce.

IWDG Reset:

- The counter begins ramp down from its reset value 0x0FFF. When it ramps down to 0x0000, the independent watchdog reset will produce.
- If the program is anomaly, the feed will not be normal, and the independent watchdog reset will produce.

- Refer to the chapter of window watchdog for details

Software Reset:

- Set SCB_AIRCR [SYSRESETREQ] as 1, and software reset will produce.

CPU Lockup Reset:

- Configure the LOCKUPEN bit of control status register (RCC_CSR) as 1, and enable CPU lockup reset;
- CPU lockup reset will produce when CPU enters the lockup status.

PVD Reset:

- Configure the PVDRSTEN bit of control status register (RCC_CSR) as 1, enable PVD reset;
- Configure the PVDE bit of power control register 1 (PWR_CR) as 1, enable PVD;
- Configure the PLS bit of power control register 1 (PWR_CR) and select the PVD threshold;
- Detect VDD power. When the VDD power is lower than the selected threshold voltage, PVD reset will produce.

Low Power Reset:

- To prevent the application from inadvertently entering low-power mode, the system can be reset before such entry occurs by configuring the nRST_STOP bit in the option byte space to 0.
- By configuring the nRST_STOP bit in the option byte space to 0, the system will perform a reset instead of entering Stop mode.
- Please refer to the Embedded Flash Memory section for details.

6.2 Clock unit

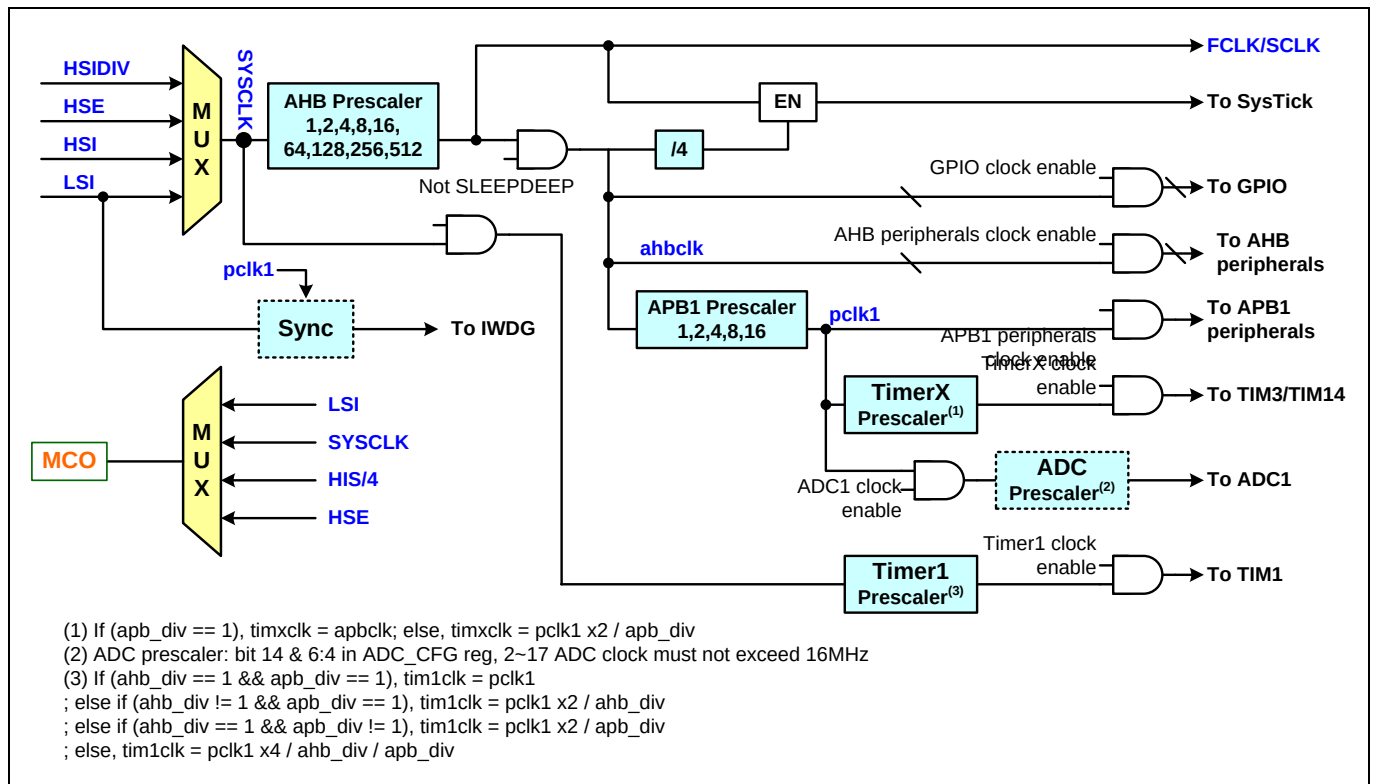
6.2.1 Overview

Four configurable independent clocks:

- High-speed external clock (HSE)
- High-speed internal clock divided by 6 (HSIDIV)
- High-speed internal clock (HSI)
- Low-speed internal clock (LSI)

6.2.2 Functional block diagram

Figure 6-2 Clock tree



6.2.3 Main characteristics

Respectively configure the clock frequency of AHB and APB1 bus via the prescaler control bit of the clock register (RCC_CFGR). The maximum frequency of AHB and APB1 bus clock is 48MHz.

6.2.4 Functional description

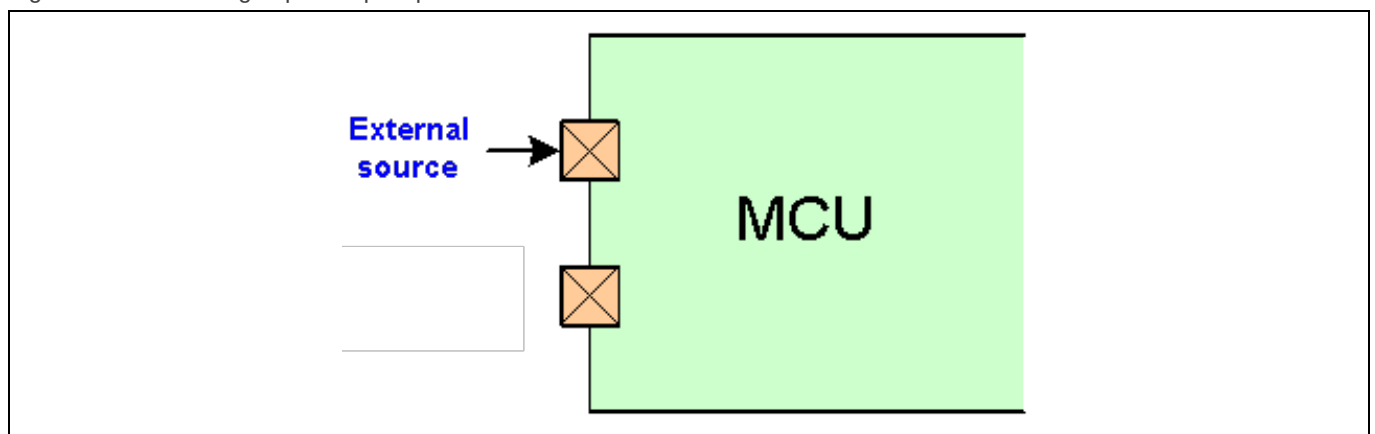
6.2.4.1 External high-speed clock (HSE)

One variety of high-speed external clock source:

- External high-speed input speed

This clock input mode is selected by configuring the HSEON bit of the clock control register (RCC_CR).

Figure 6-3 External high-speed input speed



The figure above is the module block diagram of high-speed external input clock

The high-speed external clock has characteristics below:

- External high-speed clock source must be provided
- Input frequency range 4~24MHz
- External high-speed clock signal (square wave, sine wave) with 50% duty cycle. Please refer to the chapter of electric characteristics of the data manual for details
- When HSE is enabled, OSC_IN pin must be driven

Precaution:

- Once HSE is enabled, relevant configuration of HSE cannot be changed. When the configuration needs to be changed, it's mandatory to disable HSE first.

The process to open the HSE configuration:

- Configure the HSEON bit of clock control register (RCC_CR) as 1, and enable HSE;
- Set the HSERDY bit of clock control register (RCC_CR) as 1. It means that HSE is ready, and valid clock signal will be output. At this time, it may be selected as the system clock or peripheral clock source.

6.2.4.2 High-speed internal clock (HSI)

HSI clock signal is produced by the internal 48MHz oscillator. HSI clock source is opened by default when the chip is power on.

Steps to enable HSI configuration:

- Configure the HSION bit of clock control register (RCC_CR) as 1, and enable HSI;
- Set the HSIRDY bit of clock control register (RCC_CR) as 1. It means that HSI is ready, and valid clock signal will be output. At this time, it may be selected as the system clock or peripheral clock source.

Precaution:

- Once HSI is enabled, relevant configuration of HSI cannot be changed. When the configuration needs to be changed, it's mandatory to disable HSI first.

6.2.4.3 Low-speed internal clock (LSI)

LSI acts as the low power clock source, and it provides the clock source for the independent watchdog. The clock center frequency is around 40 kHz. Refer to the chapter of electric characteristics of data manual.

Steps to enable LSI configuration:

- Set the LSION bit of control status register (RCC_CSR) as 1, enable LSI;
- Set the LSIRDY bit of control status register (RCC_CSR) as 1. It means that LSI is ready, and valid clock can be exported.

Precaution:

- Once LSI is enabled, relevant LSI configuration cannot be changed. If it's required to change configuration, disable LSI first.

6.2.4.4 Interrupt

Table 6-1 RCC global interrupt

Interrupt event	Event flag bit	Enable control bit	Flag clear bit
RCC_HSERDY	HSERDYF	HSERDYIE	HSERDYC
RCC_HSIRDY	HSIRDYF	HSIRDYIE	HSIRDYC
RCC_LSIRDY	LSIRDYF	LSIRDYIE	LSIRDYC

Note: The above flag bit, control bit and clear bit can be configured by the clock interrupt register (RCC_CIR).

6.2.4.5 System clock selection (SWS)

Four system clock sources:

- High-speed internal clock (HSIDIV) (default after power on)
- High-speed internal clock (HSI)
- High-speed external clock (HSE)
- Low-speed internal clock (LSI)

System clock configuration steps:

- Enable the required system clock source (HSIDIV, HSI, HSE, LSI). Each clock is enabled differently. For specific method, please view (chapter of HSI, HSE, LSI);
- Set the bit of the selected clock source RDY signal as 1. It means that the system clock source is ready (when the target clock source is ready, the system clock may change over);
- Select the system clock by configuring the SW bit of clock configuration register (RCC_CFGR);
- Read the SWS bit of clock configuration register (RCC_CFGR), and judge the current system clock source.

6.2.4.6 System clock frequency switch

The system clock frequency switches from low speed to high speed or from high speed to low speed. It's recommended to switch the medium speed frequency for transition. The interval switch time from high speed to low speed is 1us at least.

6.2.4.7 Peripheral reset

Achieve the corresponding peripheral software reset by APB1 peripheral reset register (RCC_APB1RSTR) and AHB peripheral reset register (RCC_AHBSTR).

6.2.4.8 Microcontroller clock output (MCO)

The microcontroller clock output (MCO) allows the clock to export to the external MCO pin. The configuration register of the corresponding GPIO port must be configured as the multiplex output function. Select any one of the 5 clock signals below as MCO output clock:

Table 6-2 Correlation of MCO and clock source

MCO bit of clock configuration register (RCC_CFGR)	Clock source
00x	No clock output
010	LSI
100	SYSCLK
101	HSI/4
110	HSE

6.2.4.9 Independent watchdog clock

When the independent watchdog is enabled by hardware, LSI oscillator will be automatically opened, and cannot be closed;

When the independent watchdog is enabled by software, LSI oscillator needs to be enabled and opened by software. When LSI oscillator is ready, the clock is supplied to IWDG, and LSI can be closed by software.

6.3 Register description

6.3.1 Register overview

Table 6-3 RCC register overview

Offset	Acronym	Register Name	Reset
0x00	RCC_CR	Clock Control Register	0x00000001
0x04	RCC_CFGR	Clock Configuration Register	0x00000000
0x08	RCC_CIR	Clock Interrupt Register	0x00000000
0x10	RCC_APB1RSTR	APB1 Peripheral Reset Register	0x00000000
0x14	RCC_AHBENR	AHB Peripheral Clock Enable Register	0x00000014
0x1C	RCC_APB1ENR	APB1 Peripheral Clock Enable Register	0x00000000
0x24	RCC_CSR	Control Status Register	0x08000000
0x28	RCC_AHBRSTR	AHB Peripheral Reset Register	0x00000000
0x40	RCC_SYSCFG	System Configuration Register	0x00000003

6.3.2 RCC_CR Clock Control Register

Address offset: 0x00

Reset value: 0x0000 0001

Access: No wait state, word, half word and byte access

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved														HSERDY	HSEON
														r	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved														HSIRDY	HSION
														r	rw

Bit	Field	Description
31 : 18	Reserved	Reserved, must be kept at reset value.
17	HSERDY	External High-speed Clock Ready Flag Set by hardware. 0: high-speed external crystal oscillator is not ready 1: high-speed external crystal oscillator is ready
16	HSEON	External High-speed Clock Enable Set '1' or clear '0' by software. When entering the standby or stop mode, the bit is cleared '0' by hardware. When HSE has been or will be selected as the clock source by system clock, this bit is forbidden to be reset. 0: Disable high-speed external crystal oscillator 1: Enable high-speed external crystal oscillator
15 : 2	Reserved	Reserved, must be kept at reset value.
1	HSIRDY	Internal High-speed Clock Ready Flag Set '1' by hardware. It means that the internal clock is ready. When HSION bit is cleared, HSIRDY turns "0" after 3 AHB clock periods. 0: Internal high-speed clock is not ready 1: Internal high-speed clock is ready
0	HSION	Internal High-speed Clock Enable Set "1" or clear "0" by software. When leaving standby or stop mode or external oscillator used as system clock, produce fault, and the bit is set as "1" by hardware to force opening of internal oscillator. When the system clock has used or will use HSI as the clock source, disable resetting the bit. 0: Disable internal high-speed clock 1: Enable internal high-speed clock

6.3.3 RCC_CFGR Clock Configuration Register

Address offset: 0x04

Reset value: 0x0000 0000

Access: No wait state, word, half word and byte access

Only when access occurs during clock change over, insert 1 or 2 wait state.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.					MCO			Res.							
					rw										
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.					PPRE1			HPRE				SWS		SW	
					rw			rw				r		rw	

Bit	Field	Description
31 : 27	Reserved	Reserved, must be kept at reset value.
26 : 24	MCO	Micro Controller Clock Output Set '1' or clear '0' by software 00x: No clock output 010: LSI clock output 100: System clock (SYSCLK) 8 division output 101: HSI 4 division output 110: HSE clock output Other: No clock output Note: This clock output may be stopped when starting or switching MCO clock source. When system clock is exported to MCO pin, please guarantee that the output clock frequency is no more than 50MHz
23 : 11	Reserved	Reserved, must be kept at reset value.
10 : 8	PPRE1	APB1 Prescaler Coefficient Set the software to control the APB1 clock (PCLK2) prescaler factor. 0xx: HCLK no division 100: HCLK 2 division 101: HCLK 4 division 110: HCLK 8 division 111: HCLK 16 division
7 : 4	HPRE	AHB Prescaler Coefficient Set the software to control the APB clock prescaler factor. 0xxx: SYSCLK no division 1000: SYSCLK 2 division 1001: SYSCLK 4 division 1010: SYSCLK 8 division 1011: SYSCLK 16 division 1100: SYSCLK 64 division 1101: SYSCLK 128 division 1110: SYSCLK 256 division 1111: SYSCLK 512 division Note: When the prescaler coefficient of AHB clock is greater than 1, it's mandatory to enable prefetch buffer. See the chapter of flash memory for details.
3 : 2	SWS	System Clock Switch Status 00: Select HSI divided by 6 output as system clock 01: Select HSE output as system clock 10: Select HSI output as system clock 11: Select LSI output as system clock
1 : 0	SW	System Clock Switch Select system clock source by software configuration When returning from stop or standby mode or directly or indirectly as system clock HSE produces fault, the hardware will enforce selection of HSI as system clock. 00: Select HSI divided by 6 output as system clock 01: Select HSE output as system clock 10: Select HSI output as system clock 11: Select LSI output as system clock

6.3.4 RCC_CIR Clock Interrupt Register

Address offset: 0x08

Reset value: 0x0000 0000

Access: No wait period, word, half word and byte access

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.												HSERDYC	HSIRDYC	Res.	LSIRDYC
												w1c	w1c		w1c
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.				HSERDYIE	HSIRDYIE	Res.	LSIRDYIE	Res.				HSERDYF	HSIRDYF	Res.	LSIRDYF
				rw	rw		rw					r	r		r

Bit	Field	Description
31 : 20	Reserved	Reserved, must be kept at reset value.
19	HSERDYC	HSE Ready Interrupt Clear Set '1' by software to clear HSE ready interrupt flag bit HSERDYF. 0: Invalid 1: Clear HSE ready interrupt flag bit HSERDYF
18	HSIRDYC	HSI Ready Interrupt Clear Set '1' by software to clear HSI ready interrupt flag bit HSIRDYF. 0: Invalid 1: Clear HSI ready interrupt flag bit HSIRDYF
17	Reserved	Reserved, must be kept at reset value.
16	LSIRDYC	LSI Ready Interrupt Clear Set '1' by software to clear LSI ready interrupt flag bit LSIRDYF 0: Invalid 1: Clear LSI ready interrupt flag bit LSIRDYF
15 : 12	Reserved	Reserved, must be kept at reset value.
11	HSERDYIE	HSE Ready Interrupt Enable Set '1' by software to enable or clear "0" to disable external oscillator ready interrupt. 0: Disable HSE ready interrupt 1: Enable HSE ready interrupt
10	HSIRDYIE	HSI Ready Interrupt Enable Set '1' by software to enable or clear "0" to disable external oscillator ready interrupt. 0: Disable HSI ready interrupt 1: Enable HSI ready interrupt
9	Reserved	Reserved, must be kept at reset value.
8	LSIRDYIE	LSI Ready Interrupt Enable Set '1' by software to enable or clear "0" to disable internal 40KHz oscillator ready interrupt. 0: Disable LSI ready interrupt 1: Enable LSI ready interrupt
7 : 4	Reserved	Reserved, must be kept at reset value.
3	HSERDYF	HSE Ready Interrupt Flag When the high-speed external clock is ready, set '1' by hardware. Clear by setting HSERDYC bit as "1" by software. 0: Produce clock ready interrupt without external oscillator 1: Clock ready interrupt caused by external oscillator
2	HSIRDYF	HSI Ready Interrupt Flag When the high-speed internal clock is ready, set '1' by hardware. Clear by setting HSIRDYC bit as "1" by software. 0: Produce clock ready interrupt without internal HSI oscillator 1: Clock ready interrupt caused by internal HSI oscillator
1	Reserved	Reserved, must be kept at reset value.
0	LSIRDYF	LSI Ready Interrupt Flag When the low-speed internal clock is ready, set '1' by hardware. Clear by setting LSIRDYC bit as "1" by software. 0: Produce clock ready interrupt without internal 40KHz oscillator 1: Clock ready interrupt caused by internal 40KHz oscillator

6.3.5 RCC_APB1RSTR APB1 Peripheral Reset Register

Address offset: 0x10

Reset value: 0x0000 0000

Access: No wait period, word, half word and byte access

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	SYSCFG	DBG	PWR	Res.				CPT	Res.	I2C1	Res.			USART2	USART1
	rw	rw	rw					rw		rw				rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.			SPI1	Res.		ADC1	Res.					TIM14	TIM1	TIM3	Res.
			rw			rw						rw	rw	rw	

Bit	Field	Description
31	Reserved	Reserved, must be kept at reset value.
30	SYSCFG	SYSCFG Reset Set to "1" or clear "0" by software 0: Invalid 1: Reset
29	DBG	DBG Reset Set to "1" or clear "0" by software 0: Invalid 1: Reset
28	PWR	Power Interface Reset Set to "1" or clear "0" by software 0: Invalid 1: Reset
27 : 24	Reserved	Reserved, must be kept at reset value.
23	CPT	Comparator Reset Set to "1" or clear "0" by software 0: Invalid 1: Reset
22	Reserved	Reserved, must be kept at reset value.
21	I2C1	I2C1 Reset Set to "1" or clear "0" by software 0: Invalid 1: Reset
20 : 18	Reserved	Reserved, must be kept at reset value.
17	USART2	USART2 Reset Set to "1" or clear "0" by software 0: Invalid 1: Reset
16	USART1	USART1 Reset Set to "1" or clear "0" by software 0: Invalid 1: Reset
15 : 13	Reserved	Reserved, must be kept at reset value.
12	SPI1	SPI1 Reset Set to "1" or clear "0" by software 0: Invalid 1: Reset
11 : 10	Reserved	Reserved, must be kept at reset value.
9	ADC1	ADC1 Reset Set to "1" or clear "0" by software 0: Invalid 1: Reset
8 : 4	Reserved	Reserved, must be kept at reset value.
3	TIM14	TIM14 Reset Set to "1" or clear "0" by software 0: Invalid 1: Reset

Bit	Field	Description
2	TIM1	TIM1 Reset Set to "1" or clear "0" by software 0: Invalid 1: Reset
1	TIM3	TIM3 Reset Set to "1" or clear "0" by software 0: Invalid 1: Reset
0	Reserved	Reserved, must be kept at reset value.

6.3.6 RCC_AHBENR AHB Peripheral Clock Enable Register

Address offset: 0x14

Reset value: 0x0000 0014

Access: No wait period, word, half word and byte access

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.													GPIOB	GPIOA	Res.
													rw	rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.									CRC	Res.	FLASH	Res.	SRAM	Res.	
									rw		rw		rw		

Bit	Field	Description
31 : 19	Reserved	Reserved, must be kept at reset value.
18	GPIOB	GPIOB Clock Enable Set "1" or clear "0" by software 0: Clock disable 1: Clock enable
17	GPIOA	GPIOA Clock Enable Set "1" or clear "0" by software 0: Clock disable 1: Clock enable
16 : 7	Reserved	Reserved, must be kept at reset value.
6	CRC	CRC Clock Enable Set "1" or clear "0" by software 0: Clock disable 1: Clock enable
5	Reserved	Reserved, must be kept at reset value.
4	Flash	FLASH Clock Enable Set "1" or clear "0" by software 0: Clock disable 1: Clock enable
3	Reserved	Reserved, must be kept at reset value.
2	SRAM	SRAM Clock Enable Set "1" or clear "0" by software 0: Clock disable 1: Clock enable
1 : 0	Reserved	Reserved, must be kept at reset value.

6.3.7 RCC_APB1ENR APB1 Peripheral Clock Enable Register

Address offset: 0x1C

Reset value: 0x0000 0000

Access: No wait period, word, half word and byte access

Note: When peripheral clock is not enabled, the software cannot read the value of peripheral register

MG32F04A064 User Guide

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	SYSCFG	DBG	PWR	Res.				CPT	Res.	I2C1	Res.			USART2	USART1
	rw	rw	rw					rw		rw				rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.			SPI1	Res.		ADC1	Res.				TIM14	TIM1	TIM3	Res.	
			rw			rw					rw	rw	rw		

Bit	Field	Description
31	Reserved	Reserved, must be kept at reset value.
30	SYSCFG	SYSCFG Clock Enable Set to "1" or clear to "0" by software 0: Clock disable 1: Clock enable
29	DBG	DBG Clock Enable Set to "1" or clear to "0" by software 0: Clock disable 1: Clock enable
28	PWR	Power Clock Enable Set to "1" or clear to "0" by software 0: Clock disable 1: Clock enable
27 : 24	Reserved	Reserved, must be kept at reset value.
23	CPT	Comparator Clock Enable Set to "1" or clear to "0" by software 0: Clock disable 1: Clock enable
22	Reserved	Reserved, must be kept at reset value.
21	I2C1	I2C1 Clock Enable Set to "1" or clear to "0" by software 0: Clock disable 1: Clock enable
20 : 18	Reserved	Reserved, must be kept at reset value.
17	USART2	USART2 Clock Enable Set to "1" or clear to "0" by software 0: Clock disable 1: Clock enable
16	USART1	USART1 Clock Enable Set to "1" or clear to "0" by software 0: Clock disable 1: Clock enable
15 : 13	Reserved	Reserved, must be kept at reset value.
12	SPI1	SPI1 Clock Enable Set to "1" or clear to "0" by software 0: Clock disable 1: Clock enable
11 : 10	Reserved	Reserved, must be kept at reset value.
9	ADC1	ADC1 Clock Enable Set to "1" or clear to "0" by software 0: Clock disable 1: Clock enable
8 : 4	Reserved	Reserved, must be kept at reset value.
3	TIM14	TIM14 Clock Enable Set to "1" or clear to "0" by software 0: Clock disable 1: Clock enable
2	TIM1	TIM1 Clock Enable Set to "1" or clear to "0" by software 0: Clock disable 1: Clock enable

Bit	Field	Description
1	TIM3	TIM3 Clock Enable Set to "1" or clear to "0" by software 0: Clock disable 1: Clock enable
0	Reserved	Reserved, must be kept at reset value.

6.3.8 RCC_CSR Control Status Register

Address offset: 0x24

Reset value: 0x0800 0000

Access: 0-3 wait state, word, half word and byte access

In case of continuous access to the register, insert the wait state.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LPWR RSTF	Res.	IWDGRS TF	SFTRST F	PORRS TF	PINRST F	Res.	RMVF	LOCKUP F	PVDRST F	Res.					
r		r	r	r	r		w1c	r	r						
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.								LOCKUP EN	PVDRST EN	Res.				LSIRDY	LSION
								rw	rw					r	rw

Bit	Field	Description
31	LPWRRSTF	Low Power Reset Flag In case of Low Power reset, set'1' by hardware, and enable clear by power reset or write RMVF bit to clear by software. 0: No Low Power reset 1: Low Power reset
30	Reserved	Reserved, must be kept at reset value.
29	IWDGRSTF	Independent Watchdog Reset Flag In case of independent watchdog reset, set'1' by hardware, and enable clear by power reset or write RMVF bit to clear by software. 0: No independent watchdog reset 1: Independent watchdog reset
28	SFTRSTF	Software Reset Flag In case of software reset, set'1' by hardware, and enable clear by power reset or write RMVF bit to clear by software. 0: No software reset 1: Software reset
27	PORRSTF	POR/PDR Reset Flag In case of POR/PDR reset, set'1' by hardware, and enable clear by power reset or write RMVF bit to clear by software. 0: No POR/PDR reset 1: POR/PDR reset
26	PINRSTF	NRST PIN Reset Flag In case of NRST PIN reset, set'1' by hardware, and enable clear by power reset or write RMVF bit to clear by software 0: No NRST PIN reset 1: NRST PIN reset
25	Reserved	Reserved, must be kept at reset value.
24	RMVF	Remove Reset Flag Clear the reset flag by setting '1' by software. 0: Invalid 1: Clear reset flag
23	LOCKUPF	CPU Lockup Reset Flag In case of CPU lockup reset, set'1' by hardware, and enable clear by power reset or write RMVF bit to clear by software 0: No CPU lockup reset 1: CPU lockup reset

Bit	Field	Description
22	PVDRSTF	PVD Reset Flag In case of PVD reset, set '1' by hardware, and enable clear by power reset or write RMVF bit to clear by software 0: No PVD reset 1: PVD reset
21 : 8	Reserved	Reserved, must be kept at reset value.
7	LOCKUPEN	CPU Lockup Reset Enable 0: Disable CPU lockup reset 1: Enable CPU lockup reset
6	PVDRSTEN	PVD Reset Enable 0: Disable PVD reset 1: Enable PVD reset
5 : 2	Reserved	Reserved, must be kept at reset value.
1	LSIRDY	Internal Low-speed Oscillator Ready Setting '1' or clearing '0' by hardware indicates whether the internal 40KHz oscillator is ready. When LSION is cleared, LSIRDY is cleared after 3 AHB clocks. 0: Internal 40KHz oscillator clock not ready 1: Internal 40KHz oscillator clock ready
0	LSION	Internal Low-speed Oscillator Enable Setting '1' or clearing '0' by software, or clear by power reset 0: Disable internal 40KHz oscillator 1: Enable internal 40KHz oscillator

6.3.9 RCC_AHBRSTR AHB Peripheral Reset Register

Address offset: 0x28

Reset value: 0x0000 0000

Access: No wait period, word, half word and byte access

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.													GPIOB	GPIOA	Res.
													rw	rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.															

Bit	Field	Description
31 : 19	Reserved	Reserved, must be kept at reset value.
18	GPIOB	GPIOB Reset Set "1" or clear "0" by software 0: Invalid 1: Reset
17	GPIOA	GPIOA Reset Set "1" or clear "0" by software 0: Invalid 1: Reset
16 : 0	Reserved	Reserved, must be kept at reset value.

6.3.10 RCC_SYSCFG System Configuration Register

Address offset: 0x40

Reset value: 0x0000 0003

Access: 0-3 wait state, word, half-word and byte access

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
Res.												SFT_NRST_RMP	SECTOR1KCFG	PROG_CHECK_EN	
												rw	rw	r	

Bit	Field	Description
31 : 3	Reserved	Reserved, must be kept at reset value.
2	SFT_NRST_RMP	SFT_NRST_RMP: software mapping of nRST. This bit is only cleared when the power is reset or when it is set to '0' by software. 1: PA14 mapped to nRST 0: no action Note: The SFT_NRST_RMP bit of RCC_SYSCFG is set to '1', PA14 is mapped to the external reset of nRST, and the low level is at least 4us.
1	SECTOR_1K_CFG	SECTOR_1K_CFG: size for Flash erase. 1: 1K bytes 0: 512 bytes
0	PROG_CHECK_EN	Check whether the data in Flash is 0xFF when writing Flash 1: Check (hardware is fixed to 1) 0: Not check

7 GPIO General-Purpose I/Os Overview

Each general-purpose IO (GPIO) port, can be individually configured by two 32-bit control registers (GPIOx_CRL/GPIOx_CRH) and two 32-bit multiplexed control registers (GPIOx_AFRL, GPIOx_AFRH) in eight modes: analog input \ input floating \ input pull-up \ input pull-down \ output push-pull \ output open-drain \ alternate function push-pull and alternate function open-drain.

Each I/O port bit is freely programmable. All registers can be accessed as 32-bit (words), 16-bit (half-words) or 8-bit (bytes). The GPIOx_BSRR and GPIOx_BRR control registers in the GPIO register group are able to modify the GPIOx_ODR bit to output 0 or 1 independently through write accesses.

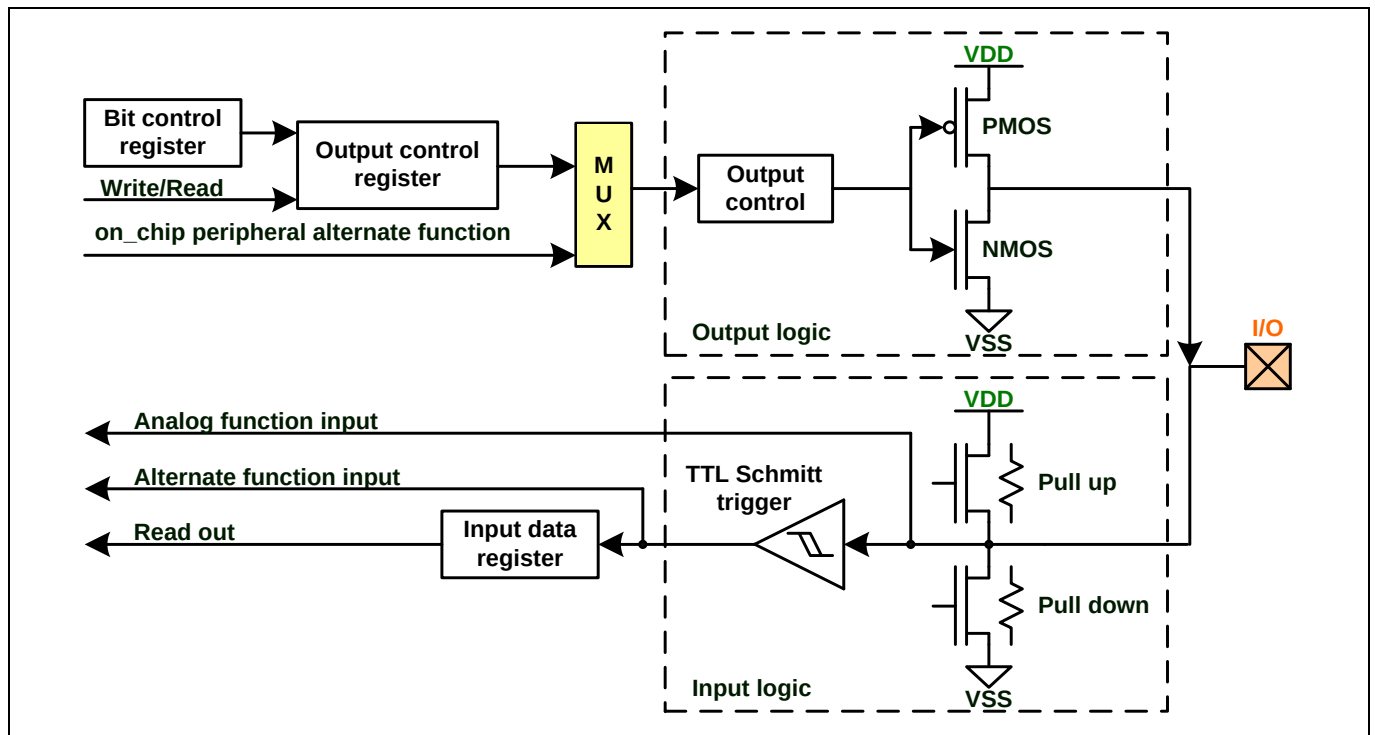
7.2 Main characteristics

- Each IOP write operation can modify one or more bits of the corresponding GPIOx_ODR.
- Supports single-cycle fast access
- All I/Os support programmable EXTI configuration register to output externally triggered interrupts
- Support GPIO locking mechanism
- Supported input states: floating, pull-up/down, analog
- Supported output states: push-pull or open drain with pull-up/down
- Input floating as default, configurable input/output direction
- The chip provides dedicated I/O pins capable of driving up to five 8-segment common-cathode LED Displays. For details, please refer to 18.2 GPIOA Seven-segment display driver.

7.3 Functional description

7.3.1 Functional block diagram

Figure 7-1 Standard I/O port



7.3.2 GPIO port configuration

Table 7-1 Port bit configuration table (take port0 as an example)

Pin_mode		Pull up/down	DCR[1:0]		CNF0		MODE0	ODRx
Analog input		x	x	x	0	0	00	x
General input Alternate input		Floating	x	x	0	1		x
		Pull-up	x	x	1	0		1
		Pull-down	x	x	1	0		0
General output	Push-pull	x	x	x	0	0	01 10 11	0 or 1
	Open-drain	Floating	x	0	0	1		0 or 1
		Pull-up	1	1	0	1		0 or 1
		Pull-down	0	1	0	1		0 or 1
Alternate output	Push-pull	x	x	x	1	0		x
	Open-drain	Floating	x	0	1	1		x
		Pull-up	1	1	1	1		x
		Pull-down	0	1	1	1		x

Note: x represents “don’t care” for I/Os in the corresponding mode. ODR0 represents the bit 0 of the output data register.

Input and output follow these configurations:

- General-purpose input:

The user should configure the CNF0 in the GPIOx_CRL to choose the Input mode.

- General-purpose output:

Push-Pull output: The user configures MODE0 to choose the output speed and sets CNF0=00;

Open-Drain output: The user configures MODE0 to choose the output speed and sets CNF0=01. To activate the pull-up/pull-down feature for pins, the GPIOx_DCR register should be configured individually. This feature is invalid if it is not an output open-drain mode.

- Alternate function:

Configure the AFRLx [3:0] and AFRHx [3:0] registers to choose the alternate function:

Alternate push-pull output: The user configures MODE0 to choose the output speed and sets CNF0=10;

Alternate open-drain output: The user configures MODE0 to choose the output speed and sets CNF0=11.

To activate the pull-up/pull-down feature for IO in the output mode, the GPIOx_DCR register should be configured individually. This feature is invalid if it is not an output open-drain mode.

During and just after reset, the GPIO ports are configured in Input Floating mode. The Serial-Wired Debug pins default to input pull-up (PU)/pull-down (PD).

When configured in general purpose output mode, the value of the output data register (GPIOx_ODR) is output on the corresponding I/O pin. The Input Data register (GPIOX_IDR) captures the data present on the I/O pin at every AHB clock cycle.

Note: Not all chips contain the JTAG and SWD ports. For specific chip configuration, refer to the chip datasheet.

- PA14: SWCLK in PD
- PA13: SWDIO in PU

7.3.3 Alternate function

The corresponding IO alternate functions are enabled by setting the alternate function register.

- When an IO is configured as Alternate Function Input, the port should choose pull-up, pull-down or floating input.
- When an IO is configured as Alternate Function Output, the port should choose output push-pull or open-drain mode.

- When an IO is configured with bidirectional alternate function, the port should choose output push-pull or open-drain mode. In this case the input is configured in input floating mode. In the open-drain mode, the GPIOx_DCR register is configured to choose weak pull-up or weak pull-down resistor.

If a port bit is configured as Alternate Function Output, this connects the port to the output signal of an on-chip peripheral. If a GPIO pin is configured as Alternate Function Output solely via software, but peripheral is not activated, its output is not specified.

7.3.4 GPIO locking mechanism

It is possible to freeze the IO configuration by applying the GPIO locking mechanism. When the LOCK mechanism has been applied on a port, it is no longer possible to modify the port configuration until the next reset.

The write sequence of the LOCK key:

- GPIOx_LCKR [16] = '1' + LCKR [15:0].
- GPIOx_LCKR [16] = '0' + LCKR [15:0].
- GPIOx_LCKR [16] = '1' + LCKR [15:0].

To lock the PA [0] port of GPIOA, follow these configurations:

- GPIOA->GPIOA_LCKR=0x10001.
- GPIOA->GPIOA_LCKR=0x00001.
- GPIOA->GPIOA_LCKR=0x10001.

After three steps above are finished, the bit 16 of the GPIOA_LCKR register is set. The GPIOA_LCKR register can no longer be written until the next soft reset. The bit 16 of the GPIOA_LCKR register remains 1 and PA [0] keeps its configuration before the locking.

When the port is locked, the value of the port bit can no longer be modified until the soft reset. Each lock bit in the GPIOx_LCKR register freezes 4 bits in the port configuration registers (GPIOx_CRL and GPIOx_CRH).

Notes:

Only the value of PA [0] is locked due to the proceeding configuration. It is still possible to configure the values of PA [15:1] and other GPIO control registers.

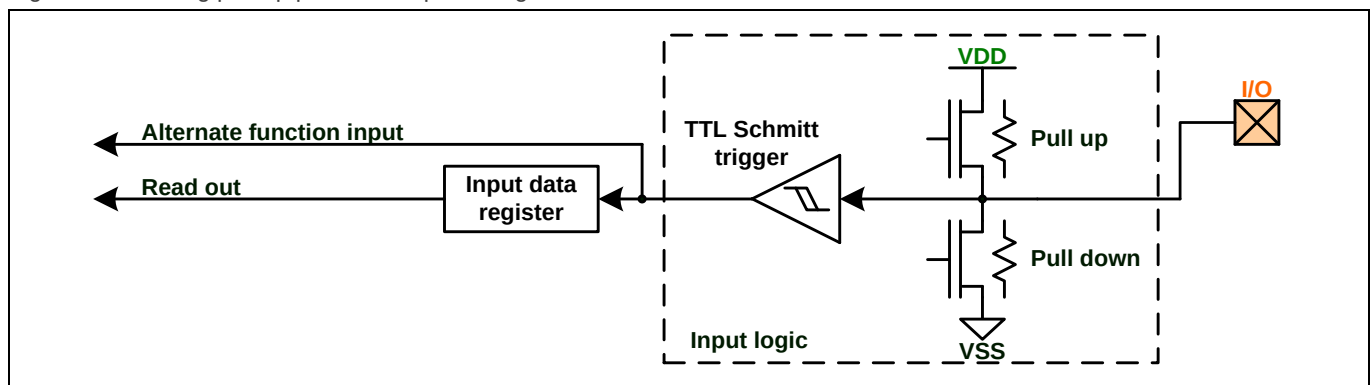
7.3.5 Input configuration

When the I/O port is program as Input:

- The Schmitt Trigger Input is activated.
- The Output Buffer is disabled.
- Floating, pull-up or pull-down input mode can be used.
- The data present on the I/O pin is sampled into the Input Data register every AHB clock cycle.
- A read access to the Input Data register gets the I/O state.

The figure below shows the input configuration of the I/O port:

Figure 7-2 Floating/pull-up/pull-down input configuration



To configure the PA [0] as pull-up input in the GPIOA, refer to the following:

- GPIOA->GPIOA_ODR=0x0001.
- GPIOA->GPIOA_CRL=0x00000008.

To configure the PA [0] as pull-down input in the GPIOA, refer to the following:

- GPIOA->GPIOA_ODR=0x0000.
- GPIOA->GPIOA_CRL=0x00000008.

Notes:

To configure the port as pull-up input, the corresponding bit in the GPIO_ODR register should output 1 firstly.

To configure the port as pull-down input, the corresponding bit in the GPIO_ODR register should output 0 firstly.

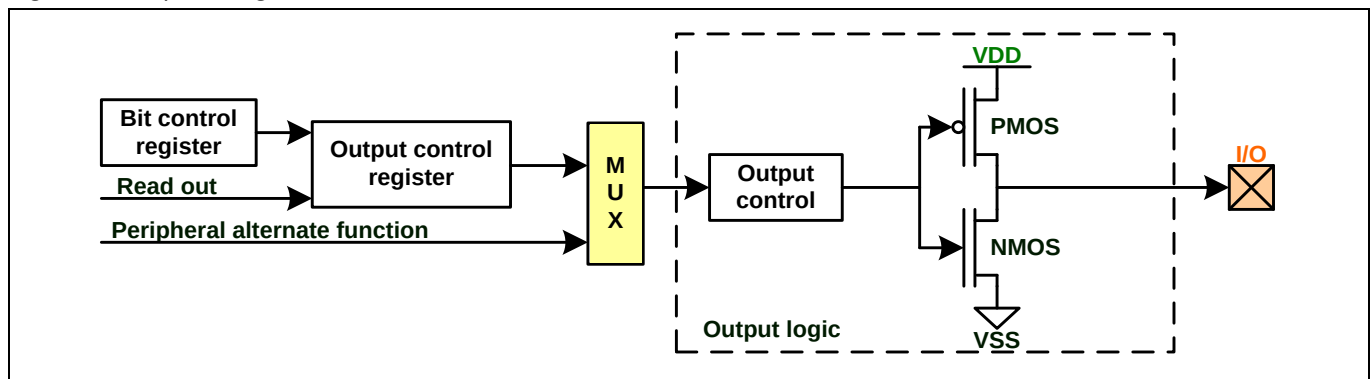
7.3.6 Output configuration

When the GPIO pin is configured as output:

- The Schmitt Trigger Input is activated.
- The Output Buffer is enabled.
- In general purpose output mode, the weak pull-up and pull-down resistors are disabled.
- Open Drain Mode: When the Port Output Data Register is set to 0, the corresponding pin outputs low level; when the Port Output Data Register is set to 1, the corresponding pin is in Hi-Z.
- Push-Pull Mode: When the output register is set to 0, the corresponding pin outputs low level; when the output register is set to 1, the corresponding pin outputs high level.
- A read access to the Port Output Data register gets the last written value.
- A read access to the Port Input Data register gets the current I/O state.

The figure below shows the output configuration of the I/O port:

Figure 7-3 Output configuration



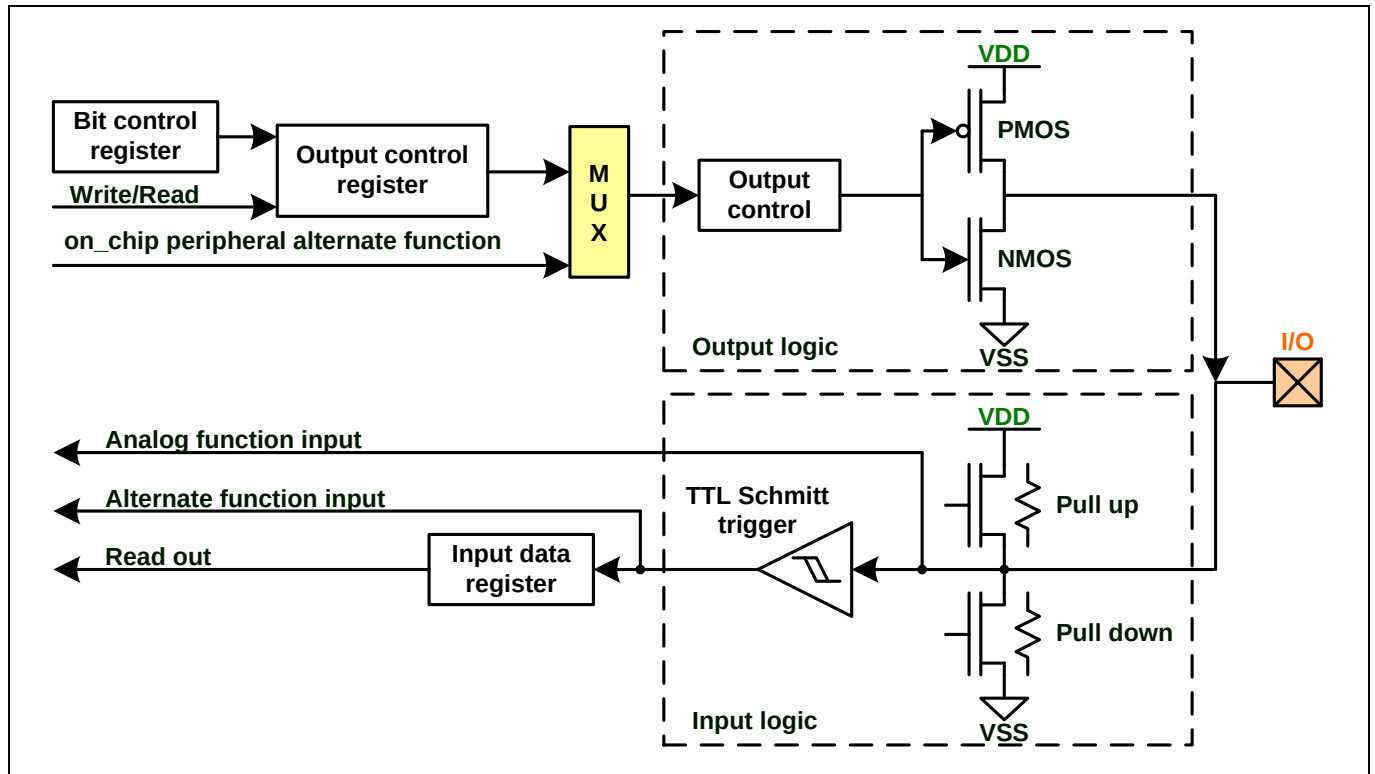
7.3.7 Alternate function configuration

When the pin is configured as alternate function:

- The Schmitt Trigger Input is activated.
- The Output Buffer can be configured as Open Drain or Push-Pull.
- In the output open-drain mode, the GPIOx_DCR register is configured to choose weak pull-up or weak pull-down resistor.
- The weak pull-up or weak pull-down resistor can be chosen when the pin is configured as Input.
- The data present on the I/O pin is sampled into the Input Data register every AHB clock cycle.

The figure below shows the alternate function configuration of the I/O port. Refer to AFRL and AFRH registers as well as the datasheet for further information.

Figure 7-4 Alternate function configuration



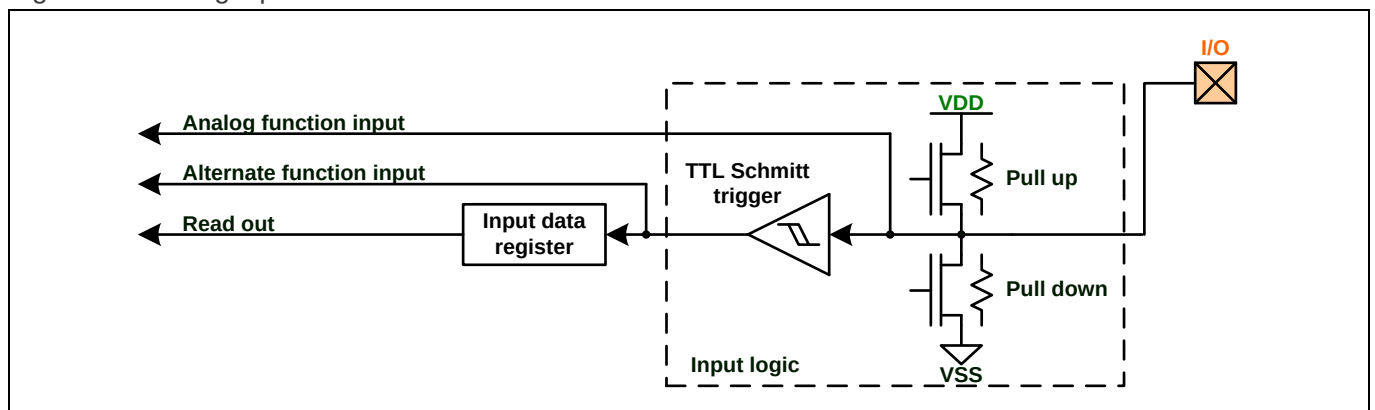
7.3.8 Analog input configuration

When the I/O port is configured as analog input configuration:

- The Output Buffer is disabled.
- The Schmitt Trigger Input is de-activated.
- The weak pull-up and pull-down resistors are disabled.
- The Port Input Data Register remains 0.

The figure below shows the analog input configuration of the I/O port:

Figure 7-5 Analog input



7.3.9 External clock used as alternate GPIO port

The external HSE clock can be used as alternate GPIO. When the corresponding clock PAD is used as the GPIO, the user should turn off the external clock first and then set the GPIO function as normal. For specific mapping, please refer to the chip datasheet.

7.3.10 SWD alternate function remapping

The SWD interface signals are mapped on the GPIO ports as shown in the table below:

Table 7-2 SWD alternate function remapping

Alternate function	GPIO port
SWDIO	PA13
SWCLK	PA14

7.3.11 NRST alternate function remapping

By default, GPIO PA[14] functions as a GPIO pin. To configure it for the NRST function, follow these steps:

- Configure GPIOA_AFRH[27:24] to a non-zero value to disable the default pull-down function of PA[14] when used as SWCLK.
- Set the SFT_NRST_RMP bit of the RCC_SYSCFGR register to 1.

Note: When PA[14] functions as NRST, the chip does not provide an internal pull-up resistor; an external pull-up resistor or a high-drive signal is required.

7.4 Register

7.4.1 Overview of registers

Table 7-3 Overview of GPIO registers

Offset	Acronym	Register Name	Reset
0x00	GPIOx_CRL	Port configuration register low	0x44444444
0x04	GPIOx_CRH	Port configuration register high	0x44444444
0x08	GPIOx_IDR	Port input data register	0x0000XXXX
0x0C	GPIOx_ODR	Port output data register	0x00000000
0x10	GPIOx_BSRR	Port bit set/reset register	0x00000000
0x14	GPIOx_BRR	Port bit reset register	0x00000000
0x18	GPIOx_LCKR	Port configuration lock register	0x00000000
0x1C	GPIOx_DCR	Port output open drain control register	0x00000000
0x20	GPIOx_AFRL	Port alternate function register low	0xFFFFFFFF
0x24	GPIOx_AFRH	Port alternate function register high	0xFFFFFFFF

7.4.2 GPIOx_CRL Port Configuration Register Low

Address offset: 0x00

Reset value: GPIOA_CRL : 0x4444 4444 · GPIOB_CRL : 0x0044 4444

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CNF7		MODE7		CNF6		MODE6		CNF5		MODE5		CNF4		MODE4	
rw		rw		rw		rw		rw		rw		rw		rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNF3		MODE3		CNF2		MODE2		CNF1		MODE1		CNF0		MODE0	
rw		rw		rw		rw		rw		rw		rw		rw	

Bit	Field	Description
31:30	CNF7	Port configuration bits (y=7..0) Set MODEy to be 0 and the port as input mode while configuring the CNFy bit to choose the input mode : 00: Analog input mode 01: Input floating mode
27:26	CNF6	
23:22	CNF5	
19:18	CNF4	

Bit	Field	Description
15:14	CNF3	10: Input pull-up/pull-down mode 11: Reserved Set MODEy not to be 0 and the port as output mode while configuring the CNFy bit to choose the output mode : 00: General purpose output push-pull 01: General purpose output open-drain 10: Alternate function output push-pull 11: Alternate function output open-drain Port input/output configuration (MODEy)(y = 0..7) The corresponding I/O port is configured via software; refer to Port bit configuration table. Set MODEy is not equal to 0, different configurations result in different rates : 00: input mode; As for the other configuration output speeds, please refer to the datasheet.
11:10	CNF2	
7:6	CNF1	
3:2	CNF0	
29:28	MODE7	
25:24	MODE6	
21:20	MODE5	
17:16	MODE4	
13:12	MODE3	
9:8	MODE2	
5:4	MODE1	
1:0	MODE0	

7.4.3 GPIOx_CRH Port Configuration Register High

Address offset: 0x04

Reset value: GPIOA_CRH : 0x4444 4444

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CNF15		MODE15		CNF14		MODE14		CNF13		MODE13		CNF12		MODE12	
rw		rw		rw		rw		rw		rw		rw		rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNF11		MODE11		CNF10		MODE10		CNF9		MODE9		CNF8		MODE8	
rw		rw		rw		rw		rw		rw		rw		rw	

Bit	Field	Description
31:30	CNF15	Port configuration bits (y=15..8) Set MODEy to be 0 and the port as input mode while configuring the CNFy bit to choose the input mode : 00: Analog input mode 01: Input floating mode 10: Input pull-up/pull-down mode 11: Reserved Set MODEy not to be 0 and the port as output mode while configuring the CNFy bit to choose the output mode : 00: General purpose output push-pull 01: General purpose output open-drain 10: Alternate function output push-pull 11: Alternate function output open-drain Port input/output configuration (MODEy)(y = 15..8) The corresponding I/O port is configured via software; refer to Port bit configuration table. Set MODEy is not equal to 0, different configurations result in different rates : 00: input mode; As for the other configuration output speeds, please refer to the datasheet.
27:26	CNF14	
23:22	CNF13	
19:18	CNF12	
15:14	CNF11	
11:10	CNF10	
7:6	CNF9	
3:2	CNF8	
29:28	MODE15	
25:24	MODE14	
21:20	MODE13	
17:16	MODE12	
13:12	MODE11	
9:8	MODE10	
5:4	MODE9	
1:0	MODE8	

7.4.4 GPIOx_IDR Port Input Data Register

Address offset: 0x08

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
IDRy(y=15~0)															

r

Bit	Field	Description
31:16	Reserved	Always read as 0
15:0	IDRy	Port input data (y=15..0) The reads represent the corresponding I/O states.

7.4.5 GPIOx_ODR Port Output Data Register

Address offset: 0xC

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
ODRy(y=15-0)															
rw															

Bit	Field	Description
31:16	Reserved	Always read as 0
15:0	ODRy	Port output data (y=15..0) When configured in general purpose output mode, the written value is output to the corresponding I/O. Note: The ODR bits can be individually set and cleared by setting the GPIOx_BSRR (x=A..H) Register.

7.4.6 GPIOx_BSRR Port Bit Set/Reset Register

Address offset: 0x10

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BRy(y=15-0)															
w															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BSy(y=15-0)															
w															

Bit	Field	Description
31:16	BRy	Port Reset bit y (y=15..0) Writing it to 0 will make the corresponding ODRY bit unchanged Writing it to 1 will clear the corresponding ODRY bit
15:0	BSy	Port Set bit y (y=15..0) Writing it to 0 will make the corresponding ODRY bit unchanged Writing it to 1 will set the corresponding ODRY bit to 1 Note: When both BSy and BRy bits are written to 1, BSy has priority over BRy.

7.4.7 GPIOx_BRR Port Bit Reset Register

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
BRy(y=15-0)															
w															

Bit	Field	Description
31:16	Reserved	Always read as 0
15:0	BRy	Port Reset bit y (y=15..0) Writing it to 0 will make the corresponding ODRY bit unchanged

Writing it to 1 will clear the corresponding ODRY bit

7.4.8 GPIOx_LCKR Port Configuration Lock Register

Address offset: 0x18

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															LCKK
															rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LCKy(y=15-0)															
w															

Bit	Field	Description
31:17	Reserved	Always read as 0
16	LCKK	Lock key This bit can be read anytime. It can only be modified using the Lock Key Writing Sequence. 0: port configuration lock key not active 1: port configuration lock key active. GPIOx_LCKR register is locked until the next soft reset. LOCK key sequence: write 1 -> write 0 -> write 1
15:0	LCKy	Port x Lock bit y (y = 15..0) These bits are read write but can only be written when the LCKK bit is 0. 0: port configuration not locked 1: port configuration locked

7.4.9 GPIOx_DCR Port Output Open Drain Control Register

Address offset: 0x1C

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PX15		PX14		PX13		PX12		PX11		PX10		PX9		PX8	
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PX7		PX6		PX5		PX4		PX3		PX2		PX1		PX0	
rw															

Bit	Field	Description
31:2	PX15-PX1	See PX0
1:0	PX0	PX0[1:0]: 11: output open-drain mode with port pull-up 01: output open-drain mode with port pull-down x0: output open-drain mode without port pull-up/pull-down

7.4.10 GPIOx_AFR1 Port Alternate Function Register Low

Address offset: 0x20

Reset value: GPIOA_AFR1 : 0xFFFF FFFF · GPIOB_AFR1 : 0x0000 00FF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AFR7				AFR6				AFR5				AFR4			
rw				rw				rw				rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AFR3				AFR2				AFR1				AFR0			
rw				rw				rw				rw			

Bit	Field	Description
31:0	AFRy	Port x alternate function bit y (y = 0..7), which can be accessed by software to configure the function. 0000: AF0 0001: AF1 0010: AF2 0011: AF3 0100: AF4 0101: AF5 0110: AF6 0111: AF7 1000: AF8 1001: AF9 1010: AF10 1011: AF11 1100: AF12 1101: AF13 1110: AF14 1111: AF15

7.4.11 GPIOx_AFRH Port Alternate Function Register High

Address offset: 0x24

Reset value: GPIOA_AFRH: 0xF00F FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AFR15				AFR14				AFR13				AFR12			
rw				rw				rw				rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AFR11				AFR10				AFR9				AFR8			
rw				rw				rw				rw			

Bit	Field	Description
31:0	AFRy	Port x alternate function bit y (y = 8..15), which can be accessed by software to configure the function. 0000: AF0 0001: AF1 0010: AF2 0011: AF3 0100: AF4 0101: AF5 0110: AF6 0111: AF7 1000: AF8 1001: AF9 1010: AF10 1011: AF11 1100: AF12 1101: AF13 1110: AF14 1111: AF15

8 EXTI Interrupt and Event Overview

Nestled vector interrupt controller (NVIC) connects the processor core, and manages the anomaly and interrupt processing with low latency. NVIC includes 2-bit interrupt priority levels, which can thus provide 4 interrupt priority levels. For other more anomalies and NVIC programming details, please refer to the Cortex-Mx Technology Reference Manual.

The EXTI module includes the edge detection circuit, which can produce the interrupt request or wake-up event, and the edge detection supports the rising edge, falling edge or any edge configuration. Each edge detection circuit supports the independent enable and mask.

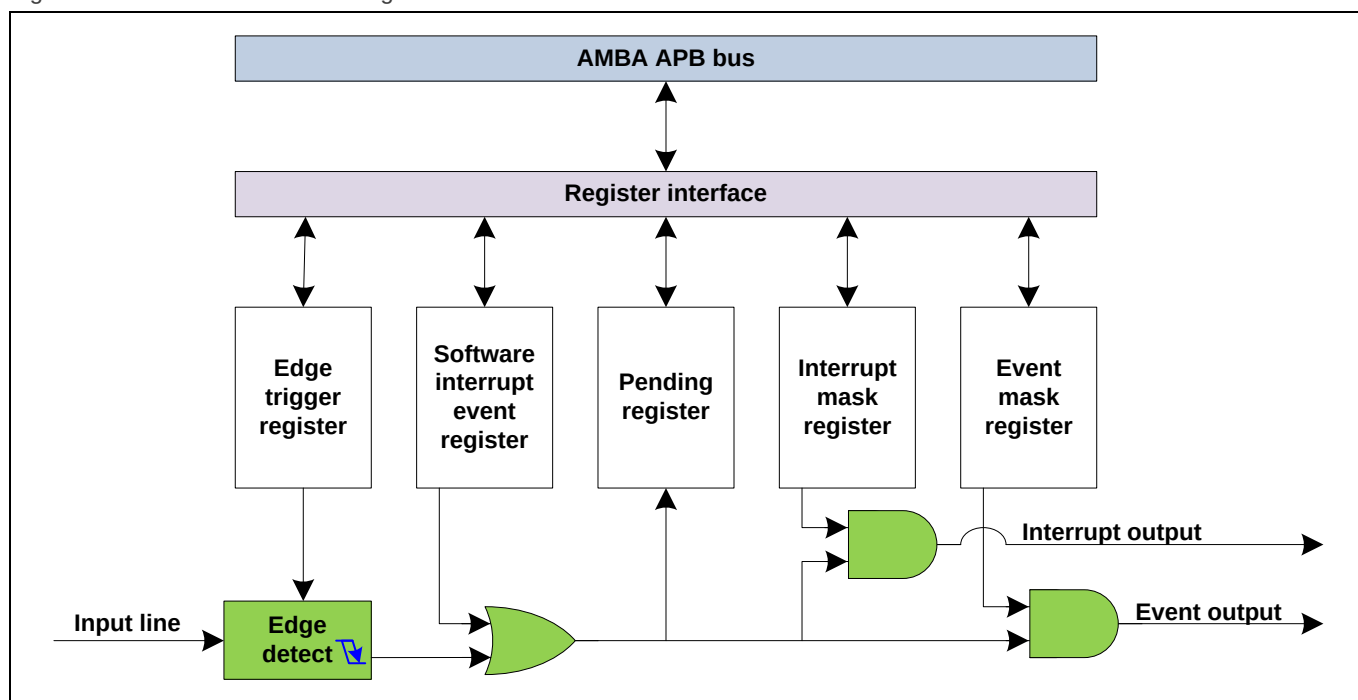
8.2 Main characteristics

- Independently trigger and mask each interrupt
- Configure interrupt/event output by software
- Produce the wake up event to wake up the low power mode
- The pending register saves the status of each corresponding interrupt line
- All GPIOs supports the trigger source configured as EXTI
- Support the rising edge trigger, falling edge trigger and any edge trigger

8.3 Function description

8.3.1 Function block diagram

Figure 8-1 EXTI structure block diagram



8.3.2 Interrupt and anomaly vector

Under Handler mode, Cortex-M0 processor and nested interrupt vector control (NVIC) can process all anomalies based on priority. In case of anomaly, the system will pop the current processing work stack after interrupt service. Simultaneous processing of vector and current work status improves the interrupt efficiency. The table below presents the anomaly type and interrupt vector.

Table 8-1 Abnormal vector

Position	Priority	Priority type	Name	Description	Address
				Reserved	0x0000 0000

Position	Priority	Priority type	Name	Description	Address
	-3	Fixed	Reset	Reset	0x0000 0004
	-2	Fixed	NMI	Non-maskable interrupt RCC Clock security system (CSS) connected to NMI	0x0000 0008
	-1	Fixed	Hardware fault	All types of faults	0x0000 000C

Table 8-2 Interrupt vector

Position	Priority	Priority type	Name	Description	Address
	3	Settable	SVCall	System service call via SWI instruction	0x0000 002C
	4	-		Reserved	0x0000 0030
		-		Reserved	0x0000 0034
	5	Settable	PendSV	Pending system service	0x0000 0038
	6	Settable	SysTick	System tick timer	0x0000 003C
0	7	Settable	WWDG_IWDG	WWDG and IWDG thru EXTI 17	0x0000_0040
1	8	Settable	PVD	PVD thru EXTI 16	0x0000_0044
2	9	Settable	Reserved	Reserved	0x0000_0048
3	10	Settable	FLASH	Flash interrupt	0x0000_004C
4	11	Settable	RCC	RCC interrupt	0x0000_0050
5	12	Settable	EXTI[1:0]	EXTI line [1:0] interrupt	0x0000_0054
6	13	Settable	EXTI[3:2]	EXTI line [3:2] interrupt	0x0000_0058
7	14	Settable	EXTI[15:4]	EXTI line [15:4] interrupt	0x0000_005C
8	15	Settable	Reserved	Reserved	0x0000_0060
9	16	Settable	Reserved	Reserved	0x0000_0064
10	17	Settable	Reserved	Reserved	0x0000_0068
11	18	Settable	Reserved	Reserved	0x0000_006C
12	19	Settable	ADC1_COMP	ADC1 global interrupt Comparator global interrupt	0x0000_0070
13	20	Settable	TIM1_BRK_UP_TRG_COM	TIM1 brake refresh trigger COM interrupt	0x0000_0074
14	21	Settable	TIM1_CC	TIM1 capture compare interrupt	0x0000_0078
15	22	Settable	Reserved	Reserved	0x0000_007C
16	23	Settable	TIM3	TIM3 global interrupt	0x0000_0080
17	24	Settable	Reserved	Reserved	0x0000_0084
18	25	Settable	Reserved	Reserved	0x0000_0088
19	26	Settable	TIM14	TIM14 global interrupt	0x0000_008C
20	27	Settable	Reserved	Reserved	0x0000_0090
21	28	Settable	Reserved	Reserved	0x0000_0094
22	29	Settable	Reserved	Reserved	0x0000_0098
23	30	Settable	I2C1	I2C1 global interrupt	0x0000_009C
24	31	Settable	Reserved	Reserved	0x0000_00A0
25	32	Settable	SPI1	SPI1 global interrupt	0x0000_00A4
26	33	Settable	Reserved	Reserved	0x0000_00A8
27	34	Settable	USART1	USART1 global interrupt	0x0000_00AC
28	35	Settable	USART2	USART2 global interrupt	0x0000_00B0
29	36	Settable	Reserved	Reserved	0x0000_00B4
30	37	Settable	Reserved	Reserved	0x0000_00B8
31	38	Settable	Reserved	Reserved	0x0000_00BC

8.3.3 Wake-up event management

All EXTI lines support the generation of interrupt or the event is used to wake the system from the low power mode. The user executes the WFE instruction to enter the corresponding low power mode. Configure the EXTI line

event output to wake up the system. The user executes the WFI to enter the low power mode, and configures EXTI line interrupt output to wake up the system. For the detailed configuration, please refer to the section of power control.

8.3.4 Interrupt function description

To enable interrupt function and produce interrupt, configure the edge detection trigger register as the required trigger type, and open the corresponding bit of interrupt mask register allowing interrupt request. When the corresponding external interrupt line detects the configured trigger condition, it produces an interrupt request and the corresponding pending position of register is 1. Write the corresponding bit of pending register as 1 and clear the interrupt.

To configure the generation event, configure the edge detection trigger register as the required trigger type, and open the corresponding bit of interrupt mask register allowing interrupt request. When the corresponding external interrupt line detects the configured trigger condition, it produces an interrupt request.

Enabling the corresponding bit of the software interrupt/event register can also produce the interrupt/event request.

8.3.5 Hardware interrupt output

Specific steps to configure hardware interrupt source:

- Open the mask bit of the corresponding interrupt line (EXTI_IMR), enable interrupt.
- Configure the trigger register bit (EXTI_RISE/EXTI_FALL) of the corresponding interrupt line;
- Open the NVIC connected interrupt channel so that the interrupt request can be transmitted to CPU and correctly responded.

When configuring EXTIx (x=31~0) line and produce interrupt output, the corresponding bit of EXTI_PR register will be set 1. It's required to clear the corresponding pending bit of EXTI_PR register before again detecting the rolling of EXTIx (x=31~0) line, and producing the interrupt.

There are three ways to clear the pending bit of EXTI_PR register:

- Write the pending bit of EXTI_PR register as 1.
- When configuring the rising edge trigger selection register (EXTI_RTSTR), writing 0 to the corresponding bit will clear the pending bit. When configuring the falling edge trigger selection register (EXTI_FTSTR), writing 0 in the corresponding bit will clear the pending bit.
- Clear by changing the edge detection polarity of EXTI line.

8.3.6 Hardware event output

Specific steps to configure hardware event source:

- Open the mask bit (EXTI_EMR) of the corresponding event line;
- Configure the trigger register bit (EXTI_RISE/EXTI_FALL) of the corresponding event line.

8.3.7 Software interrupt and event output

Specific steps to support configuration of interrupt and event by software:

- Enable event or interrupt enable bit (EXTI_IMR, EXTI_EMR).
- Configure the corresponding bit of the software interrupt event register as 1 (EXTI_SWIER).

8.3.8 External interrupt mapping

All GPIOs are used as the EXTI trigger source for producing the interrupt or event request. Configure SYSCFG_EXTICRx register, and meanwhile support the internal module (including PVD and IWDG) trigger. The specific connection relationships are shown as below:

Table 8-3 EXTI trigger source

External interrupt line	IO mapping	Control bit
EXTI0	PA0;PB	SYSCFG_EXTICR1 register EXTI0

External interrupt line	IO mapping	Control bit
EXTI1	PA1;PB1	SYSCFG_EXTICR1 register EXTI1
EXTI2	PA2	SYSCFG_EXTICR1 register EXTI2
EXTI3	PA3	SYSCFG_EXTICR1 register EXTI3
EXTI4	PA4	SYSCFG_EXTICR2 register EXTI4
EXTI5	PA5	SYSCFG_EXTICR2 register EXTI5
EXTI6	PA6	SYSCFG_EXTICR2 register EXTI6
EXTI7	PA7	SYSCFG_EXTICR2 register EXTI7
EXTI8	PA8	SYSCFG_EXTICR3 register EXTI8
EXTI9	PA9	SYSCFG_EXTICR3 register EXTI9
EXTI10	PA10	SYSCFG_EXTICR3 register EXTI10
EXTI11	PA11	SYSCFG_EXTICR3 register EXTI11
EXTI12	PA12	SYSCFG_EXTICR4 register EXTI12
EXTI13	PA13	SYSCFG_EXTICR4 register EXTI13
EXTI14	PA14	SYSCFG_EXTICR4 register EXTI14
EXTI15	PA15	SYSCFG_EXTICR4 register EXTI15

Other external interrupt/event controller connection:

- EXTI line 16 connected to PVD output
- EXTI line 17 connected to IWDG output
- EXTI line 19 connected to COMP output

8.4 Register

8.4.1 Register overview

Table 8-4 EXTI register overview

Offset	Acronym	Register Name	Reset
0x00	EXTI_IMR	Interrupt mask register	0x00000000
0x04	EXTI_EMR	Event mask register	0x00000000
0x08	EXTI_RTSTR	Rising edge trigger selection register	0x00000000
0x0C	EXTI_FTSR	Falling edge trigger selection register	0x00000000
0x10	EXTI_SWIER	Software interrupt event register	0x00000000
0x14	EXTI_PR	Pending register	0x00000000

8.4.2 EXTI_IMR Interrupt Mask Register

Address offset: 0x0

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.												IMR19	IMR18	IMR17	IMR16
												rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IMR15	IMR14	IMR13	IMR12	IMR11	IMR10	IMR9	IMR8	IMR7	IMR6	IMR5	IMR4	IMR3	IMR2	IMR1	IMR0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit	Field	Description
31:20	Reserved	Reserved, always read as 0
19:0	IMRx	Line x interrupt enable bit 1: Configure the bit as 1, enable line x corresponding interrupt 0: Configure the bit as 0, disable line x corresponding interrupt

8.4.3 EXTI_EMR Event Mask Register

Address offset: 0x04

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.												EMR19	EMR18	EMR17	EMR16
												rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EMR15	EMR14	EMR13	EMR12	EMR11	EMR10	EMR9	EMR8	EMR7	EMR6	EMR5	EMR4	EMR3	EMR2	EMR1	EMR0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit	Field	Description
31:20	Reserved	Reserved, always read as 0
19:0	EMRx	Line x event enable bit 1: Configure the bit as 1, enable line x corresponding event 0: Configure the bit as 0, disable line x corresponding event

8.4.4 EXTI_RTISR Rising Edge Trigger Selection Register

Address offset: 0x08

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.												TR19	TR18	TR17	TR16
												rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TR15	TR14	TR13	TR12	TR11	TR10	TR9	TR8	TR7	TR6	TR5	TR4	TR3	TR2	TR1	TR0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit	Field	Description
31:20	Reserved	Reserved, always read as 0
19:0	TRx	Line x corresponding interrupt or event trigger polarity 1: Configure the bit as 1, enable line x corresponding rising edge trigger interrupt or event 0: Configure the bit as 0, disable line x corresponding rising edge trigger interrupt or event

8.4.5 EXTI_FTSR Falling Edge Trigger Selection Register

Address offset: 0x0C

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.												TR19	TR18	TR17	TR16
												rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TR15	TR14	TR13	TR12	TR11	TR10	TR9	TR8	TR7	TR6	TR5	TR4	TR3	TR2	TR1	TR0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit	Field	Description
31:20	Reserved	Reserved, always read as 0
19:0	TRx	Line x corresponding interrupt or event trigger polarity 1: Configure the bit as 1, enable line x corresponding falling edge trigger interrupt or event 0: Configure the bit as 0, disable line x corresponding falling edge trigger interrupt or event

8.4.6 EXTI_SWIER Software Interrupt Event Register

Address offset: 0x10

Reset value: 0x0000 0000

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

Res.												SWIER19	SWIER18	SWIER17	SWIER16
												rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SWIER15	SWIER14	SWIER13	SWIER12	SWIER11	SWIER10	SWIER9	SWIER8	SWIER7	SWIER6	SWIER5	SWIER4	SWIER3	SWIER2	SWIER1	SWIER0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit	Field	Description
31:20	Reserved	Reserved, always read as 0
19:0	SWIERx	Line x software interrupt or event enable Writing 1 will set the corresponding pending bit of EXTI_PR register. Meanwhile, configuring the corresponding bit of EXTI_IMR or EXTI_EMR as 1 can produce interrupt or event.
		Note: Writing 1 to the corresponding bit of EXTI_PR register may clear the bit

8.4.7 EXTI_PR Software Interrupt Event Pending Register

Address offset: 0x14

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.												PR19	PR18	PR17	PR16
												rc_w1	rc_w1	rc_w1	rc_w1
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PR15	PR14	PR13	PR12	PR11	PR10	PR9	PR8	PR7	PR6	PR5	PR4	PR3	PR2	PR1	PR0
rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1

Bit	Field	Description
31:20	Reserved	Reserved, always read as 0
19:0	PRx	Line x trigger pending bit 1: Selected trigger request 0: No trigger request In case of the selected edge event on the external interrupt line, the bit is set as 1. Writing 1 clears the bit, or clear by changing the edge detection polarity.

9 ADC Analog-to-Digital ConverterADC introduction

The 12-bit ADC is a successive approximation analog-to-digital converter (SAR A/D converter).

A/D converter supports multiple operation modes: single conversion, continuous conversion, or automatic scan through selected channel. The ADC converter can be started by software, external pin trigger, and individual timers.

The window comparator (analog watchdog) allows the application to detect if the input voltage goes outside the user-defined high or low thresholds.

The ADC input clock is generated from the PCLK1 clock divided by a prescaler and it must not exceed 16MHz...

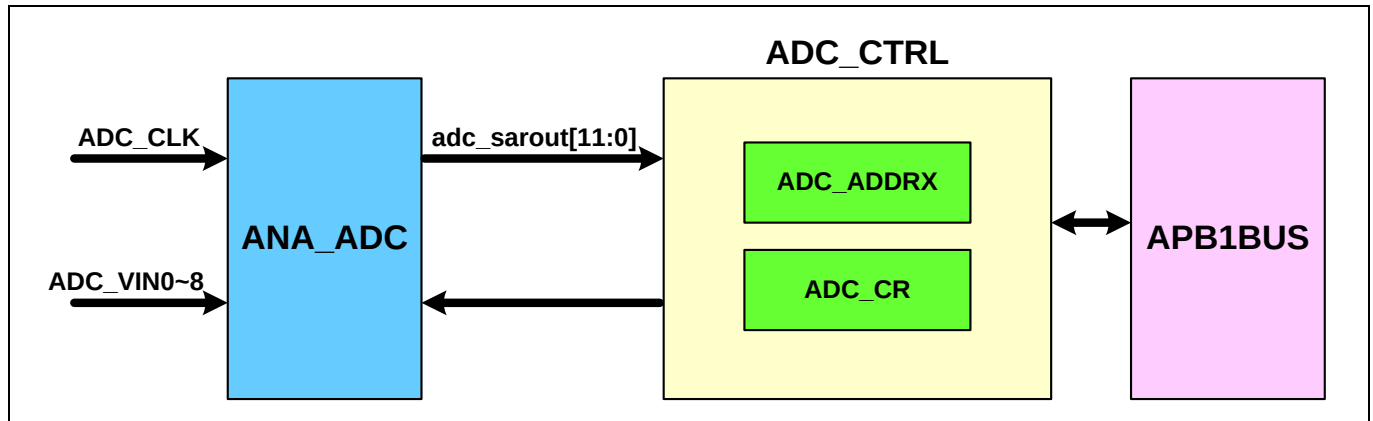
9.2 ADC main features

- SAR ADC with up to 12-bit programmable resolution; no less than 8 multiplexed external input channels and 1 internal channel
- Conversion rate up to 1Msps
- Support normal operation mode:
 - ◆ Single conversion mode: A/D converter does one conversion in the specified channel
 - ◆ One-cycle scan mode: A/D converter does one-cycle conversion in all specified channels (from low sequence number channel to high sequence number channel, or from high sequence number channel to low sequence number channel)
 - ◆ Continuous scan mode: A/D converter does continuous one-cycle conversions until the software stops A/D conversion. A/D conversion must stop if the conversion channel needs to be changed in the midway. The conversion can restart after configuring the corresponding register.
- Support arbitrary channel operation mode:
 - ◆ Single conversion mode: A/D converter does one conversion in the specified channel
 - ◆ One-cycle scan mode: A/D converter does one-cycle conversion in all specified channels (in any sequence)
 - ◆ Continuous scan mode: A/D converter does continuous one-cycle conversions until the software stops A/D conversion. A/D conversion doesn't have to stop if the conversion channel needs to be changed. The conversion in a new channel will start in the next scan cycle after configuring the corresponding register
- Support the configuration of channel sampling time and resolution via software
- A/D conversion start condition:
 - ◆ By software
 - ◆ By external trigger with the configuration of delay via software
- A/D conversion start conditions
 - ◆ Software start
 - ◆ Triggered start with configurable trigger delay
 - ◆ Timer1/3 matching or TRGO signal, external EXTI signal source
- Analog watchdog for comparing the conversion result with the specified value; An interrupt generation can be set by the user when the conversion result matches with the specified value.

