



武汉芯源半导体有限公司

WUHAN XINYUAN SEMICONDUCTOR CO., LTD

CW32L012 Reference Manual

ARM® Cortex®-M0+ Low-power 32-bit MCU

Rev 1.0

June 2026

<https://www.whxy.com>

Statement

Important notice

This reference manual is aimed at application developers and provides complete information on how to use the CW32L012 microcontroller memory and peripherals. This operation applies only to CW32L012 devices.

Disclaimer

Wuhan Xinyuan Semiconductor Co., Ltd. reserves the right to make changes, corrections, enhancements, and modifications to the product and this document at any time without notice. Buyers should obtain the latest product information on the official website.

Copyright statement

The copyright of this manual belongs to Wuhan Xinyuan Semiconductor Co., LTD., and we reserve the right of final interpretation and modification of this manual and statement.

Technical support

If you have any comments or suggestions in the process of using, please feel free to contact us.

Website: <https://www.whxy.com>.

Address: 5 WuDaYuan 3rd Road, Donghu High-tech Development Zone Wuhan Hubei, China.

Postcode: 430070.



Contents

Statement.....	i
1 Documentation conventions.....	1
1.1 Typographical conventions.....	1
1.2 Register protocol.....	2
2 System and memory overview.....	3
2.1 System architecture.....	3
2.2 Memory organization.....	5
2.2.1 Overview.....	5
2.2.2 Memory map and register boundary addresses.....	7
2.3 On-chip SRAM memory.....	9
2.4 On-chip FLASH memory.....	10
2.5 One-time programmable OTP memory.....	11
2.6 System boot configuration.....	12
2.7 Precautions.....	13
3 Power control (PWR) and power consumption.....	14
3.1 Overview.....	14
3.2 Power supply supervisor.....	15
3.3 Operating mode.....	16
3.3.1 Entering Sleep mode or DeepSleep mode.....	17
3.3.2 Exiting Sleep mode or DeepSleep mode.....	18
3.3.3 Operating mode and reset source.....	20
3.4 Low-power consumption applied.....	21
3.5 Cortex®-M0+ kernel system control register (SCB->SCR).....	22
4 Reset and clock control (RCC).....	23
4.1 System reset.....	23
4.1.1 Power on reset/Brown-out reset (POR/BOR).....	23
4.1.2 Pin input reset (NRST).....	24
4.1.3 IWDG/WWDT reset.....	24
4.1.4 LVD low voltage detection reset.....	24
4.1.5 Kernel SYSRESETREQ reset.....	24
4.1.6 Kernel LOCKUP fault reset.....	24
4.2 Peripheral reset.....	25
4.3 Clock and Control.....	26
4.3.1 Overview.....	26
4.3.2 System clock and operating mode.....	28



4.3.3	HSE clock.....	29
4.3.4	HSIOSC clock.....	31
4.3.5	LSE clock.....	32
4.3.6	LSI clock.....	34
4.3.7	SysClk system clock.....	34
4.3.8	On-chip peripheral clock control.....	35
4.4	Clock Start, calibration, and status detection.....	36
4.4.1	Clock start.....	36
4.4.2	Clock calibration.....	37
4.4.3	Clock state detection.....	38
4.4.3.1	Clock stability detection.....	38
4.4.3.2	Clock start-up failure detection.....	39
4.4.3.3	In-flight failure detection of clocks.....	40
4.4.4	Clock verification and output.....	41
4.5	SysClk system clock switch.....	42
4.5.1	Standard clock switching process.....	43
4.5.2	HSI clock switching process between different frequencies.....	44
4.5.3	Example of switching from other clocks to LSE.....	44
4.5.4	Example of switching from other clocks to HSE.....	45
4.5.5	Example of switching from other clocks to LSI.....	45
4.5.6	Example of switching from other clocks to HSI.....	46
4.6	List of registers.....	47
4.7	Register description.....	48
4.7.1	SYSCTRL_CR0 System control register 0.....	48
4.7.2	SYSCTRL_CR1 System control register 1.....	49
4.7.3	SYSCTRL_CR2 System control register 2.....	51
4.7.4	SYSCTRL_HSI Internal high frequency clock control register.....	52
4.7.5	SYSCTRL_LSI Internal low frequency clock control register.....	53
4.7.6	SYSCTRL_HSE External high frequency crystal control register.....	54
4.7.7	SYSCTRL_LSE External low frequency crystal control register.....	56
4.7.8	SYSCTRL_IER System interrupt enable control register.....	57
4.7.9	SYSCTRL_ISR System interrupt flag register.....	58
4.7.10	SYSCTRL_ICR System interrupt flag clear register.....	60
4.7.11	SYSCTRL_AHBEN AHB peripheral clock enable control register.....	61
4.7.12	SYSCTRL_APBEN1 APB peripheral clock enable control register 1.....	62
4.7.13	SYSCTRL_APBEN2 APB peripheral clock enable control register 2.....	64
4.7.14	SYSCTRL_AHBRST AHB peripheral reset control register.....	65
4.7.15	SYSCTRL_APBRS1 APB peripheral reset control register 1.....	66
4.7.16	SYSCTRL_APBRS2 APB peripheral reset control register 2.....	68
4.7.17	SYSCTRL_RESETFLAG System reset flag register.....	69



4.7.18	SYSCTRL_DEBUG Debug status timer control register.....	70
4.7.19	SYSCTRL_MCO System clock output control register.....	71
5	Interrupt.....	72
5.1	Overview.....	72
5.2	Main features.....	73
5.3	Interrupt priority.....	74
5.4	Interrupt vector table.....	75
5.5	Interrupt related registers.....	77
5.5.1	NVIC interrupt enable and disable enable.....	77
5.5.2	NVIC interrupt pending and clear pending.....	77
5.5.3	NVIC interrupt priority.....	77
5.5.4	NVIC interrupt mask.....	78
5.5.5	Peripheral interrupt enable.....	78
5.6	List of registers.....	79
5.7	Register description.....	80
5.7.1	NVIC_IUSER Interrupt enable setting register.....	80
5.7.2	NVIC_ICER Interrupt enable clear register.....	80
5.7.3	NVIC_ISPR Interrupt pending setup register.....	81
5.7.4	NVIC_ICPR Interrupt pending clear register.....	81
5.7.5	NVIC_IPR0 Interrupt priority control register 0.....	82
5.7.6	NVIC_IPR1 Interrupt priority control register 1.....	83
5.7.7	NVIC_IPR2 Interrupt priority control register 2.....	83
5.7.8	NVIC_IPR3 Interrupt priority control register 3.....	84
5.7.9	NVIC_IPR4 Interrupt priority control register 4.....	84
5.7.10	NVIC_IPR5 Interrupt priority control register 5.....	85
5.7.11	NVIC_IPR6 Interrupt priority control register 6.....	85
5.7.12	NVIC_IPR7 Interrupt priority control register 7.....	86
6	RAM memory.....	87
6.1	Overview.....	87
6.2	Main features.....	88
6.3	RAM memory operation.....	89
6.3.1	Read operation.....	89
6.3.2	Write operation.....	89
6.4	Parity check function.....	90
6.5	List of registers.....	91
6.6	Register description.....	92
6.6.1	RAM_IER Interrupt enable control register.....	92
6.6.2	RAM_ADDR Parity error address register.....	92



6.6.3	RAM_ISR Interrupt flag register.....	92
6.6.4	RAM_ICR Interrupt flag clear register.....	92
7	FLASH Memory.....	93
7.1	Overview.....	93
7.2	Main features.....	94
7.3	FLASH memory organization.....	95
7.4	FLASH memory read wait cycle configuration.....	96
7.5	FLASH memory operation.....	97
7.5.1	Page erase.....	98
7.5.2	Write operation.....	99
7.5.3	Read operation.....	100
7.6	FLASH memory protection.....	101
7.6.1	Erase and write protection.....	102
7.6.2	Erase and write PC page protection.....	103
7.6.3	Read protection.....	103
7.6.4	Security runtime library protection.....	104
7.7	FLASH memory programming.....	105
7.8	Precautions.....	106
7.9	List of registers.....	107
7.10	Register description.....	108
7.10.1	FLASH_CR1 Control register 1.....	108
7.10.2	FLASH_CR2 Control register 2.....	109
7.10.3	FLASH_PAGELOCK Erase lock register.....	110
7.10.4	FLASH_IER Interrupt enable register.....	111
7.10.5	FLASH_ISR Interrupt flag register.....	112
7.10.6	FLASH_ICR Interrupt flag clear register.....	113
7.10.7	FLASH_SDKCFR Security runtime library area register.....	113
8	Direct memory access (DMA).....	114
8.1	Overview.....	114
8.2	Main features.....	115
8.3	Functional block diagram.....	116
8.4	DMA transmit modes.....	118
8.4.1	Overview of transmit mode.....	118
8.4.2	Software trigger BLOCK transmit mode.....	120
8.4.3	Software trigger BULK transmit mode.....	121
8.4.4	Hardware trigger BLOCK transmit mode.....	122
8.4.5	Hardware trigger BULK transmit mode.....	123



8.5	Other configuration of DMA.....	124
8.5.1	Data width.....	124
8.5.2	Data block size.....	124
8.5.3	Number of data blocks.....	124
8.5.4	Channel priority.....	124
8.5.5	Channel auto repeat.....	124
8.6	DMA interrupt.....	125
8.7	List of registers.....	126
8.8	Register description.....	127
8.8.1	DMA_ISR Interrupt flag register.....	127
8.8.2	DMA_ICR Interrupt flag clear register.....	128
8.8.3	DMA_CSRy Channel y control and status register.....	129
8.8.4	DMA_TRIGy Channel y trigger source control register.....	131
8.8.5	DMA_CNTy Channel y transmit number register.....	133
8.8.6	DMA_SRCADDRy Channel y transmit source address register.....	133
8.8.7	DMA_DSTADDRy Channel y transmit destination address register.....	133
8.8.8	DMA_CCNTy Channel y current remaining transmit count register.....	134
8.8.9	DMA_CSRCADDRy Channel y current transmit source address register.....	134
8.8.10	DMA_CDSTADDRy Channel y current transmit destination address register.....	134
9	General-purpose input/output (GPIO).....	135
9.1	Overview.....	135
9.2	Main features.....	136
9.3	Functional description.....	137
9.3.1	Functional block diagram.....	137
9.3.2	Digital output.....	138
9.3.3	Digital input.....	139
9.3.4	Analogue function.....	140
9.3.5	Multiplexing function.....	141
9.3.6	Interrupt function.....	143
9.3.7	Other functions.....	144
9.4	Programming example.....	145
9.4.1	Digital output programming example.....	145
9.4.2	Digital input programming example.....	145
9.4.3	Analog functional programming example.....	145
9.4.4	Multiplexed function programming example.....	145
9.4.5	Interrupt function programming example.....	145
9.5	List of registers.....	146
9.6	Register description.....	147



9.6.1	GPIOx_DIR GPIO I/O direction register (x = A/B/C/F).....	147
9.6.2	GPIOx_OPENDRAIN GPIO output mode register (x = A/B/C/F).....	147
9.6.3	GPIOx_PDR GPIO pull-down resistor register (x = A/B/C/F).....	147
9.6.4	GPIOx_PUR GPIO pull-up resistor register (x = A/B/C/F).....	148
9.6.5	GPIOx_AFRH GPIO multiplexing function register high segment (x = A/B/C/F).....	148
9.6.6	GPIOx_AFRL GPIO multiplexing function register low segment (x = A/B/C/F).....	148
9.6.7	GPIOx_ANALOG GPIO analog digital configuration register (x = A/B/C/F).....	149
9.6.8	GPIOx_RISEIE GPIO rising edge interrupt enable register (x = A/B/C/F).....	149
9.6.9	GPIOx_FALLIE GPIO falling edge interrupt enable register (x = A/B/C/F).....	149
9.6.10	GPIOx_ISR GPIO interrupt flag register (x = A/B/C/F).....	150
9.6.11	GPIOx_ICR GPIO interrupt flag clear register (x = A/B/C/F).....	150
9.6.12	GPIOx_FILTER GPIO interrupt digital filter configuration register (x = A/B/C/F).....	150
9.6.13	GPIOx_IDR GPIO input data register (x = A/B/C/F).....	150
9.6.14	GPIOx_ODR GPIO output data register (x = A/B/C/F).....	151
9.6.15	GPIOx_BRR GPIO port bit set clear register (x = A/B/C/F).....	151
9.6.16	GPIOx_BSRR GPIO port bit-setting clear register (x = A/B/C/F).....	152
9.6.17	GPIOx_TOG GPIO port bit flip register (x = A/B/C/F).....	152
10	Cyclic redundancy check (CRC).....	153
10.1	Overview.....	153
10.2	Main features.....	154
10.3	Functional description.....	155
10.3.1	Algorithmic modes.....	156
10.3.2	Input data bit width.....	157
10.4	Programming examples.....	158
10.4.1	CRC16_CCITT algorithm mode.....	158
10.5	List of registers.....	159
10.6	Register description.....	160
10.6.1	CRC_CR Control register.....	160
10.6.2	CRC_DR Data register.....	160
10.6.3	CRC_RESULT Result register.....	160
11	Extended arithmetic unit (EAU).....	161
11.1	Overview.....	161
11.2	Main features.....	162
11.3	Operation definition.....	163
11.4	Status flag.....	164
11.5	Programming examples.....	165
11.5.1	Unsigned division programming example.....	165
11.5.2	Signed division programming example.....	165



11.5.3	Square root arithmetic programming example.....	166
11.6	List of registers.....	167
11.7	Register descriptions.....	168
11.7.1	EAU_CSR Control and status register.....	168
11.7.2	EAU_DIVIDEND Dividend/radicand register.....	169
11.7.3	EAU_DIVISOR Divisor register.....	169
11.7.4	EAU_QUOTIENT Quotient/square root register.....	169
11.7.5	EAU_REMAINDER Remainder register.....	170
12	Coordinate rotation digital computer algorithm (CORDIC).....	171
12.1	Overview.....	171
12.2	Main features.....	172
12.3	Operation definition.....	173
12.4	Status flag.....	175
12.5	List of registers.....	176
12.6	Register description.....	177
12.6.1	CORDIC_CSR Control and status register.....	177
12.6.2	CORDIC_X Data X register.....	178
12.6.3	CORDIC_Y Data Y register.....	178
12.6.4	CORDIC_Z Data Z register.....	178
13	Real time clock (RTC).....	179
13.1	Overview.....	179
13.2	Main features.....	180
13.3	Functional description.....	181
13.3.1	Functional block diagram.....	181
13.3.2	Clock and prescaler.....	182
13.3.3	Real time clock and calendar.....	183
13.3.4	RTC initialization settings.....	184
13.3.5	Register lock function.....	185
13.3.6	Register access operations.....	185
13.3.7	RTCOUT output.....	186
13.3.8	1Hz signal output.....	186
13.3.9	Clock error compensation.....	187
13.3.10	Alarm clock A and alarm clock B.....	188
13.3.11	Period interrupt function.....	189
13.3.12	Automatic wake-up function.....	189
13.3.13	Timestamp function.....	190
13.3.14	RTC interrupt.....	190
13.4	List of registers.....	191



13.5	Register description.....	192
13.5.1	RTC_KEY Key value register.....	192
13.5.2	RTC_CR0 Control register 0.....	192
13.5.3	RTC_CR1 Control register 1.....	193
13.5.4	RTC_CR2 Control register 2.....	194
13.5.5	RTC_PSC Clock prescaler register.....	195
13.5.6	RTC_DATE Date register.....	195
13.5.7	RTC_TIME Time register.....	196
13.5.8	RTC_SSCNT Subsecond count register.....	196
13.5.9	RTC_ALARM A Alarm clock A control register.....	197
13.5.10	RTC_ALARMB Alarm clock B control register.....	198
13.5.11	RTC_TAMPDATE Timestamp date register.....	198
13.5.12	RTC_TAMPTIME Timestamp time register.....	199
13.5.13	RTC_IER Interrupt enable register.....	199
13.5.14	RTC_ISR Interrupt flag register.....	200
13.5.15	RTC_ICR Interrupt flag clear register.....	201
13.5.16	RTC_AWTARR Wake-up timer reload value register.....	201
13.5.17	RTC_AWTCNT Wake-up timer count value register.....	202
13.5.18	RTC_COMPCFR1 Wake-up clock error compensation register 1.....	202
14	Basic timer (BTIM).....	203
14.1	Overview.....	203
14.2	Main features.....	204
14.3	Functional description.....	205
14.3.1	Functional block diagram.....	205
14.3.1.1	Counting unit.....	205
14.3.1.2	Update event (UEV).....	207
14.3.1.3	Trigger input channel.....	208
14.3.1.4	Trigger output channel.....	208
14.3.1.5	Reset input channel.....	208
14.3.1.6	Flip output unit.....	209
14.3.2	Operating modes.....	210
14.3.2.1	Internal count mode.....	211
14.3.2.2	External count mode.....	212
14.3.2.3	Trigger start mode.....	213
14.3.2.4	Gated count mode.....	214
14.3.3	UIF bit remap.....	215
14.4	BTIM interrupt.....	216
14.5	Trigger ADC.....	217
14.6	Direct memory access (DMA).....	218



14.7	Debugging support.....	219
14.8	Programming examples.....	220
14.8.1	Internal count mode programming example.....	220
14.8.2	External count mode programming example.....	220
14.8.3	Trigger start mode programming example.....	221
14.8.4	Gated count mode programming example.....	221
14.8.5	Counter reset programming example.....	222
14.9	List of registers.....	223
14.10	Register description.....	224
14.10.1	BTIMx_CR1 Control register 1.....	224
14.10.2	BTIMx_CR2 Control register 2.....	225
14.10.3	BTIMx_SMCR Slave mode control register.....	225
14.10.4	BTIMx_DIER DMA/Interrupt enable register.....	227
14.10.5	BTIMx_ISR Interrupt flag register.....	227
14.10.6	BTIMx_ICR Interrupt flag clear register.....	228
14.10.7	BTIMx_EGR Event generation register.....	228
14.10.8	BTIMx_CNT Count register.....	229
14.10.9	BTIMx_PSC Prescaler register.....	229
14.10.10	BTIMx_ARR Auto reload register.....	229
15	Low power timer (LPTIM).....	230
15.1	Overview.....	230
15.2	Main features.....	231
15.3	Functional description.....	232
15.3.1	Functional block diagram.....	232
15.3.1.1	Clock source.....	233
15.3.1.2	Prescaler.....	233
15.3.1.3	Filter unit.....	234
15.3.2	Start method.....	235
15.3.3	Trigger source.....	236
15.3.4	Operating mode.....	237
15.3.5	Timeout function.....	239
15.3.6	Waveform output control.....	239
15.3.7	Register update.....	240
15.3.8	Count function.....	240
15.3.9	Encoding mode.....	241
15.4	Debug support.....	243
15.5	Programming examples.....	244
15.5.1	External pulse counting routine.....	244



15.6	List of registers.....	245
15.7	Register description.....	246
15.7.1	LPTIM_CFGR Configuration register.....	246
15.7.2	LPTIM_CR0 Control register 0.....	249
15.7.3	LPTIM_ARR Auto-reload register.....	249
15.7.4	LPTIM_CNT Counter register.....	250
15.7.5	LPTIM_CMP Compare register.....	250
15.7.6	LPTIM_IER Interrupt enable register.....	251
15.7.7	LPTIM_ISR Interrupt and status register.....	252
15.7.8	LPTIM_ICR Interrupt clear register.....	253
15.7.9	LPTIM_CR1 Control register 1.....	254
16	General-purpose timer (GTIM).....	255
16.1	Overview.....	255
16.2	Main features.....	256
16.3	Functional description.....	257
16.3.1	Functional block diagram.....	257
16.3.1.1	Clock source.....	258
16.3.1.2	Prescaler.....	258
16.3.1.3	Counter and count mode.....	259
16.3.1.4	Reload register.....	266
16.3.1.5	Update event UEV.....	267
16.3.1.6	Single pulse mode.....	268
16.3.1.7	External trigger input channel.....	269
16.3.1.8	Input capture channel.....	270
16.3.1.9	Output compare channel.....	271
16.3.2	Operating modes.....	272
16.3.2.1	Internal clock mode.....	272
16.3.2.2	External clock mode.....	273
16.3.2.3	Reset mode.....	275
16.3.2.4	Gated mode.....	276
16.3.2.5	Trigger mode.....	277
16.3.2.6	Combined reset + trigger mode.....	277
16.3.2.7	Combined gated + reset mode.....	278
16.3.2.8	Quadrature encoder mode.....	279
16.3.2.9	Clock + direction encoder mode.....	282
16.3.2.10	Directional clock encoder mode.....	282
16.3.3	Input capture function.....	284
16.3.3.1	PWM input mode.....	285
16.3.3.2	Input capture source.....	286



16.3.4	Output compare function.....	287
16.3.4.1	Match output.....	289
16.3.4.2	Forced output	290
16.3.4.3	PWM output.....	290
16.3.4.4	Asymmetric PWM output.....	293
16.3.4.5	Combined PWM output.....	295
16.3.4.6	Retriggerable single pulse mode.....	297
16.3.4.7	Count direction output.....	297
16.3.4.8	Reference signal clear function.....	298
16.3.5	UIF bit remap.....	299
16.3.6	Timer cascade ITR.....	300
16.3.7	On-chip peripheral interconnect ETR.....	302
16.4	GTIM interrupt.....	303
16.5	Trigger ADC.....	304
16.6	Direct memory access (DMA).....	305
16.7	Debugging support.....	306
16.8	Programming examples.....	307
16.8.1	External clock mode 1 programming example.....	307
16.8.2	External clock mode 2 programming example.....	307
16.8.3	Reset mode programming example.....	308
16.8.4	Gated mode programming example.....	308
16.8.5	Trigger mode programming example.....	309
16.8.6	Quadrature encoder mode programming example.....	309
16.8.7	Input capture programming example.....	310
16.8.7.1	Basic input capture mode.....	310
16.8.7.2	PWM input mode.....	311
16.8.8	Output compare programming example.....	312
16.9	List of registers.....	313
16.10	Register description.....	314
16.10.1	GTIMx_CR1 Controls register 1.....	314
16.10.2	GTIMx_CR2 Control register 2.....	315
16.10.3	GTIMx_SMCR Slave mode control register.....	318
16.10.4	GTIMx_DIER DMA/Interrupt enable register.....	322
16.10.5	GTIMx_ISR Interrupt flag register.....	324
16.10.6	GTIMx_ICR Interrupt flag clear register.....	326
16.10.7	GTIMx_EGR Event generation register.....	327
16.10.8	GTIMx_CCMR1CAP Capture mode register 1.....	328
16.10.9	GTIMx_CCMR1CMP Compare mode register 1.....	330
16.10.10	GTIMx_CCMR2CAP Capture mode register 2.....	333



16.10.11	GTIMx_CCMR2CMP Compare mode register 2.....	334
16.10.12	GTIMx_CCER Capture/compare enable register.....	335
16.10.13	GTIMx_CNT Count register.....	337
16.10.14	GTIMx_PSC Prescaler register.....	337
16.10.15	GTIMx_ARR Auto-reload register.....	338
16.10.16	GTIMx_CCR1 Capture/compare register 1.....	338
16.10.17	GTIMx_CCR2 Capture/compare register 2.....	339
16.10.18	GTIMx_CCR3 Capture/compare register 3.....	339
16.10.19	GTIMx_CCR4 Capture/compare register 4.....	340
16.10.20	GTIMx_ECR Encoder control register.....	341
16.10.21	GTIMx_TISEL TI input selection register.....	342
16.10.22	GTIMx_AF1 Multiplexing function option register 1.....	344
16.10.23	GTIMx_AF2 Multiplexing function option register 2.....	345
17	Advanced-control timer (ATIM).....	346
17.1	Overview.....	346
17.2	Main features.....	347
17.3	Functional description.....	348
17.3.1	Functional block diagram.....	348
17.3.1.1	Clock source.....	349
17.3.1.2	Prescaler.....	349
17.3.1.3	Counter and count mode.....	350
17.3.1.4	Reload register.....	357
17.3.1.5	Repeat counter.....	358
17.3.1.6	Update event UEV	359
17.3.1.7	Single pulse mode.....	360
17.3.1.8	External trigger input channel.....	361
17.3.1.9	Input capture channel.....	362
17.3.1.10	Output compare channel.....	363
17.3.2	Operating modes.....	364
17.3.2.1	Internal clock mode.....	364
17.3.2.2	External clock mode.....	365
17.3.2.3	Reset mode.....	367
17.3.2.4	Gated mode.....	368
17.3.2.5	Trigger mode.....	369
17.3.2.6	Combined reset + trigger mode.....	369
17.3.2.7	Combined gated + reset mode.....	370
17.3.2.8	Quadrature encoder mode.....	370
17.3.2.9	Clock + direction encoder mode.....	373
17.3.2.10	Directional clock encoder mode.....	373
17.3.3	Input capture function.....	375



17.3.3.1	PWM input mode.....	376
17.3.3.2	Input capture source.....	377
17.3.4	Output compare function.....	378
17.3.4.1	Match output.....	379
17.3.4.2	Forced output	379
17.3.4.3	PWM output.....	380
17.3.4.4	Asymmetric PWM output.....	382
17.3.4.5	Combined PWM output.....	384
17.3.4.6	Phase shift PWM mode.....	386
17.3.4.7	Complementary output and dead-time insertion.....	387
17.3.4.8	Brake function.....	389
17.3.4.9	Retriggerable single pulse mode.....	393
17.3.4.10	Count direction output.....	393
17.3.4.11	Reference signal clear function.....	394
17.3.5	UIF bit remap.....	395
17.3.6	Timer cascade ITR.....	395
17.3.7	On-chip peripheral interconnect ETR.....	396
17.4	ATIM interrupt.....	397
17.5	Trigger ADC.....	398
17.6	Direct memory access (DMA).....	399
17.7	Debugging support.....	400
17.8	Programming examples.....	401
17.8.1	External clock mode 1 programming example.....	401
17.8.2	External clock mode 2 programming example.....	401
17.8.3	Reset mode programming example.....	402
17.8.4	Gated mode programming example.....	402
17.8.5	Trigger mode programming example.....	403
17.8.6	Quadrature encoder mode programming example.....	403
17.8.7	Input capture programming example.....	404
17.8.7.1	Basic input capture mode.....	404
17.8.7.2	PWM input mode.....	405
17.8.8	Output compare programming example.....	406
17.9	List of registers.....	407
17.10	Register description.....	409
17.10.1	ATIM_CR1 Control register 1.....	409
17.10.2	ATIM_CR2 Control register 2.....	411
17.10.3	ATIM_SMCR Slave mode control register.....	414
17.10.4	ATIM_DIER DMA/Interrupt enable register.....	418
17.10.5	ATIM_ISR Interrupt flag register.....	420



17.10.6	ATIM_ICR Interrupt flag clear register.....	423
17.10.7	ATIM_EGR Event generation register.....	425
17.10.8	ATIM_CCMR1CAP Capture mode register 1.....	427
17.10.9	ATIM_CCMR1CMP Compare mode register 1.....	429
17.10.10	ATIM_CCMR2CAP Capture mode register 2.....	432
17.10.11	ATIM_CCMR2CMP Compare mode register 2.....	433
17.10.12	ATIM_CCMR3CAP Capture mode register 3.....	434
17.10.13	ATIM_CCMR3CMP Compare mode register 3.....	435
17.10.14	ATIM_CCER Capture/compare enable register.....	436
17.10.15	ATIM_CNT Count register.....	439
17.10.16	ATIM_PSC Prescaler register.....	439
17.10.17	ATIM_ARR Auto reload register.....	439
17.10.18	ATIM_RCR Repeat count register.....	440
17.10.19	ATIM_CCR1 Capture/compare register 1.....	440
17.10.20	ATIM_CCR2 Capture/compare register 2.....	441
17.10.21	ATIM_CCR3 Capture/compare register 3.....	441
17.10.22	ATIM_CCR4 Capture/compare register 4.....	441
17.10.23	ATIM_CCR5 Capture/compare register 5.....	441
17.10.24	ATIM_CCR6 Capture/compare register 6.....	442
17.10.25	ATIM_BDTR Brake and deadband register.....	443
17.10.26	ATIM_DTR2 Dead time register 2.....	446
17.10.27	ATIM_ECR Encoder control register.....	447
17.10.28	ATIM_TISEL1 TI input selection register 1.....	448
17.10.29	ATIM_TISEL2 TI input selection register 2.....	449
17.10.30	ATIM_AF1 Multiplexing function option register 1.....	450
17.10.31	ATIM_AF2 Multiplexing function option register 2.....	452
18	Hall sensor dedicated timer (HALLTIM).....	454
18.1	Overview.....	454
18.2	Main features.....	455
18.3	Functional description.....	456
18.3.1	Functional block diagram.....	456
18.3.1.1	Operation clock.....	457
18.3.1.2	Filter unit.....	457
18.3.2	Operation description.....	458
18.3.2.1	Overflow signal.....	458
18.3.2.2	PWM waveform output.....	458
18.3.2.3	Input capture.....	458
18.4	HALLTIM interrupt.....	459
18.5	Trigger output signal TRGO.....	460



18.6	Direct memory access (DMA).....	461
18.7	List of registers.....	462
18.8	Register description.....	463
18.8.1	HALLTIM_CR Control register.....	463
18.8.2	HALLTIM_DIER DMA/Interrupt enable control register.....	464
18.8.3	HALLTIM_CNT Count value register.....	464
18.8.4	HALLTIM_ARR Auto reload register.....	465
18.8.5	HALLTIM_WIDTH HALL pulse width register.....	465
18.8.6	HALLTIM_CCR Compare register.....	465
18.8.7	HALLTIM_ISR Interrupt flag register.....	465
18.8.8	HALLTIM_ICR Interrupt flag clear register.....	466
18.8.9	HALLTIM_STATE Status register.....	466
19	Independent watchdog timer (IWDG).....	467
19.1	Overview.....	467
19.2	Main features.....	468
19.3	Functional description.....	469
19.3.1	Functional block diagram.....	469
19.3.2	Operating modes.....	469
19.3.3	Window options.....	470
19.3.4	Register lock function.....	470
19.3.5	Start refresh and stop.....	470
19.3.6	Status register.....	471
19.3.7	Timing duration setting.....	472
19.4	Programming examples.....	473
19.4.1	Configure IWDG as independent watchdog.....	473
19.4.2	Configure IWDG as window watchdog.....	473
19.4.3	Refresh IWDG (feed dog operation).....	473
19.5	List of registers.....	474
19.6	Register description.....	475
19.6.1	IWDG_KR Key-value register.....	475
19.6.2	IWDG_CR Control register.....	475
19.6.3	IWDG_ARR Reload register.....	476
19.6.4	IWDG_CNT Count value register.....	476
19.6.5	IWDG_WINR Window register.....	476
19.6.6	IWDG_SR Status register.....	477
20	Window watchdog timer (WWDG).....	478
20.1	Overview.....	478
20.2	Main features.....	479



20.3	Functional description.....	480
20.3.1	Functional block diagram.....	480
20.3.2	Operating mode.....	480
20.3.3	Refresh counter.....	481
20.3.4	Dog-feeding time.....	481
20.3.5	Reset and interrupt.....	482
20.4	Programming examples.....	483
20.5	List of registers.....	484
20.6	Register description.....	485
20.6.1	WWDT_CR0 Control register 0.....	485
20.6.2	WWDT_CR1 Control register 1.....	485
20.6.3	WWDT_SR Status register.....	486
21	Universal asynchronous receiver/transmitter (UART).....	487
21.1	Overview.....	487
21.2	Main features.....	488
21.3	Functional description.....	489
21.3.1	Functional block diagram.....	489
21.3.2	Synchronous mode.....	491
21.3.2.1	Baud rate setting.....	491
21.3.2.2	Data transmission and reception.....	492
21.3.3	Asynchronous mode.....	493
21.3.3.1	Data frame format.....	493
21.3.3.2	Fractional baud rate generator.....	494
21.3.3.3	Transmit control.....	498
21.3.3.4	Receive control.....	499
21.3.3.5	Single-wire half-duplex mode.....	501
21.3.3.6	Multi-machine communication.....	502
21.3.3.7	Hardware flow control.....	504
21.3.3.8	RS485 driver enable.....	506
21.3.3.9	LoopBack mode.....	507
21.3.3.10	Input signal source/level shift.....	508
21.3.4	LIN Mode.....	509
21.3.4.1	LIN transmit.....	509
21.3.4.2	LIN receive.....	510
21.3.5	Timer operating mode.....	511
21.3.5.1	Waiting for timeout detection mode.....	511
21.3.5.2	Receive idle detection mode.....	511
21.3.5.3	Auto baud rate detection mode.....	512
21.3.5.4	General purpose timer mode.....	512



21.4	Low-power mode.....	513
21.5	UART interrupt.....	514
21.6	Direct memory access (DMA).....	515
21.7	Programming examples.....	516
21.7.1	Asynchronous full-duplex programming examples.....	516
21.7.1.1	Transmit data in query mode.....	516
21.7.1.2	Receive data in query mode.....	517
21.7.1.3	Transmit data in interrupt mode.....	518
21.7.1.4	Receive data in interrupt mode.....	519
21.7.1.5	DMA transmits data.....	520
21.7.1.6	DMA receives data.....	521
21.7.2	Synchronous half-duplex programming examples.....	522
21.7.2.1	Transmit data in query mode.....	522
21.7.2.2	Receive data in query mode.....	522
21.7.3	Single-wire half-duplex programming examples.....	523
21.7.3.1	Transmit data in query mode.....	523
21.7.3.2	Receive data in query mode.....	524
21.8	List of registers.....	525
21.9	Register description.....	526
21.9.1	UARTx_CR1 Control register 1.....	526
21.9.2	UARTx_CR2 Control register 2.....	528
21.9.3	UARTx_CR3 Control register 3.....	530
21.9.4	UARTx_BRR1 Baud rate counter integer part register.....	530
21.9.5	UARTx_BRRF Baud rate counter fractional register.....	531
21.9.6	UARTx_TIMARR Timer reload register.....	531
21.9.7	UARTx_TIMCNT Timer count register.....	531
21.9.8	UARTx_TDR Transmit data register.....	532
21.9.9	UARTx_RDR Receive data register.....	532
21.9.10	UARTx_IER Interrupt enable register.....	532
21.9.11	UARTx_ISR Interrupt flag register.....	534
21.9.12	UARTx_ICR Interrupt flag clear register.....	536
21.9.13	UARTx_ADDR Slave address register.....	537
21.9.14	UARTx_MASK Slave address mask register.....	537
21.9.15	UARTx_RXMATCH Receive data match register.....	537
22	Serial peripheral interface (SPI).....	538
22.1	Overview.....	538
22.2	Main features.....	539
22.3	Functional description.....	540
22.3.1	Functional block diagram.....	540



22.3.2	Communication timing and data format.....	542
22.3.3	Slave select pin CS.....	544
22.3.4	Full-duplex mode.....	545
22.3.5	Single-wire half-duplex mode.....	548
22.3.6	Simplex mode.....	550
22.3.6.1	Master single transmitting/slave single receiving.....	551
22.3.6.2	Master single receiving/slave single transmitting.....	552
22.3.7	Multi-machine communication.....	553
22.3.8	Status flag.....	555
22.3.9	Error flag.....	556
22.4	SPI interrupts.....	557
22.5	Programming examples.....	558
22.5.1	Full duplex/master mode.....	558
22.5.1.1	Transmit and receive in query mode.....	558
22.5.1.2	Transmit and receive in interrupt mode.....	559
22.5.2	Single-wire half-duplex/master mode.....	560
22.5.2.1	Transmit in query mode.....	560
22.5.2.2	Receive in query mode.....	561
22.5.3	Simplex mode/master mode.....	562
22.5.3.1	Transmit in query mode.....	562
22.5.3.2	Receive in query mode.....	563
22.6	List of registers.....	564
22.7	Register description.....	565
22.7.1	SPI_CR1 Control register 1.....	565
22.7.2	SPI_CR2 Control register 2.....	567
22.7.3	SPI_CR3 Control register 3.....	567
22.7.4	SPI_SSI Slave select register.....	568
22.7.5	SPI_IER Interrupt enable register.....	569
22.7.6	SPI_ISR Interrupt flag register.....	570
22.7.7	SPI_ICR Interrupt flag clear register.....	571
22.7.8	SPI_DR Data register.....	571
23	I2C interface.....	572
23.1	Overview.....	572
23.2	Main features.....	573
23.2.1	Master.....	573
23.2.2	Slave.....	573
23.3	Protocol description.....	574
23.3.1	Protocol frame format.....	574
23.3.2	Transmission response.....	576



23.3.3	Conflict detection and arbitration.....	577
23.4	Functional description.....	579
23.4.1	Functional block diagram.....	579
23.4.1.1	Clock and reset.....	580
23.4.1.2	Input filter.....	580
23.4.1.3	Address comparator.....	581
23.4.1.4	Command/transmit FIFO.....	581
23.4.1.5	Receive FIFO and data match.....	582
23.4.1.6	Slave transmits/receives data.....	582
23.4.2	Arbitration and synchronization logic.....	583
23.4.2.1	SCL synchronization.....	583
23.4.2.2	SDA arbitration.....	583
23.4.3	Timing configuration.....	584
23.4.3.1	Master timing.....	584
23.4.3.2	Slave timing.....	586
23.4.4	Operating modes.....	587
23.4.4.1	Master transmit mode.....	588
23.4.4.2	Master receive mode.....	589
23.4.4.3	Slave receive mode.....	590
23.4.4.4	Slave transmit mode.....	591
23.4.4.5	Broadcast receive mode.....	592
23.4.5	Error flag.....	593
23.5	I2C interrupt.....	594
23.6	Programming examples.....	595
23.6.1	Master transmits example.....	595
23.6.2	Master receives example.....	596
23.6.3	Slave receives example.....	597
23.6.4	Slave transmits example.....	597
23.7	List of registers.....	598
23.8	Register description.....	599
23.8.1	I2Cx_INSEL Bus input signal selection register.....	599
23.8.2	I2Cx_MCR0 Master control register 0.....	600
23.8.3	I2Cx_MCR1 Master control register 1.....	601
23.8.4	I2Cx_MCR2 Master control register 2.....	602
23.8.5	I2Cx_MCR3 Master control register 3.....	603
23.8.6	I2Cx_MCR4 Master control register 4.....	603
23.8.7	I2Cx_MCCR Master timing configuration register.....	604
23.8.8	I2Cx_MMATCH Master data match register.....	604
23.8.9	I2Cx_MFIFOCR Master FIFO control register.....	605
23.8.10	I2Cx_MFIFOSR Master FIFO status register.....	605



23.8.11	I2Cx_MTDR Master transmit data register.....	605
23.8.12	I2Cx_MRDR Master receive data register.....	606
23.8.13	I2Cx_MDER Master DMA enable register.....	606
23.8.14	I2Cx_MIER Master interrupt enable register.....	607
23.8.15	I2Cx_MISR Master interrupt flag register.....	608
23.8.16	I2Cx_MICR Master interrupt flag clear register.....	609
23.8.17	I2Cx_SCR0 Slave control register 0.....	610
23.8.18	I2Cx_SCR1 Slave control register 1.....	611
23.8.19	I2Cx_SCR2 Slave control register 2.....	612
23.8.20	I2Cx_SCR3 Slave control register 3.....	613
23.8.21	I2Cx_SAMR Slave address match register.....	613
23.8.22	I2Cx_STAR Slave transmit response register.....	614
23.8.23	I2Cx_STDR Slave transmit data register.....	614
23.8.24	I2Cx_SRDR Slave receive data register.....	614
23.8.25	I2Cx_SASR Slave address status register.....	615
23.8.26	I2Cx_SDER Slave DMA enable register.....	615
23.8.27	I2Cx_SIER Slave interrupt enable register.....	615
23.8.28	I2Cx_SISR Slave interrupt flag register.....	617
23.8.29	I2Cx_SICR Slave interrupt flag clear register.....	618
24	Infrared modulation transmitter (IR).....	619
24.1	Overview.....	619
24.2	Main features.....	620
24.3	Functional description.....	621
24.3.1	Infrared modulation mode.....	622
24.3.2	Infrared modulation initialization configuration.....	624
24.3.3	Infrared reception.....	624
24.4	IRMOD_CR Infrared modulation control register.....	625
25	Analog-to-digital converter (ADC).....	626
25.1	Overview.....	626
25.2	Main features.....	627
25.3	Functional block diagram.....	628
25.4	Conversion timing, conversion speed, conversion accuracy, and conversion results.....	629
25.4.1	Conversion timing.....	629
25.4.2	Conversion speed.....	631
25.4.3	Conversion results.....	631
25.5	Operating modes.....	632
25.5.1	Sequence continuous conversion mode (CONT = 1).....	633
25.5.2	Sequence scan conversion mode (CONT = 0).....	635



25.6	External trigger source.....	637
25.7	Analog watchdog.....	638
25.8	BGR module.....	639
25.9	Temperature sensor.....	640
25.10	ADC interrupts.....	641
25.11	List of registers.....	642
25.12	Register description.....	643
25.12.1	ADCx_CR Control register.....	643
25.12.2	ADCx_START Startup register.....	644
25.12.3	ADCx_SAMPLE Sample cycle configuration register.....	644
25.12.4	ADCx_SQRCFR Sequence conversion channel configuration register.....	645
25.12.5	ADCx_AWDTR Analogue watchdog threshold register.....	645
25.12.6	ADCx_AWDCR Analogue watchdog configuration register.....	646
25.12.7	ADCx_TRIGGER External trigger register.....	648
25.12.8	ADCx_IER Interrupt enable register.....	651
25.12.9	ADCx_ISR Interrupt flag register.....	652
25.12.10	ADCx_ICR Interrupt flag clear register.....	652
25.12.11	ADCx_RESULT0 Sequence conversion result register 0.....	653
25.12.12	ADCx_RESULT1 Sequence conversion result register 1.....	653
25.12.13	ADCx_RESULT2 Sequence conversion result register 2.....	653
25.12.14	ADCx_RESULT3 Sequence conversion result register 3.....	653
25.12.15	ADCx_RESULT4 Sequence conversion result register 4.....	653
25.12.16	ADCx_RESULT5 Sequence conversion result register 5.....	654
25.12.17	ADCx_RESULT6 Sequence conversion result register 6.....	654
25.12.18	ADCx_RESULT7 Sequence conversion result register 7.....	654
25.12.19	BGR_CR Control register.....	654
26	Digital to analog converter (DAC).....	655
26.1	Overview.....	655
26.2	Main features.....	656
26.3	Functional block diagram.....	657
26.4	Digital signal conversion source.....	658
26.5	Data format.....	659
26.5.1	Single-channel mode.....	659
26.5.2	Dual-channel mode.....	660
26.6	Trigger source.....	661
26.7	Direct memory access (DMA).....	662
26.8	DAC interrupt.....	663



26.9	List of registers.....	664
26.10	Register description.....	665
26.10.1	DAC_CR0 Control register 0.....	665
26.10.2	DAC_SWTRGR Software trigger register.....	669
26.10.3	DAC_DHR12R1 Channel 1 12-bit right-aligned data hold register.....	669
26.10.4	DAC_DHR12L1 Channel 1 12-bit left-aligned data hold register.....	669
26.10.5	DAC_DHR8R1 Channel 1 8-bit right-aligned data hold register.....	670
26.10.6	DAC_DHR12R2 Channel 2 12-bit right-aligned data hold register.....	670
26.10.7	DAC_DHR12L2 Channel 2 12-bit left-aligned data hold register.....	670
26.10.8	DAC_DHR8R2 Channel 2 8-bit right-aligned data hold register.....	670
26.10.9	DAC_DHR12RD Dual-channel 12-bit right-aligned data hold register.....	671
26.10.10	DAC_DHR12LD Dual-channel 12-bit left-aligned data hold register.....	671
26.10.11	DAC_DHR8RD Dual-channel 8-bit right-aligned data hold register.....	671
26.10.12	DAC_DOR1 Channel 1 data output register.....	672
26.10.13	DAC_DOR2 Channel 2 data output register.....	672
26.10.14	DAC_CR1 Control register 1.....	673
27	Analog voltage comparator (VC).....	674
27.1	Overview.....	674
27.2	Main features.....	675
27.3	Functional description.....	676
27.3.1	Functional block diagram.....	676
27.3.2	I/O pins.....	678
27.3.3	Latency/Response time.....	678
27.3.4	Polarity selection.....	678
27.3.5	Digital filter.....	679
27.3.6	Hysteresis function.....	680
27.3.7	Window compare function.....	680
27.3.8	BLANK window function.....	681
27.4	VC interrupts.....	682
27.5	Programming examples.....	683
27.6	List of registers.....	684
27.7	Register description.....	685
27.7.1	VC12_REF VC1/VC2 resistor divider control register.....	685
27.7.2	VC34_REF VC3/VC4 resistor divider control register.....	685
27.7.3	VCx_CR0 Control register 0.....	686
27.7.4	VCx_CR1 Control register 1.....	687
27.7.5	VCx_CR2 Control register 2.....	688
27.7.6	VCx_SR Status register	691



28	Low voltage detector (LVD).....	692
28.1	Overview.....	692
28.2	Main features.....	693
28.3	Functional description.....	694
28.3.1	Functional block diagram.....	694
28.3.2	Hysteresis function.....	695
28.3.3	Digital filter.....	696
28.4	LVD interrupts.....	697
28.5	Programming examples.....	698
28.5.1	Brown-out reset programming example.....	698
28.5.2	Interrupt programming example.....	698
28.6	List of registers.....	699
28.7	Register description.....	700
28.7.1	LVD_CR0 Control register 0.....	700
28.7.2	LVD_CR1 Control register 1.....	701
28.7.3	LVD_SR Status register.....	702
29	Operational amplifier (OPA).....	703
29.1	Overview.....	703
29.2	Main features.....	704
29.3	Functional description.....	705
29.3.1	Functional block diagram.....	705
29.3.2	Input/output pins.....	705
29.3.3	Calibration function.....	706
29.4	Programming examples.....	707
29.5	List of registers.....	708
29.6	Register description.....	709
29.6.1	OPAx_CR Control register.....	709
29.6.2	OPAx_CAL Calibration register.....	710
30	Debug interface (DBG).....	713
30.1	Overview.....	713
30.2	Serial wire debug port SWD.....	714
30.3	Debug communication protocol.....	716
30.3.1	Transmission protocol format.....	716
30.3.2	SW-DP state machine.....	717
30.3.3	Read and write access for SW-DP and SW-AP.....	718
30.3.4	SW-DP register.....	719
30.3.5	SW-AP register.....	721



30.4	Kernel debugging.....	722
30.5	Breakpoint unit BPU.....	723
30.6	Data observation points and tracking DWT.....	724
30.7	Debug components DBG.....	725
30.8	Cautions.....	726
31	Digital signature.....	727
31.1	Overview.....	727
31.2	Product unique identification (UID) register (80bit).....	728
31.3	Product model register.....	729
31.4	FLASH capacity register.....	730
31.5	SRAM capacity register.....	731
31.6	Pin number register.....	732
32	Revision history.....	733



1 Documentation conventions

1.1 Typographical conventions

The typographical conventions used in this document are:

<i>Italic</i>	Indicates a special term or citation, or prompts a note.
Bold	Indicates a component or signal name, such as for title descriptions without chapter numbers, titles for diagrams, etc.
English uppercase	Used for terms that have a specific technical meaning and are included in the glossary, such as register or bit field names.
Color text	Indicates a link. Including: • URL external link, such as https://www.whxy.com. • Cross-references, internal chapters of this paper refer to each other. • Internal links, figures, tables or appendices, glossary entries.



1.2 Register protocol

Symbol example	Description
0x53CA	A numeric string prefixed with '0x' represents a hexadecimal number, and the value is composed of 0~9 or A~F.
REGNAME[n]	A specific bit of REGNAME, REGNAME can be a register or register field, for example, CR1[2] refers to bit 2 of the CR1 register (field).
REGNAME[m:n]	A specific bit of REGNAME, REGNAME can be a register or a register field, for example, CR1[7:5] refers to the 7th to 5th bits of the CR1 register (field).
RW	Read and write, software can read and write these bit fields.
RO	Read-only, software can only read these bit fields.
WO	Write-only, software can only write to this bit field, and invalid data will be returned when reading this bit field.
RW0	Software can read and write this bit field, only 0 can be written to this bit field, and writing 1 has no effect on this bit field.
RW1	Software can read and write this bit field, only 1 can be written to this bit field, and writing 0 has no effect on this bit field.
R0W1	When the software reads this bit field as 0, only 1 can be written to this bit field, and writing 0 has no effect on this bit field.
R1W0	When the software reads this bit field as 1, only 0 can be written to this bit field, and writing 1 has no effect on this bit field.
RFU	Reserved bit field, please keep the default value.
Word	The data length of a word is 32bit.
Half Word	The data length of a half word is 16bit.
Byte	The data length of one byte is 8bit.