9.3 ADC system block diagram

The system block diagram of the ADC is as follows:

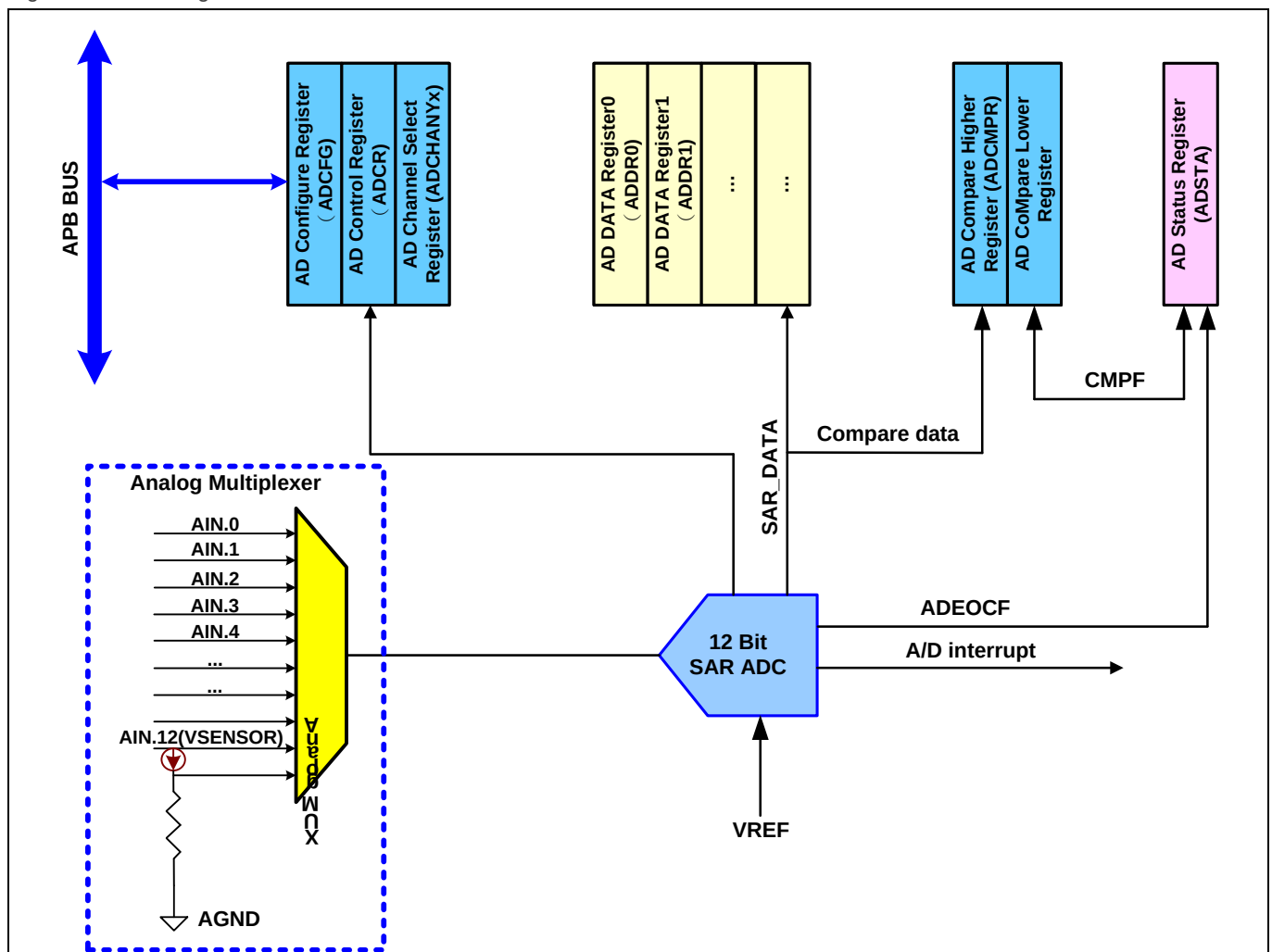
Figure 9-1 ADC system block diagram



9.4 ADC Functional Description

The ADC diagram is as follows:

Figure 9-2 ADC diagram



Note: V_SENSOR (internal reference voltage) is in the ADC AIN12 channel.

9.4.1 ADC on-off control

The ADC can be powered-on by setting the ADON bit in the ADCFG register. When the ADEN bit is set for the first time, it wakes up the ADC from Power Down mode.

Conversion starts when ADST bit is set for the ADCR register after an ADC power-up delay has elapsed (around 200ns).

The conversion can be stopped by resetting the ADST bit, and the ADC put in power down mode by resetting the ADEN bit

9.4.2 Channel Selection

ADC1 has 8 multiplexed channels including external input channels and internal 1.2V reference voltage channel. Each external input channel has an independent enable bit that can be set by CHANY_NUM, ADC_CHANY0 and ADC_CHANY1.

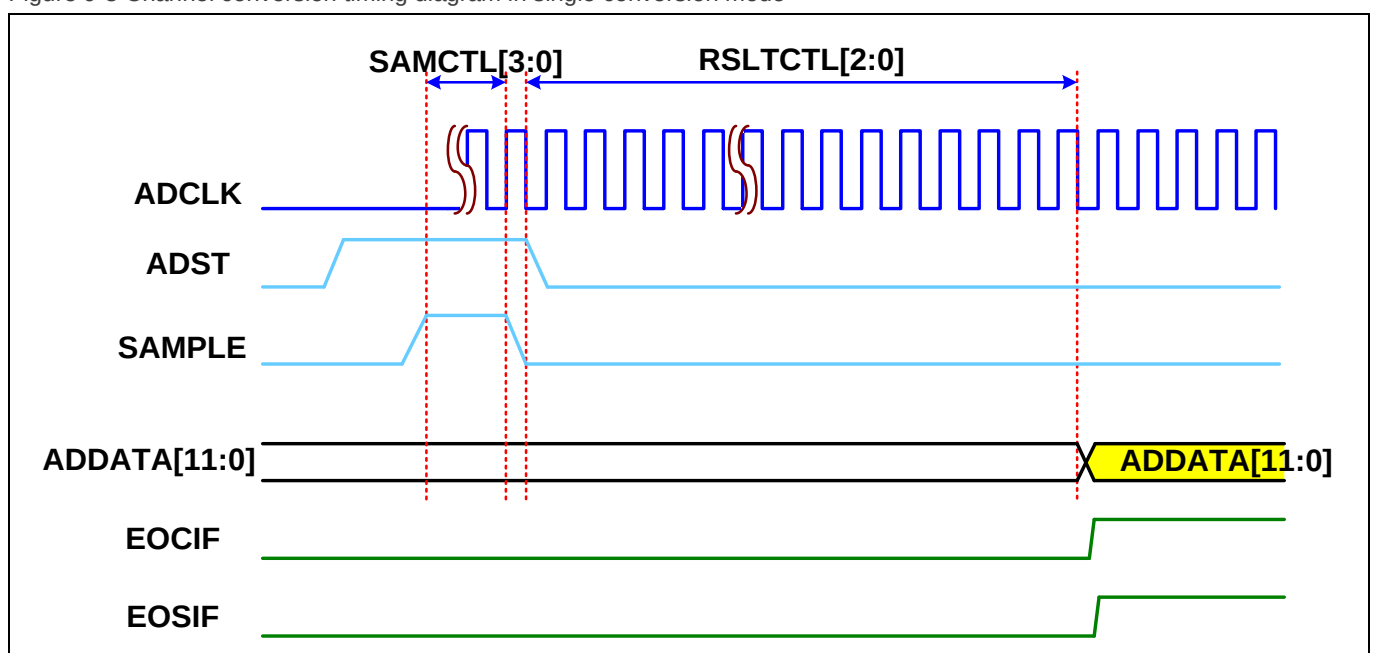
9.5 Arbitrary channel operation mode

9.5.1 Single Conversion Mode

In single conversion mode the A/D converter does one conversion in the corresponding channel. The specific procedures are as follows:

- Set by software the ADC_ANY_CFG, ADC_CHANY0, and ADC_CHANY1 registers, the conversion channel, and the CHANY_MDEN bit. (only CHANY_SELO has to be set for the single conversion mode)
- Set the ADST bit in the ADCR register by software, external trigger input, or timer overflow and start the A/D conversion.
- After A/D conversion, the converted data is stored in the data registers (ADDATA and ADDRn).
- After A/D conversion, set the ADIF bit of the status register ADSTA to '1'. At this time, an AD EOC (End Of Conversion) interrupt request is generate if the ADIE bit in the control register ADCR is set to '1'.
- ADST bit remains 1 during the A/D conversion period. Once the A/D channel sampling is complete, the ADST is automatically cleared and the A/D converter enters the idle mode.
- If the software updates ADC_ANY_CFG, ADC_CHANY0, and ADC_CHANY1 during the A/D conversion period, the hardware will not update these settings immediately. The updates will be applied when the conversion of currently configured channels is complete, waiting for setting ADST by software next time.

Figure 9-3 Channel conversion timing diagram in single conversion mode

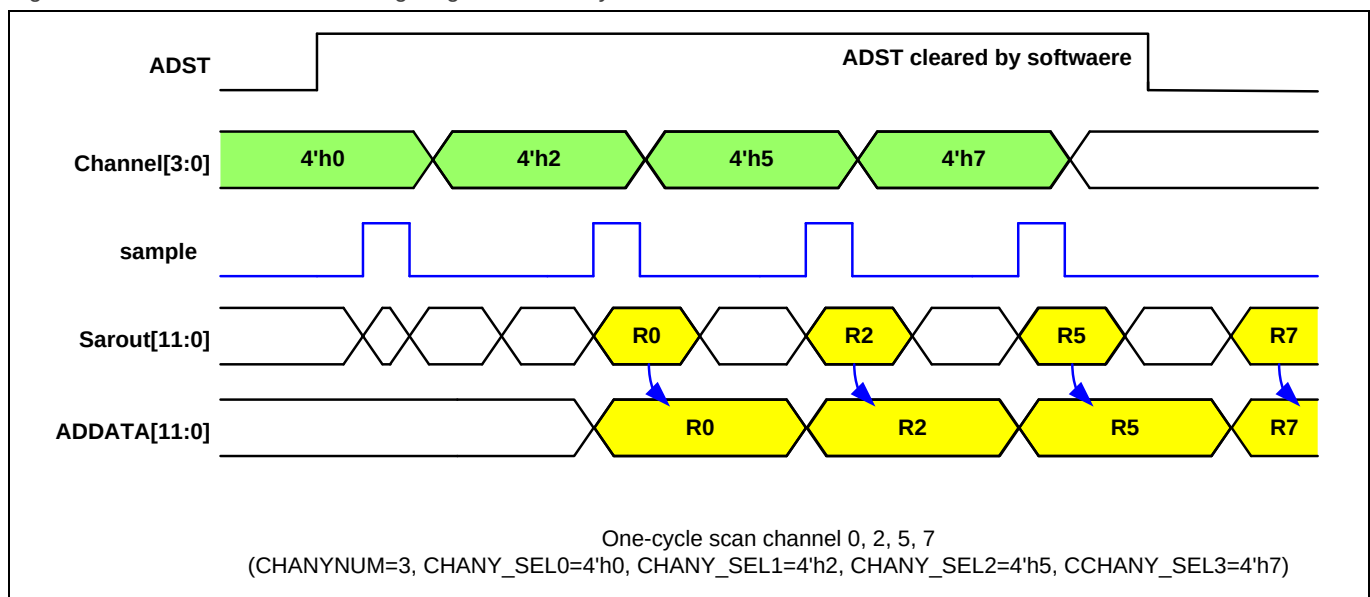


9.5.2 One-cycle scan mode

In one-cycle scan mode the A/D converter does one conversion according to the software configuration. The specific procedures are as follows:

- Set by software the ADC_ANY_CFG, ADC_CHANY0, and ADC_CHANY1 registers, desired channels and quantity, and the CHANY_MDEN bit.
- Set the ADST bit in the ADCR register by software or external trigger with the configuration of trigger delay via software. A/D conversion direction is from CHANY_SEL0 to CHANY_SEL12. The number of conversion channels is configured by CHANY_NUM. The number is arbitrarily configured from CHANY_SEL0 to CHANY_SEL12, which can be identical or not.
- After the end of A/D conversion in each channel, the converted values will be loaded to the data register of corresponding channel in order. The ADIF EOC flag is set. At this time, an AD EOC interrupt request is generate if the ADIE bit in the control register ADCR is set to '1'.
- Once the final A/D channel sampling is complete, the ADST is automatically cleared and the A/D converter enters the idle mode.
- If the software updates ADC_ANY_CFG, ADC_CHANY0, and ADC_CHANY1 during the A/D conversion period, the hardware will not update these settings immediately. The updates will be applied when the conversion of currently configured channels is complete, waiting for setting ADST by software next time.

Figure 9-4 Channel conversion timing diagram in One-cycle conversion mode



9.5.3 Continuous Scan Mode

In continuous scan mode the A/D converter does continuous conversions according to the software configuration until the software stops conversion. The specific procedures are as follows:

- Set by software the ADC_ANY_CFG, ADC_CHANY0, and ADC_CHANY1 registers, desired channels and quantity, and the CHANY_MDEN bit.
- Set the ADST bit in the ADCR register by software or external trigger with the configuration of trigger delay via software. A/D conversion direction is from CHANY_SEL0 to CHANY_SEL12. The number of conversion channels is configured by CHANY_NUM. The number is arbitrarily configured from CHANY_SEL0 to CHANY_SEL12, which can be identical or not.
- After the end of A/D conversion in each channel, the converted values will be loaded to the data register of corresponding channel in order. The ADIF EOC flag is set. At this time, an AD EOC interrupt request is generate if the ADIE bit in the control register ADCR is set to '1'.
- The A/D conversion continues as long as the ADST bit remains 1. Once the ADST bit is cleared and the current A/D conversion is complete, the A/D converter enters the idle mode.

- If the software updates ADC_ANY_CFG, ADC_CHANY0, and ADC_CHANY1 during the A/D conversion period, the hardware will not update these settings immediately. The updates will be applied when the conversion of current configured channels is complete, i.e. a conversion in the new channel starts in the next scan cycle.

Figure 9-5 Channel conversion timing diagram in Continuous scan mode

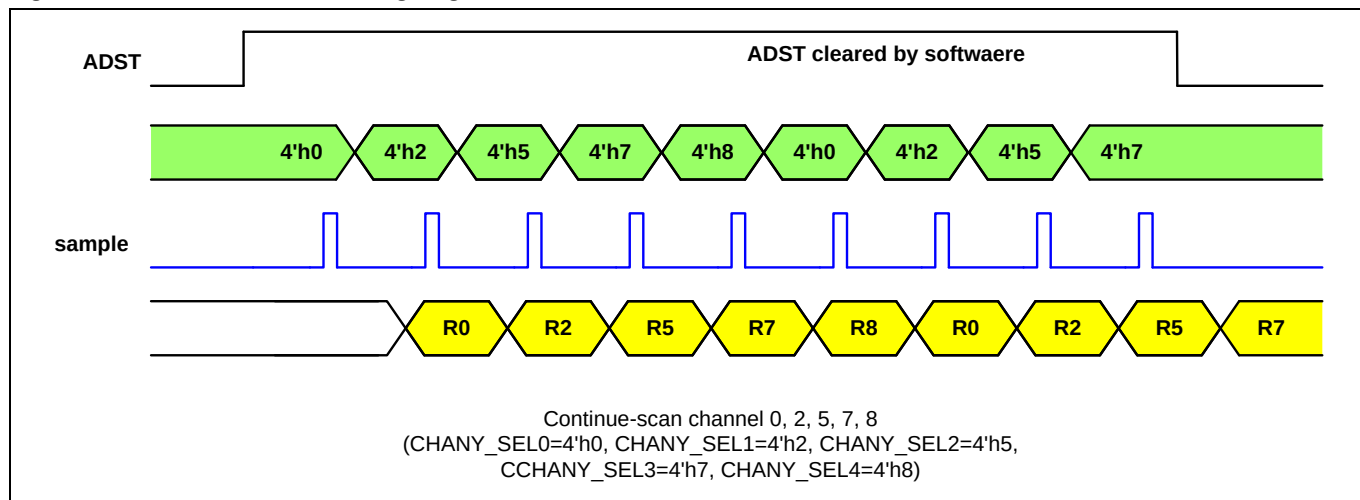
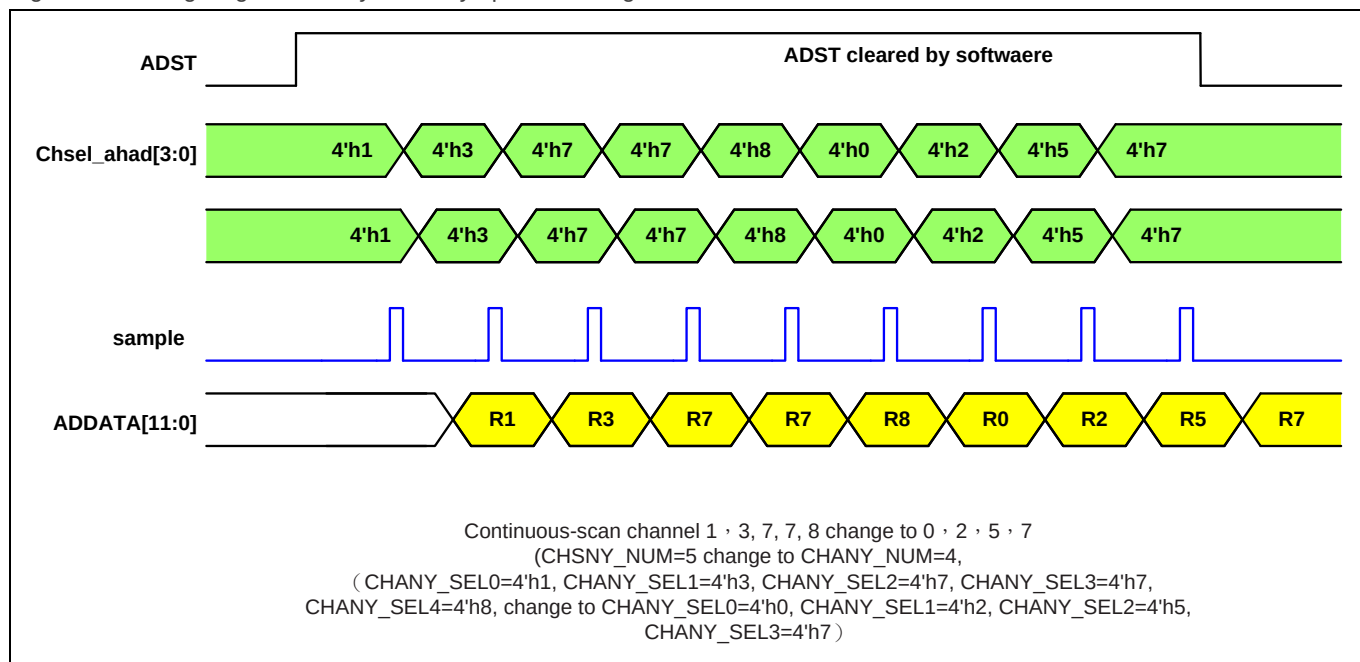


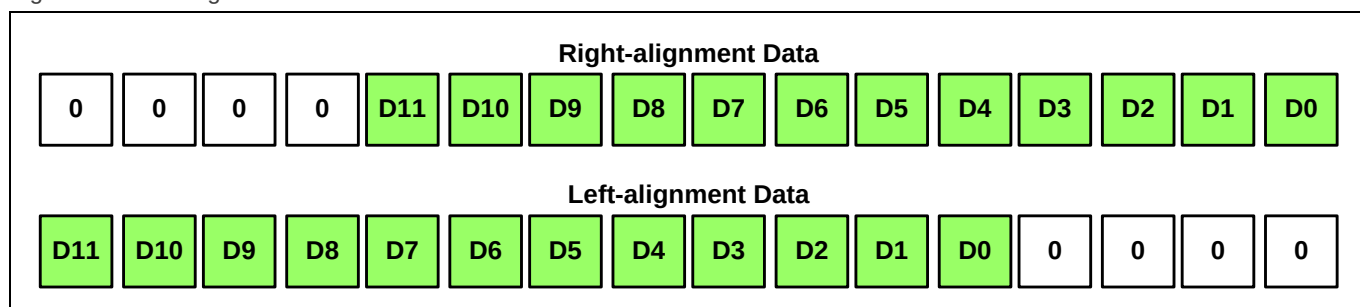
Figure 9-6 Timing diagram with dynamically updated configuration, in Continuous scan mode



9.6 Data Alignment

The ALIGN bit in the ADCR register selects the alignment mode for storing converted data. The data can be left-aligned or right-aligned, as shown in the figure below.

Figure 9-7 Data alignment mode



9.7 Programmable Resolution

The ADC conversion significance can be modified using the RSLTCTL [2:0] bits in the ADC_CFG register to accelerate the data conversion speed. The significant data bits are aligned from the higher bits of 12-bit data.

9.8 Programmable Sampling Time

The ADC clock ADCLK is generated from the PCLK 1 divided by a prescaler. The prescale coefficient can be determined by setting the ADCPRE bit in the ADCFG register, i.e. $PCLK1 / (N + 2)$ divided by a prescaler to be the ADC clock. ADC samples the input voltage for a number of ADC_CLK cycles which can be modified using the SAMCTL [3:0] bits in the ADC_CFG registers.

Set the ADC resolution as n bits (n=8, 9, 10, 11, 12). The sampling cycles of each channel is m.

The sampling frequency is calculated as follows:

$$F_{\text{sample}} = F_{\text{ADCLK}} / (m + n + 0.5).$$

Assuming the resolution is configured as 12 bits and each channel has a sampling period of 3.5T, then $F_{\text{sample}} = F_{\text{ADCLK}} / 16$.

The total conversion time is calculated as follows:

$$T_{\text{CONV}} = \text{Sampling time} + 12.5 \text{ conversion cycles}.$$

For example:

When ADCCLK = 16MHz and the sampling time is 3.5 periods, $T_{\text{CONV}} = 3.5 + 12.5 = 16$, cycle=1μs.

9.9 Conversion on external trigger

ADC conversion can be triggered by an external event (e.g. timer capture, EXTI line). After the trigger signal is generated, the sampling is delayed by N PCLK1 clock cycles before starting again. In case of trigger scan mode, only the first channel sample is delayed, the rest of the channels are started immediately after the end of the previous sample.

If the TRGEN bit of the ADCR register is set, an external event can be used to trigger the conversion. The external trigger source can be selected by setting the TRGSEL bit.

For specific external trigger source selection, refer to the description of the relevant bits in the ADCR control register (ADCR.TRGSEL).

External triggering can be set for delay control, refer to the description of the relevant bits in the ADCR control register (ADCR.TRGSHIFT).

9.10 Internal Reference Voltage

The internal signal source channel of ADC is connected to an internal reference voltage of 1.2V. This channel converts the 1.2V reference voltage output to a digital value.

The internal reference voltage has an independent enable bit that can be enabled or disabled by setting the corresponding bit in the register.

9.11 Monitoring ADC Conversion Results in Window Comparator Mode

An upper limit comparison register and a lower limit comparison register are provided in the comparison mode. The monitoring channel can be selected by setting the CMPCH bit via software.

Providing $CPMHDATA \geq CPMLDATA$, and the comparison result is greater than or equal to the specified CPMHDATA value or less than the specified CPMLDATA value in the ADCMPR register, the ADWIF bit in the status register ADSTA is set to 1.

Providing $CPMHDATA < CPMLDATA$, and the comparison result is equal to the specified CPMHDATA value or between two specified values, the ADWIF bit in the status register ADSTA is set to 1.

An interrupt request is generate if the ADWIE bit in the control register ADCR is set.

9.12 ADC register description

9.12.1 Register Overview

Table 9-1 ADC register overview

Offset	Acronym	Register Name	Reset
0x00	ADC_ADDDATA	A/D data register	0x00000000
0x04	ADC_ADCFG	A/D configuration register	0x00000000
0x08	ADC_ADCR	A/D control register	0x00000000
0x10	ADC_ADCMPR	A/D analog watchdog comparison register	0x00000000
0x14	ADC_ADSTA	A/D status register	0x00000000
0x18~0x48	ADC_ADDR 0~12	A/D channel data register	0x00000000
0x58	ADC_ADSTA_EXT	A/D Extended State Register	0x00000000
0x5C	ADC_CHANY0	A/D arbitrary channel selection register 0	0x00000000
0x60	ADC_CHANY1	A/D arbitrary channel selection register 1	0x00000000
0x64	ADC_ANY_CFG	A/D arbitrary channel configuration register	0x00000000
0x68	ADC_ANY_CR	A/D arbitrary channel control register	0x00000000

9.12.2 A/D data register (ADC_ADDDATA)

Address offset: 0x00

Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.										VALID	OVERRUN	CHANNELSEL			
										r	r	r			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA															
r															

Bit	Field	Description
31:22	Reserved	Reserved, always read as 0.
21	VALID	Data Valid Flag (Read-only) 1: DATA[11:0] data is valid 0: DATA[11:0] data is invalid This bit is set after the corresponding analog channel conversion is complete. This bit is cleared by hardware after reading the ADDDATA register.
20	OVERRUN	Data Overrun Flag (Read-only) 1: DATA[11:0] data is overrun 0: DATA[11:0] data is the last conversion result Before the new conversion result is loaded to the register, OVERRUN is set to 1 if the DATA [11:0] is not read. This bit is cleared by hardware after reading the ADDDATA register.

Bit	Field	Description
19:16	CHANNELSEL	4 bits show the channel corresponding to the current data (Channel selection) 0000: conversion data of channel 0. 0001: conversion data of channel 1. 0010: conversion data of channel 2. 0011: conversion data of channel 3. 0100: conversion data of channel 4. 0101: conversion data of channel 5. 0110: conversion data of channel 6. 0111: conversion data of channel 7. 1000: conversion data of channel 8. 1001: conversion data of channel 9. 1010: conversion data of channel 10. 1011: conversion data of channel 11. 1100: conversion data of internal reference voltage. Other: Invalid.
15:0	DATA	12-bit A/D conversion result (Transfer data) Data can be left or right aligned as configured.

9.12.3 A/D configuration register (ADC_ADCFG)

Address offset: 0x04

Reset value: 0x00000000

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.	ADCPREL	SAMCTL				RSLTCTL			ADCPREH			VSEN	Res.	ADWEN	ADEN
	rw	rw				rw			rw			rw		rw	rw

Bit	Field	Description
31:15	Reserved	Reserved, always read as 0.
14	ADCPREL	ADC Prescaler Low Bits Prescaler coefficient ADCPRE={ADCPREH, ADCPREL}
13: 10	SAMCTL	Channel x Sample time selection These bits are used to independently select the sample time for each channel. In a sample cycle, the channel selection bit should remain unchanged. 0000: 2.5 cycles 0100: 42.5 cycles 0001: 8.5 cycles 0101: 56.5 cycles 0010: 14.5 cycles 0110: 72.5 cycles 0011: 29.5 cycles 0111: 240.5 cycles 1000: 3.5 cycles 1001: 4.5 cycles 1010: 5.5 cycles 1011: 6.5 cycles 1100: 7.5 cycles Others: Reserved
9: 7	RSLTCTL	ADC Conversion Data Resolution Selection 000: 12-bit effective 001: 11-bit effective 010: 10-bit effective 011: 9-bit effective 100: 8-bit effective Others: Reserved
6: 4	ADCPREH	ADC Prescaler High Bits Prescaler coefficient ADCPRE = {ADCPREH, ADCPREL} ADC clock division : div= (ADCPRE+2)
3	VSEN	Voltage Sensor Enable 1: Internal voltage sensor enabled 0: Internal voltage sensor disabled
2	Reserved	Reserved and always read as 0.
1	ADWEN	ADC window comparison enable 1: A/D window comparison enabled 0: A/D window comparison disabled

Bit	Field	Description
0	ADEN	A/D conversion enable (ADC enable) 1 : Enable 0 : Disable

9.12.4 ADC_ADCR Control Register

Offset address: 0x08

Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.						TRG_EDGE	Res.	TRGSHIFT			TRGSELH		Res.		
						rw		rw			rw				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMPCH				ALIGN	ADMD		ADST	Res.	TRGSELL			Res.	TRGEN	AWDIE	ADIE
rw				rw	rw		rw		rw				rw	rw	rw

Bit	Field	Description
31:26	Reserved	Reserved, always read as 0.
25 : 24	TRG_EDGE	Trigger Sources Edge Selection 00: Dual edge trigger 01: Falling edge trigger 10: Rising edge trigger 11: Trigger blocked
23:22	Reserved	Reserved, always read as 0.
21 : 19	TRGSHIFT	External trigger shift sample After a trigger signal is generated, the sampling starts after a delay of N PCLK1 clock cycles. In case of trigger scan mode, the sampling in the rest channels starts immediately after the last sampling is complete. 000: No delay 001: 4 cycles 010: 16 cycles 011: 32 cycles 100: 64 cycles 101: 128 cycles 110: 256 cycles 111: 512 cycles
18 :17	TRGSELH	External Trigger Sources Select For High Bits TRGSEL={TRGSELH,TRGSELL}
16	Reserved	Reserved, always read as 0.
15 : 12	CMPCH	Window comparison channel selection 0000: conversion result for selecting comparison channel 0 0001: conversion result for selecting comparison channel 1 0010: conversion result for selecting comparison channel 2 0011: conversion result for selecting comparison channel 3 0100: conversion result for selecting comparison channel 4 0101: conversion result for selecting comparison channel 5 0110: conversion result for selecting comparison channel 6 0111: conversion result for selecting comparison channel 7 1000: conversion result for selecting comparison channel 8 1001: conversion result for selecting comparison channel 9 1010: conversion result for selecting comparison channel 10 1011: conversion result for selecting comparison channel 11 1100: conversion result for selecting internal comparison reference voltage 1111: All scan channels Others: Invalid
11	ALIGN	Data Alignment 1: Left alignment 0: Right alignment

Bit	Field	Description
10 : 9	ADMD	A/D conversion mode (ADC mode) 00: Single conversion 01: Single-cycle scan 10: Continuous scan 11: Reserved When the conversion mode is changed, the software has to disable the ADST bit.
8	ADST	A/D conversion start 1: Conversion start 0: Conversion end or idle state There are two ways of setting ADST: In the single or one-cycle mode, ADST is automatically cleared by hardware after conversion. In continuous scan mode the A/D converter does continuous conversions until the software writes '0' to the bit or the system is reset.
7	Reserved	Reserved, always read as 0.
6: 4	TRGSELL	External trigger selection, bits [18:17,6:4]: Select the external trigger source. 00000: TIM1_CC1 00001: TIM1_CC2 00010: TIM1_CC3 00011: Reserved 00100: TIM3_TRGO 00101: TIM1_CC4 and TIM1_CC5 00110: TIM3_CC1 00111: EXTI Line 11 01000: TIM1_TRGO 01001: Reserved 01010: Reserved 01011: Reserved 01100: TIM3_CC4 01101: Reserved 01110: Reserved 01111: EXTI Line 15 10000: TIM1_CC4 10001: TIM1_CC5 Others: Invalid
3	Reserved	Reserved, always read as 0.
2	TRGEN	External hardware trigger source (External trigger enable) 1: enable A/D conversion by external trigger signal 0: not enable A/D conversion by external trigger signal
1	AWDIE	ADC Window Comparator Interrupt Enable 1: Enable A/D window comparator interrupt 0: Disable A/D window comparator interrupt
0	ADIE	ADC Interrupt Enable 1: Enable A/D interrupt 0: Disable A/D interrupt An A/D EOC interrupt request is generated if the ADIF bit is set.

9.12.5 A/D window comparison register (ADC_ADCMPR)

Address offset: 0x10

Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.				CMPHDATA											
				rw											
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.				CMLPLDATA											
				rw											

Bit	Field	Description
31:28	Reserved	Reserved, always read as 0.
27: 16	CMPHDATA	Compare data high limit The 12-bit value will be compared with the conversion result of the specified channel.
15:12	Reserved	Reserved, always read as 0.
11 : 0	CMPLDATA	Compare data low limit The 12-bit value will be compared with the conversion result of the specified channel.

9.12.6 ADC_ADSTA Status Register

Offset address: 0x14

Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OVERRUN												VALID			
r												r			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VALID								CHANNEL				Res.	BUSY	AWDIF	ADIF
r								r					r	rc_w1	rc_w1

Bit	Field	Description
31:20	OVERRUN	Data overrun flag for channel 0 ~ 11 (Overrun flag) Read-only.
19:8	VALID	Valid flag for channel 0 ~ 11 (Valid flag) Read-only.
7:4	CHANNEL	Current conversion channel These 4 bits indicate the channel in conversion when BUSY = 1. These 4 bits indicate the channel for the next conversion when BUSY = 0
3	Reserved	Reserved, always read as 0.
2	BUSY	Busy/Idle 1: A/D converter is busy 0: A/D converter is idle
1	AWDIF	ADC window comparator interrupt flag Providing CPMHDATA ≥ CPMLDATA, and the selected ADC channel comparison result is greater than or equal to the specified CMPHDATA value or less than the specified CMPLDATA value in the ADCMPR register, the ADWIF bit in the status register ADSTA is set to 1. Providing CPMHDATA < CPMLDATA, and the selected ADC channel comparison result is equal to the specified CPMHDATA value or between two specified values, the ADWIF bit in the status register ADSTA is set to 1. This flag bit is cleared by writing '1'
0	ADIF	ADC interrupt flag This bit is set by the hardware when the channel group conversion is completed and cleared by software. 1: A/D conversion completed 0: A/D conversion not completed This flag bit is cleared by writing '1'

9.12.7 A/D data register (ADC_ADDR0 ~ 12)

Address offset: 0x18~0x48

Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.										VALID	OVERRUN	Res.			
										r	r				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

DATA															
r															
Bit	Field			Description											
31:22	Reserved			Reserved, always read as 0.											
21	VALID			Valid Flag (read-only): 1: DATA [11:0] bits data is valid. 0: DATA [11:0] bits data is invalid. This bit is set after the corresponding analog channel conversion is complete. This bit is cleared by hardware after reading the ADDATA register.											
20	OVERRUN			Data overrun flag (read-only) 1: DATA [11:0] data is overrun 0: DATA [11:0] data is the last conversion result Before the new conversion result is loaded to the register, OVERRUN is set to '1' if the DATA [11:0] is not read. This bit is cleared by hardware after reading the ADDATA register.											
19 : 16	Reserved			Reserved, always read as 0.											
15 : 0	DATA			12-bit A/D conversion result in the channel (Transfer data) Data can be left or right aligned as configured.											

9.12.8 A/D Extended State Register (ADC_ADSTA_EXT)

Address offset: 0x58

Reset value: 0x00000000

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.											OVERRUN	Res.			VALID
											r				r

Bit	Field			Description											
31:5	Reserved			Reserved, always read as 0.											
4	OVERRUN			Data overrun flag for internal voltage sensor (Overrun Flag)											
3:1	Reserved			Reserved, always read as 0.											
0	VALID			Valid flag for internal voltage sensor (Valid Flag)OR											

9.12.9 A/D arbitrary channel selection register (ADC_CHANY0)

Address offset: 0x5C

Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CHANY_SEL7				CHANY_SEL6				CHANY_SEL5				CHANY_SEL4			
rw				rw				rw				rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CHANY_SEL3				CHANY_SEL2				CHANY_SEL1				CHANY_SEL0			
rw				rw				rw				rw			

Bit	Field			Description											
31:28	CHANY_SEL7			Can be configured to any channel from channel 0 ~ 12.											
27:24	CHANY_SEL6			Can be configured to any channel from channel 0 ~ 12.											
23:20	CHANY_SEL5			Can be configured to any channel from channel 0 ~ 12.											
19:16	CHANY_SEL4			Can be configured to any channel from channel 0 ~ 12.											

Bit	Field	Description
15:12	CHANY_SEL3	Can be configured to any channel from channel 0 ~ 12.
11:8	CHANY_SEL2	Can be configured to any channel from channel 0 ~ 12.
7:4	CHANY_SEL1	Can be configured to any channel from channel 0 ~ 12.
3:0	CHANY_SEL0	Can be configured to any channel from channel 0 ~ 12.

Note: In the one-cycle or continuous scan modes, the ADC_CHANY0 shadow register is started by hardware. Before ADC starts to work, writing to ADC_CHANY0 by software also writes to its shadow register. When ADC is working, only the shadow register is updated if the ADC_CHANY0 value is changed. Besides that, when ADC starts to convert the last channel, the value of the shadow register will update to ADC_CHANY0, so as to complete the dynamic channel switch.

9.12.10 A/D arbitrary channel selection register 1 (ADC_CHANY1)

Address offset: 0x60

Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.												CHANY_SEL12			
												rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CHANY_SEL11				CHANY_SEL10				CHANY_SEL9				CHANY_SEL8			
rw				rw				rw				rw			

Bit	Field	Description
31:20	Reserved	Reserved, always read as 0.
19:16	CHANY_SEL12	Can be configured to any channel from channel 0 ~ 12.
15:12	CHANY_SEL11	Can be configured to any channel from channel 0 ~ 12.
11:8	CHANY_SEL10	Can be configured to any channel from channel 0 ~ 12.
7:4	CHANY_SEL9	Can be configured to any channel from channel 0 ~ 12.
3:0	CHANY_SEL8	Can be configured to any channel from channel 0 ~ 12.

Note: In the one-cycle or continuous scan modes, the ADC_CHANY1 shadow register is started by hardware. Before ADC starts to work, writing to ADC_CHANY1 by software also writes to its shadow register. When ADC is working, only the shadow register is updated if the ADC_CHANY1 value is changed. Besides that, when ADC starts to convert the last channel, the value of the shadow register will update to ADC_CHANY1, so as to complete the dynamic channel switch.

9.12.11 A/D arbitrary channel configuration register (ADC_ANY_CFG)

Address offset: 0x64

Reset value: 0x00000000

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.												CHANY_NUM			
												rw			

Bit	Field	Description
31:4	Reserved	Reserved, always read as 0.

Bit	Field	Description
3: 0	CHANY_NUM	Number of Any Channel Mode: 0000 : Channel 0 0001 : Channel 0~1 0010 : Channel 0~2 0011 : Channel 0~3 0100 : Channel 0~4 0101 : Channel 0~5 0110 : Channel 0~6 0111 : Channel 0~7 1000 : Channel 0~8 1001 : Channel 0~9 1010 : Channel 0~10 1011 : Channel 0~11 1100 : Channel 0~12 Other : Invalid

Note: In the one-cycle or continuous scan modes, the ADC_NUM shadow register is started by hardware. Before ADC starts to work, writing to ADC_NUM by software also writes to its shadow register. When ADC is working, only the shadow register is updated if the ADC_NUM value is changed. Besides that, when ADC starts to convert the last channel, the value of the shadow register will update to ADC_NUM, so as to complete the dynamic channel switch.

9.12.12 A/D arbitrary channel control register (ADC_ANY_CR)

Address offset: 0x68

Reset value: 0x00000000

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.														CHANY_MDEN	
														rw	

Bit	Field	Description
31:1	Reserved	Reserved, always read as 0.
0	CHANY_MDEN	Arbitrary channel configuration mode enable bit: 1: Enable 0: Disable When this bit is enabled, the function of ADC channel configuration become changed. There are two controlling parts: CHANY_NUM is used to configure the number of channels from channel 0 ~ 12. CHANY_SELO ~ CHANY_SEL12 is used to configure them as an arbitrary ADC channel.

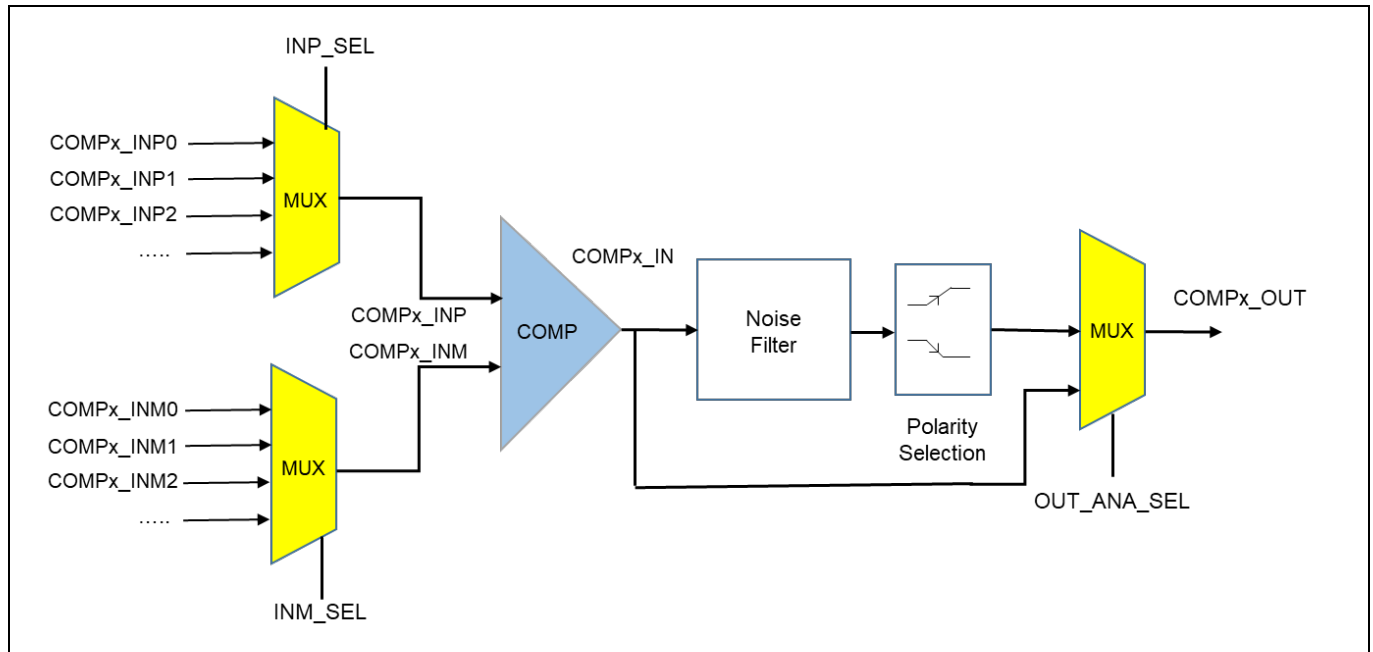
Note: In the arbitrary mode plus the one-cycle/continuous scan mode, the ADST bit in the ADC_ADCR must be disabled before closing the ADC. Then the user should judge whether the BUSY bit in the ADC_ADSTA is 0. That is to say, the CHANY_MDEN bit in the ADC_ANY_CR should be disabled when the ADC conversion is complete.

10 COMP Comparator Overview

The chip embeds one general purpose comparator (COMP1). The comparator has an integrated digital filter, and its comparison result can be output to a timer or used to generate an interrupt, or a wake-up event triggering the low-power mode. It can also be used with a timer to combine with a PWM output from the timer, generating a cycle-by-cycle current control loop.

10.2 Function block diagram

Figure 10-1 Comparator block diagrams



10.3 Main characteristics

- Analog input from alternate I/O pins;
- Programmable hysteresis voltage;
- Comparison result filtering is supported and filtering cycle is configurable;
- Outputs to I/O pins or timers;
- Supports to wake up the CPU from Sleep and Stop modes through the external event EXTI;
- Each comparator supports 4 positive and 5 negative inputs with polling function;
- Supports the polling switch in fixed cycles;
- Supports the control of polling channel 1/2/3 or 1/2;
- Supports synchronized polling of positive and negative inputs or fixed negative inputs

10.4 Functional description

10.4.1 COMP clocks and reset

The COMP input clock is synchronous with the APB1 CLK. Before using the comparator, the comparator clock should be enabled by setting the corresponding comparator clock enable bit in the RCC controller. Configuring the corresponding comparator reset control bit in the RCC controller allows to reset the comparator via software.

10.4.2 COMP switch

Before using the comparator, the COMP is powered-on by setting the EN bit in the COMPx_CSR register. When the EN bit is set to 1, the comparator (COMP) wakes up from the power-down state; when the EN bit is reset (EN set to 0), the comparator stops operating.

10.4.3 COMP inputs and outputs

The I/Os used as comparator inputs must be configured in analog input mode in the GPIO registers. The compared analog result could apply the filtering function (refer to the OFLT configuration in the COMPx_CSR register), or selectively connected to output end (refer to the OUT_ANA_SEL configuration in the COMPx_CSR register). The output can be internally redirected to a variety of timer input (refer to the OUT_SEL configuration in the COMPx_CSR register), or redirected to I/Os.

10.4.4 COMP channel selection

Each COMP has 4 positive input channels to be selected from four external pins, and has 5 negative input channels to be selected from three external pins and the internal pin connected the CRV output. The CRV voltage can be the partial voltage value of the internal 1.2V reference voltage (VREFINT) or the external voltage (VDDA).

In normal operation mode, the input channel of COMP is selected by software. In polling operation mode, comparison results of multiple channels are monitored in a time-shared manner via software polling. Logically, it is similar to simultaneous operation of several comparators.

In normal operation mode, the COMP compares selected signals from INP and INM ports. The specific procedures are as follows:

- Set the INP_SEL bit and the INM_SEL bit in the COMPx_CSR register to select desired signals;
- By setting the EN bit in the COMPx_CSR register, the COMP is powered-on and starts to work.
- The comparison results are stored in the OUT bit of the COMPx_CSR register.
- Moreover, it is required to set the CRV_SEL bit in the COMP_CRV register and then set the CRV_EN bit when the INM_SEL of COMP selects CRV (performed before the step 2 above).

In polling operation mode, the signals from the INP port in the COMP polls periodically. The FIXN bit in the COMPx_POLL register can be used to configure whether the signals from INM port follow the INP changes. The INM_SEL bit in the COMPx_CSR can also be used to configure them. Please note that the INP_SEL bit in the COMPx_CSR will be invalid when the polling function is enabled. Similarly, if the FIXN bit in the COMPx_POLL register determines the INM port follow the INP polling changes, then the INM_SEL bit in the COMPx_CSR will be invalid too. The specific procedures are as follows:

- Set the PERIOD bit in the COMPx_POLL register to select the desired polling wait cycles;
- Set the FIXN bit in the COMPx_POLL register to determine whether the signals from INM port follow the INP polling changes;
- Set the POLL_CH bit in the COMPx_POLL register to determine whether the desired polling channel is 1/2/3 or 1/2;
- Set the POLL_EN bit in the COMPx_POLL register to enable the polling function;
- By setting the EN bit in the COMPx_CSR register, the COMP is powered-on and starts to work.
- The polling comparison results are stored in the POUT bit of the COMPx_POLL register, where the results of polling channel 3/2/1 are individually stored in the POUT [2], POUT [1], and POUT [0].

10.4.5 Interrupt and wakeup

The comparator outputs can be connected to the events controller via hardware inside the chip. Each comparator has its own EXTI line and can generate events to exit the low-power mode. Refer to Interrupt and events section in the reference manual for more details.

10.4.6 Comparator LOCK mechanism

The comparators can be used for safety purposes, such as over-current or thermal protection. For some specific applications, it is necessary to ensure that the comparator programming cannot be altered in case of spurious modification access or program counter corruption.

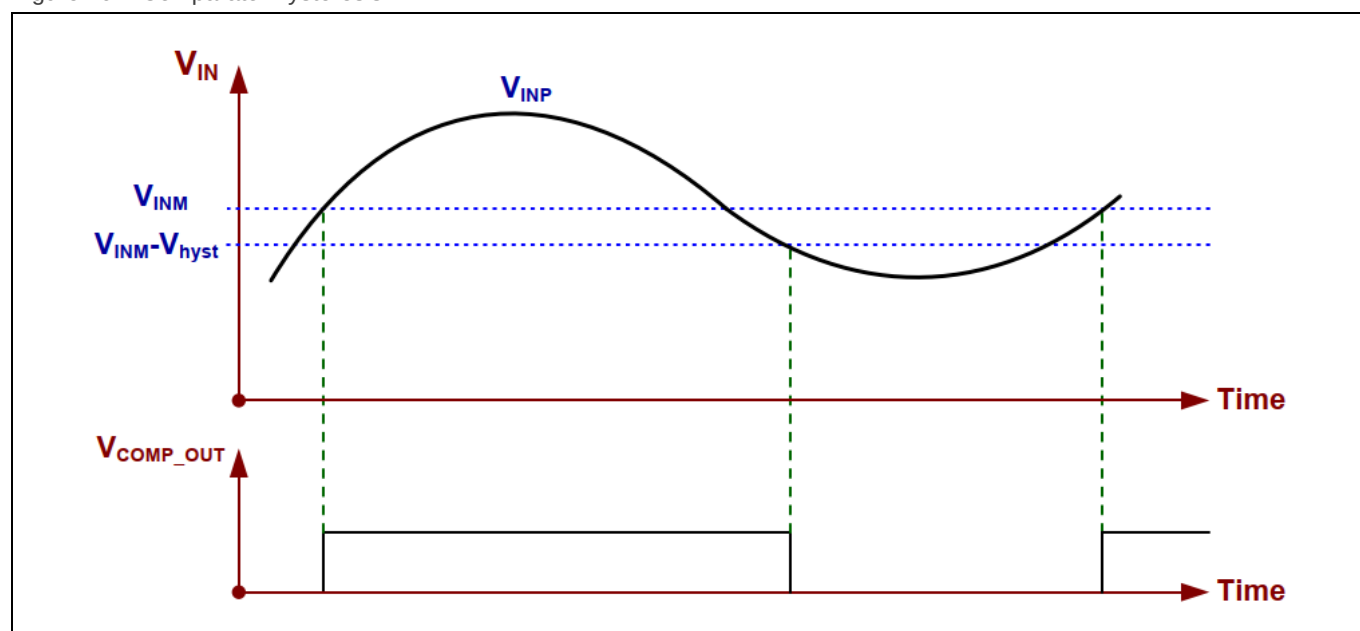
For this purpose, the comparator control status registers can be read-only.

Once the programming is completed, the LOCK bit is set to 1. This causes the whole COMPx_CSR register to become read-only, including the LOCK bit, which can only be reset by an MCU reset. (For details, please refer to the LOCK configuration in the COMPx_CSR register.)

10.4.7 Hysteresis voltage

To avoid invalid input in case of noisy signals, the comparator supports configurable hysteresis level (for details, please refer to the HYST configuration in the COMPx_CSR register).

Figure 10-2 Comparator hysteresis



10.5 Register

Table 10-1 Overview of COMP registers

Offset	Acronym	Register Name	Reset
0x0	COMPx_CSR (x=1)	COMP x (x=1) control status register	0x00000000
0x18	COMP_CRV	COMP comparator reference voltage register	0x00000000
0x1C	COMPx_POLL (x=1)	COMP x (x=1, 2) polling register	0x00000000

10.5.1 COMP control and status register (COMPx_CSR)(x=1)

Address offset: 0x0

Reset value: 0x00000000

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

LOCK	OUT	OUT_AN A_SEL										OFLT			HYST	
rw	r	rw										rw			rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
POL	Res.	OUT_SEL				Res.	INP_SEL		INM_SEL			Res.			EN	
rw		rw					rw		rw						rw	

Bit	Field	Description
31	LOCK	Comparator Register Lock This bit is write-once. It is set by software. It can be cleared by a system reset. Once set, it allows to have all control bits of comparator x as read-only. 1: COMPx_CSR is read-only 0: COMPx_CSR is read-write
30	OUT	Comparator x Output This bit indicates the comparator x output state. 1: Output is high (Positive input is greater than negative input in voltage) 0: Output is low (Positive input is less than negative input in voltage)
29	OUT_ANA_SEL	Comparator x output source selection 1: Select the analog output signal 0: Select the analog output signal that has been synchronized Note: When the COMP needs to generate an interrupt to wake up the CPU while in low-power mode, OUT_ANA_SEL can be set to 1.
28: 21	Reserved	Always read as 0
20: 18	OFLT	Comparator x Output Filter Period These bits control the comparator x output filter period. When the comparator output signal is shorter than the filter period width, it is considered as invalid to be filtered out, otherwise it is considered as valid. n indicates the number of Polling Wait Cycle configured with COMPx_POLL->PERIOD. 111: 128 * n clock cycles 110: 64 * n clock cycles 101: 32 * n clock cycles 100: 16 * n clock cycles 011: 8 * n clock cycles 010: 4 * n clock cycles 001: 2 * n clock cycles 000: no filter
17: 16	HYST	Comparator x Hysteresis These bits control the comparator x hysteresis level. 11: 85mV 10: 45mV 01: 22mV 00: 0mV
15	POL	Comparator x Output Polarity This bit is used to select the comparator x output polarity. 1: Output is inverted from the terminal 0: Output is not inverted from the terminal
14	Reserved	Always read as 0.
13: 10	OUT_SEL	Comparator x Output Selection These bits select the destination of the comparator x output. 0001: Timer 3 input capture 2 0010: Timer 1 break input 0110: Timer 1 Ocrefclear input 1010: Timer 3 input capture 1 1011: Timer 3 Ocrefclear input Other: No selection
9	Reserved	Always read as 0.
8: 7	INP_SEL	Comparator x Positive Input Selection These bits are used to select the source connected to the positive input end of the comparator x. 00: COMP1_INP0(PB3), corresponding to the Input Positive Channel 0; 01: COMP1_INP1(PB4), corresponding to the Input Positive Channel 1; 10: COMP1_INP2(PB5), corresponding to the Input Positive Channel 2; 11: COMP1_INP3(PA1), corresponding to the Input Positive Channel 3;

Bit	Field	Description
6: 4	INM_SEL	Comparator x Negative Input Selection These bits are used to select the source connected to the negative input end of the comparator x. 000 : COMP1_INM0 (PA3) , corresponding to the Input Negative Channel 0; 001 : COMP1_INM1 (PB2) , corresponding to the Input Negative Channel 1; 010 : COMP1_INM2 (PA4) , corresponding to the Input Negative Channel 2; 011 : COMP1_INM3 (PA12) , corresponding to the Input Negative Channel 3; 100 : COMP1_INM4 (CRV) , corresponding to the Input Negative Channel 4; Other: No selection
3: 1	Reserved	Always read as 0.
0	EN	Comparator x Enable This bit switches ON/OFF the comparator. 1: Comparator x enabled 0: Comparator x disabled

10.5.2 COMP comparator reference voltage register (COMP_CRV)

Address offset: 0x18

Reset value: 0x00000000

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
CRV_EN	Res.		CRV_SRC	Res.				CRV_SEL							
rw			rw					rw							

Bit	Field	Description
31: 16	Reserved	Always read as 0
15	CRV_EN	Comparator Reference Voltage Enable 1: COMP external reference voltage (CRV) enabled 0: COMP external reference voltage (CRV) disabled
14: 13	Reserved	Always read as 0
12	CRV_SRC	Comparator Reference Voltage Source Select 0: V _{REFINT} (internal voltage) 1: VDDA (external voltage) Note: If you want to select internal voltage as CRV source, the internal voltage sensor must be enabled beforehand by setting the VS_EN bit in the SYSCFG_SENSORCR register.
11: 8	Reserved	Always read as 0
7: 0	CRV_SEL	Comparator Reference Voltage Select $V_{CRV} = VDDA * (CRV_SEL / 256)$ or $V_{REFINT} * (CRV_SEL / 256)$

10.5.3 COMP polling register (COMPx_POLL) (x=1)

Address offset: 0x1C

Reset value: 0x00000000

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

Res.

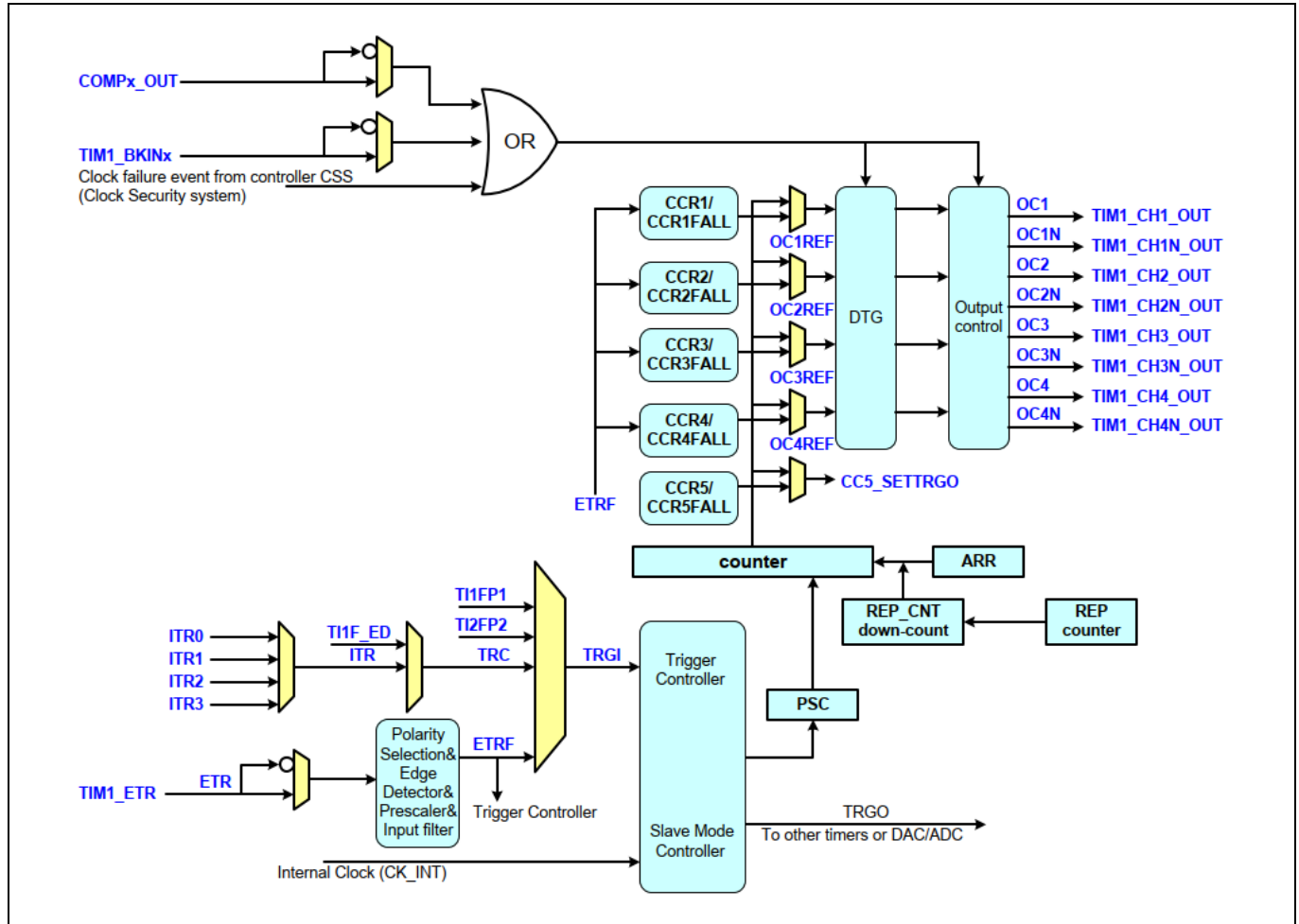
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.					POUT			Res.	PERIOD			Res.	FIXN	POLL_CH	POLL_EN
					r				rw				rw	rw	rw

Bit	Field	Description
31: 11	Reserved	Always read as 0
10: 8	POUT	Polling Output This bit is read-only. It reflects the polling channel output state. POUT[0] corresponds to channel 1. POUT[1] corresponds to channel 2. POUT[2] corresponds to channel 3. 1: Output is high (positive input above negative input) 0: Output is low (positive input below negative input)
7	Reserved	Always read as 0
6: 4	PERIOD	Polling Wait Cycle : Switch to the next polling channel every n PCLK1 cycles. PERIOD : n clock cycles 111: 128 clock cycles 110: 64 clock cycles 101: 32 clock cycles 100: 16 clock cycles 011: 8 clock cycles 010: 4 clock cycles 001: 2 clock cycles 000: 1 clock cycles
3	Reserved	Always read as 0
2	FIXN	Polling Negative Input Fix 1: Polling channel negative input is fixed. It is determined by the INM_SEL in the CSR register. When FIXN is 1, the INP_SEL is invalid. 0: Polling channel negative input is not fixed. It changes simultaneously with the INP channel. When FIXN is 0, the INM_SEL and INP_SEL is invalid.
1	POLL_CH	Comparator Polling Channel 1: polling channel 1/2/3 0: polling channel 1/2
0	POLL_EN	Comparator Polling Enable 1: Comparator polling enabled. 0: Comparator polling disabled.

11 TIM1 Advanced-Control Timer Introduction

TIM1 is composed of a 16 bit real-time programmable prescaler and a 16-bit auto reload counter with the programmable counting direction, and can conveniently provide with the counting and timing function. The counter is driven by a programmable prescaler. The advanced timer can be used for a variety of purposes, offering features such as PWM output, complementary outputs with programmable dead-time, and single-pulse mode output.

Figure 11-1 Advanced-control timer block diagram



11.2 Main features

TIM1 timer features include:

- 16 bit real-time programmable prescaler, division factor: 1-65536.
- Clock source: Internal clock source, internal trigger input (ITRx).
- 16 bit auto reload counter (counting direction: up, down, up/down).
- 8 bit programmable repeat counting function, repeat counter can be auto reloaded (auto reload at update event)
- Synchronization circuit to control the timer with external signals and to interconnect several timers to each other.
- Trigger input can serve as external clock or be managed cycle-by-cycle.
- 5 comparison channels (channels 1–4 support output, while channel 5 only provides interrupt and internal trigger signals)
- 4 output channels with complementary output functionality
- Output compare (control output waveform or indicator counter already finishes timing).
- PWM output (dead-time programmable) (edge alignment or central alignment mode)
- Break input will put the timer output signal in a safe state (reset state or preset state, choice by user).
- One-pulse mode.
- Interrupt request event: update event, trigger event, input capture, output compare or break input.
- The output values for channels 1–4 can be set via software.

11.3 Functional description

11.3.1 Clock

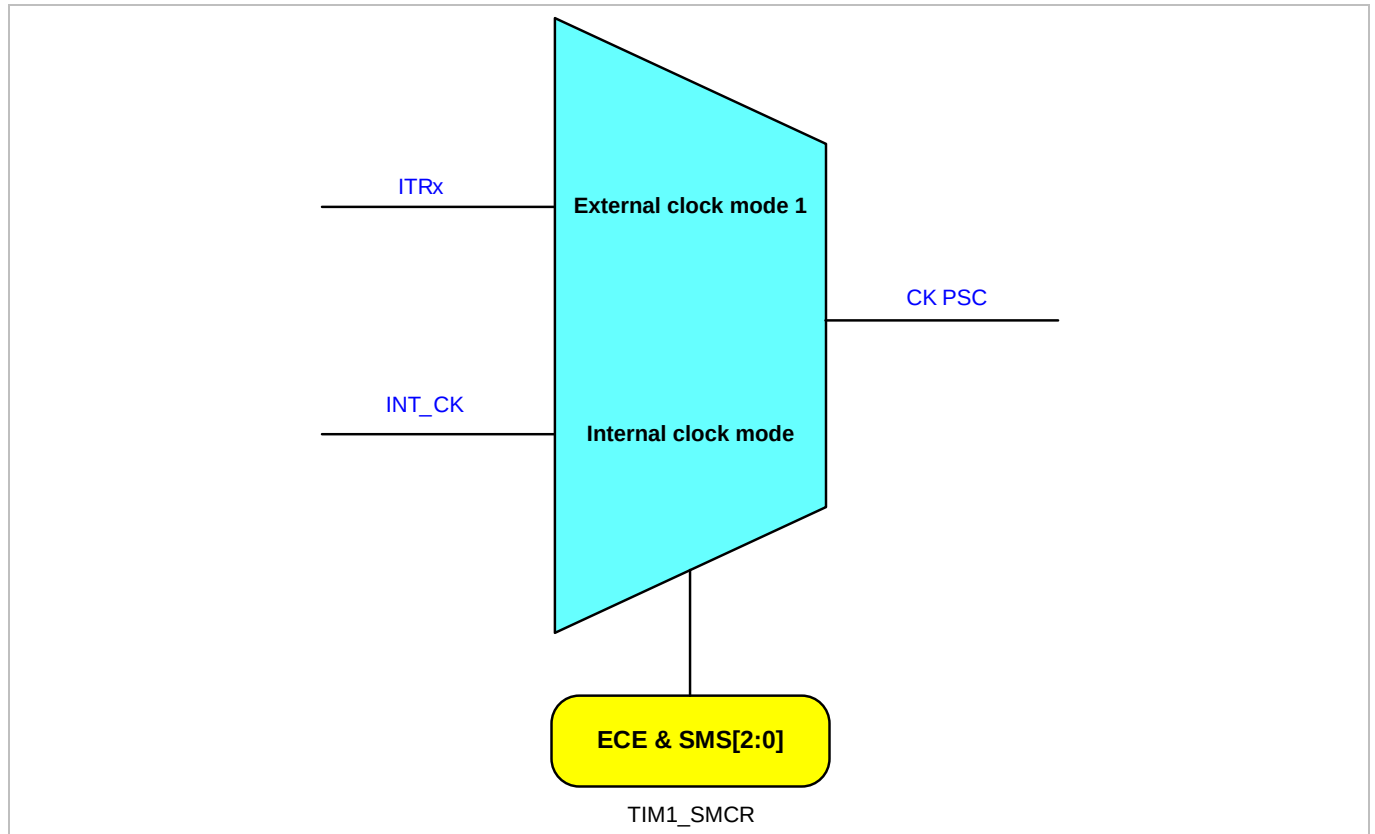
11.3.1.1 Clock selection

Several types of clock sources of the counter below:

- Internal clock (INT_CK)
- External clock mode 1: External trigger input TRGI (including ITRx)

The diagram of several types of clock choice above:

Figure 11-2 Clock selection



11.3.1.1.1 Internal clock source (INT_CK)

In case of SMS =000 of TIM1_SMCR register, disable the slave mode. Enable the counter, and the prescaler clock is driven by internal clock. In this case, the counter clock is the clock derived from the internal clock via frequency division.

11.3.1.1.2 External clock mode 1(external trigger input TRGI, including ITRx)

In case of SMS = 111 of TIM1_SMCR register, select the external clock mode 1 (TRGI). The counter is driven by the rising edge or falling edge of the selected input signal.

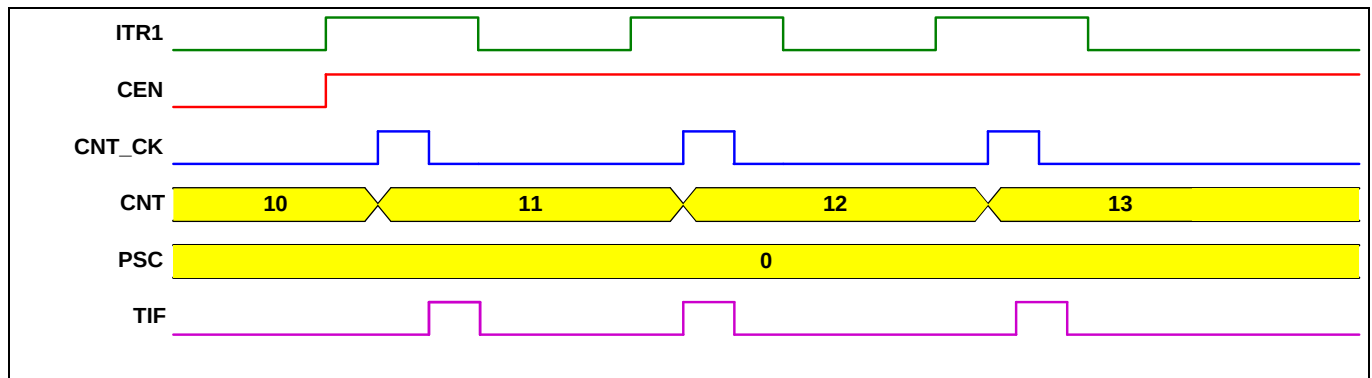
For example: count up at the rising edge on TI1 input. Specific configuration:

1. Configure TIM1_SMCR register TS=001, select ITR1 as the trigger input source; configure TIM1_SMCR register SMS=111, select the external clock mode 1.

2. Configure TIM1_CR1 register DIR=0, select up counting mode; configure TIM1_CR1 register CEN=1 and start the counter.

ITR1 input the active edge, count up once and the TIF flag is set to 1 by hardware. The delay between the active edge of ITR1 and actual clock of counter depends on the ITR1 input pin sync circuit.

Figure 11-3 Control circuit in external clock mode 1



11.3.1.2 Time base unit

TIM1 time base unit mainly includes: Counter register (TIM1_CNT), prescaler register (TIM1_PSC), auto reload register (TIM1_ARR) and repetition counter register (TIM1_RCR).

The counter unit is composed of a 16 bit counter with its related auto reload register, the counter can count up, down or both up and down.

The counter clock is provided by prescaler. The prescaler is composed of the prescaler counter with its related register. The division factor is 1-65536. It can write any time, and gets active at the next update.

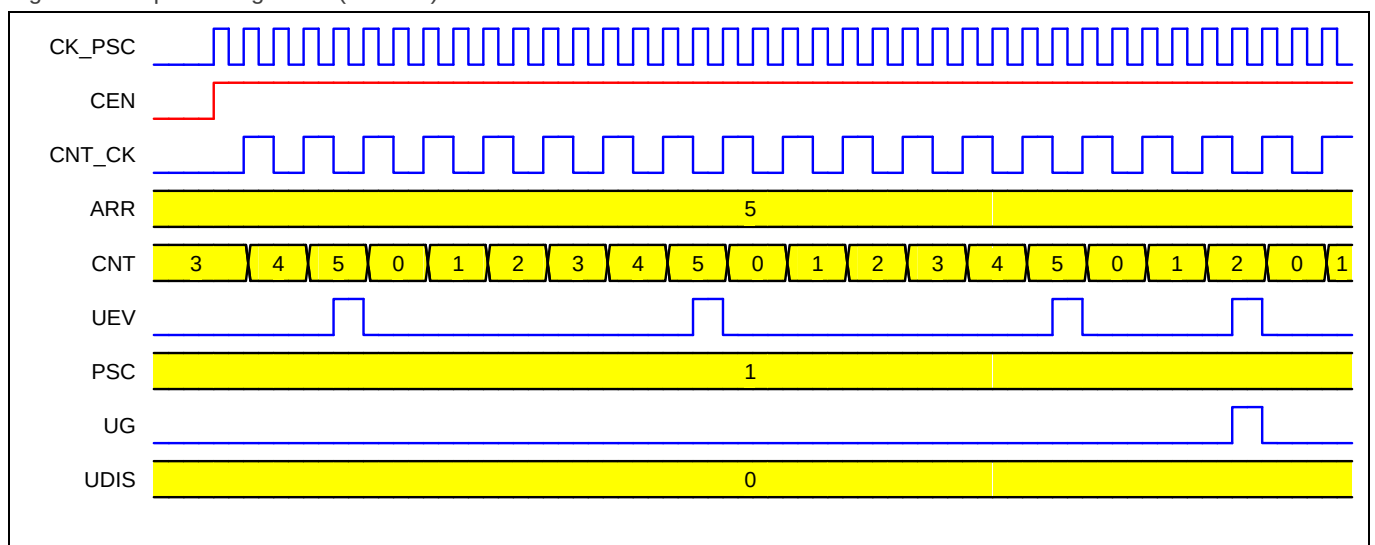
The auto preload register has a 16 bit shadow register with preload function. Set TIM1_CR1 register ARPE bit to select the content of ARR register are transferred into the shadow register permanently or at each update event.

Figure 11-4 Auto preload

Configure DIR bit and CMS bit of TIM1_CR1 register to select the count mode. There are three count modes, up counting mode, down counting mode and central alignment mode (up/ down counting mode). Detailed descriptions are provided for each count mode.

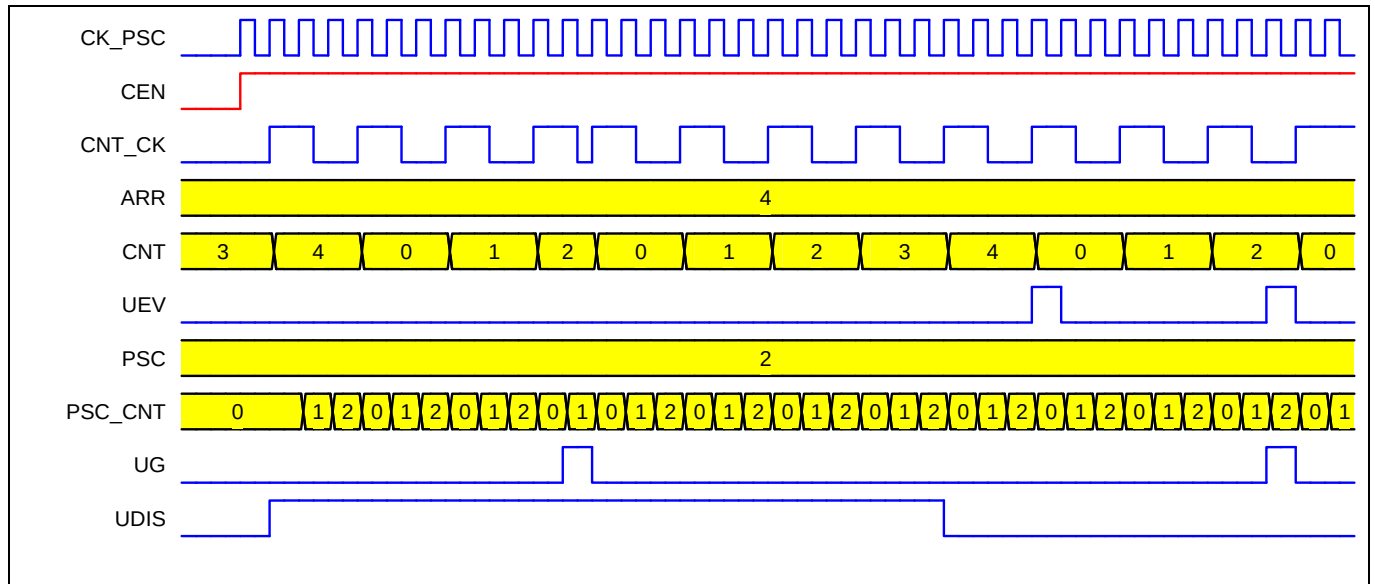
Configure TIM1_CR1 register CMS=0 , DIR=0, select up counting mode.

Figure 11-5 Up counting mode (UDIS=0)



megawin

Figure 11-6 Up counting mode (UDIS =1, disable update event)



Note: In case of update event:

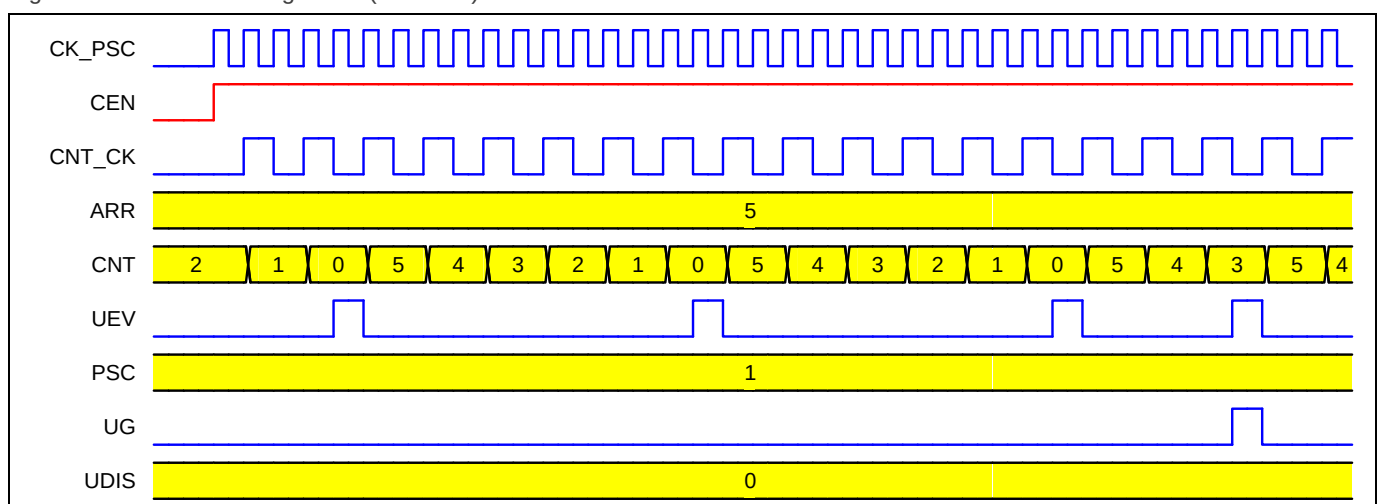
- Repeat counter is loaded with RCR register value, and down count again.
- ARR register value is loaded into ARR shadow register.
- Prescaler preload value becomes active.

11.3.1.3.2 Down counting mode

Configure TIM1_CR1 register CMS=0, DIR = 1, select down counting mode.

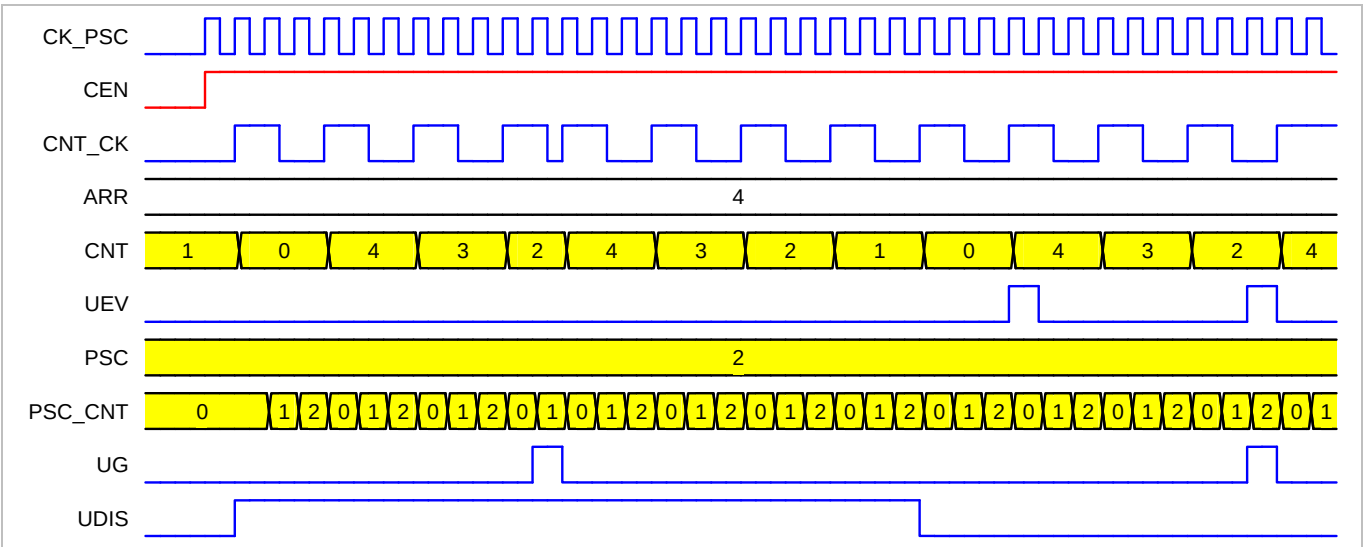
In down counting mode, the counter begins down count from the auto preload value TIM1_ARR. When counting to 0, generate an underflow event (update event). When the user enables the repeat count function, the repeat counter conducts down count at each underflow event. When the repeat counter down to 0 from the set value, generate the update event. Set TIM1_EGR register UG =1, also generate an update event. After the update event, begin down count (TIM1_CR1 register UDIS=0) from the auto load value TIM1_ARR.

Figure 11-7 Down counting mode (UDIS=0)



Configuring TIM1_CR1 register UDIS=1, disable update event. When the counter generates the underflow event, it doesn't generate update event. Configuring UG=1 also doesn't generate update event. However, the counter and prescaler counter will initialize, and count begins from TIM1_ARR.

Figure 11-8 Down counting mode (UDIS=1 disables update event)



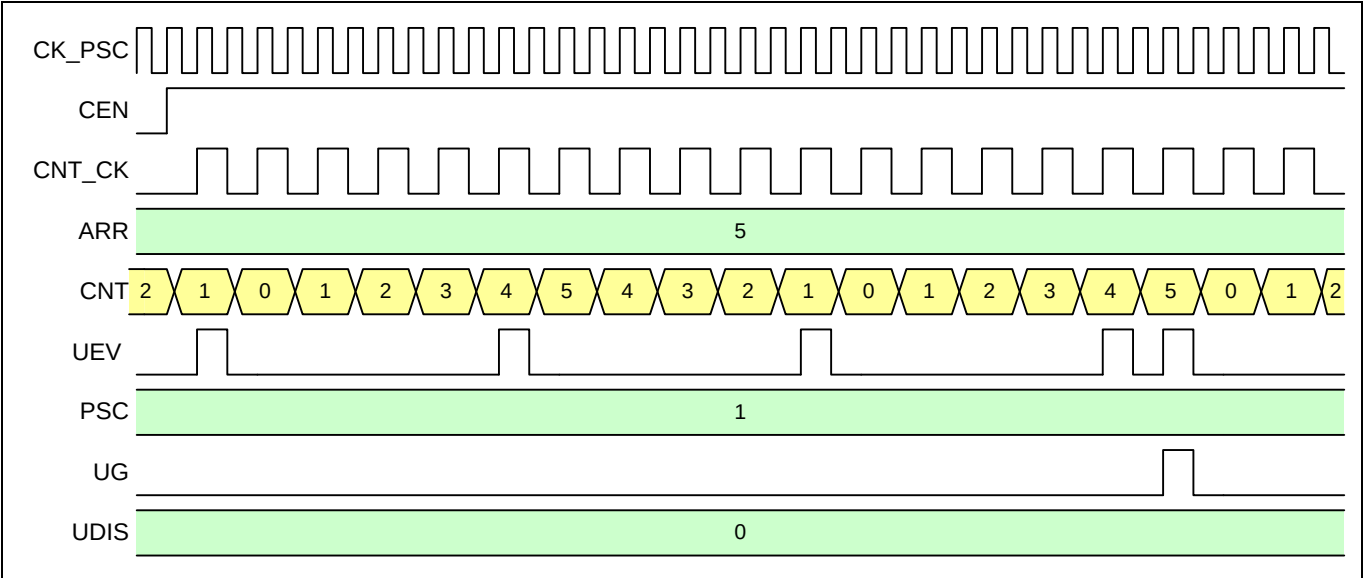
11.3.1.3.3 Central count mode (up/down count mode)

Configure TIM1_CR1 register CMS $\neq$ 0 (write DIR failure), and select central alignment count mode.

Central alignment count mode, up count and down count run alternatively. In case of up count to ARR-1, generate an overflow event. Begin down count to 1 from ARR, generate an underflow event, and begin up count from 0.

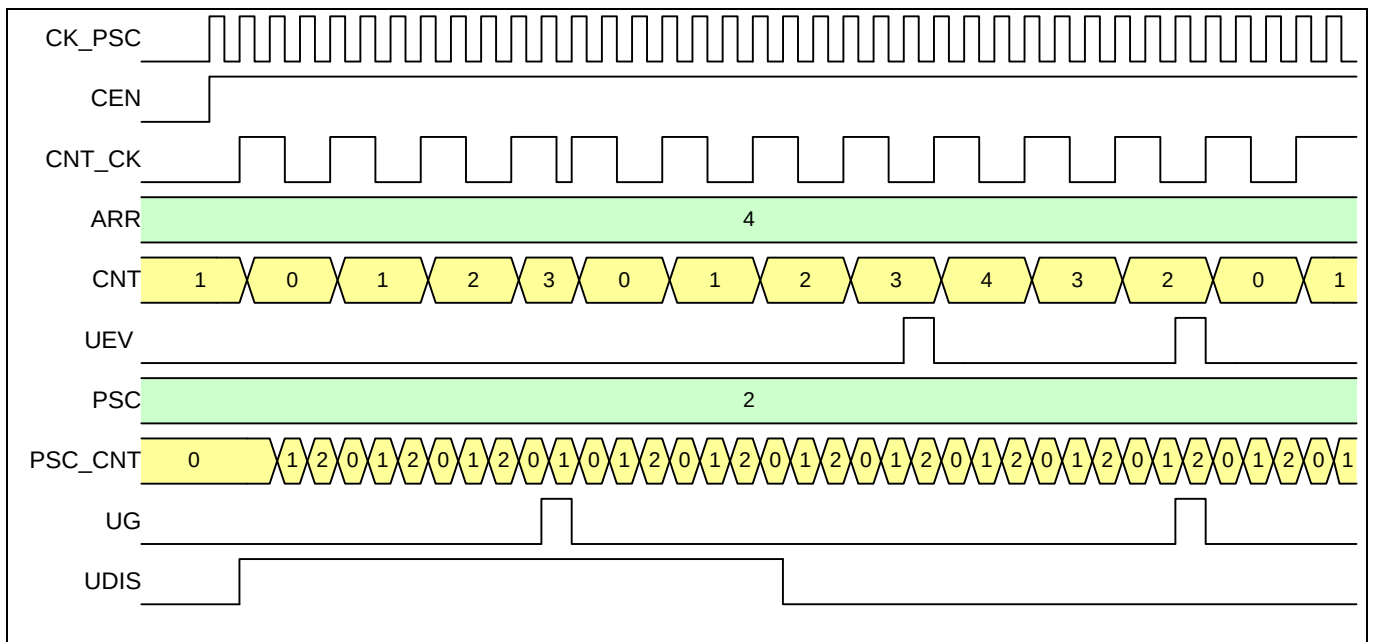
When the user enables repeat count function, the repeat counter count down once at each overflow or underflow event. When the repeat counter down to 0 from the set value, generate an update event; setting TIM1_EGR register UG =1 also generate an update event. After update event, the counter begins up count again from 0 (TIM1_CR1 register UDIS=0).

Figure 11-9 Central count mode (UDIS=0)



When configuring TIM1_CR1 register UDIS=1, disable update event. When the counter generates the overflow or underflow event, it doesn't generate update event. Configuring UG=1 also doesn't generate update event. However, the counter and prescaler counter will initialize, and the count begins from 0.

Figure 11-10 Central count mode (UDIS=1 disables update event)



11.3.2 Repeat counter

The repeat counter may adjust the frequency of the update event. During up count in the edge alignment mode, the repeat counter counting down at each overflow of the counter; during down count, the repeat counter counting down at each underflow of the counter. In the central alignment mode, the repeat counter counting down at the counter overflow and underflow. Configuring REP of the TIM1_RCR register to adjust the frequency of the update event. The repeat counter generates the update event after REP+1 count cycle. In the central alignment mode, the writing REP value decides whether the update event generates at overflow or underflow.

In case of the update event, REP value will be updated to the real-time repeat counter REP_CNT. To enable the real-time writing of REP_CNT achieves the flexible adjustment of the occurrence time of the update event.

Figure 11-11 Repeat count time sequence in the central alignment mode

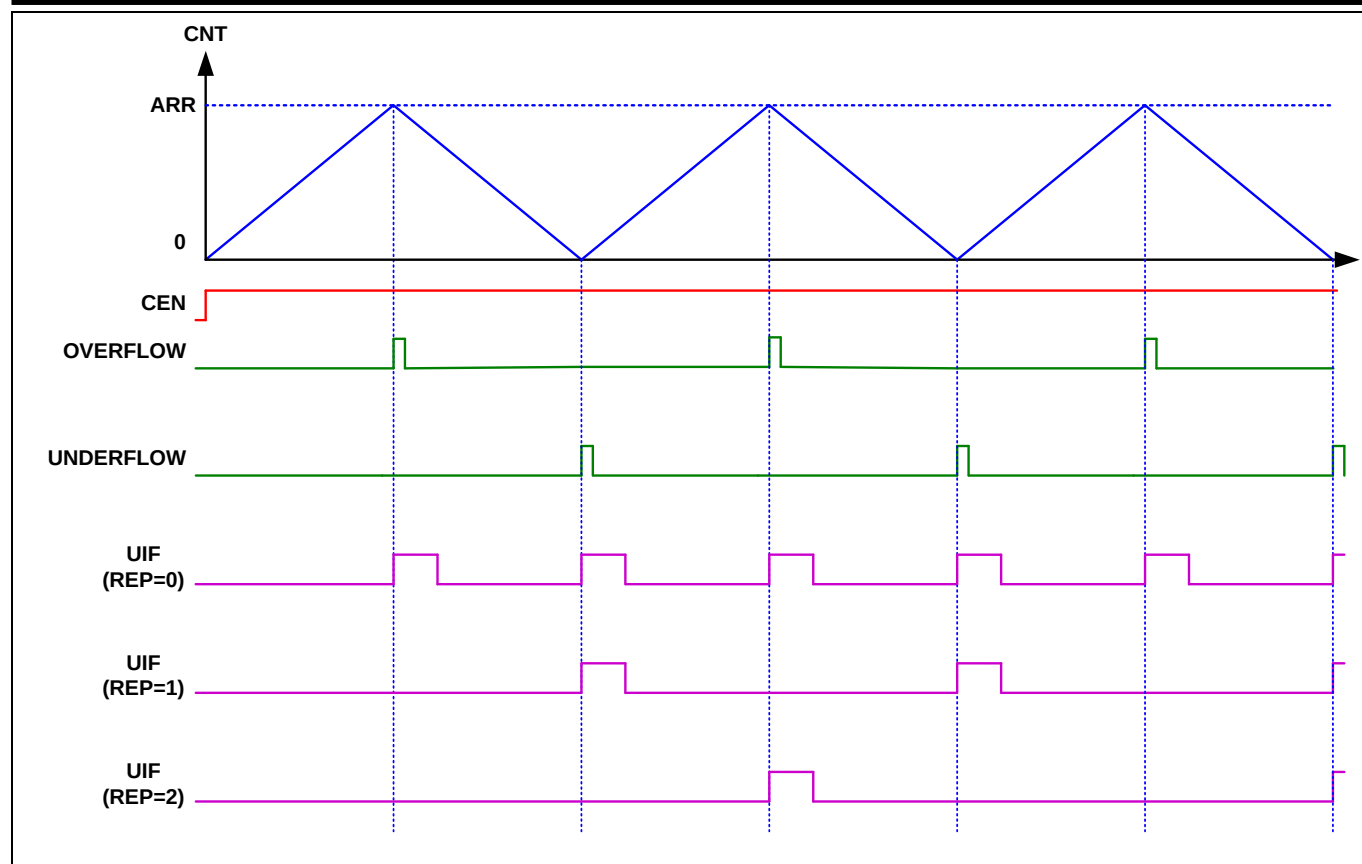


Figure 11-12 Up count time sequence in the edge alignment mode

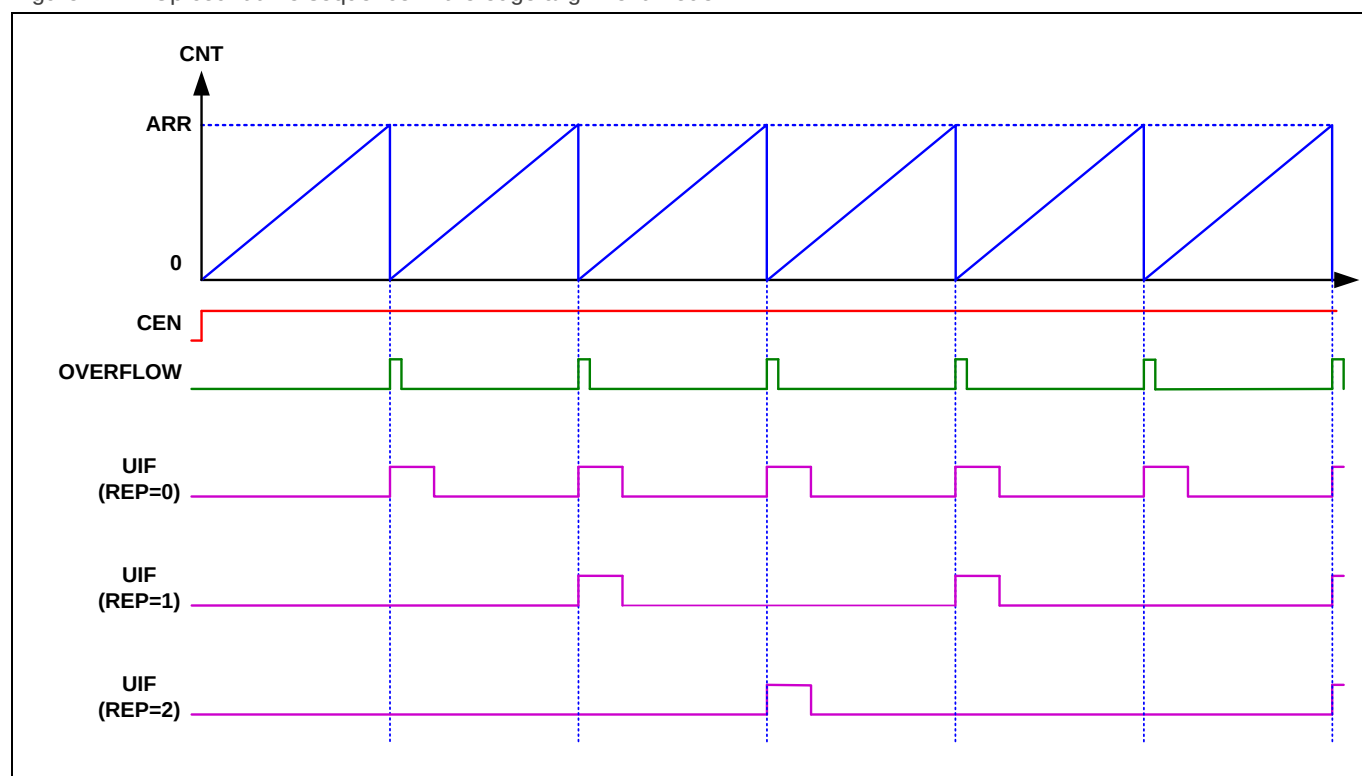
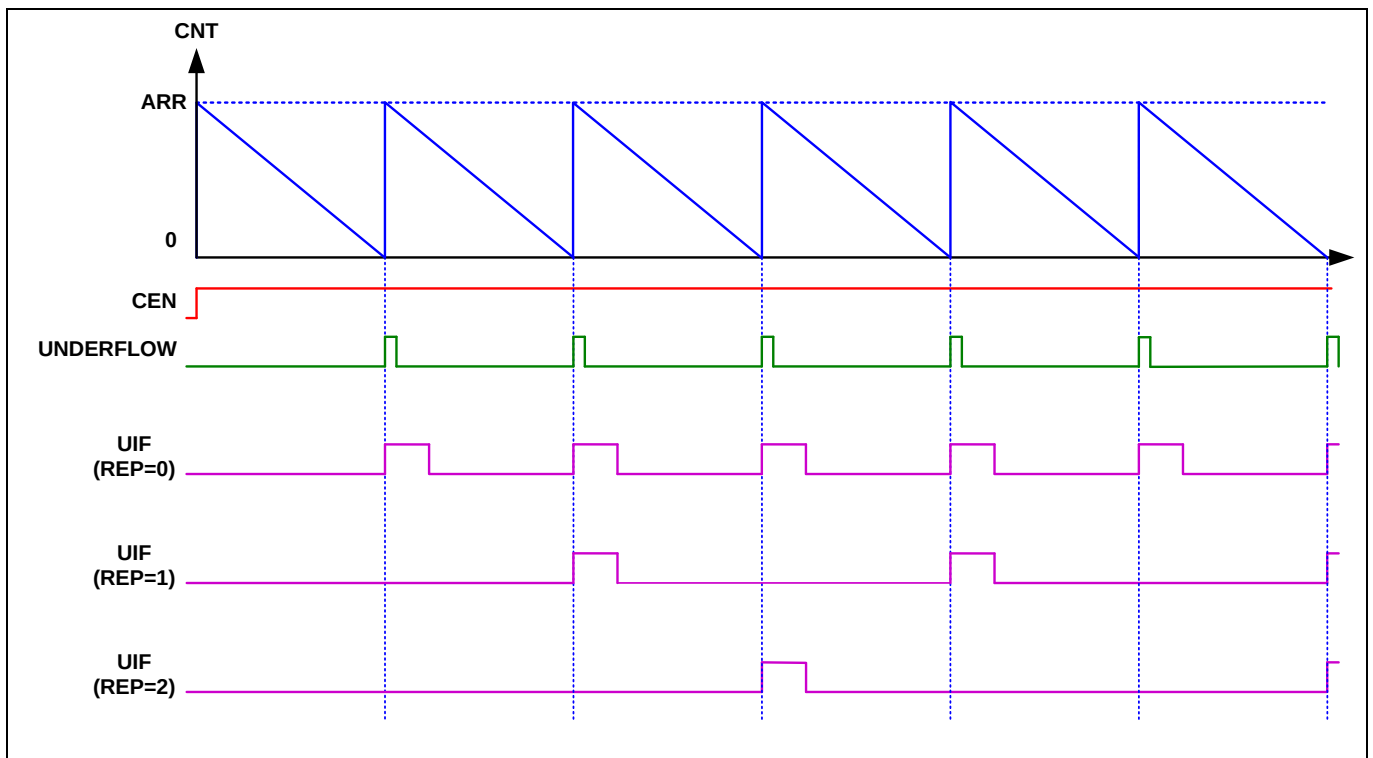


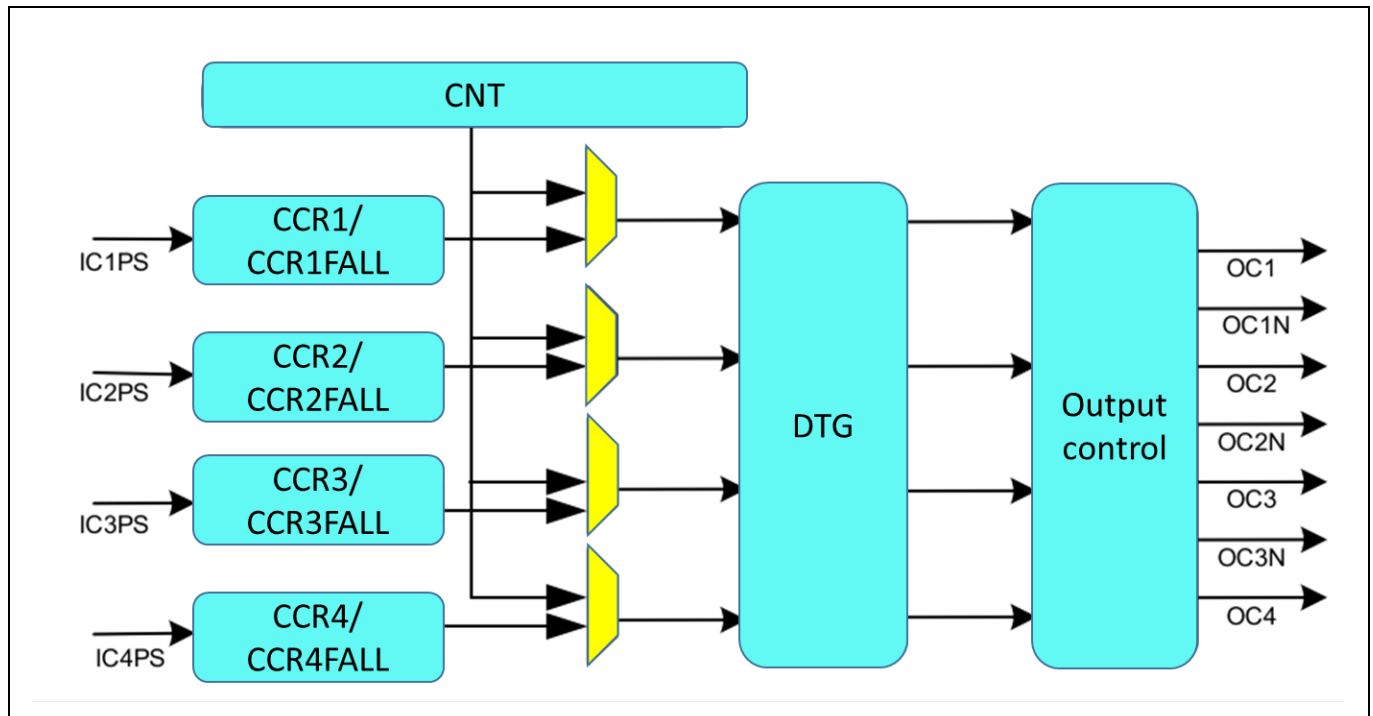
Figure 11-13 Down count time sequence in the edge alignment mode



11.3.3 Output compare

The compare output of the capture/compare channel is composed of the comparator, output control circuit and capture/compare register. Its structure is shown below:

Figure 11-14 Output compare block diagram



In output compare mode, the content of compare register is loaded into the shadow register, and then compare the content of shadow register with the current value of counter. The compare module includes a compare register (preload register) and a shadow register. When read and write, only programmed the compare register.

11.3.3.1 Force output

Configure TIM1_CCMRx register CCxS = 00 and set the channel CCx as output mode. Configure TIM1_CCMRx register OCxM bit, and directly enforce the output compare signal as active or inactive level, independently of the output compare result. Configure TIM1_CCMRx register OCxM = 100, and enforce output compare signal as inactive level. And OCxREF is enforced as low level. Configure TIM1_CCMRx register OCxM = 101, and enforce output compare signal as active level. And the OCxREF is enforced as high level (OCxREF always active as high level).

Note: In force output mode, TIM1_CCRx shadow register and counter output compare is still performed. The corresponding flag bit of compare result will also be changed. When enable the corresponding interrupt, it will also generate the corresponding interrupt.

11.3.3.2 Output compare

In the output compare mode, when the values of counter and capture compare register matches, different waveform can be exported according to the OCxM bit of TIM1_CCMRx register.

For example, when the value of counter and capture/compare register matches, the functions in the output compare mode:

1. When compare match, OCxM value is different, output channel x signal OCx run differently:

- ◆ OCxM = 000: OCx signal keeps its level.
- ◆ OCxM = 001: OCx signal is set as active level.
- ◆ OCxM = 010: OCx signal is set as inactive level.

- ◆ OCxM = 011: OCx signal toggle.

2. When match, set the flag bit of status register (CCxIF bit of TIM1_SR register).

3. Configure TIM1_DIER register CCxIE =1, generate an interrupt when match.

The output compare mode can also be used to export a single pulse (one-pulse mode).

Configuration steps of channel 1 output compare mode:

1. Configure counter clock (select clock source, configure prescaler factor).

2. Configure TIM1_ARR and TIM1_CCR1 register.

3. Configure TIM1_DIER register CC1IE =1, enable compare 1 interrupt.

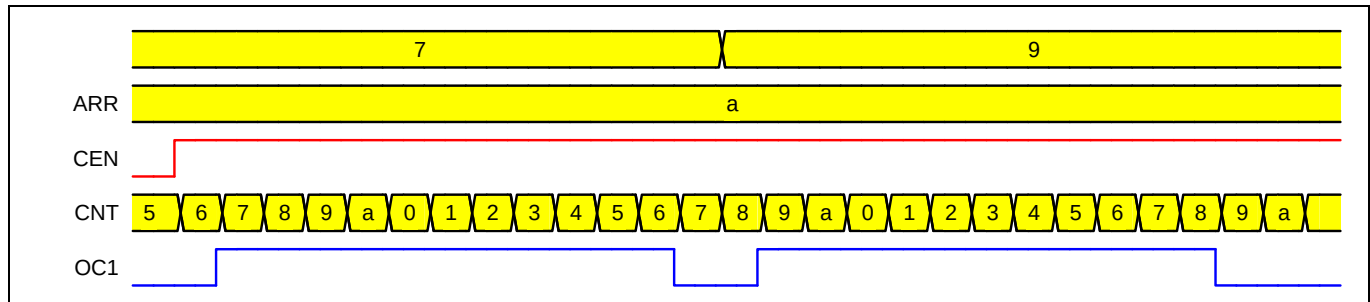
4. Configure output mode:

- ◆ Configure TIM1_CCMR1 register OC1M = 011, toggle when OC1 compare match.
- ◆ Configure TIM1_CCMR1 register OC1PE = 0, disable preload function of TIM1_CCR1 register.
- ◆ Configure TIM1_CCMR1 register CC1P = 1, OC1 is active low polarity.
- ◆ Configure TIM1_CCER register CC1E = 1, enable channel 1 output, OC1 signal is exported to the corresponding output pin.

5. Configure TIM1_CR1 register CEN =1, enable the counter.

Configure TIM1_CCMRx register OCxPE=0, disable TIM1_CCRx register preload function, write TIM1_CCRx register any time, and the write value updated immediately. Configure TIM1_CCMRx register OCxPE=1, enable TIM1_CCRx register preload function, Updated only at the next update event. The figure below is an example.

Figure 11-15 Output compare mode, OC1 signals toggle when match



Note: In the output compare mode, the update event has no effect on output result. In the force output mode, the compare output between TIM1_CCRx shadow register and counter still performed. The corresponding flag bit of compare result is also be changed. When enabling the corresponding interrupt, it still generates the corresponding interrupt.

11.3.3.3 PWM output

In the PWM mode, according to the value of TIM1_ARR register and TIM1_CCRx register, generate a frequency and duty cycle controllable PWM waveform.

Configure OCxM=110 or OCxM=111 of TIM1_CCMRx register corresponding to the channel x, select channel x to PWM mode 1 or PWM mode 2. In the PWM mode, the counter will always compare with CCRx. According to configuration and compare results, the channel x exports different signals. Therefore, TIM1 can generate 4 PWM output signals with independent duty under the same frequency. In PWM mode, enable the preload function of TIM1_CCRx and TIM1_ARR. The written value of TIM1_CCRx preload register and TIM1_ARR preload register will become effective at the next update event, and load the corresponding shadow register. Therefore, in the PWM mode, before enabling counter, it's required to set TIM1_EGR UG=1, and generate the update event to initialize all registers.

Configure TIM1_CCER register CCxP, select OCx's active polarity. Configure TIM1_CCER register CCxE, CCxNE bit and TIM1_BDTR register MOE, OSS1, OSSR bit and control OCx output enable. Configure TIM1_CR1 register CMS bit, and select edge alignment or central alignment PWM signal.

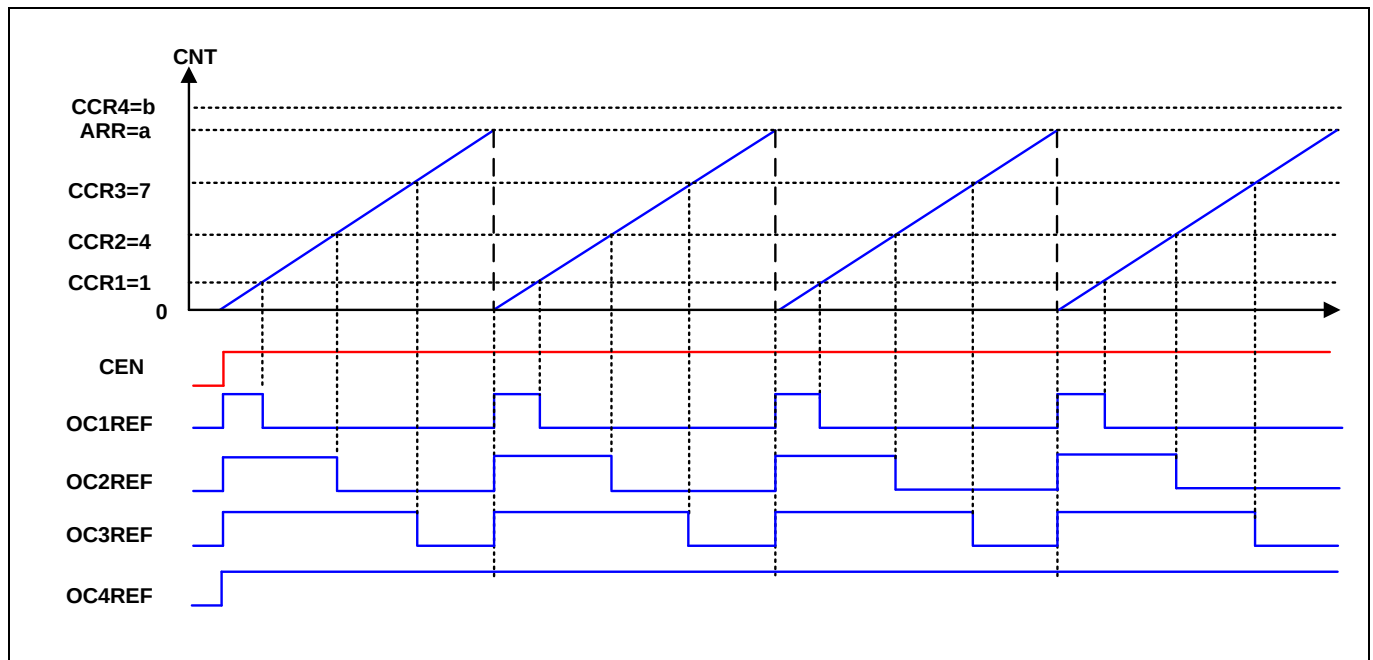
- CMS=00, edge alignment mode, configure DIR, select count mode (up or down).
- CMS=01, central alignment mode 1.
- CMS=10, central alignment mode 2.

- CMS=11, central alignment mode 3.

11.3.3.3.1 PWM edge alignment mode - up counting mode

Based on up counting mode configuration, configure TIM1_CCMRx register CCxS=00, select output mode, OCxM=110, select PWM mode 1. In case of $TIM1_CNT < TIM1_CCRx$, channel x (OCxREF) is in active level. Otherwise, it's inactive level. If $TIM1_CCRx$ compare value is greater than the auto reload value ($TIM1_ARR$), OCxREF stays in active level. When the compare value is 0, OCxREF is at inactive level. The figure below is the example of waveform in PWM mode 1 during edge alignment up count in case of CCR1=1, CCR2=4, CCR3=7, CCR4=b and ARR=a.

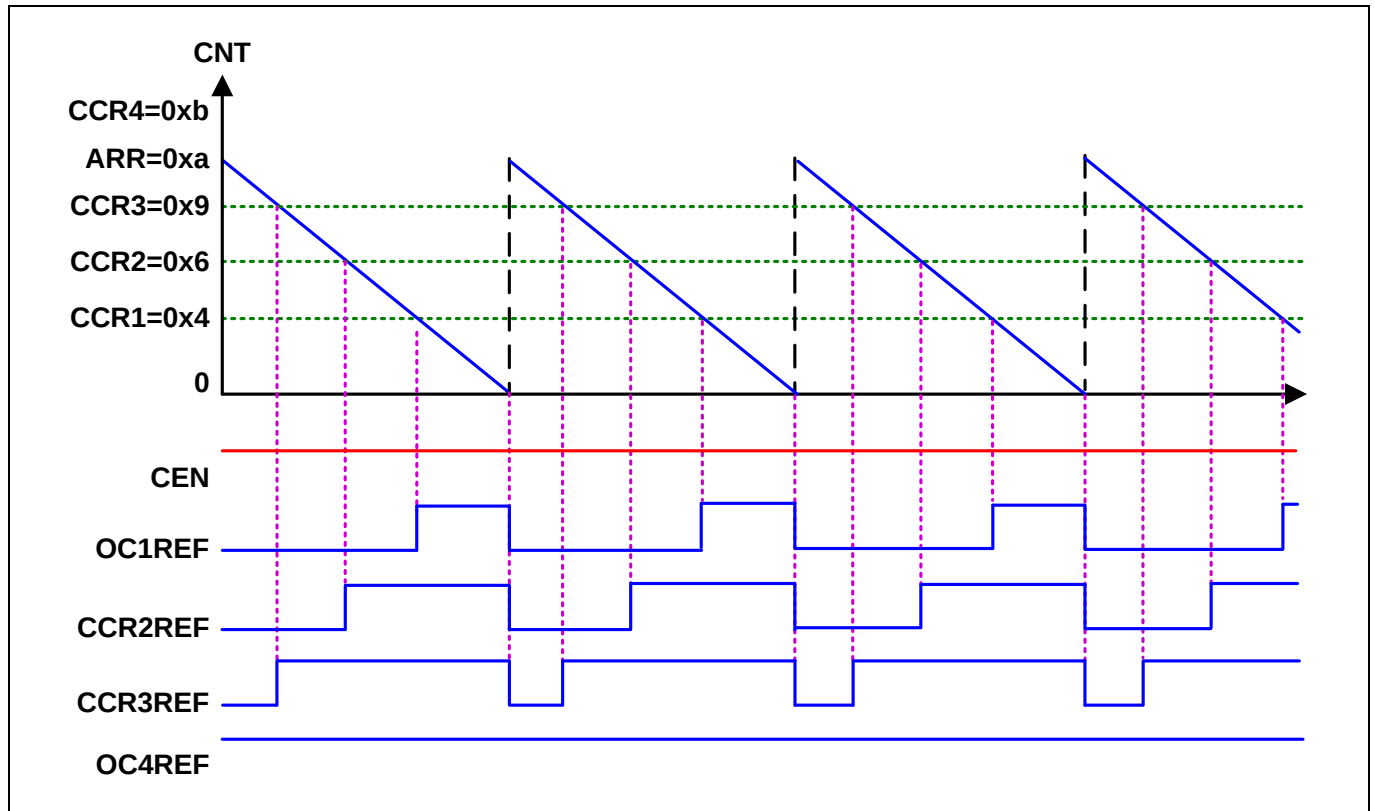
Figure 11-16 Waveform in PWM mode 1 during edge alignment up count



11.3.3.3.2 PWM edge alignment mode -down counting mode

Based on the down counting mode configuration, configure TIM1_CCMRx register CCxS=00, select output mode, OCxM=110, select PWM mode 1. In case of $TIM1_CNT > TIM1_CCRx$, channel x (OCxREF) is at inactive level. Or else, it's at active level. The figure is an example of waveform in PWM mode 1 during edge alignment down count in case of CCR1=4, CCR2=6, CCR3=9, CCR4=b and ARR=a.

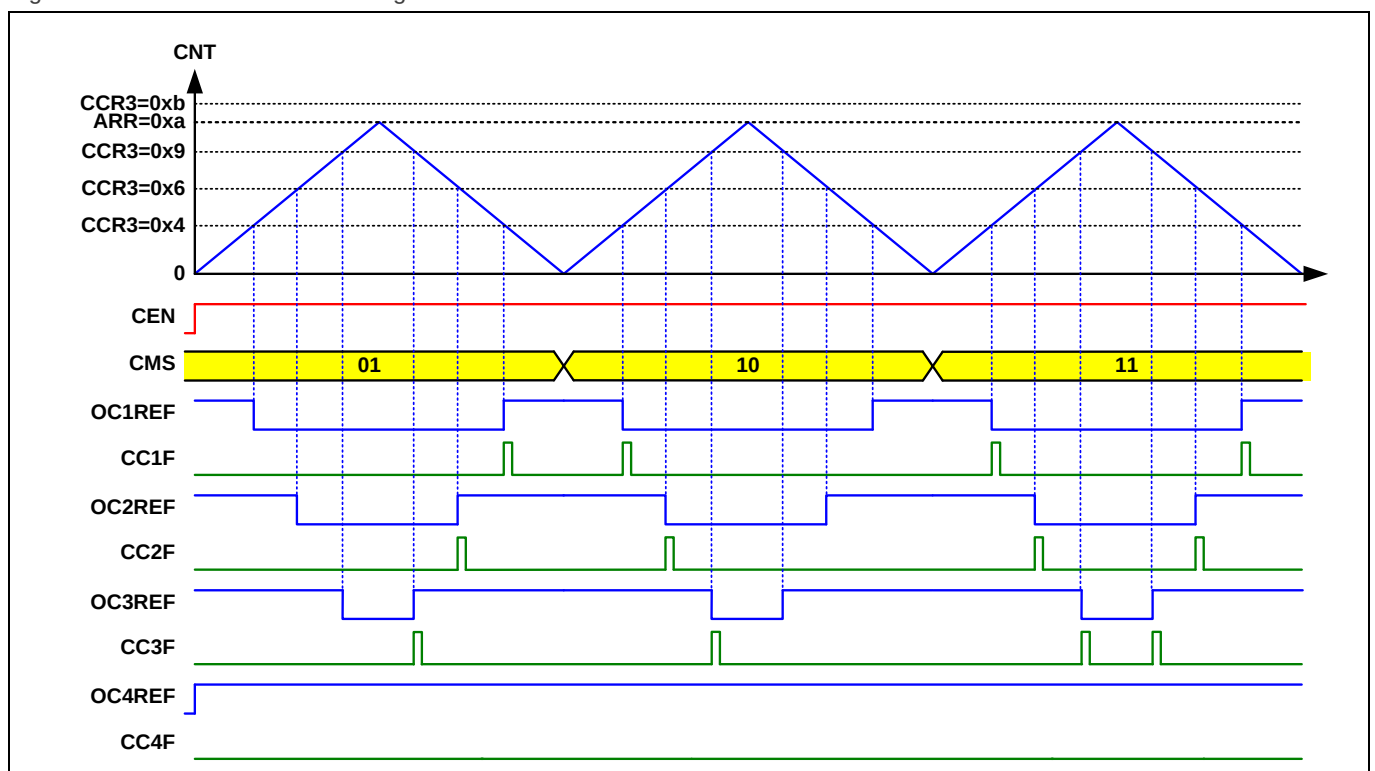
Figure 11-17 Waveform in PWM mode 1 during edge alignment down count



11.3.3.3 PWM central alignment mode

Configure TIM1 counter as central alignment count mode. Configure TIM1_CCMRx register CCxS=00 and select output mode. According to the CMS with different configurations, the output compare interrupt flag bit is set during up count (CMS =01), set during down count (CMS =10) or set during up or down count (CMS =11). The figure below is an example of waveform in central alignment PWM mode 1 (CMS =1) during CCR1=4, CCR2=6, CCR3=9, CCR4=b, ARR=a.

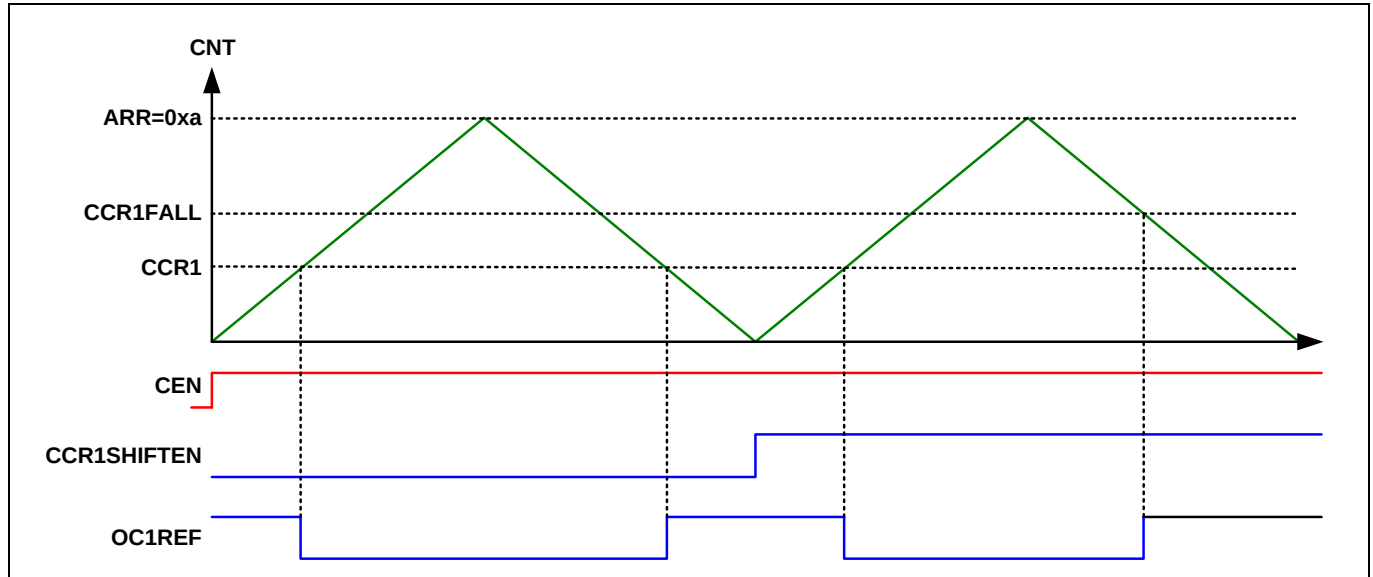
Figure 11-18 Waveform in central alignment PWM mode 1



11.3.3.3.4 Phase move function under PWM central alignment mode

Configure PDER (channel x output PWM phase shift enable bit) and CCRxFALL register (channel x capture/compare value in PWM central alignment down count mode), set the channel output PWM phase shift function. Configure CCRxFALL and CCRx, PWM output programmable phase shift waveform, and it can shift left or right.

Figure 11-19 Move phase function

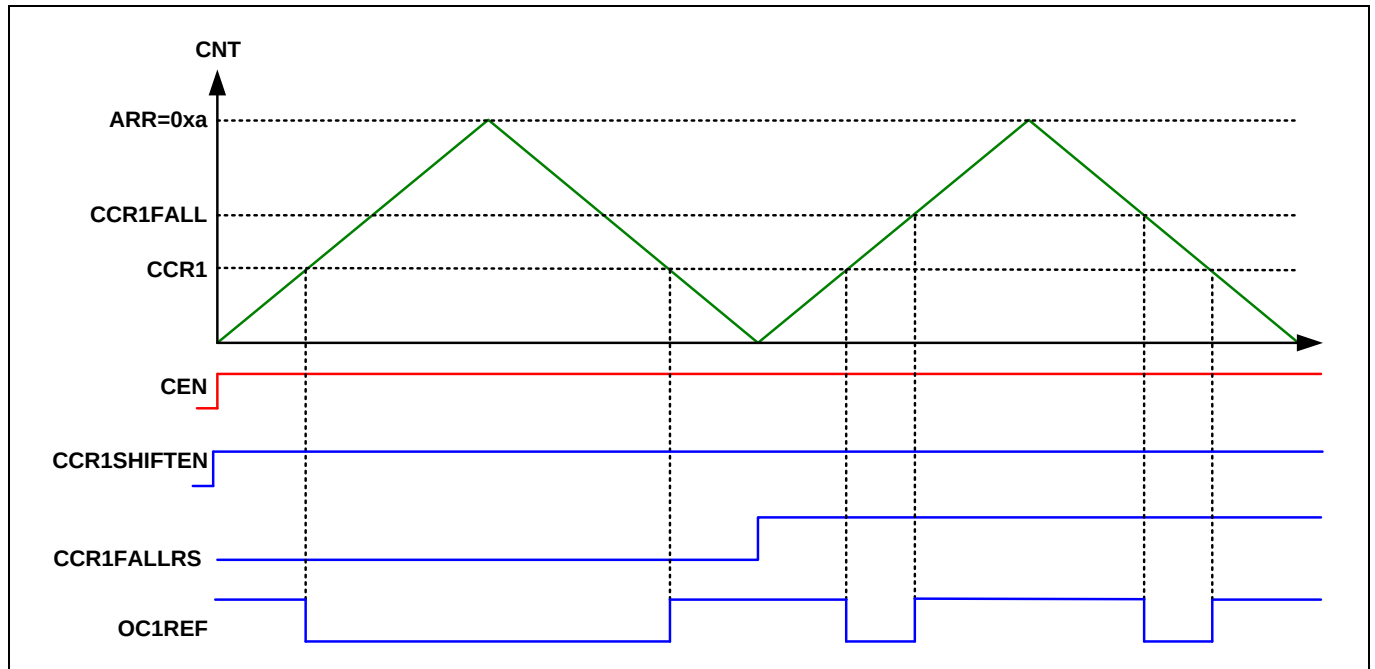


Note:

- In central alignment mode, use the current up/down count configuration, and the count direction depends on the current DIR value.
- In the central alignment mode, it's better not to change the counter value or it may generate unexpected result. When the counter is in up counting, write the counter value $> \text{TIMx_ARR}$, and the counter will continue up count. Writing 0 or ARR will immediately update the count direction. However, it won't generate the update event.
- In the central alignment mode, before enabling counter, configure TIMx_EGR register UG =1, generate a software update. Update all registers, don't change the counter value after enabling the counter.

In phase-shift mode, the CCRxFALLRS bit of the PDER register can be configured to be active, this enables CCRxFALL to trigger a toggle of OCxREF during the up-counting phase.

Figure 11-20 Triggering of CCRxFALL incrementing count in phase-shift mode



- As shown in Figure 11-20, in phase-shift mode, when CCRxFALLRS is active, OCxREF can be triggered during both up-counting and down-counting.
- Perform a toggle (i.e., when the CNT value is between CCRx and CCRxFALL, OCxREF can output a low or high level depending on the mode).
- Note that unlike CCRxFALL, which triggers only during the down-counting phase, when CCRxFALLRS is enabled, you must set the CCRxFALL value to be greater than the CCRx value; otherwise, OCxREF will not toggle and will instead maintain its previous level.
- By configuring and adjusting the values of the four sets of CCRx and CCRxFALL registers (and updating these values during commutation update events), this feature enables the generation of four complementary outputs, thereby supporting waveforms with a 180° phase relationship.
- As shown in Figures 11-21 and 11-22 below:

Figure 11-21

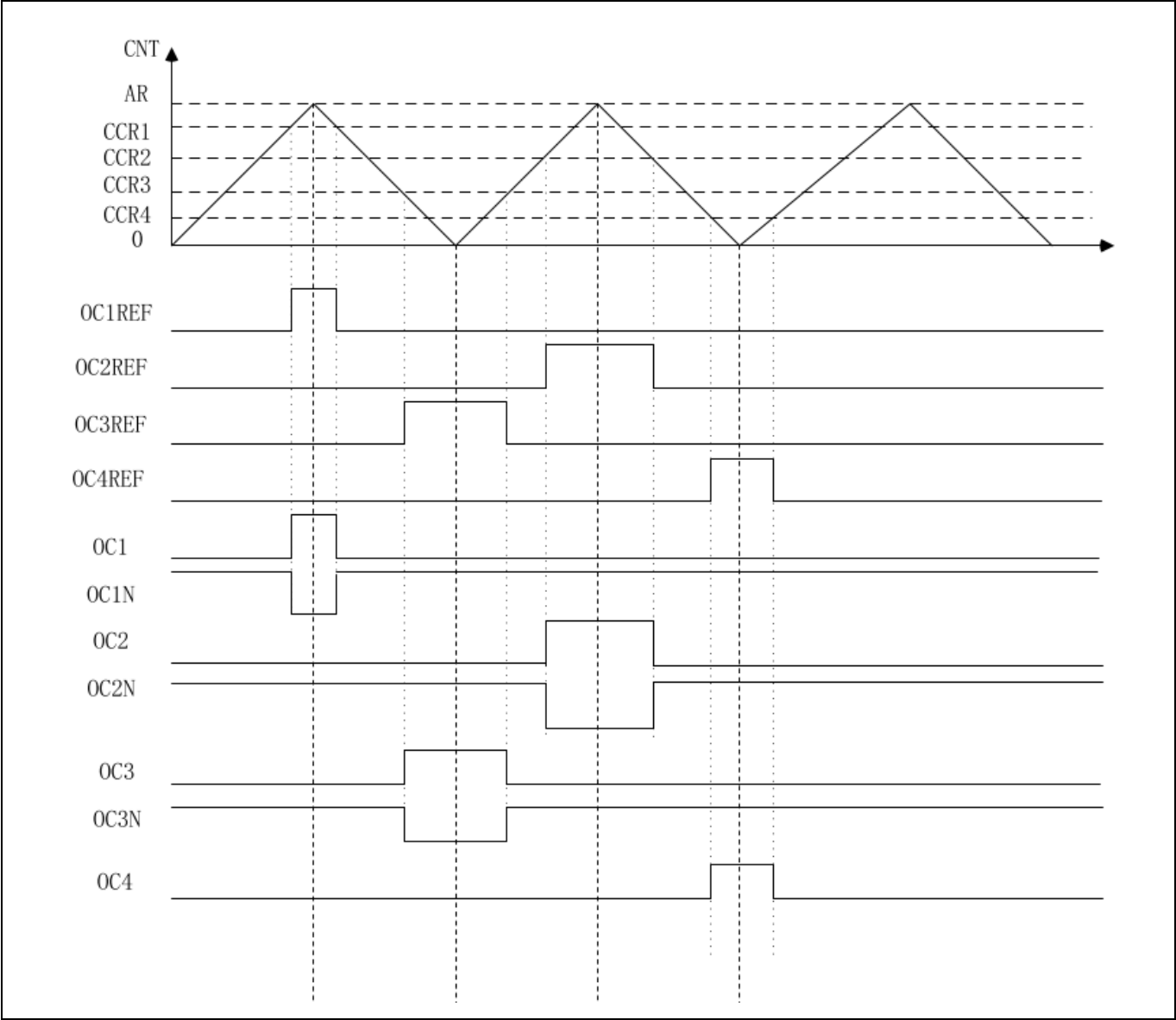
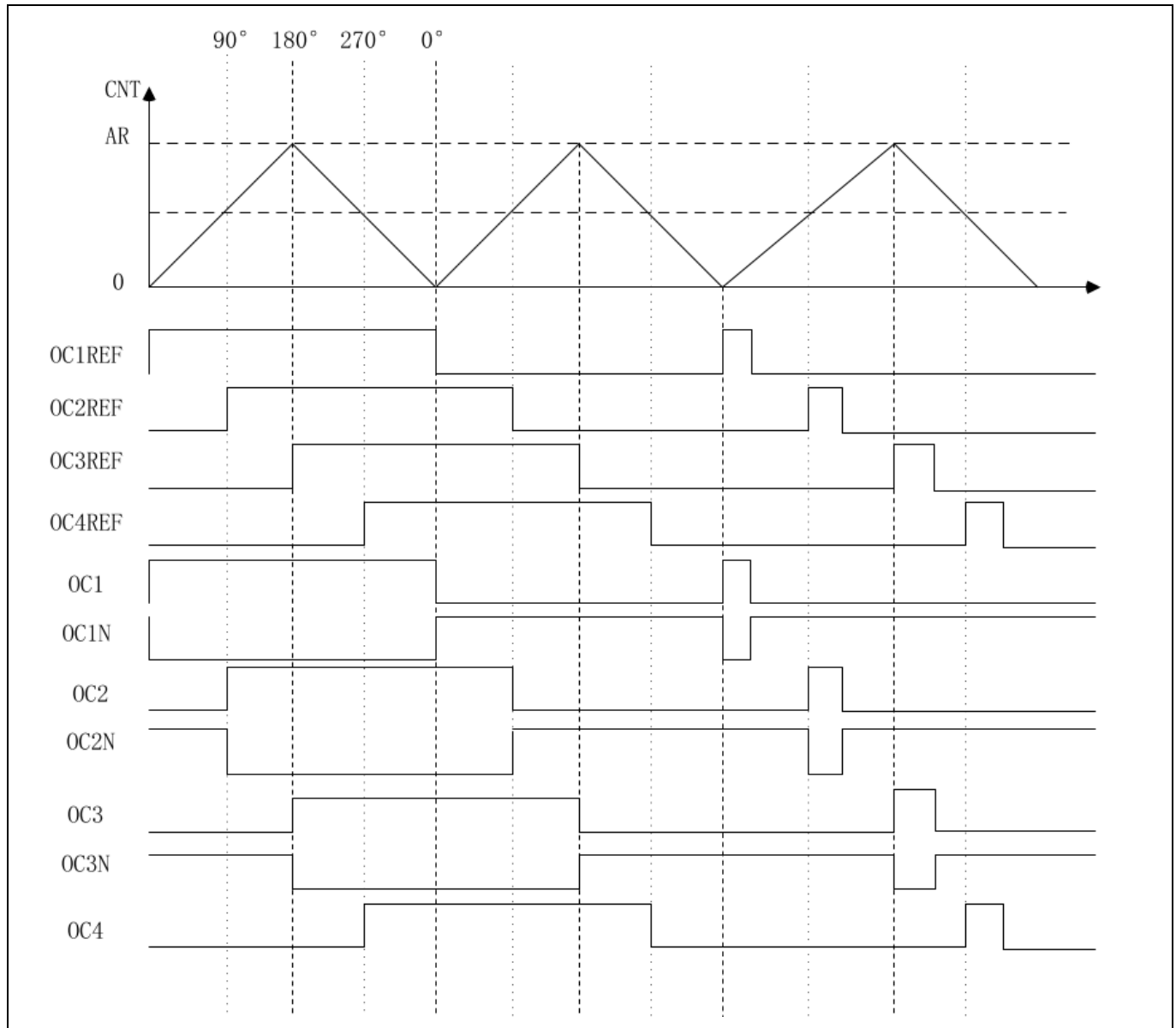
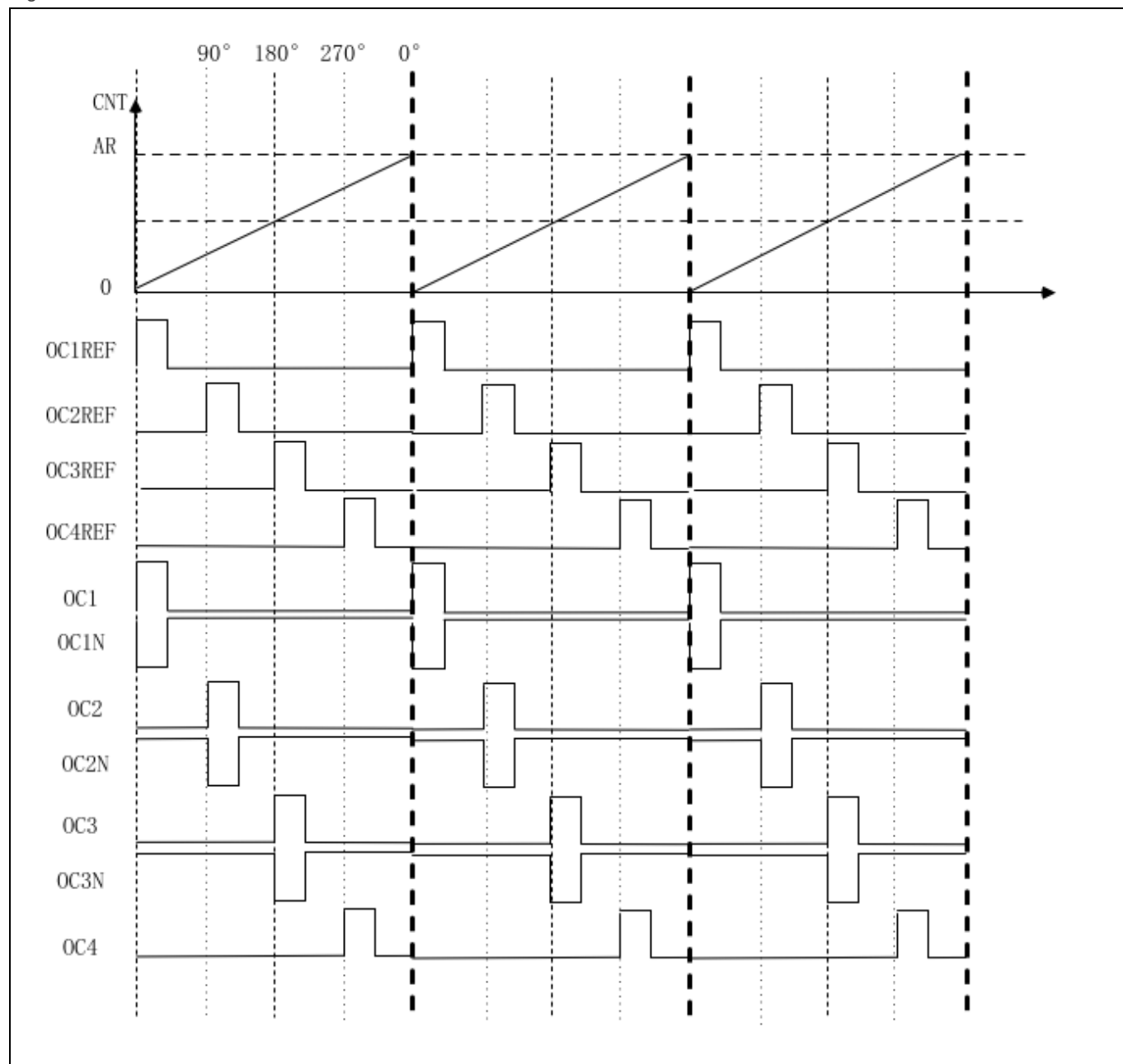


Figure 11-22



- In addition, leveraging the capability of CCRx and CCRxFALL to trigger actions based on up-counting and down-counting, a new function has been added: forcing the counter to zero upon an up-count overflow; this enables a 180° phase shift to be achieved within half a counting cycle.
- When the CNTFORCECL bit of the PDER register is enabled, it is possible to achieve phase-quadrature waveform output within a single triangular wave cycle.
- As shown in Figure 11-23 below:

Figure 11-23



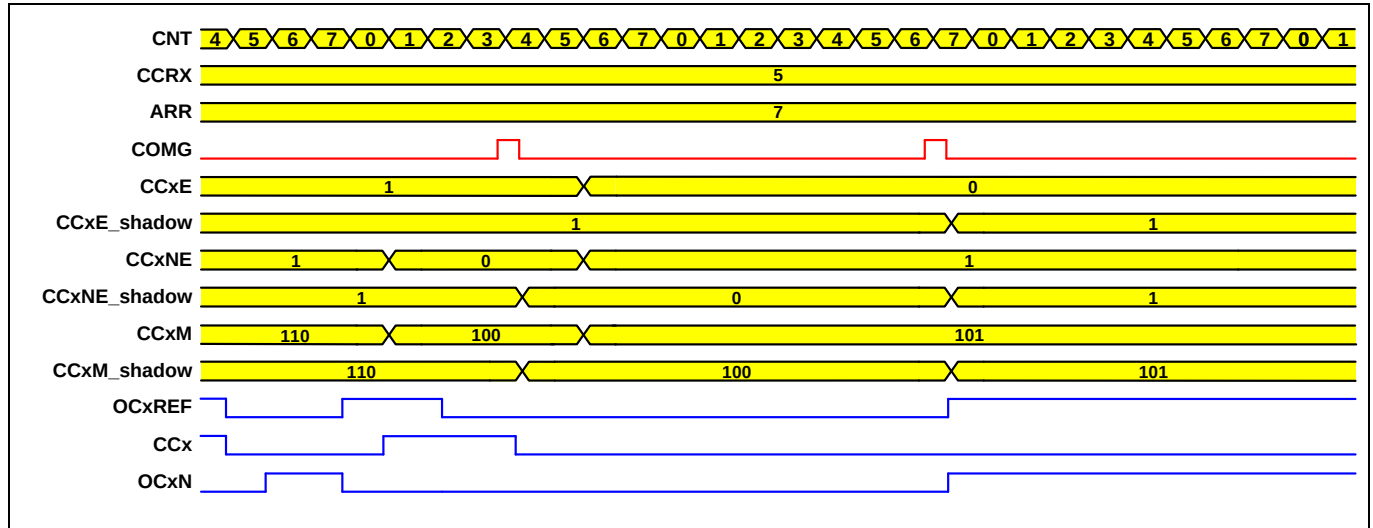
11.3.3.3.5 6-step PWM output

Configure OCxM and select output mode. CCxE=1 and CCxNE=1, Open channel x and complementary channel output enable. In channel x, generate complementary output. These functions are in preload bit. In case of COM commutation event, these preload bits will be loaded in the corresponding shadow register. Thus, writing these bits won't influence the current output, and can change the configuration of all the channels at the same time. Configure TIM1_EGR register COMG =1 or generate COM event at TRGI rising edge.

In case of COM event, COM interrupt flag will be set by hardware. Configure TIM1_DIER register COMIE=1 and COMDE = 1, generate COM event will generate one COM interrupt at the same time.

The figure below shows the OCx and OCxN outputs under different configurations when a COM event occurs.

Figure 11-24 Example of six step PWM with COM (OSSR = 1)

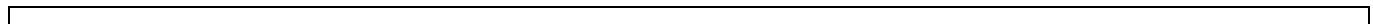


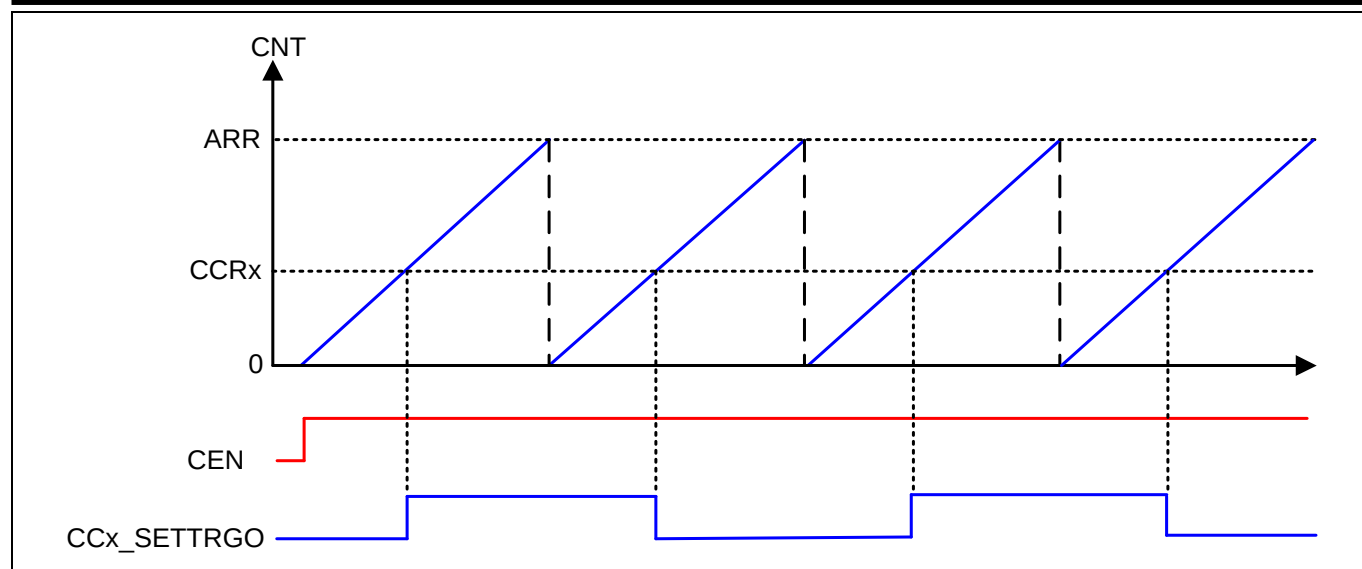
11.3.3.3.6 Trigger Source Output

In PWM mode, CCx_SETTRGO signal can be used to trigger modules such as ADC. This chapter only introduces CCx_SETTRGO. Please refer to the ADC chapter for the trigger logic of this signal, detailed trigger source selection, trigger edge selection, and other information.

In edge alignment mode, every time a match is compared (the current count value of TIMx_CNT is equal to TIMx_CCRx), CCx_SETTRGO undergoes a flip. The following figure shows CCx_SETTRGO in edge aligned incremental counting mode example of output.

Figure 11-25 CCx_SETTRGO in edge aligned incremental counting mode output example

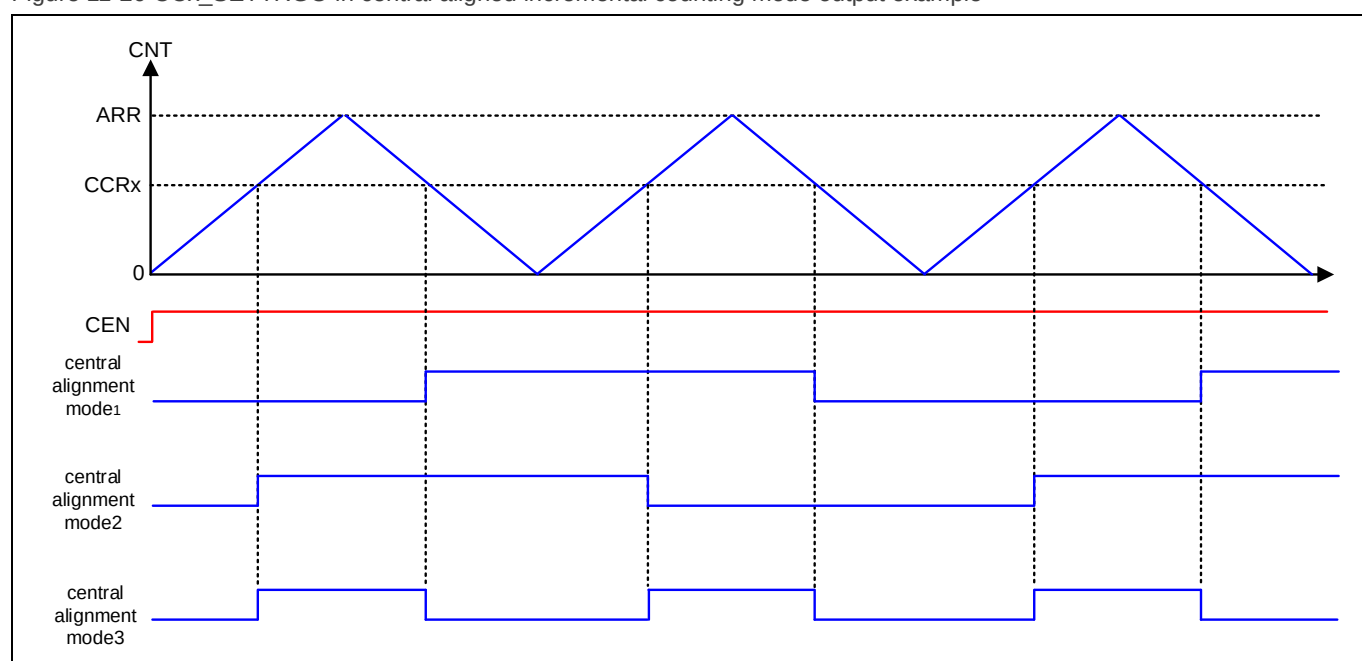




In central alignment mode 1, when comparing and matching the decreasing counting cycles, CCx_SETTRGO undergoes a flip. In central alignment mode 2, when comparing and matching the increasing counting cycles, CCx_SETTRGO undergoes a flip. In central alignment mode 3, CCx_SETTRGO is used when the increasing or decreasing counting cycles match undergoes a flip. The following figure shows CCx_SETTRGO in central alignment mode example of output.

Note: When using phase-shifting mode, the matching target for decreasing count comparison is CCRxFALL.

Figure 11-26 CCx_SETTRGO in central aligned incremental counting mode output example



11.3.3.4 Complementary output and dead-time insertion

OCx and OCxN are a pair of complementary output channels. TIM1 all channels can export complementary signals with switching-off and the switching-on instants of the outputs. It also has a programmable dead-time area. The user adjusts the dead time according to the connected output devices and their characteristics (intrinsic delays of level-shifters, delays due to power switches).

TIM1_BDTR register DTG [7:0] bit defines the dead-time when inserting the complementary output. The specific calculation mode is given in the table below:

Table 11-1 Dead-time calculation

DTG[7: 5]	DT
0xx	$DT = DTG[7 : 0] \times T_{dtg} \text{ (} T_{dtg} = T_{DTS} \text{)}$
10x	$DT = (64 + DTG[5 : 0]) \times T_{dtg} \text{ (} T_{dtg} = 2 \times T_{DTS} \text{)}$
110	$DT = (32 + DTG[4 : 0]) \times T_{dtg} \text{ (} T_{dtg} = 8 \times T_{DTS} \text{)}$
111	$DT = (32 + DTG[4 : 0]) \times T_{dtg} \text{ (} T_{dtg} = 16 \times T_{DTS} \text{)}$

For example, in case of $T_{DTS}=125\text{ns}$, possible dead time:

- In case of step time 125ns, dead time 0-15875ns.
- In case of step time 250ns, dead time 16us -31750ns.
- In case of step time 1us, dead time 32us -63us.
- In case of step time 2us, dead time 64us -126us.

In case of no break circuit, configure $CCxE=1$ and $CCxNE=1$, enable dead-time insert. Otherwise, configure $MOE = 1$.

Configure TIM1_CCER register $CCxP$ and $CCxNP$, and independently select polarity for each output (main output OCx or complementary output $OCxN$).

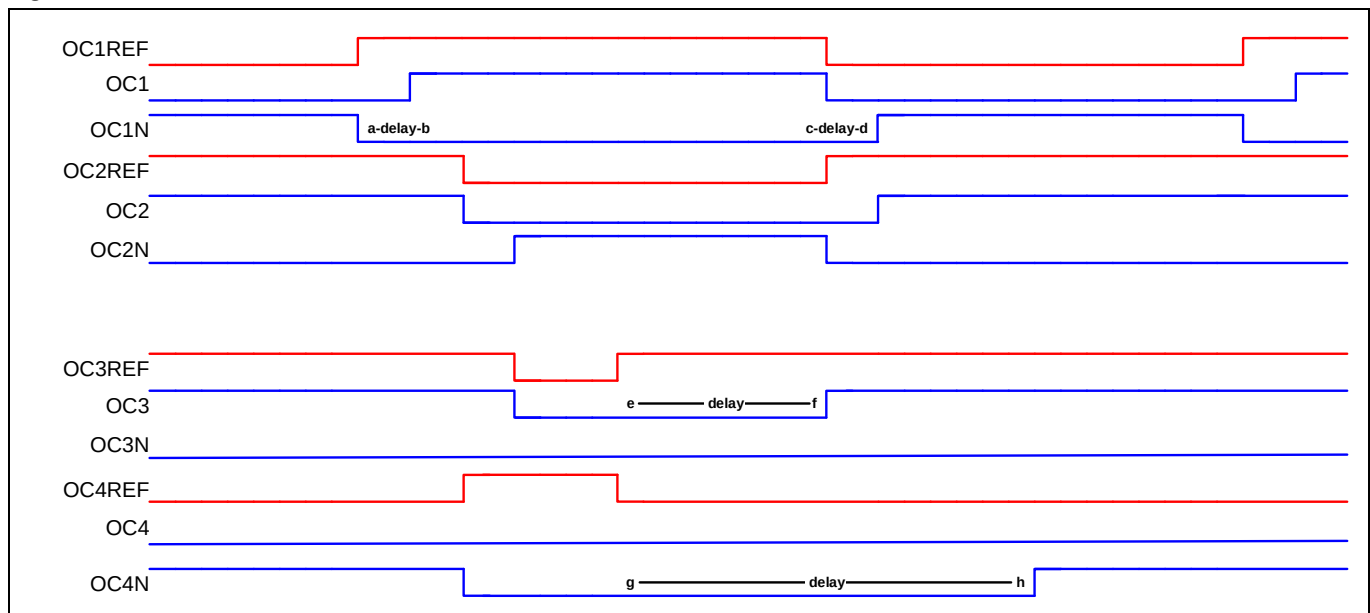
Configure TIM1_CCER register $CCxE$ and $CCxNE$ bit. Different combinations of MOE , $OISx$, $OISxN$, $OSSI$ and $OSSR$ in TIM1_BDTR and TIM1_CR2 register can control the complementary signal OCx and $OCxN$ output. The specific combination control configuration is given in the complementary output channel OCx and $OCxN$ control bit in table 3, table 4, table 5 and table 6 in this chapter.

For example: OCx and $OCxN$ are active high. In PWM mode, in case of match, export reference signal $OCxREF$ signal toggle and output signal OCx and reference signal are same. However, OCx signal's rising edge has a latency with response to the reference signal; complementary output signal $OCxN$ and reference signal are opposite. $OCxN$ signal's rising edge has a latency with response to the reference signal's falling edge.

Note: Dead time cannot be more than or equal to the duty cycle of OCx or $OCxN$ signal (active output). Or else, OCx or $OCxN$ signal are always inactive values.

The figures below show the relation of dead-time generator output signal and current reference signal $OCxREF$.

Figure 11-27 Dead-time insertion



11.3.3.5 Break function

Break input pin, clock failure event and comparator output may serve as TIM1 break source. The clock failure event is generated by the clock safety system of the reset clock counter.

During the use of break function, OCx and $OCxN$ output signal levels are controlled by the function level combination: TIM1_BDTR register MOE , $OSSI$ and $OSSR$ bit, and TIM1_CR2 register $OISx$ and $OISxN$ bit. In case

of break event, OCx and OCxN output cannot be active at the same time. The output state of complementary output channel OCx and OCxN with the break function is shown in the table below.

Table 11-2 In case of MOE=1, OSSI=0/1, OSSR=0:

CCxE	CCxNE	OCx	OCxN
0	0	OCx=0 , OCx_EN=0	OCxN=0 , OCxN_EN=0
0	1	OCx=0 , OCx_EN=0	OCxN=OCxREF+Polarity , OCxN_EN=1
1	0	OCx=OCxREF+Polarity , OCx_EN=1	OCxN=0 , OCxN_EN=0
1	1	OCx=OCxREF+Polarity+dead-time, OCx_EN=1	OCxN=OCxREF+reverse phase +Polarity+dead-time, OCxN_EN=1

Table 11-3 In case of MOE=1, OSSI=0/1, OSSR=1

CCxE	CCxNE	OCx	OCxN
0	0	OCx=0 , OCx_EN=0	OCxN=0 , OCxN_EN=0
0	1	OCx=CCxP , OCx_EN=1	OCxN=OCxREF+Polarity , OCxN_EN=1
1	0	OCx=OCxREF+Polarity, OCx_EN=1	OCxN=CCxNP , OCxN_EN=1
1	1	OCx=OCxREF+Polarity+dead-time, OCx_EN=1	OCxN=OCxREF reverse phase+Polarity+dead-time, OCxN_EN=1

Table 11-4 In case of MOE=0, OSSI=0, OSSR=0/1

CCxE	CCxNE	OCx	OCxN
0	0	OCx_EN=0 , OCxN_EN=0 Insync : OCx=CCxP , OCxN=CCxNP In case the clock is available: After a dead-time, OCx=OISx , OCxN=OISxN OISx and OISxN don't correspond to the active level of OCx and OCxN	
0	1	OCx_EN=0, OCxN_EN=0 Insync : OCx=CCxP , OCxN=CCxNP In case the clock is available: After a dead-time, OCx=OISx , OCxN=OISxN OISx and OISxN don't correspond to the active level of OCx and OCxN	
1	0	OCx_EN=0 , OCxN_EN=0 Insync : OCx=CCxP , OCxN=CCxNP In case the clock is available: After a dead-time, OCx=OISx , OCxN=OISxN OISx and OISxN don't correspond to the active level of OCx and OCxN	
1	1	OCx_EN=0 , OCxN_EN=0 Insync : OCx=CCxP , OCxN=CCxNP In case the clock is available: After a dead-time, OCx=OISx , OCxN=OISxN OISx and OISxN don't correspond to the active level of OCx and OCxN	

Table 11-5 In case of MOE=0, OSSI=1, OSSR=0/1

CCxE	CCxNE	OCx	OCxN
0	0	OCx_EN=1 , OCxN_EN=1 Insync : OCx=CCxP , OCxN=CCxNP In case the clock is available: After a dead-time, OCx=OISx , OCxN=OISxN OISx and OISxN don't correspond to the active level of OCx and OCxN	
0	1	OCx_EN=1, OCxN_EN=1 Insync : OCx=CCxP, OCxN=CCxNP In case the clock is available: After a dead-time, OCx=OISx , OCxN=OISxN OISx and OISxN don't correspond to the active level of OCx and OCxN	
1	0	OCx_EN=1 , OCxN_EN=1 Insync : OCx=CCxP , OCxN=CCxNP In case the clock is available: After a dead-time, OCx=OISx , OCxN=OISxN OISx and OISxN don't correspond to the active level of OCx and OCxN	
1	1	OCx_EN=1 , OCxN_EN=1 Insync : OCx=CCxP , OCxN=CCxNP In case the clock is available: After a dead-time, OCx=OISx , OCxN=OISxN OISx and OISxN don't correspond to the active level of OCx and OCxN	

Note: When the channel output and complementary output are closed, OISx, OISxN, CCxP and CCxNP must be configured as 0.

After system reset, MOE =0, disable break function. Configure TIM1_BKINF register BKIN_SEL and select break channels, it supports the selection of multiple break channels, it will enter the break state if break signal from any selected channel is active. Configure BKINF of TIM1_BKINF register, enable the break signal filtering function. Configure BKINF of TIM1_BKINF register, select the sampling frequency of break digital filter. Before changing the sampling frequency of break digital filter, turn off the break filter function. Configure TIM1_BDTR register BKE =1 and enable break function. Configure TIM1_BDTR register BKP bit and select break input signal polarity. BKP and BKE can also be written at the same time and become active in the next clock period.

Because MOE is asynchronously cleared, insert a synchronous circuit in between the actual signal and synchronous control, and generate latency between the synchronous and asynchronous signal (write MOE =1 when the state is 0; input an dummy instruction for latency after write and before read. Otherwise, correct reading cannot be guaranteed).

In case of break events, MOE is asynchronously cleared. According to OSSI configuration OCx/OCxN output, put it in inactive state, idle state and reset state; in case of MOE=0, the output is determined by TIM1_CR2 register OISx bit. In case of OSSI =0, the timer disables the output. Otherwise, it enables the output. When using the complementary output, the output is first put in the reset state. The dead-time is generated again. After the dead-time, the output level is determined by OISx and OISxN.

Configure TIM1_DIER register BIE =1. In case of a break event, generate a break interrupt. Configure TIM1_BDTR register AOE =1, auto set MOE bit when the next update event occurred.

Note: The break input is acting on level. When the break input is active, cannot (auto or via software) set MOE, and the status flag BIF cannot be cleared.

In the break circuit, enable write protection and safeguard application safety. Allow the user to lock the dead-time duration, OCx/OCxN polarity and disabled state, OCxM configuration, break enable and polarity. Configure TIM1_BDTR register LOCK bit, select lock levels (three levels of lock). Lock can only be written once after system reset.