2 System and memory overview

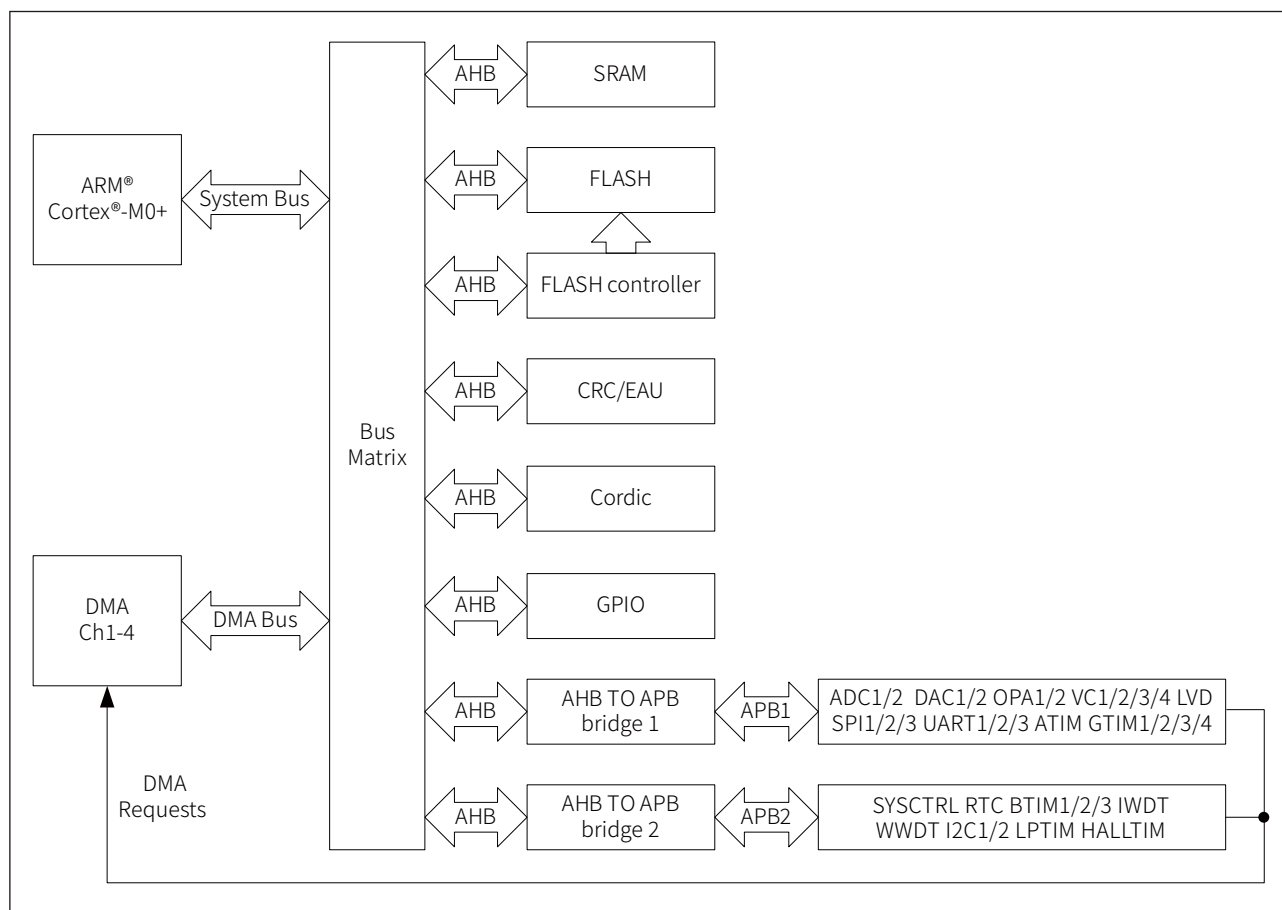
2.1 System architecture

The consists of CW32L012 microcontroller system contains:

- 2 master device:
 - ARM® Cortex®-M0+ core
 - Direct memory access (DMA)
- Multiple slave device:
 - On-chip SRAM
 - On-chip FLASH
 - FLASH controller
 - Redundancy calculation unit (CRC)
 - Extended arithmetic unit (EAU)
 - CORDIC arithmetic unit
 - GPIO port
 - AHB to APB1 conversion bridge and all devices on the APB1 bus
 - AHB to APB2 conversion bridge and all devices on the APB2 bus

The master and slave devices are connected through AHB bus matrix, and the system architecture is shown as follows:

Figure 2-1 System architecture



- System bus
Realize the connection of peripheral bus and bus matrix of M0+ microprocessor.
- DMA bus
AHB bus and bus matrix connection for DMA.
- Bus matrix:
Manages the access arbitration between M0+ microprocessor and DMA to FLASH, SRAM and all peripherals. The arbitration control adopts the round-robin scheduling algorithm to balance the load and ensure the bus utilization efficiency.
- AHB TO APB Bridge 1/2:
Provides a fully synchronized connection from the AHB bus to the APB1/APB2 bus, that is, the bridge between the AHB and APB buses.



2.2 Memory organization

2.2.1 Overview

The CW32L012 core is a 32-bit ARM® Cortex®-M0+ microprocessor with a maximum addressing space of 4GB. The built-in program memory, data memory, peripherals and port registers of the chip are uniformly addressed in the same 4GB linear address space.

Bytes in memory are organized in little-endian format. The lowest byte data of a word storage space is the least significant bit of the word, and the highest byte data is the most significant bit. E.g:

Store 0x1122 3344 in the memory space at address 0x2000 0000, the actual result is:

0x20000000 bytes store 0x44,

0x20000001 bytes store 0x33,

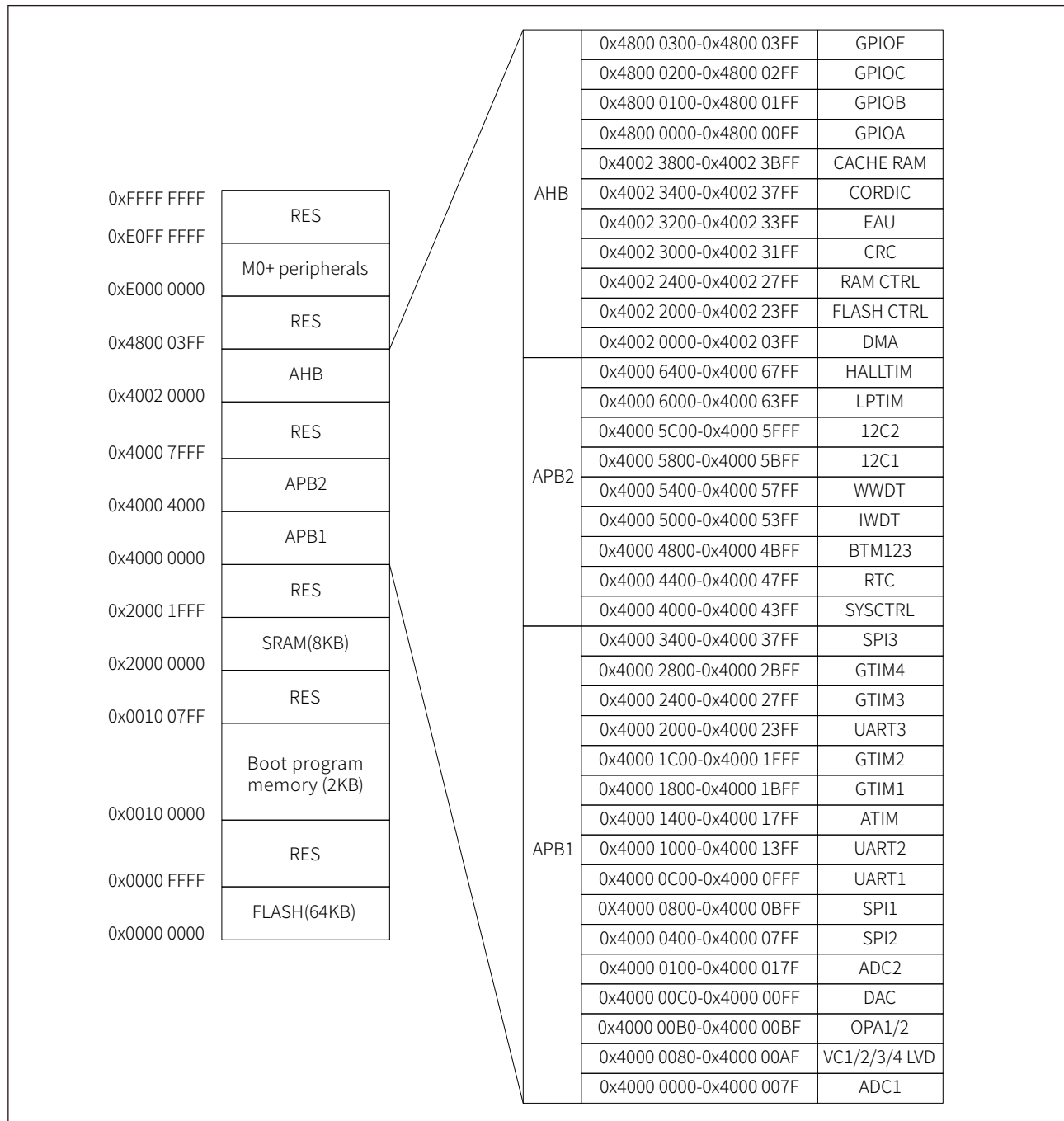
0x20000002 bytes store 0x22,

0x20000003 bytes store 0x11.

The system address allocation is shown in the figure below, RES is the reserved area:



Figure 2-2 System address assignment



2.2.2 Memory map and register boundary addresses

The detailed starting address space allocation of on-chip memory and peripherals is shown in the following table:

Table 2-1 CW32L012 memory and peripheral register boundary addresses

Device or bus	Boundary address	Size	Peripheral
Main FLASH memory	0x0000 0000 - 0x0000 FFFF	64KB	Main FLASH
Boot program memory	0x0010 0000 - 0x0010 09FF	2KB	BootLoader
OTP memory	0x0010 0760 - 0x0010 0775	22B	OTP
SRAM memory	0x2000 0000 - 0x2000 1FFF	8KB	SRAM
APB1 peripheral	0x4000 0000 - 0x4000 03FF	1KB	ADC12/VC1234/LVD/OPA12/DAC
	0x4000 0400 - 0x4000 07FF	1KB	SPI2
	0x4000 0800 - 0x4000 0BFF	1KB	SPI1
	0x4000 0C00 - 0x4000 0FFF	1KB	UART1
	0x4000 1000 - 0x4000 13FF	1KB	UART2
	0x4000 1400 - 0x4000 17FF	1KB	ATIM
	0x4000 1800 - 0x4000 1BFF	1KB	GTIM1
	0x4000 1C00 - 0x4000 1FFF	1KB	GTIM2
	0x4000 2000 - 0x4000 23FF	1KB	UART3
	0x4000 2400 - 0x4000 27FF	1KB	GTIM3
	0x4000 2800 - 0x4000 2BFF	1KB	GTIM4
	0x4000 3400 - 0x4000 37FF	1KB	SPI3
	0x4000 4000 - 0x4000 43FF	1KB	SYSCTRL
APB2 peripheral	0x4000 4400 - 0x4000 47FF	1KB	RTC
	0x4000 4800 - 0x4000 4BFF	1KB	BTIM123
	0x4000 5000 - 0x4000 53FF	1KB	IWDT
	0x4000 5400 - 0x4000 57FF	1KB	WWDT
	0x4000 5800 - 0x4000 5BFF	1KB	I2C1
	0x4000 5C00 - 0x4000 5FFF	1KB	I2C2
	0x4000 6000 - 0x4000 63FF	1KB	LPTIM
	0x4000 6400 - 0x4000 67FF	1KB	HALLTIM
	0x4002 0000 - 0x4002 03FF	1KB	DMA
AHB peripheral	0x4002 2000 - 0x4002 23FF	1KB	FLASH CTRL
	0x4002 2400 - 0x4002 27FF	1KB	RAM CTRL



Table 2-1 CW32L012 memory and peripheral register boundary addresses (continued)

Device or bus	Boundary address	Size	Peripheral
AHB peripheral (continued)	0x4002 3000 - 0x4002 31FF	0.5KB	CRC
	0x4002 3200 - 0x4002 33FF	0.5KB	EAU
	0x4002 3400 - 0x4002 37FF	1KB	CORDIC
	0x4002 3800 - 0x4002 3BFF	1KB	CACHE_RAM
	0x4800 0000 - 0x4800 03FF	1KB	GPIOA/B/C/F
M0+ peripheral	0xE000 0000 - 0xE00F FFFF	1MB	M0+ peripheral



2.3 On-chip SRAM memory

The CW32L012 integrates 8KB on-chip SRAM, and the starting address is 0x2000 0000. SRAM supports access in 3 widths of byte (8bit), half word (16bit) or full word (32bit), and can be accessed by CPU at the maximum system clock frequency with zero latency.

Parity check

The SRAM supports the parity check function, which is enabled by default after the chip is powered on, and cannot be configured by the user.

The data bus of SRAM is 36bit, including 32bit of data bits and 4bit of parity bits (each 8bit of data bits is equipped with 1bit of parity bits), which is used to increase the robustness of the memory.

The parity bits are computed and stored when the CPU writes to the SRAM, and they are automatically checked when reading. After the system detects that the parity check fails, it will generate the corresponding interrupt flag.

For a detailed introduction to SRAM, see chapter [6 RAM memory](#).



2.4 On-chip FLASH memory

The on-chip FLASH memory consists of two physical areas: the main FLASH memory and the boot program memory.

- Main FLASH memory, a total of 64KB, the address space is 0x0000 0000 - 0x0000 FFFF. This area is mainly used to store application code and user data, which is programmable by the user.
- Boot program memory, a total of 2KB, the address space is 0x0010 0000 - 0x0010 07FF. This area is mainly used to store the BootLoader startup program, which has been programmed when the chip leaves the factory and cannot be changed by the user.

The FLASH controller implements various operations on FLASH (erase, write, read) , and the internal prefetch cache mechanism can speed up CPU code execution.

FLASH supports access with 3 widths of byte (8bit), half word (16bit) or full word (32bit), and the highest frequency of access is 24MHz. If the HCLK clock frequency of the system configuration is higher than 24MHz, a reasonable response waiting time must be configured through the WAIT bit field of the FLASH control register FLASH_CR2 to ensure that the FLASH is correctly accessed.

For a detailed introduction to FLASH, see chapter [7 FLASH Memory](#).



2.5 One-time programmable OTP memory

There is 22B memory space inside the chip as the OTP area, and the address space is 0x0010 0760 – 0x0010 0775. This memory area can be read and written in the same way as the FLASH area, but it can only be written once, and then it can only be read, not erased or written. This area is mainly used to store unchangeable information, such as user-defined product UID, algorithm key, etc., which are written when the device leaves the factory.



2.6 System boot configuration

The device supports 2 different boot modes, which are selected by the BOOT pin state, as shown in the following table:

Table 2-2 Boot mode configuration

Boot mode configuration	Mode
BOOT = 0	Boot from main FLASH memory, run user program.
BOOT = 1	Start from the boot program memory, and fix the BootLoader program of the running chip. At this time, the user can use the ISP communication protocol to perform FLASH programming through the UART1 interface (PA13/PA14).

The boot mode selection circuit only samples the state of the BOOT pin when the chip is de-reset, so the user must set the level state of the BOOT pin as needed before the chip is de-reset to determine the startup mode after the chip is reset.

After the system boot is completed, the CPU obtains the address of the top of the stack from the 0x0000 0000 address of the memory and starts executing the code from the address indicated by 0x0000 0004 of the memory.

The BootLoader program is located in the bootloader memory area and is programmed by the device provider at the time of manufacture. The user can program the FLASH via UART1 (pin PA13/PA14) using the ISP communication protocol.



2.7 Precautions

CW32L012 needs to pay attention to the following matters when using:

- A few registers such as FLASH, SRAM and GPIOx_ODR, CRC_DR support 8-bit/16-bit/32-bit access mode, and other peripherals only support 32-bit access mode. 8-bit/16-bit access to an address that does not support 8-bit/16-bit access mode will generate a HardFault hardware error exception.
- The RES areas in the system address allocation in [Figure 2-2: System address assignment](#) are reserved address spaces without corresponding physical devices. If these reserved address spaces are accessed, a HardFault hardware error exception will occur.
- After the chip is reset, all peripheral clocks except SYSTICK and SRAM are turned off. Before using these peripherals, the user must pass the peripheral clock enable control register (AHB peripheral clock enable control register SYSCTRL_AHBEN, APB peripheral clock Enable control register SYSCTRL_APBEN1/2) to open the configuration clock and working clock of the corresponding peripheral. Failure to turn on the peripheral configuration clock will result in an access failure: writing data fails, and no HardFault hardware error exception will be generated, and the result of reading data is not available. If the operating clock of the peripheral is not turned on, the main function of the peripheral is invalid.



3 Power control (PWR) and power consumption

3.1 Overview

The CW32L012 needs 2 sets of power supply when working: working power supply (VDD, VSS) and analog power supply (VDDA, VSSA). The user can choose to use the same power supply for the working power supply and the analog power supply, or use two different power supplies, but the difference between the two should not exceed 0.3V. Please refer to the relevant chapters of the datasheet for details of the power supply.

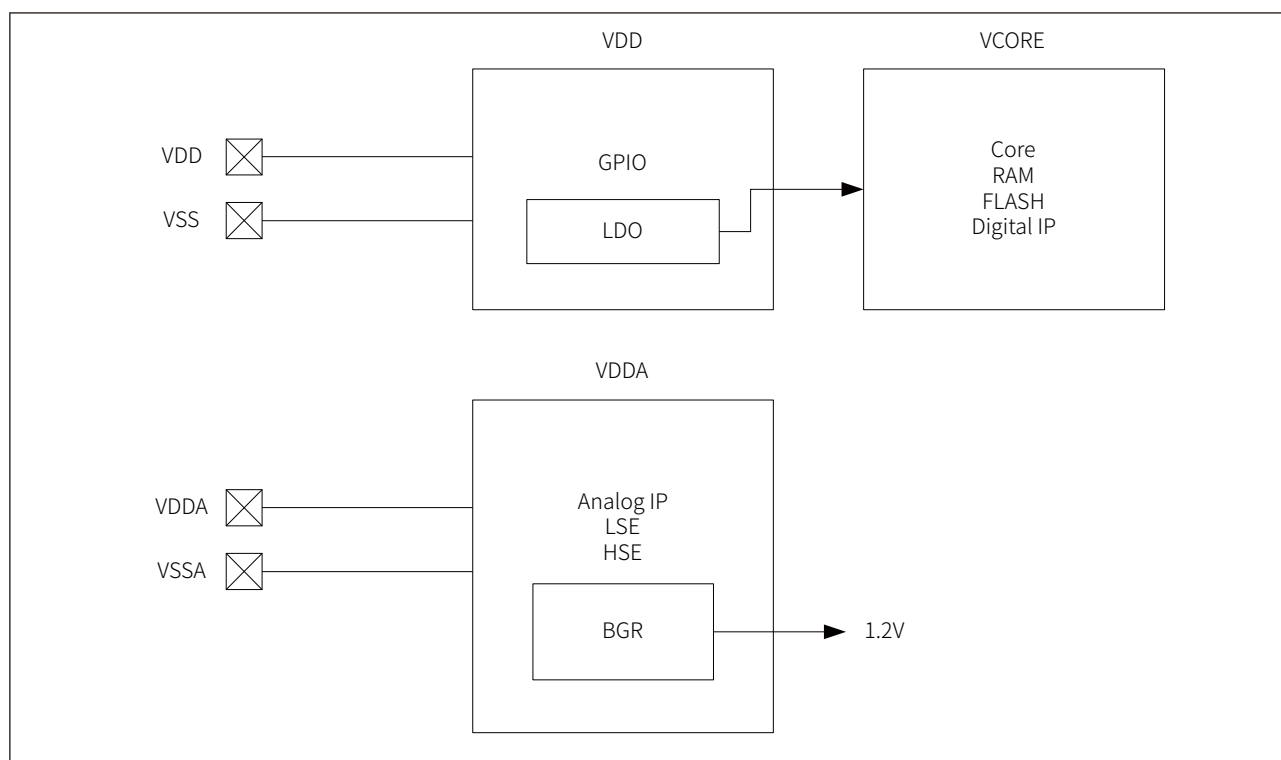
The CW32L012 embeds a 1.5V low dropout LDO voltage regulator to supply power to the digital power domain inside the chip.

The chip has a built-in reference voltage generator (BGR) circuit, which can provide reference voltage for other analog modules.

Attention: The difference between the supply voltage and the reference voltage must be higher than 0.3V. Please refer to the relevant section of the datasheet for detailed parameters of the voltage reference.

The division and allocation of power domains within the chip are shown in the following figure:

Figure 3-1 System power distribution



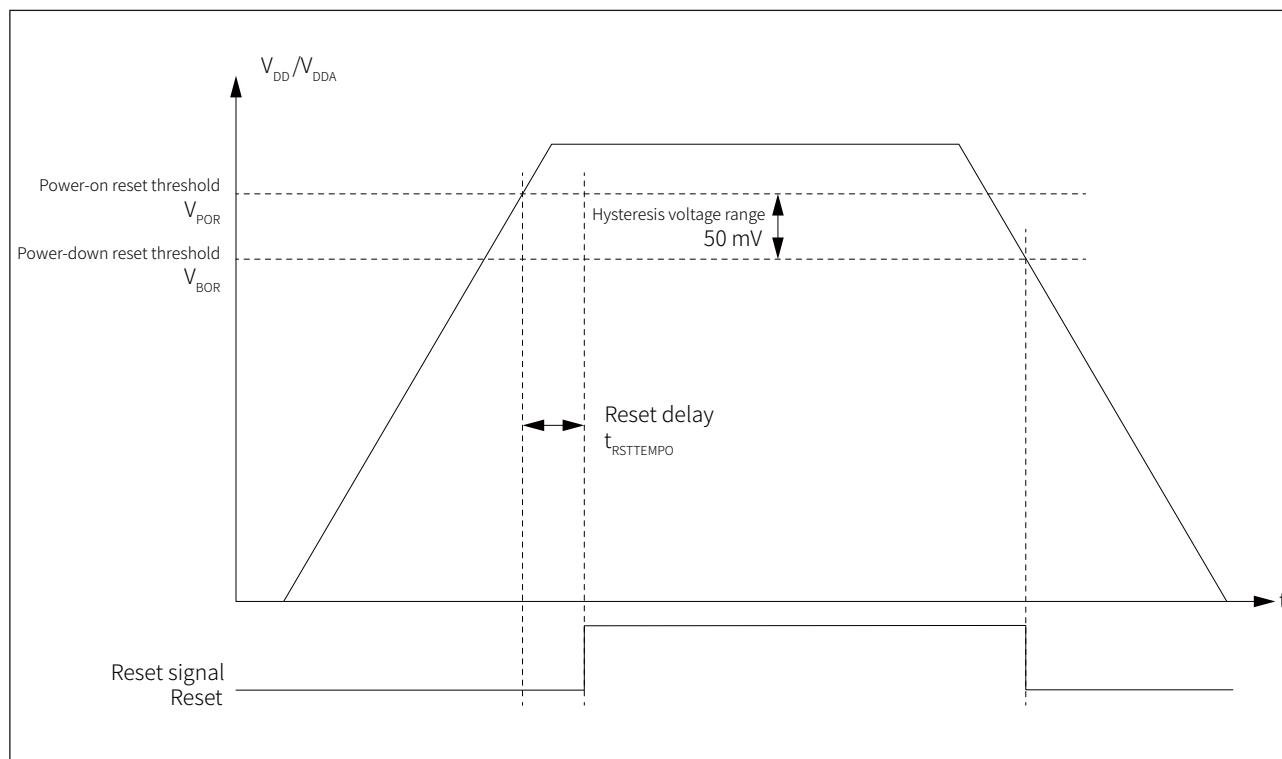
3.2 Power supply supervisor

Power on reset (POR)/ Brown-out reset (BOR)

The CW32L012 integrates power-on reset (POR) and Brown-out reset (BOR) power monitoring circuits, and is always in working state after power-on. POR/BOR monitors VDD and VDDA power supply voltages. When the monitored power supply voltage is lower than the reset threshold (V_{BOR}), the system will enter the reset state. Users do not need to add an external hardware reset circuit.

The reset signal waveforms of the power-on and power-down phases are shown in the following figure:

Figure 3-2 Power on reset/power down reset waveform



For more details on the power on/power down reset threshold, refer to the electrical characteristics section in the datasheet.

3.3 Operating mode

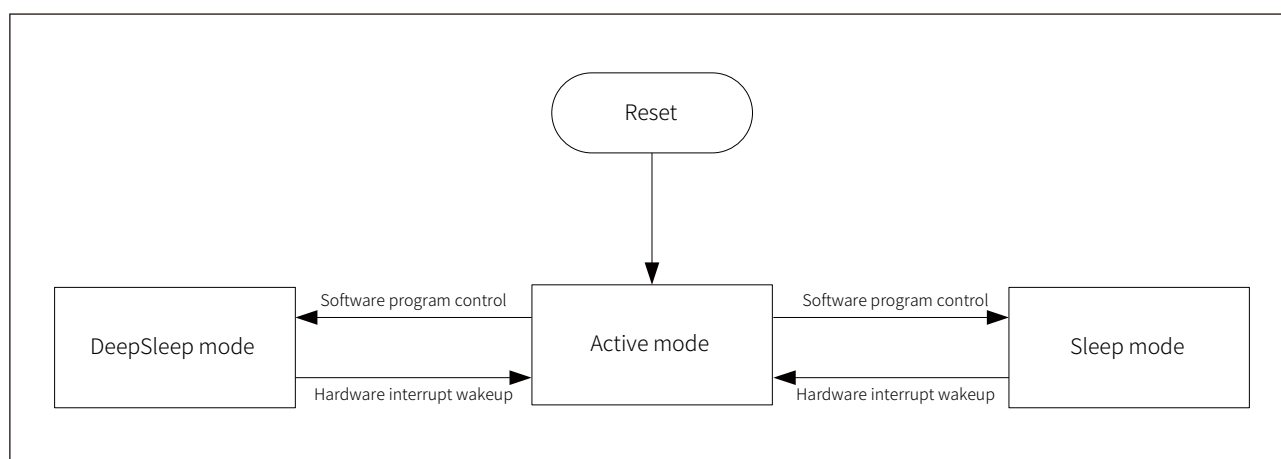
The CW32L012 supports 3 operating modes, and the integrated power management module of the built-in power supply automatically completes the unified management. The three working modes are:

- Active mode
- Sleep mode
- DeepSleep mode

After the power is turned on, the system automatically enters the active mode. Users can enter two low-power operating states of sleep or deep sleep through software programs; In the low-power operating state, the wake-up mechanism can be triggered by a hardware interrupt to return the system to the active mode.