The figure below shows the example of response break output:

Figure 11-28 Response break output (OISx=0 , OISxN=0)

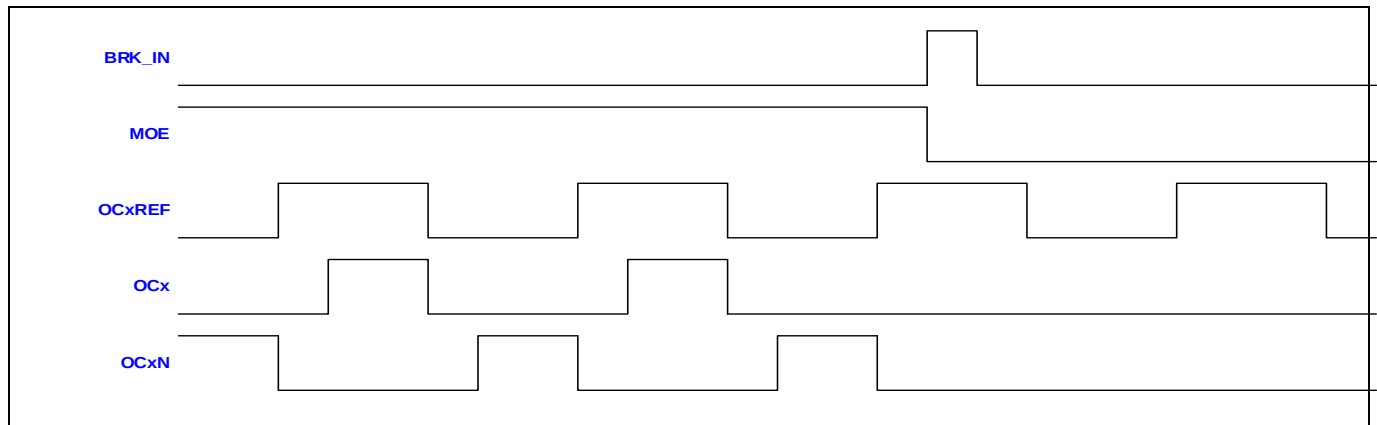
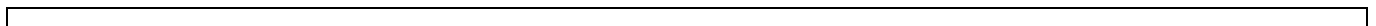


Figure 11-29 Response break output (OISx=0, OISxN=1)



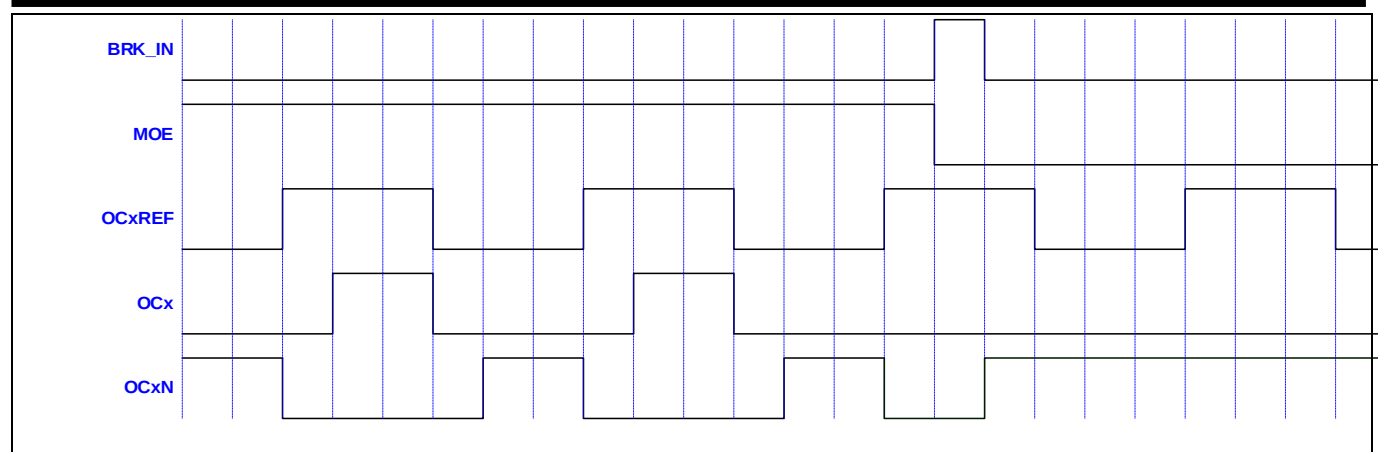


Figure 11-30 Response break output (OISx=1,OISxN=0)

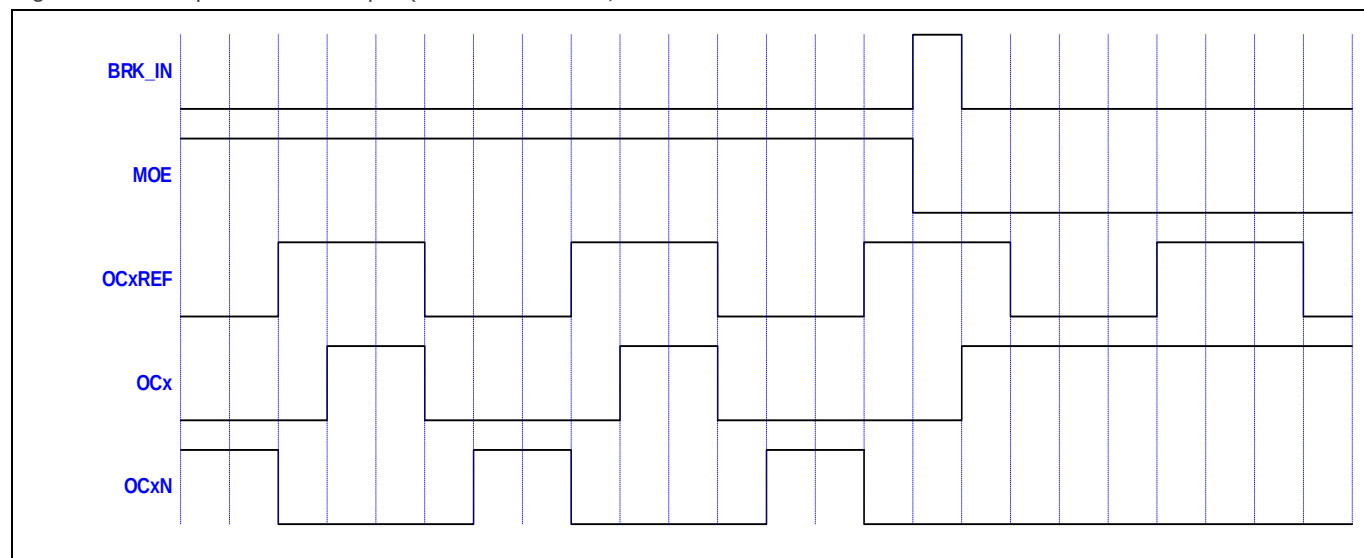
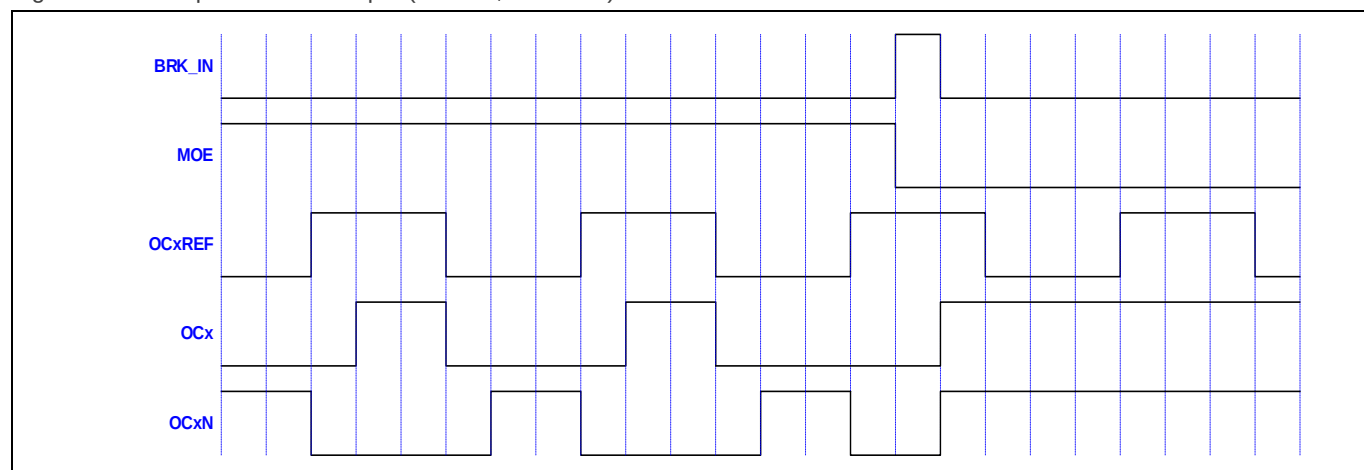


Figure 11-31 Response break output (OISx=1, OISxN=1)



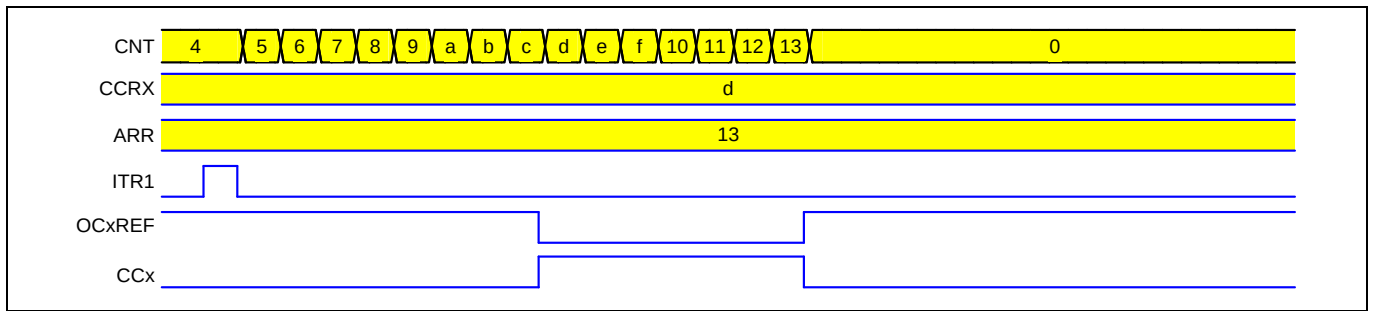
11.3.3.6 One-pulse output

In the one-pulse mode (OPM), the counter responds to a stimulus and generates a pulse with width programmable. Configure TIM1_CR1 register OPM =1. Select one-pulse mode, trigger signal's active edge or configuring CEN =1 can enable the counter. CEN =1 remains until the next update event or configure CEN =0.

The necessary condition to generate pulse is the difference of compare value and initial value of the counter. Necessary configuration prior to counter enable:

- Up counting mode: Counter $CNT < CCRx \leq ARR$.
- Down counting mode: Counter $CNT > CCRx$.

Figure 11-32 one-pulse mode



For example, when ITR1 detects the rising edge, and after latency t_{DELAY} , OC2 generates a positive pulse with the length of t_{PULSE} .

Configure ITR1 serves as the trigger source:

1. Configure TIM1_SMCR register TS = 001, ITR1 as the slave mode counter trigger (TRGI).
2. Configure TIM1_SMCR register SMS = 110, select trigger mode, ITR1 enables the counter.

OPM waveform is determined by TIM1_ARR and TIM1_CCR1 (consider clock frequency and counter prescaler): t_{DELAY} is determined by TIM1_CCR1 register value and CNT initial value; TIM1_ARR - TIM1_CCR1 value is t_{PULSE} .

Next example is generating a negative pulse. In case of compare and match, generate a waveform from 1 to 0. When the counter preload value is achieved, generate a waveform from 0 to 1:

1. Configure TIM1_CCMR1 register OC1M = 111, select PWM mode 2.
2. Configure TIM1_CCER register CC1P = 1, export active low level.
3. Configure TIM1_CCMR1 OC1PE = 1 and TIM1_CR1 register ARPE, enable preload register.
4. Configure TIM1_CCR1 register and TIM1_ARR register.
5. Configure TIM1_EGR register UG = 1, generate an update event.
6. Wait for an external trigger event on ITR1.

In this case, TIM1_CR1 register DIR=0, CMS=0, OPM= 1, stop counting in the next update event (when the counter toggles to 0 from the auto load value).

11.3.3.6.1 OCx fast enable

OCx fast enable is a special scenario of one-pulse mode. In the one-pulse mode, set TIM1_CCMR register OCxFE=1, and enforce OCxREF to directly respond to stimulus without relying on the compare result of counter and compare value. The output waveform is the same as the waveform when compare and match. Thus, it can remove the compare time and export the compare result fast. OCx rapid output enables only becomes active in the PWM mode.

11.3.4 Slave mode

11.3.4.1 Reset mode

Configure TIM1_SMCR register SMS=100, select slave reset mode. In this mode, the TRGI input event will clear and restart the counter.

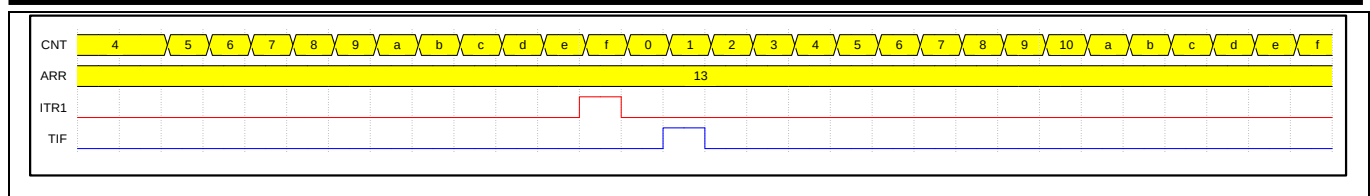
For example, ITR1 trigger counter restart:

1. Configure TIM1_SMCR register SMS = 100 and select the slave mode as reset mode. Configure TIM1_SMCR register TS = 001, select ITR1 as the trigger input of sync timer.
2. Configure TIM1_CR1 register DIR=0, select the count direction as the up count; configure PSC=0, no division; configure CEN=1, enable the counter.

The counter clock source is provided by the internal clock. When detecting ITR1 rising edge, clear and restart the counter. The trigger interrupt flag is set by hardware.

TIM1_ARR = 0x13 time sequence in reset mode

Figure 11-33 Control time sequence in reset mode



11.3.4.2 Gate mode

Configure TIM1_SMCR register SMS=101, select slave gate mode. When TRGI input is active level, the counter always starts. Otherwise, the counter stops (but without reset operation), and the counter start and stop are countable.

For example, the counter only counts when ITR1 is high:

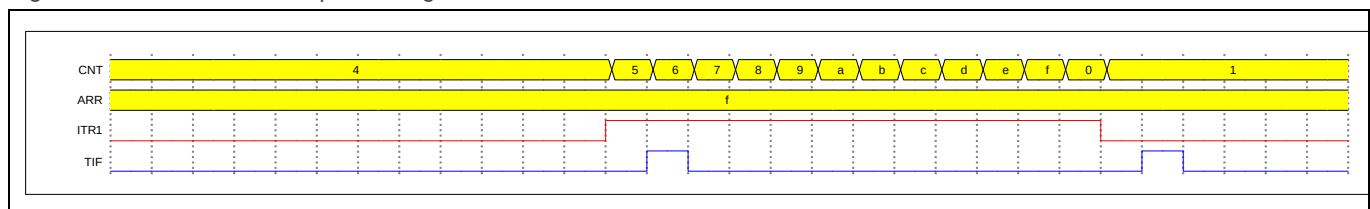
1. Configure TIM1_SMCR register SMS=101, select slave gate mode; configure TIM1_SMCR register TS=001, select the ITR1 as the sync counter trigger input.

2. Configure TIM1_CR1 register DIR=0, select count direction as the up count; configure PSC=0, no division; configure CEN=1, enable the counter.

The counter clock source is provided by the internal clock. When detecting ITR1 high level, the counter count starts. When ITR1 is low level, the counter stops. When the counter starts or stops, set TIF as 1.

TIM1_ARR=0xf time sequence in gate mode.

Figure 11-34 Control time sequence in gate mode



11.3.4.3 Trigger mode

Configure TIM1_SMCR register SMS=110 and select slave trigger mode. When TRGI input is active edge, the counter starts counting. The timer startup is controlled, and the stop is not controlled.

For example, the counter count starts at the rising edge of ITR1 input:

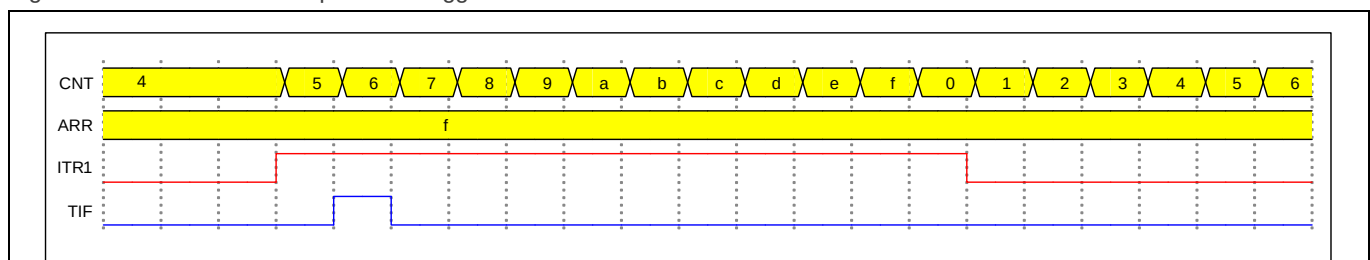
1. Configure TIM1_SMCR register SMS = 110, select slave trigger mode; configure TIM1_SMCR register TS=001, select ITR1 as the trigger input of counter.

2. Configure TIM1_CR1 register DIR=0, select count direction as the up count; configure PSC=0, no division.

The counter clock source is provided by internal clock. When detecting ITR1 rising edge, the counter count starts.

TIM1_ARR =0xf sequence in the trigger mode.

Figure 11-35 Control time sequence in trigger mode

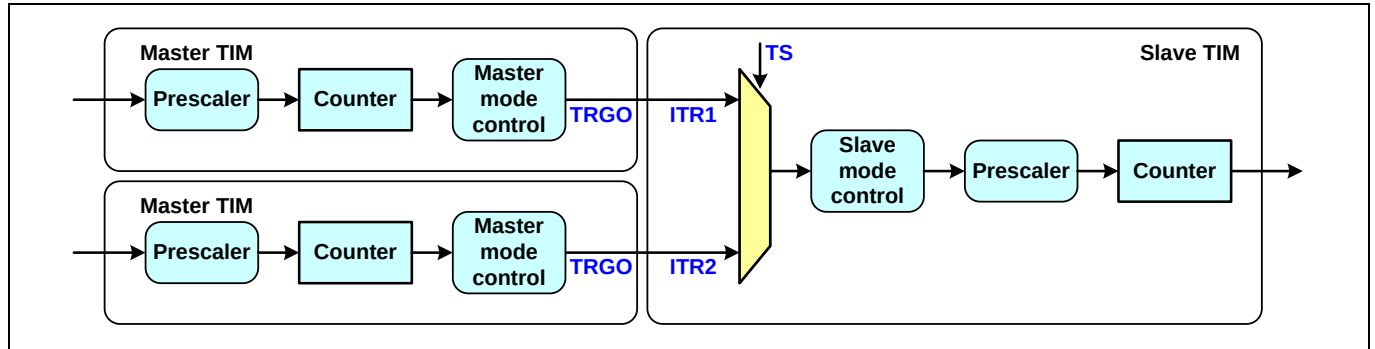


11.3.5 Timer synchronization

Different timers are internally connected, achieve timer cascading or synchronization.

Timer synchronization connectivity:

Figure 11-36 Timer connectivity



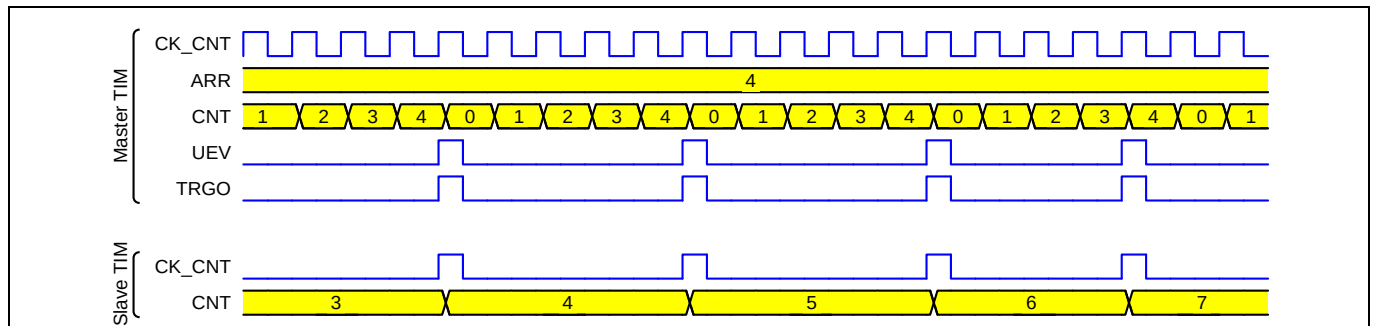
Several typical applications of connectivity.

Use one timer as the prescaler of another timer

For example, use master timer as slave timer prescaler, the time sequence diagram is shown above:

1. Configure master timer CR2 register MMS=010, master timer update event as trigger input (TRGO), master timer exports a cyclic signal at each update event.
2. Configure master timer ARR register, set master timer output period.
3. Configure slave timer SMCR register TS, select slave timer trigger source as master timer TRGO;
4. Configure slave timer SMCR register SMS=111, select slave external trigger mode 1.
5. Configure master timer CR1 register CEN=1, start master timer.
6. Configure slave timer CR1 register CEN=1, start slave timer.

Figure 11-37 Use master timer as slave timer prescaler



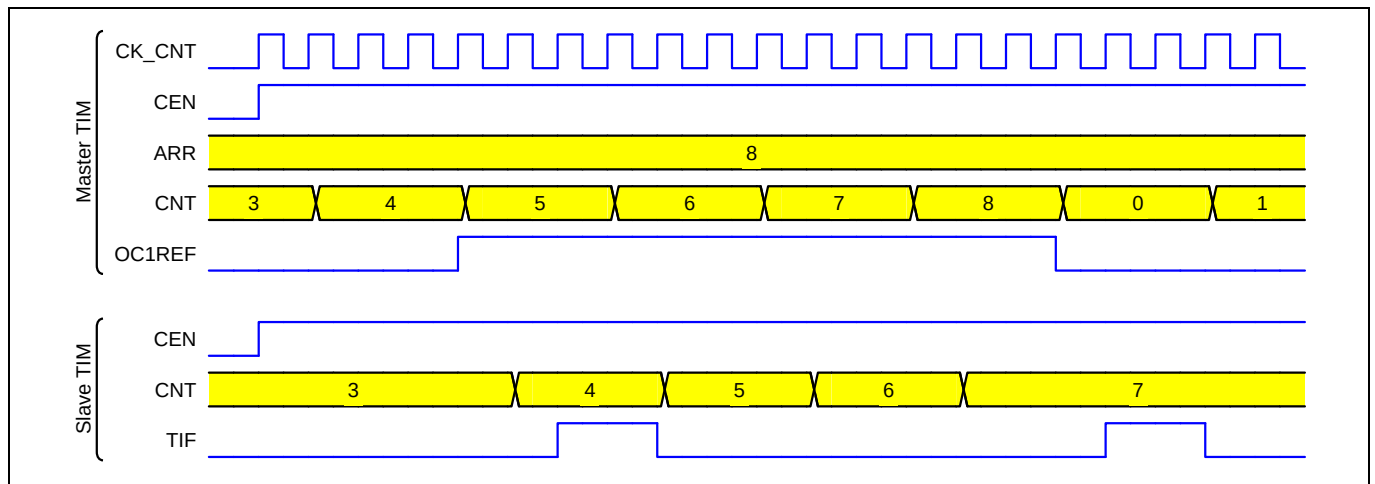
Use one timer to enable the other timer

For example: Use master timer to enable slave timer, and the time sequence diagram is shown below:

When master timer OC1REF is high, slave timer counter count begins. The clock frequency of both timers is provided by CK_CNT/3. Specific configuration:

1. Configure master timer CR2 register MMS=100, configure master timer output compare reference signal (OC1REF) as trigger output (TRGO).
2. Configure master timer CCR1 register, ARR register, CCMR1 register OC1M bit, configure the output waveform of master timer output signal TRGO.
3. Configure slave timer SMCR register TS, configure master timer OC1REF as slave timer trigger output.
4. Configure slave timer SMCR register SMS=101, configure slave timer as gate mode.
5. Configure slave timer CR1 register CEN=1, enable slave timer.
6. Configure master timer CR1 register CEN=1, enable master timer.

Figure 11-38 Use master timer to enable slave timer



Use one timer to start another timer

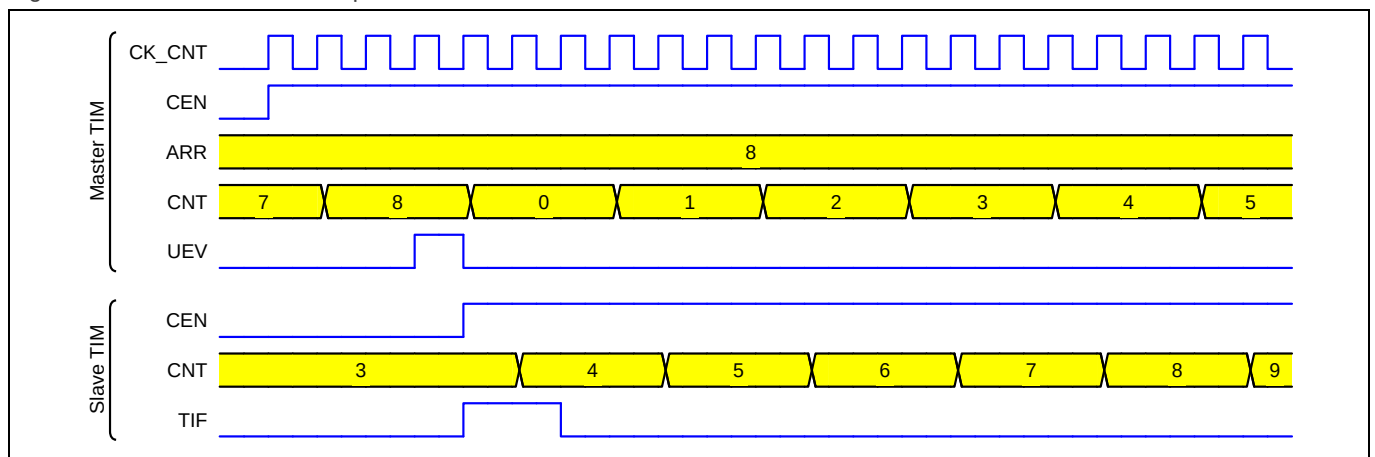
For example: Use master timer update event to enable slave timer, and the time sequence is shown in the figure below:

When master timer generate the update event, slave timer receives the trigger signal, slave timer CEN is auto set by hardware. Slave timer counter count begins. The clock frequency of two timers is provided by CK_CNT/3.

The specific configuration:

1. Configure master timer CR2 register MMS=010, select master timer update event as trigger output (TRGO);
2. Configure master timer ARR register, configure the period of update event;
3. Configure slave timer SMCR register TS, configure master timer TRGO as slave timer trigger input;
4. Configure slave timer SMCR register SMS=110, configure slave timer as trigger mode;
5. Configure master timer CR1 register CEN=1, enable master timer.

Figure 11-39 Use master timer update event to start slave timer



Use one external trigger to start two timers in step

For example: Use master timer ETR rising edge start master timer and slave timer, the time sequence is shown in the figure below:

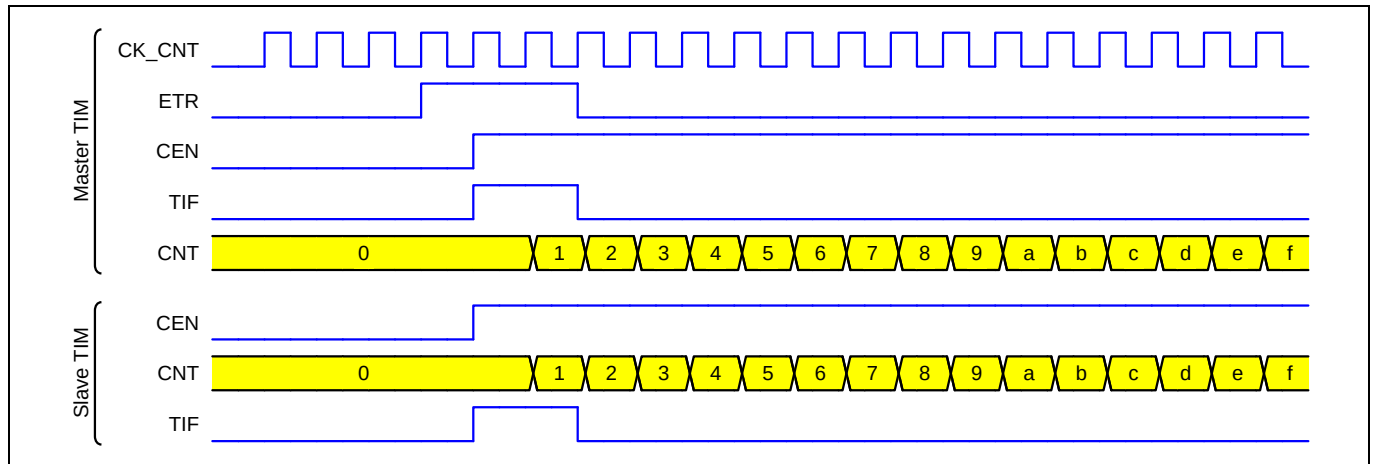
In order to ensure simultaneous start two timers, master timer must be configured in master/slave mode. Specific configuration:

1. Configure master timer CR2 register MMS=001, make master timer enable signal CEN as trigger output (TRGO).
2. Configure master timer SMCR register TS=111, make ETR as trigger input.
3. Configure master timer SMCR register SMS=110, configure master timer as trigger mode.
4. Configure slave timer SMCR register TS, configure master timer trigger output as slave timer trigger input.

5. Configure slave timer SMCR register SMS=110, configure slave timer as trigger mode.

When master timer ETR generates the rising edge, two timers start simultaneously (according to internal clock). The counter count begins, and two TIF flags are set.

Figure 11-40 master timer ETR start master and slave timer in step



11.3.6 Debug mode

In the debug mode, configure DBG_CR register DBG_TIM1_STOP=1, TIM1 counter stops counting. (Details in the debug section)

In debug mode, configure the DOS bits in the TIM1_DOCR register to set the output state.

11.3.7 Interrupt

TIM1 interrupt includes: compare 1 interrupt, compare 2 interrupt, compare 3 interrupt, compare 4 interrupt, compare 5 interrupt, update interrupt, COM interrupt, trigger interrupt and break interrupt. When the corresponding interrupt enable bit is set, generate the corresponding event and interrupt.

Table 11-6 List of interrupt events

Interrupt event	Flag bit	Enable bit
Compare 1 interrupt	CC1IF	CC1IE
Compare 2 interrupt	CC2IF	CC2IE
Compare 3 interrupt	CC3IF	CC3IE
Compare 4 interrupt	CC4IF	CC4IE
Compare 5 interrupt	CC5IF	CC5IE
Update interrupt	UIF	UIE
COM interrupt	COMIF	COMIE
Trigger interrupt	TIF	TIE
Break interrupt	BIF	BIE

11.3.8 Software Output Settings

TIM1 allows for the simple configuration of OCx and OCxN output values for channels CH1~4 via the SOER and SOCR registers, covering both direct output and output states when the brake is active.

The specific steps are as follows:

Set SOEN to 1 in the SOER register to enable the software output function for channels CH1 through CH4.

1. If only software output is required, configure the CH bits corresponding to SFTEN and SFTV in the SOCR register and set the desired OCx and OCxN output levels; this configuration takes effect immediately.
2. When braking occurs, the software output is interrupted, and the MOE output pulls the corresponding PWM signal low; once braking ceases, the Automatic Output Enable (AOE) function—provided it is active and an update event occurs—restores the software output.

3. If software output is required only during a brake event, configure the CH bits corresponding to BSFTEN and BSFTV in the SOCR register and set the desired OCx and OCxN output levels; the configured output takes effect only when a brake event occurs. Once the brake condition is cleared, if Automatic Output Enable (AOE) is active and an update event occurs, the software-driven brake output is released, and normal PWM output resumes.
4. If both the standard software output and the braking-state software output are configured, the standard software output is produced during normal operation, while the output configured for the braking state is produced when braking occurs.

11.4 Register

Table 11-7 TIM1 register overview

Offset	Acronym	Register Name	Reset
0x00	TIM1_CR1	Control Register 1	0x0000
0x04	TIM1_CR2	Control Register 2	0x0000
0x08	TIM1_SMCR	Slave Mode Control Register	0x0000
0x0C	TIM1_DIER	Interrupt Enable Register	0x0000 0000
0x10	TIM1_SR	Status Register	0x0000 0000
0x14	TIM1_EGR	Event Generation Register	0x0000 0000
0x18	TIM1_CCMR1	Compare Mode Register 1	0x0000
0x1C	TIM1_CCMR2	Compare Mode Register 2	0x0000
0x20	TIM1_CCER	Compare Enable Register	0x0000
0x24	TIM1_CNT	Counter	0x0000
0x28	TIM1_PSC	Prescaler	0x0000
0x2C	TIM1_ARR	Auto Reload Register	0x0000
0x30	TIM1_RCR	Repeat Count Register	0x0000
0x34	TIM1_CCR1	Compare Register 1	0x0000
0x38	TIM1_CCR2	Compare Register 2	0x0000
0x3C	TIM1_CCR3	Compare Register 3	0x0000
0x40	TIM1_CCR4	Compare Register 4	0x0000
0x44	TIM1_BDTR	Break And Dead-Time Register	0x0000 0000
0x54	TIM1_CCMR3	Compare Mode Register 3	0x0000
0x58	TIM1_CCR5	Compare Register 5	0x0000
0x5C	TIM1_PDER	PWM Phase Shift Enable Register	0x0000
0x60~0x70	TIM1_CCRxFALL	PWM Phase Shift Count Down Compare Register 1~5	0x0000
0x74	TIM1_BKINF	Break Input Filter Register	0x0000 0000
0x80	TIM1_DOGR	Debug Output Control Register	0x0000
0x84	TIM1_SOER	Software Output Global Enable Register	0x0000
0x88	TIM1_SOCR	Software Input Enable/Output Value Register	0x0000
0x8C	TIM1_BKSR	Break Input Status Register	0x0000

11.4.1 TIM1_CR1 Control Register 1

Address offset: 0x00

Reset value: 0x0000

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

Reserved	CKD	ARPE	CMS	DIR	OPM	URS	UDIS	CEN
	rW	rW	rW	rW	rW	rW	rW	rW

Bit	Field	Description
15 : 10	Reserved	Reserved, must be kept at reset value.
9 : 8	CKD	Clock division Division ratio between the timer clock (CK_INT) frequency and the dead-time and sampling clock used by the dead-time generators and the digital filters (ETR, Tlx). 00 : $t_{DTS} = t_{INT_CK}$ 01 : $t_{DTS} = 2 \times t_{INT_CK}$ 10 : $t_{DTS} = 4 \times t_{INT_CK}$ 11 : Reserved, do not program this value
7	ARPE	Auto reload preload enable 0: Disable the shadow register of TIM1_ARR register 1: Enable the shadow register of TIM1_ARR register
6 : 5	CMS	Center alignment mode selection 00: Edge alignment mode. Count direction depends on DIR bit 01: Central alignment mode 1. The counter alternatively conducts up and down count. The channel is in output mode. Only in down count, compare interrupt flag bit is set 10: Central alignment mode 2. The counter alternatively conducts up and down count. The channel is in output mode. Only in up count, compare interrupt flag bit is set 11: Central alignment mode 3. The counter alternatively conducts up and down count. The channel is in output mode. In up and down count, compare interrupt flag bit is set Note: During count, the alignment mode change is disabled.
4	DIR	Count direction 0: up count 1: down count Note: When the counter is configured as central alignment mode, the bit is read only.
3	OPM	one-pulse mode 0: Disable one-pulse mode. In case of update event, the counter count continues 1: Enable one-pulse mode. In case of the next update event (clear CEN bit), the counter count stops
2	URS	Update request source This bit is set and cleared by software, select update event source. 0: The event below may generate a update interrupt or DMA request: - Counter overflow/underflow - Set UG bit - Update generation through the slave mode controller 1: Only in counter overflow/underflow, generate update interrupt or DMA request
1	UDIS	Update disables This bit is used to enable or disable the update event 0: Update event (UEV) enabled. 1: Update event disabled. The Update event is not generated, shadow registers keep their value (ARR, PSC, CCRx). However, the counter and the prescaler are reinitialized if the EGR.UG bit is set, the counter is reinitialized if a hardware reset is received from the slave mode controller.
0	CEN	Counter enable 0: Counter disabled 1: Counter enabled Note: External clock, gated mode can work only if the CEN bit has been previously set by software. However trigger mode can set the CEN bit automatically by hardware.

11.4.2 TIM1_CR2 Control Register 2

Address offset: 0x04

Reset value: 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	OIS4	OIS3N	OIS3	OIS2N	OIS2	OIS1N	OIS1	Res.	MMS			RES.	CCUS	Res.	CCPC
	rw	rw	rw	rw	rw	rw	rw		rw				rw		rw

Bit	Field	Description
15	Reserved	Reserved, must be kept at reset value.
14	OIS4	Output idle state 4 (OC4 output). Refer to OIS1 bit.
13	OIS3N	Output idle state 3 (OC3N output). Refer to OIS1N bit.
12	OIS3	Output idle state 3 (OC3 output). Refer to OIS1 bit.
11	OIS2N	Output idle state 2 (OC2N output). Refer to OIS1N bit.
10	OIS2	Output idle state 2 (OC2 output). Refer to OIS1 bit.
9	OIS1N	(Output idle state 1) (OC1N output) 0: In case of MOE =0, OC1N =0 after dead-time 1: In case of MOE =0, OC1N =1 after dead-time Note: After setting LOCK (TIM1_BKR register) level 1, 2 or 3, this bit cannot be changed.
8	OIS1	(Output idle state 1) (OC1 output) 0: In case of MOE=0, if OC1N is implemented, OC1=0 after dead-time 1: In case of MOE=0, if OC1N is implemented, OC1 = 1 after dead-time Note: After setting LOCK (TIM1_BKR register) level 1, 2 or 3, this bit cannot be changed.
7	Reserved	Reserved, must be kept at reset value.
6 : 4	MMS	Master mode selection These bits control TRGO signal selection, used to select the sync information sent to the slave timers in master mode: 000: Reset TIM1_EGR register UG bit generate one pulse trigger output (TRGO). 001: Enable The Counter Enable signal CNT_EN is used as trigger output (TRGO). It is useful to start several timers at the same time or to control a window in which a slave timer is enable. The Counter Enable signal is generated by a logic OR between CEN control bit and the trigger input when configured in gated mode. When the Counter Enable signal is controlled by the trigger input, there is a delay on TRGO, except if the master/slave mode is selected. 010: Update Update event is selected as TRGO. 011: Compare When a compare match occurred, send a positive pulse as TRGO. 100: Compare OC1REF signal is used as trigger output (TRGO) 101: Compare OC2REF signal is used as trigger output (TRGO) 110: Compare OC3REF signal is used as trigger output (TRGO) 111: Compare OC4REF signal is used as trigger output (TRGO)
3	Reserved	Reserved, must be kept at reset value.
2	CCUS	Compare control update selection 0: CCPC=1, they are updated by setting the COMG=1 only. 1 : CCPC=1, they are updated by setting the COMG=1 or when a rising edge occurs on TRGI. Note: This bit acts only on channels that have a complementary output.
1	Reserved	Reserved, must be kept at reset value.
0	CCPC	Compare preloaded control bit 0: CCxE, CCxNE and OCxM bit preload disable 1: CCxE, CCxNE and OCxM bit preload enable Note: This bit acts only on channels that have a complementary output.

11.4.3 TIM1_SMCR Slave Mode Control Register

Address offset: 0x08

Reset value: 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved										TS[4:3]		Reserved			
										rw					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								MSM	TS[2:0]		OCCS	SMS			
								rw	rw		rw	rw			

Bit	Field	Description
31 : 22	Reserved	Reserved, must be kept at reset value.
21 : 20	TS[4:3]	Trigger selection Trigger input source selection. Refer to the TS[2:0] bit description.
19 : 8	Reserved	Reserved, must be kept at reset value.
7	MSM	Master/slave mode 0: No action 1: The effect of an event on the trigger input (TRGI) is delayed allowing a perfect synchronization between the current timer and its slaves (through TRGO). It is useful if we want to synchronize several timers on a single external event.
6 : 4	TS[2:0]	Trigger selection These bits, combined with bits TS[4:3], are used to trigger selection of the input source. 00000 : Internal trigger 0 (ITR0) 00001 : Internal trigger 1 (ITR1) 00010 : Internal trigger 2 (ITR2) 00011 : Internal trigger 3 (ITR3) 00100 : Reserved 00101 : Reserved 00110 : Reserved 00111 : Reserved Others: Reserved For more details on ITRx, refer to the chip-specific configuration section. Note: After the slave mode enable, these bits cannot be changed
3	OCCS	Timer compare output signal (OCxREF) clear selection In PWM mode, clear the compare output signal (OCxREF) 0: External trigger signal as clear signal 1: Comparator output (COMP) as clear signal Note: It's only applicable to products that support external trigger or have a built-in comparator (COMP)
2 : 0	SMS	Slave mode selection When external signals are selected the active edge of the trigger signal (TRGI) is linked to the polarity selected on the external input. 000: Close slave mode - In case of CEN =1, the prescaler is directly driven by the internal clock. 001: Reserved 010: Reserved 011: Reserved 100: Reset mode -The rising edge of the selected trigger input (TRGI) reinitialize the counter and generate an update event. 101: Gate mode - When the trigger input (TRGI) is high, the counter count begins and generate an update event. When the trigger input turns low, the counter count stops (but without reset) and generates an update event. The counter start and stop are controlled. 110: Trigger mode -The counter starts in the rising edge of the trigger input TRGI (but without reset) and generate an update event. Only the counter start is controlled. 111: External clock mode 1 -The rising edge of the selected trigger input (TRGI) drives the counter and generate an update event.

Note: For internal trigger interconnection relationships, refer to the interconnection table between TIM in the chip-specific configuration section.

11.4.4 TIM1_DIER Interrupt Enable Register

Address offset: 0x0C

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															CC5IE
															rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.								BIE	TIE	COMIE	CC4IE	CC3IE	CC2IE	CC1IE	UIE
								rw	rw	rw	rw	rw	rw	rw	rw

Bit	Field	Description
31 : 17	Reserved	Reserved, must be kept at reset value.
16	CC5IE	Compare 5 interrupt enable 0: Compare 5 interrupt disable 1: Compare 5 interrupt enable
15 : 8	Reserved	Reserved, must be kept at reset value.
7	BIE	Break interrupts enable 0: Break interrupts disable 1: Break interrupts enable
6	TIE	Trigger interrupts enable 0: Break interrupts disable 1: Break interrupts enable
5	COMIE	Enable COM interrupt 0: COM interrupts disable 1: COM interrupts enable
4	CC4IE	Enable compare 4 interrupts 0: Compare interrupt 4 disable 1: Compare interrupt 4 enable
3	CC3IE	Enable compare 3 interrupts 0: Compare interrupt 3 disable 1: Compare interrupt 3 enable
2	CC2IE	Enable compare 2 interrupts 0: Compare interrupt 2 disable 1: Compare interrupt 2 enable
1	CC1IE	Enable compare 1 interrupt 0: Compare interrupt 1 disable 1: Compare interrupt 1 enable
0	UIE	Enable update interrupt 0: Update event interrupt disable 1: Update event interrupt enable

11.4.5 TIM1_SR Status Register

Address offset: 0x10

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															CC5IF
															rw0c
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								BIF	TIF	COMIF	CC4IF	CC3IF	CC2IF	CC1IF	UIF
								r_w0c							

Bit	Field	Description
31 : 17	Reserved	Reserved, must be kept at reset value.

Bit	Field	Description
16	CC5IF	Compare 5 interrupt flags Refer to CC1IF description.
15 : 8	Reserved	Reserved, must be kept at reset value.
7	BIF	Break interrupt flag When the break input is active, the bit is set by hardware. If the Break input is inactive, the bit is set as 0 by software 0: No break event occurred 1: Detect active level on Break input
6	TIF	Trigger interrupt flag In case of trigger event (When the slave mode counter is in any other mode except for the gate mode, detect the active edge in TRGI input terminal, or detect any edge in gate mode), the bit is set by hardware. It's cleared by software. 0: No trigger event occurred 1: Trigger interrupt pending
5	COMIF	COM interrupt flag In case of COM event (compare control bit: CCxE, CCxNE, OCxM updated), the bit is set by hardware. It is cleared by software. 0 : No COM event occurred 1: COM interrupt pending
4	CC4IF	Compare 4 interrupt flag Refer to CC1IF description
3	CC3IF	Compare 3 interrupt flag Refer to CC1IF description
2	CC2IF	Compare 2 interrupt flag Refer to CC1IF description.
1	CC1IF	Compare 1 interrupt flag When the counter value and compare value match, the bit is set by hardware except for the central alignment mode (the bit is set in the central alignment mode according to TIM1_CR1.CMS[1:0]). It's cleared by software. 0: No match 1: TIM1_CNT value and TIM1_CCR1 value match
0	UIF	Update interrupt flag In case of update event, the bit is set by hardware. It's cleared by software. 0: No update interrupt occurred 1: update interrupt pending This bit is set by hardware when the registers are updated: - In case of TIM1_CR1 register UDIS=0 and REP_CNT=0, when the counter generates overflow/underflow. -In case of TIM1_CR1 register UDIS=0, URS=0, when TIM1_EGR register UG =1. - In case of TIM1_CR1 register UDIS=0, URS=0, when update generation through the slave mode controller.

11.4.6 TIM1_EGR Event Generation Register

Address offset: 0x14

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															CC5G
															W
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								BG	TG	COMG	CC4G	CC3G	CC2G	CC1G	UG
								W	W	W	W	W	W	W	W

Bit	Field	Description
31 : 17	Reserved	Reserved, must be kept at reset value.
16	CC5G	Compare 5 generation Refer to CC1G description.
15 : 8	Reserved	Reserved, must be kept at reset value.

Bit	Field	Description
7	BG	Break generation 0: No action 1: generate a break event. MOE =0, BIF =1; when producing the corresponding interrupt and DMA, generate the corresponding interrupt and DMA. It's cleared by hardware.
6	TG	Trigger generation 0: No action 1: generate a trigger event. TIM1_SR register TIF =1. When enabling the corresponding interrupt and DMA, generate the corresponding interrupt and DMA. It's auto cleared by hardware.
5	COMG	Compare control update generation 0: No action 1: Compare event control update. It's auto cleared by hardware. In case of CCPC =1, enable update CCxE, CCxNE and OCxM bit. Note: This bit is only active for the complementary output channel.
4	CC4G	Compare 4 generation Refer to CC1G description.
3	CC3G	Compare 3 generation Refer to CC1G description.
2	CC2G	Compare 2 generation Refer to CC1G description.
1	CC1G	Compare 1 generation Refer to CC1G description. This bit is set by software. It generates a compare event, and is auto cleared by hardware. 0: No action 1 : Generate a compare event in channel CC1: Set CC1IF =1. When enabling the corresponding interrupt and DMA, generate the corresponding interrupt and DMA.
0	UG	Update event generation 0 : No action 1: Initialize the counter, and generate an update event. It's auto cleared by hardware. When selecting the central alignment or up counting mode, the counter is clear; otherwise (down counting mode) the counter auto reload value is loaded. The prescaler counter is cleared at the same time.

11.4.7 TIM1_CCMR1 Compare Mode Register 1

Address offset: 0x18

Reset value: 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC2CE	OC2M			OC2PE	OC2FE	Reserved		OC1CE	OC1M			OC1PE	OC1FE	Reserved	
rw	rw			rw	rw			rw	rw			rw	rw		

Bit	Field	Description
15	OC2CE	Channel 2 output and compare clear enable Refer to OC1CE description.
14 : 12	OC2M	Channel 2 output and compare mode Refer to OC1M description.
11	OC2PE	Channel 2 output and compare preload enable Refer to OC1PE description.
10	OC2FE	Channel 2 output and compare quick enable Refer to OC1FE description.
9 : 8	Reserved	Reserved, must be kept at reset value.
7	OC1CE	Channel 1 compare clear enable 0: OC1REF is not affected by ETR 1: When detecting ETR as active level, clear OC1REF

Bit	Field	Description
6 : 4	OC1M	Channel 1 output compare mode The bit has defined the output reference signal OC1REF action. OC1REF has determined OC1 and OC1N value. OC1REF is active at high level. The active level of OC1, OC1N depends on CC1P and CC1NP bit. 000 : Freeze. TIM1_CCR1 and TIM1_CNT compare results has no effect on OC1REF. 001: Set as high when configuration. When TIM1_CNT value and TIM1_CCR1 value are same, enforce OC1REF as high level 010: Set as low when configuration. When TIM1_CNT value and TIM1_CCR1 value are same, enforce OC1REF as low level 011: Toggle when match. When TIM1_CCR1=TIM1_CNT, OC1REF toggle. 100: Enforce as low. Enforce OC1REF at low level 101: Enforce as high. Enforce OC1REF at high level 110 : PWM mode 1. During up count, in case of TIM1_CNT<TIM1_CCR1, enforce OC1REF is at high level. Or else, it's at low level. During down count, in case of TIM1_CNT > TIM1_CCR1, enforce OC1REF is at low level. Or else, it's at high level. 111 : PWM mode 2. During up count, in case of TIM1_CNT<TIM1_CCR1, channel 1 enforce OC1REF is at low level. Or else, it's at high level. During down count, in case of TIM1_CNT > TIM1_CCR1, enforce OC1REF is at high level. Or else, it's at low level. Note 1: In case of LOCK level 3 (TIM1_BDTR register LOCK bit), the bit cannot be changed. Note 2: In PWM mode 1 or PWM mode 2, only when the compare result changes or it changes over from the freeze mode to PWM mode in the output compare mode, OC1REF level may change.
3	OC1PE	Channel 1 output compare preload enable 0: Disable TIM1_CCR1 register preload function. The value written into TIM1_CCR1 register becomes active immediately 1: Enable TIM1_CCR1 register preload function. Only read and write the preload register. TIM1_CCR1 preload value becomes active at the update event Note 1: When the LOCK level is 3 (TIM1_BDTR register LOCK bit), the bit cannot be changed. Note 2: Only in the one-pulse mode (TIM1_CR1 register OPM= 1), it has no influence whether the preload register is set. Under other scenarios, it's required to set the preload register. Otherwise, the follow up action is not certain.
2	OC1FE	Channel 1 output compare quick enable In case of the bit is set 1, when the channel is configured as PWM mode, it speeds up response of the compare output to the trigger time. The output channel deems the active edge of the trigger input signal as one compare match. Therefore, OC is set as compare level, but is irrelevant to the compare result. 0: channel 1 output compare fast enable. 1: channel 1 output compare fast disable.
1 : 0	Reserved	Reserved, must be kept at reset value.

11.4.8 TIM1_CCMR2 Compare Mode Register 2

Address offset: 0x1C

Reset value: 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC4CE	OC4M			OC4PE	OC4FE	Reserved		OC3CE	OC3M			OC3PE	OC3FE	Reserved	
rw	rw			rw	rw			rw	rw			rw	rw		

Bit	Field	Description
15	OC4CE	Channel 4 output compare clear enable Refer to OC3CE description
14 : 12	OC4M	Channel 4 output compare mode Refer to OC3M description
11	OC4PE	Channel 4 output compare preload enable Refer to OC3PE description
10	OC4FE	Channel 4 output compare rapid enable Refer to OC3FE description
9 : 8	Reserved	Reserved, must be kept at reset value.

Bit	Field	Description
7	OC3CE	Channel 3 output compare clear enable 0: OC3REF is not affected by ETR input 1: When detecting ETR input is at active level, clear OC3REF
6 : 4	OC3M	Channel 3 output compare mode The bit has defined the output reference signal OC3REF action. OC3REF has determined OC3 and OC3N value. OC3REF is active at high level. The active level of OC3, OC3N depends on CC3P and CC3NP bit. 000 : Freeze. TIM1_CCR3 and TIM1_CNT compare results has no effect on OC3REF. 001: Set as high when configuration. When TIM1_CNT value and TIM1_CCR3 value are same, enforce OC3REF as high level 010: Set as low when configuration. When TIM1_CNT value and TIM1_CCR3 value are same, enforce OC3REF as low level 011: Toggle when match. When TIM1_CCR3=TIM1_CNT, OC3REF toggle. 100: Enforce as low. Enforce OC3REF at low level 101: Enforce as high. Enforce OC3REF at high level 110: PWM mode 1. During up count, in case of TIM1_CNT<TIM1_CCR3, enforce OC3REF as high level, otherwise its low level; during down count, in case of TIM1_CNT > TIM1_CCR3, enforce OC3REF as low level, otherwise its high level. 111: PWM mode 2. During up count, in case of TIM1_CNT<TIM1_CCR3, enforce OC3REF as low level, otherwise its high level; during down count, in case of TIM1_CNT > TIM1_CCR3, enforce OC3REF as high level, otherwise its low level. Note 1: In case of LOCK level 3 (TIM1_BDTR register LOCK bit), the bit cannot be changed. Note 2: In PWM mode 1 or PWM mode 2, only when the compare result changes or it changes over from the freeze mode to PWM mode in the output compare mode, OC3REF level may change.
3	OC3PE	Channel 3 output compare preload enable 0: Disable TIM1_CCR3 register preload function, the value written into TIM1_CCR3 register becomes active immediately 1: Enable TIM1_CCR3 register preload function. The read and write only works on the preload register. TIM1_CCR3 preload value becomes active when the update event is available Note 1: When the LOCK level is set 3 (TIM1_BDTR register LOCK bit), the bit cannot be changed. Note 2: Only in the one-pulse mode (TIM1_CR1 register OPM= 1), it's not required to set the preload register. Under other scenarios, it's required to set the preload register. Otherwise, the follow up action is not certain.
2	OC3FE	Channel 3 output compare quick enable In case of the bit 1, when the channel is configured as PWM mode, it speeds up response of the compare output to the trigger time. The output channel deems the active edge of the trigger input signal as one compare match. Therefore, OC is set as compare level, but is irrelevant to the compare result. 0: channel 3 output compare fast disable. 1: channel 3 output compare fast enable.
1 : 0	Reserved	Reserved, must be kept at reset value.

11.4.9 TIM1_CCER Compare Enable Register

Address offset: 0x20

Reset value: 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	CC4P	CC4E	CC3NP	CC3NE	CC3P	CC3E	CC2NP	CC2NE	CC2P	CC2E	CC1NP	CC1NE	CC1P	CC1E

rw

Bit	Field	Description
15 : 14	Reserved	Reserved, must be kept at reset value.
13	CC4P	Channel 4 compare output polarity Refer to CC1P description.
12	CC4E	Channel 4 compare enable Refer to CC1E description

Bit	Field	Description
11	CC3NP	Channel 3 complementary output polarity Refer to CC1NP description
10	CC3NE	Channel 3 complementary output enable Refer to CC1NE description
9	CC3P	Channel 3 output polarity Refer to CC1P description
8	CC3E	Channel 3 output enable Refer to CC1E description
7	CC2NP	Channel 2 complementary output polarity Refer to CC1NP description
6	CC2NE	Channel 2 complementary output enable Refer to CC1NE description
5	CC2P	Channel 2 output polarity Refer to CC1P description
4	CC2E	Channel 2 output enable Refer to CC1E description
3	CC1NP	Channel 1 complementary output polarity This bit has defined the input signal polarity: 0: OC1N is active at high level 1: OC1N is active at low level Note: When LOCK level (TIM1_BDTR register LCKK bit) is set 3 or 2, the bit cannot be changed.
2	CC1NE	Channel 1 complementary output enable 0: Close channel 1 complementary output. OC1N output disable. 1: OC1N signal exports to the corresponding output pin. The output level relies on the values in MOE, OSSR, OSSI, OIS1, OIS1N and CC1E.
1	CC1P	Channel 1 output polarity This bit defines the output signal polarity: 0: OC1 is active at high level 1: OC1 is active at low level Note: When LOCK level (TIM1_BDTR register LCKK) is set 3 or 2, this bit cannot be changed.
0	CC1E	Channel 1 output enable 0: Close. OC1 output disable 1: Enable. The output level relies on the values in MOE, OSSR, OSSI, OIS1, OIS1N and CC1NE.

11.4.10 TIM1_CNT Counter

Address offset: 0x24

Reset value: 0x0000

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

CNT

rw

Bit	Field	Description
15 : 0	CNT	Counter value

11.4.11 TIM1_PSC Prescaler

Address offset: 0x28

Reset value: 0x0000

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

PSC

rw

Bit	Field	Description
15 : 0	PSC	Prescaler value Counter clock frequency (ck_cnt) = $f_{CK_PSC} / (PSC+1)$ In case of update event, PSC value is loaded into the current prescaler register.

11.4.12 TIM1_ARR Auto Reload Register

Address offset: 0x2C

Reset value: 0x0000

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

Bit	Field	Description
15 : 0	ARR	Auto reload value These bits define the auto reload value of the counter. When auto reload value is 0, the counter doesn't work.

11.4.13 TIM1_RCR Repeat Count Register

Address offset: 0x30

Reset value: 0x0000

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

Bit	Field	Description
15 : 8	REP_CNT	Real-time value written by the repeat counter In the repeat count mode, it's to change real-time phase move of the detection point of update interrupt flag bit (UIF) Note: After the update event, write these bits. Write REP_CNT before the update event will make the bit move inactive.
7 : 0	REP	Repeat counter value The repeat counter value defines the generation speed of the update event. When the repeat counter value reduces to 0, it generate the update event. If the update interrupt is enabled, it will simultaneously affect the update interrupt generation speed. The written REP value becomes active at the next update event. In PWM mode, (REP+1) corresponds to: In the edge alignment mode, the number of PWM period In the central alignment mode, the number of PWM half period

11.4.14 TIM1_CCR1 Compare Register 1

Address offset: 0x34

Reset value: 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR1															
rw															

Bit	Field	Description
15 : 0	CCR1	Channel 1 compare value If the preload function is not selected in TIM1_CCMR1 register (OC1PE bit), the write value is immediately transmitted to the current compare shadow register. Or else, the write value is only loaded into the compare register in case of update event. The current compare shadow register is involved in the compare with the counter TIM1_CNT, and the compare result is reflected to the output signal of OC1 port.

11.4.15 TIM1_CCR2 Compare Register 2

Address offset: 0x38

Reset value: 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR2															
rw															

Bit	Field	Description
15 : 0	CCR2	Channel 2 compare value Refer to CCR1 description.

11.4.16 TIM1_CCR3 Compare Register 3

Address offset: 0x3C

Reset value: 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR3															
rw															

Bit	Field	Description
15 : 0	CCR3	Channel 3 compare value Refer to CCR1 description.

11.4.17 TIM1_CCR4 Compare Register 4

Address offset: 0x40

Reset value: 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR4															
rw															

Bit	Field	Description
15 : 0	CCR4	Channel 4 compare value Refer to CCR1 description.

11.4.18 TIM1_BDTR Break and Dead-Time Register

Address offset: 0x44

Reset value: 0x0000 0000

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

Note: According to the lock setting, DOE, AOE, BKP, BKE, OSSR, OSSR and DTG bits can be write protected; it's necessary to configure them during the first write into TIM1_BDTR register. Details are given in the chapter of complementary output and dead-time insert.

Bit	Field	Description
31 : 17	Reserved	Reserved, must be kept at reset value.
16	DOE	Direct output enable When the break is active, MOE is set 0, active. 0 : After break output, wait for a dead time and export the idle state (output disable). 1 : Immediately export the idle state (output disable). Note: When LOCK level 1 (TIM1_BDTR register LOCK bit) is set, the bit cannot be changed.
15	MOE	Main output enable When the channel x is configured as output, according to AOE bit setting value, the bit can be cleared or auto set by software. When the break input is active, the bit is cleared by hardware. 0 : Disable OCx and OCxN output or enforce as the idle state (output disable). 1 : When setting the corresponding enable bit (TIM1_CCER register CCxE, CCxNE), OCx and OCxN output enable
14	AOE	Auto output enable 0: MOE cannot be set by hardware 1: MOE can be set by software or auto set by hardware at the next update event when the break is inactive Note: When LOCK level 1 (TIM1_BDTR register LOCK bit) is set, the bit cannot be changed.
13	BKP	Break input polarity 0: Break input is active low 1: Break input is active high Note: When LOCK level 1 (TIM1_BDTR register LOCK bit) is set, the bit cannot be changed.
12	BKE	Break function enable 0 : Break input disabled 1 : Break input enabled Note 1: When LOCK level 1 (TIM1_BDTR register LOCK bit) is set, the bit cannot be changed. Note 2: Break input includes pin input and comparator compare result input. Before break function is enabled, configure BKIN_SEL bit in TIM1_BKINF register to select break source.
11	OSSR	Off state selection in the run mode This bit only applies in case of MOE =1 and the channel is in complementary output. 0: When the timer inactive, disable OC/OCN output 1: When the timer inactive, in case of CCxE = 1 or CCxNE = 1, first enable OC/OCN and export inactive level, and then set OC/OCN enable output signal Note: When LOCK level 2 (TIM1_BDTR register LOCK bit) is set, the bit cannot be changed.
10	OSSI	Off state selection in the idle mode This bit only applies in case of MOE =0 and the channel is in output. 0: When the timer inactive, disable OC/OCN output 1: When the timer inactive, in case of CCxE = 1 or CCxNE = 1, first OC/OCN exports inactive level, and then set OC/OCN enable output signal. Note: When LOCK level 2 (TIM1_BDTR register LOCK bit) is set, the bit cannot be changed.

Bit	Field	Description
9 : 8	LOCK	Lock configuration This bit defines the register write protection function. 00: Write protection function closes, the register doesn't have the write protection 01: Lock level 1, unable to write TIM1_BDTR register DOE, DTG, BKE, BKP and AOE bit and TIM1_CR2 register OISx/OISxN bit 10: Lock level 2, unable to write lock level 1 bit, or CC polarity level as well as OSSR/OSSI bit 11 : Lock level 3, unable to write lock level 2 bit, or CC control bit Note: The LOCK bits can be written only once after the reset. Once the TIM1_BDTR register has been written, their content is frozen until the next reset.
7 : 0	DTG	Dead-time generator setup adjustment These bits define the dead-time duration of the complementary output. Note: When LOCK level (TIM1_BDTR register LOCK bit) is set 1, 2 or 3, the bit cannot be changed.

11.4.19 TIM1_CCMR3 Compare Mode Register 3

Address offset: 0x54

Reset value: 0x0000

The channel only applies to the output compare mode

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												OC5PE	Reserved		
												rw			

Bit	Field	Description
15 : 4	Reserved	Reserved, must be kept at reset value.
3	OC5PE	Output compare 5 preload enable 0: Disable TIM1_CCR5 register preload function, and the value written into TIM1_CCR5 register will become effective immediately. 1: Enable TIM1_CCR5 register preload function. Only read and write the preload register. TIM1_CCR5 preload value becomes active at the update event Note 1: When LOCK level 3 is set (TIM1_BDTR register LOCK level), this bit cannot be changed.
2 : 0	Reserved	Reserved, must be kept at reset value.

11.4.20 TIM1_CCR5 Compare Register 5

Address offset: 0x58

Reset value: 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR5															
rw															

Bit	Field	Description
15 : 0	CCR5	Compare 5 value CC5 channel is only configured as output: If the preload function is not selected in the TIM1_CCMR3 register (OC5PE bit), the write value will immediately be transmitted to the corresponding compare shadow register. Or else, only in case of the update event, the preload value will be transmitted to the corresponding compare shadow register. The compare shadow register is involved in the counter TIM1_CNT compare. Because CC5 channel is the internal channel and cannot be transmitted to the pin, the compare result is used for the internal trigger event.

11.4.21 TIM1_PDER PWM Phase Shift Enable Register

Address offset: 0x5C

Reset value: 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		CCR5FALLRS	CCR4FALLRS	CCR3FALLRS	CCR2FALLRS	CCR1FALLRS	CNTFORCECL	Reserved		CCR5_SHIFT_EN	CCR4_SHIFT_EN	CCR3_SHIFT_EN	CCR2_SHIFT_EN	CCR1_SHIFT_EN	CCDREPE
		rw	rw	rw	rw	rw	rw			rw	rw	rw	rw	rw	rw

Bit	Field	Description
15 : 14	Reserved	Reserved, must be kept at reset value.
13	CCR5FALLRS	When PWM phase-shift output is enabled for Channel 5, triggering based on CCR5FALL is permitted during the up-counting phase. 0: Disable CCR5FALL trigger during the up-counting phase. 1: Enable CCR5FALL trigger during the up-counting phase. For details, refer to the CCRxFALL register description regarding the phase-shifting operation.
12	CCR4FALLRS	When PWM phase-shift output is enabled for Channel 4, triggering based on CCR4FALL is permitted during the up-counting phase. 0: Disable CCR4FALL trigger during the up-counting phase. 1: Enable CCR4FALL trigger during the up-counting phase. For details, refer to the CCRxFALL register description regarding the phase-shifting operation.
11	CCR3FALLRS	When PWM phase-shift output is enabled for Channel 3, triggering based on CCR3FALL is permitted during the up-counting phase. 0: Disable CCR3FALL trigger during the up-counting phase. 1: Enable CCR3FALL trigger during the up-counting phase. For details, refer to the CCRxFALL register description regarding the phase-shifting operation.
10	CCR2FALLRS	When PWM phase-shift output is enabled for Channel 2, triggering based on CCR2FALL is permitted during the up-counting phase. 0: Disable CCR2FALL trigger during the up-counting phase. 1: Enable CCR2FALL trigger during the up-counting phase. For details, refer to the CCRxFALL register description regarding the phase-shifting operation.
9	CCR1FALLRS	When PWM phase-shift output is enabled for Channel 1, triggering based on CCR1FALL is permitted during the up-counting phase. 0: Disable CCR1FALL trigger during the up-counting phase. 1: Enable CCR1FALL trigger during the up-counting phase. For details, refer to the CCRxFALL register description regarding the phase-shifting operation.
8	CNTFORCECL	When PWM phase-shift is enabled for channel x, the counter is forced to zero upon an overflow. 0: Disable counter reset on overflow. 1: Enable counter reset on overflow.
7 : 6	Reserved	Reserved, must be kept at reset value.
5	CCR5_SHIFT_EN	Enable channel 5 output PWM phase shift bit 0: Channel 5 output PWM phase shift disable 1: Channel 5 output PWM phase shift enable Details given in CCRxFALL register phase shift operation
4	CCR4_SHIFT_EN	Enable channel 4 output PWM phase shift bit 0: Channel 4 output PWM phase shift disable 1: Channel 4 output PWM phase shift enable Details given in CCRxFALL register phase shift operation
3	CCR3_SHIFT_EN	Enable channel 3 output PWM phase shift bit 0: Channel 3 output PWM phase shift disable 1: Channel 3 output PWM phase shift enable Details given in CCRxFALL register phase shift operation

Bit	Field	Description
2	CCR2_SHIFT_EN	Enable channel 2 output PWM phase shift bit 0: Channel 2 output PWM phase shift disable 1: Channel 2 output PWM phase shift enable Details given in CCRxFALL register phase shift operation
1	CCR1_SHIFT_EN	Enable channel 1 output PWM phase shift bit 0: Channel 1 output PWM phase shift disable 1: Channel 1 output PWM phase shift enable Details given in CCRxFALL register phase shift operation
0	CCDREPE	Enable DMA to generate the update request at each underflow or overflow 0 : DAM update request is generated according to the repeat count register value. 1 : Enable DMA to generate the update request at each underflow or overflow

11.4.22 TIM1_CCRxFALL PWM Phase Shift Count Down Compare Register

Address offset: 0x60 ~ 0x70

Reset value: 0x0000

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

Bit	Field	Description
15 : 0	CCRxFALL	Channel x compare value during count down in PWM central alignment mode PWM phase shift function: Enable PDER register PWM phase shift. According to the required phase shift, configure CCRxFALL and CCRx, PWM exports programmable phase shift waveform, and can shift left or right. Note: x=1~5

11.4.23 TIM1_BKINF Break Input Filter Register

Address offset: 0x74

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	COMPBKIN_SEL	Reserved						IOBKIN_SEL	Res.	BKINF				BKINF
		rw							rw		rw				rw

Bit	Field	Description
31 : 14	Reserved	Reserved, must be kept at reset value.
13	COMPBKIN_SEL	COMP break input channel select COMPBKIN_SEL[0] : 0: COMP1_OUT break input disable 1: COMP1_OUT break input enable
12 : 7	Reserved	Reserved, must be kept at reset value.
6	IOBKIN_SEL	IO break input channel select IOBKIN_SEL[0]: 0: TIM1_BKIN1 break input disable 1: TIM1_BKIN1 break input enable
5	Reserved	Reserved, must be kept at reset value.

Bit	Field	Description
4 : 1	BKINF	Break input filter 0000: 2 cycles 0001: 4 cycles 0010: 8 cycles 0011: 16 cycles 0100: 32 cycles 0101: 64 cycles 0110: 128 cycles 0111: 256 cycles 1000: 384 cycles 1001: 512 cycles 1010: 640 cycles 1011: 768 cycles 1100: 896 cycles 1101: 1024 cycles 1110: 1152 cycles 1111: 1280 cycles Note: Before changing the sampling frequency of break digital filter, turn off the break filter function.
0	BKINFE	Break input filter enable 1: Break input filter enable 0: Break input filter disable Note: Enable the filter after the break condition is configured. Break filter is used for level active signal.

11.4.24 TIM1_DOCR Debug Mode Output Control Register

Address offset: 0x80

Reset value: 0x0000

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

Bit	Field	Description
15 : 2	Reserved	Reserved, must be kept at reset value.
1 : 0	DOS	Debug halt Output Select 00: Debug halt is active; OCx/OCxN maintains the output value it held prior to entering debug mode. (No impact on the original output.). 01: Debug halt is active; OCx/OCxN outputs a low level. 10: Debug halt is active; OCx outputs a high level, and OCxN outputs a low level. 11: When debug halt is active and a brake event is triggered, the OCx/OCxN outputs drive to the levels defined by the break function settings.

11.4.25 TIM1_SOER Software Output Global Enable Register

Address offset: 0x84

Reset value: 0x0000

5	1	4	1	3	1	2	1	1	0	1	9	8	7	6	5	4	3	2	1	0
Reserved																				S
																				OEN
																				rw

Bit	Field	Description
15 : 1	Reserved	Reserved, must be kept at reset value.
0	SOEN	Software Output Global Enable 0: Disable the software-configured output function for CH1–CH4. 1: Enable the software-configured output function for CH1–CH4.

11.4.26 TIM1_SOCR Software Output Configuration Register

Address offset: 0x88

Reset value: 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BSFTEN								BSFTV							
rw								rw							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SFTEN								SFTV							
rw								rw							

Bit	Field	Description
31 : 24	BSFTEN	CH1~CH4 Break Software Output Enable 0: Disable OCx/OCxN Software Output Configuration. 1: Enable OCx/OCxN Software Output Configuration. X=1~4, BSFTEN[6:0] correspond to OC4, OC3N, OC3, OC2N, OC2, OC1N, and OC1, respectively.
23 : 16	BSFTV	CH1~CH4 Break Software Output Value 0: OCx/OCxN Software output setting 0. 1: OCx/OCxN software output setting 1. X=1~4, BSFTV[6:0] correspond to OC4, OC3N, OC3, OC2N, OC2, OC1N, and OC1, respectively.
15 : 8	SFTEN	CH1~CH4 Software Output Enable 0: Disable OCx/OCxN Software Output Configuration. 1: Enable OCx/OCxN Software Output Configuration. X=1~4, SFTEN[6:0] correspond to OC4, OC3N, OC3, OC2N, OC2, OC1N, and OC1, respectively.
7 : 0	SFTV	CH1~CH4 Break Software Output Value 0: OCx/OCxN Software output setting 0. 1: OCx/OCxN software output setting 1. X=1~4, SFTV[6:0] correspond to OC4, OC3N, OC3, OC2N, OC2, OC1N, and OC1, respectively.

11.4.27 TIM1_BKSR Break status register

Address offset: 0x8C

Reset value: 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													BCPTF0	BIOF	Res.
													r_w0c	r_w0c	

Bit	Field	Description
15 : 3	Reserved	Reserved, must be kept at reset value.
2	BCPTF0	CMP0 Break input status flag 0: No CMP0 break input. 1: CMP0 break input occurred. When a break input signal is received, the bit is set to 1 by hardware and cleared by software.

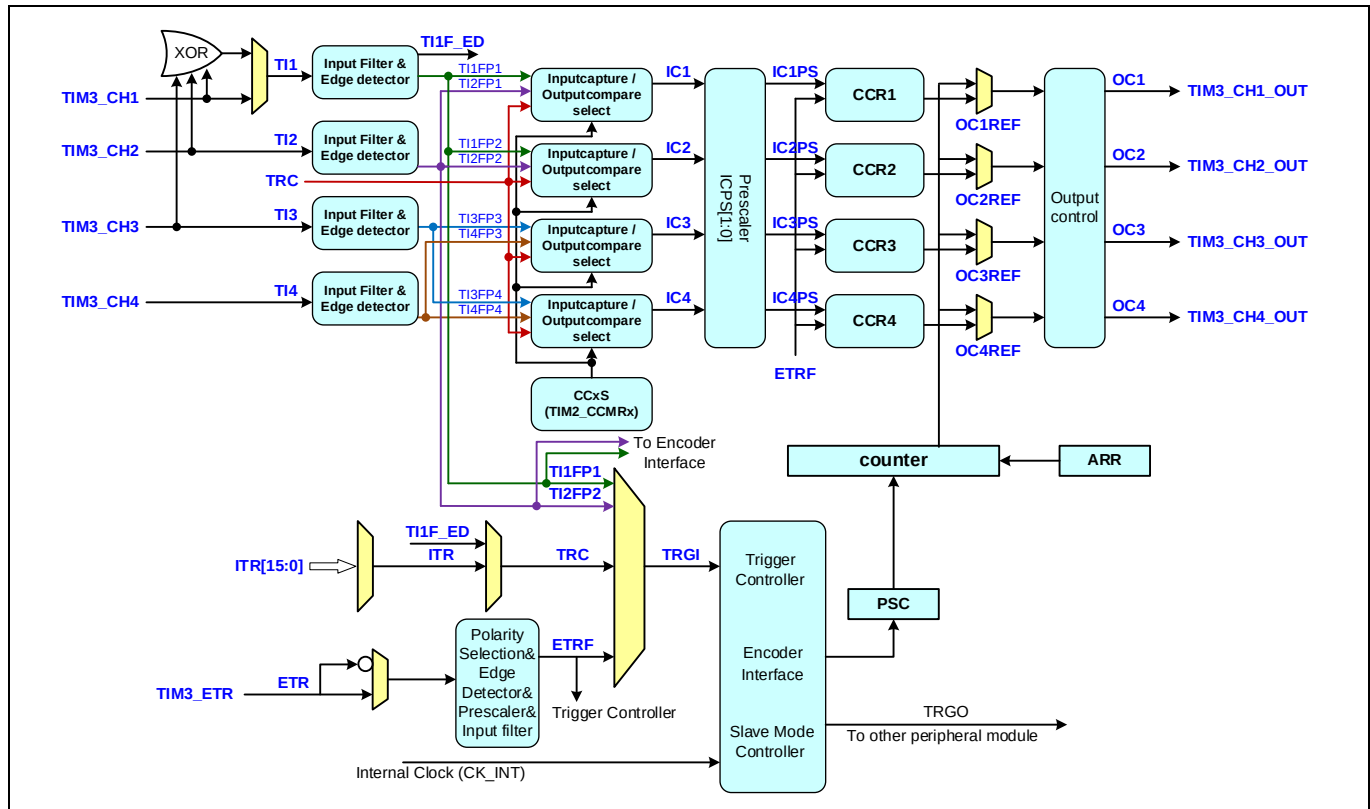
Bit	Field	Description
1	BIOF	IO Break input status flag 0: No IO break input. 1: IO break input occurred. When a break input signal is received, the bit is set to 1 by hardware and cleared by software.
0	Reserved	Reserved, must be kept at reset value.

12 TIM3 16-Bit General-purpose Timer Overview

TIM3 is composed of a 16 bit real-time programmable prescaler and a 16-bit auto reload counter with the programmable counting direction, and can conveniently provide with the counting and timing function. The counter is driven by a programmable prescaler. The general-purpose timer can be used for a variety of purposes, such as input capture function (measuring input signal pulse width or frequency, PWM input, etc), output compare function (PWM output, one-pulse mode, etc.)

12.2 Function block diagram

Figure 12-1 TIM3 block diagram



TIM3 structure block diagram is mainly composed of input capture unit, output compare unit, and counter unit.

12.3 Main characteristics

- 16 bit real-time programmable prescaler, division factor: 1-65536.
- Clock source: Internal clock source, external clock input (TIMx, ETRx), internal trigger input (ITRx).
- 16 bit auto reload counter (counting direction: up, down, up/down).
- Synchronization circuit to control the timer with external signals and to interconnect several timers to each other.
- Input capture: measure input signal pulse width or period.
- Trigger input can serve as external clock or be managed cycle-by-cycle.
- Support interfaces such as encoder and hall-sensor.
- 4 output channels
- Output compare (control output waveform or indicator counter already finishes timing).
- PWM output (edge alignment or central alignment mode)
- One-pulse mode.
- Interrupt request event: update event, trigger event, input capture, output compare.

12.4 Function description

12.4.1 Clock

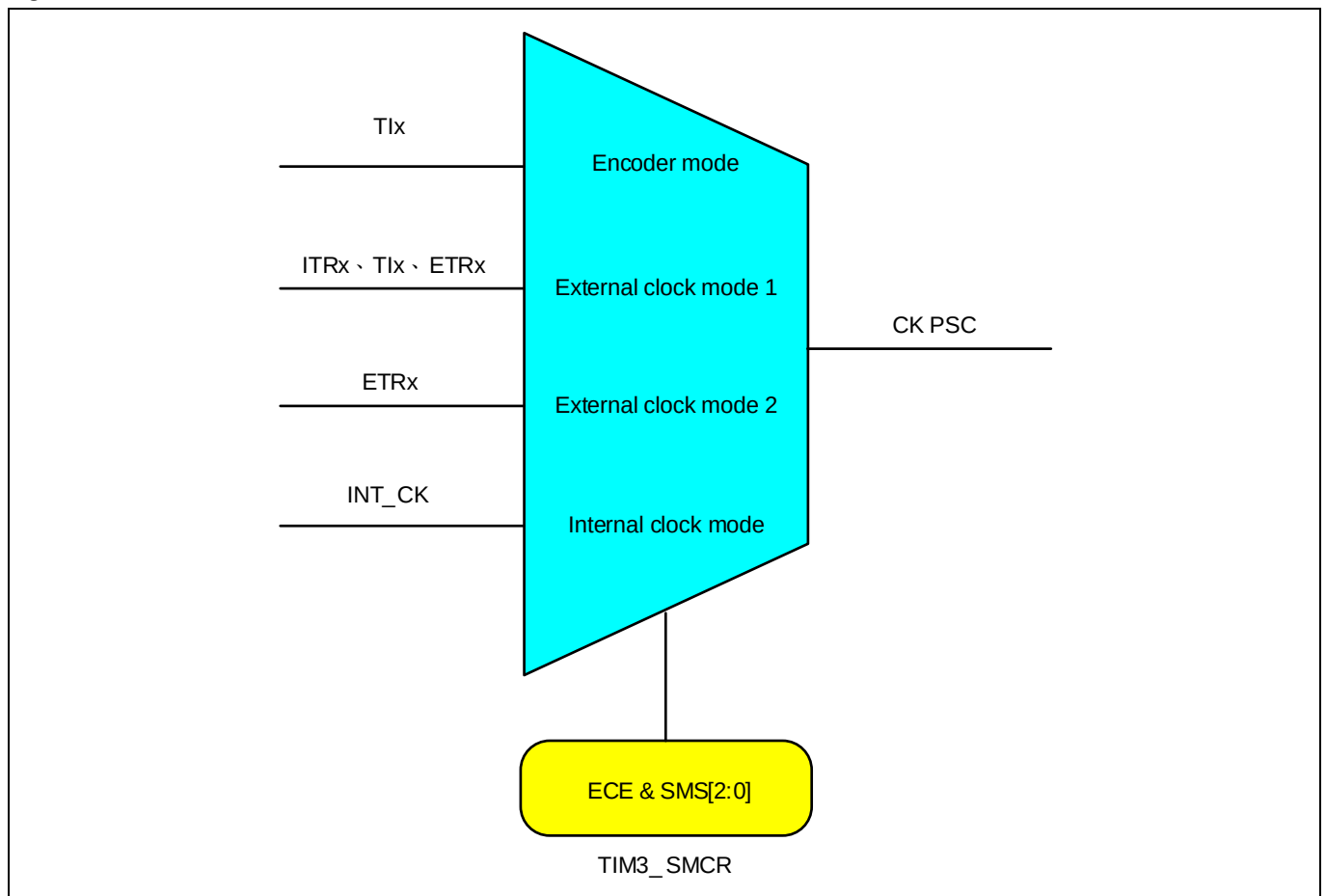
12.4.1.1 Clock selection

Several types of clock sources of the counter below:

- Internal clock (INT_CK)
- External clock mode 1: External trigger input TRGI (including TIx, ITRx, ETRx)
- External clock mode 2: External trigger input ETR (including ETRx)
- Encoder mode

The diagram of several types of clock choice above:

Figure 12-2 Clock selection



12.4.1.1.1 Internal clock source (INT_CK)

In case of SMS = 000 of TIM3_SMCR register, disable the slave mode. Enable the counter, and the prescaler clock is driven by internal clock.

12.4.1.1.2 External clock mode 1 (external trigger input TRGI, including TIx, ITRx, ETRx)

In case of SMS = 111 of TIM3_SMCR register, select the external clock mode 1 (TRGI). The counter is driven by the rising edge or falling edge of the selected input signal.

For example: count up at the rising edge on TI1 input. Specific configuration:

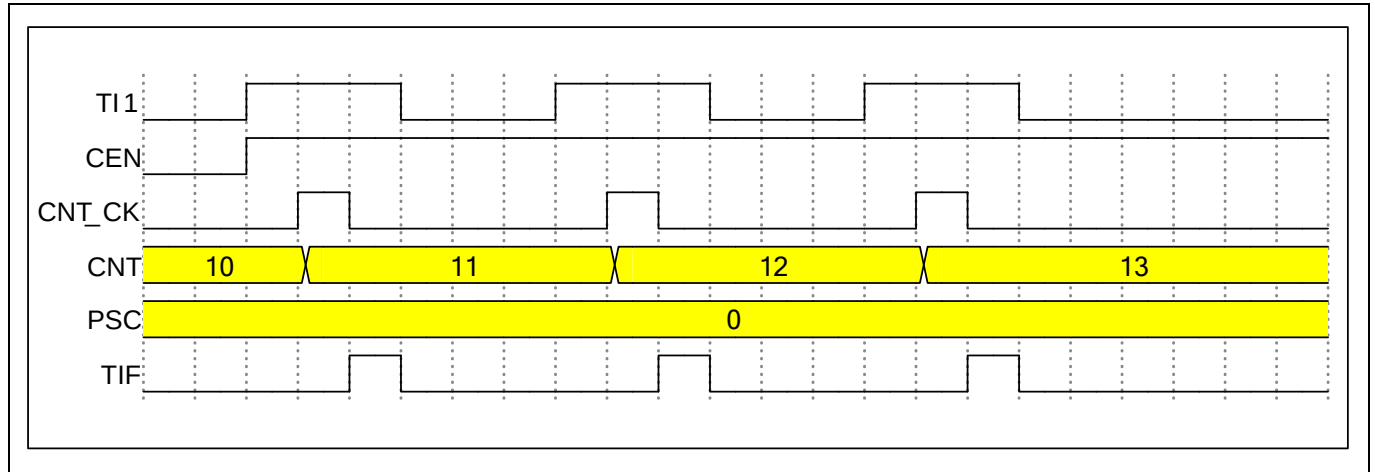
1. Configure TIM3_CCMR1 register CC1S=01, CC1 channel is configured as input, and IC1 is mapped on TI1; Configure TIM3_CCMR1 register IC1F[3 : 0], set input filter bandwidth; configure TIM3_CCER register CC1P=0, and select the rising edge as the active edge.

2. Configure TIM3_SMCR register TS=101, select TI1 as the trigger input source; configure TIM3_SMCR register SMS=111, select the external clock mode 1.

3. Configure TIM3_CR1 register DIR=0, select up counting mode; configure TIM3_CR1 register CEN=1 and start the counter.

TI1 input the active edge, count up once and TIF flag bit is set by hardware. The delay between the active edge of TI1 and actual clock of counter depends on the TI1 input pin sync circuit.

Figure 12-3 Control circuit in external clock mode 1



12.4.1.1.3 External clock mode 2(external trigger input ETR, including ETRx)

In case of TIM3_SMCR register ECE=1, enable external clock mode 2, and the counter is driven by the active edge of ETR signal.

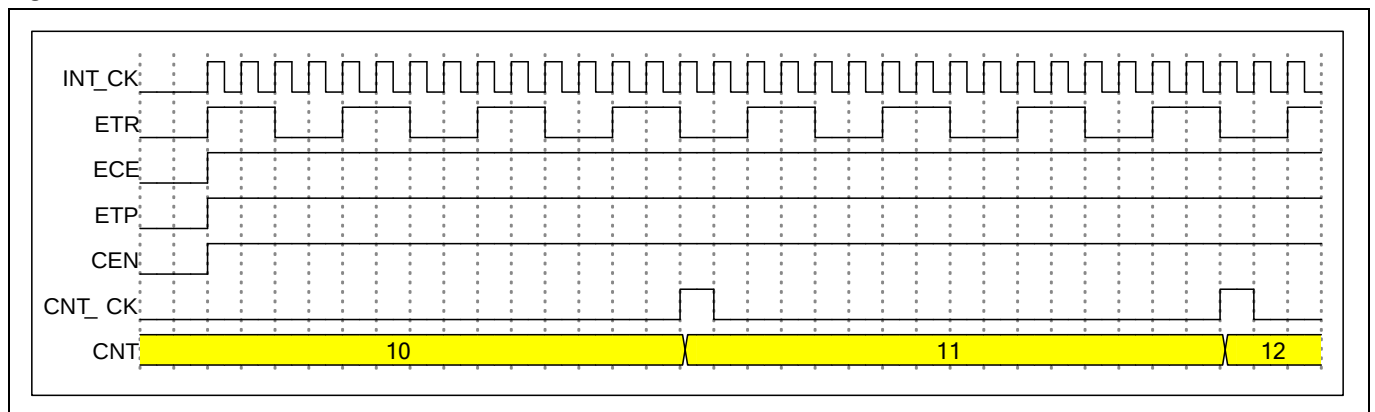
For example: count up once of each 4 falling edge on ETR Specific configuration:

1. Configure TIM3_SMCR register ETF [3:0] =0010, count up once of each 4 active edges on ETR; configure TIM3_SMCR register ETP=1, select falling edge as active edge; configure TIM3_SMCR register ECE=1, select external clock mode 2.

2. Configure TIM3_CR1 register DIR=0, select up counting mode; Configure TIM3_CR1 register CEN=1, start the counter.

The delay between the falling edge of ETR and actual clock of counter depends on the sync circuit of ETR signal terminal.

Figure 12-4 Control circuit in external clock mode 2



12.4.1.1.4 Encoder mode

Refer to the chapter of slave mode-encoder interface for details.

12.4.1.2 Time base unit

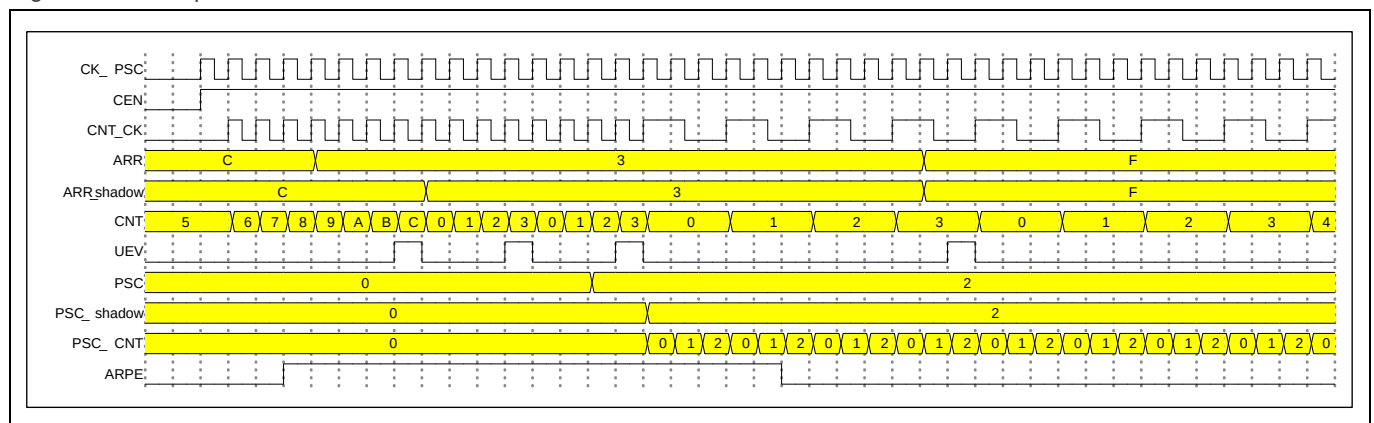
TIM3 time base unit mainly includes: Counter register (TIM3_CNT), prescaler register (TIM3_PSC), auto reload register (TIM3_ARR).

The counter unit is composed of a 16 bit counter with its related auto reload register, the counter can count up, down or both up and down.

The counter clock is provided by prescaler. The prescaler is composed of the prescaler counter with its related register. The division factor is 1-65536. It can write any time, and gets active at the next update.

The auto preload register has a 16 bit shadow register with preload function. Set TIM3_CR1 register ARPE bit to select the content of ARR register are transferred into the shadow register permanently or at each update event.

Figure 12-5 Auto preload



12.4.1.3 Counter modes

Configure DIR bit and CMS bit of TIM3_CR1 register to select the count mode. There are three count modes, up counting mode, down counting mode and central alignment mode (up/ down counting mode). Detailed descriptions are provided for each count mode.

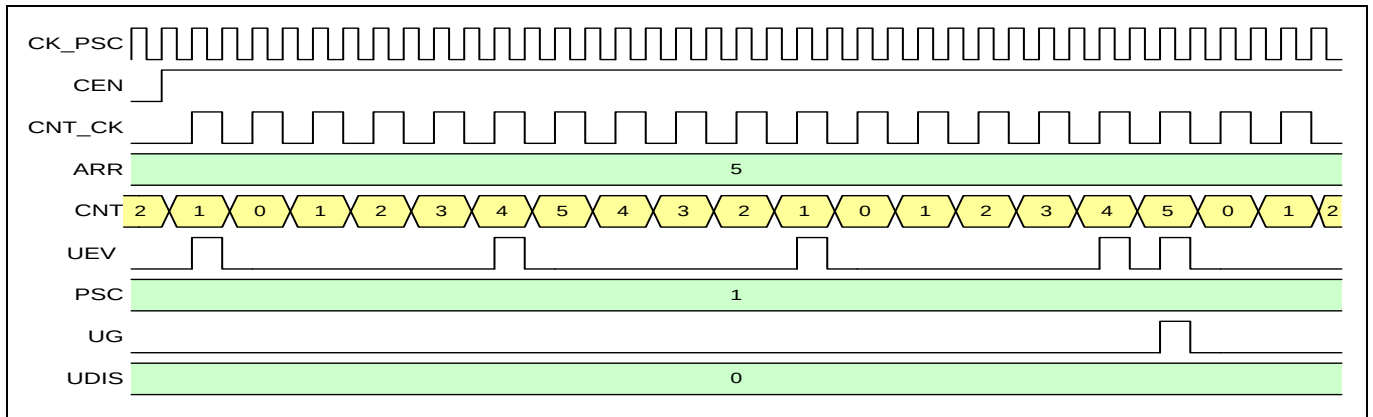
12.4.1.3.1 Up counting mode

Configure TIM3_CR1 register CMS=0, DIR=0, select up counting mode.

In up counting mode, enable TIM3_CR1 register CEN, and the counter begins up count from 0 to TIM3_ARR value and generates a counter overflow event (update). The counter begins up count again from 0. Set TIM3_EGR register UG =1, also generate an update event.

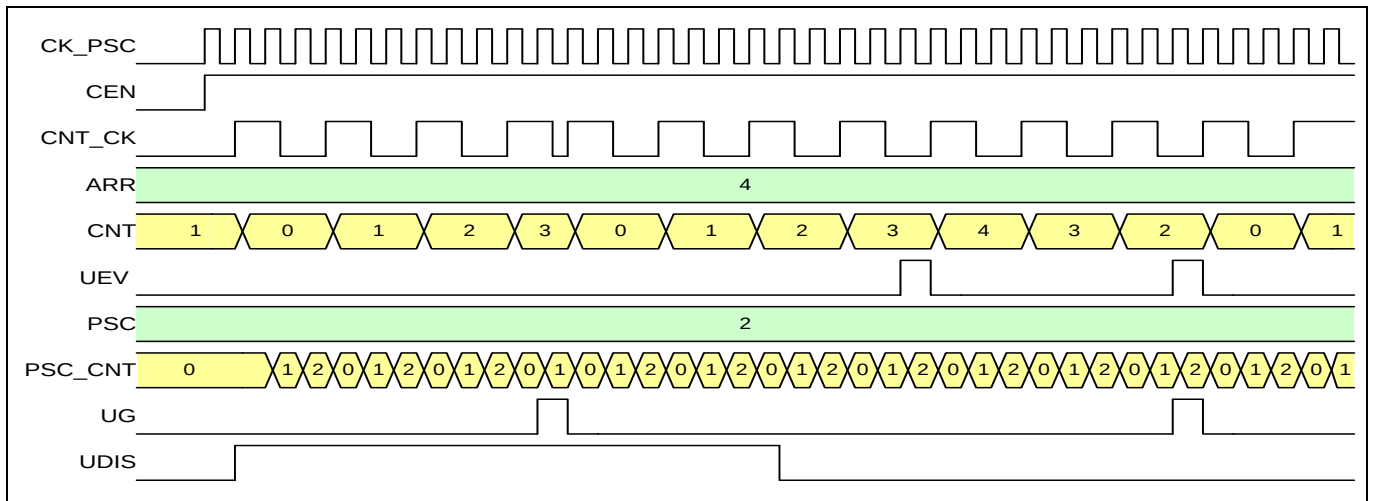
Figure 12-6 Up counting mode (UDIS=0)

Figure 12-10 Central count mode (UDIS=0)



When configuring TIM3_CR1 register UDIS=1, disable update event. When the counter generates the overflow or underflow event, it doesn't generate update event. Configuring UG=1 also doesn't generate update event. However, the counter and prescaler counter will initialize, and the count begins from 0.

Figure 12-11 Central count mode (UDIS=1 disables update event)

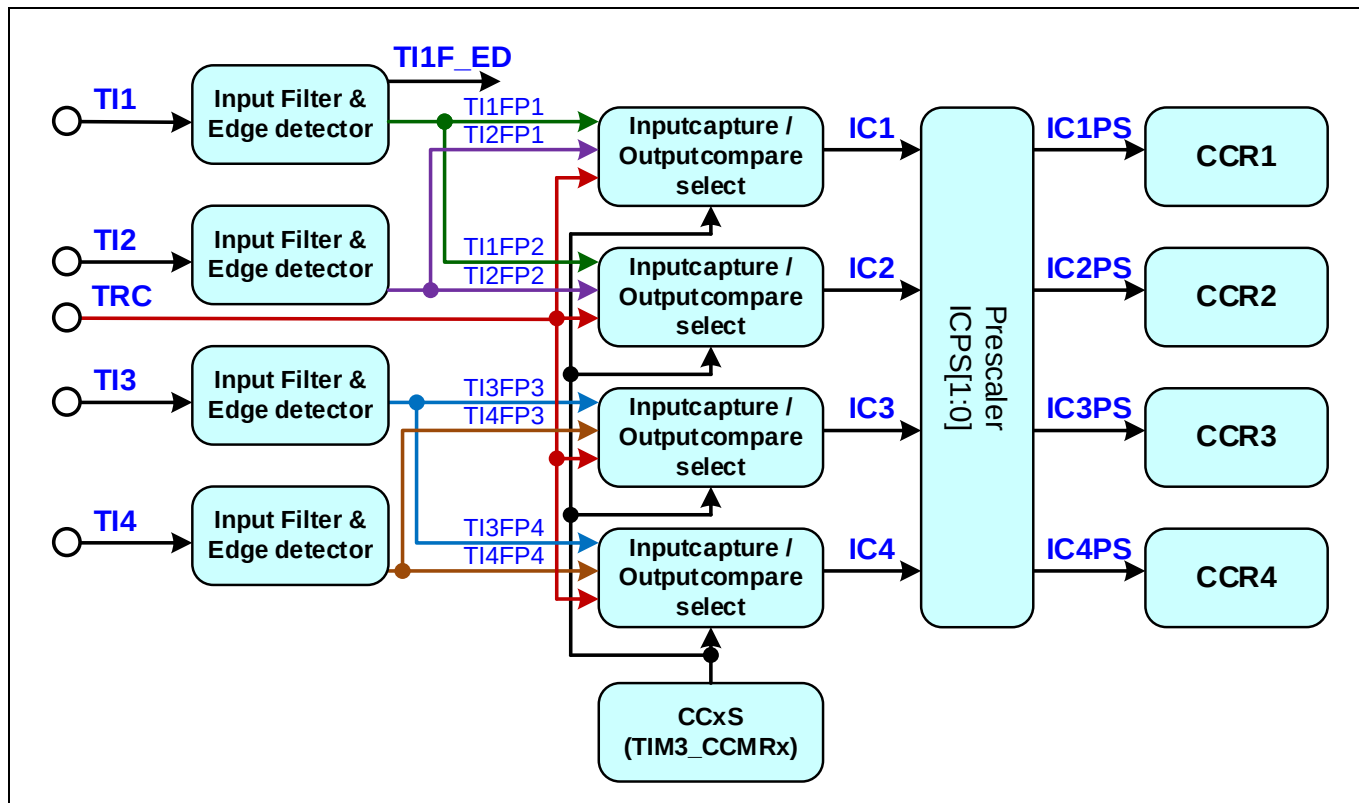


12.4.2 Input capture

12.4.2.1 Input capture

The input capture is composed of digital filter, multiplex, prescaler, etc. Its block diagram:

Figure 12-12 TIM3 input capture block diagram



Configure TIM3_CCMRx register ICxF, set the filter width of a digital filter (filter sampling frequency and digital filter width are shown in the table below). When the input signal width is greater than the filter width, the input signal is active. When the digital filter samples the input signal of the input pin TIx, generate a filtered signal TIxF. Through the edge detector with the optional polarity, generate an effective signal TIxFPx, this signal can serve as the trigger input signal of the slave mode counter. This signal generates ICxPS via the prescaler for trigger input capture event.

Table 12-1 Digital filter width and ICxF correlation

IC1F[3: 0]	Sample frequency and filter length	IC1F[3: 0]	Sample frequency and filter length
0000	No filter, sample by f_{DTS}	1000	Sampling frequency $f_{sampling}=f_{DTS}/8$, $N=6$
0001	Sampling frequency $f_{sampling}=f_{INT_CK}$, $N=2$	1001	Sampling frequency $f_{sampling}=f_{DTS}/8$, $N=8$
0010	Sampling frequency $f_{sampling}=f_{INT_CK}$, $N=4$	1010	Sampling frequency $f_{sampling}=f_{DTS}/16$, $N=5$
0011	Sampling frequency $f_{sampling}=f_{INT_CK}$, $N=8$	1011	Sampling frequency $f_{sampling}=f_{DTS}/16$, $N=6$
0100	Sampling frequency $f_{sampling}=f_{DTS}/2$, $N=6$	1100	Sampling frequency $f_{sampling}=f_{DTS}/16$, $N=8$
0101	Sampling frequency $f_{sampling}=f_{DTS}/2$, $N=8$	1101	Sampling frequency $f_{sampling}=f_{DTS}/32$, $N=5$
0110	Sampling frequency $f_{sampling}=f_{DTS}/4$, $N=6$	1110	Sampling frequency $f_{sampling}=f_{DTS}/32$, $N=6$
0111	Sampling frequency $f_{sampling}=f_{DTS}/4$, $N=8$	1111	Sampling frequency $f_{sampling}=f_{DTS}/32$, $N=8$

In the input capture mode, when detecting the active edge of signal ICx, the counter's current value is locked in the corresponding shadow register, and then loaded into the corresponding capture/compare register. When enabling interrupt or DMA, and producing the capture event, generate the corresponding interrupt or DMA request. At the capture event, the capture flag bit CCxIF of the status register (TIM3_SR) will be set. Configure CCxIF =0

or read TIM3_CCRx data, clear CCxIF flag bit. When CCxIF is not cleared, next input capture event generated, the repeat capture flag CCxOF will be set. Configure CCxOF =0, can clear CCxOF flag bit.

For example, Sample TI1 input signal's active edge. At TI1 rising edge, capture the current counter value and lock it in the TIM3_CCR1 register, procedure steps as follows:

1. Configure TIM3_CCMR1 register CC1S=01, CC1 channel is configured as input, and IC1 is mapped on TI1.
2. Configure TIM3_CCMR1 register IC1F [3:0], configure the filter width of digital filter (configure upon need).
3. Configure TIM3_CCER register CC1P=0, select capture occurs in the rising edge of TI1 signal.
4. Configure TIM3_CCMR1 register IC1PSC [1:0], select the prescaler factor.
5. Configure TIM3_CCER register CC1E = 1, channel 1 capture enable.
6. Configure TIM3_DIER register CC1IE=1, enable channel 1 interrupt request; if the chip has the built in DMA, configure CC1DE=1 of the TIMx_DIER register to allow the DMA request on capture/compare channel 1.

Note :

- When the channel is configured as input mode, TIM3_CCRx register becomes read only.
- In case of more than two continuous captures and CCxIF flag is not cleared, the repeat capture flag CCxOF is set. In order to avoid losing the capture information that may be generated before the repeat capture flag CCxOF is set. It is recommended to read data before reading the repeat capture flag.
- Set the corresponding CCxG bit of TIM3_EGR register, generate the input capture interrupt or DMA request through software.

12.4.2.2 PWM capture

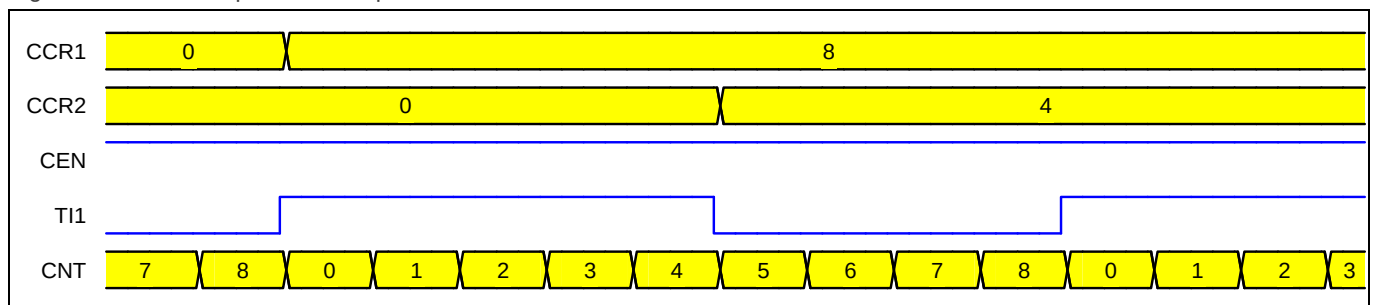
Differences of PWM input mode configuration with ordinary input capture:

- Two ICx signals are active on edges with opposite polarity and mapped on the same TIx input.
- Configure the slave mode counter at reset mode, set one TIxFP as trigger input signal.

For example: Measure TI1 PWM signal period (TIM3_CCR1 register) and duty cycle (TIM3_CCR2 register), the measuring value depends on the internal clock INT_Ck frequency and prescaler value. Specific steps:

1. Configure TIM3_CR1 register DIR =0, select counter as up counting mode.
2. Configure TIM3_CCMR1 register CC1S = 01, map IC1 on TI1, and select active input of TIM3_CCR1.
3. Configure TIM3_CCER register CC1P =0, select active polarity of TI1FP1 (active on rising edge) (capture the counter value to TIM3_CCR1, and clear the counter).
4. Configure TIM3_CCMR1 register CC2S =10, map IC2 on TI1, select active input of TIM3_CCR2.
5. Configure TIM3_CCER register CC2P =1, and select active polarity of TI2FP2 (active on falling edge) (capture counter value to TIM3_CCR2).
6. Configure TIM3_SMCR register TS = 101, select TI1FP1 as the active trigger input signal.
7. Configure TIM3_SMCR SMS = 100, set the slave mode counter as reset mode.
8. Configure TIM3_CCER register CC1E=1 and CC2E = 1. Enable CC1 channel and CC2 channel capture.

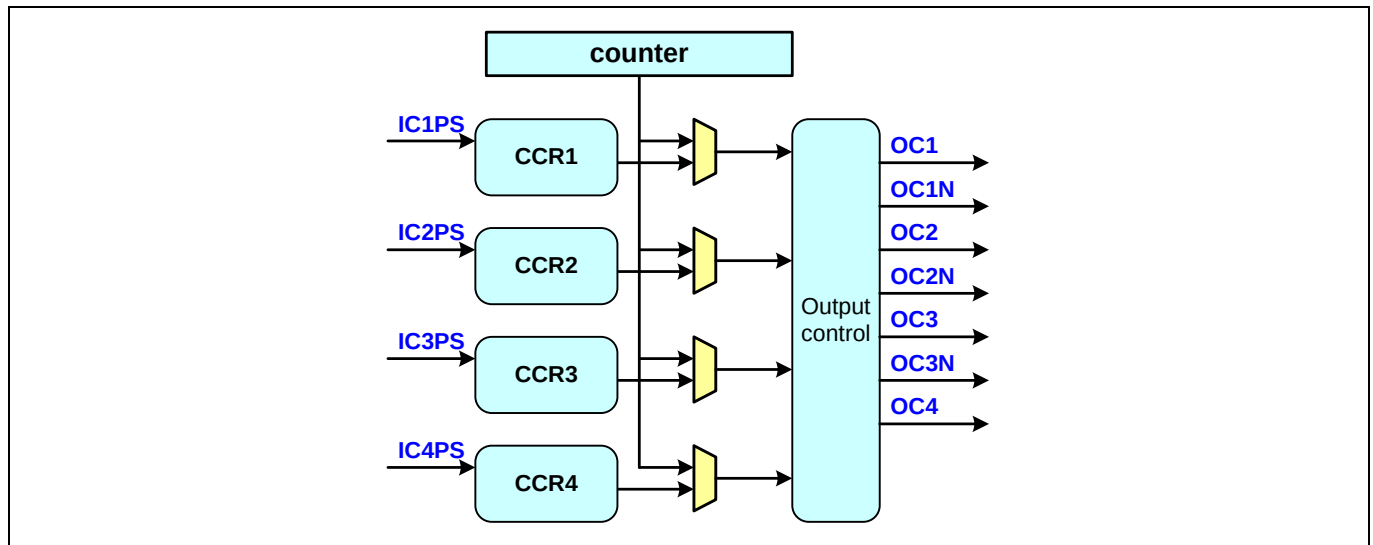
Figure 12-13 PWM input mode sequence



12.4.3 Output compare

The compare output of the capture/compare channel is composed of the comparator, output control circuit and capture/compare register. Its structure is shown below:

Figure 12-14 Output compare block diagram



In output compare mode, the content of capture/compare register is loaded into the shadow register and then compare the content of shadow register with the current value of counter. The capture/compare module includes a capture/compare register (preload register) and a shadow register. When reading and write, only programmed the capture/compare register.

12.4.3.1 Force output

Configure TIM3_CCMRx register CCxS = 00 and set the channel CCx as output mode. Configure TIM3_CCMRx register OCxM bit, and directly enforce the output compare signal as active or inactive level, independently of the output compare result. Configure TIM3_CCMRx register OCxM = 100, and enforce output compare signal as inactive level. And OCxREF is enforced at a low level. Configure TIM3_CCMRx register OCxM = 101, and enforce output compare signal as active level. And OCxREF is enforced as high level (OCxREF always active as high level).

Note: In force output mode, TIM3_CCRx shadow register and counter output compare is still performed. The corresponding flag bit of compare result will also be changed. When enable the corresponding interrupt and DMA request, it will also generate the corresponding interrupt and DMA request.

12.4.3.2 Output compare

In the output compare mode, when the values of counter and capture compare register matches, different waveform can be exported according to the OCxM bit of TIM3_CCMRx register.

For example, when the value of counter and capture/compare register matches, the functions in the output compare mode:

1. When compare match, OCxM value is different, output channel x signal OCx run differently:
 - ◆ OCxM = 000: OCx signal keeps its level.
 - ◆ OCxM = 001: OCx signal is set as active level.
 - ◆ OCxM = 010: OCx signal is set as inactive level.
 - ◆ OCxM = 011: OCx signal toggle.
2. When match, set the flag bit of status register (CCxIF bit of TIM3_SR register).
3. Configure TIM3_DIER register CCxIE =1, generate an interrupt when match.

4. Configure TIM3_DIER register CCxDE =1, generate a DMA request when match (only suitable for the built in DMA product).

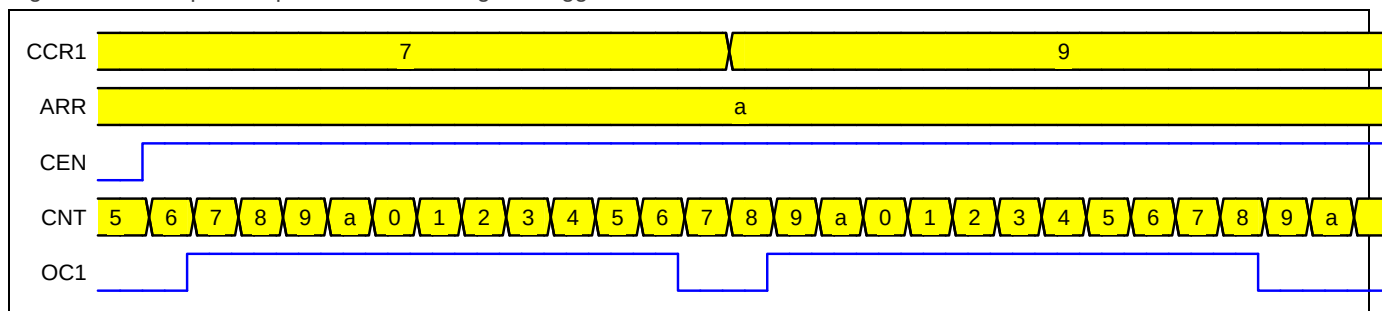
The output compare mode can also be used to export a single pulse (one-pulse mode).

Configuration steps of channel 1 output compare mode:

1. Configure counter clock (select clock source, configure prescaler factor).
2. Configure TIM3_ARR and TIM3_CCR1 register.
3. Configure TIM3_DIER register CC1IE =1, enable capture/compare 1 interrupt.
4. Configure output mode:
 - ◆ Configure TIM3_CCMR1 register OC1M = 011, toggle when OC1 compare match.
 - ◆ Configure TIM3_CCMR1 register OC1PE = 0, disable preload function of TIMx_CCR1 register.
 - ◆ Configure TIM3_CCMR1 register CC1P = 1, OC1 is active low polarity.
 - ◆ Configure TIM3_CCER register CC1E = 1, enable channel 1 output, OC1 signal is exported to the corresponding output pin.
5. Configure TIM3_CR1 register CEN =1, enable the counter.

Configure TIM3_CCMRx register OCxPE=0, disable TIM3_CCRx register preload function, write TIM3_CCRx register any time, and the write value updated immediately. Configure TIM3_CCMRx register OCxPE=1, and enable TIM3_CCRx register preload function, Updated only at the next update event. The figure below is an example.

Figure 12-15 Output compare mode, OC1 signals toggle when match



Note: In the output compare mode, the update event has no effect on output result. In the force output mode, the compare output between TIM3_CCRx shadow register and counter still performed. The corresponding flag bit of compare result is also be changed. When enabling the corresponding interrupt and DMA request, it still generates the corresponding interrupt and DMA request.

12.4.3.3 PWM output

In the PWM mode, according to the value of TIM3_ARR register and TIM3_CCRx register, generate a frequency and duty cycle controllable PWM waveform.

Configure OCxM=110 or OCxM=111 of TIM3_CCMRx register corresponding to the channel x, select channel x to PWM mode 1 or PWM mode 2. In the PWM mode, the counter will always be compared with CCRx. According to configuration and compare results, channel x exports different signals. Therefore, TIM3 can generate 4 PWM output signals with independent duty under the same frequency. In PWM mode, enable the preload function of TIM3_CCRx and TIM3_ARR. The written value of TIM3_CCRx preload register and TIM3_ARR preload register will become effective at the next update event, and load the corresponding shadow register. Therefore, in the PWM mode, before enabling counter, it's required to set TIM3_EGR UG=1, and generate the update event to initialize all registers.

Configure TIM3_CCER register CCxP, select OCx's active polarity. Configure TIM3_CCER register CCxE bit and control OCx output enable. Configure TIM3_CR1 register CMS bit, and select edge alignment or central alignment PWM signal.

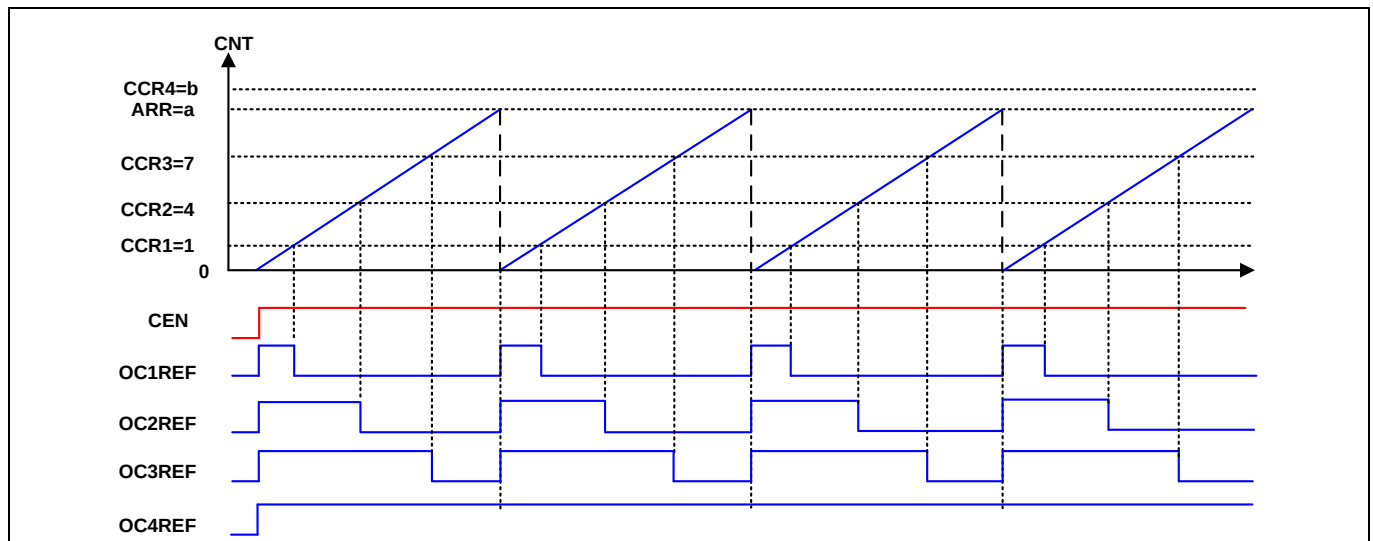
- CMS=00, edge alignment mode, configure DIR, select count mode (up or down).

- CMS=01, central alignment mode 1.
- CMS=10, central alignment mode 2.
- CMS=11, central alignment mode 3.

12.4.3.3.1 PWM edge alignment mode - up counting mode

Based on up counting mode configuration, configure TIM3_CCMRx register CCxS =00, select output mode, OCxM =110, select PWM mode 1. In the case of TIM3_CNT < TIM3_CCRx, channel x (OCxREF) is in active level. Otherwise, it's inactive level. If TIM3_CCRx compare value is greater than the auto reloads value (TIM3_ARR), OCxREF stays at an active level. When the value of the comparison is 0, OCxREF is at an inactive level. The figure below is the example of waveform in PWM mode 1 during edge alignment up count in case of CCR1 =1, CCR2 =4, CCR3 =7, CCR4 =b and ARR =a.

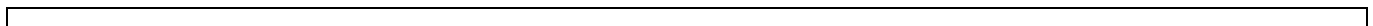
Figure 12-16 Waveform in PWM mode 1 during edge alignment up count

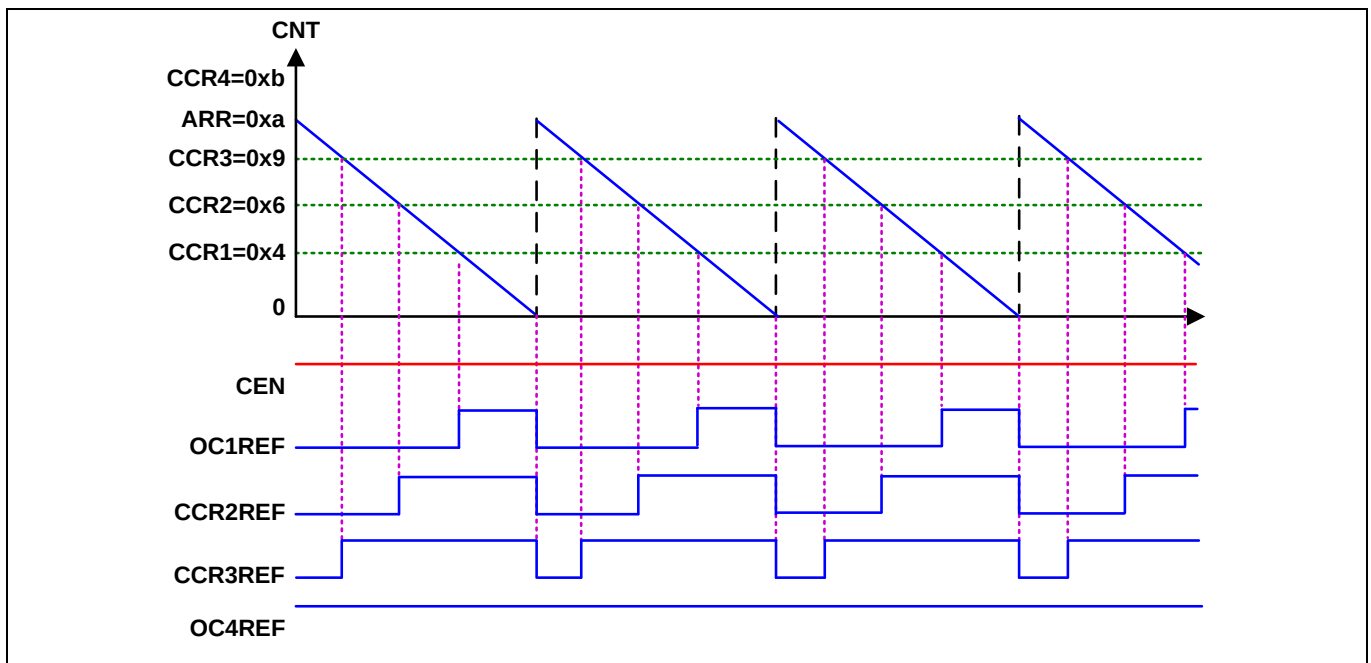


12.4.3.3.2 PWM edge alignment mode -down counting mode

Based on the down counting mode configuration, configure TIM3_CCMRx register CCxS=00, select output mode, OCxM=110, select PWM mode 1. In the case of TIM3_CNT > TIM3_CCRx, channel x (OCxREF) is at an inactive level. Or else, it's at an active level. The figure is an example of a waveform in PWM mode 1 during edge alignment down count in case of CCR1=4, CCR2=6, CCR3=9, CCR4=b and ARR=a.

Figure 12-17 Waveform in PWM mode 1 during edge alignment down count

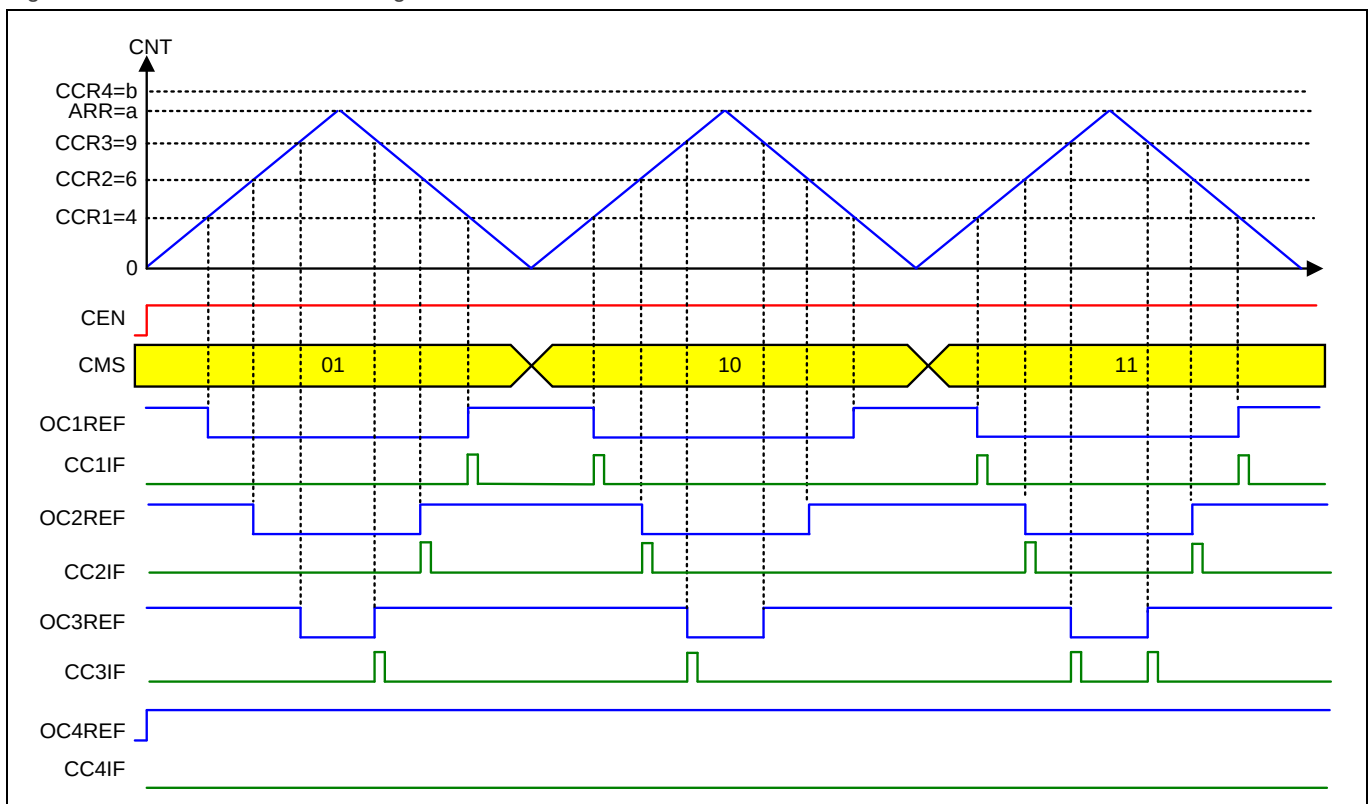




12.4.3.3.3 PWM central alignment mode

Configure TIM3 counter as central alignment count mode. Configure TIM3_CCMRx register CCxS=00 and select output mode. According to the CMS with different configurations, the output compare interrupt flag bit is set during up count (CMS =01), set during down count (CMS =10) or set during up or down count (CMS =11). The figure below is an example of waveform in central alignment PWM mode 1 (CMS =1) during CCR1=4, CCR2=6, CCR3=9, CCR4=b, ARR=a.

Figure 12-18 Waveform in central alignment PWM mode 1



Note:

- When entering central alignment mode, the current up/down counting configuration is used, and the counting direction depends on the current value of DIR.

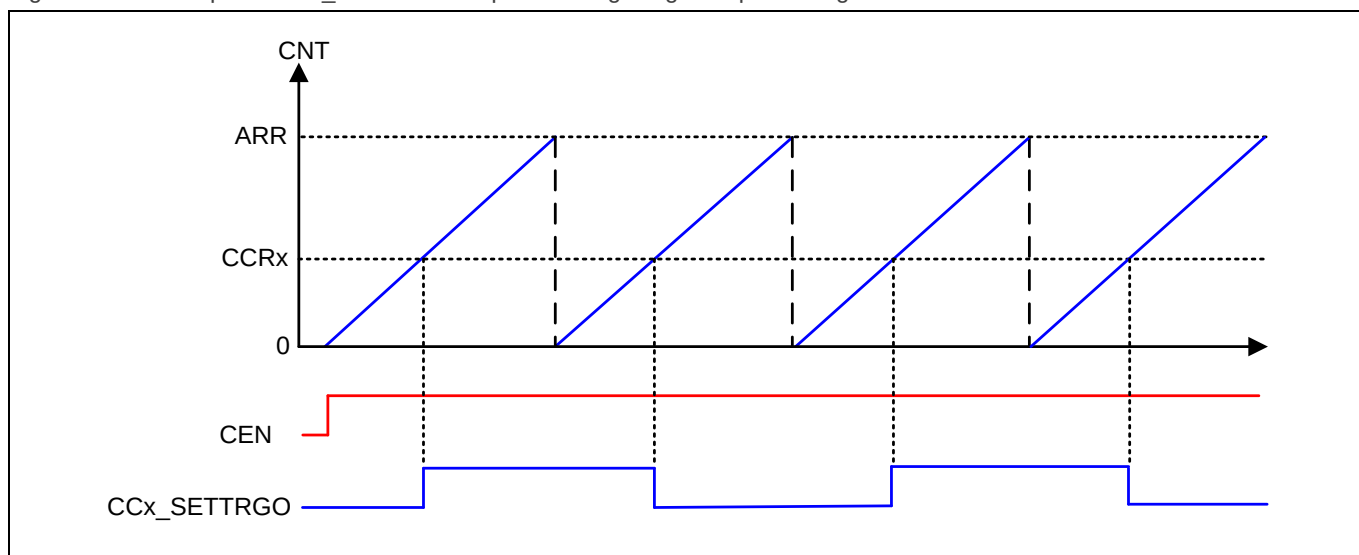
- In center alignment mode, it is best to avoid modifying the counter value, as this may lead to unpredictable results. When the counter is counting up, if a value greater than TIM3_ARR is written to it, the counter will continue to increment. Writing 0 or ARR directly will immediately update the counting direction but will not generate an update event.
- When using center alignment mode, it is recommended to set the UG bit in the TIM3_EGR register to 1 before starting the counter; this generates a software update to initialize all registers. Do not modify the counter value after the counter has started.

12.4.3.3.4 Trigger Source Output

In PWM mode, CCx_SETTRGO signal can be used to trigger modules such as ADC. This chapter only introduces trigger logic of the SETTRGO signal, please refer to the ADC chapter for the detailed trigger source selection, trigger edge selection, and other information.

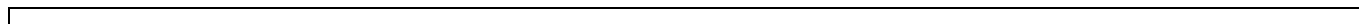
In edge alignment mode, every time compare match (the current count value of TIM3_CNT is equal to TIM3_CCRx), CCx_SETTRGO undergoes a flip. The following figure shows in edge aligned up counting mode, example of CCx_SETTRGO output.

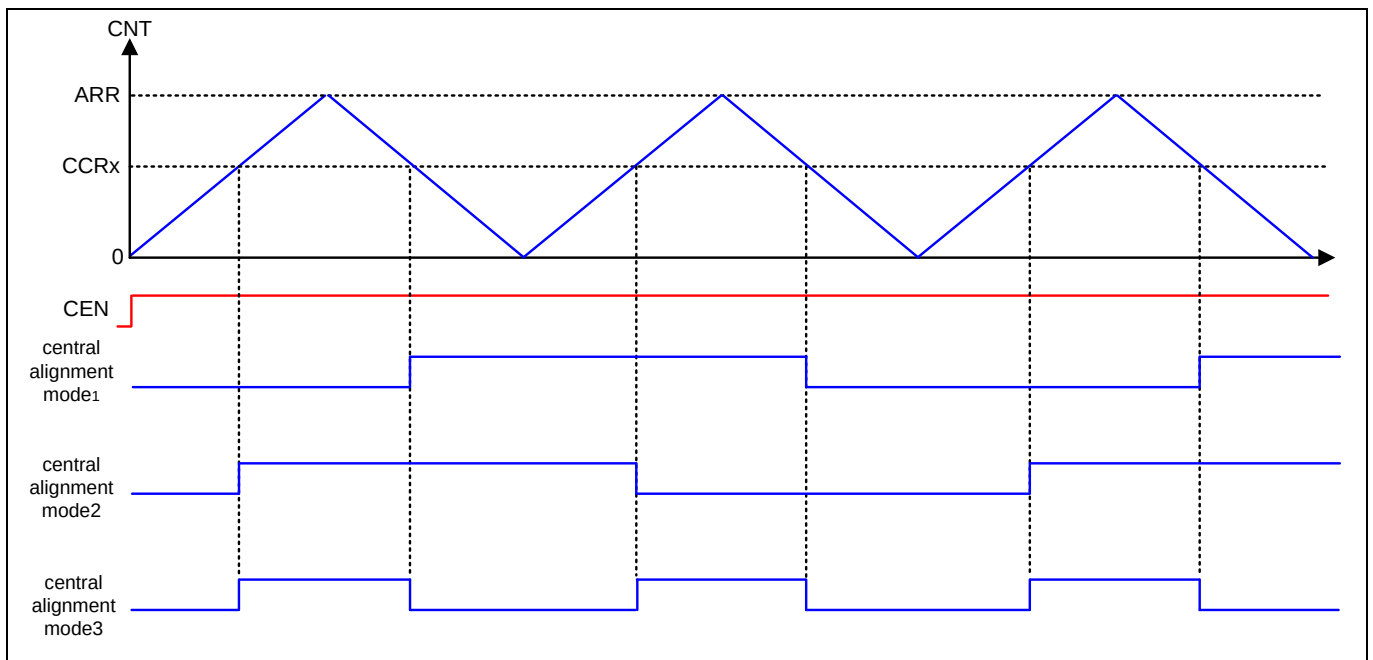
Figure 12-19 example of CCx_SETTRGO output with edge aligned up counting mode



In central alignment mode 1, when compare match the down counting cycles, CCx_SETTRGO undergoes a flip. In central alignment mode 2, when comparing match the up-counting cycles, CCx_SETTRGO undergoes a flip. In central alignment mode 3, flip occurs when the up or down counting cycles match. The following figure shows in central alignment mode, an example of CCx_SETTRGO output.

Figure 12-20 example of CCx_SETTRGO output with central alignment mode





12.4.3.4 External event clear OCxREF

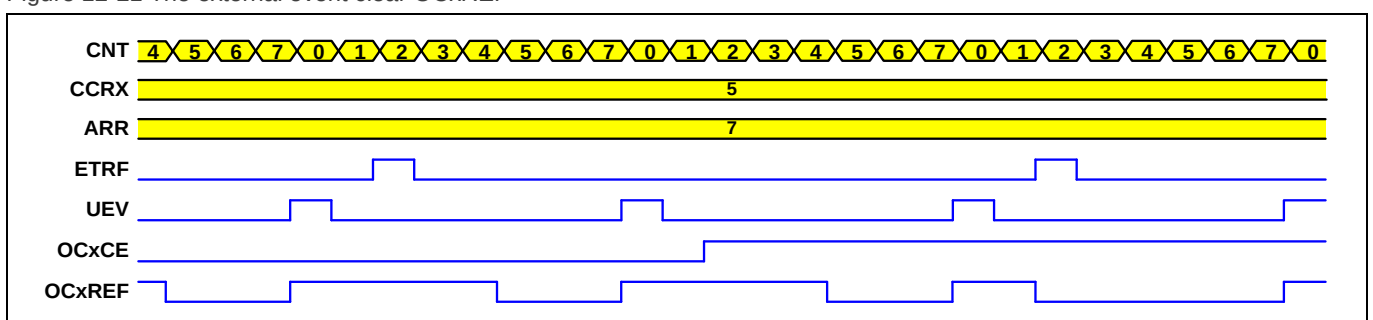
When configuring TIM3_CCMR register OCxCE=1, OCxREF can be driven low by the active level of ETRF input terminal till the next update event (UEV). The function can only be used in the output mode and PWM mode, it does not work in the force output mode.

For example, when OCxREF signal is connected to an external input, ETR configuration:

1. Configure TIM3_SMCR register ETPS [1:0] =00, close the external trigger prescaler.
2. Configure TIM3_SMCR register ECE=0, disable external clock mode 2.
3. Configure TIM3_SMCR register ETF [3:0] and ETP, configure ETR signal trigger polarity and filter width.

In the figure below, when ETRF input change triggers ETRF as high, it corresponds to different OCxCE value, and OCxREF signal action (PWM mode).

Figure 12-21 The external event clear OCxREF



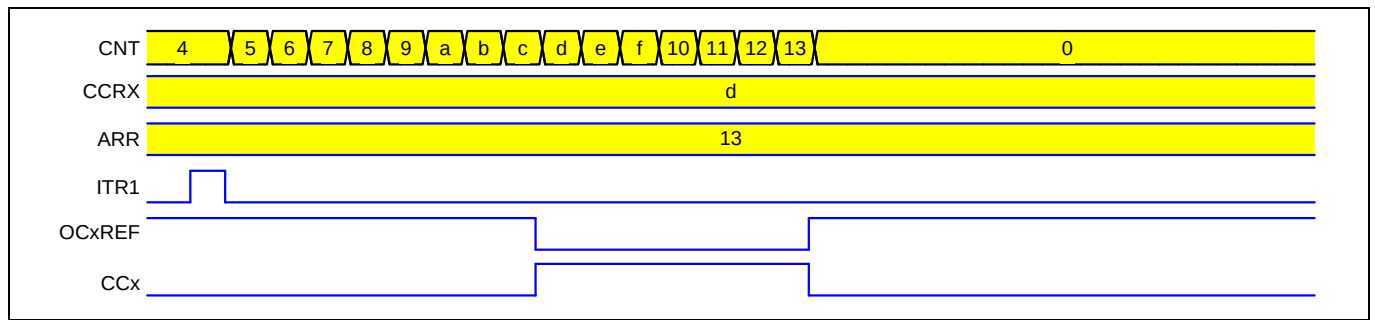
12.4.3.5 One-pulse output

In the one-pulse mode (OPM), the counter responds to a stimulus and generates a pulse with width programmable. Configure TIM3_CR1 register OPM =1. Select one-pulse mode, trigger signal's active edge or configuring CEN =1 can enable the counter. CEN =1 remains until the next update event or configure CEN =0.

The necessary condition to generate pulse is the difference of compare value and initial value of the counter. Necessary configuration prior to counter enable:

- Up counting mode: Counter $CNT < CCRx \leq ARR$.
- Down counting mode: Counter $CNT > CCRx$.

Figure 12-22 one-pulse mode



For example, when TI2 detects the rising edge, and after latency t_{DELAY} , OC2 generates a positive pulse with the length of t_{PULSE} .

Configure TI2FP2 serves as the trigger source:

1. Configure TIM3_CCMR1 register CC2S = 01, map TI2FP2 in TI2.
2. Configure TIM3_CCER register CC2P = 0, detect TI2FP2 rising edge.
3. Configure TIM3_SMCR register TS = 110, TI2FP2 as the slave mode counter trigger (TRGI).
4. Configure TIM3_SMCR register SMS = 110, select trigger mode, TI2FP2 enables the counter.

OPM waveform is determined by TIM3_ARR and TIM3_CCR1 (consider clock frequency and counter prescaler): t_{DELAY} is determined by TIM3_CCR1 register value and CNT initial value; TIM3_ARR – TIM3_CCR1 value is t_{PULSE} .

Next example is generating a negative pulse. In case of compare and match, generate a waveform from 1 to 0. When the counter preload value is achieved, generate a waveform from 0 to 1:

1. Configure TIM3_CCMR1 register OC1M = 111, select PWM mode 2.
2. Configure TIM3_CCER register CC1P = 1, export active low level.
3. Configure TIM3_CCMR1 OC1PE = 1 and TIMx_CR1 register ARPE, enable preload register.
4. Configure TIM3_CCR1 register and TIMx_ARR register.
5. Configure TIM3_EGR register UG = 1, generate an update event.
6. Wait for an external trigger event on TI2.

In this case, TIM3_CR1 registers DIR=0, CMS=0, OPM= 1, stop counting in the next update event (when the counter toggles to 0 from the auto load value).

12.4.3.5.1 OCx fast enable

OCx fast enable is a special scenario of one-pulse mode. In the one-pulse mode, set TIM3_CCMR register OCxFE=1, and enforce OCxREF to directly respond to stimulus without relying on the compare result of counter and compare value. The output waveform is the same as the waveform when compare and match. Thus, it can remove the compare time and export the compare result fast. OCx rapid output enables only becomes active in the PWM mode.

12.4.4 Slave mode

12.4.4.1 Encoder interface

The encoder interface mode means the counter count under interaction of orthogonal signals of TI1 and TI2. At the change of input source, the count direction is auto modified by hardware. Configure TIMx_SMCR register SMS bit and select input source. According to input source difference, divide the encoder interface into 3 modes: SMS =001, encoder interface mode 1; SMS =010, encoder interface mode 2; SMS =011, encoder interface mode 3; the specific count in the three modes is expressed below. Two input TI1 and TI2 are used as the interface of orthogonal encoder.

In the encoder mode, prior to the counter enable, configure the ARR register. In the encoder interface mode, it's like using an external clock with direction selection. The counter counts continuously between 0 to the auto load value of TIM3_ARR register (up count and down count are controlled by external clock).

Note: The encoder mode doesn't support the external clock mode 2.

In the encoder interface code, the counter is auto changed according to the speed and direction of increment encoder. Therefore, the counter content always points to the encoder position. The count direction corresponds to the direction of the connected sensor. The table below presents all possible combinations, assuming that TI1 and TI2 are not exchanged at the same time.

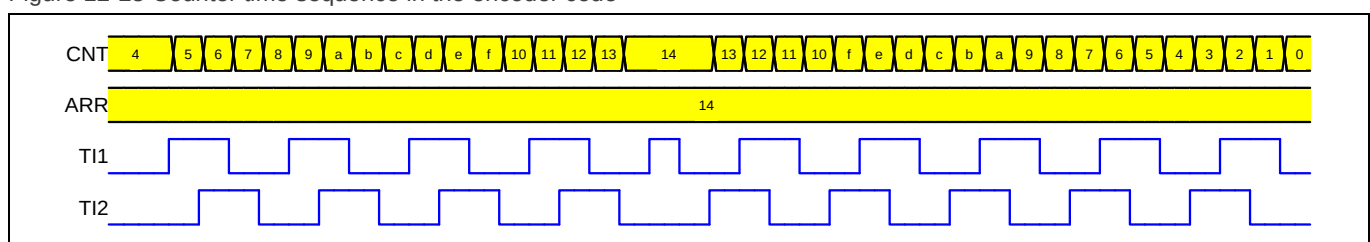
Table 12-2 Relations of count direction and encoder signal

Count mode	Relative level (TI1FP1 relative to TI2, TI2FP2 relative to TI1)	TI1FP1 signal		TI2FP2 signal	
		Rising	Falling	Rising	Falling
Encoder interface mode 1 (only count at TI2)	High level	-	-	up count	down count
Encoder interface mode 1 (only count at TI2)	Low level	-	-	down count	up count
Encoder interface mode 2 (only count at TI1)	High level	down count	up count	-	-
Encoder interface mode 2 (only count at TI1)	Low level	up count	down count	-	-
Encoder interface mode 3 (only count at TI1 and TI2)	High level	down count	up count	up count	down count
Encoder interface mode 3 (only count at TI1 and TI2)	Low level	up count	down count	down count	up count

The counter configuration and sequence figure in the encoder interface mode is below. We can see the count signal generation and direction control in the figure. Specific configuration:

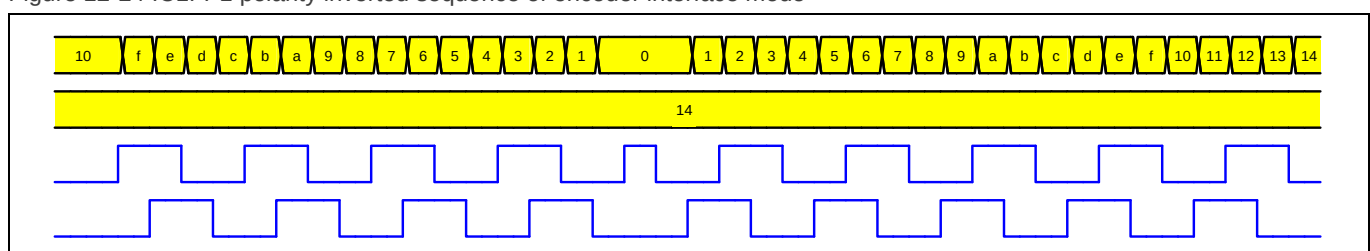
1. Configure TIM3_CCMR register CC1S=01, map IC1FP1 on TI1.
2. Configure TIM3_CCMR register CC2S=01, map IC2FP2 on TI2.
3. Configure TIM3_CCER register CC1P=0, IC1 non-inverted, IC1=TI1.
4. Configure TIM3_CCER register CC2P=0, IC2 non-inverted, IC1=TI2.
5. Configure TIM3_SMCR register SMS=011, select encoder mode 3. According to the input level of another signal, the counter counts at the edge of TI1FP1 and TI2FP2.
6. Configure TIM3_CR1 register CEN=1, enable the counter.

Figure 12-23 Counter time sequence in the encoder code



Counter sequence when the IC1FP1 polarity inverted (CC1P = 1, another configuration unchanged).

Figure 12-24 IC1FP1 polarity inverted sequence of encoder interface mode



In the encoder interface mode, the counter can provide the current position information of the sensor. Use another timer in the capture mode to measure the interval period of two encoder events and acquire dynamic

information (speed, acceleration, and deceleration). According to the interval period of two encoder events, regularly read the counter. Lock the counter value in the third input capture register (the acquire signal should be cyclic and generated by another timer). When the chip has the built in DMA, read the value upon DMA request.

12.4.4.2 Reset mode

Configure TIM3_SMCR register SMS=100, select slave reset mode. In this mode, the TRGI input event will clear and restart the counter.

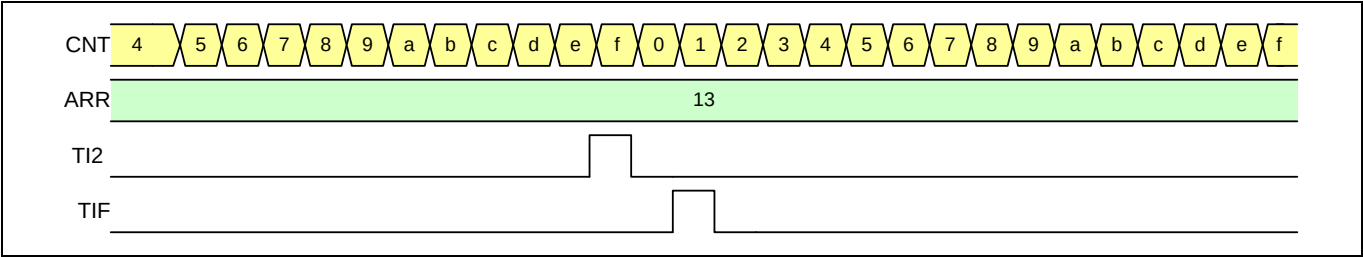
For example, TI2 input terminal falling edge trigger counter restart:

- 1. Configure TIM3_CCMR1 CC2S=01. CC2 channel is configured as input mode. IC2 is mapped on T12. Configure TIM3_CCER register CC2P=1, detect the falling edge.
- 2. Configure TIM3_SMCR register SMS = 100 and select the slave mode as reset mode. Configure TIM3_SMCR register TS =110, select timer input 2 after filter (TI2FP2) as the trigger input of sync timer.
- 3. Configure TIM3_CR1 register DIR=0, select the count direction as the up count; configure PSC=0, no division; configure CEN=1, enable the counter.

The counter clock source is provided by the internal clock. When detecting T12 falling edge, clear and restart the counter. The trigger interrupt flag is set by hardware.

TIM3_ARR = 0x13 time sequence in reset mode.

Figure 12-25 Control time sequence of reset mode



12.4.4.3 Gate mode

Configure TIM3_SMCR register SMS=101, select slave gate mode. In this mode, select active level (0: high level active, 1: low level active) according to the TIM3_CCER register CCxP value. When TRGI input is active level, the counter always starts. Otherwise, the counter stops (but without reset operation), and the counter start and stop are countable.

For example, the counter only counts when TI1 is high:

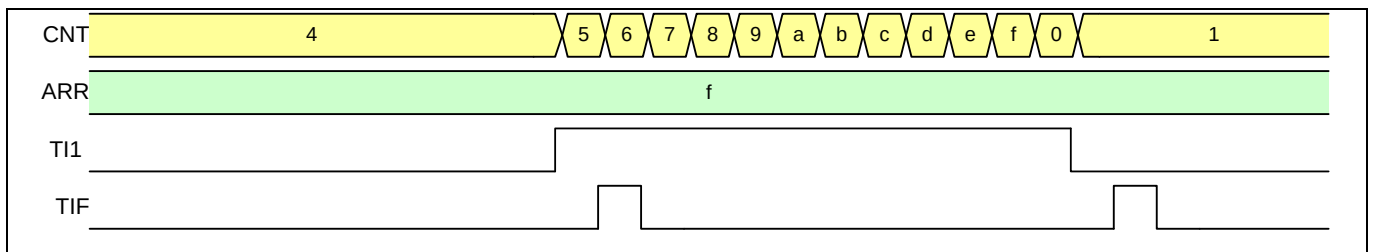
- 1. Configure CC1S=01 of TIM3_CCMR1 register. CC1 channel is configured as input mode. IC1 is mapped on TI1. Configure TIM3_CCER register CC1P=0, detect the high level of TI1.
- 2. Configure TIM3_SMCR register SMS=101, select slave gate mode; configure TIM3_SMCR register TS=101, select the timer input after filter 1 (TI1FP1) as the sync counter trigger input.
- 3. Configure TIM3_CR1 register DIR=0, select count direction as the up count; configure PSC=0, no division; configure CEN=1, enable the counter.

The counter clock source is provided by the internal clock. When detecting TI1 high level, the counter count starts. When TI1 is at a low level, the counter stops. When the counter starts or stops, set TIF as 1.

TIM3_ARR=0xf time sequence in gate mode.

Figure 12-26 Control time sequence of gate mode





12.4.4.4 Trigger mode

Configure TIM3_SMCR register SMS=110, and select slave trigger mode. Select active edge (0: rising edge active, 1: falling edge active) according to the TIM3_CCER register CCxP value. When TRGI input is active edge, the counter starts counting. The timer startup is controlled and the stop is not controlled.

For example, the counter count starts at the rising edge of TI1 input:

1. Configure CC1S=01 of TIM3_CCMR1 register. CC1 channel is configured as input mode. IC1 is mapped on TI1. Configure TIM3_CCER register CC1P=0, and detect the rising edge.

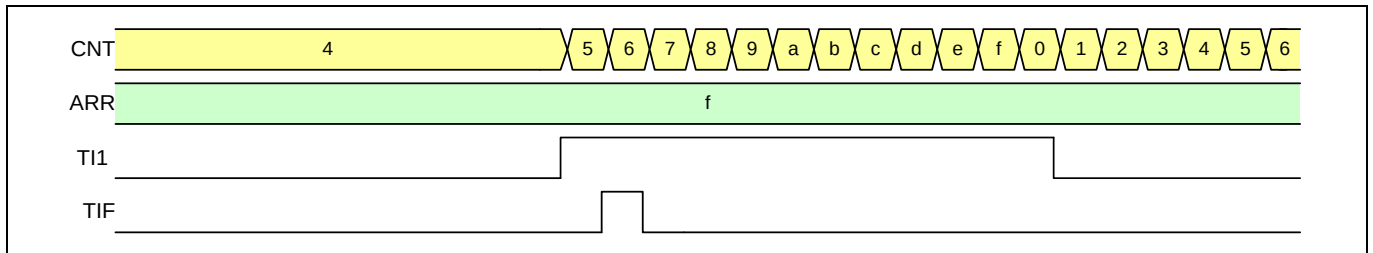
2. Configure TIM3_SMCR register SMS = 110, select slave trigger mode; configure TIM3_SMCR register TS=101, select the timer input 1 (TI1FP1) after filter as the trigger input of counter.

3. Configure TIM3_CR1 register DIR =0, select count direction as the up count; configure PSC=0, no division.

The counter clock source is provided by internal clock. When detecting TI1 rising edge, the counter count starts.

TIMx_ARR =0xf sequence in the trigger mode.

Figure 12-27 Control time sequence of trigger mode



12.4.4.5 External clock mode 2+ slave mode

When the clock source selects the external clock mode 2, and ETR signal is used as external clock input, it can be used together with the other slave mode. In this mode, the slave mode only supports the reset mode, gate mode, trigger mode, and not supports the external clock mode 1 and encoder mode.

For example, in the trigger mode, the counter counts once at each rising edge of ETR:

1. Configure TIM3_SMCR register ETF = 0000, don't use the digital filter; configure TIM3_SMCR register ETPS = 00, close the prescaler; configure TIM3_SMCR register ETP =0, detect ETR rising edge; configure TIM3_SMCR register ECE = 1, enable external clock mode 2.

2. Configure TIM3_CCMR1 register CC1S=01, CC1 channel is configured as input, and IC1 mapped on TI1 as the input capture source; configure TIM3_CCER register CC1P=0, selected rising edge is active.

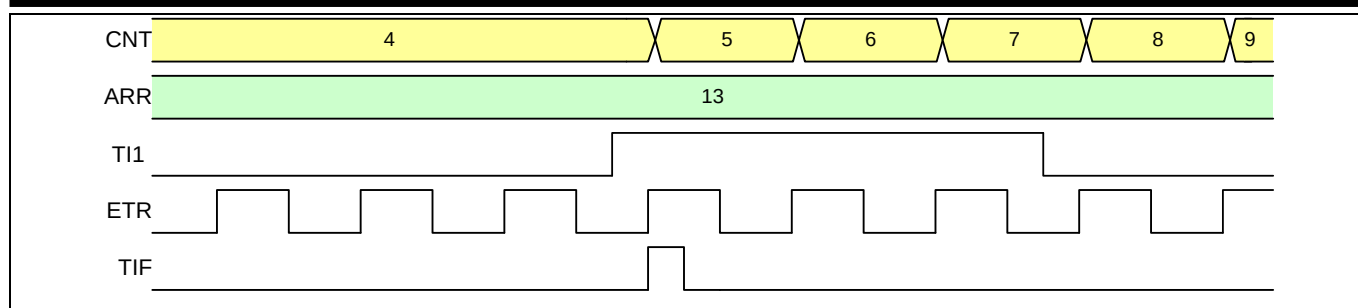
3. Configure TIM3_SMCR register SMS = 110, select slave trigger mode. Configure TIM3_SMCR register TS = 101, select TI1 as input source.

4. Configure TIM3_CR1 register DIR=0, select the count direction as up count. Configure PSC=0, no division.

The counter count begins at the rising edge of TI1, and set TIF. The latency between the ETR signal rising edge and actual reset time of counter depends on the sync circuit of ETR input terminal.

TIM3_ARR=13 time sequence in external clock mode 2 + slave mode (trigger mode).

Figure 12-28 Control time sequence of external clock mode 2 +slave mode (trigger mode)



12.4.5 Timer synchronization

Different timers are internally connected, achieve timer cascading or synchronization.

Detailed descriptions are given in the relevant chapter of TIM1.

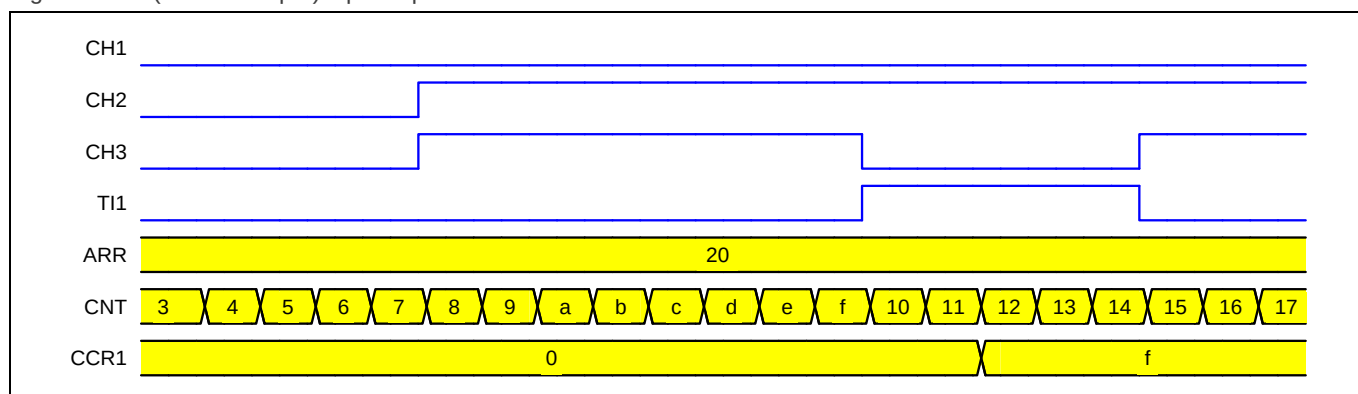
12.4.6 Timer XOR function

Configure TIM3_CR2 register TI1S=1, connect the TIM3_CH1, TIM3_CH2 and TIM3_CH3 pin to TI1 input terminal via XOR for all input modes of the timer.

For example, TIM3_CH1, TIM3_CH2 and TIM3_CH3 pins are connected to TI1 input terminal via XOR. Sample TI1 input signal's active edge. When TI1 rising edge is active, capture the current counter value, and lock it in the TIM3_CCR1 register. Specific configuration:

1. Configure TIM3_CR2 register TI1S=1, configure three inputs of the timer and connect it to TI1 input channel via XOR.
2. Configure TIM3_CCMR1 register CC1S=01, CC1 channel is configured as input, IC1 is mapped on TI1.
3. Configure TIM3_CCMR1 register IC1F [3:0], configure digital filter width (configure upon need).
4. Configure TIM3_CCER register CC1P=0, select capture in the rising edge of TI1 signal.
5. Configure TIM3_CCMR1 register IC1PS, select prescaler factor.
6. Configure TIM3_CCER register CC1E = 1, enable input/capture channel 1 capture.
7. Configure TIM3_CR1 register CEN=1, enable the counter.

Figure 12-29 (TI1 XOR input) input capture waveform



Hall interface circuit

Detailed descriptions are given in the relevant chapter of TIM1.

12.4.7 Debug mode

In the debug mode, configure DBG_CR register DBG_TIM3_STOP=1, TIM3 counter stops counting. (Details in the debug section)

12.4.8 Interrupt

TIM3 interrupt includes: capture/compare 1 interrupt, capture/compare 2 interrupt, capture/compare 3 interrupt, capture/compare 4 interrupt, update interrupt, trigger interrupt. When the corresponding interrupt enable bit is set, generate the corresponding event and interrupt.

Table 12-3 List of interrupt events

Interrupt event	Flag bit	Enable bit
Capture/compare 1 interrupt	CC1IF	CC1IE
Capture/compare 2 interrupt	CC2IF	CC2IE
Capture/compare 3 interrupt	CC3IF	CC3IE
Capture/compare 4 interrupt	CC4IF	CC4IE
Update interrupt	UIF	UIE
Trigger interrupt	TIF	TIE

12.5 Register

Table 12-4 TIM3 register overview

Offset	Acronym	Register Name	Reset
0x00	TIM3_CR1	Control Register 1	0x0000
0x04	TIM3_CR2	Control Register 2	0x0000
0x08	TIM3_SMCR	Slave Mode Control Register	0x0000 0000
0x0C	TIM3_DIER	Interrupt Enable Register	0x0000
0x10	TIM3_SR	Status Register	0x0000
0x14	TIM3_EGR	Event Generation Register	0x0000
0x18	TIM3_CCMR1	Capture/Compare Mode Register 1	0x0000
0x1C	TIM3_CCMR2	Capture/Compare Mode Register 2	0x0000
0x20	TIM3_CCER	Capture/Compare Enable Register	0x0000
0x24	TIM3_CNT	Counter	0x0000
0x28	TIM3_PSC	Prescaler	0x0000
0x2C	TIM3_ARR	Auto Reload Register	0x0000
0x34	TIM3_CCR1	Capture/Compare Register 1	0x0000
0x38	TIM3_CCR2	Capture/Compare Register 2	0x0000
0x3C	TIM3_CCR3	Capture/Compare Register 3	0x0000
0x40	TIM3_CCR4	Capture/Compare Register 4	0x0000
0x50	TIM3_OR	Input Option Register	0x0000

12.5.1 TIM3_CR1 Control Register 1

Address offset: 0x00

Reset value: 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						CKD		ARPE	CMS		DIR	OPM	URS	UDIS	CEN
						rw		rw	rw		rw	rw	rw	rw	rw

Bit	Field	Description
15 : 10	Reserved	Reserved, must be kept at reset value.