The conversion mechanism of the 3 operating modes is shown in the following figure:

Figure 3-3 The conversion mechanism of the operating modes



In the active mode, the CPU runs normally, and all module users can use it normally. In Sleep mode, the CPU stops running, all peripherals are unaffected, and all I/O pins remain in the same state. In DeepSleep mode, the CPU stops running, the high-speed clock (HSE, HSIO SC) is automatically turned off, and the low-speed clock (LSE, LSI, RC10K) remains in the original state.

The CPU and clock states in different operating modes are shown in the following table:

Table 3-1 Operating mode and CPU and clock states

Operating mode	CPU state	High-speed clock	Low-speed clock
Active mode	Run	ON/OFF	ON/OFF
Sleep mode	Stop	ON/OFF	ON/OFF
DeepSleep mode	Stop	OFF	ON

3.3.1 Entering Sleep mode or DeepSleep mode

Using the M0+ core's ARM waits for interrupt-specific instruction, WFI (Wait for Interrupt), in conjunction with the SLEEPONEXIT and SLEEPDEEP bit fields of the M0+ core's system control register (SCR, System Control Register), Sleep mode or DeepSleep mode can be entered or exited immediately (interrupt service routine).

- Entering immediately:
Executing the WFI instruction, the MCU will immediately enter Sleep mode (when SLEEPDEEP is 0) or DeepSleep mode (when SLEEPDEEP is 1).
- Entering on exit:
Setting the SLEEPONEXIT bit to 1, after exiting the lowest priority interrupt service routine, the MCU will enter Sleep mode (when SLEEPDEEP is 0) or DeepSleep mode (when SLEEPDEEP is 1), needn't execute the WFI instruction.

Table 3-2 Entering Sleep mode or DeepSleep mode

Entry method	SLEEPONEXIT bit	Condition	SLEEPDEEP bit	Enter mode
Entering immediately	-	Executing the WFI instruction.	0	Sleep mode
			1	DeepSleep mode
Entering on exit	1	After exiting the lowest priority interrupt service routine.	0	Sleep mode
			1	DeepSleep mode

Attention: Before entering DeepSleep mode, if the FLASH is performing an erase/write operation, you must wait for the FLASH_CR1.BUSY flag bit to clear to 0, and make sure FLASH_CR1.MODE is 0 at the same time.

Attention: In DeepSleep mode, the system will automatically turn off the high-speed clock. If users need to keep some peripherals running in DeepSleep mode, they must start the corresponding low-speed clock and set the peripheral clock to this low-speed clock before entering DeepSleep mode.



3.3.2 Exiting Sleep mode or DeepSleep mode

In Sleep mode or DeepSleep mode, the interrupt can be used to wake up the CPU and return to active mode. However, it is worth noting that if the user executes the WFI command in the interrupt service routine to enter Sleep (including DeepSleep), an interrupt with a higher priority than this interrupt is required to wake up the CPU, so we strongly recommend that the user is ready to enter sleep, all interrupt service routines should be processed and all interrupt requests and interrupt flags should be cleared.

In different operating modes, the types of interrupts that the CPU can respond to are shown in the following table:

Table 3-3 Operating modes and interrupt source

Interrupt number	Interrupt source	Active mode	Sleep mode	DeepSleep mode
0	IWDT	Y	Y	Y
	WWDT	Y	Y	N
1	LVD	Y	Y	Y
2	RTC	Y	Y	Y
3	FLASH	Y	Y	N
	RAM	Y	Y	N
4	RCC	Y	Y	Y
5	GPIOA	Y	Y	Y
6	GPIOB	Y	Y	Y
7	GPIOC	Y	Y	Y
8	GPIOF	Y	Y	Y
9	DMA1/2	Y	Y	N
10	DMA3/4	Y	Y	N
11	CORDIC	Y	Y	N
12	ADC1	Y	Y	N
13	ATIM	Y	Y	N
14	VC1/3	Y	Y	Y
15	VC2/4	Y	Y	Y
16	GTIM1	Y	Y	N
17	GTIM2	Y	Y	N
18	GTIM3/4	Y	Y	N
19	LPTIM	Y	Y	Y
20	BTIM1	Y	Y	N
21	BTIM2	Y	Y	N
22	BTIM3/HALLTIM	Y	Y	N



Table 3-3 Operating modes and interrupt source (continued)

Interrupt number	Interrupt source	Active mode	Sleep mode	DeepSleep mode
23	I2C1	Y	Y	N
24	I2C2	Y	Y	N
25	SPI1	Y	Y	N
26	SPI2	Y	Y	N
27	UART1	Y	Y	Y
28	UART2	Y	Y	Y
29	UART3	Y	Y	Y
30	ADC2/DAC12	Y	Y	N
31	FAULT	Y	Y	Y

Exiting Sleep mode using an interrupt, the user must enable the enable bit for this interrupt before entering Sleep (including DeepSleep mode).

After the interrupt wakes up and exits Sleep mode, the CPU will immediately enter the interrupt service routine for this interrupt. If the user does not set this interrupt service routine, and when entering sleep immediately: the CPU will continue to execute the next statement of the WFI instruction that entered sleep; when entering sleep when exiting: continue to execute the next statement of the last entered interrupt service routine. In general, based on system reliability considerations, it is strongly recommended that the user set the service routine for this interrupt, and clear the interrupt request and interrupt flag in the interrupt service routine.

When the interrupt wakes up to exit the DeepSleep mode, the CPU operation state is the same as the exit from the Sleep mode.

In DeepSleep mode, the system will automatically turn off the high-speed clock. When exiting DeepSleep, CW32L012 adds an additional system clock option for users. Users can choose to continue to use the clock used when entering DeepSleep, or select HSI 4MHz as the system clock. Configure the WAKEUPCLK bit field of the system control register SYSCTRL_CR2 to 1, then the internal high-speed clock HSI 4MHz is automatically used as the system clock after the interrupt wakes up and exits the DeepSleep mode, it can speed up the system wake-up.



3.3.3 Operating mode and reset source

Even in Sleep mode or DeepSleep mode, the CPU can respond to some reset sources. The hardware reset or software reset that the CPU can respond to in different working modes is shown in the following table:

Table 3-4 Operating mode and reset source

Reset source	Active mode	Sleep mode	DeepSleep mode
Power-on reset/Brown-out reset (POR/BOR)	Y	Y	Y
Pin input reset (NRST)	Y	Y	Y
LVD reset	Y	Y	Y
IWDT reset	Y	Y	Y
Kernel LOCKUP fault reset	Y	N	N
Kernel SYSRESETREQ reset	Y	N	N



3.4 Low-power consumption applied

In Sleep mode, the CPU stops running, and all peripherals keep running, including ARM® Cortex®-M0+ core peripherals, such as NVIC, SysTick and other peripherals. Sleep mode consumes less power than active mode.

In DeepSleep mode, the CPU stops running, the high-speed clock is turned off, the low-speed clock remains unchanged, some peripherals can be configured to continue to run, and the NVIC interrupt processing still works. DeepSleep mode consumes much less power than Sleep mode.

Users can reduce system running power consumption by:

1. Decrease system clock frequency:
 - Use low-frequency high-speed clock HSI, HSE or low-speed clock LSI, LSE.
 - Reduce the frequency of SYSCLK, HCLK, PCLK by programming the prescaler registers:
 - Set the SYSCLK bit field of the SYSCTRL_CR0 register to select the appropriate clock source.
 - Set the HCLKPRS bit field of the SYSCTRL_CR0 register to reduce the HCLK frequency.
 - Set the PCLKPRS bit field of the SYSCTRL_CR0 register to reduce the PCLK frequency.
2. Turn off clocks and peripherals which not in use during sleep:
 - AHB bus clock HCLK and APB bus clock PCLK, can be turned off as needed.
 - Turn off clocks to peripherals not related to wakeup:
 - AHB peripheral clock enable control register, SYSCTRL_AHBEN.
 - APB peripheral clock enable control register 1, SYSCTRL_AHBEN1.
 - APB peripheral clock enable control register 2, SYSCTRL_AHBEN2.



3.5 Cortex®-M0+ kernel system control register (SCB->SCR)

Address: 0xE000 ED10 Reset value: 0x0000 0000

Bit field	Name	Permission	Function description
31:5	RFU	-	Reserved bits, please keep the default value.
4	SEVONPEND	RW	When set to 1, each new suspension of an interrupt generates an event, which can be used to wake up the processor if WFE sleep is used.
3	RFU	-	Reserved bits, please keep the default value.
2	SLEEPDEEP	RW	When set to 1, enter DeepSleep mode when WFI or SLEEPONEXIT is 1 and all interrupt service routines are exited. When set to 0, enter Sleep mode when WFI or SLEEPONEXIT is 1 and all interrupt service routines are exited.
1	SLEEPONEXIT	RW	When set to 1, the processor automatically enters Sleep mode (or DeepSleep mode) when all interrupt service routines are exited. When set to 0, the hibernation function on exit is disabled.
0	RFU	-	Reserved bits, please keep the default value.



4 Reset and clock control (RCC)

4.1 System reset

The CW32L012 supports the following 6 system resets:

- Power on reset (POR)/ Brown-out reset (BOR)
- Pin input reset (NRST)
- IWDG reset
- LVD reset
- Kernel SYSRESETREQ reset
- Kernel LOCKUP fault reset

The reset range of each system reset is shown in the following table:

Table 4-1 Reset method and range

Reset method	Reset range
Power on reset (POR)/ Brown-out reset (BOR)	Entire MCU
Pin input reset (NRST)	Entire MCU (Except RTC)
IWDG reset	M0+ core/peripherals (Except RAM controller/RTC)
LVD reset	M0+ core/peripherals (Except LVD controlle section/RTC)
Kernel SYSRESETREQ reset	M0+ core (Except SWD debug logic) / Peripherals (Except RAM controller/LVD/RTC)
Kernel LOCKUP fault reset	M0+ core/peripherals (Except RAM controller/LVD/RTC)

After the system reset occurs, the CPU runs again, most of the registers are reset to default values, and the program starts to execute from the reset interrupt entry address of the interrupt vector table.

The user can query the reset source of this system reset through the system reset flag register SYSCTRL_RESETFLAG. The reset flag is set by hardware and cleared by software. It is recommended that users clear the relevant flag bits of this register to 0 after reading the flag bits to avoid confusion after the next reset.

4.1.1 Power on reset/Brown-out reset (POR/BOR)

The CW32L012 integrates special POR and BOR circuits to monitor the power supply voltage, and keeps the chip in the reset state when the power supply voltage is lower than the safe range to prevent the chip from malfunctioning during power-on/power-off. In order to ensure the stable operation of the system, the user must keep the power supply voltage within a safe range.

For the specific functions of POR and BOR, please refer to chapter [3 Power control \(PWR\) and power consumption](#).



4.1.2 Pin input reset (NRST)

The CW32L012 has a special reset input pin, inputting a low-level signal of a certain width will cause the system to reset. A dedicated anti-shake circuit is designed inside the chip, and low-level pulse signals shorter than 20μs will be shielded.

The reset input pin has a built-in pull-up resistor. If the user needs to connect an external RC circuit, the influence of the internal pull-up resistor must be considered. For the circuit characteristics of the built-in pull-up resistor, please refer to the relevant section of the datasheet.

4.1.3 IWDG/WWDT reset

The CW32L012 integrates an independent watchdog (IWDG) and a window watchdog (WWDT). When the IWDG or WWDT meets the reset condition, a reset signal will be generated to cause a system reset.

For detailed functions of IWDG and WWDT reset, please refer to chapter [19 Independent watchdog timer \(IWDG\)](#), chapter [20 Window watchdog timer \(WWDT\)](#).

4.1.4 LVD low voltage detection reset

The CW32L012 integrates a low voltage detector (LVD), which can continuously compare the selected monitoring voltage with the set LVD threshold voltage. When the comparison result meets the trigger condition, an LVD reset signal will be generated to cause a system reset.

Compared with the POR/BOR reset method, the LVD low voltage detection reset function is more powerful. The reset threshold voltage, monitoring voltage source, pulse filter width and hysteresis time can be set through the relevant registers.

For detailed functions of LVD, please refer to chapter [28 Low voltage detector \(LVD\)](#).

4.1.5 Kernel SYSRESETREQ reset

The core SYSRESETREQ reset is a software reset, which is implemented by setting the SYSRESETREQ bit field of the Application Interrupt and Reset Control Status Register (AIRCR) of the ARM® Cortex®-M0+. Setting this bit to 1 by the application program will generate a core SYSRESETREQ reset, thereby implementing a software reset.

4.1.6 Kernel LOCKUP fault reset

When the CPU encounters a serious exception (such as the read instruction is invalid, the bit width does not match the target address when accessing FLASH), it will stop the PC pointer at the current address and lock it, and generate the kernel LOCKUP fault reset signal.

After the chip is powered on, the LOCKUP reset function is disabled by default, and the user needs to manually enable it. By setting the LOCKUP bit field of the system control register SYSCTRL_CR2 to 1, the LOCKUP reset function can be enabled.



4.2 Peripheral reset

Users can use software to reset various peripherals inside CW32L012 individually. Peripheral reset can restore the registers, state machines and various control logic of each peripheral to the default state after power-on reset. The user can perform the independent reset operation of each peripheral module by setting the 3 registers: `SYSCTRL_AHBRST`, `SYSCTRL_APBRST1`, and `SYSCTRL_APBRST2`.

For power-on reset/brown-out reset (POR/BOR), the internal peripheral modules are already in the default state after reset, and the module can be initialized and configured directly, and there is no need to independently reset each module.

For other types of system resets, some peripherals may retain the working state before the reset. If the user needs to restore the peripherals to the default state of power-on, they should reset the peripherals through the corresponding reset registers before using them. The RTC module is a typical peripheral that may need to retain the current state after reset. For specific judgment logic examples, please refer to the relevant routines.



4.3 Clock and Control

4.3.1 Overview

The CW32L012 has a built-in multi-channel clock generation circuit, which generates various clocks of different frequencies through the frequency divider and multiplexer for use by the CPU and various peripherals.

The internal clock SysClk of the system is divided to provide the advanced high-performance bus clock HCLK for the CPU core, and the HCLK clock is divided to provide the advanced peripheral clock PCLK for digital and analog peripherals.

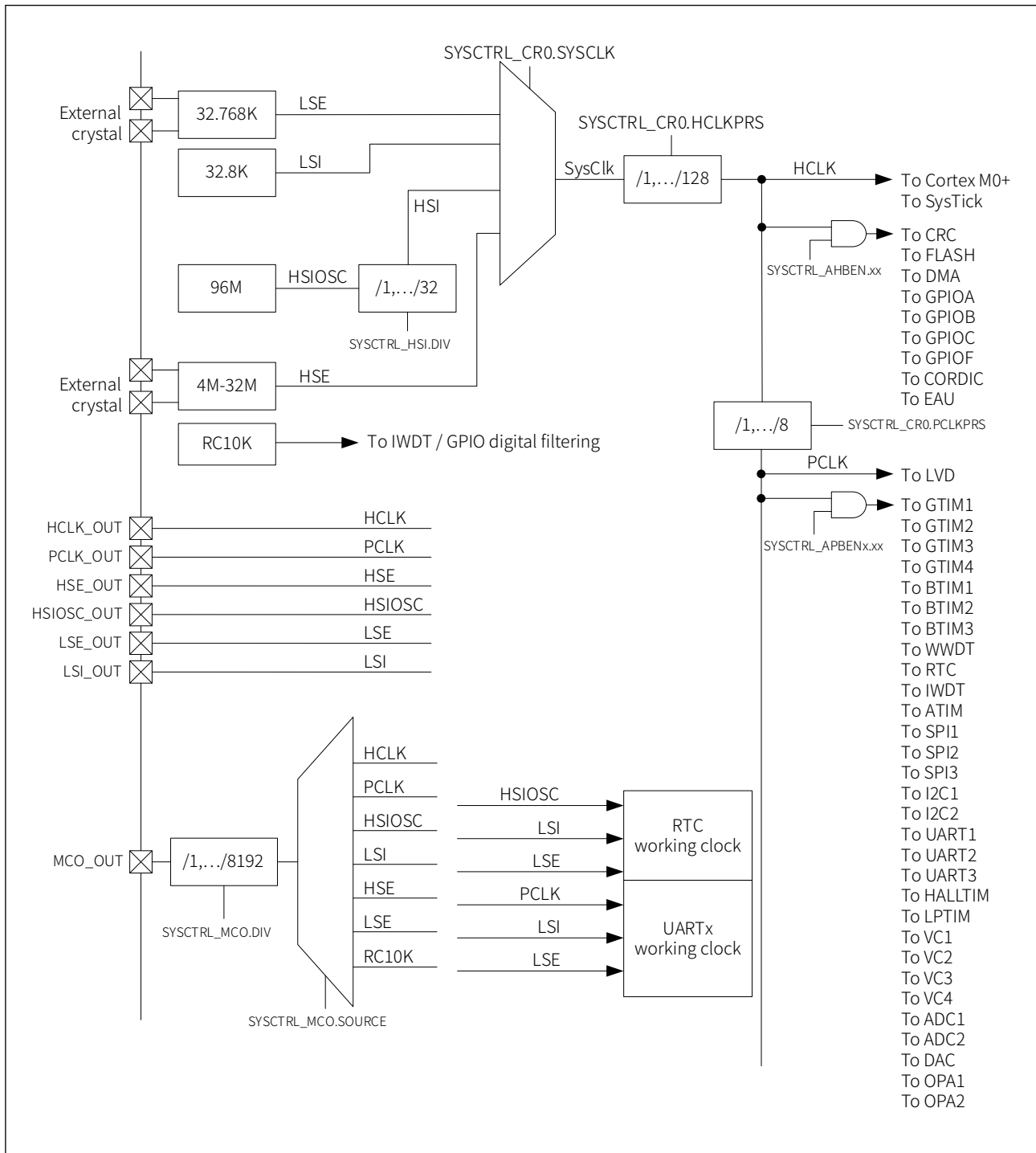
System clock SYSCLK has 4 clock sources:

- External high speed oscillator clock (HSE).
- Low-speed external oscillator clock (LSE).
- The HSI clock is generated by dividing the internal high-speed RC oscillator clock (HSIOSC).
- Low-speed internal RC oscillator clock (LSI).

The system internal clock tree is shown in the following figure:



Figure 4-1 System internal clock tree



The high-speed internal RC oscillator clock HSIOSC is divided by the frequency divider to generate the HSI clock. The frequency division coefficient is set through the DIV bit field of the built-in high-frequency clock control register SYSTRL_HSI, and the effective frequency division coefficients are 1/2/3/4/5/6/7/8/9/10/12/16/20/24/28/32.

The system clock SysClk can be selected from 4 clock sources: HSE, LSE, HSI, LSI, which can be selected by the SYSCLK bit field of the system control register SYSTRL_CR0.

Advanced high-performance bus clock HCLK is generated by the frequency division of the system clock SysClk, and is used as the configuration clock and working clock of the M0+ core, SysTick, DMA, FLASH, CRC, GPIO and other modules. The frequency division factor is set by the HCLKPRS bit field of the system control register SYSTRL_CR0, and the effective frequency division factor is 2^n ($n = 0 \sim 7$).

Advanced peripheral clock PCLK, which is generated by frequency division of HCLK, is used as the configuration clock and working clock of peripherals such as timer, SPI, and I2C. The frequency division factor is set by the PCLKPRS bit field of the system control register SYSTRL_CR0, and the effective frequency division factor is 2^n ($n = 0 \sim 3$).

In addition to the 4 clock sources used as the system clock, CW32L012 also has 1 dedicated low-speed internal clock sources:

- Internal RC10K clock.
It can be used as a count clock for the IWDG module and as a filter clock for the interrupt input signal of the GPIO port.

4.3.2 System clock and operating mode

The CW32L012 supports 3 operating modes: Active mode, Sleep mode and DeepSleep mode.

- In the Active mode, the CPU runs normally, and all module users can use it normally.
- In the Sleep mode, the CPU stops running, and the clock oscillators and peripherals remain unchanged.
- In DeepSleep mode, the CPU stops running, and the high-speed clock HSE, HSIOSC oscillator are automatically turned off to save power; The oscillators of the low-speed clocks LSE, LSI, RC10K remain unchanged; Whether the system clock SysClk and the HCLK and PCLK clocks are valid depends on the clock source status of the SysClk system clock.

After waking up from DeepSleep mode, if the WAKEUPCLK bit field of the system control register SYSTRL_CR2 is configured to 1, the system will automatically use the HSI 4MHz as the clock source of the system clock, and the original system clock source remains enabled. If SYSTRL_CR2.WAKEUPCLK is 0, the system will wait for the system clock source used before the system enters DeepSleep to stabilize before starting to run. Since the HSI starts up quickly and can quickly respond to user needs, it is recommended to switch the clock source of the system clock to the HSI clock or configure SYSTRL_CR2.WAKEUPCLK to 1 before entering DeepSleep mode.



4.3.3 HSE clock

The high-speed external clock signal (HSE) supports 2 operating modes:

- For crystal/ceramic resonators, set the MODE bit field of the external high-frequency crystal control register SYSCTRL_HSE to 0, and configure the OSC_IN and OSC_OUT pins as analog functions.
- For external clock input, set the MODE bit field of the external high-frequency crystal control register SYSCTRL_HSE to 1, and configure the OSC_IN pin as a digital input function.

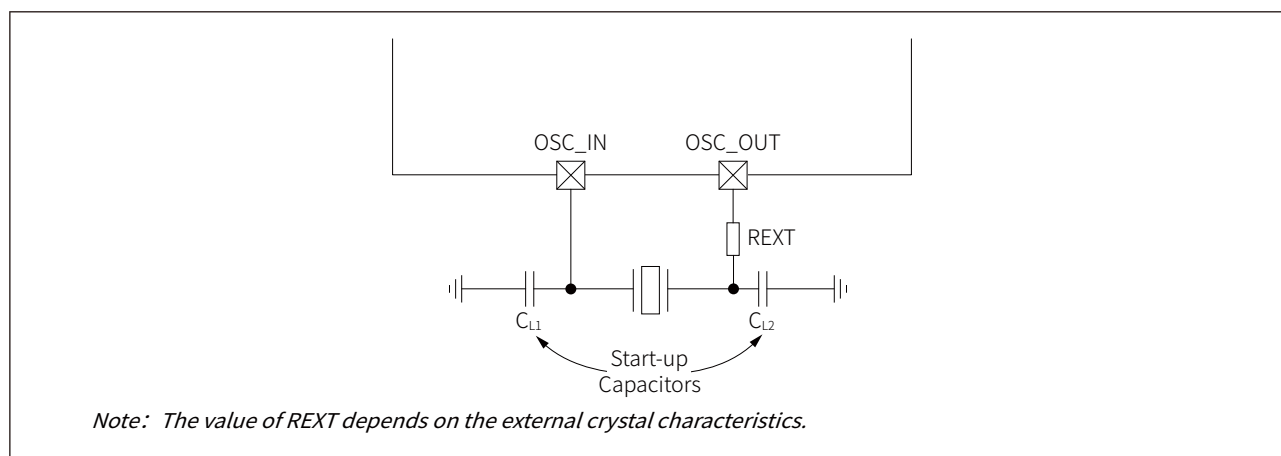
Table 4-2 HSE operating mode configuration

Operating mode	SYSCTRL_HSE.MODE	OSC_IN	OSC_OUT
Crystal/Ceramic resonators	0	Analog	Analog
External clock input	1	Digital input	Can be used as general purpose GPIO

Crystal/Ceramic resonators clock mode

The external quartz crystal/ceramic resonator is connected between the OSC_IN and OSC_OUT pins, and cooperates with the start-up capacitor to generate a stable 4~32MHz clock signal. The circuit connection is shown in the following figure:

Figure 4-2 HSE external quartz crystal/ceramic resonator connection method



The crystal/ceramic resonator and start-up capacitor must be placed as close as possible to the chip oscillator pins to minimize oscillator output distortion and start-up time.

The value of the start-up capacitor must be adjusted according to the selected crystal/ceramic resonator. The calculation method is:

$$C_L = (C_{L1} \times C_{L2}) / (C_{L1} + C_{L2}) + C_0$$

Among them,

C_L is the nominal load capacitance of the crystal;

C_0 is the stray capacitance value, which is related to the design of the circuit board. Generally, the value can be 4pF~6pF;

C_{L1} and C_{L2} are two external start-up capacitors, usually C_{L1} and C_{L2} are same.

If the user uses a crystal resonator with a nominal load capacitance value of 12.5pF, the common value of the two external starting capacitors for the crystal is 15pF.

Attention: Since there are many crystal manufacturers, the crystal characteristics of each manufacturer are different. If the output clock signal of the HSE oscillator is not ideal, you can contact the crystal supplier for coordination.

Before HSE clock is stable, users can adjust driving capability by setting PDRIVER bit field of external high-frequency crystal control register SYSCTRL_HSE. Eight driving capabilities can be selected. The stronger the driving capability, the greater the power consumption. After HSE clock is stable, users can adjust driving capability by setting DRIVER bit field of external high-frequency crystal control register SYSCTRL_HSE. Eight driving capabilities can be selected. The stronger the driving capability, the greater the power consumption. Users can balance clock reliability, start-up time, and low power consumption according to their needs.

External clock input mode

In the external clock input mode, the external clock is input from the OSN_IN pin, and the OSC_OUT pin can be used as a general-purpose GPIO. The input clock signal can be square wave, sine wave or triangular wave, the duty cycle must be between 40%~60%, and the frequency is between 4~32MHz.

When an external oscillator is used, the PA01 or PA11 pin (which needs to be configured for HEXEN multiplexed output function) outputs an enable signal that outputs different levels in chip run mode and DeepSleep mode, respectively. The output polarity is configured through the HEXENPOL bit field of the SYSCTRL_HSE register.

In the two working modes of HSE, the HSE clock oscillator is turned off by default after power-on, and is enabled by setting the HSEEN bit field of the system control register SYSCTRL_CR1 to 1.

After the HSE oscillator is started, when the internal clock monitoring module of the chip detects a certain number of HSE clock signals, the HSE clock is considered to be stable. The number of detection clocks can be set through the WAITCYCLE bit field of the external high-frequency crystal control register SYSCTRL_HSE, as shown in the following table:

Table 4-3 HSE stability detection clock signal number

SYSCTRL_HSE.WAITCYCLE	Number of HSE detection clocks
00	8192
01	32768
10	131072 (Defaults)
11	262144

The start-up state of the HSE clock can be determined through the STABLE flag of the external high-frequency crystal control register SYSCTRL_HSE. If the STABLE flag is 1, the HSE clock is stable, and if it is 0, the HSE clock is not stable.

Attention: All parameters of the HSE oscillator should be set before starting, and it is forbidden to modify the relevant parameters after starting.



4.3.4 HSIOSC clock

The HSIOSC clock is generated by the internal RC oscillator, no external circuit is required, the startup speed is fast and the frequency is fixed at 96MHz.

The frequency of the RC oscillator output clock is affected by the chip processing process, operating voltage, ambient temperature and other factors. CW32L012 provides the HSIOSC clock frequency calibration function. Users can calibrate the HSIOSC clock by setting the TRIM bit field value of the built-in high-frequency clock control register SYSCTRL_HSI frequency, see section [4.4.2 Clock calibration](#) for details.

The HSIOSC internal high-speed RC oscillator is turned on by default after the chip is powered on. Users can turn it off by setting the HSIEN bit field of SYSCTRL_CR1 in the system control register to 0. If the user stops and restarts the HSIOSC oscillator, the STABLE flag of the built-in high-frequency clock control register SYSCTRL_HSI can be used to determine whether the HSI clock is stable. If the STABLE flag is 1, it means that the HSIOSC clock is stable, and if it is 0, it means that the HSIOSC clock is not stable yet.

Attention: All parameters of the HSIOSC oscillator should be set before starting, and it is forbidden to modify the relevant parameters after starting.

The HSIOSC clock outputs the HSI clock after frequency division. The frequency division coefficient is set by the DIV bit field of the built-in high-frequency clock control register SYSCTRL_HSI, and the effective frequency division coefficients are 1/2/3/4/5/6/7/8/9/10/12/16/20/ 24/ 28/32, and the default after power-on is 24. Therefore, the HSI clock has a default frequency of 4MHz.



4.3.5 LSE clock

External low-speed clock (LSE) supports 2 operating modes:

- For crystal/ceramic resonators, set the MODE bit field of the external low-frequency crystal control register SYSCTRL_LSE to 0, and configure the OSC32_IN and OSC32_OUT pins as analog functions.
- For external clock input, set the MODE bit field of the external low-frequency crystal control register SYSCTRL_LSE to 1, and configure the OSC32_IN pin as a digital input function.

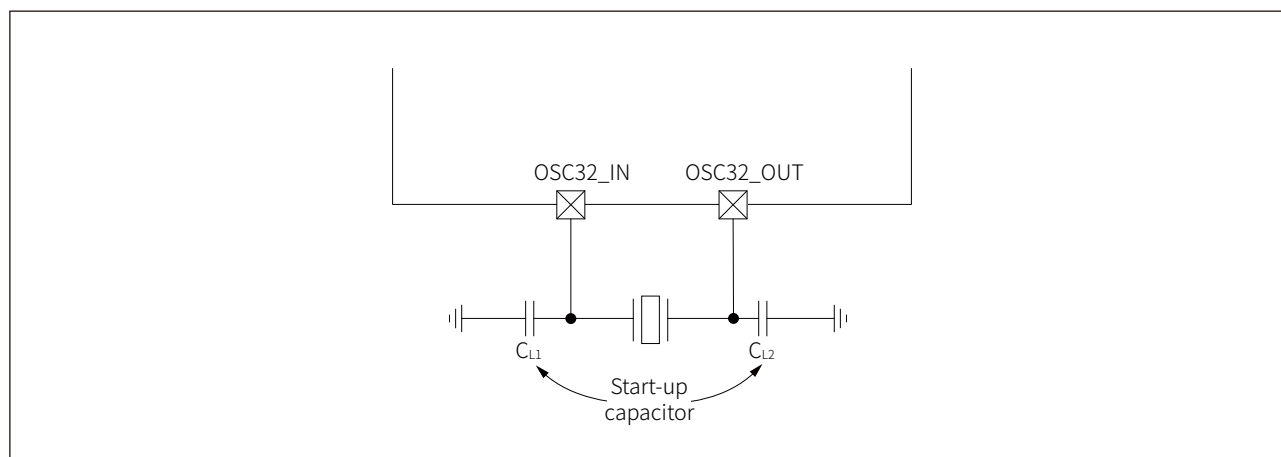
Table 4-4 LSE operating mode configuration

Operating mode	SYSCTRL_LSE.MODE	OSC32_IN	OSC32_OUT
Crystal/Ceramic resonators	0	Analog	Analog
External clock input	1	Digital input	Can be used as general purpose GPIO

Crystal/Ceramic resonators clock mode

The external low-frequency quartz crystal/ceramic resonator is connected between the two pins of OSC32_IN and OSC32_OUT. With the start-up capacitor, a stable clock signal of up to 1MHz can be generated. LSE is usually used as the operating clock of the system real-time clock RTC. At this time, a quartz crystal/ceramic resonator with a frequency of 32.768kHz should be selected. The external circuit connection is shown in the figure below:

Figure 4-3 LSE external quartz crystal/ceramic resonator connection method



The quartz crystal/ceramic resonator and start-up capacitor must be placed as close as possible to the chip oscillator pins to minimize oscillator output distortion and start-up time.

The value of the start-up capacitor must be adjusted according to the selected crystal/ceramic resonator. The calculation method is:

$$C_L = (C_{L1} \times C_{L2}) / (C_{L1} + C_{L2}) + C_0$$

Among them,

C_L is the nominal load capacitance of the crystal;

C_0 is the stray capacitance value, which is related to the design of the circuit board. Generally, the value can be 4~6pF;

C_{L1} and C_{L2} are two external start-up capacitors, usually C_{L1} and C_{L2} are same.



If the user uses a crystal resonator with a nominal load capacitance value of 12.5pF, the common value of the two external starting capacitors for the crystal is 15pF.

Attention: Since there are many crystal manufacturers, the crystal characteristics of each manufacturer are different. If the output clock signal of the LSE oscillator is not ideal, you can contact the crystal supplier for coordination.

Before the LSE clock is stabilised, the user can adjust the drive capability by setting the PDRIVER bit field of the external low frequency crystal control register SYSCTRL_LSE, there are sixteen drive capabilities to choose from, the stronger the drive capability the higher the power consumption. After the LSE clock is stabilised, the user can adjust the drive capability by setting the DRIVER bit field of the external low-frequency crystal control register SYSCTRL_LSE. There are sixteen levels of drive capability to choose from. The stronger the drive capability, the higher the power consumption. Users can balance clock reliability, startup time, and low power consumption as needed.

External clock input mode

In the external clock input mode, the external clock is input from the OSN32_IN pin, and the OSC32_OUT pin can be used as a general-purpose GPIO. The input clock signal can be square wave, sine wave or triangular wave, the duty cycle must be between 45%~55%, and the frequency can be up to 1MHz.

In the two working modes of HSE, the LSE clock oscillator is turned off by default after power-on, and is enabled by setting the LSEEN bit field of the system control register SYSCTRL_CR1 to 1.

After the LSE oscillator is started, when the internal clock monitoring module of the chip detects a certain number of LSE clock signals, the LSE clock is considered to be stable. The number of detection clocks can be set through the WAITCYCLE bit field of the external low-frequency crystal control register SYSCTRL_LSE, as shown in the following table:

Table 4-5 LSE stability detection clock signal number

SYSCTRL_LSE.WAITCYCLE	Number of LSE detection clocks
00	256
01	1024
10	4096 (Defaults)
11	16384

The start-up state of the LSE clock can be determined through the STABLE flag of the external low-frequency crystal control register SYSCTRL_LSE. If the STABLE flag is 1, the LSE clock is stable, and if it is 0, the LSE clock is not stable.

Attention: All parameters of the LSE oscillator should be set before starting, and it is forbidden to modify the relevant parameters after starting.

The LSE oscillator has a lock function. After the lock function is turned on, the software is prohibited from clearing the enable bit (SYSCTRL_CR1.LSEEN) of the LSE clock oscillator. That is, once the LSE is turned on, it cannot be turned off by software. The locking function is mainly designed to meet the requirements of some special application scenarios, such as application scenarios sensitive to RTC time loss, which can effectively prevent RTC time loss caused by misoperation. The lock function is disabled by default and is enabled by setting the LSELOCK bit field of the system control register SYSCTRL_CR1 to 1.



4.3.6 LSI clock

The LSI clock is generated by the internal low-speed RC oscillator with a default frequency of 32.8kHz. The internal low-speed RC oscillator requires no external circuitry and is less expensive than the LSE clock, but less accurate than the LSE clock.

The frequency of the RC oscillator output clock is affected by the chip processing process, operating voltage, ambient temperature and other factors. CW32L012 provides the LSI clock frequency calibration function. Users can calibrate the LSI clock frequency by setting the value of the TRIM bit field of the built-in low-frequency clock control register SYSCTRL_LSI. For details, please refer to [4.4.2 Clock calibration](#).

The LSI internal low-speed RC oscillator is disabled by default, and is enabled by setting the LSIEN bit field of the system control register SYSCTRL_CR1 to 1. After the LSI oscillator is started, if the internal clock monitoring module of the chip detects a certain number of LSI clock signals, the LSI clock is considered to be stable. The number of detection clocks can be set by the WAITCYCLE bit field of the built-in low-frequency oscillator control register SYSCTRL_LSI, as shown in the following table:

Table 4-6 LSI stability detection clock signal number

SYSCTRL_LSI.WAITCYCLE	Number of LSI detection clocks
00	6 (Defaults)
01	18
10	66
11	258

The STABLE flag of the built-in low-frequency clock control register SYSCTRL_LSI can be used to determine whether the LSI clock is stable. If the STABLE flag is 1, it means that the LSI clock is stable, and if it is 0, it means that the LSI clock is not stable.

Attention: All parameters of the LSI oscillator should be set before starting, and it is forbidden to modify the relevant parameters after starting.

4.3.7 SysClk system clock

System clock SysClk can choose from 4 clock sources, including HSE, LSE, HSI, and LSI.

After the system power-on reset is completed, HSI is selected as the clock source of SysClk by default, and the default value of the clock frequency is 4MHz.

The user can select the system clock source through the SYSCCLK bit field of the system control register SYSCTRL_CR0. After selecting a clock as the system clock source, the user cannot stop the clock by setting the enable bit SYSCTRL_CR1.xxxEN of the clock circuit to 0.



4.3.8 On-chip peripheral clock control

On-chip peripherals generally need the configuration clock and the working clock. The configuration clock is used to respond to the read and write operations of the peripheral registers by the CPU. The working clock is used for the function realization of each peripheral (such as the transmission clock of the UART, the count clock of the timer, etc.).

The configuration clock and working clock of the peripheral must be turned on before using the peripheral, otherwise the peripheral cannot work. By setting the corresponding bits of the AHB peripheral clock enable control register `SYSCTRL_AHBEN` and the APB peripheral clock enable control register `SYSCTRL_APBEN1` and `SYSCTRL_APBEN2` to 1, the configuration clock and working clock of the corresponding peripherals are turned on.

Peripherals such as RTC, UART, and FLASH only open the configuration clock, and the working clock is configured through the clock source selection register of each module.

When the peripherals do not need to be used, the peripherals can be disabled by turning off the configuration clocks and operating clocks of the peripherals, which can effectively reduce the power consumption of the chip.

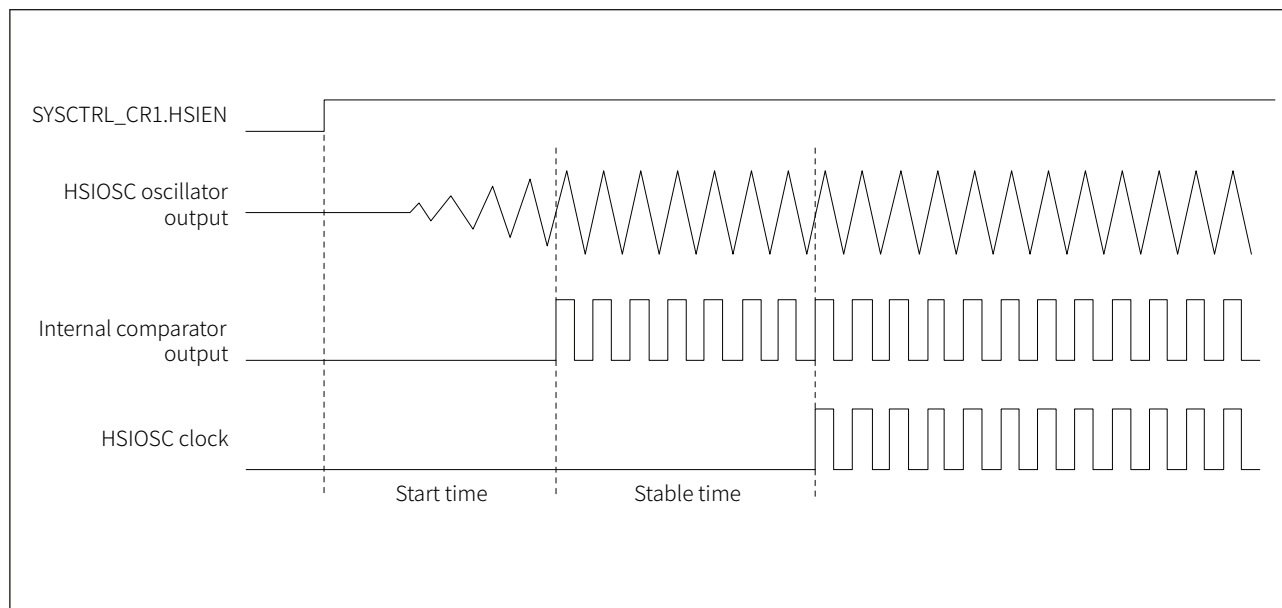


4.4 Clock Start, calibration, and status detection

4.4.1 Clock start

The clock source startup process of CW32L012 is similar. Take HSIOSC as an example to illustrate the stabilization process after clock startup, as shown in the following figure:

Figure 4-4 HSIOSC clock startup process



When SYSCTRL_CR1.HSIEN is set to 1, the HSIOSC clock oscillation circuit starts to work, but the output clock signal amplitude is very small at this time. After the start-up time stage, the amplitude and duty cycle of the output clock signal can meet the needs of the internal sampling circuit and enter the stabilization time stage. In the stable time stage, the internal clock monitoring circuit of the chip counts the clock signals output by the HSIOSC. When the count value reaches the set number, the HSIOSC clock signal is considered to be stable, and the HSIOSC clock stability flag bit SYSCTRL_HSI.STABLE is set to 1.

The clock start-up process of other clock oscillators is similar, but note that the number of detection clocks in the stable phase, the detection time and the number of detection clocks for start-up failure detection are different for each clock oscillator. For details, see the description of each clock oscillator.

4.4.2 Clock calibration

Clock calibration is mainly for HSIOSC clock and LSI clock. Clock frequency calibration is achieved by adjusting the TRIM value of the oscillator.

HSIOSC clock calibration

The safe working range of HSIOSC is 90~100MHz . If it exceeds the safe range, the chip may be abnormal. When the chip leaves the factory, the calibration parameters of the 96MHz frequency point have been pre-adjusted and stored in FLASH. The application program only needs to read and write the calibration value in FLASH to SYSCTRL_HSI.TRIM to obtain a precise 96MHz clock. The storage address of the 96MHz frequency calibration value is: 0x0010 07C0 - 0x0010 07C1. If other frequency clocks are needed, users need to adjust the value of SYSCTRL_HSI.TRIM by themselves.

LSI clock calibration

The LSI output clock frequency range is $32.8\text{kHz} \pm 10\%$. If the LSI output clock frequency is adjusted outside this range, the clock may be abnormal. When the chip leaves the factory, the calibration parameters of the 32.8kHz frequency point are pre-adjusted and stored in the FLASH. The application program only needs to read the calibration value in the FLASH and write it into SYSCTRL_LSI.TRIM to obtain an accurate 32.8kHz clock. 32.8kHz frequency calibration value storage address: 0x0010 07C2 - 0x0010 07C3. If other frequency clocks are needed, users need to adjust the value of SYSCTRL_LSI.TRIM by themselves.



4.4.3 Clock state detection

The chip has the functions of clock stability detection, start-up failure detection, and clock failure detection during clock startup, and supports automatic clock source switching when the currently selected system clock source fails.

4.4.3.1 Clock stability detection

The 4 clock sources, HSE, LSE, HSIOOSC, and LSI, all support the clock stability detection function. Users can determine the clock status through the stability flag of the corresponding clock source. The clock stable flag is cleared by hardware when the clock source is turned off, and set 1 by hardware after the clock source is enabled and stable.

Note that the clock stable flag is only for the clock start-up process. During the clock stable operation process, the detection of clock operation failure will not affect the clock stable flag.

Taking the HSIOOSC clock source as an example, the HSIOOSC clock stability flag and clock stability interrupt flag are described as follows:

- `SYSCTRL_HSI.STABLE`
Cleared by hardware when HSIOOSC is turned off, and set by hardware when it detects that the HSIOOSC clock is stable.
- `SYSCTRL_ISR.HSISTABLE`
It belongs to the same signal as `SYSCTRL_HSI.STABLE`. Although this signal is located in the interrupt flag register `SYSCTRL_ISR`, it will not cause an interrupt request after being set to 1 by hardware.
- `SYSCTRL_ISR.HSIRDY`
When the hardware detects that the clock changes from an unstable state to a stable state (that is, the `SYSCTRL_HSI.STABLE` flag changes from 0 to 1), it is set to 1, which is the clock stability interrupt flag. Users can clear this flag by setting `SYSCTRL_ICR.HSIRDY` to 0.



4.4.3.2 Clock start-up failure detection

The CW32L012 supports external clock (HSE and LSE) start-up failure detection function. After the external clock is started, the clock detection logic counts the detected clock signals, and if a set number of clock signals are detected within the detection time, the clock starts to vibrate successfully, otherwise the start-up fails.

The number of clocks for HSE start-up failure detection is set through `SYSCTRL_HSE.WAITCYCLE`. The detection time does not need to be set by the user, but is automatically set by the system, as shown in the following table:

Table 4-7 HSE start-up failure detection time and number of detection clocks

<code>SYSCTRL_HSE.WAITCYCLE</code>	Detection time	Number of detection clocks
00	65ms	8192
01	65ms	32768
10	65ms	131072
11	130ms	262144

The number of clocks for LSE start-up failure detection is set through `SYSCTRL_LSE.WAITCYCLE`. The detection time does not need to be set by the user, but is automatically set by the system, as shown in the following table:

Table 4-8 LSE start-up failure detection time and number of detection clocks

<code>SYSCTRL_LSE.WAITCYCLE</code>	Detection time	Number of detection clocks
00	1s	256
01	1s	1024
10	1s	4096
11	2s	16384

The external clock (HSE and LSE) failure detection function is disabled by default, and the corresponding interrupt flag will not be generated if the HSE/LSE fails to start. By setting `SYSCTRL_CR1.HSECCS` to 1 and `SYSCTRL_CR1.LSECCS` to 1 to enable the fault detection function of the external clocks HSE and LSE, respectively.

After enabling the fault detection function of the external clock, if the HSE or LSE fails to start, a start-up failure interrupt flag will be generated (corresponding `SYSCTRL_ISR.HSEFAIL` or `SYSCTRL_ISR.LSEFAIL` is set to 1); if the interrupt is enabled (corresponding `SYSCTRL_IER.HSEFAIL` or `SYSCTRL_IER.LSEFAIL` is set to 1), the CPU will execute the interrupt service routine to handle the failure of clock start-up.

The `LSEEN` can be turned off by software after generating the start-up failure interrupt flag `LSEFAIL`, and then the `LSEFAIL` flag can be cleared by software. If `LSEEN` is not turned off, `LSEFAIL` is generated at intervals as shown in [Table 4-8: LSE start-up failure detection time and number of detection clocks](#).



4.4.3.3 In-flight failure detection of clocks

The CW32L012 supports external clock (HSE or LSE) in-flight failure detection function. During the stable operation of the external clock, the clock detection logic continues to count the HSE and LSE clock signals with a certain detection period: If a set number of clock signals are detected during the detection cycle, the operation is normal, otherwise the operation is disabled.

The in-flight failure detection cycle of HSE clock is configured by `SYSCTRL_HSE.DETCNT`, the actual time is `SYSCTRL_HSE.DETCNT/fLSI`, and the number of detected clocks is 0x20000. Considering that clocks have certain deviation, in order to ensure reliable detection function, HSE detection cycle parameter `HSE.DETCNT` shall be configured with certain margin, which is generally configured as $8000/f_{HSE}$ according to HSE operation frequency (where f_{HSE} is HSE clock frequency in MHz).

Example, if HSE clock frequency is 4MHz, `SYSCTRL_HSE.DETCNT` shall be configured to 2000.

The in-flight failure detection cycle of the LSE clock is not configurable, and is fixed at 256 LSI clock cycles, and the number of detected clocks is 128 LSE.

The external clock LSE failure detection function is disabled by default, and the corresponding interrupt flag will not be generated if the HSE or LSE fails during operation. By setting `SYSCTRL_CR1.HSECCS` to 1 and `SYSCTRL_CR1.LSECCS` to 1, the HSE and LSE clock failure detection function is enabled respectively.

After the failure detection function of the external clock is enabled, if an HSE or LSE failure is detected during operation, a in-flight failure flag will be generated (corresponding `SYSCTRL_ISR.HSEFAULT` or `SYSCTRL_ISR.LSEFAULT` is set to 1), if the interrupt is enabled (corresponding `SYSCTRL_IER.HSEFAULT` or `SYSCTRL_IER.LSEFAULT` is set to 1), the CPU will execute the interrupt service routine for clock in-flight failure process.

After generating the in-flight failure flag `LSEFAULT` it is recommended that software turn off `LSEEN` to clear the `LSESTABLE` state, and then software clears the `LSEFAULT` flag. If `LSEEN` is not turned off, `LSEFAULT` is generated every 256 LSI clock durations (approximately 7.8ms) until LSE returns to normal.

If the current system clock source is HSE, set `SYSCTRL_CR1.CLKCCS` to 0, no action will be taken when HSE fails in-flight; set `SYSCTRL_CR1.CLKCCS` to 1, the system clock source will automatically switch to HSI 4MHz clock when HSE fails in-flight.

If the current system clock source is LSE, set `SYSCTRL_CR1.CLKCCS` to 0, no action will be taken when LSE fails in-flight; set `SYSCTRL_CR1.CLKCCS` to 1, system clock source will automatically switch to HSI 4MHz clock when LSE fails in-flight.