Bit	Field	Description
9 : 8	CKD	Clock division Division ratio between the timer clock (CK_INT) frequency and the dead-time and sampling clock used by the dead-time generators and the digital filters (ETR, TIx). 00 : $t_{DTS} = t_{INT_CK}$ 01 : $t_{DTS} = 2 \times t_{INT_CK}$ 10 : $t_{DTS} = 4 \times t_{INT_CK}$ 11 : Reserved, do not program this value
7	ARPE	Auto reload preload enable 0: Disable the shadow register of TIM3_ARR register 1: Enable the shadow register of TIM3_ARR register
6 : 5	CMS	Center alignment mode selection 00: Edge alignment mode. Count direction depends on DIR bit 01: Central alignment mode 1. The counter alternatively conducts up and down count. The channel is in output mode. Only in down count, compare interrupt flag bit is set 10: Central alignment mode 2. The counter alternatively conducts up and down count. The channel is in output mode. Only in up count, compare interrupt flag bit is set 11: Central alignment mode 3. The counter alternatively conducts up and down count. The channel is in output mode. In up and down count, compare interrupt flag bit is set Note: During count, the alignment mode change is disabled.
4	DIR	Count direction 0: up count 1: down count Note: When the counter is configured as central alignment mode or encoder mode, the bit is read only.
3	OPM	one-pulse mode 0: Disable one-pulse mode. In case of update event, the counter count continues 1: Enable one-pulse mode. In case of the next update event (clear CEN bit), the counter count stops
2	URS	Update request source This bit is set and cleared by software, select update event source. 0: The event below may generate a update interrupt request: - Counter overflow/underflow - Set UG bit - Update generation through the slave mode controller 1: Only in counter overflow/underflow, generate update interrupt request
1	UDIS	Update disable This bit is used to enable or disable the update event 0: Update event (UEV) enabled. 1: Update event disabled. The Update event is not generated, shadow registers keep their value (ARR, PSC, CCRx). However the counter and the prescaler are reinitialized if the EGR.UG bit is set or a hardware reset is received from the slave mode controller.
0	CEN	Counter enable 0: Counter disabled 1: Counter enabled Note: External clock, gated mode and encoder mode can work only if the CEN bit has been previously set by software. However trigger mode can set the CEN bit automatically by hardware.

12.5.2 TIM3_CR2 Control Register 2

Address offset: 0x04

Reset value: 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								TI1S	MMS			Reserved			
								rw	rw						

Bit	Field	Description
15 : 8	Reserved	Reserved, must be kept at reset value.
7	TI1S	TI1 selection 0: TIM3_CH1 pin is connected to TI1 input 1: TIM3_CH1, TIM3_CH2 and TIM3_CH3 pins are connected to the TI1 input after XOR combination
6 : 4	MMS	Master mode selection These bits control TRGO signal selection, used to select the sync information sent to the slave timers in master mode: 000: Reset TIM3_EGR register UG bit generate one pulse trigger output (TRGO). 001: Enable The Counter Enable signal CNT_EN is used as trigger output (TRGO). It is useful to start several timers at the same time or to control a window in which a slave timer is enable. The Counter Enable signal is generated by a logic OR between CEN control bit and the trigger input when configured in gated mode. When the Counter Enable signal is controlled by the trigger input, there is a delay on TRGO, except if the master/slave mode is selected. 010: Update Update event is selected as TRGO. 011: Capture/compare When a capture or a compare match occurred, send a positive pulse as TRGO. 100: Compare OC1REF signal is used as trigger output (TRGO) 101: Compare OC2REF signal is used as trigger output (TRGO) 110: Compare OC3REF signal is used as trigger output (TRGO) 111: Compare OC4REF signal is used as trigger output (TRGO)
3 : 0	Reserved	Reserved, must be kept at reset value.

12.5.3 TIM3_SMCR Slave Mode Control Register

Address offset: 0x08

Reset value: 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved										TS[4:3]		Reserved			
										rw					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETP	ECE	ETPS		ETF				MSM	TS			OCCS.	SMS		
rw	rw	rw		rw				rw	rw			rw	rw		

Bit	Field	Description
31: 22	Reserved	Reserved, must be kept at reset value.
21: 20	TS[4:3]	Trigger selection Trigger input source selection. Refer to the TS[2:0] bit description.
19: 16	Reserved	Reserved, must be kept at reset value.
15	ETP	External trigger polarity The bit selects ETR signal polarity. 0: High level or rising edge is active 1: Low level or falling edge is active Note: It's only suitable for supporting the external trigger product.

Bit	Field	Description
14	ECE	External clock enable The bit enables the external clock mode 2. 0: external clock mode 2 disabled 1: external clock mode 2 enabled, The counter is clocked by any active edge on the ETRFsignal. Note 1: It's only suitable for supporting the external trigger product. Note 2: Configuring ECE =1 has the same effect with configuring SMS =111 and TS =111. Note 3: TS≠111, reset mode, gate mode and trigger mode can be used together with the external clock mode 2. Note 4: If external clock mode 1 and external clock mode 2 are enabled at the same time, the external clock input is ETRF.
13: 12	ETPS	External trigger prescaler The external trigger signal ETRP frequency must be lower than 1/4 of TIM3 PCLK frequency. When input the quicker external clock, use the prescaler to reduce ETRP frequency. 00: Prescaler OFF 01: ETRP frequency divided by 2 10: ETRP frequency divided by 4 11: ETRP frequency divided by 8 Note: It's only suitable for supporting the external trigger product
11: 8	ETF	External trigger filter These bits define ETRP signal sampling frequency and ETRP digital filter bandwidth. In fact, the digital filter is an event counter, which generate an output jump after N event. 0000: No filter, f_{DTS} sampling 001: Sampling frequency $f_{sampling} = f_{INT_CK}$, $N = 2$ 0010: Sampling frequency $f_{sampling} = f_{INT_CK}$, $N = 4$ 0011: Sampling frequency $f_{sampling} = f_{INT_CK}$, $N = 8$ 0100: Sampling frequency $f_{sampling} = f_{DTS}/2$, $N = 6$ 0101: Sampling frequency $f_{sampling} = f_{DTS}/2$, $N = 8$ 0110: Sampling frequency $f_{sampling} = f_{DTS}/4$, $N = 6$ 0101: Sampling frequency $f_{sampling} = f_{DTS}/2$, $N = 8$ 0110: Sampling frequency $f_{sampling} = f_{DTS}/4$, $N = 6$ 0111: Sampling frequency $f_{sampling} = f_{DTS}/4$, $N = 8$ 1000: Sampling frequency $f_{sampling} = f_{DTS}/8$, $N = 6$ 1001: Sampling frequency $f_{sampling} = f_{DTS}/8$, $N = 8$ 1010: Sampling frequency $f_{sampling} = f_{DTS}/16$, $N = 5$ 1011: Sampling frequency $f_{sampling} = f_{DTS}/16$, $N = 6$ 1100: Sampling frequency $f_{sampling} = f_{DTS}/16$, $N = 8$ 1101: Sampling frequency $f_{sampling} = f_{DTS}/32$, $N = 5$ 1110: Sampling frequency $f_{sampling} = f_{DTS}/32$, $N = 6$ 1111: Sampling frequency $f_{sampling} = f_{DTS}/32$, $N = 8$ Note: It's only suitable for supporting the external trigger product
7	MSM	Master/slave mode 0: No action 1: The effect of an event on the trigger input (TRGI) is delayed to allow a perfect synchronization between the current timer and its slaves (through TRGO). It is useful if we want to synchronize several timers on a single external event.

Bit	Field	Description
6 : 4	TS[2:0]	Trigger selection 00000 : Internal trigger 0 (ITR0) 00001 : Internal trigger 1 (ITR1) 00010 : Internal trigger 2 (ITR2) 00011 : Internal trigger 3 (ITR3) 00100 : TI1 edge detector (TI1F_ED) 00101 : Timer input after filter 1 (TI1FP1) 00110 : Timer input after filter 2 (TI2FP2) 00111 : External trigger input (ETR) 01000 : Internal Trigger 4 (ITR4) 01001 : Reserved 01010 : Reserved 01011 : Reserved 01100 : Reserved 01101 : Reserved 01110 : Reserved 01111 : Reserved 10000 : Internal Trigger 12 (ITR12) 10001 : Internal Trigger 13 (ITR13) 10010 : Internal Trigger 14 (ITR14) 10011 : Internal Trigger 15 (ITR15) 10100 : Filtered timer input 3 (TI3FP3) 10101 : Filtered timer input 4 (TI4FP4) others : Reserved More details on ITRx refer to the table below. Note: After the slave mode enable, these bits cannot be changed
3	OCCS	Timer compare output signal(OCxREF) clear selection In PWM mode, clear the compare output signal(OCxREF) 0: External trigger signal as clear signal 1: Comparator output(COMP) as clear signal Note: It's only applicable to products that support external trigger or have a built-in comparator (COMP).
2 : 0	SMS	Slave mode selection When external signals are selected the active edge of the trigger signal (TRGI) is linked to the polarity selected on the external input. 000: Close slave mode - In case of CEN =1, the prescaler is directly driven by the internal clock. 001: Encoder mode 1 - Based on TI1FP1 level, the counter conducts up/down count along TI2FP2 edge. 010: Encoder mode 2 - Based on TI2FP2 level, the counter conducts up/down count along TI1FP1edge. 011: Encoder mode 3- Based on another input level, the counter conducts up/down count along TI1FP1 and TI2FP2 edge. 100: Reset mode -The rising edge of the selected trigger input (TRGI) reinitialize the counter and generate an update register signal. 101: Gate mode - When the trigger input (TRGI) is high, the counter count begins. When the trigger input turns low, the counter count stops (but without reset). The counter start and stop are controlled. 110: Trigger mode -The counter starts in the rising edge of the trigger input TRGI (but without reset). Only the counter start is controlled. 111: External clock mode 1 -The rising edge of the selected trigger input (TRGI) drives the counter. Note: If TI1F_EN is selected as trigger input (TS =100), don't use the gate mode. This is because that EI1F_ED exports a pulse at each TI1F change. However, in the gate mode, it's required to check the trigger input level.

Note: For internal trigger interconnection relationships, refer to the interconnection table between TIM in the chip-specific configuration section.

12.5.4 TIM3_DIER Interrupt Enable Register

Address offset: 0x0C

Reset value: 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									TIE	Res.	CC4IE	CC3IE	CC2IE	CC1IE	UIE
									rw		rw	rw	rw	rw	rw

Bit	Field	Description
15 : 7	Reserved	Reserved, must be kept at reset value.
6	TIE	Trigger interrupt enable 0: Break interrupt disable 1: Break interrupt enable
5	Reserved	Reserved, must be kept at reset value.
4	CC4IE	Enable Capture/Compare 4 interrupt 0: Capture/compare interrupt 4 disable 1: Capture/compare interrupt 4 enable
3	CC3IE	Enable Capture/Compare 3 interrupt 0: Capture/compare interrupt 3 disable 1: Capture/compare interrupt 3 enable
2	CC2IE	Enable Capture/Compare 2 interrupt 0: Capture/compare interrupt 2 disable 1: Capture/compare interrupt 2 enable
1	CC1IE	Enable Capture/Compare 1 interrupt 0: Capture/compare interrupt 1 disable 1: Capture/compare interrupt 1 enable
0	UIE	Enable update interrupt 0: Update event interrupt disable 1: Update event interrupt enable

12.5.5 TIM3_SR Status Register

Address offset: 0x10

Reset value: 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			CC4OF	CC3OF	CC2OF	CC1OF	Reserved		TIF	Res.	CC4IF	CC3IF	CC2IF	CC1IF	UIF
			r_w0c						r_w0c		r_w0c				

Bit	Field	Description
15 : 13	Reserved	Reserved, must be kept at reset value.
12	CC4OF	Capture/compare 4 over capture flag Refer to CC1OF description.
11	CC3OF	Capture/compare 3 over capture flag Refer to CC1OF description.
10	CC2OF	Capture/compare 2 over capture flag Refer to CC1OF description.
9	CC1OF	Capture/compare 1 over capture flag Only when the channel 1 is configured as input capture mode and CC1F is configured as 1, and the capture event occurs again. The flag is set by hardware. Write 0 may clear the bit. 0: No overcapture 1: Overcapture
8 : 7	Reserved	Reserved, must be kept at reset value.
6	TIF	Trigger interrupt flag In case of trigger event (When the slave mode counter is in any other mode except for the gate mode, detect the active edge in TRGI input terminal, or detect any edge in gate mode), the bit is set by hardware. It's cleared by software. 0: No trigger event occurred 1: Trigger interrupt pending
5	Reserved	Reserved, must be kept at reset value.
4	CC4IF	Capture/compare 4 interrupt flag Refer to CC1IF description

Bit	Field	Description
3	CC3IF	Capture/compare 3 interrupt flag Refer to CC1IF description
2	CC2IF	Capture/compare 2 interrupt flag Refer to CC1IF description.
1	CC1IF	Capture/compare 1 interrupt flag Channel 1 in output mode: When the counter value and compare value match, the bit is set by hardware except for the central alignment mode (the bit is set in the central alignment mode according to TIM3_CR1.CMS [1:0]). It's cleared by software. 0: No match 1: TIM3_CNT value and TIM3_CCR1 value match Channel 1 as output mode: In case of capture event, the bit is set by hardware. It's cleared by software or reading TIM3_CCR1 register 0: No input capture occurred 1: Counter value is captured to TIM3_CCR1
0	UIF	Update interrupt flag In case of update event, the bit is set by hardware. It's cleared by software. 0: No update interrupt occurred 1: update interrupt pending This bit is set by hardware when the registers are updated: - In case of TIM3_CR1 register UDIS=0 and REP_CNT=0, when the counter generates overflow/underflow. -In case of TIM3_CR1 register UDIS=0, URS=0, when TIMx_EGR register UG =1. - In case of TIM3_CR1 register UDIS=0, URS=0, when update generation through the slave mode controller.

12.5.6 TIM3_EGR Event Generation Register

Address offset: 0x14

Reset value: 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									TG	Res.	CC4G	CC3G	CC2G	CC1G	UG
									W		W	W	W	W	W

Bit	Field	Description
15 : 7	Reserved	Reserved, must be kept at reset value.
6	TG	Trigger generation 0: No action 1: generate a trigger event. TIM3_SR register TIF =1. When enabling the corresponding interrupt, generate the corresponding interrupt. It's auto cleared by hardware.
5	Reserved	Reserved, must be kept at reset value.
4	CC4G	Capture/compare 4 generation Refer to CC1G description.
3	CC3G	Capture/compare 3 generation Refer to CC1G description.
2	CC2G	Capture/compare 2 generation Refer to CC1G description.
1	CC1G	Capture/compare 1 generation This bit is set by software. It generates a capture/compare event, and is auto cleared by hardware. 0: No action 1 : Generate a capture/compare event in channel CC1: When the channel CC1 is configured as output: Set CC1IF =1. When enabling the corresponding interrupt, generate the corresponding interrupt. When the channel CC1 is configured input: The current value of the counter is captured in TIMx_CCR1 register, set CC1IF =1. When enabling the corresponding interrupt, generate the corresponding interrupt. When CC1IF is set, set CC1OF =1.

Bit	Field	Description
0	UG	Update event generation 0 : No action 1: Initialize the counter, and generate an update event. It's auto cleared by hardware. When selecting the central alignment or up counting mode, the counter is clear; otherwise (down counting mode) the counter auto reload value is loaded. The prescaler counter is cleared at the same time.

12.5.7 TIM3_CCMR1 Capture/Compare Mode Register 1

Address offset: 0x18

Reset value: 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC2CE.	OC2M			OC2PE	OC2FE	CC2S		OC1CE.	OC1M			OC1PE	OC1FE	CC1S	
IC2F				IC2PSC		CC2S		IC1F				IC1PSC		CC1S	
rw	rw			rw	rw	rw		rw	rw			rw	rw	rw	

The channel may be used for input (capture mode) or output (compare mode). The channel direction is defined by the corresponding CCxS. The register's bit other than CCxS functions differently in the input mode and output mode. OCxx describes the channel function in the output mode. ICxx describes the channel function in the input mode.

Output compare mode:

Bit	Field	Description
15	OC2CE	Output compare 2 clear enable Refer to OC1CE description
14 : 12	OC2M	Channel 2 output and compare mode Refer to OC1M description.
11	OC2PE	Channel 2 output and compare preload enable Refer to OC1PE description.
10	OC2FE	Channel 2 output and compare quick enable Refer to OC1FE description.
9 : 8	CC2S	Channel 2 capture/compare 2 selection This bit defines the channel direction and input signal selection. Only when the channel closes, these bits can be written: 00 : Channel 2 is configured as output 01 : Channel 2 is configured as input, IC2 is mapped on TI2 10 : Channel 2 is configured as input, IC2 is mapped on TI1 11 : Channel 2 is configured as input, IC2 is mapped on TRC. The mode only works when the internal generator input is selected (TIM3_SMCR register TS bit selection)
7	OC1CE	Output compare 1 clear enable 0: OC1REF is unaffected by the ETR input. 1: When detecting ETR as active level, clear OC1REF.

Bit	Field	Description
6 : 4	OC1M	Channel 1 output compare mode The bit has defined the output reference signal OC1REF action. OC1REF has determined OC1 and OC1N value. OC1REF is active at high level. The active level of OC1, OC1N depends on CC1P and CC1NP bit. 000 : Freeze. TIM3_CCR1 and TIM3_CNT compare results has no effect on OC1REF. 001: Set as high when configuration. When TIM3_CNT value and TIM3_CCR1 value are same, enforce OC1REF as high level 010: Set as low when configuration. When TIM3_CNT value and TIM3_CCR1 value are same, enforce OC1REF as low level 011: Toggle when match. When TIM3_CCR1=TIM3_CNT, OC1REF toggle. 100: Enforce as low. Enforce OC1REF at low level 101: Enforce as high. Enforce OC1REF at high level 110: PWM mode 1. During up count, in case of TIM3_CNT<TIM3_CCR1, enforce OC1REF is at high level. Or else, it's at low level. During down count, in case of TIM3_CNT > TIM3_CCR1, enforce OC1REF is at low level. Or else, it's at high level. 111: PWM mode 2. During up count, in case of TIM3_CNT<TIM3_CCR1, channel 1 enforce OC1REF is at low level. Or else, it's at high level. During down count, in case of TIM3_CNT > TIM3_CCR1, enforce OC1REF is at high level. Or else, it's at low level. Note 1: In case of LOCK level 3 (TIM3_BDTR register LOCK bit) and CC1S = 00 (the channel is configured as output), the bit cannot be changed. Note 2: In PWM mode 1 or PWM mode 2, only when the compare result changes or it changes over from the freeze mode to PWM mode in the output compare mode, OC1REF level may change.
3	OC1PE	Channel 1 output compare preload enable 0: Disable TIM3_CCR1 register preload function. The value written into TIM3_CCR1 register becomes active immediately 1: Enable TIM3_CCR1 register preload function. Only read and write the preload register. TIM3_CCR1 preload value becomes active at the update event Note 1: When the LOCK level is 3 (TIM3_BDTR register LOCK bit) and CC1S = 00 (when the channel is configured as output), the bit cannot be changed. Note 2: Only in the one-pulse mode (TIM3_CR1 register OPM= 1), it has no influence whether the preload register is set. Under other scenarios, it's required to set the preload register. Otherwise, the follow up action is not certain.
2	OC1FE	Channel 1 output compare quick enable In case of the bit is set 1, when the channel is configured as PWM mode, it speeds up response of the capture/compare output to the trigger time. The output channel deems the active edge of the trigger input signal as one compare match. Therefore, OC is set as compare level, but is irrelevant to the compare result. 0: channel 1 output compare fast enable. 1: channel 1 output compare fast disable.
1 : 0	CC1S	Channel 1 capture/compare selection This bit defines the channel direction and input signal selection. Only when the channel closes, these bits can be written: 00 : Channel 1 is configured as output 01 : Channel 1 is configured as input, IC1 is mapped on TI1 10 : Channel 1 is configured as input, IC1 is mapped on TI2 11 : Channel 1 is configured as input, IC1 is mapped on TRC. This mode only works when the internal trigger input is selected (TIM3_SMCR register TS bit selection)

Input capture mode:

Bit	Field	Description
15 : 12	IC2F	Input capture 2 filter Refer to IC1F description
11 : 10	IC2PSC	Input/capture 2 prescaler Refer to IC1PSC description

Bit	Field	Description
9 : 8	CC2S	Channel 2 capture/compare selection This bit defines the channel direction and input signal selection. Only when the channel closes, these bits can be written: 00 : Channel 2 is configured as output 01 : Channel 1 is configured as input, IC2 is mapped on TI2 10 : Channel 2 is configured as input, IC2 is mapped on TI1 11 : Channel 2 is configured as input, IC2 is mapped on TRC. This mode only works when the internal trigger input is selected (TIM3_SMCR register TS bit selection)
7 : 4	IC1F	Channel 1 input capture filter The digital filter is composed of an event counter. It records an output jump after N input events. These bits define IC1 input signal sampling frequency and digital filter length. 0000: No filter, f_{DTS} sampling 001: Sampling frequency $f_{sampling} = f_{INT_CK}$, $N = 2$ 0010: Sampling frequency $f_{sampling} = f_{INT_CK}$, $N = 4$ 0011: Sampling frequency $f_{sampling} = f_{INT_CK}$, $N = 8$ 0100: Sampling frequency $f_{sampling} = f_{DTS}/2$, $N = 6$ 0101: Sampling frequency $f_{sampling} = f_{DTS}/2$, $N = 8$ 0110: Sampling frequency $f_{sampling} = f_{DTS}/4$, $N = 6$ 0111: Sampling frequency $f_{sampling} = f_{DTS}/4$, $N = 8$ 1000: Sampling frequency $f_{sampling} = f_{DTS}/8$, $N = 6$ 1001: Sampling frequency $f_{sampling} = f_{DTS}/8$, $N = 8$ 1010: Sampling frequency $f_{sampling} = f_{DTS}/16$, $N = 5$ 1011: Sampling frequency $f_{sampling} = f_{DTS}/16$, $N = 6$ 1100: Sampling frequency $f_{sampling} = f_{DTS}/16$, $N = 8$ 1101: Sampling frequency $f_{sampling} = f_{DTS}/32$, $N = 5$ 1110: Sampling frequency $f_{sampling} = f_{DTS}/32$, $N = 6$ 1111: Sampling frequency $f_{sampling} = f_{DTS}/32$, $N = 8$
3 : 2	IC1PSC	Channel 1 input/capture prescaler This bit defines the ratio of the prescaler acting on IC1. The prescaler is reset as soon as CC1E=0(TIM3_CCER register). 00: No prescaler, the capture input detects each edge to trigger one capture 01: Trigger one capture for every 2 events 10: Trigger one capture for every 4 events 11: Trigger one capture for every 8 events
1 : 0	CC1S	Channel 1 capture/compare selection This bit defines the channel direction and input signal selection. Only when the channel closes, these bits can be written: 00: Channel 1 is configured as output 01: Channel 1 is configured as input, IC1 is mapped on TI1 10: Channel 1 is configured as input, IC1 is mapped on TI2 11: Channel 1 is configured as input, IC1 is mapped on TRC. This mode only works when the internal trigger input is selected (TIM3_SMCR register TS bit selection)

12.5.8 TIM3_CCMR2 Capture/Compare Mode Register 2

Address offset: 0x1C

Reset value: 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC4CE	OC4M			OC4PE	OC4FE	CC4S		OC3CE	OC3M			OC3PE	OC3FE	CC3S	
IC4F				IC4PSC		CC4S		IC3F				IC3PSC		CC3S	
rw	rw			rw	rw	rw		rw	rw			rw	rw	rw	

Compare output mode:

Bit	Field	Description
15	OC4CE	Channel 4 output compare clear enable Refer to OC3CE description

Bit	Field	Description
14 : 12	OC4M	Channel 4 output compare mode Refer to OC3M description
11	OC4PE	Channel 4 output compare preload enable Refer to OC3PE description
10	OC4FE	Channel 4 output compare rapid enable Refer to OC3FE description
9 : 8	CC4S	Channel 4 capture/compare selection This bit defines the channel direction and input signal selection. Only when the channel closes, these bits can be written: 00: Channel 4 is configured as output 01: Channel 4 is configured as input, IC4 is mapped on TI4 10: Channel 4 is configured as input, IC4 is mapped on TI3 11: Channel 4 is configured as input, IC4 is mapped on TRC. This mode only works when the internal trigger input is selected (TIM3_SMCR register TS bit selection)
7	OC3CE	Channel 3 output compare clear enable 0: OC3REF is not affected by ETR input 1: When detecting ETR input is at active level, clear OC3REF
6 : 4	OC3M	Channel 3 output compare mode The bit has defined the output reference signal OC3REF action. OC3REF has determined OC3 and OC3N value. OC3REF is active at high level. The active level of OC3, OC3N depends on CC3P and CC3NP bit. 000 : Freeze. TIM3_CCR3 and TIM3_CNT compare results has no effect on OC3REF. 001: Set as high when configuration. When TIM3_CNT value and TIM3_CCR3 value are same, enforce OC3REF as high level 010: Set as low when configuration. When TIM3_CNT value and TIM3_CCR3 value are same, enforce OC3REF as low level 011: Toggle when match. When TIM3_CCR3=TIM3_CNT, OC3REF toggle. 100: Enforce as low. Enforce OC3REF at low level 101: Enforce as high. Enforce OC3REF at high level 110: PWM mode 1. During up count, in case of TIM3_CNT<TIM3_CCR3, enforce OC3REF as high level, otherwise its low level; during down count, in case of TIM3_CNT > TIM3_CCR3, enforce OC3REF as low level, otherwise its high level. 111: PWM mode 2. During up count, in case of TIM3_CNT<TIM3_CCR3, enforce OC3REF as low level, otherwise its high level; during down count, in case of TIM3_CNT > TIM3_CCR3, enforce OC3REF as high level, otherwise its low level. Note 1: In case of LOCK level 3 (TIM3_BDTR register LOCK bit) and CC3S = 00 (the channel is configured as output), the bit cannot be changed. Note 2: In PWM mode 1 or PWM mode 2, only when the compare result changes or it changes over from the freeze mode to PWM mode in the output compare mode, OC3REF level may change.
3	OC3PE	Channel 3 output compare preload enable 0: Disable TIM3_CCR3 register preload function, the value written into TIM3_CCR3 register becomes active immediately 1: Enable TIM3_CCR3 register preload function. The read and write only works on the preload register. TIM3_CCR3 preload value becomes active when the update event is available Note 1: When the LOCK level is set 3 (TIM3_BDTR register LOCK bit) and CC3S =0 (the channel is configured as output), the bit cannot be changed. Note 2: Only in the one-pulse mode (TIM3_CR1 register OPM= 1), it's not required to set the preload register. Under other scenarios, it's required to set the preload register. Otherwise, the follow up action is not certain.
2	OC3FE	Channel 3 output compare quick enable In case of the bit 1, when the channel is configured as PWM mode, it speeds up response of the capture/compare output to the trigger time. The output channel deems the active edge of the trigger input signal as one compare match. Therefore, OC is set as compare level, but is irrelevant to the compare result. 0: channel 3 output compare fast disable. 1: channel 3 output compare fast enable.

Bit	Field	Description
1 : 0	CC3S	Channel 3 capture/compare selection The bit has defined the channel direction and input signal selection. Only when the channel closes, these bits can be written: 00: Channel 3 is configured as input 01: Channel 3 is configured as input, IC3 is mapped on TI3 10: Channel 3 is configured as input, IC3 is mapped on TI4 11: Channel 3 is configured as input, IC3 is mapped on TRC. This mode only works when the internal trigger input is selected (TIM3_SMCR register TS bit selection)

Input capture mode:

Bit	Field	Description
15 : 12	IC4F	Input capture 4 filter Refer to IC3F description
11 : 10	IC4PSC	Input/capture 4 prescaler Refer to IC3PSC description
9 : 8	CC4S	Channel 4 capture/compare selection This bit defines the channel direction and input signal selection. Only when the channel closes, these bits can be written: 00 : Channel 4 is configured as output 01 : Channel 4 is configured as input, IC4 is mapped on TI4 10 : Channel 4 is configured as input, IC2 is mapped on TI3 11 : Channel 4 is configured as input, IC4 is mapped on TRC. This mode only works when the internal trigger input is selected (TIM3_SMCR register TS bit selection)
7 : 4	IC3F	Channel 3 input capture filter The digital filter is composed of an event counter. It records an output jump after N input events. These bits define IC1 input signal sampling frequency and digital filter length. 0000: No filter, f_{DTS} sampling 001: Sampling frequency $f_{sampling} = f_{INT_CK}$, $N = 2$ 0010: Sampling frequency $f_{sampling} = f_{INT_CK}$, $N = 4$ 0011: Sampling frequency $f_{sampling} = f_{INT_CK}$, $N = 8$ 0100: Sampling frequency $f_{sampling} = f_{DTS}/2$, $N = 6$ 0101: Sampling frequency $f_{sampling} = f_{DTS}/2$, $N = 8$ 0110: Sampling frequency $f_{sampling} = f_{DTS}/4$, $N = 6$ 0111: Sampling frequency $f_{sampling} = f_{DTS}/4$, $N = 8$ 1000: Sampling frequency $f_{sampling} = f_{DTS}/8$, $N = 6$ 1001: Sampling frequency $f_{sampling} = f_{DTS}/8$, $N = 8$ 1010: Sampling frequency $f_{sampling} = f_{DTS}/16$, $N = 5$ 1011: Sampling frequency $f_{sampling} = f_{DTS}/16$, $N = 6$ 1100: Sampling frequency $f_{sampling} = f_{DTS}/16$, $N = 8$ 1101: Sampling frequency $f_{sampling} = f_{DTS}/32$, $N = 5$ 1110: Sampling frequency $f_{sampling} = f_{DTS}/32$, $N = 6$ 1111: Sampling frequency $f_{sampling} = f_{DTS}/32$, $N = 8$
3 : 2	IC3PSC	Channel 3 input/capture prescaler This bit defines the ratio of the prescaler acting on IC3. The prescaler is reset as soon as CC3E=0(TIM3_CCER register). 00: No prescaler, the capture input detects each edge to trigger one capture 01: Trigger one capture for every 2 events 10: Trigger one capture for every 4 events 11: Trigger one capture for every 8 events
1 : 0	CC3S	Channel 3 capture/compare selection This bit defines the channel direction and input signal selection. Only when the channel closes, these bits can be written: 00: Channel 3 is configured as output 01: Channel 3 is configured as input, IC3 is mapped on TI3 10: Channel 3 is configured as input, IC3 is mapped on TI4 11: Channel 3 is configured as input, IC3 is mapped on TRC. This mode only works when the internal trigger input is selected (TIM3_SMCR register TS bit selection)

12.5.9 TIM3_CCER Capture/Compare Enable Register

Address offset: 0x20

Reset value: 0x0000

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

Bit	Field	Description
15	CC4NP	Channel 4 input/capture complementary output polarity Refer to CC1NP description
14	Reserved	Reserved, must be kept at reset value.
13	CC4P	Channel 4 capture/compare output polarity Refer to CC1P description.
12	CC4E	Channel 4 capture/compare enable Refer to CC1E description
11	CC3NP	Channel 3 input/capture complementary output polarity Refer to CC1NP description
10	Reserved	Reserved, must be kept at reset value.
9	CC3P	Channel 3 input capture output polarity Refer to CC1P description
8	CC3E	Channel 3 input/capture output enable Refer to CC1E description
7	CC2NP	Channel 2 input/capture complementary output polarity Refer to CC1NP description
6	Reserved	Reserved, must be kept at reset value.
5	CC2P	Channel 2 input capture output polarity Refer to CC1P description
4	CC2E	Channel 2 input/capture output enable Refer to CC1E description
3	CC1NP	Channel 1 input capture polarity When Channel 1 is configured as an output, this bit is invalid. When Channel 1 is configured as an input, the combined use of CC1P and CC1NP defines the input signal polarity and level; for details, refer to the ICx polarity/level selection table.
2	Reserved	Reserved, must be kept at reset value.
1	CC1P	Channel 1 input/capture output polarity When channel 1 is configured as output, the bit defines the output signal polarity: 0: OC1 is active at high level 1: OC1 is active at low level When Channel 1 is configured as an input, the combined use of CC1P and CC1NP determines the polarity and level of the input signal; for details, please refer to the ICx polarity/level selection table.
0	CC1E	Channel 1 input/capture output enable When channel 1 is configured as output: 0: Close. OC1 output disable 1: Enable. The OC1 signal is output to the corresponding output pin. CC1 channel is configured as input: This bit determines whether the input capture function is enabled. 0 : Capture disable 1 : Capture enable

In input mode, ICx polarity/level selection is given in the Table below:

Table 12-5 ICx polarity/level selection

CCxP	CCxNP	ICx polarity/level
0	0	Rising edge active/high level active
1	0	Falling edge active/low level active
1	1	Rising edge or falling edge active/low level active
0	1	Reserved

12.5.10 TIM3_CNT Counter

Address offset: 0x24

Reset value: 0x0000

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

Bit	Field	Description
15 : 0	CNT	Counter value

12.5.11 TIM3_PSC Prescaler

Address offset: 0x28

Reset value: 0x0000

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

Bit	Field	Description
15 : 0	PSC	Prescaler value Counter clock frequency (ck_cnt) = $f_{CK_PSC} / (PSC + 1)$ In case of update event, PSC value is loaded into the current prescaler register.

12.5.12 TIM3_ARR Auto Reload Register

Address offset: 0x2C

Reset value: 0x0000

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

Bit	Field	Description
15 : 0	ARR	Auto reload value These bits define the auto reload value of the counter. When auto reload value is 0, the counter doesn't work.

12.5.13 TIM3_CCR1 Capture/Compare Register 1

Address offset: 0x34

Reset value: 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR1															
rw															

Bit	Field	Description
15 : 0	CCR1	Channel 1 capture/compare value When channel 1 is configured as input: CCR1 value determines the counter value of the previous capture event (this register is only readable). When channel 1 is configured as output: If the preload function is not selected in TIM3_CCMR1 register (OC1PE bit), the write value is immediately transmitted to the current capture/compare shadow register. Or else, the write value is only loaded into the capture/compare register in case of update event. The current capture/compare shadow register is involved in the compare with the counter TIM3_CNT, and the compare result is reflected to the output signal of OC1 port.

12.5.14 TIM3_CCR2 Capture/Compare Register 2

Address offset: 0x38

Reset value: 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR2															
rw															

Bit	Field	Description
15 : 0	CCR2	Channel 2 capture/compare value Refer to CCR1 description.

12.5.15 TIM3_CCR3 Capture/Compare Register 3

Address offset: 0x3C

Reset value: 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR3															
rw															

Bit	Field	Description
15 : 0	CCR3	Channel 3 capture/compare value Refer to CCR1 description.

12.5.16 TIM3_CCR4 Capture/Compare Register 4

Address offset: 0x40

Reset value: 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR4															
rw															

Bit	Field	Description
15 : 0	CCR4	Channel 4 capture/compare value Refer to CCR1 description.

12.5.17 TIM3_OR Input Option Register

Address offset: 0x50

Reset value: 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								TI4_RMP		Reserved				ETR_RMP	
								rw						rw	

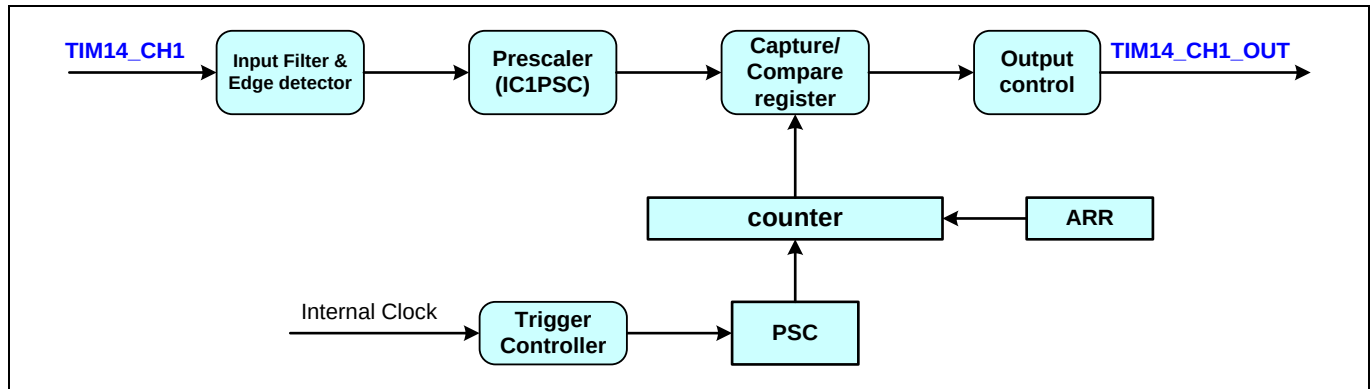
Bit	Field	Description
15:8	Reserved	Reserved, must be kept at reset value.
7:6	TI4_RMP	TI4 multiplex 00: CH4 GPIO input 01: LSI clock input 10: Reserved 11: HSE_CLK_DIV_128 clock input
1:0	ETR_RMP	ETR multiplex 00 : Reserved 01 : LSI clock input 10 : Reserved 11 : HSE_CLK_DIV_128 clock input

13 TIM14 Basic Timer Overview

TIM14 is composed of a 16 bit real-time programmable prescaler and a 16-bit auto reload counter, and can conveniently provide with the counting and timing function. The counter is driven by a programmable prescaler. The basic timer can be used for a variety of purposes, such as input capture function (measuring input signal pulse width or frequency, etc.), output compare function (PWM output, compare output, etc.).

13.2 Function block diagram

Figure 13-1 TIM14 block diagram



TIM14 structure block diagram is mainly composed of input capture unit, output compare unit, and counter unit.

13.3 Main characteristics

- 16 bit real-time programmable prescaler, division factor: 1-65536.
- 16 bit auto reload counter (counting direction: up).
- Input capture: measure input signal pulse width or period.
- Output compare (control output waveform or indicator counter already finishes timing).
- One-pulse mode.
- PWM output (edge alignment mode)
- Interrupt/DMA request event: update event, input capture, output compare.

13.4 Function description

13.4.1 Clock

13.4.1.1 Clock selection

The clock of the counter is provided by the internal clock (INT_CK).

13.4.1.2 Time base unit

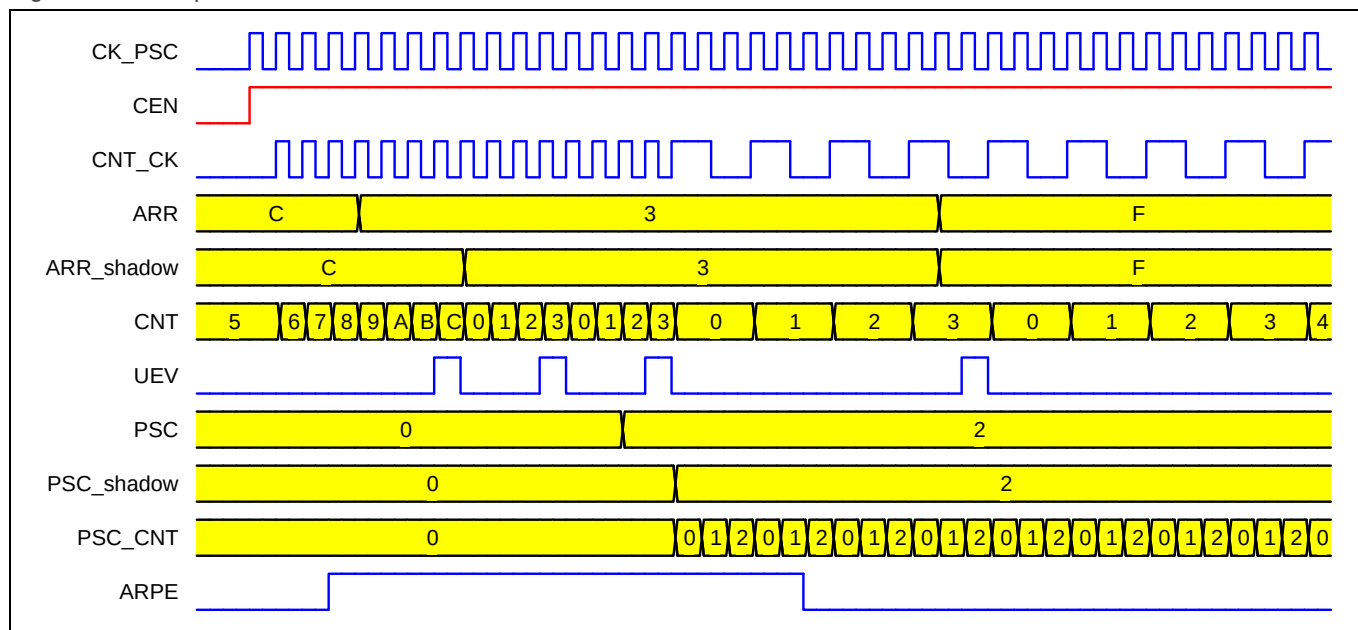
TIM14 time base unit mainly includes: Counter register (TIM14_CNT), prescaler register (TIM14_PSC), auto reload register (TIM14_ARR).

The counter unit is composed of a 16 bit counter with its related auto reload register, the counter can count up.

The counter clock is provided by prescaler. The prescaler is composed of the prescaler counter with its related register. The division factor is 1-65536. It can write any time, and gets active at the next update.

The auto preload register has a 16 bit shadow register with preload function. Set TIM14_CR1 register ARPE bit to select the content of ARR register are transferred into the shadow register permanently or at each update event.

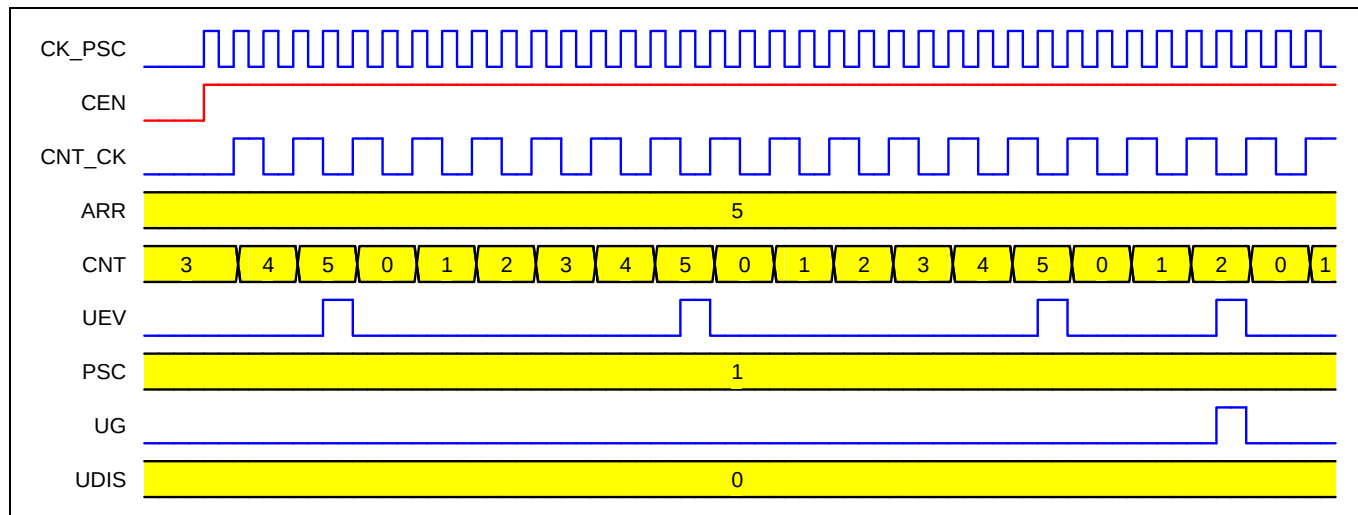
Figure 13-2 Auto preload



13.4.1.3 Counter modes

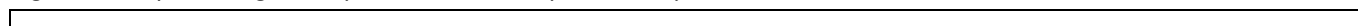
TIM14 supports only the up counting mode. Enable TIM14_CR1 register CEN, and the counter begins up count from 0 to TIM14_ARR value and generates a counter overflow event (update). The counter begins up count again from 0. Set TIM14_EGR register UG =1, also generate an update event.

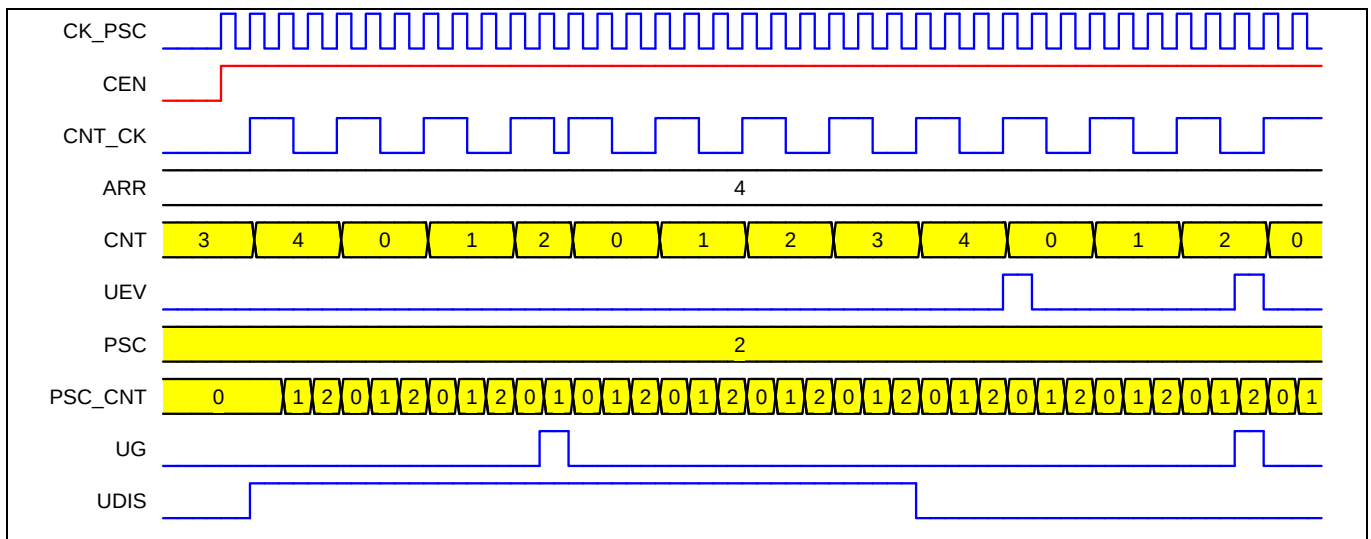
Figure 13-3 Up counting mode (UDIS=0)



Configuring TIM14_CR1 register UDIS=1, disable update event. In case of counter overflow event, the update event is not generated. Configuring UG=1, the update event is not generated. However, the counter and prescaler counter will initialize, and begin up count from 0.

Figure 13-4 Up counting mode (UDIS =1, disable update event)





Note: In case of update event:

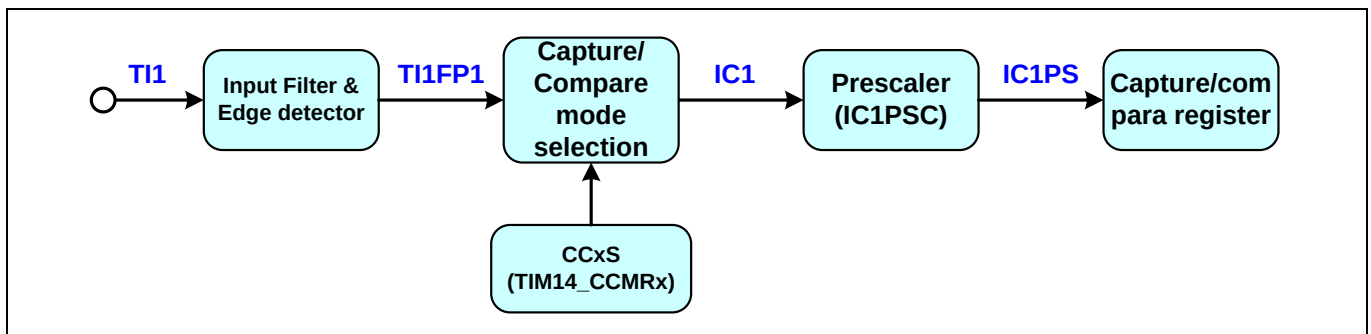
- ARR register value is loaded into ARR shadow register.
- Prescaler preload value becomes active.

13.4.2 Input capture

13.4.2.1 Input capture

The input capture is composed of digital filter, edge detector, prescaler, capture compare register, etc. Its block diagram:

Figure 13-1 TIM14 input capture block diagram



Configure TIM14_CCMR1 register IC1F, set the filter width of a digital filter (filter sampling frequency and digital filter width are shown in the table below). When the input signal width is greater than the filter width, the input signal is active. When the digital filter samples the input signal of the input pin TI1, generate a filtered signal TI1F. Through the edge detector with the optional polarity, generate an effective signal IC1. This signal generates IC1PS via the prescaler for trigger input capture event.

Table 13-1 Digital filter width and IC1F correlation

IC1F[3: 0]	Sample frequency and filter length	IC1F[3: 0]	Sample frequency and filter length
0000	No filter, sample by f_{DTS}	1000	Sampling frequency $f_{sampling}=f_{DTS} / 8$, $N=6$
0001	Sampling frequency $f_{sampling}=f_{INT_CK}$, $N=2$	1001	Sampling frequency $f_{sampling}=f_{DTS} / 8$, $N=8$
0010	Sampling frequency $f_{sampling}=f_{INT_CK}$, $N=4$	1010	Sampling frequency $f_{sampling}=f_{DTS} / 16 \cdot N=5$
0011	Sampling frequency $f_{sampling}=f_{INT_CK}$, $N=8$	1011	Sampling frequency $f_{sampling}=f_{DTS} / 16$, $N=6$
0100	Sampling frequency $f_{sampling}=f_{DTS} / 2$, $N=6$	1100	Sampling frequency $f_{sampling}=f_{DTS} / 16$, $N=8$

IC1F[3: 0]	Sample frequency and filter length	IC1F[3: 0]	Sample frequency and filter length
0101	Sampling frequency $f_{\text{sampling}}=f_{\text{DTS}}/2$, N=8	1101	Sampling frequency $f_{\text{sampling}}=f_{\text{DTS}}/32$, N=5
0110	Sampling frequency $f_{\text{sampling}}=f_{\text{DTS}}/4$, N=6	1110	Sampling frequency $f_{\text{sampling}}=f_{\text{DTS}}/32$, N=6
0111	Sampling frequency $f_{\text{sampling}}=f_{\text{DTS}}/4$, N=8	1111	Sampling frequency $f_{\text{sampling}}=f_{\text{DTS}}/32$, N=8

In the input capture mode, when detecting the active edge of signal IC1, the counter's current value is locked in the corresponding shadow register, and then loaded into the corresponding capture/compare register. When enabling interrupt or DMA, and producing the capture event, generate the corresponding interrupt or DMA request. At the capture event, the capture flag bit CC1IF of the status register (TIM14_SR) will be set. Configure CC1IF =0 or read TIM14_CCR1 data, clear CC1IF flag bit. When CC1IF is not cleared, next input capture event generated, the repeat capture flag CC1OF will be set. Configure CC1OF =0, can clear CC1OF flag bit.

For example, Sample TI1 input signal's active edge. At TI1 rising edge, capture the current counter value and lock it in the TIM14_CCR1 register, procedure steps as follows:

1. Configure TIM14_CCMR1 register CC1S=01, CC1 channel is configured as input, and IC1 is mapped on TI1.
2. Configure TIM14_CCMR1 register IC1F [3:0], configure the filter width of digital filter (configure upon need).
3. Configure TIM14_CCER register CC1P=0, select capture occurs in the rising edge of TI1 signal.
4. Configure TIM14_CCMR1 register IC1PSC [1:0], select the prescaler factor.
5. Configure TIM14_CCER register CC1E = 1, channel 1 capture enable.
6. Configure TIM14_DIER register CC1IE=1, enable channel 1 interrupt request; if the chip has the built in DMA, configure CC1DE=1 of the TIM14_DIER register to allow the DMA request on capture/compare channel 1.

Note :

- When the channel is configured as input mode, TIM14_CCR1 register becomes read only.

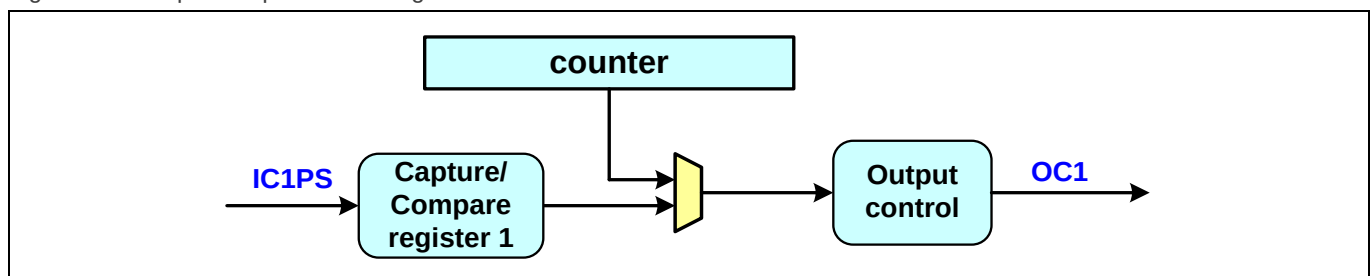
In case of more than two continuous captures and CC1IF flag is not cleared, the repeat capture flag

- CC1OF is set. In order to avoid losing the capture information that may be generated before the repeat capture flag CC1OF is set. It is recommended to read data before reading the repeat capture flag.
- Set the corresponding CC1G bit of TIM14_EGR register, generate the input capture interrupt or DMA request through software.

13.4.3 Output compare

The compare output of the capture/compare channel is composed of the comparator, output control circuit and capture/compare register. Its structure is shown below:

Figure 13-2 Output compare block diagram



In output compare mode, the content of capture/compare register is loaded into the shadow register, and then compare the content of shadow register with the current value of counter. The capture/compare module includes a capture/compare register (preload register) and a shadow register. When read and write, only programmed the capture/compare register.

13.4.3.1 Force output

Configure TIM14_CCMR1 register CC1S = 00 and set the channel CC1 as output mode. Configure TIM14_CCMR1 register OC1M bit, and directly enforce the output compare signal as active or inactive level, independently of the output compare result. Configure TIM14_CCMR1 register OC1M = 100, and enforce output compare signal as inactive level. And OC1REF is enforced as low level. Configure TIM14_CCMR1 register OC1M = 101, and enforce output compare signal as active level. And the OC1REF is enforced as high level (OC1REF always active as high level).

Note: In force output mode, TIM14_CCR1 shadow register and counter output compare is still performed. The corresponding flag bit of compare result will also be changed. When enable the corresponding interrupt and DMA request, it will also generate the corresponding interrupt and DMA request.

13.4.3.2 Output compare

In the output compare mode, when the values of counter and capture compare register are matches, different waveform can be exported according to the OC1M bit of TIM14_CCMR1 register.

For example, when the value of counter and capture/compare register matches, the functions in the output compare mode:

1. When compare match, OC1M value is different, output channel 1 signal OC1 run differently:

- ◆ OC1M = 000: OC1signal keeps its level.
- ◆ OC1M = 001: OC1 signal is set as active level.
- ◆ OC1M = 010: OC1 signal is set as inactive level.
- ◆ OC1M = 011: OC1 signal toggle.

2. When match, set the flag bit of status register (CC1IF bit of TIM14_SR register).

3. Configure TIM14_DIER register CC1IE =1, generate an interrupt when match.

4. Configure TIM14_DIER register CC1DE =1, generate a DMA request when match (only suitable for the built in DMA product).

The output compare mode can also be used to export a single pulse (one-pulse mode).

Configuration steps of channel 1 output compare mode:

1. Configure counter clock (select clock source, configure prescaler factor).

2. Configure TIM14_ARR and TIM14_CCR1 register.

3. Configure TIM14_DIER register CC1IE =1, enable capture/compare 1 interrupt.

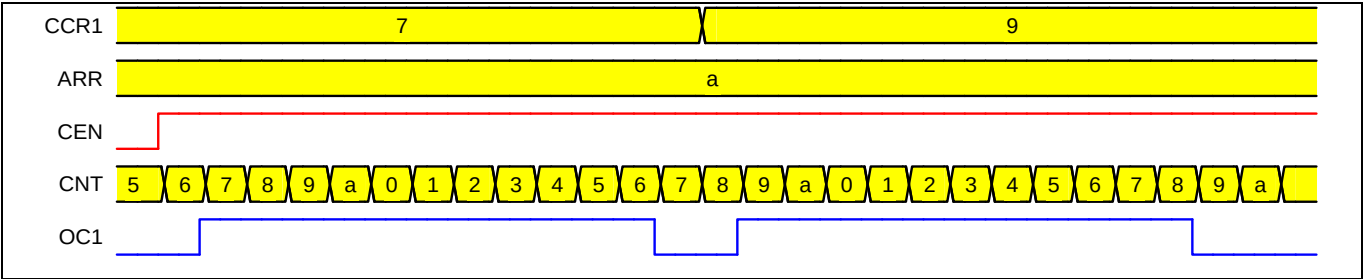
4. Configure output mode:

- ◆ Configure TIM14_CCMR1 register OC1M = 011, toggle when OC1 compare match.
- ◆ Configure TIM14_CCMR1 register OC1PE = 0, disable preload function of TIM14_CCR1 register.
- ◆ Configure TIM14_CCMR1 register CC1P = 1, OC1 is active low polarity.
- ◆ Configure TIM14_CCER register CC1E = 1, enable channel 1 output, OC1 signal is exported to the corresponding output pin.

5. Configure TIM14_CR1 register CEN =1, enable the counter.

Configure TIM14_CCMR1 register OC1PE=0, disable TIM14_CCR1 register preload function, write TIM14_CCR1 register any time, and the write value updated immediately. Configure TIM14_CCMR1 register OC1PE=1, enable TIM14_CCR1 register preload function, Updated only at the next update event. The figure below is an example.

Figure 13-3 Output compare mode, OC1 signals toggle when match



Note: In the output compare mode, the update event has no effect on output result. In the force output mode, the compare output between TIM14_CCR1 shadow register and counter still performed. The corresponding flag bit of compare result is also be changed. When enabling the corresponding interrupt and DMA request, it still generates the corresponding interrupt and DMA request.

13.4.3.3 PWM output

In the PWM mode, according to the value of TIM14_ARR register and TIM14_CCR1 register, generate a frequency and duty cycle controllable PWM waveform.

Configure OC1M=110 or OC1M=111 of TIM14_CCMR1 register corresponding to the channel 1, select channel 1 to PWM mode 1 or PWM mode 2. In the PWM mode, the counter will always compare with CCR1. According to configuration and compare results, the channel x exports different signals. Therefore, TIM14 can generate 1 PWM output signals with independent duty under the same frequency. In PWM mode, enable the preload function of TIM14_CCR1 and TIM14_ARR. The written value of TIM14_CCR1 preload register and TIM14_ARR preload register will become effective at the next update event, and load the corresponding shadow register. Therefore, in the PWM mode, before enabling counter, it's required to set TIM14_EGR UG=1, and generate the update event to initialize all registers.

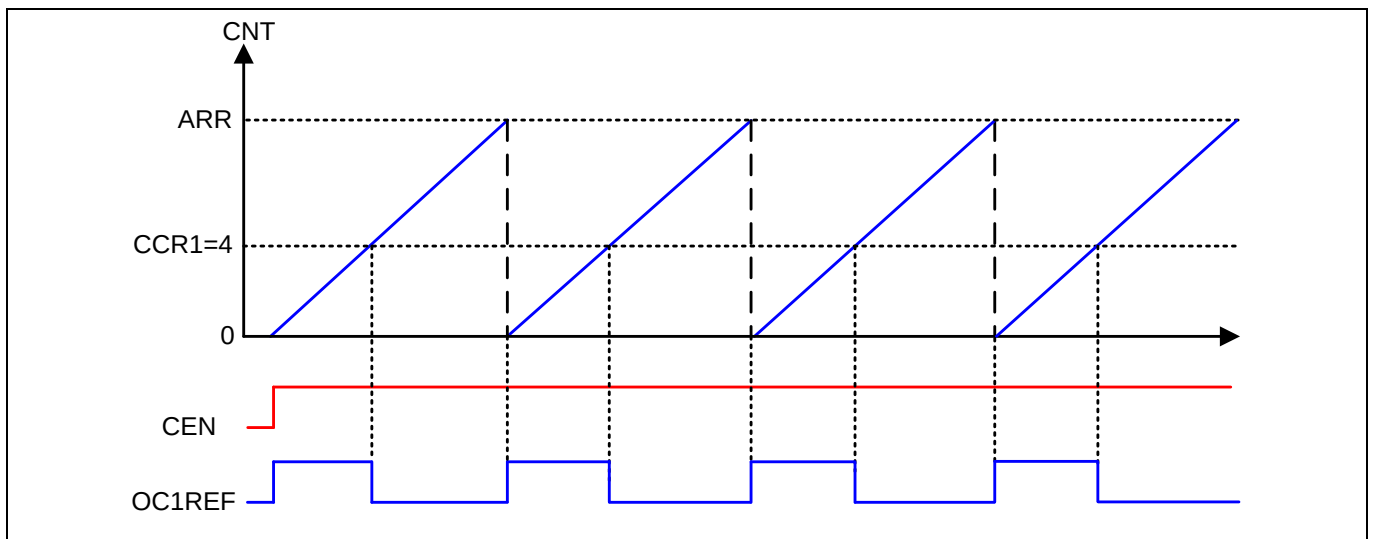
Configure TIM14_CCER register CC1P, select OC1's active polarity. Configure TIM14_CCER register CC1E to control OC1 output enable.

13.4.3.3.1 PWM edge alignment mode - up counting mode

Based on up counting mode configuration, configure TIM14_CCMR1 register CC1S =00, select output mode, OC1M =110, select PWM mode 1. In case of TIM14_CNT <TIM14_CCR1, channel 1 (OC1REF) is in active level. Otherwise, it's inactive level. If TIM14_CCR1 compare value is greater than the auto reload value (TIM14_ARR), OC1REF stays in active level. When the compare value is 0, OC1REF is at inactive level. The figure below is the example of waveform in PWM mode 1 during edge alignment up count in case of CCR1=4, and ARR =a.

Figure 13-4 Waveform in PWM mode 1 during edge alignment up count





13.4.3.4 One-pulse output

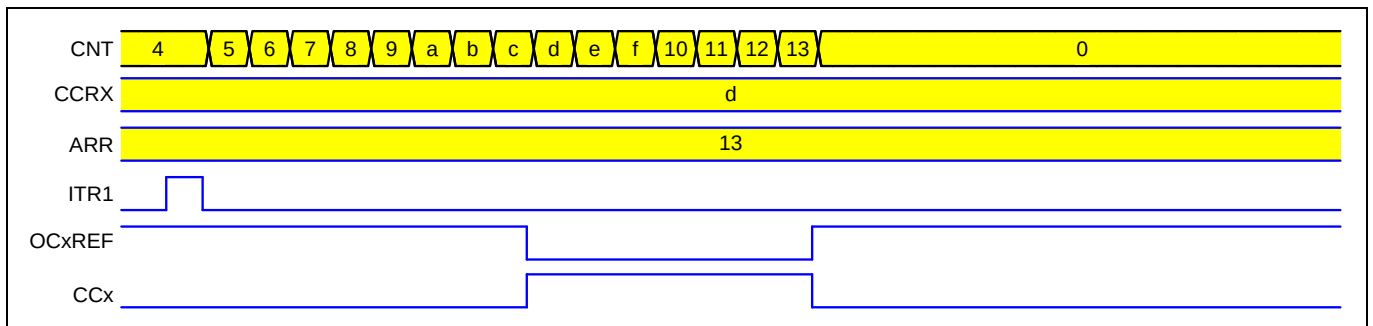
In the one-pulse mode (OPM), the counter responds a stimulus, and generate a pulse with width programmable. Configure TIM14_CR1 register OPM =1. Select one-pulse mode, trigger signal's active edge or configuring CEN =1 can enable the counter. CEN =1 remains until the next update event or configure CEN =0.

The necessary condition to generate pulse is the difference of compare value and initial value of the counter. Necessary configuration prior to counter enable:

- Up counting mode: Counter $CNT < CCR1 \leq ARR$.

The figure below is one pulse mode:

Figure 13-5 one-pulse mode



13.4.4 Debug mode

In the debug mode, configure DBG_CR register DBG_TIM14_STOP=1, TIM14 counter stops counting. (details in the debug section)

13.4.5 Interrupt

TIM14 interrupt includes: capture/compare 1 interrupt and update interrupt. When the corresponding interrupt enable bit is set, generate the corresponding event and interrupt.

Table 13-2 List of interrupt events

Interrupt event	Flag bit	Enable bit
Capture/compare 1 interrupt	CC1IF	CC1IE
Update interrupt	UIF	UIE

13.5 Register

Table 13-3 TIM14 register overview

Offset	Acronym	Register Name	Reset
0x00	TIM14_CR1	Control Register 1	0x0000
0x0C	TIM14_DIER	Interrupt Enable Register	0x0000
0x10	TIM14_SR	Status Register	0x0000
0x14	TIM14_EGR	Event Generation Register	0x0000
0x18	TIM14_CCMR1	Capture/Compare Mode Register 1	0x0000
0x20	TIM14_CCER	Capture/Compare Enable Register	0x0000
0x24	TIM14_CNT	Counter	0x0000
0x28	TIM14_PSC	Prescaler	0x0000
0x2C	TIM14_ARR	Auto Reload Register	0x0000
0x34	TIM14_CCR1	Capture/Compare Register 1	0x0000
0x44	TIM14_BDTR	Break And Dead-Time Register	0x0000

13.5.1 TIM14_CR1 Control Register 1

Address offset : 0x00

Reset value : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						CKD		ARPE	Reserved			OPM	URS	UDIS	CEN
						rw		rw				rw	rw	rw	rw

Bit	Field	Description
15 : 10	Reserved	Reserved, must be kept at reset value.
9 : 8	CKD	Clock division Division ratio between the timer clock (CK_INT) frequency and the dead-time and sampling clock used by the dead-time generators and the digital filters (TI1). 00 : $t_{DTS} = t_{INT_CK}$ 01 : $t_{DTS} = 2 \times t_{INT_CK}$ 10 : $t_{DTS} = 4 \times t_{INT_CK}$ 11 : Reserved, do not program this value
7	ARPE	Auto reload preload enable 0: Disable the shadow register of TIM14_ARR register 1: Enable the shadow register of TIM14_ARR register
6 : 4	Reserved	Reserved, must be kept at reset value.
3	OPM	one-pulse mode 0: Disable one-pulse mode. In case of update event, the counter count continues 1: Enable one-pulse mode. In case of the next update event (clear CEN bit), the counter count stops
2	URS	Update request source Software configures this bit, select update event source. 0: The event below may generate an update interrupt request: - Counter overflow - Set UG bit 1: Only in counter overflow, generate update interrupt request

Bit	Field	Description
1	UDIS	Update disable This bit is used to enable or disable the update event 0: Update event (UEV) enabled. 1: Update event disabled. The Update event is not generated, shadow registers keep their value (ARR, PSC, CCR1). However the counter and the prescaler are reinitialized if the EGR.UG bit is set.
0	CEN	Counter enable 0: Counter disabled 1: Counter enabled

13.5.2 TIM14_DIER Interrupt Enable Register

Address offset : 0x0C

Reset value : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved														CC1IE	UIE
														rW	rW

Bit	Field	Description
15 : 2	Reserved	Reserved, must be kept at reset value.
1	CC1IE	Enable capture/compare 1 interrupt 0: Capture/compare interrupt 1 disable 1: Capture/compare interrupt 1 enable
0	UIE	Enable update interrupt 0: Update event interrupt disable 1: Update event interrupt enable

13.5.3 TIM14_SR Status Register

Address offset : 0x10

Reset value : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						CC1OF	Reserved							CC1IF	UIF
						r_w0c								r_w0c	r_w0c

Bit	Field	Description
15 : 10	Reserved	Reserved, must be kept at reset value.
9	CC1OF	Capture/compare 1 over capture flag Only when the channel 1 is configured as input capture mode and CC1F is configured as 1, and the capture event occurs again. The flag is set by hardware. Write 0 may clear the bit. 0: No overcapture 1: Overcapture
8 : 2	Reserved	Reserved, must be kept at reset value.
1	CC1IF	Capture/compare 1 interrupt flag Channel 1 in output mode: When the counter value and compare value match, the bit is set by hardware, It's cleared by software. 0: No match 1: TIM14_CNT value and TIM14_CCR1 value match Channel 1 as output mode: In case of capture event, the bit is set by hardware. It's cleared by software or reading TIM14_CCR1 register 0: No input capture occurred 1: Counter value is captured to TIM14_CCR1

Bit	Field	Description
0	UIF	Update interrupt flag In case of update event, the bit is set 1 by hardware. It's cleared 0 by software. 0: No update interrupt occurred 1: update interrupt pending TIM1_EGR register UG =1 or counter overflow generate update event.

13.5.4 TIM14_EGR Event Generation Register

Address offset : 0x14

Reset value : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved														CC1G	UG
														W	W

Bit	Field	Description
15 : 2	Reserved	Reserved, must be kept at reset value.
1	CC1G	Capture/compare 1 generation This bit is set by software. It generates a capture/compare event, and is auto cleared by hardware. 0: No action 1 : Generate a capture/compare event in channel CC1: When the channel CC1 is configured as output: Set CC1IF =1. When enabling the corresponding interrupt, generate the corresponding interrupt. When the channel CC1 is configured input: Set CC1IF =1. When enabling the corresponding interrupt, generate the corresponding interrupt. When CC1IF is set, set CC1OF =1.
0	UG	Update generation 0 : No action 1: Initialize the counter, and generate an update event. It's auto cleared by hardware.

13.5.5 TIM14_CCMR1 Capture/Compare Mode Register 1

Address offset : 0x18

Reset value : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Reserved								Res.	OC1M			OC1PE	Res.	CC1S		
								IC1F				IC1PSC		CC1S		
								rw	rw			rw		rw	rw	

The channel may be used for input (capture mode) or output (compare mode). The channel direction is defined by the corresponding CC1S. The register's other bits functions differently in the input mode and output mode. OC1x describes the channel function in the output mode. IC1x describes the channel function in the input mode.

Output compare mode:

Bit	Field	Description
15 : 7	Reserved	Reserved, must be kept at reset value.

Bit	Field	Description
6 : 4	OC1M	Output compare mode The bit has defined the output reference signal OC1REF action. OC1REF has determined OC1 value. OC1REF is active at high level. The active level of OC1 depends on CC1P bit. 000 : Freeze. TIM14_CCR1 and TIM14_CNT compare results has no effect on OC1REF. 001: Set as high when configuration. When TIM14_CNT value and TIM14_CCR1 value are same, enforce OC1REF as high level 010: Set as low when configuration. When TIM14_CNT value and TIM14_CCR1 value are same, enforce OC1REF as low level 011: Toggle when match. When TIM14_CCR1=TIM14_CNT, OC1REF toggle. 100: Enforce as low. Enforce OC1REF at low level 101: Enforce as high. Enforce OC1REF at high level 110 : PWM mode 1. During up count, in case of TIM14_CNT<TIM14_CCR1, enforce OC1REF is at high level. Or else, it's at low level. 111 : PWM mode 2. During up count, in case of TIM14_CNT<TIM14_CCR1, channel 1 enforce OC1REF is at low level. Or else, it's at high level. Note: In PWM mode 1 or PWM mode 2, only when the compare result changes or it changes over from the freeze mode to PWM mode in the output compare mode, OC1REF level may change.
3	OC1PE	Output compare preload enable 0: Disable TIM14_CCR1 register preload function. The value written into TIM14_CCR1 register becomes valid immediately 1: Enable TIM14_CCR1 register preload function. Only read and write the preload register. TIM14_CCR1 preload value becomes valid during the update event Note: Only in the one-pulse mode (TIM14_CR1 register OPM= 1), it has no influence whether the preload register is set. Under other scenarios, it's required to set the preload register. Otherwise, the follow up action is not certain.
2	Reserved	Reserved, must be kept at reset value.
1 : 0	CC1S	Channel 1 Capture/Compare selection This bit defines the channel direction and input signal selection. Only when the channel closes, these bits can be written: 00 : Channel 1 is configured as output 01 : Channel 1 is configured as input 10 : Reserved 11 : Reserved

Input capture mode

Bit	Field	Description
15 : 8	Reserved	Reserved, must be kept at reset value.

Bit	Field	Description
7 : 4	IC1F	Channel 1 input capture filter The digital filter is composed of an event counter. It records an output jump after N input events. These bits define IC1 input signal sampling frequency and digital filter length. 0000 : No filter, f_{DTS} sampling 0001 : Sampling frequency $f_{sampling}=f_{INT_CK} \cdot N=2$ 0010 : Sampling frequency $f_{sampling}=f_{INT_CK} \cdot N=4$ 0011 : Sampling frequency $f_{sampling}=f_{INT_CK} \cdot N=8$ 0100 : Sampling frequency $f_{sampling}=f_{DTS} / 2 \cdot N=6$ 0101 : Sampling frequency $f_{sampling}=f_{DTS} / 2 \cdot N=8$ 0110 : Sampling frequency $f_{sampling}=f_{DTS} / 4 \cdot N=6$ 0111 : Sampling frequency $f_{sampling}=f_{DTS} / 4 \cdot N=8$ 1000 : Sampling frequency $f_{sampling}=f_{DTS} / 8 \cdot N=6$ 1001 : Sampling frequency $f_{sampling}=f_{DTS} / 8 \cdot N=8$ 1010 : Sampling frequency $f_{sampling}=f_{DTS} / 16 \cdot N=5$ 1011 : Sampling frequency $f_{sampling}=f_{DTS} / 16 \cdot N=6$ 1100 : Sampling frequency $f_{sampling}=f_{DTS} / 16 \cdot N=8$ 1101 : Sampling frequency $f_{sampling}=f_{DTS} / 32 \cdot N=5$ 1110 : Sampling frequency $f_{sampling}=f_{DTS} / 32 \cdot N=6$ 1111 : Sampling frequency $f_{sampling}=f_{DTS} / 32 \cdot N=8$
3 : 2	IC1PSC	Channel 1 input capture prescaler This bit defines the ratio of the prescaler acting on IC1. The prescaler is reset as soon as CC1E=0 (TIM14_CCER register). 00: No prescaler, the capture input detects each edge to trigger one capture 01: Trigger one capture for every 2 events 10: Trigger one capture for every 4 events 11: Trigger one capture for every 8 events
1 : 0	CC1S	Channel 1 capture/Compare selection This bit defines the channel direction and input signal selection. Only when the channel closes, these bits can be written: 00: Channel 1 is configured as output 01: Channel 1 is configured as input 10: Reserved 11: Reserved

13.5.6 TIM14_CCER Capture/Compare Enable Register

Address offset : 0x20

Reset value : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												CC1NP	Res.	CC1P	CC1E
												r/w		r/w	r/w

Bit	Field	Description
15 : 4	Reserved	Reserved, must be kept at reset value.
3	CC1NP	Channel 1 input capture 1 polarity This bit is not valid when Channel 1 is configured as an output. When channel 1 is configured as input, CC1P/CC1NP match use has defined the input signal polarity and level. Details are given in the IC1 polarity/level selection table.
2	Reserved	Reserved, must be kept at reset value.
1	CC1P	Channel 1 capture/Compare output polarity When Channel 1 is configured as an output, this bit defines the output signal polarity. 0: OC1 active at high level 1: OC1 active at low level When channel 1 is configured as input, CC1P/CC1NP match use has defined the input signal polarity and level. Details are given in the IC1 polarity/level selection table.

Bit	Field	Description
0	CC1E	Channel 1 capture/compare output enable When channel 1 is configured as an output. 0: off. OC1 output is disabled 1: on. OC1 signal is output to the corresponding output pin CC1 Channel configured as input. This bit determines whether the input capture function is enabled. 0: Capture disable 1: Capture enable

The polarity/level selection of IC1 in the input mode is shown in the following table:

Table 13-4 IC1 polarity/level selection table

CC1P	CC1NP	IC1 polarity/level
0	0	Rising edge active/high level active
1	0	Falling edge active/low level active
1	1	Rising or falling edge active/low level active
0	1	Reserved

13.5.7 TIM14_CNT Counter

Address offset : 0x24

Reset value : 0x0000

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

Bit	Field	Description
15 : 0	CNT	Count value

13.5.8 TIM14_PSC Prescaler

Address offset : 0x28

Reset value : 0x0000

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

Bit	Field	Description
15 : 0	PSC	Prescaler value Counter clock frequency (ck_cnt) = $f_{CK_PSC} / (PSC + 1)$ In case of an update event, PSC value is loaded into the current prescaler register.

13.5.9 TIM14_ARR Auto Reload Register

Address offset : 0x2C

Reset value : 0x0000

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

Bit	Field	Description
15 : 0	ARR	Auto-reload value

These bits define the auto reload value of the counter. When auto reload value is 0, the counter doesn't work.

13.5.10 TIM14_CCR1 Capture/Compare Register 1

Address offset : 0x34

Reset value : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR1															
rw															

Bit	Field	Description
15 : 0	CCR1	Channel 1 capture/compare value When channel 1 is configured as input: CCR1 value determines the counter value of the previous capture event (this register is only readable). When channel 1 is configured as output: If the preload function is not selected in TIM14_CCMR1 register (OC1PE bit), the write value is immediately transmitted to the current capture/compare shadow register. Or else, the write value is only loaded into the capture/compare register in case of update event. The current capture/compare shadow register is involved in the compare with the counter TIM14_CNT, and the compare result is reflected to the output signal of OC1 port.

13.5.11 TIM14_BDTR Break and Dead-Time Register

Address offset : 0x44

Reset value : 0x0000

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

Bit	Field	Description
15	MOE	Main output enable 0 : Disable OC1 output or enforce as the idle state 1: If the corresponding enable bit (CC1E of TIM14_CCER register) is set, the OC1 output is turned on
14 : 0	Reserved	Reserved, must be kept at reset value.

14 IWDG Independent watchdogIntroduction

The built-in independent watchdog offers enhanced safety, precise timing, and operational flexibility. It is used to detect and resolve malfunctions caused by software errors, triggering a system reset when the counter reaches a specified timeout value. The independent watchdog (IWDG) is driven by a dedicated low-speed clock (LSI), ensuring it remains functional even if the main clock fails.

The independent watchdog is best suited to applications which require the watchdog to run as a totally independent process outside the main application, but have lower timing accuracy constraints.

14.2 Main performance of IWDG

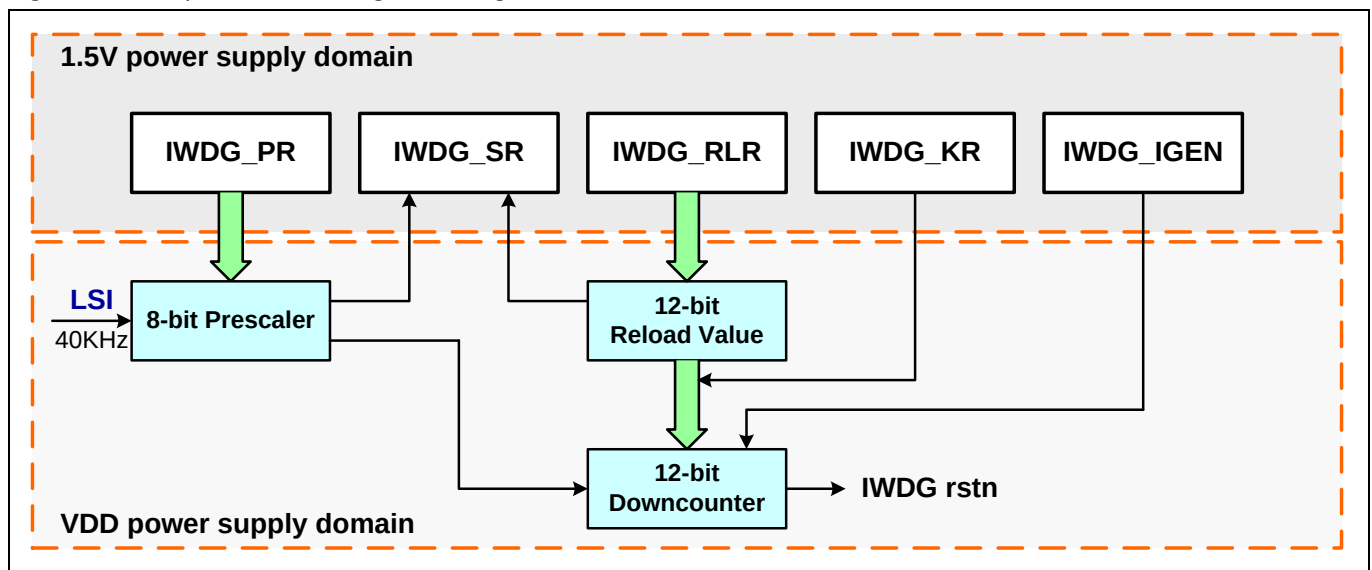
- Free running down counter
- The clock is provided by an independent oscillator (can operate in shutdown mode)
- After the watchdog is activated, a reset is generated when the counter counts to 0x0000.

14.3 IWDG Function Description

Write 0xCCCC in the key register (IWDG_KR). Start activating the independent watchdog; At this point, the counter starts to count down from its reset value 0xFFFF. When the counter reaches the end of 0x000, a reset signal (IWGD_RESET) will be generated. At any time, as long as the key register IWDG_KR_ Write 0xAAAA and IWDG in KR_ The value in RLR will be reloaded into the counter to avoid a watchdog reset.

The following figure shows the functional block diagram of the independent watchdog module.

Figure 14-1 Independent Watchdog Block Diagram



Note: The watchdog function is in the VDD power supply area, which means it can still operate normally in shutdown mode.

Table 14-1 IWDG timeout (For example, the LSI clock frequency is 40 KHz)

Prescale Coefficient	PR[2:0] Bits	Min time (ms) RL[11:0]=0x000	Max time (ms)
/4	0	0.1	409.6
/8	1	0.2	819.2
/16	2	0.4	1638.4
/32	3	0.8	3276.8
/64	4	1.6	6553.6
/128	5	3.2	13107.2
/256	(6 or 7)	6.4	26214.4

Note: These times are given based on a 40 KHz clock. In fact, the oscillator frequency inside the MCU will vary between 30 KHz and 60 KHz. In addition, even if the frequency of the oscillator is precise, the exact

timing still depends on the phase difference between the APB interface clock and the oscillator clock, so there will always be a complete oscillator cycle that is uncertain.

14.3.1 Hardware Watchdog

If the user activates the 'Hardware Watchdog' function in selecting bytes (please refer to the 'Embedded Flash' section), the watchdog will automatically start running after the system is powered on and reset; If the software does not write the corresponding value to the key register before the counter count ends, the system will generate a reset.

14.3.2 Register Access Protection

IWDG_PR, IWDG_RLR and IWDG_IGEN register has write protection function. To modify the values of these three registers, you must first write 0x5555 to the IWDG_KR register. Writing different values to this register will disrupt the order of operations and the register will be protected again. The reload operation (i.e. writing 0xAAAA) will also activate the write protection function. The status register indicates whether the pre division value and down counter are being updated.

14.3.3 Debug mode

When the microcontroller enters the Debug menu (CPU core stops), according to the DBG in the debugging module_ IWDG_ The status of the STOP configuration bit allows the IWDG counter to continue working or stop. Please refer to the chapter on debugging module for details.

14.4 Register

14.4.1 Overview of registers

Table 14-2 Overview of IWDG registers

Offset	Acronym	Register Name	Reset
0x00	IWDG_KR	Key register	0x00000000
0x04	IWDG_PR	Prescaler register	0x00000000
0x08	IWDG_RLR	Reload register	0x00000FFF
0x0C	IWDG_SR	Status register	0x00000000
0x10	IWDG_CR	Control register	0x00000000
0x14	IWDG_IGEN	Interrupt generate register	0x00000FFF
0x18	IWDG_CNT	Counter register	0x00000000

14.4.2 IWDG_KR Key Register

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
KEY															
W															

Bit	Field	Description
31:16	Reserved	Reserved, must be kept at reset value

Bit	Field	Description
15:0	KEY	Key value (write-only register) These bits must be written by software at regular intervals with 0xAAAA to feed the dog, otherwise a reset signal is generated to reset the system when the counter decrements to 0x0000. Writing 0x5555 enables access to the IWDG_PR, IWDG_RLR, IWDG_IGEN, and IWDG_CR [IRQ_SEL] registers. Writing 0xCCCC to start the watchdog.

14.4.3 IWDG_PR Prescaler Register

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.													PR		
													rw		

Bit	Field	Description
31:3	Reserved	always read as 0
2:0	PR	Prescaler divider These bits have write protection settings. Select the prescalerpre factor of the counter clock by setting these bits. To change the prescaler factor, the PVU bit of the IWDG_SR register must be 0. 000: prescaler divider = 4 100: prescaler divider = 64 001: prescaler divider = 8 101: prescaler divider = 128 010: prescaler divider = 16 110: prescaler divider = 256 011: prescaler divider = 32 111: prescaler divider = 256 Note: Performing a read operation on this register will return the prescaler frequency value from the VDD voltage domain. If a write operation is in progress, the value read back may be invalid. Therefore, the read value is valid only when the PVU bit of the IWDG_SR register is 0.

14.4.4 IWDG_RLR Reload Register

Address offset: 0x08

Reset value: 0x0000 0FFF

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.				RL											
				rw											

Bit	Field	Description
31:12	Reserved	Reserved, must be kept at reset value
11:0	RL	Watchdog counter reload value These bits are write-protected. They are used to define the reload value for the watchdog counter; whenever 0xAAAA is written to the IWDG_KR register, the reload value is transferred to the counter. The counter then begins counting down from this value. The counter then begins counting down from this value. The watchdog timeout period can be calculated using this reload value and the clock prescaler value. Note: Performing a read operation on this register will return the prescaler frequency value from the VDD voltage domain. If a write operation is in progress, the value read back may be invalid. Therefore, the read value is valid only when the RVU bit of the IWDG_SR register is 0.

14.4.5 IWDG_SR Status Register

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.												UPDATE	IVU	RVU	PVU
												r	r	r	r

Bit	Field	Description
31:4	Reserved	Reserved, must be kept at reset value
3	UPDATE	Watchdog reload value update flag This bit is set to '1' by hardware to indicate that 0xAAAA has been written to the IWDG_KR register. When the watchdog reload value is written to the counter, the counter is updated, and this bit is cleared to '0' by hardware.
2	IVU	Watchdog Interrupt Generate value update This bit is set by hardware to indicate that an update of the interrupt generate value is ongoing. It is cleared by hardware when the interrupt generate value update operation is completed in the VDD voltage domain (takes up to 5 oscillator cycles at LSI). The interrupt generate value can be updated only when IVU bit is cleared.
1	RVU	Watchdog counter reload value update This bit is set when an update of the reload value is ongoing. It is cleared when the reload value update operation is completed (takes up to 5 oscillator cycles at LSI). The reload value can be updated only when RVU bit is cleared.
0	PVU	Watchdog prescaler value update This bit is set when an update of the prescaler value is ongoing. It is cleared when the prescaler value update operation is completed (takes up to 5 oscillator cycles at LSI). The prescaler value can be updated only when PVU bit is cleared.

Note: If multiple reload values, pre frequency values, or interrupt generation values are used in the application, the pre load value must be changed again after the RVU bit is cleared, the pre frequency value can be changed again after the PVU bit is cleared, and the interrupt generation value must be changed again after the IVU bit is cleared. However, after the pre division or reinstallation values are updated, there is no need to wait for the RVU or PVU to reset, and the following code can continue to be executed.

14.4.6 IWDG_CR Control Register

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
Res.													IRQ_CLR	IRQ_SEL	
													rW	rW	

Bit	Field	Description
31:2	Reserved	Reserved, must be kept at reset value
1	IRQ_CLR	IWDG interrupt clear 1: Interrupt cleared 0: No effect
0	IRQ_SEL	IWDG overflow operation select 1: interrupt generation enabled after overflow 0: reset generation enabled after overflow

14.4.7 IWDG_IGEN Interrupt Generate Register

Address offset: 0x14

Reset value: 0x0000 0FFF

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.				IGEN											
				rw											

Bit	Field	Description
31:12	Reserved	Reserved, must be kept at reset value
11:0	IGEN	IWDG Interrupt Generate value These bits are write protected. Used to define the threshold for the watchdog counter to generate an interrupt. Whenever the counter value decreases to the threshold, an interrupt is generated. This register can only be modified when the IVU bit in the IWDG_SR register is 0. Note: Performing a read operation on this register will return the interrupt generated value from the VDD voltage domain. If a write operation is in progress, the return value may be invalid. Only when the IVU bit of the IWDG_SR register is 0, the return value must be valid.

14.4.8 IWDG_CNT Counter Register

Address offset: 0x18

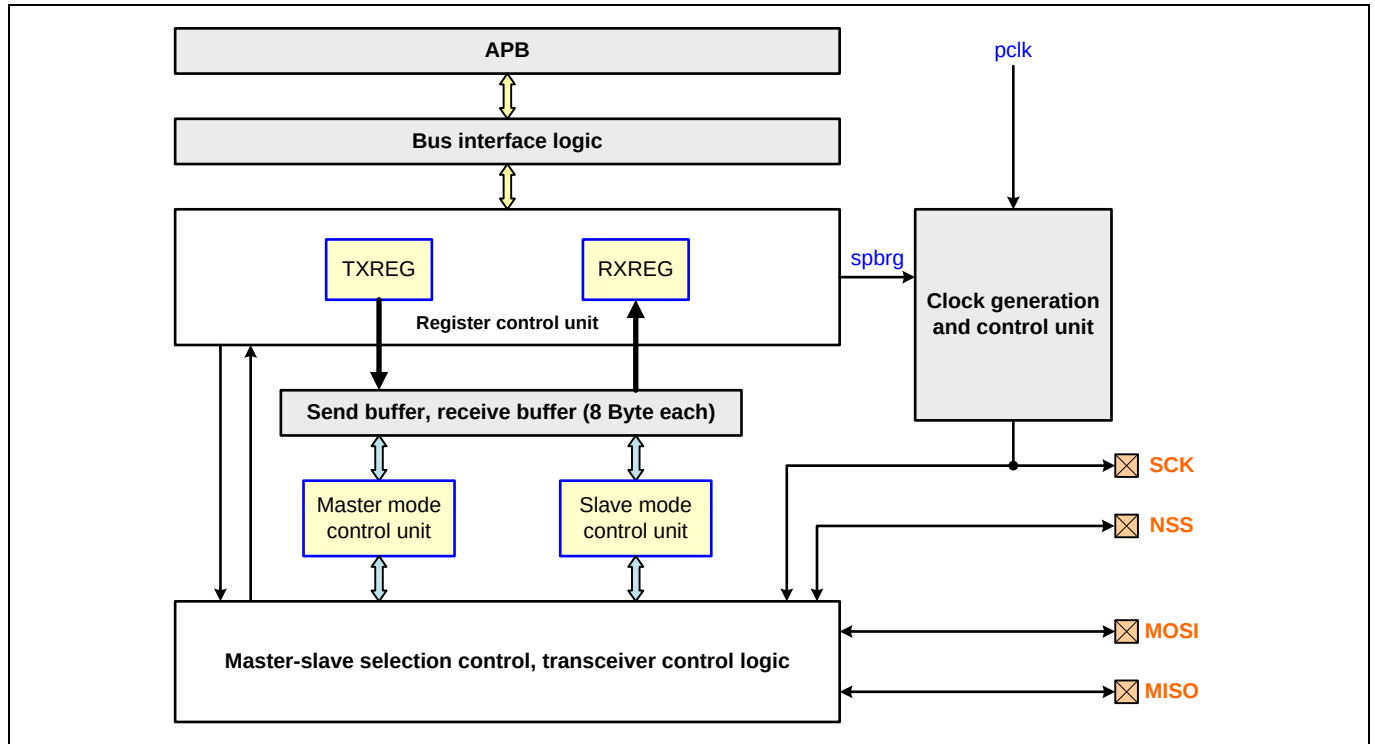
Reset value: 0x0000 0001(reset by RCC and system)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.													IWDG_CNT		
													r		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IWDG_CNT								IWDG_PS							
r								r							

Bit	Field	Description
31:19	Reserved	Reserved, must be kept at reset value
18:8	IWDG_CNT	IWDG counter value
7 : 0	IWDG_PS	Value of the prescaler counter of the IWDG clock

15 SPI Serial Peripheral Interface Function block diagram

Figure 15-1 SPI function block diagram



15.2 Introduction

The SPI (Serial Peripheral Interface) interface is widely used on board level communication between different devices, such as extended serial Flash and ADC. Many IC manufacturers produce devices that support SPI interfaces.

SPI allows the MCU to communicate with external devices in full duplex, synchronous and serial mode. The application software can communicate by the query status or SPI interrupt.

15.3 SPI function description

15.3.1 Overview

SPI supports the simultaneous receipt and transmission of 1~32 bit data. SPI can be configured as the slave mode or the master mode under the host environment. The software configures the CPOL bit and the CPHA bit of the universal control register (CCTL) and select four possible transmission sequences among the clock and data. Configure LSBFE bit to select whether MSB is in front or LSB is in front for data transmission.

SPI transmits data in the rising edge or falling edge of SCK, and receives data in the effective edge of the opposite clock.

SPI is used to exchange data. Data must be read after the transmission, even if the data is not valid data. Besides, the clock phase bit and polarity of the host and other communication slave unit must be same.

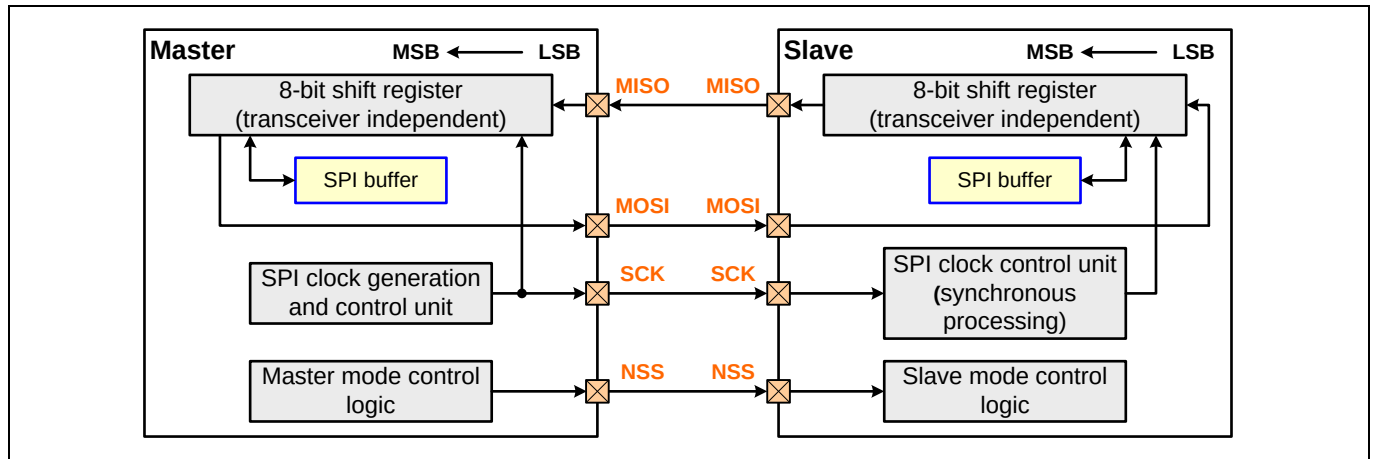
SPI is connected with the external device via 4 pins:

1. **MISO**: Master equipment input, slave device output pin. The transmission direction is from the slave equipment to the master equipment.
2. **MOSI**: Master equipment output, slave device input pin. The transmission direction is from the master equipment to the slave equipment.
3. **SCK**: SCK: Serial port clock, generated by master equipment and transmitted to the slave equipment.

4. **NSS**: Slave equipment selection. This pin is the selectable function under SPI protocol for master equipment to select the slave equipment to communicate with it (when there are multiple slave equipment) to achieve the one-to-one communication between the master and slave equipment and avoid the conflict between equipment on the data line. When the NSS pin function is activated, configure SPI_I2S_GCTL.MODE as '1' and SPI works in the master mode. Configure SPI_I2S_NSSR.NSS as '0' and NSS pin exports the low level, thereby enabling data communication between the slave equipment connected with the master equipment (configure SPI_I2S_GCTL.MODE as '0') and it.

The figure below shows one-to-one interconnected communication between the master equipment and slave equipment:

Figure 15-2 Single master and slave application



Same pins of SPI master and slave equipment are connected. The data transmission in the figure is serial communication from the highest valid bit to the lowest valid bit.

The master equipment is responsible for initiating the communication request and the slave equipment is responsible for response. The slave equipment gets the clock signal provided by master equipment via SCK pin, thereby enabling master and slave equipment to use the same clock for synchronous full duplex communication.

For the slave equipment, MOSI pin inputs the transmission data from the master equipment, and MISO pin exports the response data to the master equipment.

15.3.1.1 Phase bit and polarity of clock signal

CPOL and CPHA bit of SPI_CCTL register respectively control the clock polarity and phase bit of the clock. By combination, 4 sequence relations of clock/data can be made available.

Clock polarity refers to the level of SCK clock under the clock idle status: When configuring CPOL bit as '0', SCK clock keeps the low level under the idle status; otherwise, SCK clock keeps the high level under the idle status. Both master and slave equipment will be affected by CPOL control bit.

The clock phase bit determines the input data sampling sequence: When configuring CPHA bit as '0', the first data bit will be sampled in the second clock edge of SCK; otherwise, the first data bit will be sampled in the first clock edge of SCK.

Besides, CPHASEL bit is '0' after system power on and reset. When the software adjusts this bit as '1', change the data sampling sequence and CPHA bit function may change. For example during CPHASEL=1, CPHA=0: The first data bit will be sampled in the first clock edge of SCK (CPOL bit '0' rising edge; '1' falling edge).

Therefore, the expected use of clock/data sequence relations should be based on the configuration of CPOL, CPHA and CPHASEL bits.

Precautions for the time sequence configuration:

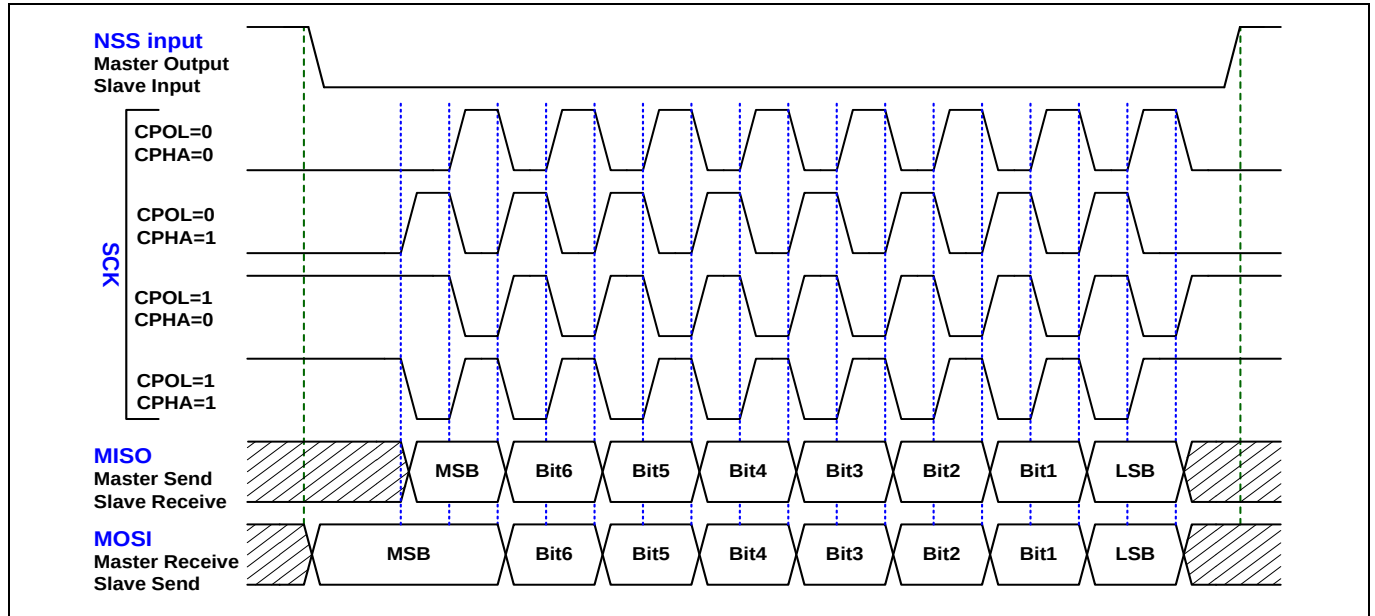
SPI cannot change CPOL/CPHA during work (When change is needed, first close the SPI enable bit SPIEN).

The master/slave equipment communicate in step. Therefore, the time sequence configuration should be consistent.

In the idle status, SCK level must be consistent with the corresponding polarity of CPOL configuration.

The figure below illustrates 4 scenarios of different CPHA and CPOL bit combinations during SPI transmission (configure CPHASEL bit as '1'), as well as the sequence of master/slave equipment SCK/MISO/MOSI/NSS pins.

Figure 15-3 Data clock sequence



It should be noted, configure SPI_GCTL.NSS=1, namely when NSS pin function is auto controlled by hardware. After data communication, the hardware will auto pull up this pin (as illustrated in the figure above). When configuring SPI_GCTL.NSS=0, NSS output status will be controlled by the NSS bit of register SPI_NSSR from the chip (changeover of NSS pin output status by software).

15.3.1.2 High speed transmission

With response to the board level sensitivity in the high speed transmission mode, configure SPI_CCTL register TXEDGE/RXEDGE bit to adjust the time point of data transmission/receipt sampling. During high speed transmission, the baud rate generator for the master equipment is configured at SPBRG≤4 and output SCK clock is quicker (≥10MHz); during low speed transmission, the baud rate generator for the master equipment is configured at SPBRG>4, and output SCK clock is slower (<10MHz).

In the slave mode, when TXEDGE bit is '1', once the software configures TXREG to write data, rather than waiting for SCK clock input edge, make transmission to the MISO pin line; when TXEDGE bit is '0', the slave equipment will always wait for a valid clock edge before transmitting data to MISO pin line.

15.3.1.3 Data frame format

Configure LSBFE bit of SPI_CCTL register, and determine the data bit output sequence, from the lowest valid bit to the highest valid bit when the LSBFE bit is '1' or from the highest valid bit to the lowest valid bit in case of '0' (default).

Configure SPILEN bit of SPI_CCTL register, and determine the data frame length. When SPILEN bit is '1' (default), the data frame length is 8 bit. In case of '0', the data frame length is 7 bits. SPI transmission and receipt is controlled by the data frame format configuration.

Besides, configure SPI_GCTL.DW8_32=1 and register SPI_EXTCTL to achieve the data format of any frame length (frame length range within 1~32 bit); any frame length configuration also supports LSBFE bit function (LSB or MSB preferred).

15.3.2 Main characteristics of SPI

- Fully compatible with Motorola SPI specification
- Support full duplex sync transmission in 3 lines
- Bit programmable baud rate generator
- Support master and slave mode
- Support communication between one host and multiple slave units

- When the SPI is in the master/slave mode, the clock can be up to PCLK/2 、PCLK/4 (PCLK as APB clock)
- Programmable clock polarity and phase bit
- Programmable data frame length (fixed 8 bit or 7 bit frame length, any frame length of 1~32 bits)
- Programmable data sequence, MSB is in front or LSB is in front (support transmission and receipt of data with any frame length of 1~32 or fixed frame length)
- 4 byte receipt/ transmission buffer, the following interrupt event or status should be available for software configuration and use:
 - ◆ Transmission buffer is empty
 - ◆ Transmission buffer and transmission shift register are simultaneously empty
 - ◆ Transmission end underflow
 - ◆ Receive valid byte
 - ◆ Receipt buffer overflow
 - ◆ Receipt buffer full
 - ◆ Receive specified byte in the master mode

15.3.3 SPI slave mode

When SPI serves as the slave equipment, SCK pin put comes from the serial clock of the master equipment. Therefore, when the baud rate generator is not used while the slave equipment works, it's not necessary to configure the register SPI_SPBRG (invalid under the slave equipment).

15.3.3.1 Configuration steps

1. Set SPI_GCTL.SPILEN to define the data frame format as 7 bit or 8 bit.
2. Configure the CPOL, CPHA/CPHASEL bit of the register SPI_CCTL to determine the sequence mode.
3. Configure SPI_CCTL.LSBFE, and determine the data frame receipt and transmission sequence (LSB or MSB bit preferred).
4. Configure MODE bit of register SPI_GCTL as '0' (slave mode), SPIEN bit as '1' (SPI function enable), and configure GPIO function pins required for SPI operation.
5. Configure TXEN and RXEN bit of register SPI_GCTL as '1', and open transmission and receipt permission (during transmission, it's required to write data into the register SPI_TXREG). SPI will receive MOSI pin data under the slave mode, and export data from MISO pin.

Note: It's mandatory to keep consistent the sequence mode and data frame receipt and transmission of master and slave equipment to guarantee normal transmission of data.

15.3.3.2 Data transmission

Write data to the transmission data register SPI_TXREG, and the whole data is transmitted to the transmission buffer.

When the slave equipment receives the SCK clock signal as well as the first data bit of MOSI pin. The slave equipment uses SCK change edge and transmit the data to the MISO pin bit by bit. The transmission data process conforms to the relevant sequence of data/clock (determined by the CPOL, CPHA/CPHASEL bit).

However, during high speed transmission (configure SPI_CCTL.TXEDGE=1), the data won't change according to the input SCK clock edge, but instead transmit data to MISO pin in advance along the internal PCLK clock edge (it won't be earlier than the SCK clock edge of the previous bit of data receipt sampling).

When the first bit of data is transmitted, the hardware will assert SPI_INTSTAT.TX_INTF flag, and the software will use this flag to write TXREG for continuous data transmission (configure SPI_INTEN.TX_IEN bit as '1' to produce CPU interrupt).

Note: The slave unit clock signal is provided by the host. Therefore, the precondition of continuous transmission must be that the host can provide the continuous clock.

15.3.3.3 Data receipt

When the slave equipment receives a complete data from the MOSI pin:

- This data along the shift register will be transmitted to the receipt buffer in the final sampling clock edge. The hardware will simultaneously assert the SPI_INTSTAT.RX_INTF flag. And then the software will read the SPI_RXREG and acquire the data from the receipt buffer.
- The software configures SPI_INTEN.RX_IEN bit as '1' to open the interrupt enable, and use CPU interrupt to acquire the receipt data.

15.3.4 SPI master mode

When SPI serves as the master equipment, export the serial clock to the SCK pin for use by the slave equipment.

15.3.4.1 Configuration steps

1. Set SPI_SPBREG to define the serial clock baud rate.
2. Configure SPI_CCTL register CPOL, CPHA/CPHASEL, and determine the time sequence mode.
3. Configure SPI_CCTL.SPILEN to define 8 or 7 bit data frame format. Configure SPI_GCTL.DW8_32 as '1', and configure SPI_EXTCTL register to define any frame format (SPILEN should be fixed as '1').
4. Configure SPI_CCTL.LSBFE to determine the data receipt and transmission sequence (LSB or MSB bit preferred).
5. In case of only receipt of data rather than transmission, configure SPI_RXDNR register to define the received byte (upon receipt of the specified number of byte, SCK clock output will end and keep CPOL bit configuration status).
6. Configure MODE bit of the register SPI_GCTL as '1' (master mode), and SPIEN bit as '1' (SPI function enable), and configure GPIO function pin required for the SPI work.
7. Configure TXEN and RXEN bit of register SPI_GCTL as '1', and open transmission and receipt permission (during transmission, it's required to write data into the register SPI_TXREG after opening TXEN). SPI will export clock SCK and sync data MOSI to the pin in the master mode, and sample input data from MISO pin. NSS is the optional output function of master equipment.

Note: It's mandatory to keep consistent the sequence mode and data frame receipt and transmission of master and slave equipment to guarantee normal transmission of data.

15.3.4.2 Data transmission

Configure TXEN bit as '1', and write data to the transmission data register TXREG. The master data will be transmitted to the transmission buffer, and the master equipment begins transmission. The master equipment serially exports SCK clock and MOSI data to the pin according to the preset baud rate, and this process conforms to the relevant sequence of data/clock (as determined by CPOL, CPHA/CPHASEL). Besides, LSBFE bit determines the data serial transmission sequence.

When the 1st bit data is transmitted, the hardware will assert the SPI_INTSTAT.TX_INTF flag. The software uses this flag to write TXREG and achieve continuous data transmission (configuring SPI_INTEN.TX_IEN bit as '1' to produce CPU interrupt).

15.3.4.3 Data receipt

When the slave equipment receives a complete data from the MOSI pin:

- This data along the shift register will be transmitted to the receipt buffer in the final sampling clock edge. The hardware will simultaneously assert the SPI_INTSTAT.RX_INTF flag. And then the software will read the SPI_RXREG and acquire the data from the receipt buffer.

- The software configures SPI_INTEN.RX_IEN bit as '1' to open the interrupt enable, and use CPU interrupt to acquire the receipt data.
- In case of only receipt, and after receipt of the number of bytes as defined by RXDNR, the hardware will assert the SPI_INTSTAT.RXMATCH_INTF flag. Meanwhile, the master equipment won't transmit the clock signal, and SCK output will be kept in the CPOL bit configuration status (fixed high or low level).

15.3.5 Baud rate setting

SCK pin output clock frequency conforms to the baud rate configuration, which is obtained by the division of internal clock PCK according to the configuration value of SPI_SPBRG register. The register SPBREG controls the count period of one 16-bit counter.

According to the expected baud rate and Fpclk (APB module PCLK clock frequency), and use the formula of the table below to calculate the configuration value to the register SPBRG (X in the table below), X within the range of 2~65535.

Table 15-1 Baud rate formula

Mode	Formula
SPI mode	Baud rate = Fpclk/X

15.3.6 Interrupt

15.3.6.1 Status flag

To facilitate software operation, the application may use 4 current status flags and 7 interrupt flags to monitor SPI bus status.

The current status flag is only readable, and auto set and cleared by the hardware.

The interrupt status flag bit is set during event occurrence, and produces CPU interrupt during interrupt enable, cleared by software.

SPI has an 8 byte transmission buffer and receipt buffer. According to SPI_GCTL register DW8_32 bit setting, CPU may read and write 1 or 4 bytes each time. According to DW8_32 setting, the transmission and receipt buffer respectively have one byte or one valid data status flag.

Note: Configure SPI_GCTL.DW8_32=1, and receipt and transmission buffer have two valid data at most. When the frame length is configured as 8bit and below, one valid data is 1 byte. Its 2 bytes when the frame length is configured as a valid data within the range of 9~16bit. Its 3 bytes when the frame length is configured as a valid data within the range of 17~24bit. Its 4 bytes when the frame length is configured as a valid data within the range of 25~32bit.

Table 15-2 SPI status

Class	Status flag	Buffer and signal status
Interrupt status	TX_INTF	Transmission buffer is empty, according to DW8_32 setting to complete the write operation of transmission data register TXREG
	RX_INTF	According to DW8_32 setting, there is at least one valid data space, and complete the read operation of one receipt data register RXREG
	UNDERRUN_INTF	Transmission buffer is empty and repeat transmission
	RXOERR_INTF	Receipt buffer is not empty and covered
	RXMATCH_INTF	Non empty, the final one data is transmitted to the receipt buffer (valid in the master mode)
	RXFULL_INTF	Receipt buffer is full, and cannot receive new data
	TXEPT_INTF	Transmission buffer and shift register both are empty
Current status	RXAVL_4BYTE	Receipt buffer has the valid data over 4 bytes
	TXFULL	Transmission buffer is full
	TXEPT	Transmission buffer and shift register both are empty
	RXAVL	Receipt buffer is not empty

15.4 Register description

Table 15-3 SPI register overview

Offset	Acronym	Register Name	Reset
0x00	SPI_TXREG	Transmission Data Register	0x00000000
0x04	SPI_RXREG	Receipt Data Register	0x00000000
0x08	SPI_CSTAT	Current Status Register	0x00000001
0x0C	SPI_INTSTAT	Interrupt Status Register	0x00000000
0x10	SPI_INTEN	Interrupt Enable Register	0x00000000
0x14	SPI_INTCLR	Interrupt Clear Register	0x00000000
0x18	SPI_GCTL	Global Control Register	0x00000004
0x1C	SPI_CCTL	Universal Control Register	0x00000008
0x20	SPI_SPBRG	Baud Rate Generator	0x00000002
0x24	SPI_RXDNR	Receipt Data Number Register	0x00000001
0x28	SPI_NSSR	Slave Chip Selection Register	0x000000FF
0x2C	SPI_EXTCTL	Data Length Control Register	0x00000008

15.4.1 SPI_TXREG Transmission Data Register

Address offset: 0x00

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TXREG															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TXREG															
rw															

Bit	Field	Description
31 : 0	TXREG	Transmission data register Valid data bit is controlled by DW8_32: DW8_32=0, only valid in low 8 bit DW8_32=1, TXREG[31 : 0] valid

15.4.2 SPI_RXREG Receipt Data Register

Address offset: 0x04

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RXREG															
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RXREG															
r															

Bit	Field	Description
31 : 0	RXREG	Receipt data register Valid data bit is controlled by DW8_32. DW8_32=0: Only valid in low 8 bit DW8_32=1: RXREG[31:0] valid Note: The register is readable, but not writable.

15.4.3 SPI_CSTAT Current Status Register

Address offset: 0x08

Reset value: 0x0000 2001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.			BUSY	RXFADDR				TXFADDR				RXAVL_4BYTE	TXFULL	RXAVL	TXEPT
			r	r				r				r	r	r	r

Bit	Field	Description
31 : 13	Reserved	Always read as 0
12	BUSY	Busy flag bit Indicate SPI transmission is ongoing
11 : 8	RXFADDR	Number of valid byte in the current buffer
7 : 4	TXFADDR	Number of valid byte in the transmission buffer
3	RXAVL_4BYTE	Valid data in the receipt buffer reaches 4 byte flag bit 0: Data in receipt buffer is less than 4 byte 1: Data in receipt buffer exceeds 4 byte Note: When DW8_32 is configured to 1, this flag is set to 1 if the data amount meets or exceeds one frame; otherwise, it is set to 1 only if the amount meets or exceeds 4 bytes.
2	TXFULL	Transmission buffer full flag bit 0: Transmission buffer is not full 1: Transmission buffer is full Note: When DW8_32 is configured to 1, the hardware sets the flag after the FIFO stores 2 frames of data; otherwise, the hardware sets the flag only after the FIFO stores 8 bytes of data.
1	RXAVL	Receipt valid byte data message bit When the receipt buffer receives a complete byte data, the bit is set. 0: Receipt buffer is empty 1: Receipt buffer is not empty Note: This bit is only readable, auto set and cleared by the hardware.
0	TXEPT	Transmission empty bit 0: Transmission buffer or Shift register is not empty 1: Transmission buffer and Shift register both are empty Note: This bit is only readable, auto set and cleared by the hardware.

15.4.4 SPI_INTSTAT Interrupt Status Register

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.									TXEPT_I NTF	RXFULL _INTF	RXMAT CH_INT F	RXOER R_INTF	UNDER RUN_IN TF	RX_INT F	TX_INTF
									r	r	r	r	r	r	r
Bit	Field				Description										
31 : 7	Reserved				Reserved, always read as 0										

Bit	Field	Description
6	TXEPT_INTF	Transmitter empty interrupt flag bit Auto set by the hardware, clear by writing the INTCLR. TXEPT_ICLR bit as '1'. 0: Transmission buffer or Shift register is not empty 1: Transmission buffer and Shift register both are empty Note: This bit is the interrupt status signal while TXEPT is the status signal.
5	RXFULL_INTF	RX FIFO full interrupt flag bit Auto set by the hardware, clear by writing INTCLR. RXFULL_ICLR bit as '1'. 0: RX buffer is not full 1: RX buffer is full Note: This flag is valid only when DW8_32 is configured to 0 and EXTLEN remains at its default value of 8; a "full" status indicates that the valid data retained in the FIFO has reached 4 bytes.
4	RXMATCH_INTF	Receive specified byte interrupt flag bit (Receive data match the RXDNR number, the receive process will be completed and generate the interrupt) Auto set by the hardware, clear by writing INTCLR.RXMATCH_ICLR bit as '1'. 0: Still has not received specified byte in RXDNR register 1: Receive specified byte in RXDNR register
3	RXOERR_INTF	Receiver overflow error interrupt flag bit (Receive overrun error interrupt flag bit) Auto set by the hardware, clear by writing INTCLR. RXOERR_ICLR bit as '1'. 0: No overflow error 1: Overflow error
2	UNDERRUN_INTF	SPI slave mode underflow flag bit (SPI underrun interrupt flag bit) Auto set by the hardware, clear by writing INTCLR.UNDERRUN_ICLR bit as '1'. 0: No underrun error 1: Underrun error
1	RX_INTF	Receiver data valid interrupt flag bit Auto set by the hardware, clear by writing to the RX_ICLR bit of the INTCLR register. When the receive buffer receives complete and valid data. 0: Receipt buffer is empty 1: Receipt buffer has received a complete valid data
0	TX_INTF	Transmit FIFO available interrupt flag bit (One byte of data was sent) Auto set by the hardware, auto clear when the transmission buffer is not empty. 0: Transmission buffer is not empty 1: Transmission buffer is empty

15.4.5 SPI_INTEN Interrupt Enable Register

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
Res.									TXEPT_IEN	RXFULL_IEN	RXMATCH_IEN	RXOERR_IEN	UNDERRUN_IEN	RX_IEN	TX_IEN
									rw	rw	rw	rw	rw	rw	rw

Bit	Field	Description
31 : 7	Reserved	Reserved, always read as 0
6	TXEPT_IEN	Transmit empty interrupt enable bit 0 : Interrupt disable 1 : Interrupt enable
5	RXFULL_IEN	Receive FIFO full interrupt enable bit 0 : Interrupt disable 1 : Interrupt enable
4	RXMATCH_IEN	Receive data complete interrupt enable bit 0 : Interrupt disable 1 : Interrupt enable

Bit	Field	Description
3	R XOERR_IEN	Receipt overflow error interrupt enable bit 0 : Interrupt disable 1 : Interrupt enable
2	UNDERRUN_IEN	SPI slave mode underflow interrupt enable bit 0 : Interrupt disable 1 : Interrupt enable
1	RX_IEN	Receive FIFO interrupt enable bit 0 : Interrupt disable 1 : Interrupt enable
0	TX_IEN	Transmit FIFO empty interrupt enable bit 0 : Interrupt disable 1 : Interrupt enable

15.4.6 SPI_INTCLR Interrupt Clear Register

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.									TXEPT_I CLR	RXFULL_I CLR	RXMATCH_I CLR	R XOERR_I CLR	UNDERRUN_I CLR	RX_I CLR	Res.
									W	W	W	W	W	W	

Bit	Field	Description
31 : 7	Reserved	Reserved, always read as 0
6	TXEPT_I CLR	Transmitter empty interrupt clear bit 0 : Write 0 meaningless 1 : Write 1 to clear interrupt
5	RXFULL_I CLR	Receiver buffer full interrupt clear bit 0 : Write 0 meaningless 1 : Write 1 to clear interrupt
4	RXMATCH_I CLR	Receive completed interrupt clear bit 0 : Write 0 meaningless 1 : Write 1 to clear interrupt
3	R XOERR_I CLR	Receive overflow error interrupt clear bit 0 : Write 0 meaningless 1 : Write 1 to clear interrupt
2	UNDERRUN_I CLR	SPI slave mode overflow interrupt clear bit 0 : Write 0 meaningless 1 : Write 1 to clear interrupt
1	RX_I CLR	Receive interrupt clear bit 0 : Write 0 meaningless 1 : Write 1 to clear interrupt
0	Reserved	Reserved, always read as 0

15.4.7 SPI_GCTL Global Control Register

Address offset: 0x18

Reset value: 0x0000 0004

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved												FCR_IOR_M	NSS_S	PAD_SEL	
												rw	rw	rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PAD_SEL			Res.	DW8_32	NSS_M	Reserved						RXEN	TXEN	MODE	SPIEN
rw				rw	rw							rw	rw	rw	rw

Bit	Field	Description																																																																																																																													
31 : 20	Reserved	Reserved, always read as 0																																																																																																																													
19	FCRIORM	Full Continue receiving in only read mode 0: In read-only mode, when RXFIFO is full, stop transmitting the clock signal and the active chip select signal. 1: In read-only mode, when RXFIFO is full, continues to output the clock and the active chip select signal; Note: When the RXFIFO is full, previously received data remains in the FIFO.																																																																																																																													
18	NSS_S	NSS select signal that from software or external PAD 0: Controlled by external NSS PAD 1: Controlled by NSSR register value.																																																																																																																													
17: 13	PAD_SEL	PAD0 , PAD1 , PAD2 , PAD3 respectively correspond to SCK, MOSI, NSS and MISO pins of pins. Mapping change of signal SCL, MOSI, NSS, MISO and PAD: <table><tr><td>PAD_SEL</td><td>SCL</td><td>MOSI</td><td>NSS</td><td>MISO</td></tr><tr><td>0</td><td>PAD0</td><td>PAD1</td><td>PAD2</td><td>PAD3</td></tr><tr><td>1</td><td>PAD0</td><td>PAD1</td><td>PAD3</td><td>PAD2</td></tr><tr><td>2</td><td>PAD0</td><td>PAD2</td><td>PAD1</td><td>PAD3</td></tr><tr><td>3</td><td>PAD0</td><td>PAD2</td><td>PAD3</td><td>PAD1</td></tr><tr><td>4</td><td>PAD0</td><td>PAD3</td><td>PAD1</td><td>PAD2</td></tr><tr><td>5</td><td>PAD0</td><td>PAD3</td><td>PAD2</td><td>PAD1</td></tr><tr><td>6</td><td>PAD1</td><td>PAD0</td><td>PAD2</td><td>PAD3</td></tr><tr><td>7</td><td>PAD1</td><td>PAD0</td><td>PAD3</td><td>PAD2</td></tr><tr><td>8</td><td>PAD1</td><td>PAD2</td><td>PAD0</td><td>PAD3</td></tr><tr><td>9</td><td>PAD1</td><td>PAD2</td><td>PAD3</td><td>PAD0</td></tr><tr><td>10</td><td>PAD1</td><td>PAD3</td><td>PAD0</td><td>PAD2</td></tr><tr><td>11</td><td>PAD1</td><td>PAD3</td><td>PAD2</td><td>PAD0</td></tr><tr><td>12</td><td>PAD2</td><td>PAD0</td><td>PAD1</td><td>PAD3</td></tr><tr><td>13</td><td>PAD2</td><td>PAD0</td><td>PAD3</td><td>PAD1</td></tr><tr><td>14</td><td>PAD2</td><td>PAD1</td><td>PAD0</td><td>PAD3</td></tr><tr><td>15</td><td>PAD2</td><td>PAD1</td><td>PAD3</td><td>PAD0</td></tr><tr><td>16</td><td>PAD2</td><td>PAD3</td><td>PAD0</td><td>PAD1</td></tr><tr><td>17</td><td>PAD2</td><td>PAD3</td><td>PAD1</td><td>PAD0</td></tr><tr><td>18</td><td>PAD3</td><td>PAD0</td><td>PAD1</td><td>PAD2</td></tr><tr><td>19</td><td>PAD3</td><td>PAD0</td><td>PAD2</td><td>PAD1</td></tr><tr><td>20</td><td>PAD3</td><td>PAD1</td><td>PAD0</td><td>PAD2</td></tr><tr><td>21</td><td>PAD3</td><td>PAD1</td><td>PAD2</td><td>PAD0</td></tr><tr><td>22</td><td>PAD3</td><td>PAD2</td><td>PAD0</td><td>PAD1</td></tr><tr><td>23</td><td>PAD3</td><td>PAD2</td><td>PAD1</td><td>PAD0</td></tr></table>	PAD_SEL	SCL	MOSI	NSS	MISO	0	PAD0	PAD1	PAD2	PAD3	1	PAD0	PAD1	PAD3	PAD2	2	PAD0	PAD2	PAD1	PAD3	3	PAD0	PAD2	PAD3	PAD1	4	PAD0	PAD3	PAD1	PAD2	5	PAD0	PAD3	PAD2	PAD1	6	PAD1	PAD0	PAD2	PAD3	7	PAD1	PAD0	PAD3	PAD2	8	PAD1	PAD2	PAD0	PAD3	9	PAD1	PAD2	PAD3	PAD0	10	PAD1	PAD3	PAD0	PAD2	11	PAD1	PAD3	PAD2	PAD0	12	PAD2	PAD0	PAD1	PAD3	13	PAD2	PAD0	PAD3	PAD1	14	PAD2	PAD1	PAD0	PAD3	15	PAD2	PAD1	PAD3	PAD0	16	PAD2	PAD3	PAD0	PAD1	17	PAD2	PAD3	PAD1	PAD0	18	PAD3	PAD0	PAD1	PAD2	19	PAD3	PAD0	PAD2	PAD1	20	PAD3	PAD1	PAD0	PAD2	21	PAD3	PAD1	PAD2	PAD0	22	PAD3	PAD2	PAD0	PAD1	23	PAD3	PAD2	PAD1	PAD0
PAD_SEL	SCL	MOSI	NSS	MISO																																																																																																																											
0	PAD0	PAD1	PAD2	PAD3																																																																																																																											
1	PAD0	PAD1	PAD3	PAD2																																																																																																																											
2	PAD0	PAD2	PAD1	PAD3																																																																																																																											
3	PAD0	PAD2	PAD3	PAD1																																																																																																																											
4	PAD0	PAD3	PAD1	PAD2																																																																																																																											
5	PAD0	PAD3	PAD2	PAD1																																																																																																																											
6	PAD1	PAD0	PAD2	PAD3																																																																																																																											
7	PAD1	PAD0	PAD3	PAD2																																																																																																																											
8	PAD1	PAD2	PAD0	PAD3																																																																																																																											
9	PAD1	PAD2	PAD3	PAD0																																																																																																																											
10	PAD1	PAD3	PAD0	PAD2																																																																																																																											
11	PAD1	PAD3	PAD2	PAD0																																																																																																																											
12	PAD2	PAD0	PAD1	PAD3																																																																																																																											
13	PAD2	PAD0	PAD3	PAD1																																																																																																																											
14	PAD2	PAD1	PAD0	PAD3																																																																																																																											
15	PAD2	PAD1	PAD3	PAD0																																																																																																																											
16	PAD2	PAD3	PAD0	PAD1																																																																																																																											
17	PAD2	PAD3	PAD1	PAD0																																																																																																																											
18	PAD3	PAD0	PAD1	PAD2																																																																																																																											
19	PAD3	PAD0	PAD2	PAD1																																																																																																																											
20	PAD3	PAD1	PAD0	PAD2																																																																																																																											
21	PAD3	PAD1	PAD2	PAD0																																																																																																																											
22	PAD3	PAD2	PAD0	PAD1																																																																																																																											
23	PAD3	PAD2	PAD1	PAD0																																																																																																																											
12	Reserved	Reserved, must retain the reset value.																																																																																																																													
11	DW8_32	Transmission and receipt data register valid data selection (Valid byte or double- word data select signal) 0: Only valid in low 8 bit 1: All 32 bits of data are valid. Note : Access must be made using the specified data format.																																																																																																																													
10	NSS_M	NSS output in the hardware or software control master mode 0:Control by the NSSR register value 1:Auto control by the hardware during data transmission																																																																																																																													
9 : 5	Reserved	Reserved, must retain the reset value.																																																																																																																													
4	RXEN	Receipt enable bit 0: Receipt disable. Clear RX buffer at the same time 1: Receipt enable Note: When the SPI only works in the host receipt mode, TXEN must be set as 0.																																																																																																																													
3	TXEN	Transmission enable bit 0: Transmission disable. Clear TX buffer at the same time 1: Transmission enable Note: In the host mode, the transmission and receipt occur simultaneously																																																																																																																													
2	MODE	Host mode bit 0: Slave mode (serial clock from external host) 1: Host mode (serial clock produced by the internal BRG)																																																																																																																													
1	INTEN	SPI interrupt enable bit 0: Disable SPI interrupt 1: Enable SPI interrupt																																																																																																																													

Bit	Field	Description
0	SPIEN	SPI selection bit 0 : SPI disable (reset status) 1 : SPI enable

15.4.8 SPI_CCTL Universal Control Register

Address offset: 0x1C

Reset value: 0x0000 0008

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.					MRDECHG			Res.	CPHASE L	TXEDGE	RXEDG E	SPILEN	LSBFE	CPOL	CPHA
					rw				rw	rw	rw	rw	rw	rw	rw

Bit	Field	Description
31 : 11	Reserved	Reserved, always read as 0
10 : 8	MRDECHG	Receives the sampling clock edge adjustment function bit in master mode. 3'b000: If RXEDGE=0, the sampling point is not adjusted; If RXEDGE=1, the sampling point is shifted by SPBRG/2 mclk cycles. 3'b001: Shift the sampling point later by one MCLK cycle. 3'b010: Shift the sampling point later by 2 MCLK cycles. 3'b011: Shift the sampling point later by 3 MCLK cycles. 3'b1xx: Shift the sampling point later by 4 MCLK cycles. Note: mclk is the APB clock used by the SPI module, and SPBRG represents the setting value of the baud rate control register; when the MRDECHG setting is not 3'b000, the RXEDGE setting is disabled. In high-speed communication applications, the system can determine this configuration by adjusting for board-level routing delays; this allows for the compensation of transmission delays during SPI master-mode reception, thereby increasing transmission speed.
7	Reserved	Reserved, always read as 0
6	CPHASEL	CPHA polarity inversion selection 0: CPHA setting remains unchanged. 1: Invert CPHA setting value. In case of CPHA as 1, the first data bit sampling begins from the second clock edge. In case of CPHA as 0, the first data bit sampling begins from the first clock edge.
5	TXEDGE	Transmit data phase adjustment bit (slave mode) 0: Transmit data to the bus after one valid clock edge. To be used in the low speed mode (SPBRG > 4). 1: Immediately transmit data to the data bus. To be used in the high speed mode (SPBRG = 4).
4	RXEDGE	Receive data sampling clock edge selection bit (master mode) 0: Sample data in the middle of transmission data bit. 1: Sample clock in the tail clock edge of the transmission data (for high speed mode).
3	SPILEN	SPI character length bit This bit works after DW8_32 setting (DW8_32 =0); 0: 7 bit data 1: 8 bit data (default)
2	LSBFE	LSBFE:LSB front enable bit 0: Data transmission or receipt highest bit in front 1: Data transmission or receipt lowest bit in front
1	CPOL	Clock polarity flag bit 0: Clock in low level in idle status (between two transmissions)

		1: Clock in high level in idle status (between two transmissions)
0	CPHA	Clock phase selection bit 0: The first data bit sampling begins from the second clock edge 1: The first data bit sampling begins from the first clock edge

15.4.9 SPI_SPBRG Baud Rate Generator

Address offset: 0x20

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
SPBRG															
rw															

Bit	Field	Description
31 : 16	Reserved	Reserved, always read as 0
15 : 0	SPBRG	SPI baud rate control register for baud rate Baud rate formula: Baud rate = Fpclk/SPBRG (Fpclk is APB clock frequency) Note: Don't write 0 and 1 in the register.

15.4.10 SPI_RXDNR Receipt Data Number Register

Address offset: 0x24

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
RXDNR															
rw															

Bit	Field	Description
31 : 16	Reserved	Reserved, always read as 0
15 : 0	RXDNR	The register is used to hold a count of to be received in the next receipt process Note: The register value is valid when SPI is in the host receipt mode. The default value is 1. The register value changes by MCU write value, and don't write the '0' value in the register.

15.4.11 SPI_NSSR Slave Chip Selection Register

Address offset: 0x28

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.															NSS

	rw
--	----

Bit	Field	Description
31 : 1	Reserved	Reserved, must retain the reset value.
0	NSS	The chip selection output signal in the master mode. Low valid, the bit is invalid in the slave mode 0: Slave device is not selected (allow communication between the slave device and master mode) 1: Slave device is selected

15.4.12 SPI_EXTCTL Data Length Control Register

Address offset: 0x2C

Reset value: 0x0000 0008

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.											EXTLEN				
											rw				

Bit	Field	Description
31 : 5	Reserved	Reserved, always read as 0
4 : 0	EXTLEN	Control SPI data length 0 0000 : 32 bit 0 0001 : 1 bit 0 0010 : 2 bit 0 0011 : 3 bit 1 1100 : 28 bit 1 1101 : 29 bit 1 1110 : 30 bit 1 1111 : 31 bit Note: 1. During operation, DW8_32 must be configured to '1', the LSBFE bit of the SPI_CCTL register must be configured to '1', and the SPIEN bit must also be configured to '1'. When DW8_32 is configured to '0', EXTLEN must remain at its initial value of 8. 2. EXTLEN[4:0] should be configured when the SPI_GCTL.SPIEN bit is '0' and must not be modified during communication.

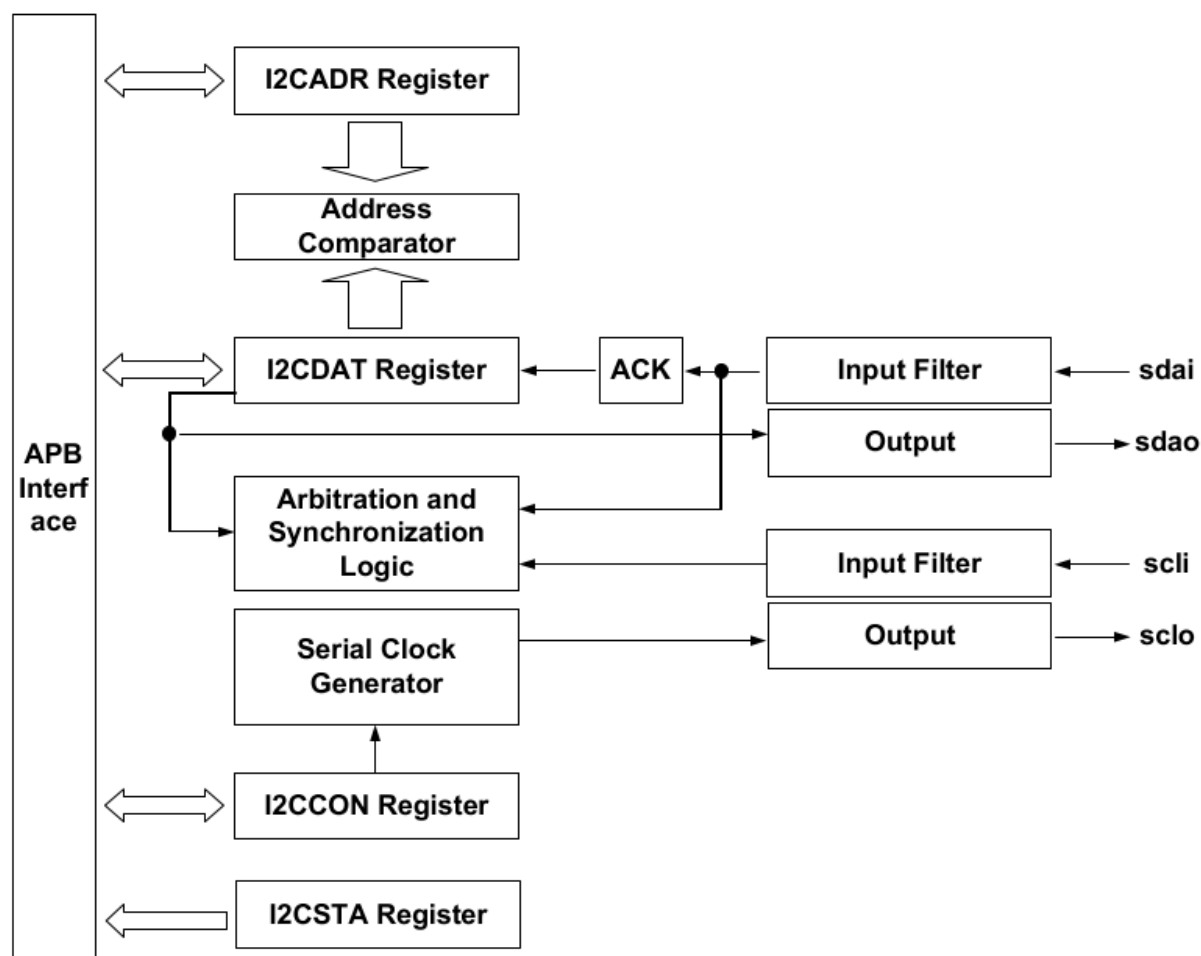
16 I2C Inter-Integrated Circuit InterfaceIntroduction

I2C (inter-integrated circuit) bus interface provides support for serial interconnection between chips in the microcontroller. It provides multi-master capability to control all I2C bus-specific sequencing, protocol, arbitration and timing.

The I2C bus is a two-wire serial interface, where the serial data line (SDA) and the serial clock line (SCL) pass information between the devices connected to the bus. Each device is addressable by a unique address and can operate as a transmitter or receiver. Besides, the device can also be considered as a master or slave when it conducts data transfer. A master is a device that initiates the data transfer on the bus and generates clock signals that allow the transfer. At this point, any device addressed is considered as a slave.

16.2Main characteristics

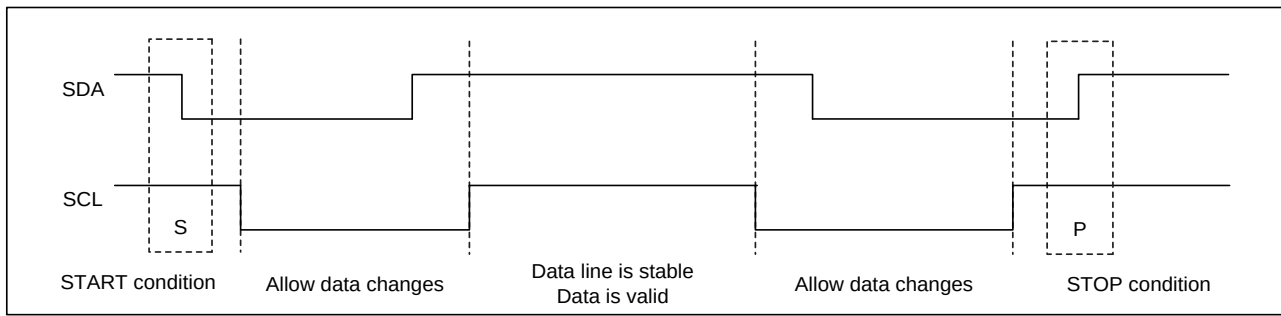
- I2C bus protocol converter/parallel-bus
- Half-duplex synchronous operation
- Master and Slave modes supported
- 7-bit/10-bit address format supported
- START, STOP, RESTART, and ACK generation and detection supported
- Standard mode (up to 100Kbps) and Fast mode (up to 400Kbps) supported
- Interrupt or polled-mode operation supported



--	--	--

Pin Name	Attribute	Description
I2C_SCL	I/O	I2C clock
I2C_SDA	I/O	I2C data

When the bus is idle, SCL and SDA are pulled up by external pull-up resistors. The data transfer initiated in the Master begins with a START condition. A high to low transition on the SDA line while SCL is high defines a START condition. The data transfer in the Master is terminated by the Master issuing a STOP condition. A low to high transition on the SDA line while SCL is high defines a STOP condition. A timing diagram of Start and STOP conditions is shown as below. SDA must maintain steady during the data transfer when SCL is 1.



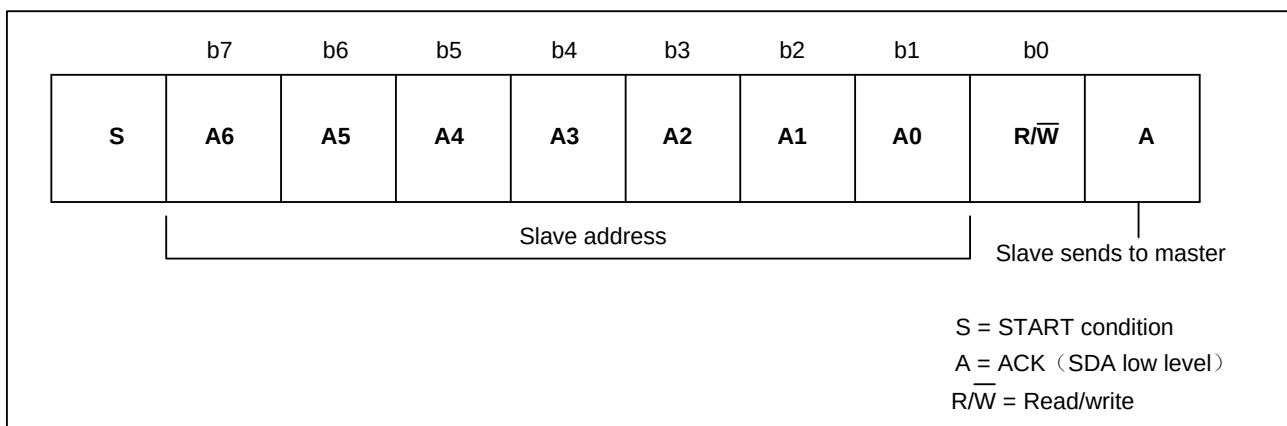
16.3.3.2 Addressing protocol

I2C supports the 7-bit addressing format.

- 7-bit address format

The slave address is the first 7 bits (bit 7:1) of the first byte sent following a START condition (S). The least significant bit (bit 0) determines the data transfer direction. If bit 0 is 0, then the Master writes data to the Slave; if bit 0 is 1, then the Master reads data from the Slave.

Figure 16-3 7-bit address format



The following table defines the special purpose and reserved addresses of the first byte for I2C:

Table 16-2 First byte for I2C

Slave Address	R/W Bit	Description
0000 000	0	General Call address. I2C places the data in the receive buffer and issues a General Call interrupt.
0000 001	1	Reserved
0000 001	x	Reserved
0000 010	x	Reserved
0000 011	x	Reserved
0000 1xx	x	Reserved
1111 1xx	x	Reserved
1111 0xx	x	Reserved

16.3.3.3 Transmitting and receiving protocol

The master can initiate data transmission and reception to or from the bus, acting as either a master-transmitter or master-receiver. A slave responds to requests from the master, acting as either a slave-transmitter or slave-receiver.

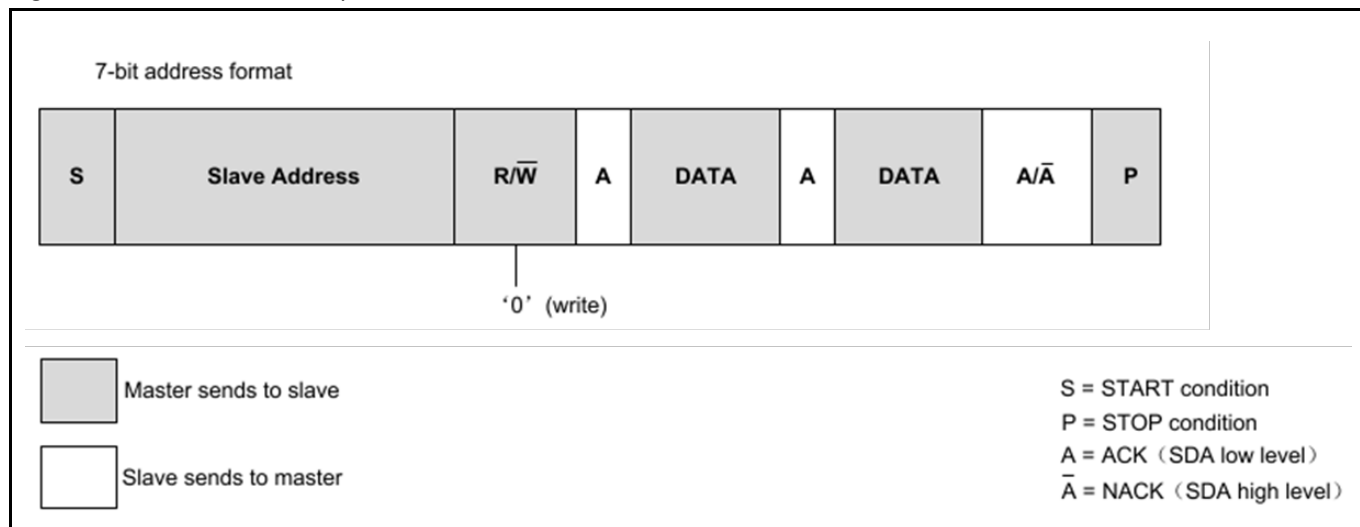
- Master-transmitter and Slave-receiver

All data is transmitted in byte format, with no limit on the number of bytes per data transfer. After the master sends the address and R/W bit or the master transmits a byte of data to the slave, the slave-receiver must respond with the acknowledge signal (ACK). When a slave-receiver does not respond with an ACK pulse, the master aborts

the transfer by issuing a STOP condition. If a slave is not able to respond, the slave must leave the SDA line high so that the master can abort the transfer.

If the master-transmitter is transmitting data as shown in the following figure, then the slave-receiver responds to the master-transmitter with an ACK pulse after every byte of data is received.

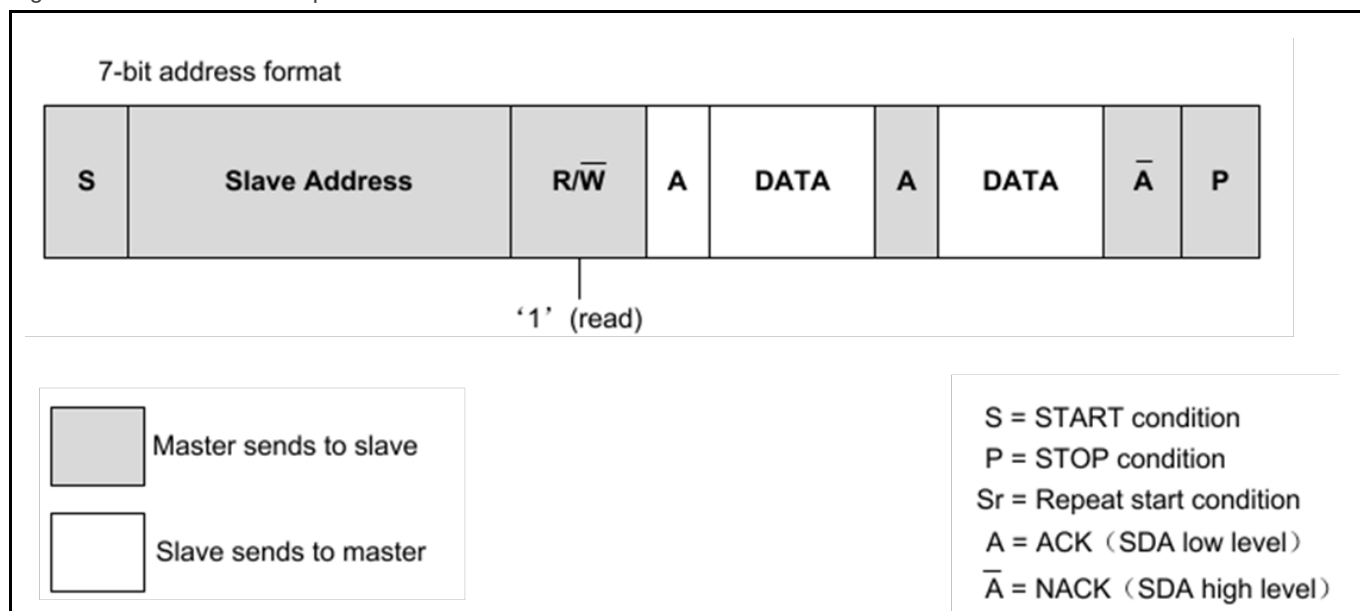
Figure 16-4 Master-transmitter protocol



● Master-receiver and Slave-transmitter

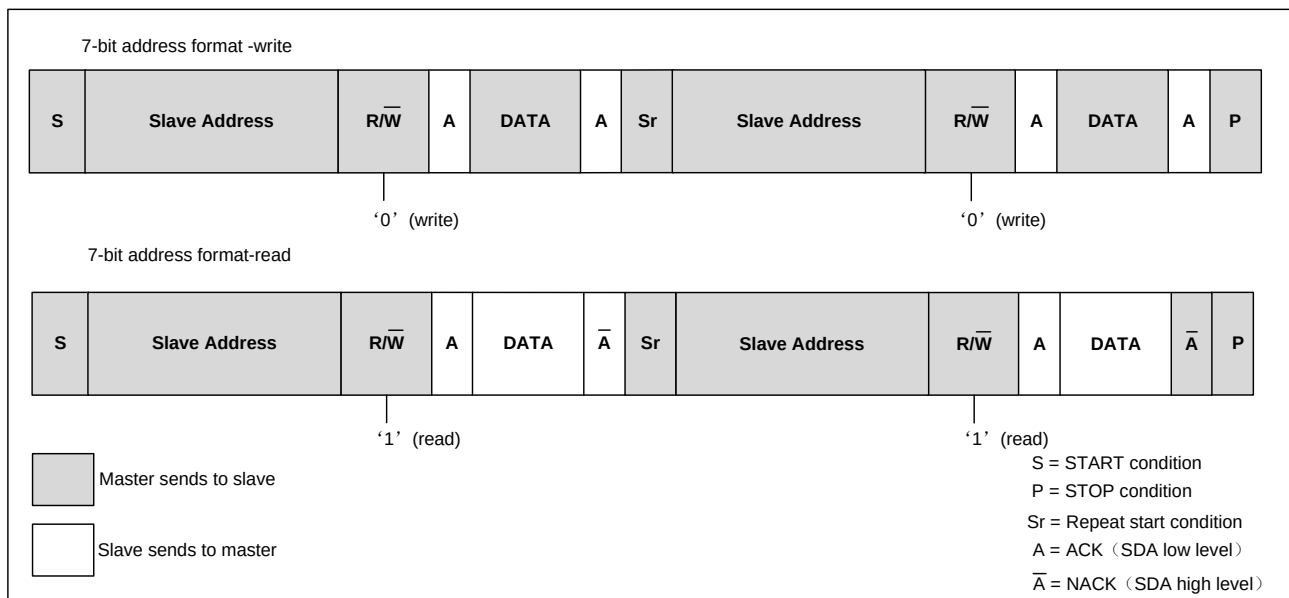
If the master is receiving data as shown in the following figure, then the master responds to the slave-transmitter with an ACK pulse after a byte of data has been received, except for the last byte. This is the way the master-receiver notifies the slave-transmitter that this is the last byte. The slave-transmitter relinquishes the SDA line after detecting the No Acknowledge (NACK) so that the master can issue a STOP condition.

Figure 16-5 Master-receiver protocol



When a master does not want to relinquish the bus with a STOP condition, the master can issue a RESTART condition. This is identical to a START condition except it occurs after the ACK. Operating in Master mode, the I2C interface can then communicate with the same slave using a transfer of a different direction.

Figure 16-6 Master-transmitter and receiver protocol with a RESTART (SR)



16.3.3.4 Arbitration

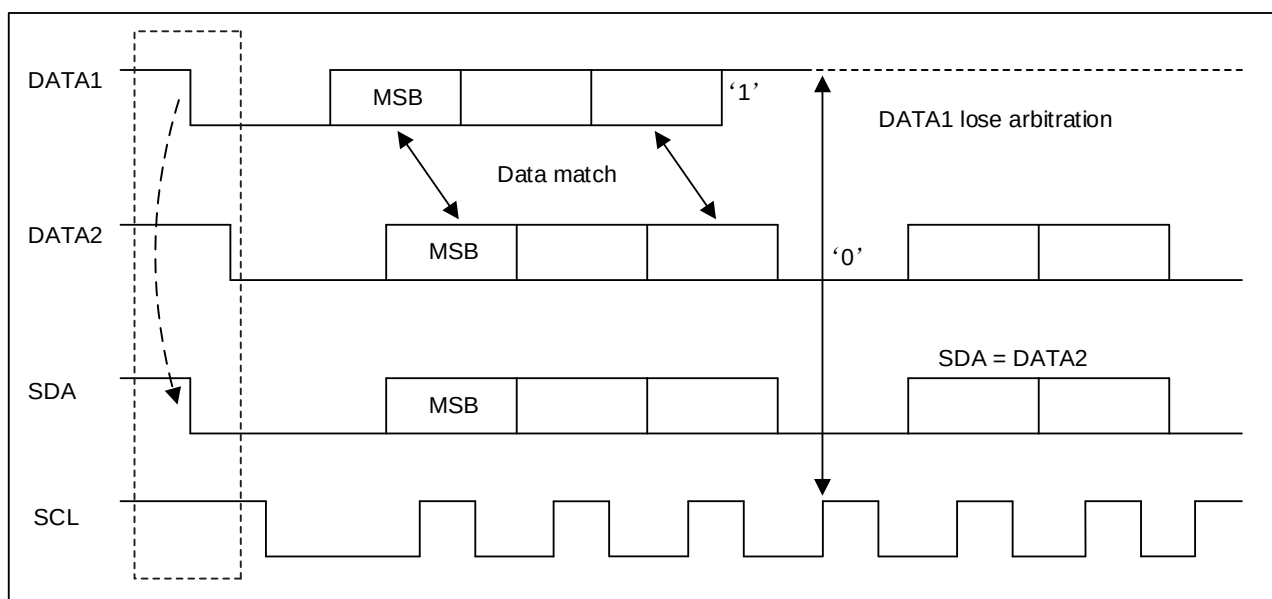
If there are several masters on the same bus, there is an arbitration procedure if they try to take control of the bus by generating the START condition at the same time (if several masters try to take control of the bus at the same time, only one master can take control of the bus with intact messages). Once a master has control of the bus, no other master can take control until the first master sends a STOP condition and places the bus in an idle state.

Arbitration takes place on the SDA line, while the SCL line is high. If there are two or more masters trying to send messages to the bus, the master, which transmits a 1 while the other master transmits 0, loses arbitration. The master that lost arbitration can continue to generate clock pulses until the end of the byte transfer. If all masters are addressing the same device, the arbitration could go into the data phase.

Upon detecting that it has lost arbitration, the I2C interface stops generating SCL.

The following figure illustrates the timing of two masters arbitrating on the bus.

Figure 16-7 Dual master arbitration



16.3.3.5 Clock synchronization

When two or more masters try to transfer information on the bus at the same time, they must arbitrate and synchronize the SCL clock. All masters generate their own clock to transfer messages. Data is only valid during the high period of clock. Clock synchronization is performed using the wired-AND connection to the SCL signal. When the master transitions the SCL clock to 0, the master starts counting the low time of the SCL and transitions the SCL to 1 at the beginning of the next clock period. However, if another master is holding the SCL line to 0, then the master goes into a wait state until the SCL transitions to 1.

All masters then count off their high time, and the master with the shortest high time transitions the SCL to 0. The masters then count out their low time and the one with the longest low time forces the other masters into a wait state. Therefore, a synchronized SCL clock is generated, which is illustrated in the following figure.

Figure 16-8 Clock synchronization (schematic diagram)

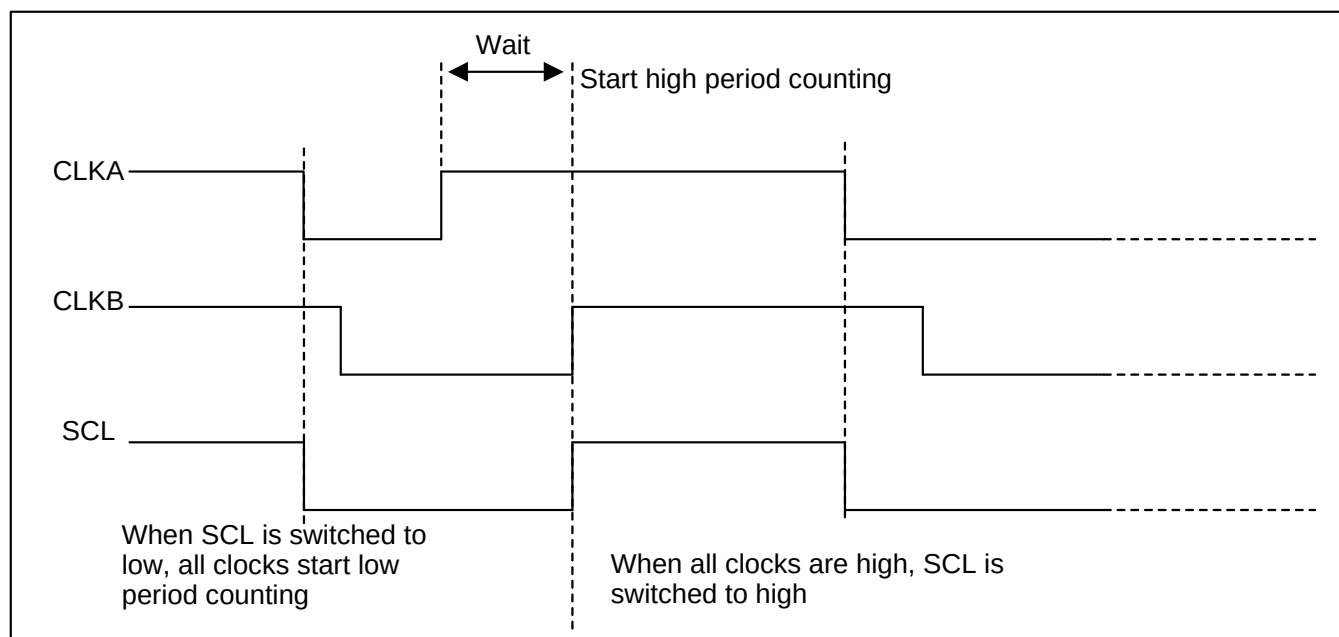
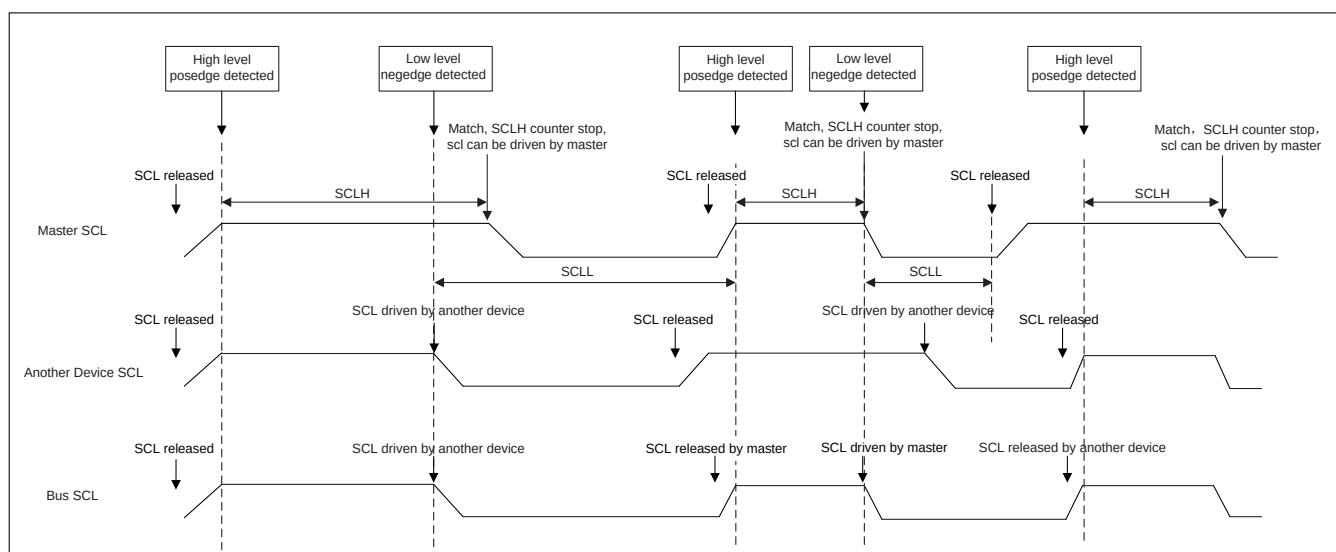


Figure 16-9 Clock synchronization (timing diagram)



16.3.3.6 SCL configuration

In master mode, the SCL output frequency can be configured via the I2C_DIV register; the formula for calculating the baud rate is as follows:

$$\text{SCL baud} = \text{pclk} / (4 * (\text{DIV} + 1))$$

(DIV>=16'h3)

16.3.4 Operating mode

I2C interface can operate in one of the four following modes:

- Slave transmitter mode
- Slave receiver mode
- Master transmitter mode
- Master receiver mode

The I2C interface can be enabled by configuring I2C_CON.ENS1; for details regarding mode switching and operation, please refer to Section 16.3.5.