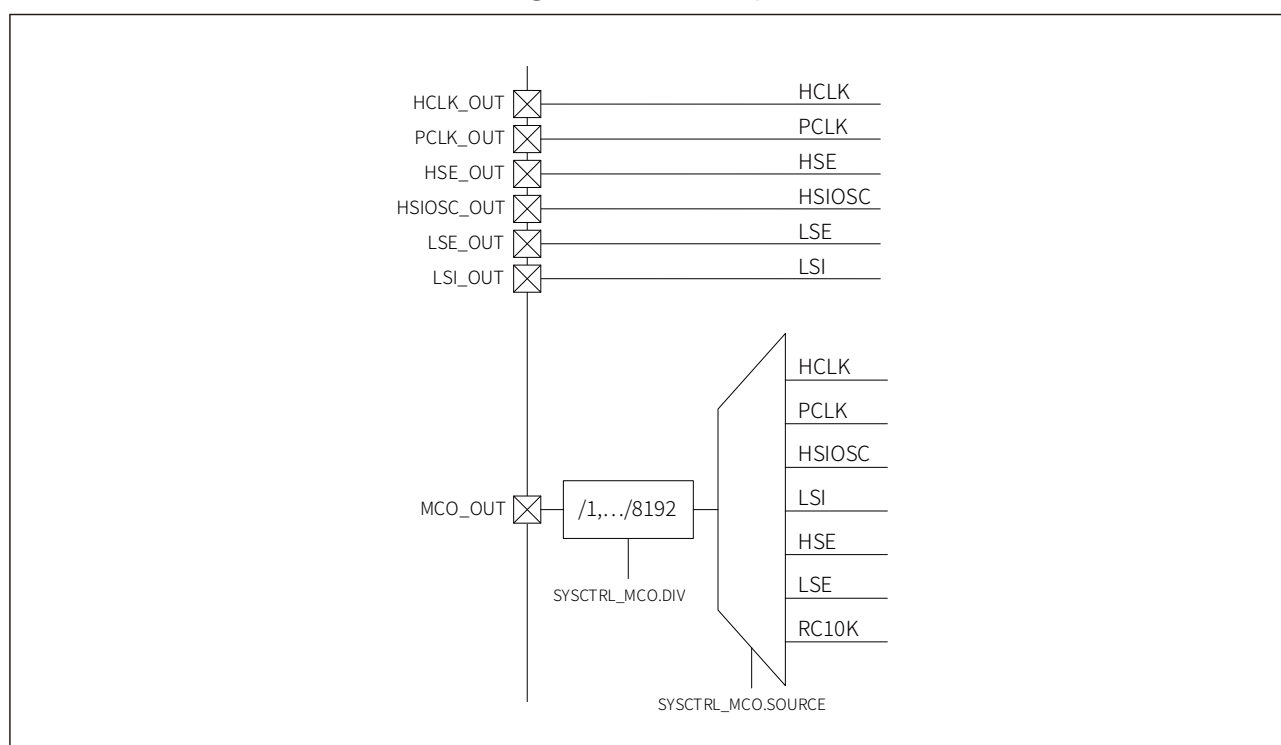
4.4.4 Clock verification and output

The CW32L012 supports outputting various internal clock signals to external pins. Users can use this function to measure the current system's CPU operating frequency, system bus frequency, peripheral operating frequency, etc.

- HCLK_OUT pin: Output HCLK clock signal.
- PCLK_OUT pin: Output PCLK clock signal.
- HSE_OUT pin: Output HSE clock signal.
- HSIOOSC_OUT pin: Output HSIOOSC clock signal.
- LSE_OUT pin: Output LSE clock signal.
- LSI_OUT pin: Output LSI clock signal.
- MCO_OUT pin: Output HCLK/PCLK/HSIOOSC/LSI/HSE/LSE/RC10K clock signal, before the clock is output to the MCO_OUT pin, it can be divided by a prescaler (effective frequency division factors are 1, 2, 4, ..., 8192), so that the low-bandwidth instrument can accurately measure the signal.

The clock output block diagram is shown in the following figure:

Figure 4-5 Clock output



4.5 SysClk system clock switch

The system clock SysClk can select 4 clock sources, including HSE, LSE, HSI, and LSI. By setting the SYSCLK bit field of the system control register SYSCTRL_CR0, it can be switched between different clock sources.

The main application scenarios include:

- Automatic clock failover
When it is detected that the current system clock source fails, it can quickly and safely switch to other available clock sources to improve product reliability.
- Power management
Switch the clock source of SysClk to low-speed clock in standby state to reduce system power consumption, and switch the clock source of SysClk to high-speed clock in normal use state to quickly respond to user needs.

The clock source safety switching rules are as follows:

- Any two of HSE, LSE, HSI and LSI clock sources can be switched to each other.
The system clock switching operation must be performed in accordance with the clock switching procedures described below, otherwise an exception may occur.

Attention: When the system clock is switched, the FLASH control register FLASH_CR2.WAIT needs to be configured synchronously to read the waiting period parameters: if the system clock frequency is not greater than 24MHz, the FLASH control register FLASH_CR2.WAIT should be set to 0; if the system clock frequency is greater than 24MHz and is not greater than 48MHz, the FLASH control register FLASH_CR2.WAIT should be set to 1; if the system clock frequency is greater than 48MHz and is not greater than 72MHz, the FLASH control register FLASH_CR2.WAIT should be set to 2; if the system clock frequency is greater than 72MHz and is not greater than 96MHz, the FLASH control register FLASH_CR2.WAIT should be set to 3; if the system clock frequency is greater than 96MHz, the FLASH control register FLASH_CR2.WAIT should be set to 4.



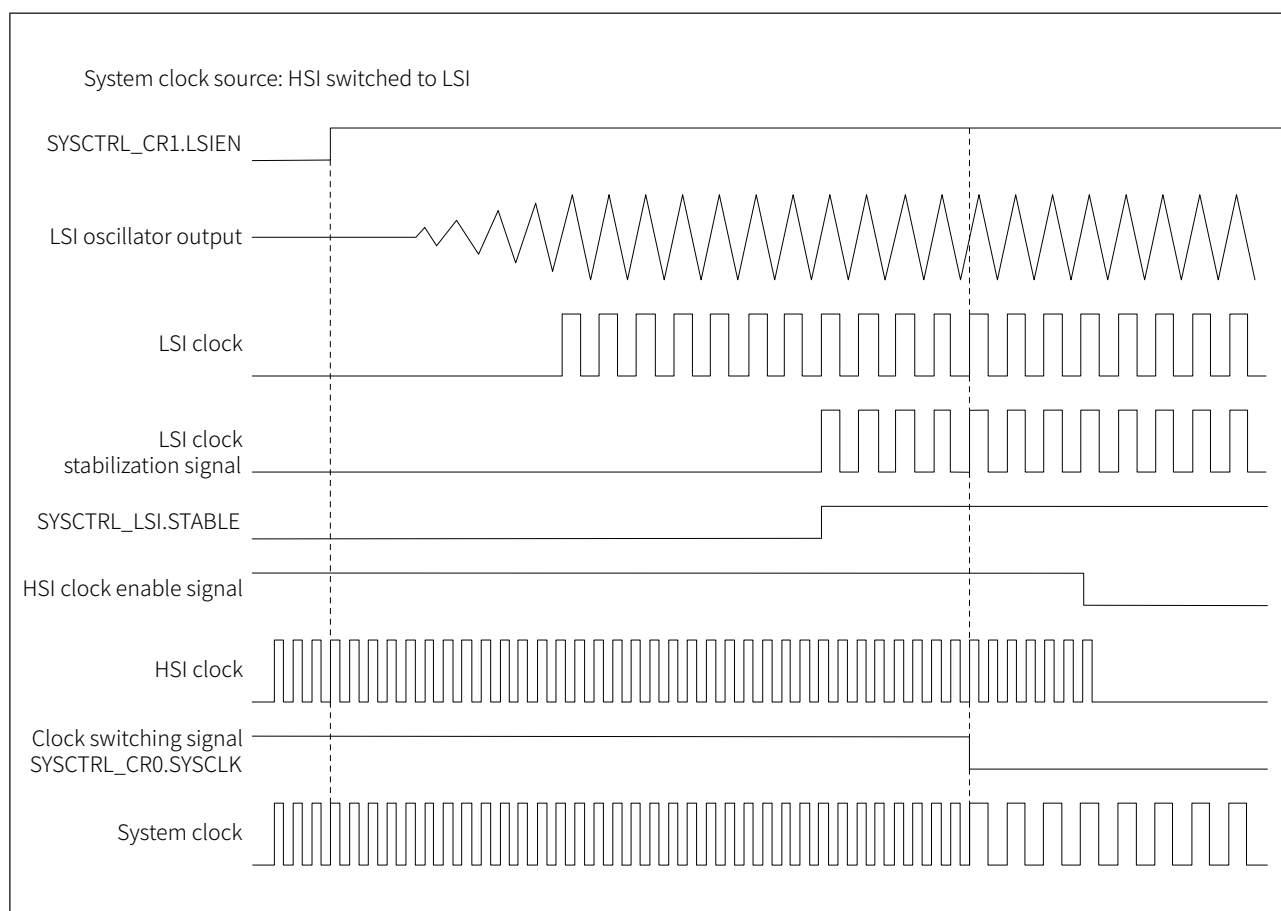
4.5.1 Standard clock switching process

The standard clock switching operation flow is as follows:

- Step 1:** If the new clock source is HSE or LSE, you need to configure the corresponding external pin to the appropriate mode: when connecting an external crystal, you need to configure it as an analog function, when connecting an external clock input, you need to set the GPIO as a digital input function and enable the port's input function.
- Step 2:** Configure the oscillation parameters of the new clock source, such as drive capability, oscillation amplitude, stabilization period, etc..
- Step 3:** Set the new clock source oscillator enable bit to 1.
- Step 4:** According to the higher frequency of the current clock source and the new clock source, configure the FLASH_CR2.WAIT read wait period parameter according to the requirements of the chapter [7 FLASH Memory](#).
- Step 5:** Wait for the new clock source to output a stable frequency (the STABLE signal of the new clock source becomes 1).
- Step 6:** Configure the system control register SYSCTRL_CR0.SYSCLK, and select the clock source of the system clock SysClk as the new clock source.
- Step 7:** According to the frequency of the new clock source, configure the FLASH_CR2.WAIT read waiting period parameter according to the requirements of the chapter [7 FLASH Memory](#).
- Step 8:** Turn off clock sources that are no longer in use.

Taking the system clock switching from HSI to LSI as an example, the switching process of the old and new clocks is shown in the following figure:

Figure 4-6 HSI clock switched to LSI clock



4.5.2 HSI clock switching process between different frequencies

Switching between different frequencies of the HSI clock does not require starting a new clock source, but only needs to change the frequency division ratio of the HSI clock prescaler. The switching speed is fast, and there is no need to wait for the clock stabilization time before and after switching. The switching operation process is as follows:

- Step 1:** According to the higher frequency of the current clock source and the new clock source, configure the FLASH_CR2.WAIT read wait period parameter according to the requirements of the chapter [7 FLASH Memory](#).
- Step 2:** Read the value of the built-in high-frequency clock control register SYSCTRL_HSI register as Value.
- Step 3:** According to the required target frequency, set the built-in high-frequency clock control register SYSCTRL_HSI.DIV to the corresponding frequency division coefficient (**Note that the SYSCTRL_HSI.TRIM field should remain unchanged.**).
- Step 4:** According to the frequency of the new clock source, configure the FLASH_CR2.WAIT read wait period parameter as required in the chapter [7 FLASH Memory](#).

4.5.3 Example of switching from other clocks to LSE

The switching operation process is as follows:

- Step 1:** Set PC14/PC15 pins as analog function ports.
- Step 2:** According to the characteristics of the external crystal of the LSE oscillator, configure the oscillation parameters of the new clock source LSE, such as drive capability (set by the external low-frequency crystal control register SYSCTRL_LSE.DRIVER), stable period (set through the external low-frequency crystal control register SYSCTRL_LSE.WAITCYCLE, it is recommended to set to 3), etc..
- Step 3:** Set the system control register SYSCTRL_CR1.LSEEN to 1 to enable the LSE crystal oscillator circuit.
Attention: The SYSCTRL_CR1 register has the KEY protection feature. The upper 16-bit data of the written data must be 0x5A5A, otherwise it cannot be written.
- Step 4:** Cyclic query and wait for the external low-frequency crystal control register SYSCTRL_LSE.STABLE stable flag to become 1, that is, wait for the LSE crystal oscillator circuit to output a stable clock.
- Step 5:** Set the system control register SYSCTRL_CR0.SYSCLK to 4, and switch the SysClk clock source to LSE.
Attention: The SYSCTRL_CR0 register has the KEY protection feature. The upper 16-bit data of the written data must be 0x5A5A, otherwise it cannot be written.
- Step 6:** Configure the FLASH_CR2.WAIT read wait period parameter according to the requirements of the chapter [7 FLASH Memory](#).
- Step 7:** Set the system control register SYSCTRL_CR1.xxxEN to 0 to turn off the original clock.



4.5.4 Example of switching from other clocks to HSE

The switching operation process is as follows:

- Step 1: Set PB07/PC13 pins as analog function ports.
- Step 2: According to the characteristics of the external crystal of the HSE oscillator, configure the oscillation parameters of the new clock source HSE, such as drive capability (through the external high-frequency crystal control register `SYSCTRL_HSE.DRIVER` configuration), stabilization period (configured through the external high-frequency crystal control register `SYSCTRL_HSE.WAITCYCLE`, it is recommended to set it to 3), etc..
- Step 3: Set the system control register `SYSCTRL_CR1.HSEEN` to 1 to enable the HSE crystal oscillator circuit.
Attention: The `SYSCTRL_CR1` register has the KEY protection feature. The upper 16-bit data of the written data must be 0x5A5A, otherwise it cannot be written.
- Step 4: According to the higher frequency of the current clock and HSE, configure the `FLASH_CR2.WAIT` read waiting period parameter according to the requirements of the chapter [7 FLASH Memory](#).
- Step 5: Cyclic query in a loop and wait for the stable flag of the external high-frequency crystal control register `SYSCTRL_HSE.STABLE` to become 1 to ensure that the crystal oscillator circuit outputs a stable clock.
- Step 6: Set the system control register `SYSCTRL_CR0.SYSCLK` to 1, and switch the SysClk clock source to HSE.
Attention: The `SYSCTRL_CR0` register has the KEY protection feature. The upper 16-bit data of the written data must be 0x5A5A, otherwise it cannot be written.
- Step 7: According to the clock frequency of the HSE, configure the `FLASH_CR2.WAIT` read waiting period parameter according to the requirements of the chapter [7 FLASH Memory](#).
- Step 8: Set the system control register `SYSCTRL_CR1.xxxEN` to 0 to turn off the original clock.

4.5.5 Example of switching from other clocks to LSI

The switching operation process is as follows:

- Step 1: Configure the built-in low-frequency clock control register `SYSCTRL_LSI.TRIM` and the built-in low-frequency clock control register `SYSCTRL_LSI.WAITCYCLE` to appropriate values.
- Step 2: Set the system control register `SYSCTRL_CR1.LSIEN` to 1 to enable the LSIRC oscillator circuit.
Attention: The `SYSCTRL_CR1` register has the KEY protection feature. The upper 16-bit data of the written data must be 0x5A5A, otherwise it cannot be written.
- Step 3: Cyclic query and wait for the built-in low-frequency clock control register `SYSCTRL_LSI.STABLE` stable flag to become 1, that is, wait for the LSI to output a stable clock.
- Step 4: Set the system control register `SYSCTRL_CR0.SYSCLK` to 3, and switch the SysClk clock source to LSI.
Attention: The `SYSCTRL_CR0` register has the KEY protection feature. The upper 16-bit data of the written data must be 0x5A5A, otherwise it cannot be written.
- Step 5: Configure the `FLASH_CR2.WAIT` read wait period parameter to 0.
- Step 6: Set the system control register `SYSCTRL_CR1.xxxEN` to 0 to turn off the original clock.



4.5.6 Example of switching from other clocks to HSI

The switching operation process is as follows:

- Step 1:** Configure the built-in high-frequency clock control registers `SYSCTRL_HSI.TRIM`, `SYSCTRL_HSI.WAITCYCLE` and `SYSCTRL_HSI.DIV` to appropriate values.
- Step 2:** Set the system control register `SYSCTRL_CR1.HSIEN` to 1 to enable the `HSIOSC` clock oscillation circuit.
Attention: The `SYSCTRL_CR1` register has the KEY protection feature. The upper 16-bit data of the written data must be 0x5A5A, otherwise it cannot be written.
- Step 3:** Cyclic query and wait for the built-in high-frequency clock control register `SYSCTRL_HSI.STABLE` stable flag to become 1, that is, wait for the `HSIOSC` to output a stable clock.
- Step 4:** According to the higher frequency of the current clock and HSI, configure the `FLASH_CR2.WAIT` read waiting period parameter according to the requirements of the chapter [7 FLASH Memory](#).
- Step 5:** Set the system control register `SYSCTRL_CR0.SYSCLK` to 0, and switch the `SysClk` clock source to HSI.
Attention: The `SYSCTRL_CR0` register has the KEY protection feature. The upper 16-bit data of the written data must be 0x5A5A, otherwise it cannot be written.
- Step 6:** According to the clock frequency of HSI, configure the `FLASH_CR2.WAIT` read waiting period parameter according to the requirements of the chapter [7 FLASH Memory](#).
- Step 7:** Set the system control register `SYSCTRL_CR1.xxxEN` to 0 to turn off the original clock.



4.6 List of registers

SYSCTRL base address: SYSCTRL_BASE = 0x4000 4000

Table 4-9 List of SYSCTRL registers

Register name	Register address	Register description
SYSCTRL_CR0	SYSCTRL_BASE + 0x00	System control register 0
SYSCTRL_CR1	SYSCTRL_BASE + 0x04	System control register 1
SYSCTRL_CR2	SYSCTRL_BASE + 0x08	System control register 2
SYSCTRL_IER	SYSCTRL_BASE + 0x0C	System interrupt enable control register
SYSCTRL_ISR	SYSCTRL_BASE + 0x10	System interrupt flag register
SYSCTRL_ICR	SYSCTRL_BASE + 0x14	System interrupt flag clear register
SYSCTRL_HSI	SYSCTRL_BASE + 0x18	Internal high frequency clock control register
SYSCTRL_HSE	SYSCTRL_BASE + 0x1C	External high frequency crystal control register
SYSCTRL_LSI	SYSCTRL_BASE + 0x20	Internal low frequency clock control register
SYSCTRL_LSE	SYSCTRL_BASE + 0x24	External low frequency crystal control register
SYSCTRL_DEBUG	SYSCTRL_BASE + 0x2C	Debug status timer control register
SYSCTRL_AHBEN	SYSCTRL_BASE + 0x30	AHB peripheral clock enable control register
SYSCTRL_APBEN2	SYSCTRL_BASE + 0x34	APB peripheral clock enable control register 2
SYSCTRL_APBEN1	SYSCTRL_BASE + 0x38	APB peripheral clock enable control register 1
SYSCTRL_AHBRST	SYSCTRL_BASE + 0x40	AHB peripheral reset control register
SYSCTRL_APBRS2	SYSCTRL_BASE + 0x44	APB peripheral reset control register 2
SYSCTRL_APBRS1	SYSCTRL_BASE + 0x48	APB peripheral reset control register 1
SYSCTRL_RESETFLAG	SYSCTRL_BASE + 0x4C	System reset flag register
SYSCTRL_MCO	SYSCTRL_BASE + 0x70	System clock output control register



4.7 Register description

See section 1 [Documentation conventions](#) for abbreviations used in register descriptions.

4.7.1 SYSCTRL_CR0 System control register 0

Address offset: 0x00 Reset value: 0x0000 0000

Bit field	Name	Permission	Function description
31:16	KEY	WO	Only when KEY is 0x5A5A, the write operation of this register is valid.
15:8	RFU	-	Reserved bits, please keep the default value.
7:5	HCLKPRS	RW	Configure the clock source of HCLK 000: Set the clock source of HCLK to SysClk. 001: Set the clock source of HCLK to SysClk/2. 010: Set the clock source of HCLK to SysClk/4. 011: Set the clock source of HCLK to SysClk/8. 100: Set the clock source of HCLK to SysClk/16. 101: Set the clock source of HCLK to SysClk/32. 110: Set the clock source of HCLK to SysClk/64. 111: Set the clock source of HCLK to SysClk/128.
4:3	PCLKPRS	RW	Configure the clock source of PCLK 00: Set the clock source of PCLK to HCLK. 01: Set the clock source of PCLK to HCLK/2. 10: Set the clock source of PCLK to HCLK/4. 11: Set the clock source of PCLK to HCLK/8.
2:0	SYSCLK	RW	Configure the clock source of SysClk 000: Set the clock source of SysClk to HSI. 001: Set the clock source of SysClk to HSE. 011: Set the clock source of SysClk to LSI. 100: Set the clock source of SysClk to LSE.



4.7.2 SYSCTRL_CR1 System control register 1

Address offset: 0x04

Reset value: 0x0000 0001

Bit field	Name	Permission	Function description
31:16	KEY	WO	Only when KEY is 0x5A5A, the write operation of this register is valid.
15:9	RFU	-	Reserved bits, please keep the default value.
8	CLKCCS	RW	Configure the crystal oscillator for safe operation in the event of a failure in operation 0: No action. 1: Automatically switch the clock source of SysClk to HSI 4MHz.
7	HSECCS	RW	High frequency crystal oscillator clock monitor enable configuration 0: Disable HSE clock monitor. 1: Enable HSE clock monitor. <i>Note: During operation, if the number of clocks output by the HSE is less than 0x20000 clocks per HSE.DETCNT LSI clock durations, the HSE is judged to be disabled during runtime.</i> <i>Note: During startup, if the condition specified by HSE.WAITCYCLE is not satisfied, the HSE is judged to startup failure.</i> <i>Note: The power consumption of DeepSleep mode will be increased by about 1μA when this function is enabled.</i>
6	LSECCS	RW	Low frequency crystal oscillator clock monitor enable configuration 0: Disable LSE clock monitor. 1: Enable LSE clock monitor. <i>Note: During operation, if the number of clocks output by the LSE is less than 128 within the duration of every 256 LSI clocks (about 7.8ms), the LSE is judged to be disabled during runtime.</i> <i>Note: During startup, if the conditions specified by LSE.WAITCYCLE are not met, it is determined that the LSE has failed to start vibrating.</i> <i>Note: Enable this feature and the power consumption of DeepSleep mode will increase by about 1μA.</i>
5	LSELOCK	RW	External low-speed clock LSE lock control 0: LSEEN can be set and cleared. 1: LSEEN only settable. <i>Note: The bit field can be reset only by a power-on reset after enabled.</i> <i>Note: This function does not work when the LSEFAIL or LSEFAULT flag is generated, i.e., at this time LSEEN can be set or cleared until the above flags are cleared by the user software.</i>
4	LSEEN	RW	External low-speed clock LSE enable control 0: Disabled 1: Enabled <i>Note: When the system enters DeepSleep, this low-speed clock will not automatically shut down.</i> <i>Note: When LSELOCK = 0, LSEEN can be turned off by software, other than that it can only be reset by power-on reset; when LSELOCK = 1, only power-on reset can reset LSEEN.</i>



Bit field	Name	Permission	Function description
3	LSIEN	RW	Internal low-speed clock LSI enable control 0: Disabled 1: Enabled <i>Note: When the system enters DeepSleep, this low-speed clock will not automatically shut down.</i>
2	RFU	-	Reserved bits, please keep the default value.
1	HSEEN	RW	External high-speed clock HSE enable control 0: Disabled 1: Enabled <i>Note: When the system enters DeepSleep, this high-speed clock will be automatically turned off.</i>
0	HSIEN	RW	Internal high-speed clock HSIOSC enable control 0: Disabled 1: Enabled <i>Note: When the system enters DeepSleep, this high-speed clock will be automatically turned off.</i>

Attention: The LSI clock can be turned on automatically by hardware in the following ways, but the SYSCTRL_CR1.LSIEN bit field will not be set. This bit field can only be set by software writing SYSCTRL_CR1.LSIEN to 1:

1. *SYSCTRL_CR1.HSECCS = 1.*
2. *SYSCTRL_CR1.LSECCS = 1.*
3. *GPIOx_FILTER.FLTCLK = 0x5 (LSI).*
4. *VCx_CR1.FLTCLK = 0x0 (LSI) and VCx_CR0.EN = 1.*
5. *LVD_CR0.FLTCLK = 0x0 (LSI) and LVD_CR0.EN = 1.*
6. *IWDG_KR = 0xCCCC (enable IWDG).*



4.7.3 SYSCTRL_CR2 System control register 2

Address offset: 0x08 Reset value: 0x0000 0C00

Bit field	Name	Permission	Function description
31:16	KEY	WO	Only when KEY is 0x5A5A, the write operation of this register is valid.
15	RAMBRKEN	RW	RAM parity error triggers ATIM brake enable control 0: Disabled 1: Enabled
14	LVDBRKEN	RW	LVD output triggers ATIM brake enable control 0: Disabled 1: Enabled
13	DSBRKEN	RW	DeepSleep triggers ATIM brake enable control 0: Disabled 1: Enabled
12	CLLBRKEN	RW	Generate HardFault or Cortex-M0+ LockUp flag to trigger ATIM brake enable control 0: Disabled 1: Enabled
11	LSEBRKEN	RW	LSE in-flight failure triggers ATIM brake enable control 0: Disabled 1: Enabled
10	HSEBRKEN	RW	HSE failure in-flight triggers ATIM brake enable control 0: Disabled 1: Enabled
9:7	RFU	-	Reserved bits, please keep the default value.
6:4	FLASHWAIT	RW	FLASH instruction fetches cycle configuration 000: 1 HCLK cycle for $HCLK \leq 24\text{MHz}$. 001: 2 HCLK cycle for $24\text{MHz} < HCLK \leq 48\text{MHz}$. 010: 3 HCLK cycle for $48\text{MHz} < HCLK \leq 72\text{MHz}$. 011: 4 HCLK cycle for $72\text{MHz} < HCLK \leq 96\text{MHz}$. <i>Note: This control bit has the same function as FLASH_CR2[2:0].</i>
3	WAKEUPCLK	RW	When DeepSleep wakes up, the source configuration of the system clock 0: Keep the original system clock source. 1: Switch the system clock source to HSI 4MHz, keep the original system clock source enabled.
2	LOCKUP	RW	Cortex-M0+ LockUp Functional configuration 0: Disabled 1: Enabled <i>Note: When this function is enabled, the CPU will reset the MCU when it reads an invalid instruction.</i>



Bit field	Name	Permission	Function description
1	SWDIO	RW	SWD Port function configuration 0: PA13/PA14 are used as SWD ports, GPIO function is not available. 1: PA13/PA14 are used as GPIO ports, SWD function is not available.
0	RFU	-	Reserved bits, please keep the default value.