16.3.4.1 Spike filtering

I2C supports a fixed spike filter. Before internal sampling, it checks whether the input signals (SCL and SDA) remain stable for three APB clock cycles.

16.3.4.2 Address matching

I2C compares the received 7-bit slave address with the address in the I2C_ADR register; if they match, a corresponding interrupt is generated. Additionally, if the I2C_ADR.GC bit is configured to 1, the received address is also compared against the broadcast address 0x0.

16.3.5 Interrupt

When any of the 25 interrupt states described in the table below occur (excluding 0xF8), I2C_CON.SI is automatically set by hardware. I2C_CON.SI must be cleared by manually writing a "0" via software; writing a "1" has no effect. Within the interrupt service routine, other register operations must be performed first; finally, the I2C_CON.SI flag bit is manually cleared by writing a '0' to it. Access to the I2C_DAT register must be performed within the interrupt service routine. After setting I2C_CON.ENS1=1 and commencing I2C data transfer, manipulations of the AA, SI, STA, and STO bits in the I2C_CON register must be performed within the interrupt service routine.

The table below lists the interrupt status, corresponding software responses, and subsequent hardware actions for the different operating modes of I2C.

16.3.5.1 Master transmitter mode

The default idle status code for I2C is 0xF8. When in state 0xF8, setting I2C_CON.STA=1 causes the I2C interface to transmit a START condition and enter master mode. Subsequently, loading the slave address and write command while in state 0x08 switches the interface to master-transmitter mode. Finally, setting I2C_CON.STO=1 at the appropriate time causes I2C to transmit a STOP condition and return to the idle state of 0xF8.

Table 16-3 I2C Master Transmit Mode Interrupt Status

Status code	I2C status	Software response					Next Action Taken by I2C Hardware
		I2C_DAT Operation	I2C_CON Operation				
			STA	STO	SI	AA	
0x08	A START condition has been transmitted	Load SLA+W	X	0	0	X	SLA+W will be transmitted; ACK or NACK will be received
0x10	A repeated START condition has been transmitted	Load SLA+W	X	0	0	X	SLA+W will be transmitted; ACK or NACK will be received

Status code	I2C status	Software response					Next Action Taken by I2C Hardware
		I2C_DAT Operation	I2C_CON Operation				
			STA	STO	SI	AA	
		Load SLA+R	X	0	0	X	SLA+R will be transmitted; ACK or NACK will be received I2C will switch to Master Receiver mode
0x18	SLA+W has been transmitted; ACK has been received	Load data byte	0	0	0	X	Data byte will be transmitted and ACK or NACK will be received
		No action	1	0	0	X	Repeated START will be transmitted
		No action	0	1	0	X	STOP condition will be transmitted and STO Flag will be reset
		No action	1	1	0	X	STOP condition followed by a START condition will be transmitted and STO Flag will be reset
0x20	SLA+W has been transmitted; NACK has been received	Load data byte	0	0	0	X	Data byte will be transmitted and ACK or NACK will be received
		No action	1	0	0	X	Repeated START will be transmitted
		No action	0	1	0	X	STOP condition will be transmitted and STO Flag will be reset
		No action	1	1	0	X	STOP condition followed by a START condition will be transmitted and STO Flag will be reset
0x28	Data byte has been transmitted; ACK has been received	Load data byte	0	0	0	X	Data byte will be transmitted and ACK or NACK will be received
		No action	1	0	0	X	Repeated START will be transmitted
		No action	0	1	0	X	STOP condition will be transmitted and STO Flag will be reset
		No action	1	1	0	X	STOP condition followed by a START condition will be transmitted and STO Flag will be reset
0x30	Data byte has been transmitted;	Load data byte	0	0	0	X	Data byte will be transmitted and

Status code	I2C status	Software response					Next Action Taken by I2C Hardware
		I2C_DAT Operation	I2C_CON Operation				
			STA	STO	SI	AA	
	NACK has been received						ACK or NACK will be received
		No action	1	0	0	X	Repeated START will be transmitted
		No action	0	1	0	X	STOP condition will be transmitted and STO Flag will be reset
		No action	1	1	0	X	STOP condition followed by a START condition will be transmitted and STO Flag will be reset
0x38	Arbitration lost in SLA+W or data bytes	No action	0	0	0	X	I2C will be released and not addressed Slave mode entered
		No action	1	0	0	X	A START condition will be transmitted when the bus becomes free

Note: SLA: Slave address.

R: Read bit (high level at SDA)

W: Write bit (low level at SDA)

16.3.5.2 Master receiver mode

The default idle status code for I2C is 0xF8. When in state 0xF8, setting I2C_CON.STA=1 causes the I2C interface to transmit a START condition and enter master mode. Subsequently, loading the slave address and read command while in state 0x08 switches the interface to master-receive mode. Finally, setting I2C_CON.STO=1 at the appropriate time causes I2C to transmit a STOP condition and return to the idle state of 0xF8.

Table 16-4 I2C Master Receiver Mode Interrupt Status

Status code	I2C status	Software response					Next Action Taken by I2C Hardware
		I2C_DAT Operation	I2C_CON Operation				
			STA	STO	SI	AA	
0x08	A START condition has been transmitted	Load SLA+R	X	0	0	X	SLA+R will be transmitted; ACK or NACK will be received
0x10	A repeated START condition has been transmitted	Load SLA+R	X	0	0	X	SLA+R will be transmitted; ACK or NACK will be received
		Load SLA+W	X	0	0	X	SLA+W will be transmitted; ACK or NACK will be received I2C will switch to Master Transmitter mode
0x38	Arbitration lost in SLA+R or NACK bit	No action	0	0	0	X	I2C will be released and not addressed Slave mode entered
		No action	1	0	0	X	A START condition will be transmitted when the bus becomes free
0x40	SLA+R has been transmitted; ACK has been received	No action	0	0	0	0	Data byte will be received and NACK will be returned
		No action	0	0	0	1	Data byte will be received and ACK will be returned
0x48	SLA+R has been transmitted; NACK has been received	No action	1	0	0	X	Repeated START will be transmitted
		No action	0	1	0	X	STOP condition will be transmitted and STO Flag will be reset
		No action	1	1	0	X	STOP condition followed by a START condition will be transmitted and STO Flag will be reset
0x50	Data byte has been received; ACK has been returned	Read data byte	0	0	0	0	Data byte will be received and NACK will be returned
		Read data byte	0	0	0	1	Data byte will be received and ACK will be returned

Status code	I2C status	Software response					Next Action Taken by I2C Hardware
		I2C_DAT Operation	I2C_CON Operation				
			STA	STO	SI	AA	
0x58	Data byte has been received; NACK has been returned	Read data byte	1	0	0	X	Repeated START will be transmitted
		Read data byte	0	1	0	X	STOP condition will be transmitted and STO Flag will be reset
		Read data byte	1	1	0	X	STOP condition followed by a START condition will be transmitted and STO Flag will be reset

16.3.5.3 Slave transmitter mode

The default idle state code for the I2C interface is 0xF8. When I2C_CON.AA is set to 1 while in this state, the I2C interface can receive its own slave address and respond with an ACK. If the received address corresponds to a read command, the interface enters slave transmitter mode; subsequently, upon detecting an I2C stop condition or completing the transmission of the final data byte, the interface returns to the idle state of 0xF8.

Table 16-5 I2C Slave Transmitter Mode Interrupt Status

Status code	I2C status	Software response					Next Action Taken by I2C Hardware
		I2C_DAT Operation	I2C_CON Operation				
			STA	STO	SI	AA	
0xA8	Own SLA+R has been received; ACK has been returned	Load data byte	X	0	0	0	Last data byte will be transmitted and NACK should be received
		Load data byte	X	0	0	1	Data byte will be transmitted and ACK should be received
0xB0	Arbitration lost in SLA+R/W as Master; own SLA+R has been received; ACK has been returned	Load data byte	X	0	0	0	Last data byte will be transmitted and NACK should be received
		Load data byte	X	0	0	1	Data byte will be transmitted and ACK should be received
0xB8	Data byte has been transmitted; ACK has been received	Load data byte	X	0	0	0	Last data byte will be transmitted and NACK should be received
		Load data byte	X	0	0	1	Data byte will be transmitted and ACK should be received
0xC0	Data byte has been transmitted; NACK has been received	No action	0	0	0	0	Switched to the not addressed Slave mode; no recognition of own SLA or General Call Address
		No action	0	0	0	1	Switched to the not addressed Slave mode; own SLA or General Call Address will be recognized

Status code	I2C status	Software response					Next Action Taken by I2C Hardware
		I2C_DAT Operation	I2C_CON Operation				
			STA	STO	SI	AA	
		No action	1	0	0	0	Switched to the not addressed Slave mode; no recognition of own SLA or General Call Address ; a START condition will be transmitted when the bus becomes free
		No action	1	0	0	1	Switched to the not addressed Slave mode; own SLA or General Call Address will be recognized; a START condition will be transmitted when the bus becomes free
0xC8	Last data byte has been transmitted; ACK has been received	No action	0	0	0	0	Switched to the not addressed Slave mode; no recognition of own SLA or General Call Address
		No action	0	0	0	1	Switched to the not addressed Slave mode; own SLA or General Call Address will be recognized
		No action	1	0	0	0	Switched to the not addressed Slave mode; no recognition of own SLA or General Call Address ; a START condition will be transmitted when the bus becomes free
		No action	1	0	0	1	Switched to the not addressed Slave mode; own SLA or General Call Address will be recognized; a START condition will be transmitted when the bus becomes free

16.3.5.4 Slave receiver mode

The default idle state code for the I2C interface is 0xF8. When I2C_CON.AA is set to 1 while in this state, the I2C interface is enabled to recognize its own slave address and respond with an ACK. Upon receiving its own slave address, if the received command is identified as a write operation, I2C enters slave receiver mode; subsequently, after receiving an I2C STOP condition, it returns to the idle state of 0xF8.

Table 16-6 I2C Slave Receiver Mode Interrupt Status

Status code	I2C status	Software response					Next Action Taken by I2C Hardware
		I2C_DAT Operation	I2C_CON Operation				
			STA	STO	SI	AA	
0x60	Own SLA+W has been received; ACK has been returned	No action	X	0	0	0	Data byte will be received and NACK will be returned
		No action	X	0	0	1	Data byte will be received and ACK will be returned
0x68	Arbitration lost in SLA+R/W as Master; own SLA+W has been received; ACK has been returned	No action	X	0	0	0	Data byte will be received and NACK will be returned
		No action	X	0	0	1	Data byte will be received and ACK will be returned
0x70	General call address(0x00) has been received; ACK has been returned	No action	X	0	0	0	Data byte will be received and NACK will be returned
		No action	X	0	0	1	Data byte will be received and ACK will be returned
0x78	Arbitration lost in SLA+R/W as Master; General call address(0x00) has been received; ACK has been returned	No action	X	0	0	0	Data byte will be received and NACK will be returned
		No action	X	0	0	1	Data byte will be received and ACK will be returned
0x80	Own slave address has been matched; data byte has been received; ACK has been returned	Read data byte	X	0	0	0	Data byte will be received and NACK will be returned
		Read data byte	X	0	0	1	Data byte will be received and ACK will be returned
0x88	Own slave address has been matched; data byte has been received; NACK has been returned	Read data byte	0	0	0	0	Switched to the not addressed Slave mode; no recognition of own SLA or General Call Address
		Read data byte	0	0	0	1	Switched to the not addressed Slave mode; own SLA or General Call Address will be recognized
		Read data byte	1	0	0	0	Switched to the not addressed Slave mode; no recognition of own SLA or General Call Address ; a START condition will be transmitted when the bus becomes free
		Read data byte	1	0	0	1	Switched to the not addressed Slave mode; own SLA or General Call Address will be recognized;

Status code	I2C status	Software response					Next Action Taken by I2C Hardware
		I2C_DAT Operation	I2C_CON Operation				
			STA	STO	SI	AA	
							a START condition will be transmitted when the bus becomes free
0x90	General Call address has been matched; data byte has been received; ACK has been returned	Read data byte	X	0	0	0	Data byte will be received and NACK will be returned
		Read data byte	X	0	0	1	Data byte will be received and ACK will be returned
0x98	General Call address has been matched; data byte has been received; NACK has been returned	Read data byte	0	0	0	0	Switched to the not addressed Slave mode; no recognition of own SLA or General Call Address
		Read data byte	0	0	0	1	Switched to the not addressed Slave mode; own SLA or General Call Address will be recognized
		Read data byte	1	0	0	0	Switched to the not addressed Slave mode; no recognition of own SLA or General Call Address ; a START condition will be transmitted when the bus becomes free
		Read data byte	1	0	0	1	Switched to the not addressed Slave mode; own SLA or General Call Address will be recognized; a START condition will be transmitted when the bus becomes free
0xA0	A STOP condition or repeated START condition has been received while still addressed as Slave	No action	0	0	0	0	Switched to the not addressed Slave mode; no recognition of own SLA or General Call Address
		No action	0	0	0	1	Switched to the not addressed Slave mode; own SLA or General Call Address will be recognized
		No action	1	0	0	0	Switched to the not addressed Slave mode; no recognition of own SLA or General Call Address ; a START condition will be transmitted when the bus becomes free

Status code	I2C status	Software response					Next Action Taken by I2C Hardware
		I2C_DAT Operation	I2C_CON Operation				
			STA	STO	SI	AA	
		No action	1	0	0	1	Switched to the not addressed Slave mode; own SLA or General Call Address will be recognized; a START condition will be transmitted when the bus becomes free

16.3.5.5 Other modes

Table 16-7. Miscellaneous Status

Status code	I2C status	Software response					Next Action Taken by I2C Hardware
		I2C_DAT Operation	I2C_CON Operation				
			STA	STO	SI	AA	
0xF8	No relevant state information available; SI = "0"	No action	No action				Wait or proceed current transfer
0x00	Bus error due to an illegal START or STOP condition	No action	0	1	0	X	The internal hardware is affected only when in master mode or in slave mode with a matched address. At this point, the bus is released, and the I2C switches to the slave mode (unmatched address state). The "STO" control bit will be reset.

16.4 Register

16.4.1 Overview of registers

Table 16-8 Overview of I2C registers

Offset	Acronym	Register Name	Reset
0x00	I2C_DAT	Data Register	0x00000000
0x04	I2C_ADR	Address Register	0x00000000
0x08	I2C_CON	Control Register	0x00000000
0x0C	I2C_STA	Status Register	0x000000F8
0x10	I2C_DIV	Frequency Divider Register	0x000000F9

16.4.2 I2C_DAT Data Register

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.	DATA
	rw

Bit	Field	Description
31:8	Reserved	Reserved, must be kept at reset value
7:0	DATA	I2C data to be transmitted or received

16.4.3 I2C_ADR Address Register

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						GC	
								rw						rw	

Bit	Field	Description
31:8	Reserved	Reserved, must be kept at reset value
7:1	ADDR	I2C address value when operating as a slave
0	GC	General Call Enable Bit. 1: General Call Address 0x0 will be recognized. 0: General Call Address 0x0 will not be recognized.

16.4.4 I2C_CON Control Register

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.									ENS1	STA	STO	SI	AA	Res.	
									rw	rw	rw	rw	rw		

Bit	Field	Description
31:7	Reserved	Reserved, must be kept at reset value
6	ENS1	I2C Enable Bit When ENS1=0, I2C is disabled, and SDA/SCL output "1". When ENS1=1, I2C is enabled.
5	STA	START Control Bit When the START flag is set by software and the I2C is in master mode, a START signal will be generated on the I2C bus. After sending START, this bit is cleared by software.
4	STO	STOP Control Bit When the STOP flag is set by software and the I2C is in master mode, a STOP signal will be generated on the I2C bus. After sending STOP, this bit is cleared by hardware.
3	SI	Interrupt Flag Bit When an interrupt occurs, the interrupt flag bit is set to "1" by hardware; writing "0" via software clears it.

Bit	Field	Description
2	AA	ACK Response Control Bit When AA=1, an ACK will be returned on the I2C bus in the following situations: - When its own slave address is received. - When I2C_ADR.GC=1 and the general call address is received. - When the I2C is in master receiver mode and receives one data byte. - When the I2C is in slave receiver mode and receives one data byte. When AA=0, an NACK will be returned on the I2C bus in the following situations: - When the I2C is in master receiver mode and receives one data byte. - When the I2C is in slave receiver mode and receives one data byte.
1:0	Reserved	Reserved, must be kept at reset value

16.4.5 I2C_STA Status Register

Address offset: 0x0C

Reset value: 0x0000 00F8

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.								STA							
								r							

Bit	Field	Description
31:8	Reserved	Reserved, must be kept at reset value
7:0	STA	I2C Status codes

16.4.6 I2C_DIV Frequency Divider Register

Address offset: 0x10

Reset value: 0x0000 00F9

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
DIV															
rw															

Bit	Field	Description
31:16	Reserved	Reserved, must be kept at reset value
15:0	DIV	SCL Frequency division factor $SCL\ baud = pclk / (4 * (DIV + 1))$ (DIV >= 16'h3)

17 USART Universal Synchronous Asynchronous Receiver

Transmitter Introduction

The universal synchronous/asynchronous receiver transmitter (USART) offers a flexible means of full-duplex data exchange with external equipment. The USART offers a very wide range of baud rates using a built-in baud rate (including integer and decimal formats) generator.

It supports asynchronous mode (UART), synchronous mode, and the single wire half duplex communication is supported in the asynchronous mode (UART).

17.2 USART features

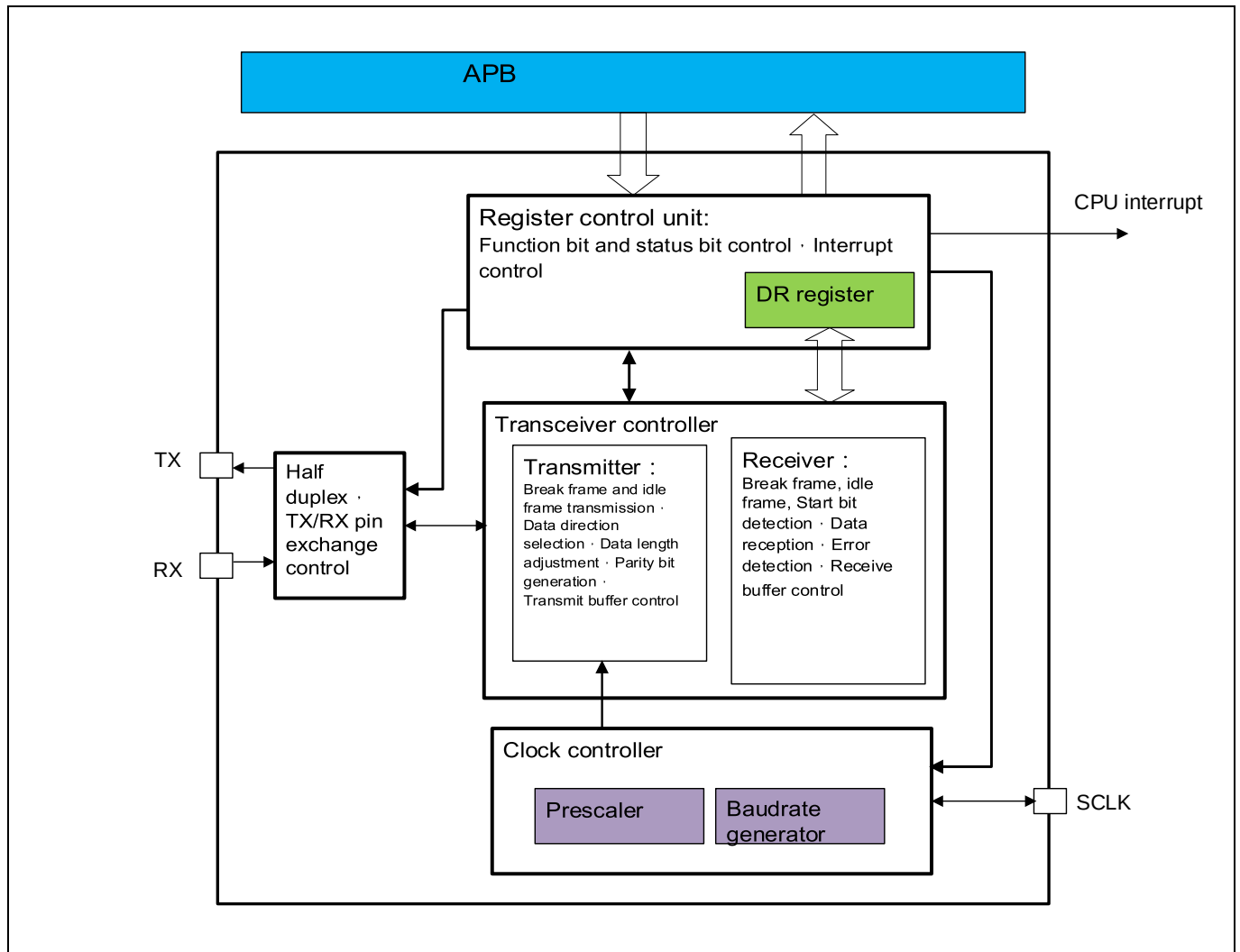
- Supports full-duplex asynchronous communications and full-duplex clock synchronous communications
- Baud rate generator (including integer and decimal formats)
 - ◆ Programmable baud rate, available for a transmitter or receiver (with a minimum division factor of 1)
- Separate transmit and receive FIFO registers, with separate enable bits for transmitter and receiver
- Supports LSB/MSB reception and transmission modes
- Programmable data word length (8 or 9 bits)
- Configurable stop bits (1/2 stop bits)
- Configurable parity check bit (odd parity, even parity, without parity check)
- Supports the idle frame's generation (automatic output when TE is enabled) and receive detection
- Support exchange of Tx and Rx pins and inversion of Tx and Rx signals
- Supports following interrupt sources:
 - ◆ Tx data register empty (TXE)
 - ◆ Transmission complete (TC)
 - ◆ Rx data valid (RXNE)
 - ◆ Rx buffer overrun (OVR)
 - ◆ Rx idle frame complete (IDLE)
 - ◆ Parity error (PE)
 - ◆ Noise flag (NF) and framing error (FE)

17.3 USART functional description

17.3.1 Functional block diagram

The functional block diagram of USART can be referred to as follows: USART can be divided into register-related control unit, receive/transmit data controller, clock controller, and pin control logic unit.

Figure 17-1 USART functional block diagram



17.3.2 Signal description

Signal Name	Type	Description
USART_SCLK	Output or input	Input or output clock pin in synchronous mode
USART_TX	Output or input	Transmit data pin, or half-duplex receive/transmit data pin
USART_RX	Input	Receive data pin (for full duplex)

17.3.3 Functional description

The full-duplex communication requires a minimum of two pins for USART: Receive Data In (RX) and Transmit Data Out (TX).

RX: External serial data is transmitted to the USART receiver through this pin. Oversampling techniques are used for data recovery by discriminating between valid incoming data and noise which is generated during the transmission.

TX: The serial data generated internally in the USART transmitter is output through this pin. When the transmitter is enabled and nothing is to be transmitted, the TX pin is at high level.

The idle state is the initial state of the bus prior to transmission or reception.

There is one start bit represented by '0'.

In the USART communication process, a data frame (8 or 9 bits) can be transmitted and received in least significant bit first (LSB) or in most significant bit first (MSB).

The stop bit indicates the end of a frame with a '1'. The number of bits can be configured as 1 or 2.

The internal baud rate generator is controlled by programming the baud rate register USART_BRR to obtain the desired baud rate for data transmission (refer to the baud rate calculation equation in section 12.3.5).

By configuring the SWAP bit in the USART_CR2 register, the interface of Tx and Rx pins can be swapped.

By configuring the RXTOG/TXTOG bit in the USART_CR2 register, the level signal at the receiving / transmitting terminals can be reversed (including the start bit and stop bit).

In addition, the USART module supports synchronous mode (different from UART), and thus the following pins are required:

- SCLK transmitter clock output or clock input: This pin is used in synchronous mode. In synchronous mode, the clock input and output functions are supported and the clock polarity and phase are configured via software.

17.3.4 Character description

Word length may be selected as being either 8 or 9 bits by programming the USART_CR1.DL bit. The TX pin is pulled down by the transmitter during the transmission start bit. It is pulled up during the transmission stop bit.

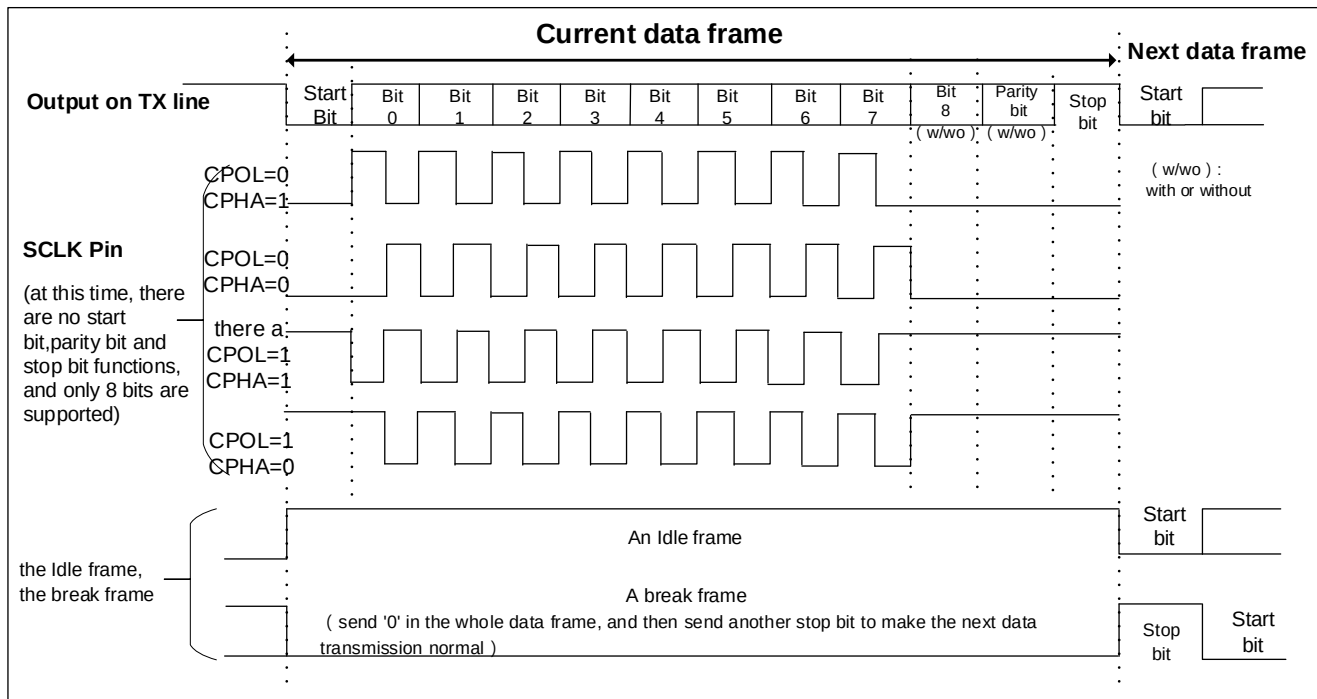
An idle frame is interpreted as an entire data frame of '1's containing the stop bit. The start bit of the next data frame follows the Idle frame.

A Break frame is interpreted as an entire data frame of '0's containing the stop bit. At the end of the Break frame, the transmitter sends another '1' stop bit to acknowledge the start bit of the next frame (generating a falling edge to be detected by the Rx end).

The clock generated by the baud rate generator is provided for the transmitter or receiver, when the enable bit is set respectively for the transmitter and receiver.

The following figure shows a diagram of data frame format, the break frame, and the Idle frame.

Figure 17-2 UART data frame type diagram



17.3.5 Baud rate generator

For each communication mode, the baud rate can be configured according to the following equation.

- For asynchronous mode (UART) and multiprocessor mode (including the multiprocessor under LIN protocol):

The communication baud rate:

$$f_{baudrate} = \frac{PCLK}{N \times (MFD + FFD/N)}$$

Error E (%):

$$E(\%) = \left\{ \frac{PCLK}{f_{baudrate} \times N \times (MFD + FFD/N)} - 1 \right\} \times 100$$

In the above equation, PCLK is the frequency of internal clock source; MFD and FFD are the integer and fractional frequency division that baud rate configuration of the USART_BRR; $N=8 \times (2 - OVER8)$. Select the oversampling mode by configuring the USART_CR1.OVER8; when OVER8=1 (8x oversampling), FFD [3:0] only use the lower 3 bits and user should configure the FFD [3] bit to 1'h0.

- For synchronous mode:

The communication baud rate:

$$f_{baudrate} = \frac{PCLK}{4 \times MFD}$$

In the above equation, PCLK is the frequency of the internal clock source; MFD is the integer baud rate configuration of the USART_BRR. In synchronous mode, fractional frequency division (FFD) is invalid, and user should configure FFD [3:0] to 4'h0.

17.3.6 Sampling

The built-in detection circuit of UART detects the start of a data frame and the RX pin is sampled. The UART uses a clock of 8 or 16 times the data baud rate to sample the data on the RX pin.

The USART_CR1.OVER8 bit can be configured to select whether the USART uses a clock of 16 or 8 times the data baud rate to sample the data on the RX pin.

When 8x oversampling (OVER8=1) is selected, a higher speed (up to $fPCLK/8$) can be obtained, but the maximum tolerance of the receiver to clock deviation will be reduced.

17.3.7 Parity check control

Parity control (generation of parity bit in transmission and parity checking in reception) can be enabled by setting the USART_CR1.PCE bit. The USART_CR1.PS bit is used to select odd parity or even parity for the data.

Even parity: the total number of '1s' inside the data and the parity bit is an even number.

Odd parity: the total number of '1s' inside the data and the parity bit is an odd number.

When parity control is effective :

- The transmitter will automatically generate a parity bit and output it before the stop bit.
- The receiver will detect the parity bit and judge whether it is wrong. If the parity bit is wrong, the hardware will automatically set USART_SR.PE flag, but the current received data will still be transferred from the shift register to USART_DR register.

17.3.8 Transmitter

When the USART_CR1.TE bit is set, the transmitter is enabled and the serial data is output on the TX pin. The transmit data register USART_DR and the internal transmit shift register form a double buffer structure to transmit data continuously. As for UART mode, the data length (8 bits or 9 bits) can be selected by configuring the USART_CR1.DL bit.

17.3.8.1 Character transmission

During a USART transmission, data is written from the USART_DR register, and the data frame byte shifts out least significant bit first (USART_CR1.MLS=0) or most significant bit first (USART_CR1.MLS=1) on the TX pin via the transmit shift register.

Data transmission order: a start bit, character, a parity bit (with or without), stop bits.

The number of stop bits can be configured in the register USART_CR2.STOP [1:0] as 1 or 2.

The USART_CR1.TE bit should not be reset before data transmission is complete. Otherwise, the baud rate generator will stop generating the clocks immediately. Subsequently, the latter part of the data being transmitted are lost.

17.3.8.2 Send break frame

Setting the USART_CR1.SBK bit to transmit a break frame. If the SBK bit is set to '1' during the transmission, the break frame can't be sent on the TX pin until completing the current character transmission.

SBK bit is cleared automatically by hardware when the break frame is completed, and sending an additional high-level stop bit (to ensure that the start bit of the next frame data can be detected).

The length of break frames depends on the data frame length (CR2.DL), parity enable bit (CR1. PS) and stop bit (CR2. STOP). For example, when there is no parity bit and the stop bit is 1 bit, the break frame is 10 consecutive '0' bits if CR2.DL=0, while it is 11 consecutive '0' bits if CR2.DL=1.

17.3.8.3 Transmission configuration procedure

Refer to the following procedures to configure the USART to transmit data frames.

1. Configure the pin functions required by USART.
2. Enable the USART (USART_CR1.UE=1).
3. Configure the USART_BRR.
4. Configure the USART_CR1, USART_CR2, and USART_CR3 registers according to the requirements such as data frame transmission.
5. Enable the transmitter (USART_CR1.TE=1); set USART_CR1.TXEIEN=1 if the transmit data register empty interrupt is needed.
6. Wait until the transmit data register is empty. Write the communication data to USART_DR, and the data is transferred to the transmit shift register. Then the transmission starts.
7. Repeat step 6 if the data should be transmitted continuously.
8. Confirm that the transmission is complete by checking the USART_SR.TC bit. If TCIE=1 is configured, a transition complete interrupt is generated after the transmission of the last frame is complete.

Note: The USART's transmitter supports two types of interrupts, namely the transmit data register empty interrupt TXE and the transition complete interrupt TC, which can be queried by the status bits in the USART_SR register. If TXEIE=1, a TXE interrupt is generated when the value of USART_DR register is transferred to the transmit shift register. If TCIE=1, a TC interrupt is generated when the last bit of data is transmitted and no further data is written to the USART_DR register.

17.3.9 Receiver

The data register USART_DR and the internal receive shift register form a double buffer structure to receive data continuously.

As for UART mode, the data length (8 bits or 9 bits) can be selected by configuring the USART_CR1.DL bit.

After the receiver enable bit USART_CR.RE is set to '1' and a start bit is detected, the data on the RX pin is received into the receive shift register. When a data frame is received, the data is transferred from the receive shift register to the data register USART_DR. Meanwhile, the status flag RXNE will be set to '1'. If RXNEIE=1, the RXNE interrupt request is allowed. When the CPU uses this request to read the received data, the data can be read only once per request.

Data reception order: start bit -> data bit (MSB/LSB) -> parity bit (with or without) -> stop bits.

17.3.9.1 Receive break frame

When the USART receiver recognizes a break frame, it sets the USART_SR.FE flag (equivalent to receiving a '0' in the stop bit).

17.3.9.2 Receive idle character

When UART works normally, the receiver will set the USART_SR.IDLE flag when it receives an idle frame. Configure IDLEIEN=1 to allow the IDLE interrupt request.

17.3.9.3 Reception configuration procedure

Refer to the following procedures to configure the USART to receive data frames.

1. Configure the functional pins required by USART.
2. Enable the USART (USART_CR1.UE=1).
3. Configure the USART_BRR.
4. Configure the USART_CR1, USART_CR2, and USART_CR3 registers according to the requirements such as data frame.
5. Configure the USART_BRR to set the communication baud rate (not necessary if the clock source is external).
6. Enable the receiver (USART_CR1.RE=1), set USART_CR1.RXNEIEN=1 if the receive interrupt is needed.
7. When a start bit is detected, the receiver receives the data into the receive shift register and checks the parity and stop bits. There are three error flags in total: PE, FE, and ORE. When no error occurs, the received data is transferred from the receive shift register to the USART_DR register, and the RXNE flag is set to '1'.
8. The received data can be read through the RXNE interrupt. Repeat step 7 if the data should be received continuously.
9. If any receive error has been detected during reception, the corresponding error flag can be set.

Note: To prevent overflow error, the RXNE bit must be cleared (software reads the data register USART_DR) before the end of the next character reception. Data reception can no longer be performed when any of the PE, FE, or ORE reception errors occur, but data reception can also be restarted by clearing all error flags.

- When an overrun error occurs, the received data is lost and the ORE status bit is set to '1', but the RXNE interrupt is not generated.
- When a parity error occurs, the received data is transferred to USART_DR and the PE status bit is set to '1', but the RXNE interrupt is not generated.
- When a framing error occurs, the received data is transferred to USART_DR and the FE status bit is set to '1', but the RXNE interrupt is not generated.

17.3.10 Synchronous mode

By configuring USART_CR1.SAS bit to '1' to enable synchronous mode (Clock pin function will be valid at the same time).

In synchronous mode, user should configure USART_CR2.HDSEL bit as '0'.

Synchronous mode supports master mode and slave mode: In master mode, the clock generated by the internal baud rate generator, it is used and output at the same time. In slave mode, the clock is input by SCLK pin. In the synchronous mode, USART can achieve data communication with SPI (At this time, user should configure the clock polarity and clock phase of SPI and USART to be consistent).

17.3.10.1 Clock description

Configure the USART_CR2.CLKEN bit to '1' to enable the clock pin function. In addition, select whether to use the internal baud rate clock or the input clock from the SCLK pin for data communication according to the USART_CR3.CKINE bit.

When the internal baud rate clock is selected, a synchronous clock can be output via the SCLK pin.

The transmission and reception of 1 data frame consist of 8 clock pulses.

When both RE and TE are '0', the clock output is stopped and fixed at the level configured by USART_CR2.CPOL.

The clock polarity is selected by configuring the USART_CR2.CPOL bit; the external clock phase is selected by configuring the USART_CR2.CPHA bit.

17.3.10.2 Clock synchronization description

When the SCLK pin is used as the clock output of the transmitter, the clock is output only in the data segment. 8 clock pulses are output for one frame of data. After the last bit is sent, the communication line holds the value of the last bit and the clock output is fixed at a high or low level (determined by the CPOL bit).

The USART receiver works differently in synchronous mode than in asynchronous mode. If RE=1, the data is sampled on the changing SCLK edge (rising or falling edge, depending on the CPOL and CPHA bit configuration) without any oversampling. Sufficient setup time and hold time must be ensured at this point to comply with the timing requirements (similar to SPI protocol).

When the internal clock source is used, the baud rate generated by the internal baud rate generator is calculated in the following equation:

$$f_{baudrate} = \frac{PCLK}{4 \times MFD}$$

Where the communication baud rate is in MBps; PCLK is the frequency of the internal clock source; MFD is the integer baud rate configuration of the USART_BRR (Note that in synchronous mode $MFD \geq 2$ should be configured). In synchronous mode, fractional frequency division (FFD) is invalid, and user should configure FFD [3:0] to 4'h0.

Therefore, when the internal clock source is used and MFD=2, the maximum baud rate for synchronous mode is PCLK/8 (MBps).

When the external clock source is used, the maximum frequency required for the external input clock is PCLK/8 (MHz), at which time the maximum baud rate is also PCLK/8 (MBps).

17.3.11 Single-wire half-duplex communication

Configure the USART_CR3.HDSEL bit to '1' to enable the single-wire half-duplex mode.

In single-wire half-duplex mode, the TX and RX pins are connected through the internal logic of the chip, meanwhile:

- The RX pin is left suspended and not involved in the transmission. The TX of the USART is directly connected to the TX of another USART during transmission.
- When data is being transmitted, the TX remains occupied until the stop bit is sent.
- TX is always released when no data is transmitted. Thus, it acts as a standard IO in idle or in reception. It means that the IO corresponding to TX must be configured as floating input (or output high open-drain) when not driven by the USART.

Apart from the configuration of the single-wire pins, the rest configuration is similar to what is done in normal transmission.

Before the communication, both USARTs have RXEN enabled to stay in the wait-to-receive state. When the communication is required, both USARTs have to agree on which will send, with RE off and TE enabled in the USART_CR1 register of the sender. If both USARTs try to send data, a conflict will occur (the hardware will not block USARTs from sending: when the transmitters enable bit TE is set, TX will transmit data as long as USART_DR is written).

17.3.12 Interrupts

The USART module supports following interrupt sources:

Table 17-1 UART interrupt requests

Interrupt Event	Interrupt Status Bit	Enable Bit	UART	Synchronous mode
Transmit data register empty	TXE	TXEIEEN	√	√
Transmission complete	TC	TCIEEN	√	√
Receive data register full	RXNE	RXNEIEEN	√	√
Idle line detected	IDLE	IDLEIEEN	√	-
Parity error	PE	PEIEEN	√	-
Noise flag	NF	ERRIEN	√	-
Overrun error	ORE	ERRIEN	√	√
Framing error	FE	ERRIEN	√	-

Note: "√" means the interrupt is used. "-" indicates that the interrupt is not used.

17.4 Register

17.4.1 Overview of registers

Table 17-2 Overview of USART registers

Offset	Acronym	Register Name	Reset
0x00	USART_SR	Status register	0x0000_00C0
0x04	USART_DR	Data register	0x0000_01FF
0x08	USART_BRR	Baud rate register	0x0000_0000
0x0C	USART_CR1	Control register 1	0x0000_0000
0x10	USART_CR2	Control register 2	0x0000_0000
0x14	USART_CR3	Control register 3	0x0000_6000

17.4.2 USART_SR status register

Address offset: 0x00

Reset value: 0x0000 00C0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								TXE	TC	RXNE	IDLE	ORE	NF	FE	PE
								r	r	rc_w0	r	r	r	r	r

Bit	Field	Description
31:8	Reserved	Reserved, must be kept at reset value
7	TXE	Transmit data register Empty 0: Transmit data register is not empty (data is not transferred to the shift register) 1: Transmit data register is empty (data is transferred to the shift register) TXE bit is set and cleared automatically by hardware. The hardware will clear TXE when data is not transferred to the shift register (when written to DR register); the hardware will set this bit to '1' when data is transferred from DR to shift register.
6	TC	Transmission Complete 0: Transmission is not complete 1: Transmission is complete TC clearing condition: Write the transmit data to the data register when TE=1. TC setting condition: TE=0 or the transmit data register USART_DR is not updated when the last bit of the data frame is sent.

Bit	Field	Description
5	RXNE *Note 1	Receive data register not empty 0: Valid data is not received 1: Valid data is received Note: RXNE bit is set and cleared by hardware. It can also be cleared by writing '0' to it. RXNE bit is set automatically by hardware when the valid data is received. It is cleared by hardware after reading the received data.
4	IDLE *Note 2	IDLE frame detected 0: No Idle frame is detected 1: Idle frame is detected This bit is set automatically by hardware when an idle frame is detected.
3	ORE *Note 2	OverRun error 0: No overrun error 1: Overrun error is detected Note: This bit is set automatically by hardware when a new data frame is received while RXNE=1 (there already exists readable data).
2	NF *Note 2	Noise detected flag 0: No noise is detected 1: Noise is detected Note: This bit is set automatically by hardware when noise is detected on a received signal line.
1	FE *Note 2	Framing error 0: No framing error is detected 1: Framing error occurred This bit is set automatically by hardware when: In asynchronous (UART) mode, the stop bit of the received data frame is low. Note: The received data will be transferred from the shift register to the data register when FE=1, but no RXNE interrupt request signal will be generated, and the subsequent data reception will be stopped.
0	PE *Note 2	Parity error 0: No parity error 1: Parity error This bit is set automatically by hardware when a parity error occurs during data reception. Note: The received data will be transferred from the shift register to the data register when PE=1, but no RXNE interrupt request signal will be generated, and the subsequent data reception will be stopped.

*Note 1: This bit can be cleared via software by writing '0' to it.

*Note 2: This bit can be cleared by software sequence (reading the status register and then performing a read access to the USART_DR data register).

17.4.3 USART_DR data register

Address offset: 0x04

Reset value: 0x0000 01FF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved							DR[8:0]								
							rw								

Bit	Field	Description
31:9	Reserved	Reserved, must be kept at reset value
8:0	DR[8:0]	Transmit /Receive data register Contains the Received or Transmitted data character, depending on whether it is read from or written to. When reading, it means receiving data; when transmitting, it means sending data. The most significant bit DR [8], is valid only in asynchronous (UART) mode when the data length is set to 9 bits (DL=1).

17.4.4 USART_BRR baud rate register

Address offset: 0x08

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved												MFD [15:12]			
												rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MFD [11:0]												FFD[3:0]			
rw												rw			

Bit	Field	Description
31:16	Reserved	Reserved, must be kept at reset value
19:4	MFD [15:0]	Mantissa frequency division of baudrate These 16 bits define the mantissa division of the USART baud rate. Before transmission or reception is enabled (TE or RE set to 1), this value must be changed according to the baud rate requirement.
3:0	FFD[3:0]	Fraction frequency division of baudrate These 4 bits define the fraction division of the USART baud rate. Before transmission or reception is enabled (TE or RE set to 1), this value must be configured according to the baud rate requirement. Note: When FFD [3:0] =4'h0, the fractional division function is disabled. In asynchronous (UART) mode and if USART_CR1.OVER8=1, FFD [3] is invalid so please configure FFD [3] =0. In synchronous mode, the fractional division is invalid, and should configure FFD [3:0] =4'h0.

17.4.5 USART_CR1 control register 1

Address offset: 0x0C

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved														SAS	MLS
														rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OVER8	Res.	UE	DL	Res.	PCE	PS	PEIEN	TXEIE	TCIEN	RXNEIE N	IDLEIEN	TE	RE	Res.	SBK
rw		rw	rw		rw	rw	rw	rw	rw	rw	rw	rw	rw		rw

Bit	Field	Description
31:18	Reserved	Reserved, must be kept at reset value
17	SAS	Synchronous/Asynchronous mode selection 0: UART mode (asynchronous) 1: Synchronous mode (synchronous) Note: User should configure this bit at TE=0 and RE=0.
16	MLS	MSB/LSB mode selection 0: LSB mode 1: MSB mode Note: User should configure this bit at TE=0 and RE=0.
15	OVER8	UART oversampling mode 0: 16x oversampling 1: 8x oversampling Note: User should configure this bit at TE=0 and RE=0.
14	Reserved	Reserved, must be kept at reset value

Bit	Field	Description
13	UE	USART enable 0: USART prescaler and pin outputs disabled 1: USART enabled When this bit is cleared, the USART prescaler and outputs are stopped and the current transition is stopped in order to reduce power consumption. Note: This bit is set and cleared by software.
12	DL	Data length 0: 8 bits 1: 9 bits Note: User should configure this bit at TE=0 and RE=0.
11	Reserved	Reserved, must be kept at reset value
10	PCE	Parity control enable 0: Parity control disabled 1: Parity control enabled Note: This bit is set or cleared by software. It must be kept at reset value in synchronous mode.
9	PS	Parity selection 0: Even parity 1: Odd parity Note: This bit is set and cleared by software. It is only valid when PCE=1.
8	PEIEN	PE interrupt enable 0: PE interrupt request disabled 1: PE interrupt request enabled Note: This bit is set and cleared by software.
7	TXEIEEN	TXE interrupt enable 0: TXE interrupt request disabled 1: TXE interrupt request enabled Note: This bit is set and cleared by software.
6	TCIEN	Transmission complete interrupt enable 0: TC interrupt request disabled 1: TC interrupt request enabled Note: This bit is set and cleared by software.
5	RXNEIEN	RXNE interrupt enable 0: RXNE interrupt request disabled 1: RXNE interrupt request enabled Note: This bit is set and cleared by software.
4	IDLEIEN	IDLE interrupt enable 0: IDLE interrupt request disabled 1: IDLE interrupt request enabled Note: This bit is set and cleared by software.
3	TE	Transmitter enable 0: Transmitter disabled 1: Transmitter enabled Note: This bit is set and cleared by software. In synchronous mode, if simultaneous transmission and reception are required, user must configure both TE and RE bits at the same time to ensure proper timing sequence of clock and data.
2	RE	Receiver enable 0: Receiver disabled 1: Receiver enabled Note: This bit is set and cleared by software. In synchronous mode, if simultaneous transmission and reception are required, user must configure both RE and TE bits at the same time to ensure proper timing sequence of clock and data.
1	Reserved	Reserved, must be kept at reset value
0	SBK	Send break 0 : No break frame is transmitted 1: Break frame will be transmitted This bit set is used to send break frame. It can be set by software and cleared automatically by hardware after sending the break frame.

17.4.6 USART_CR2 control register 2

Address offset: 0x10

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SWAP	Res.	STOP[1:0]		Res.	CPOL	CPHA	Res.								
rw		rw			rw	rw									

Bit	Field	Description
31:15	Reserved	Reserved, must be kept at reset value
15	SWAP	Swap pin between input and output 0: IO pin functions are not swapped 1: Input and output of IO pin functions are swapped Note: Once the SWAP bit is set, the MODE in the GPIOx_CRL register needs to be changed, e.g. the original input mode changes to the output mode.
14	Reserved	Reserved, must be kept at reset value
13:12	STOP[1:0]	STOP bit UART mode: 00: 1 stop bit 10: 2 stop bits 01: Reserved 11: Reserved
11	Reserved	Reserved, must be kept at reset value
10	CPOL	Clock polarity 0: The clock is at a low level when idle 1: The clock is at a high level when idle Note: This bit works in conjunction with the CPHA bit to produce the desired clock/data relationship (only valid in synchronous mode).
9	CPHA	Clock phase 0: The first clock transition is the first data capture edge 1: The second clock transition is the first data capture edge. Note: This bit works in conjunction with the CPOL bit to produce the desired clock/data relationship (only valid in synchronous mode).
8:0	Reserved	Reserved, must be kept at reset value

17.4.7 USART_CR3 control register 3

Address offset: 0x14

Reset value: 0x0000 6000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.		TXTOG	RXTOG	Reserved											CKINE
		rw	rw												rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.				ONEBIT	Res.							HDSEL	Res.		ERRIEN
				rw								rw			rw

Bit	Field	Description
31:30	Reserved	Reserved, must be kept at reset value
29	TXTOG	Transmit toggle bit 0: Transmit toggle disabled 1: Transmit signal level's toggle enabled
28	RXTOG	Receive toggle bit 0: Receive toggle disabled 1: Receive signal level's toggle enabled

Bit	Field	Description
27:17	Reserved	Reserved, must be kept at reset value
16	CKINE	Clock input permission in synchronous mode This bit is used to control whether the clock is input externally (This bit is valid when CR1. SAS bit is '1'): 0: Clock is not input externally 1: Clock is input externally Note: User should configure this bit at TE=0 and RE=0.
15:12	Reserved	Reserved, must be kept at reset value
11	ONEBIT	UART one sample bit method enable 0: Three samples (majority decision) 1: One sample Note: User should configure this bit at TE=0 and RE=0. When one sample bit method is selected, the noise detected flag (USART_SR.NF) will be invalid.
10:4	Reserved	Reserved, must be kept at reset value
3	HDSEL	Single wire Half-duplex selection 0: Full-duplex mode 1: Half-duplex mode
2:1	Reserved	Reserved, must be kept at reset value
0	ERRIEN	Error interrupt enable 0: Error interrupt request disabled 1: Error interrupt request enabled Error interrupts include FE, ORE, and NF. Note: when reading or writing to DR, user can configure ERRIEN=1 to allow interrupt requests that sending to CPU for USART's abnormal communication.

18 SYSCFG System Controller

18.1 Introduction

The chip is composed of a set of system configuration registers. Main functions of these registers:

- Manage external interrupt connected to GPIO port (pin configuration)
- Remap memory to code initial area
- System level configuration of some peripherals
-

18.2 GPIOA 7-segment display driver

The chip provides dedicated I/O pins capable of driving up to five 8-segment common-cathode LED displays. Up to five specific I/O pins on GPIOA can be configured for current-sinking mode (maximum 80 mA) to control the common cathodes of digital tube displays; up to eight specific I/O pins can be configured for constant-current source mode (with selectable settings of 2.5, 5, 7.5, or 10 mA) to control the segment selection of the displays.

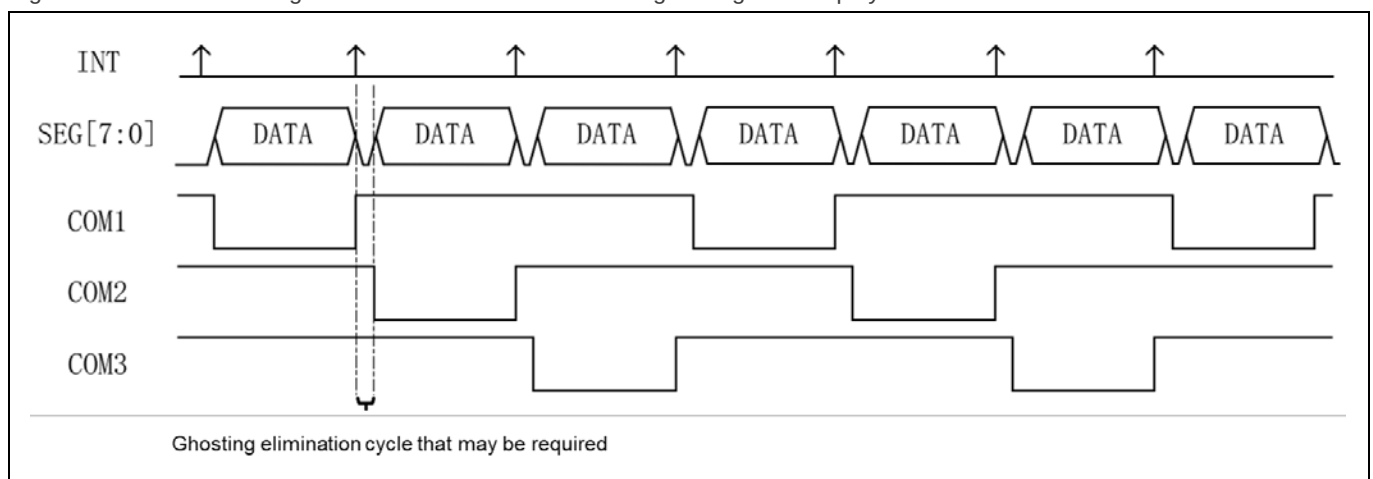
In current-sinking mode, this I/O features enhanced current-sinking capability but otherwise functions identically to a standard GPIO. In constant-current source mode, the IO input function is forcibly disabled; configuring ODR = 1 outputs current, while configuring ODR = 0 cuts off the current and leaves the IO floating, if the corresponding LEDIOx[1] is configured to 1, the IO can be forced to output 0.

The display process is described as follows:

- Configure RCC->AHBENR[17] to enable the GPIOA clock, and RCC->APB1ENR[30] to enable the SYSCFG clock.
- Configure GPIOA by switching the required GPIO pins to general-purpose push-pull output mode.
- Configure SYSCRG_LEDIOCR and SYSCREG_HCIOCR to switch the required GPIOs to current-sink mode or constant-current source mode.
- Set up a timer-based polling mechanism or interrupt, and configure GPIOA_ODR to perform periodic segment switching and digit scanning.
- Adjust the blanking period based on the display output.

A typical output waveform is shown in the figure below:

Figure 18-1 Schematic diagram of drive waveforms for a 3-digit 7-segment display



Note: Before entering low-power modes such as Stop or Deep Stop, the ODR must be configured to turn off the digital tube display; once the system exits low-power mode, the ODR must be reconfigured to drive the display again.

18.3 Internal Reference Voltage

The chip provides an internal reference voltage for use by the Analog-to-Digital Converter (ADC) and the comparator (COMP). The ADC can sample and convert the internal reference voltage value via channel 12, while the comparator allows the internal reference voltage to be selected as the inverting input via inverting input channel 4.

The internal reference voltage is derived from two sources. The first source is an analog voltage sensor that outputs a fixed 1.2V reference voltage; the second source is an analog temperature sensor that outputs a voltage inversely proportional to temperature.

When only the VS_EN bit in SYSCFG_SENSORCR is set, the internal reference voltage selects the output voltage of the analog voltage sensor; when only the TS_EN bit in SYSCFG_SENSORCR is set, the internal reference voltage selects the output voltage of the analog temperature sensor. If both TS_EN and VS_EN are set, the internal reference voltage is selected from the analog voltage sensor output; in this state, the source of the internal reference voltage can be rapidly switched by clearing or setting VS_EN.

18.4 Register

18.4.1 Register overview

Table 18-1 SYSCFG register overview

Offset	Acronym	Register Name	Reset
0x00	SYSCFG_CFGR	SYSCFG configuration register	0x00000000
0x08	SYSCFG_EXTICR1	SYSCFG external interrupt configuration register 1	0x00000000
0x0C	SYSCFG_EXTICR2	SYSCFG external interrupt configuration register 2	0x00000000
0x10	SYSCFG_EXTICR3	SYSCFG external interrupt configuration register 3	0x00000000
0x14	SYSCFG_EXTICR4	SYSCFG external interrupt configuration register 4	0x00000000
0x1C	SYSCFG_SENSORCR	SYSCFG Analog Sensor Configuration Register	0x00000000
0x20	SYSCFG_LEDIOCR	SYSCFG Constant-Current Source IO Configuration Register	0x00000000
0x24	SYSCFG_HCIOCR	SYSCFG Sink Current I/O Configuration Register	0x00000000

18.4.2 SYSCFG_CFGR Configuration Register

This register has two control bits MEM_MODE, and is used to configure the initial address 0x00000000.

Address offset: 0x0

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.														MEM_MODE	
														rw	

Bit	Field	Description
31: 2	Reserved	Reserved, must retain the reset value

Bit	Field	Description
1:0	MEM_MODE	Memory Mode Selection Bit Control memory internal mapping to address 0x0000 0000. x0: Main Flash memory mapping to 0x0000 0000 01: System Flash mapping to 0x0000 0000 11: Built-in RAM mapping to 0x0000 0000

18.4.3 SYSCFG_EXTICR1 External Interrupt Configuration Register 1

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
EXTI3				EXTI2				EXTI1				EXTI0			
rw				rw				rw				rw			

Bit	Field	Description
31:16	Reserved	Reserved, must retain the reset value
15:0	EXTIx	EXTIx configuration (x=0 ...3) Select the input source of EXTIx external interrupt. 0000: PA[x] pin 0001: PB[x] pin

18.4.4 SYSCFG_EXTICR2 External Interrupt Configuration Register 2

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
EXTI7				EXTI6				EXTI5				EXTI4			
rw				rw				rw				rw			

Bit	Field	Description
31:16	Reserved	Reserved, must retain the reset value
15:0	EXTIx	EXTIx configuration (x=4...7) Select the input source of EXTIx external interrupt. 0000: PA[x] pin 0001: PB[x] pin

18.4.5 SYSCFG_EXTICR3 External Interrupt Configuration Register 3

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
EXTI11				EXTI10				EXTI9				EXTI8			

rw	rw	rw	rw
----	----	----	----

Bit	Field	Description
31:16	Reserved	Reserved, must retain the reset value
15:0	EXTIx	EXTIx configuration (x=8...11) Select the input source of EXTIx external interrupt. 0000: PA[x] pin 0001: PB[x] pin

18.4.6 SYSCFG_EXTICR4 External Interrupt Configuration Register 4

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
EXTI15				EXTI14				EXTI13				EXTI12			
rw				rw				rw				rw			

Bit	Field	Description
31:16	Reserved	Reserved, must retain the reset value
15:0	EXTIx	EXTIx configuration (x=12...15) Select the input source of EXTIx external interrupt. 0000: PA[x] pin 0001: PB[x] pin

18.4.7 SYSCFG_SENSORCR Analog Sensor Configuration Register

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.														TS_EN	VS_EN
														rw	rw

Bit	Field	Description
31:2	Reserved	Reserved, must retain the reset value
1	TS_EN	Temperature sensor enable 0 : Disable analog temperature sensor 1 : Enable analog temperature sensor
0	VS_EN	Voltage sensor enable 0 : Disable analog voltage sensor 1 : Enable analog voltage sensor

18.4.8 SYSCFG_LEDIOCR Constant-Current Source IO Configuration Register

Address offset: 0x20

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LEDIO7.				LEDIO6				LEDIO5				LEDIO4			
rw				rw				rw				rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LEDIO3				LEDIO2				LEDIO1				LEDIO0			
rw				rw				rw				rw			

Bit	Field	Description
31:28	LEDIO7	Constant Current Source IO 7 Configuration (PA15) Refer to LEDIO0 bit
27:24	LEDIO6	Constant Current Source IO 6 Configuration (PA12) Refer to LEDIO0 bit
23:20	LEDIO5	Constant Current Source IO 5 Configuration (PA11) Refer to LEDIO0 bit
19:16	LEDIO4	Constant Current Source IO 4 Configuration (PA10) Refer to LEDIO0 bit
15:12	LEDIO3	Constant Current Source IO 3 Configuration (PA9) Refer to LEDIO0 bit
11:8	LEDIO2	Constant Current Source IO 2 Configuration (PA8) Refer to LEDIO0 bit
7:4	LEDIO1	Constant Current Source IO 1 Configuration (PA3) Refer to LEDIO0 bit
3:0	LEDIO0	Constant Current Source IO 0 Configuration (PA2) [3:2] : Constant current source IO current range 11 : 10.0mA 10 : 7.5mA 01 : 5.0mA 00 : 2.5mA [1] : N-type transistor configuration for a constant current source IO 1 : When ODR = 0, the constant current source IO outputs 0. 0 : When ODR = 0, the constant current source IO is floating. [0] : Constant current source IO enable 1 : Constant-current source mode 0 : GPIO mode

18.4.9 SYSCFG_HCIOCR Sink Current I/O Configuration Register

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.											HCI04	HCI03	HCI02	HCI01	HCI00
											rw	rw	rw	rw	rw

Bit	Field	Description
31:5	Reserved	Reserved, must retain the reset value
4	HCI04	Sink Current IO 4 Refer to HCI00 bit
3	HCI03	Sink Current IO 3 Refer to HCI00 bit
2	HCI02	Sink Current IO 2 Refer to HCI00 bit
1	HCI01	Sink Current IO 1 Refer to HCI00 bit
0	HCI00	Sink Current IO 0 1 : Sinking current mode 0 : GPIO mode

19 Device Electronic Signature

19.1 Overview

The Device electronic signature is stored in the system memory area in the Flash memory module, and can be read using the SWD or the CPU. It contains factory program identification data that allow the user firmware or other external devices to automatically match its interface to the characteristics of the microcontroller.

The unique device identifier is ideally suited:

- for use as serial numbers
- for use as security keys in order to increase the security of code in Flash memory while
- using and combining this unique ID with software encryption-decryption algorithm before programming the internal Flash memory
- to activate secure boot processes
- The 96-bit unique device identifier provides a reference number which is unique for any microcontroller and in any context. These bits can never be altered by the user.
- The 96-bit unique device identifier can also be read in single bytes (8 bits)/half-words (16 bits)/words (32 bits) in different ways.

19.2 Register description

Base address: 0x1FFF F7E8

Table 19-1 Device electronic signature register overview

Offset	Acronym	Register Name	Reset
0x00	UID1	Unique identifier	0xFFFFFFFF
0x04	UID2	Unique identifier	0xFFFFFFFF
0x08	UID3	Unique identifier	0xFFFFFFFF

19.2.1 UID1 Unique Identifier

Address offset : 0x00

Reset value: This value is written by default.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
U_ID (31: 16)															
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
U_ID (15: 0)															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															
r															

19.2.2 UID2 Unique Identifier

Address offset : 0x04

Reset value: This value is written by default.

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

U_ID (63: 48)

r

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

U_ID (47: 32)

r

Bit	Field	Description
31 : 0	U_ID (63 : 32)	U_ID : 63 : 32 unique ID bits

19.2.3 UID3 Unique Identifier

Address offset : 0x08

Reset value: This value is written by default.

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

U_ID (95: 80)

r

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

U_ID (79: 64)

r

Bit	Field	Description
31 : 0	U_ID (95 : 64)	U_ID : 95 : 64 unique ID bits

20 DBG Debug Support

20.1 Introduction

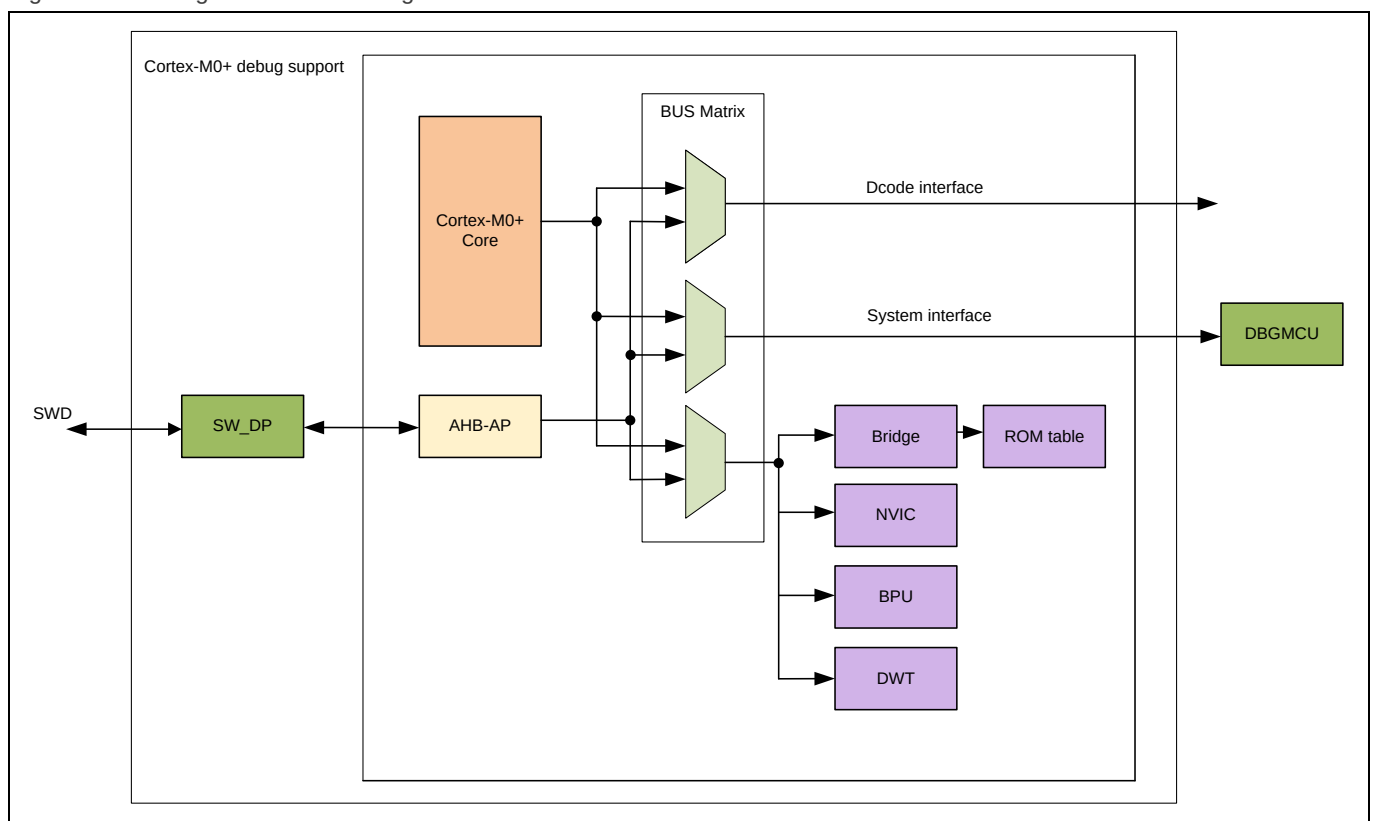
The MCU core includes the debug module mainly used for the function commissioning. When the core gets address (address breakpoint) or accesses data (data breakpoint), the hardware debug module may control the core to stop. The user can enquire the core internal status and system status. After enquiry, the core may continue to execute the current program.

When the chip and debugger connect and begin debugging, the debugger auto calls the core debug module for debug operation.

20.2 Function descriptions

20.2.1 Function block diagram

Figure 20-1 Debug function block diagram



Cortex-M0 core includes the debug unit, and it's composed of:

SWDP: SW debug port

BPU: Breakpoint debug unit

DWT: Data watch point and trace

20.2.2 SWD internal pull up and down

SWD pin input directly controls the debug mode, and cannot be floated. To guarantee that I/O level is controllable, SWD pin has the built in pull up and down resistance.

- SWDIO: Internal pull up
- SWCLK: Internal pull down

The software can use these I/O ports as the ordinary I/O port. At this time, the pull up/down function closes by default. Refer to the chapter of General Purpose Input/output GPIO.

20.2.3 SWD debug port

The chip's two ordinary I/O ports are used as SWD-DP interface pin. These pins in different packages all support SWD debug port.

Table 20-1 SWD debug port pin

SWJ-DP port pin name	SW debug interface		Pin allocation
	Type	Debug function	
SWDIO	Input/output	Serial data input/output	PA13
SWCLK	Input	Serial clock	PA14

20.3 ID code

The chip has many ID codes, as seen below:

Table 20-2 ID code

ID Name	Chip
DEV_ID	0x4C514700
CPU TAP SW ID	0x0BC11477

20.3.1 Microcontroller device ID code

The microcontroller contains the device ID code. This ID defines MCU chip version, and is mapped on the external APB bus. This code can be acquired by the user code and debug interface.

20.3.2 Cortex JEDEC-106 ID code

The microcontroller has a JEDEC-106ID code. It's located in the 4KB ROM table mapped to the internal PPB bus address 0xE00FF000_0xE00FFFFF

20.4 SW debug port

20.4.1 SW protocol introduction

This sync serial protocol uses 2 pins: Clock signal (SWCLK) and two-way data signal (SWDIO) from host to target.

As the two-way data line, SWDIO pin has the built in pull up resistor, and the extra external resistor is not required.

Data transmission begins from the low bit, and allow reading and writing the register DPACC and APACC.

According to the protocol, when SWDIO changes direction, insert one conversion time (one Bit time by default, adjustment by SWCLK). In this period, any device cannot drive the signal line.

20.4.2 SW protocol series

One series contains three phases:

- The host sends 8 bit request packet;
- The target sends 3 bit acknowledgement response;
- According to the configuration direction, the host or target sends 33 bit (including one check bit) data;

Table 20-3 8bit request packet

Bit	Name	Description
0	Beginning	Must be 1
1	APnDP	0: Access DP 1: Access AP
2	RnW	0: Write request 1: Read request
4: 3	A[3 : 2]	DP or AP register address
5	Parity	Previous bit check bit
6	Stop	0
7	Park	Cannot be driven by host. Because of pull up, the target is always read as 1

Note: Each request packet is followed by 1 bit conversion time. For more information on DAPCC and APACC register, refer to the relevant CPU technology data sheet of ARM.

Table 20-4 3bit response package

Bit	Name	Description
2: 0	ACK	001: Fail 010: Wait 100: Success

Note : When the response (ACK) signal is one of scenarios above, the response bit is followed by one conversion time.

Table 20-5 33bit data packet

Bit	Name	Description
31: 0	WDATA/RDATA	Read or write data
32	Parity	32 bit data parity check bit

Note: Wait for one conversion time after read data bit.

20.4.3 SW-DP status unit (Reset, Idle states, ID code)

SW-DP status unit uses the internal ID code to identify SW_DP. Observe JEP-106 standard. For specific information, please refer to the relevant ARM manual.

SW-DP status unit doesn't work until before the debugger reads ID.

- In case of power on reset, or DP switches from JTAG to SWD, or exceeds high level of 50 periods, SW-DP status unit is in the reset status;
- If the low level of at least 2 periods appear after RESET status, the status unit will switch to the IDLE status;
- When the status unit is in the reset status, first switch to the IDLE status, and read DP-SW ID register. Otherwise, the debugger cannot conduct other normal transmission, and ACK Fault will appear;

20.4.4 DP and AP read/write access

- DP read operation has no delay: The debugger directly acquires data (in case of ACK return success status) or waits (in case of ACK return wait status);
- AP read operation has latency. It means that the previous read operation result can only be acquired during the next operation. If the next operation is not access to AP, it's mandatory to read DP-RDBUFF register to acquire the previous read operation result;
- DP-CTRL/STAT register READOK flag bit is refreshed after AP read operation and RDBUFF read operation to inform the debugger whether AP read operation succeeds;

- SW-DP has the write buffer (DP and AP have the write buffer) so that the write operation can be accepted when other transmissions are ongoing. If the write buffer is full, the debugger will acquire one wait ACK response. Read IDCODE register, read CTRL/STAT register and write ABORT register operation remain accepted when the write buffer is full;
- Because SWCLK and HCLK are not in step, insert two extra SWCLK periods after the write operation (after parity check bit) to ensure that the internal write operation is correctly completed. These two extra clock periods should be inserted when the line is in IDLE status. This operation step is particularly important during writing CTRL/STAT register to propose one power on request. Otherwise, the next operation (valid operation after core power on) will execute immediately. It will lead to failure;

20.4.5 SW-DP register

In case of APnDP=0, access to these registers below.

Table 20-6 SW-DP register

A[3:2]	Read/write	SELECT register's CTRLSEL bit	Register	Description
00	Read		IDCODE	Fixed as 0x0BB11477 (for identifying SW-DP)
00	Write		ABORT	
01	Read/Write	0	DP-CTRL/STAT	Request one system or debug power on operation; configure AP access operation mode; Control compare, check operation; Read some status bits (overflow, power on response).
01	Read/Write	1	WIRE CONTROL	Configure the serial communication physical layer protocol (such as conversion time length, etc.)
10	Read		READ RESEND	Allow to restore data from one error debug transmission rather than repeat the initial AP transmission.
10	Write		SELECT	Select current access port and valid 4 word register window.
11	Read/Write		READ BUFFER	This register will capture the data result of the previous read operation from AP. Therefore, the data can be acquired without again enabling the new AP transmission

20.4.6 SW-AP register

In case of APnDP=1, access and AP register access address is composed of two parts:

- A[3: 2] value
- DP SELECT register's current value

20.5 MCU debug module (DBGMCU)

MCU debug module provides the debugger assist function below:

- Support low power mode
- Breakpoint timer and watchdog clock control

20.5.1 Debug support at low power mode

MCU has multiple low power modes. It can close CPU clock, reduce CPU power consumption, and enter the low power mode by executing WFE or WFI command. FCLK and AHB bus clock HCLK are mandatory for debug operation, and cannot be closed. MCU is equipped with some registers to change the low power mode characteristics, thus supporting the debug code in the low power mode. Specific configurations:

- Enter the sleep mode. In order to make HCLK and FCLK have the same clock, the debugger must set the DBG_CR register DBG_SLEEP bit.

- Enter the stop mode. It's mandatory to configure DBG_STOP bit. The operation will activate the internal oscillator HSI, thus providing the clock for FCLK and HCLK.

20.5.2 Support timer, watchdog

In case of the breakpoint, select the timer's work mode according to the application of timer and watchdog;

- Counter count continues. The application is in the PWM wave control motor
- Counter count stops. The application is in the watchdog counter.

20.6 Register

20.6.1 Register overview

Base address: 0x4001 3400

Table 20-7 DBG register overview

Offset	Acronym	Register Name	Reset
0x00	DBG_IDCODE	DBG ID encode register	0x4C514700
0x04	DBG_CR	DBG control register	0x00000000

20.6.2 DBG_IDCODE ID Encode Register

Address offset: 0x00 (only support 32-bit access, read only)

Reset value: 0x4C514700

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DEV_ID															
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DEV_ID															
r															
Bit	Field		Description												
31:0	DEV_ID		Device Identifier Read only register, always read as reset value												

20.6.3 DBG_CR Control Register

Address offset: 0x04 (only support 32 bit access)

Reset value: 0x0000 0000(POR reset only, not reset by system reset)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.													DBG_ TIM14_ S TOP	Res.	
													rw		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DBG_ TIM3_ PWM_ OFF	Res.	DBG_ TIM1_ PWM_ OFF	DBG_ TIM3_ STOP	Res.	DBG_ TIM1_ STOP	Res.	DBG_ IWDG_ S TOP	Res.				DBG_ STOP_ FOR_ LDO	Res.	DBG_ STOP	DBG_ SLEEP
rw		rw	rw		rw		rw					rw		rw	rw
Bit		Field				Description									
31:19		Reserved				Reserved, must retain the reset value									
18		DBG_TIM14_STOP				TIM14 stops work when the core enters the debug mode 0: TIM14 still works normally when the timer is selected 1: TIM14 stops when the timer is selected									

Bit	Field	Description
17:16	Reserved	Reserved, must retain the reset value
15	DBG_TIM3_PWM_OFF	TIM3 PWM outputs 0 when the core enters the debug mode 0: TIM3 PWM normal outputs 1: TIM3 PWM outputs 0
14	Reserved	Reserved, must retain the reset value
13	DBG_TIM1_PWM_OFF	TIM1 PWM outputs 0 when the core enters the debug mode 0: TIM1 PWM normal outputs 1: TIM1 PWM outputs 0
12	DBG_TIM3_STOP	TIM3 stops work when the core enters the debug mode 0: TIM3 still works normally when the timer is selected 1: TIM3 stops when the timer is selected
11	Reserved	Reserved, must retain the reset value
10	DBG_TIM1_STOP	TIM1 stops work when the core enters the debug mode 0: TIM1 still works normally when the timer is selected 1: TIM1 stops when the timer is selected
9	Reserved	Reserved, must retain the reset value
8	DBG_IWDG_STOP	Independent watchdog stops work This bit is not related to whether the core enters the debug status 0: The watchdog counter still works normally 1: The watchdog counter stops work
7:4	Reserved	Reserved, must retain the reset value
3	DBG_STOP_FOR_LDO	Debug stop mode LDO status 0: LDO enters the low power mode, and normally enters the STOP mode 1: LDO doesn't enter the low power mode, PLL sustains power on, and cannot truly enter STOP mode. CPU enters DEEPSLEEP, and HCLK closes
2	Reserved	Reserved, must retain the reset value
1	DBG_STOP	Debug stop mode 0: In the stop mode, the clock controller disables all clocks (including HCLK and FCLK). When exiting from STOP mode, the clock configuration is consistent with the configuration after reset. If it's required to again enable PLL, HSE clock, the software must reconfigure the clock control system. 1: In the stop mode, FCLK and HCLK open, and HSI provides the clock. When exiting from STOP mode, make reconfiguration by software if it's required to again enable PLL, HSE clock
0	DBG_SLEEP	Debug sleep mode 0: In the sleep mode, clock FCLK opens, FCLK keeps the configured system clock by default, and HCLK closes. The sleep mode won't reset the configured clock system. When exiting the sleep mode, the software doesn't need to reconfigure the system clock 1: In the sleep mode, FCLK and HCLK clock is provided by the formally configured system.

21 History Records

Table 21-1 History records

Date	Version	Content
2026/09/01	0.01	Initial Version