4.7.4 SYSCTRL_HSI Internal high frequency clock control register

Address offset: 0x18 Reset value: 0x0001 F772

Bit field	Name	Permission	Function description
31:16	RFU	-	Reserved bits, please keep the default value.
15	STABLE	RO	HSIOSC clock stable status bits 0: HSIOSC clock has not stabilized 1: HSIOSC clock has stabilized
14:11	DIV	RW	HSI clock and HSIOSC clock division factor configuration 0000: HSI = HSIOSC/32 0001: HSI = HSIOSC/1 0010: HSI = HSIOSC/2 0011: HSI = HSIOSC/3 0100: HSI = HSIOSC/4 0101: HSI = HSIOSC/5 0110: HSI = HSIOSC/6 0111: HSI = HSIOSC/7 1000: HSI = HSIOSC/8 1001: HSI = HSIOSC/9 1010: HSI = HSIOSC/10 1011: HSI = HSIOSC/12 1100: HSI = HSIOSC/16 1101: HSI = HSIOSC/20 1110: HSI = HSIOSC/24 (Default) 1111: HSI = HSIOSC/28
10:0	TRIM	RW	Clock frequency adjustment, changing the value of this register bit can adjust the oscillation frequency of the HSIOSC. Every time the TRIM value increases by 1, the oscillation frequency of the HSIOSC increases by about 0.2%, and the total adjustment range is 90~100MHz. The calibration value of 96MHz has been saved in FLASH, and the accurate frequency can be obtained by reading the calibration value in FLASH and writing it into SYSCTRL_HSI.TRIM, the hardware default value is 0x172. 96MHz calibration value address: 0x0010 07C0 – 0x0010 07C1



4.7.5 SYSCTRL_LSI Internal low frequency clock control register

Address offset: 0x20 Reset value: 0x---- ----

Bit field	Name	Permission	Function description
31:16	RFU	-	Reserved bits, please keep the default value.
15	STABLE	RO	LSI clock stable status bit 0: LSI clock has not stabilized 1: LSI clock has stabilized
14:12	RFU	-	Reserved bits, please keep the default value.
11:10	WAITCYCLE	RW	Internal low-speed clock LSI stabilization time selection 00: 6 cycles 01: 18 cycles 10: 66 cycles 11: 258 cycles
9	RFU	-	Reserved bits, please keep the default value.
8:0	TRIM	RW	Clock frequency adjustment, change the value of this register bit to adjust the oscillation frequency of LSI. Every increase of TRIM value by 1 will increase the oscillation frequency of LSI by about 0.4%, the total adjustment range is 30kHz~36kHz. 1 set of frequency calibration value has been saved in FLASH, read the preset calibration value from FLASH, write it to this bit field to get the accurate LSI frequency. 32.8kHz calibration value address: 0x0010 07C2 - 0x0010 07C3.



4.7.6 SYSCTRL_HSE External high frequency crystal control register

Address offset: 0x1C Reset value: 0x0027 FF22

Bit field	Name	Permission	Function description
31:25	RFU	-	Reserved bits, please keep the default value.
24	DIGFLT	RW	External high-speed clock HSE digital filter control 0: Disable HSE digital filter function. 1: Enable the HSE digital filter function, the HSE frequency should not be greater than 8MHz. <i>Note: It is recommended to enable the filtering function only when the interference signal is extremely strong.</i> <i>Note: When HCLK source is HSE, it is forbidden to modify this control bit.</i>
23:20	PDRIVER	RW	HSE crystal drive capability configuration before HSE clock stabilizes; see DRIVER bit field description.
19	STABLE	RO	HSE clock stable status bit 0: HSE clock has not been stabilized. 1: HSE clock has been stabilized.
18:8	DETCNT	RW	HSE in-flight failure detection cycle configuration In consideration of clock errors, the DETCNT should generally be set to a value of $8000/f_{HSE}$. Example: HSE is 4MHz, then DETCNT should be configured to 2000. <i>Note: DETCNT must be greater than 0.</i>
7	HEXENPOL	RW	Output to external oscillator enable signal polarity configuration 0: Output high level in Active mode, output low level in DeepSleep mode. 1: Output low level in Active mode, output high level in DeepSleep mode. <i>Note: It is necessary to configure the corresponding pin as HEXEN multiplexed output.</i>
6	MODE	RW	External high-speed clock HSE operating mode configuration 0: Crystal mode, whose output clock is generated by the crystal resonating with the internal circuit. 1: Input mode, whose output clock comes from the OSC_IN pin.
5:4	WAITCYCLE	RW	HSE stability detection and startup failure detection clock cycle number selection 00: Wait for 8192 HSE clock signals. 01: Wait for 32768 HSE clock signals. 10: Wait for 131072 HSE clock signals. 11: Wait for 262144 HSE clock signals. <i>Note: When HSECCS is enabled, the startup failure detection circuit will detect the number of HSE output clock signals within a limited time, and the four limit times are 65ms, 65ms, 65ms, and 130ms respectively.</i>



Bit field	Name	Permission	Function description
3:0	DRIVER	RW	HSE crystal drive capability configuration 0000: Minimum drive capability. ... 0111: Maximum drive capability. <i>Note: You need to select the appropriate drive capability according to the crystal characteristics, load capacitance, and parasitic parameters of the board. The stronger the drive capability, the higher the power consumption; the weaker the drive capability, the lower the power consumption.</i>



4.7.7 SYSCTRL_LSE External low frequency crystal control register

Address offset: 0x24 Reset value: 0x0000 0A22

Bit field	Name	Permission	Function description
31:19	RFU	-	Reserved bits, please keep the default value.
18	STABLE	RO	LSE clock stable status bit 0: LSE clock has not stabilized. 1: LSE clock has stabilized.
17	PINLOCK	RW	External low-speed clock LSE pin locked to analog state 0: LSE pin rewritable. 1: LSE pin locked to analog state. <i>Attention: Once enabled, only power-on reset can reset this bit field.</i> <i>Attention: This function takes effect only when SYSCTRL_CR1.LSELOCK is 1.</i> <i>Attention: This function does not work when the LSEFAIL or LSEFAULT flag is generated, i.e., at this time LSEEN can be set or cleared until the above flags are cleared by the user software.</i>
16:12	RFU	-	Reserved bits, please keep the default value.
11:8	PDRIVER	RW	Configuration of LSE crystal drive capability before LSE clock stabilizes; see DRIVER bit-field description.
7	RFU	-	Reserved bits, please keep the default value.
6	MODE	RW	External low-speed clock LSE operating mode configuration 0: Crystal mode, whose output clock is generated by the crystal resonating with the internal circuit. 1: Input mode, whose output clock comes from the OSC32_IN pin.
5:4	WAITCYCLE	RW	Selection of the number of clock cycles for LSE stability detection and start-up failure detection 00: Waiting for 256 LSE clock signals. 01: Waiting for 1024 LSE clock signals. 10: Waiting for 4096 LSE clock signals. 11: Waiting for 16384 LSE clock signals. <i>Note: When LSECCS is enabled, the start-up failure detection circuit will detect the number of LSE clock signals within a limited time. The four-stage limited time is 1s, 1s, 1s, and 2s respectively.</i>
3:0	DRIVER	RW	LSE crystal drive capability configuration 0000: Minimum drive capability. ... 1111: Maximum drive capability. <i>Note: The appropriate drive capability needs to be selected based on the crystal characteristics, load capacitance, and board parasitics. The stronger the driving ability, the greater the power consumption, and the weaker the driving ability, the smaller the power consumption.</i>

Attention: Only a power-on reset can reset this register.



4.7.8 SYSCTRL_IER System interrupt enable control register

Address offset: 0x0C

Reset value: 0x0000 0000

Bit field	Name	Permission	Function description
31:16	KEY	WO	Only when KEY is 0x5A5A, the write operation of this register is valid.
15:9	RFU	-	Reserved bits, please keep the default value.
8	HSEFAULT	RW	HSE operation fault interrupt enable control 0: Disable HSE operation fault interrupt. 1: Enable HSE operation fault interrupt.
7	LSEFAULT	RW	LSE Fail-Run Interrupt Enable Control 0: Disable LSE run-fail interrupt. 1: Enable LSE run-fail interrupt.
6	HSEFAIL	RW	HSE startup failure interrupt enable control 0: Disable HSE startup failure interrupt. 1: Enable HSE startup failure interrupt.
5	LSEFAIL	RW	LSE startup failure interrupt enable control 0: Disable LSE start failure interrupt. 1: Enable LSE start failure interrupt.
4	LSERDY	RW	LSE stable interrupt enable control 0: Disable LSE stable interrupt. 1: Enable LSE stable interrupt.
3	LSIRDY	RW	LSI stable interrupt enable control 0: Disable LSI stable interrupt. 1: Enable LSI stable interrupt.
2	RFU	-	Reserved bits, please keep the default value.
1	HSERDY	RW	HSE stable interrupt enable control 0: Disable HSE stable interrupt. 1: Enable HSE stable interrupt.
0	HSIRDY	RW	HSIOSC Stable Interrupt Enable Control 0: Disable HSIOSC stable interrupt. 1: Enable HSIOSC stable interrupt.

Note: The FAULT interrupt (vector number 47) is executed when HSEFAULT interrupts and LSEFAULT interrupts are generated, and the RCC global interrupt (vector number 20) is executed when the other interrupt is generated.



4.7.9 SYSCTRL_ISR System interrupt flag register

Address offset: 0x10 Reset value: 0x0000 0801

Bit field	Name	Permission	Function description
31:16	RFU	-	Reserved bits, please keep the default value.
15	LSESTABLE	RO	LSE clock stable status bit; hardware set to clear 0: LSE clock has not stabilized. 1: LSE clock has stabilized.
14	LSISTABLE	RO	LSI clock stable status bit; hardware set to clear 0: LSI clock has not stabilized. 1: LSI clock has stabilized.
13	RFU	-	Reserved bits, please keep the default value.
12	HSESTABLE	RO	HSE clock stable status bit; hardware set to clear 0: HSE clock has not stabilized. 1: HSE clock has stabilized.
11	HSISTABLE	RO	HSIOSC clock stable status bit; hardware set to clear 0: HSIOSC clock has not stabilized. 1: HSIOSC clock has stabilized.
10:9	RFU	-	Reserved bits, please keep the default value.
8	HSEFAULT	RO	HSE clock operation failure interrupt flag bit 0: HSE clock is running normally. 1: Oscillation stopped during HSE clock operation.
7	LSEFAULT	RO	LSE Clock run failure interrupt flag 0: LSE clock is running fine. 1: Oscillation stopped during LSE clock operation. <i>Note: After generating LSEFAULT it is recommended that software turn off LSEEN to clear the LSESTABLE state and then software clear the LSEFAULT flag. If LSEEN is not turned off, LSEFAULT will be generated every 256 LSI clock durations (approximately 7.8ms) until the LSE returns to normal.</i>
6	HSEFAIL	RO	HSE clock start-up failure interrupt flag 0: HSE clock starts up successfully. 1: HSE clock start-up failed.
5	LSEFAIL	RO	LSE clock start-up failure interrupt flag 0: LSE Clock starts up successfully. 1: LSE clock start-up failed. <i>Note: After generating LSEFAIL, LSEEN can be turned off by software and then the LSEFAIL flag can be cleared by software. If LSEEN is not turned off, LSEFAIL is generated at intervals described in the WAITCYCLE bit field of the SYSCTRL_LSE register.</i>
4	LSERDY	RO	LSE clock stable interrupt flag 0: LSE clock has not stabilized. 1: LSE clock has stabilized. <i>Note: Clearing ICR.LSEFAULT, ICR.LSEFAIL, or ICR.LSERDY can clear the LSERDY flag synchronously.</i>



Bit field	Name	Permission	Function description
3	LSIRDY	RO	LSI clock stable interrupt flag 0: LSI clock has not stabilized. 1: LSI clock has stabilized.
2	RFU	-	Reserved bits, please keep the default value.
1	HSERDY	RO	HSE clock stable interrupt flag 0: HSE clock has not stabilized. 1: HSE clock has stabilized. <i>Note: Clearing ICR.HSEFAULT, ICR.HSEFAIL or ICR.HSERDY can clear the HSERDY flag synchronously.</i>
0	HSIRDY	RO	HSIOSC clock stable interrupt flag 0: HSIOSC clock has not stabilized. 1: HSIOSC clock has stabilized.

Note: The FAULT interrupt (vector number 47) is executed when the HSEFAULT interrupt and LSEFAULT interrupt are generated, and the RCC global interrupt (vector number 20) is executed when other interrupts are generated.

Note: The STABLE signal of the clock is set by the hardware and cleared by the hardware, and the RDY signal is set by the hardware and cleared by the software writing the corresponding ICR register bit field.



4.7.10 SYSCTRL_ICR System interrupt flag clear register

Address offset: 0x14 Reset value: 0x0000 00FF

Bit field	Name	Permission	Function description
31:9	RFU	-	Reserved bits, please keep the default value.
8	HSEFAULT	R1W0	HSE clock operation failure interrupt flag clear control W0: Clear the corresponding flag in the ISR register. W1: No function. <i>Note: Clearing HSEFAULT flag will clear the HSERDY flag synchronously.</i>
7	LSEFAULT	R1W0	LSE clock operation failure interrupt flag clear control W0: Clear the corresponding flag in the ISR register. W1: No function. <i>Note: Clearing LSEFAULT flag will clear the LSERDY flag synchronously.</i>
6	HSEFAIL	R1W0	HSE clock start-up failure interrupt flag clear control W0: Clear the corresponding flag in the ISR register. W1: No function. <i>Note: Clearing HSEFAIL flag will clear the HSERDY flag synchronously.</i>
5	LSEFAIL	R1W0	LSE clock start-up failure interrupt flag clear control W0: Clear the corresponding flag in the ISR register. W1: No function. <i>Note: Clearing LSEFAIL flag will clear the LSERDY flag synchronously.</i>
4	LSERDY	R1W0	LSE clock stable interrupt flag clear control W0: Clear the corresponding flag in the ISR register. W1: No function.
3	LSIRDY	R1W0	LSI clock stable interrupt flag clear control W0: Clear the corresponding flag in the ISR register. W1: No function.
2	RFU	-	Reserved bits, please keep the default value.
1	HSERDY	R1W0	HSE clock stable flag clear control W0: Clear the corresponding flag in the ISR register. W1: No function.
0	HSIRDY	R1W0	HSIOSC clock stable interrupt flag clear control W0: Clear the corresponding flag in the ISR register. W1: No function.



4.7.11 SYSCTRL_AHBEN AHB peripheral clock enable control register

Address offset: 0x30

Reset value: 0x0000 0000

Bit field	Name	Permission	Function description
31:16	KEY	WO	Only when KEY is 0x5A5A, the write operation of this register is valid.
15:9	RFU	-	Reserved bits, please keep the default value.
8	CORDIC	RW	CORDIC module configuration clock and working clock enable control 0: Disabled 1: Enabled
7	GPIOF	RW	GPIOF port configuration clock and working clock enable control 0: Disabled 1: Enabled
6	GPIOC	RW	GPIOC port configuration clock and working clock enable control 0: Disabled 1: Enabled
5	GPIOB	RW	GPIOB port configuration clock and working clock enable control 0: Disabled 1: Enabled
4	GPIOA	RW	GPIOA port configuration clock and working clock enable control 0: Disabled 1: Enabled
3	EAU	RW	EAU module configuration clock and working clock enable control 0: Disabled 1: Enabled
2	CRC	RW	CRC module configuration clock and working clock enable 0: Disabled 1: Enabled
1	FLASH	RW	FLASH module configuration clock enable control 0: Disabled 1: Enabled
0	DMA	RW	DMA module configuration clock and working clock enable control 0: Disabled 1: Enabled



4.7.12 SYSCTRL_APBEN1 APB peripheral clock enable control register 1

Address offset: 0x38

Reset value: 0x0000 0000

Bit field	Name	Permission	Function description
31:16	KEY	WO	Only when KEY is 0x5A5A, the write operation of this register is valid.
15	RFU	-	Reserved bits, please keep the default value.
14	SPI3	RW	SPI3 module configuration clock and working clock enable control 0: Disabled 1: Enabled
13	SPI2	RW	SPI2 module configuration clock and working clock enable control 0: Disabled 1: Enabled
12	GTIM4	RW	GTIM4 module configuration clock and working clock enable control 0: Disabled 1: Enabled
11	GTIM3	RW	GTIM3 module configuration clock and working clock enable control 0: Disabled 1: Enabled
10:9	RFU	-	Reserved bits, please keep the default value.
8	UART3	RW	UART3 module configuration clock and working clock enable control 0: Disabled 1: Enabled
7	GTIM2	RW	GTIM2 module configuration clock and working clock enable control 0: Disabled 1: Enabled
6	GTIM1	RW	GTIM1 module configuration clock and working clock enable control 0: Disabled 1: Enabled
5	ATIM	RW	ATIM module configuration clock and working clock enable control 0: Disabled 1: Enabled
4	UART2	RW	UART2 module configuration clock enable control 0: Disabled 1: Enabled
3	UART1	RW	UART1 module configuration clock enable control 0: Disabled 1: Enabled
2	SPI1	RW	SPI1 module configuration clock and working clock enable control 0: Disabled 1: Enabled



Bit field	Name	Permission	Function description
1	VC	RW	VC module configuration clock enable control 0: Disabled 1: Enabled
0	ADC	RW	ADC module configuration clock and working clock enable control 0: Disabled 1: Enabled



4.7.13 SYSCTRL_APBEN2 APB peripheral clock enable control register 2

Address offset: 0x34 Reset value: 0x0000 0100

Bit field	Name	Permission	Function description
31:16	KEY	WO	Only when KEY is 0x5A5A, the write operation of this register is valid.
15:13	RFU	-	Reserved bits, please keep the default value.
12	HALLTIM	RW	HALLTIM module configuration clock and working clock enable control 0: Disabled 1: Enabled
11	I2C2	RW	I2C2 module configuration clock enable control 0: Disabled 1: Enabled
10	DAC	RW	DAC module configuration clock enable control 0: Disabled 1: Enabled
9	OPA	RW	OPA module configuration clock enable control 0: Disabled 1: Enabled
8	RFU	-	Reserved bits, please keep the default value.
7	LPTIM	RW	LPTIM module configuration clock enable control 0: Disabled 1: Enabled
6	I2C1	RW	I2C1 module configuration clock enable control 0: Disabled 1: Enabled
5	WWDT	RW	WWDT module configuration clock and working clock enable control 0: Disabled 1: Enabled
4	IWDT	RW	IWDT module configuration clock enable control 0: Disabled 1: Enabled
3	RFU	-	Reserved bits, please keep the default value.
2	BTIM	RW	BTIM1-3 module configuration clock and working clock enable control 0: Disabled 1: Enabled
1	RTC	RW	RTC module configuration clock enable control 0: Disabled 1: Enabled
0	RFU	-	Reserved bits, please keep the default value.



4.7.14 SYSCTRL_AHBRST AHB peripheral reset control register

Address offset: 0x40

Reset value: 0x0000 007F

Bit field	Name	Permission	Function description
31:9	RFU	-	Reserved bits, please keep the default value.
8	CORDIC	RW	CORDIC module reset control 0: Module is in reset state. 1: Module works fine.
7	GPIOF	RW	GPIOF module reset control 0: Module is in reset state. 1: Module works fine.
6	GPIOC	RW	GPIOC module reset control 0: Module is in reset state. 1: Module works fine.
5	GPIOB	RW	GPIOB module reset control 0: Module is in reset state. 1: Module works fine.
4	GPIOA	RW	GPIOA module reset control 0: Module is in reset state. 1: Module works fine.
3	EAU	RW	EAU module reset control 0: Module is in reset state. 1: Module works fine.
2	CRC	RW	CRC module reset control 0: Module is in reset state. 1: Module works fine.
1	FLASH	RW	FLASH module reset control 0: Module is in reset state. 1: Module works fine.
0	DMA	RW	DMA module reset control 0: Module is in reset state. 1: Module works fine.



4.7.15 SYSCTRL_APBRST1 APB peripheral reset control register 1

Address offset: 0x48

Reset value: 0x0000 01FF

Bit field	Name	Permission	Function description
31:15	RFU	-	Reserved bits, please keep the default value.
14	SPI3	RW	SPI3 module reset control 0: Module is in reset state. 1: Module works fine.
13	SPI2	RW	SPI2 module reset control 0: Module is in reset state. 1: Module works fine.
12	GTIM4	RW	GTIM4 module reset control 0: Module is in reset state. 1: Module works fine.
11	GTIM3	RW	GTIM3 module reset control 0: Module is in reset state. 1: Module works fine.
10:9	RFU	-	Reserved bits, please keep the default value.
8	UART3	RW	UART3 module reset control 0: Module is in reset state. 1: Module works fine.
7	GTIM2	RW	GTIM2 module reset control 0: Module is in reset state. 1: Module works fine.
6	GTIM1	RW	GTIM1 module reset control 0: Module is in reset state. 1: Module works fine.
5	ATIM	RW	ATIM module reset control 0: Module is in reset state. 1: Module works fine.
4	UART2	RW	UART2 module reset control 0: Module is in reset state. 1: Module works fine.
3	UART1	RW	UART1 module reset control 0: Module is in reset state. 1: Module works fine.
2	SPI1	RW	SPI1 module reset control 0: Module is in reset state. 1: Module works fine.
1	VC	RW	VC module reset control 0: Module is in reset state. 1: Module works fine.



Bit field	Name	Permission	Function description
0	ADC	RW	ADC module reset control 0: Module is in reset state. 1: Module works fine.



4.7.16 SYSCTRL_APBRST2 APB peripheral reset control register 2

Address offset: 0x44

Reset value: 0x0000 00FF

Bit field	Name	Permission	Function description
31:13	RFU	-	Reserved bits, please keep the default value.
12	HALLTIM	RW	HALLTIM module reset control 0: Module is in reset state. 1: Module works fine.
11	I2C2	RW	I2C2 module reset control 0: Module is in reset state. 1: Module works fine.
10	DAC	RW	DAC module reset control 0: Module is in reset state. 1: Module works fine.
9	OPA	RW	OPA module reset control 0: Module is in reset state. 1: Module works fine.
8	RFU	-	Reserved bits, please keep the default value.
7	LPTIM	RW	LPTIM module reset control 0: Module is in reset state. 1: Module works fine.
6	I2C1	RW	I2C1 module reset control 0: Module is in reset state. 1: Module works fine.
5	WWDT	RW	WWDT module reset control 0: Module is in reset state. 1: Module works fine.
4	IWDT	RW	IWDT module reset control 0: Module is in reset state. 1: Module works fine.
3	RFU	-	Reserved bits, please keep the default value.
2	BTIM	RW	BTIM1-3 module reset control 0: Module is in reset state. 1: Module works fine.
1	RTC	RW	RTC module reset control 0: Module is in reset state. 1: Module works fine.
0	RFU	-	Reserved bits, please keep the default value.



4.7.17 SYSCTRL_RESETFLAG System reset flag register

Address offset: 0x4C Reset value: 0x---- ----

Bit field	Name	Permission	Function description
31:10	RFU	-	Reserved bits, please keep the default value.
9	SYSRESETREQ	RW0	Cortex-M0+ CPU SYSRESETREQ soft reset flag 0: No CPU SYSRESETREQ soft reset occurred. 1: CPU SYSRESETREQ soft reset has occurred. Write 0 to clear, write 1 to invalid.
8	LOCKUP	RW0	Cortex-M0+ CPU Lockup reset flag 0: No Lockup reset occurred. 1: Lockup reset has occurred. Write 0 to clear, write 1 to invalid.
7	RFU	-	Reserved bits, please keep the default value.
6	RSTB	RW0	NRST pin reset flag 0: No Pin reset occurred. 1: Pin reset has occurred. Write 0 to clear, write 1 to invalid.
5	WWDT	RW0	WWDT reset flag, requires software initialization and clearing 0: No WWDT reset occurred. 1: WWDT reset has occurred. Write 0 to clear, write 1 to invalid.
4	IWDT	RW0	IWDT reset flag 0: No IWDT reset occurred. 1: IWDT reset has occurred. Write 0 to clear, write 1 to invalid.
3	LVD	RW0	LVD reset flag 0: No LVD reset occurred. 1: LVD reset has occurred. Write 0 to clear, write 1 to invalid.
2:1	RFU	-	Reserved bits, please keep the default value.
0	POR	RW0	POR/BOR reset flag 0: No POR/BOR reset occurred. 1: POR/BOR reset has occurred. Write 0 to clear, write 1 to invalid.



4.7.18 SYSCTRL_DEBUG Debug status timer control register

Address offset: 0x2C Reset value: 0x0000 07E7

Bit field	Name	Permission	Function description
31:12	RFU	-	Reserved bits, please keep the default value.
11	HALLTIM	RW	In the debugging state, HALLTIM count function configuration 0: Normal count. 1: Pause count.
10	WWDT	RW	In the debugging state, WWDT count function configuration 0: Normal count. 1: Pause count.
9	IWDT	RW	In the debugging state, IWDT count function configuration 0: Normal count. 1: Pause count.
8	RTC	RW	In the debugging state, RTC count function configuration 0: Normal count. 1: Pause count.
7	RFU	-	Reserved bits, please keep the default value.
6	LPTIM	RW	In the debugging state, LPTIM count function configuration 0: Normal count. 1: Pause count.
5	BTIM123	RW	In the debugging state, BTIM1/BTIM2/BTIM3 count function configuration 0: Normal count. 1: Pause count.
4	GTIM4	RW	In the debugging state, GTIM4 count function configuration 0: Normal count. 1: Pause count.
3	GTIM3	RW	In the debugging state, GTIM3 count function configuration 0: Normal count. 1: Pause count.
2	GTIM2	RW	In the debugging state, GTIM2 count function configuration 0: Normal count. 1: Pause count.
1	GTIM1	RW	In the debugging state, GTIM1 count function configuration 0: Normal count. 1: Pause count.
0	ATIM	RW	In the debugging state, ATIM count function configuration 0: Normal count. 1: Pause count.



4.7.19 SYSCTRL_MCO System clock output control register

Address offset: 0x70

Reset value: 0x0000 0000

Bit field	Name	Permission	Function description
31:8	RFU	-	Reserved bits, please keep the default value.
7:4	DIV	RW	MCO output frequency division control 0000: 1 frequency division. 0001: 2 frequency division. 0010: 4 frequency division. 0011: 8 frequency division. 0100: 16 frequency division. 0101: 32 frequency division. 0110: 64 frequency division. 0111: 128 frequency division. 1000: 256 frequency division. 1001: 512 frequency division. 1010: 1024 frequency division. 1011: 2048 frequency division. 1100: 4096 frequency division. 1101: 8192 frequency division. 1110: Reserved. 1111: Reserved.
3:0	SOURCE	RW	MCO output signal source configuration 0000: Zero output 0001: HCLK 0010: PCLK 0011: HSIOSC 0100: LSI 0101: HSE 0110: LSE 1001: RC10K <i>Note: When HSE uses an external injection clock, the MCO output is HSE_EX; when LSE uses an external injection clock, the MCO output is LSE_EX.</i>



5 Interrupt

5.1 Overview

Nested vectored interrupt controller (NVIC) for the ARM® Cortex®-M0+ core to manage interrupts and exceptions. The NVIC is tightly coupled to the processor core, enabling low-latency exception and interrupt handling.

The processor supports up to 32 interrupt request (IRQ) inputs and supports multiple internal exceptions.

This chapter only introduces the 32 external interrupt requests (IRQ0 ~ IRQ31) of the processor. Please refer to the "ARM® Cortex®-M0+ Technical Reference Manual" and "ARM® v6-M Architecture Reference Manual" for details of the processor's internal exceptions.



5.2 Main features

- 16 inner exceptions
- 32 maskable external interrupts
- 4 programmable priorities
- Low-latency exception and interrupt handling
- Support for interrupt nesting
- Interrupt vector table remapping



5.3 Interrupt priority

The external interrupt can be set to 4 levels of priority, the highest priority is "0", the lowest priority is "3", and the default value is "0".

If a higher priority interrupt occurs while the processor is executing an interrupt handler, the interrupt is preempted. If an interrupt occurs with the same or lower priority as the interrupt being processed, the interrupt will not be preempted, but the state of the new interrupt will change to pending. If multiple pending interrupts have the same priority, the pending interrupt with the smaller interrupt number takes precedence. For example, if both IRQ [0] and IRQ [1] are pending, and both have the same priority, then IRQ [0] is processed first.



5.4 Interrupt vector table

When the ARM® Cortex®-M0+ responds to an interrupt, the processor automatically fetches the start address of the Interrupt Service Routine (ISR) from the interrupt vector table in the memory. The interrupt vector table includes the initial value of the main stack pointer (MSP), the entry address of the service routine for internal exceptions and external interrupts. Each interrupt vector occupies 1 word (4 bytes), and the storage address of the interrupt vector is the vector number multiplied by 4, as shown in the following table:

Table 5-1 Interrupt vector table

Vector number	External interrupt number (IRQ#)	Priority	Interrupt source	Introduction	Address
0	-	-	-	MSP initial value	0x0000 0000
1	-	-3	Reset	Reset vector	0x0000 0004
2	-	-2	NMI	Non-maskable interrupt ^[1]	0x0000 0008
3	-	-1	HardFault	Hardware error exception (fault)	0x0000 000C
4-10	-	-	-	Reserved	0x0000 0010~002B
11	-	Configurable	SVCALL	Hypervisors invoked via SWI instructions	0x0000 002C
12-13	-	-	-	Reserved	0x0000 0030~0037
14	-	Configurable	PendSV	Suspendable requests for system services	0x0000 0038
15	-	Configurable	SysTick	System tick timer	0x0000 003C
16	0	Configurable	WDT ^[2]	Window watchdog and independent watchdog interrupt	0x0000 0040
17	1	Configurable	LVD	LVD global interrupt	0x0000 0044
18	2	Configurable	RTC	RTC global interrupt	0x0000 0048
19	3	Configurable	FLASH/EEPROM ^[2]	FLASH/EEPROM global interrupt	0x0000 004C
20	4	Configurable	RCC	RCC global interrupt ^[3]	0x0000 0050
21	5	Configurable	GPIOA	GPIOA global interrupt	0x0000 0054
22	6	Configurable	GPIOB	GPIOB global interrupt	0x0000 0058
23	7	Configurable	GPIOC	GPIOC global interrupt	0x0000 005C
24	8	Configurable	GPIOF	GPIOF global interrupt	0x0000 0060
25	9	Configurable	DMA1/2	DMA channel 1 and DMA 2 global interrupt	0x0000 0064



Table 5-1 Interrupt vector table (continued)

Vector number	External interrupt number (IRQ#)	Priority	Interrupt source	Introduction	Address
26	10	Configurable	DMA3/4	DMA channel 3 and DMA 4 global interrupt	0x0000 0068
27	11	Configurable	CORDIC	CORDIC global interrupt	0x0000 006C
28	12	Configurable	ADC1	ADC1 global interrupt	0x0000 0070
29	13	Configurable	ATIM	ATIM global interrupt	0x0000 0074
30	14	Configurable	VC1/3	VC1/3 global interrupt	0x0000 0078
31	15	Configurable	VC2/4	VC2/4 global interrupt	0x0000 007C
32	16	Configurable	GTIM1	GTIM1 global interrupt	0x0000 0080
33	17	Configurable	GTIM2	GTIM2 global interrupt	0x0000 0084
34	18	Configurable	GTIM3/4	GTIM3/4 global interrupt	0x0000 0088
35	19	Configurable	LPTIM	LPTIM global interrupt	0x0000 008C
36	20	Configurable	BTIM1	BTIM1 global interrupt	0x0000 0090
37	21	Configurable	BTIM2	BTIM2 global interrupt	0x0000 0094
38	22	Configurable	BTIM3/ HALLTIM	BTIM3 and HALLTIM global interrupt	0x0000 0098
39	23	Configurable	I2C1	I2C1 global interrupt	0x0000 009C
40	24	Configurable	I2C2	I2C2 global interrupt	0x0000 00A0
41	25	Configurable	SPI1	SPI1 global interrupt	0x0000 00A4
42	26	Configurable	SPI2/3	SPI2/3 global interrupt	0x0000 00A8
43	27	Configurable	UART1	UART1 global interrupt	0x0000 00AC
44	28	Configurable	UART2	UART2 global interrupt	0x0000 00B0
45	29	Configurable	UART3	UART3 global interrupt	0x0000 00B4
46	30	Configurable	ADC2/DAC	ADC2 and DAC global interrupt	0x0000 00B8
47	31	Configurable	FAULT	HSE/LSE operation fail interrupt ^[4]	0x0000 00BC

[1] NMI is not used in CW32L012.

[2] Since the interrupts of some peripherals multiplex an IRQ interrupt source, the user should first check the interrupt flag bit in the interrupt service routine to determine the peripheral that generates the interrupt.

[3] HSE, LSE clock signal start-up failure and LSI, LSE, HSIOSC, HSE clock signal stability correspond to RCC global interrupt.

[4] The failure of the HSE or LSE clock signal during operation corresponds to a FAULT interrupt.



5.5 Interrupt related registers

5.5.1 NVIC interrupt enable and disable enable

The ARM® Cortex®-M0+ processor supports up to 32 external interrupt sources, corresponding to the 32 enable bits of the interrupt enable setting register NVIC_ISER and the 32 disable bits of the interrupt enable clear register NVIC_ICER. Setting the enable bit to 1 enables interrupts; Set the disable bit to 1, disable interrupts.

The above NVIC interrupt enable is only for the processor NVIC, whether the interrupt of the peripheral is enabled is also controlled by the interrupt control register of the corresponding peripheral.

5.5.2 NVIC interrupt pending and clear pending

When an interrupt occurs, if the system is processing an interrupt of the same priority or higher, the system will not process the interrupt immediately, but will set the interrupt status to pending and save it in the interrupt pending status register; The pending state will remain valid until the processor enters this interrupt processing, unless the pending state is manually cleared. When the processor begins to enter the interrupt processing, the hardware will automatically clear the corresponding interrupt pending state.

The user can set the status of this interrupt to the pending state by setting the corresponding bit of the interrupt pending setting register NVIC_ISPR. If the system is not processing an interrupt with the same priority or higher priority, the interrupt will be immediately responded and deal with.

The user can set the state of this interrupt to the pending clear state by setting the corresponding bit of the interrupt pending clear register NVIC_ICPR.

5.5.3 NVIC interrupt priority

The interrupt priority control registers NVIC_IPR0~NVIC_IPR7 are used to set the interrupt priority of IRQ0~IRQ31. Each interrupt source uses 8 bits. In CW32L012 only the upper two bits are used, and up to 4 interrupt priorities can be set.

Attention: The setting of the interrupt priority register of ARM® Cortex®-M0+ should be before the interrupt is enabled. The user cannot change the interrupt priority after the interrupt is enabled, which will lead to unpredictable results.



5.5.4 NVIC interrupt mask

In some special occasions, all interrupts need to be disabled, which can be realized by using the interrupt mask register PRIMASK. Only the lowest 1 bit of PRIMASK is valid. If this bit is set to 1, all external interrupts and exceptions except NMI and hardware error exceptions are disabled; after clearing to 0, it is allowed to respond to interrupts and exceptions. This bit defaults to 0 after reset.

ARM® Cortex®-M0+ has dedicated ARM instructions for modifying PRIMASK register, CPSIE i and CPSID i, please refer to "ARM® v6-M Architecture Reference Manual" for details.

Assembly instruction example reference:

```
CPSIE i ; Clear PRIMASK (Enable interrupt)
CPSID i ; Set up PRIMASK (Disable interrupts)
```

C-language (Calling the CMSIS device driver library) example reference:

```
void __enable_irq(void);    //Clear PRIMASK
void __disable_irq(void);   //Set up PRIMASK
```

5.5.5 Peripheral interrupt enable

Peripheral modules generally have their own interrupt enable registers. When using interrupts, you must first enable the peripheral interrupt enable, and refer to [Table 5-1: Interrupt vector table](#) to enable the NVIC interrupt for the interrupt source. For the interrupt enable of specific peripherals, please refer to the description of the relevant peripheral modules.



5.6 List of registers

NVIC base address: NVIC_BASE = 0xE000 E000

Table 5-2 List of NVIC registers

Register name	Register address	Register description
NVIC_ISER	NVIC_BASE + 0x100	IRQ0~IRQ31 Interrupt enable setting register
NVIC_ICER	NVIC_BASE + 0x180	IRQ0~IRQ31 Interrupt enable clear register
NVIC_ISPR	NVIC_BASE + 0x200	IRQ0~IRQ31 Interrupt pending setup register
NVIC_ICPR	NVIC_BASE + 0x280	IRQ0~IRQ31 Interrupt pending clear register
NVIC_IPR0	NVIC_BASE + 0x400	IRQ0~IRQ3 Interrupt priority control register 0
NVIC_IPR1	NVIC_BASE + 0x404	IRQ4~IRQ7 Interrupt priority control register 1
NVIC_IPR2	NVIC_BASE + 0x408	IRQ8~IRQ11 Interrupt priority control register 2
NVIC_IPR3	NVIC_BASE + 0x40C	IRQ12~IRQ15 Interrupt priority control register 3
NVIC_IPR4	NVIC_BASE + 0x410	IRQ16~IRQ19 Interrupt priority control register 4
NVIC_IPR5	NVIC_BASE + 0x414	IRQ20~IRQ23 Interrupt priority control register 5
NVIC_IPR6	NVIC_BASE + 0x418	IRQ24~IRQ27 Interrupt priority control register 6
NVIC_IPR7	NVIC_BASE + 0x41C	IRQ28~IRQ31 Interrupt priority control register 7



5.7 Register description

See section 1 [Documentation conventions](#) for abbreviations used in register descriptions.

5.7.1 NVIC_ISER Interrupt enable setting register

Address offset: 0x100 Reset value: 0x0000 0000

Bit field	Name	Permission	Function description
31:0	SETIRQ	RW	Set to enable external interrupt IRQ0~IRQ31; write "1" to set, write "0" is invalid. [0]: IRQ0 [1]: IRQ1 [2]: IRQ2 [31]: IRQ31 The read value indicates the current interrupt enable state.

5.7.2 NVIC_ICER Interrupt enable clear register

Address offset: 0x180 Reset value: 0x0000 0000

Bit field	Name	Permission	Function description
31:0	CLRIRQ	RW	Set to disable external interrupt IRQ0~IRQ31; write "1" to set, write "0" is invalid. [0]: IRQ0 [1]: IRQ1 [2]: IRQ2 [31]: IRQ31 The read value indicates the current interrupt enable state.



5.7.3 NVIC_ISPR Interrupt pending setup register

Address offset: 0x200 Reset value: 0x0000 0000

Bit field	Name	Permission	Function description
31:0	SETPEND	RW	Set the pending state of external interrupt IRQ0~IRQ31; write "1" to set, write "0" is invalid. [0]: IRQ0 [1]: IRQ1 [2]: IRQ2 [31]: IRQ31 The read value indicates the current interrupt pending state.

5.7.4 NVIC_ICPR Interrupt pending clear register

Address offset: 0x280 Reset value: 0x0000 0000

Bit field	Name	Permission	Function description
31:0	CLRPEND	RW	Clear the pending state of external interrupt IRQ0~IRQ31; write "1" to clear, write "0" is invalid. [0]: IRQ0 [1]: IRQ1 [2]: IRQ2 [31]: IRQ31 The read value indicates the current interrupt pending state.



5.7.5 NVIC_IPR0 Interrupt priority control register 0

Address offset: 0x400 Reset value: 0x0000 0000

Bit field	Name	Permission	Function description
31:30	IPRIRQ3	RW	The priority of interrupt IRQ3, 00 has the highest priority and 11 has the lowest priority.
29:24	RFU	-	Reserved bits, please keep the default value.
23:22	IPRIRQ2	RW	The priority of interrupt IRQ2, 00 has the highest priority and 11 has the lowest priority.
21:16	RFU	-	Reserved bits, please keep the default value.
15:14	IPRIRQ1	RW	The priority of interrupt IRQ1, 00 has the highest priority and 11 has the lowest priority.
13:8	RFU	-	Reserved bits, please keep the default value.
7:6	IPRIRQ0	RW	The priority of interrupt IRQ0, 00 has the highest priority and 11 has the lowest priority.
5:0	RFU	-	Reserved bits, please keep the default value.



5.7.6 NVIC_IPR1 Interrupt priority control register 1

Address offset: 0x404 Reset value: 0x0000 0000

Bit field	Name	Permission	Function description
31:30	IPRIRQ7	RW	The priority of interrupt IRQ7, 00 has the highest priority and 11 has the lowest priority.
29:24	RFU	-	Reserved bits, please keep the default value.
23:22	IPRIRQ6	RW	The priority of interrupt IRQ6, 00 has the highest priority and 11 has the lowest priority.
21:16	RFU	-	Reserved bits, please keep the default value.
15:14	IPRIRQ5	RW	The priority of interrupt IRQ5, 00 has the highest priority and 11 has the lowest priority.
13:8	RFU	-	Reserved bits, please keep the default value.
7:6	IPRIRQ4	RW	The priority of interrupt IRQ4, 00 has the highest priority and 11 has the lowest priority.
5:0	RFU	-	Reserved bits, please keep the default value.

5.7.7 NVIC_IPR2 Interrupt priority control register 2

Address offset: 0x408 Reset value: 0x0000 0000

Bit field	Name	Permission	Function description
31:30	IPRIRQ11	RW	The priority of interrupt IRQ11, 00 has the highest priority and 11 has the lowest priority.
29:24	RFU	-	Reserved bits, please keep the default value.
23:22	IPRIRQ10	RW	The priority of interrupt IRQ10, 00 has the highest priority and 11 has the lowest priority.
21:16	RFU	-	Reserved bits, please keep the default value.
15:14	IPRIRQ9	RW	The priority of interrupt IRQ9, 00 has the highest priority and 11 has the lowest priority.
13:8	RFU	-	Reserved bits, please keep the default value.
7:6	IPRIRQ8	RW	The priority of interrupt IRQ8, 00 has the highest priority and 11 has the lowest priority.
5:0	RFU	-	Reserved bits, please keep the default value.



5.7.8 NVIC_IPR3 Interrupt priority control register 3

Address offset: 0x40C Reset value: 0x0000 0000

Bit field	Name	Permission	Function description
31:30	IPRIRQ15	RW	The priority of interrupt IRQ15, 00 has the highest priority and 11 has the lowest priority.
29:24	RFU	-	Reserved bits, please keep the default value.
23:22	IPRIRQ14	RW	The priority of interrupt IRQ14, 00 has the highest priority and 11 has the lowest priority.
21:16	RFU	-	Reserved bits, please keep the default value.
15:14	IPRIRQ13	RW	The priority of interrupt IRQ13, 00 has the highest priority and 11 has the lowest priority.
13:8	RFU	-	Reserved bits, please keep the default value.
7:6	IPRIRQ12	RW	The priority of interrupt IRQ12, 00 has the highest priority and 11 has the lowest priority.
5:0	RFU	-	Reserved bits, please keep the default value.

5.7.9 NVIC_IPR4 Interrupt priority control register 4

Address offset: 0x410 Reset value: 0x0000 0000

Bit field	Name	Permission	Function description
31:30	IPRIRQ19	RW	The priority of interrupt IRQ19, 00 has the highest priority and 11 has the lowest priority.
29:24	RFU	-	Reserved bits, please keep the default value.
23:22	IPRIRQ18	RW	The priority of interrupt IRQ18, 00 has the highest priority and 11 has the lowest priority.
21:16	RFU	-	Reserved bits, please keep the default value.
15:14	IPRIRQ17	RW	The priority of interrupt IRQ17, 00 has the highest priority and 11 has the lowest priority.
13:8	RFU	-	Reserved bits, please keep the default value.
7:6	IPRIRQ16	RW	The priority of interrupt IRQ16, 00 has the highest priority and 11 has the lowest priority.
5:0	RFU	-	Reserved bits, please keep the default value.



5.7.10 NVIC_IPR5 Interrupt priority control register 5

Address offset: 0x414 Reset value: 0x0000 0000

Bit field	Name	Permission	Function description
31:30	IPRIRQ23	RW	The priority of interrupt IRQ23, 00 has the highest priority and 11 has the lowest priority.
29:24	RFU	-	Reserved bits, please keep the default value.
23:22	IPRIRQ22	RW	The priority of interrupt IRQ22, 00 has the highest priority and 11 has the lowest priority.
21:16	RFU	-	Reserved bits, please keep the default value.
15:14	IPRIRQ21	RW	The priority of interrupt IRQ21, 00 has the highest priority and 11 has the lowest priority.
13:8	RFU	-	Reserved bits, please keep the default value.
7:6	IPRIRQ20	RW	The priority of interrupt IRQ20, 00 has the highest priority and 11 has the lowest priority.
5:0	RFU	-	Reserved bits, please keep the default value.

5.7.11 NVIC_IPR6 Interrupt priority control register 6

Address offset: 0x418 Reset value: 0x0000 0000

Bit field	Name	Permission	Function description
31:30	IPRIRQ27	RW	The priority of interrupt IRQ27, 00 has the highest priority and 11 has the lowest priority.
29:24	RFU	-	Reserved bits, please keep the default value.
23:22	IPRIRQ26	RW	The priority of interrupt IRQ26, 00 has the highest priority and 11 has the lowest priority.
21:16	RFU	-	Reserved bits, please keep the default value.
15:14	IPRIRQ25	RW	The priority of interrupt IRQ25, 00 has the highest priority and 11 has the lowest priority.
13:8	RFU	-	Reserved bits, please keep the default value.
7:6	IPRIRQ24	RW	The priority of interrupt IRQ24, 00 has the highest priority and 11 has the lowest priority.
5:0	RFU	-	Reserved bits, please keep the default value.



5.7.12 NVIC_IPR7 Interrupt priority control register 7

Address offset: 0x41C Reset value: 0x0000 0000

Bit field	Name	Permission	Function description
31:30	IPRIRQ31	RW	The priority of interrupt IRQ31, 00 has the highest priority and 11 has the lowest priority.
29:24	RFU	-	Reserved bits, please keep the default value.
23:22	IPRIRQ30	RW	The priority of interrupt IRQ30, 00 has the highest priority and 11 has the lowest priority.
21:16	RFU	-	Reserved bits, please keep the default value.
15:14	IPRIRQ29	RW	The priority of interrupt IRQ29, 00 has the highest priority and 11 has the lowest priority.
13:8	RFU	-	Reserved bits, please keep the default value.
7:6	IPRIRQ28	RW	The priority of interrupt IRQ28, 00 has the highest priority and 11 has the lowest priority.
5:0	RFU	-	Reserved bits, please keep the default value.



6 RAM memory

6.1 Overview

The CW32L012 integrates 8KB embedded RAM for users to use, which is used to store various data in the process of program execution. The starting address of the RAM is 0x2000 0000, and the data is stored in the little-endian mode in the RAM, that is, the lowest byte address space stores the least significant byte data of the data.



6.2 Main features

- Support access in 3 widths of byte (8bit), half word (16bit) or full word (32bit).
- Zero latency, can be accessed by CPU at maximum system clock frequency.
- Support parity check function.



6.3 RAM memory operation

The RAM memory operations that the user can perform include: read operations and write operations.

The read and write operations of RAM support three-bit widths of 8bit, 16bit and 32bit. The user program can complete the read and write by directly accessing the absolute address, but it should be noted that the data bit width of read and write must be aligned with the corresponding address boundary, otherwise read and write the operation is invalid and will result in a HardFault hardware error exception.

6.3.1 Read operation

The read operation supports 3 different bit widths, which can be read by direct access to the absolute address, but it should be noted that the read data bit width must be aligned with the corresponding address boundary.

Code example:

8bit read:

```
tempdata = * ( ( uint8_t * ) 0x2000 0001 );
```

16bit read:

```
tempdata = * ( ( uint16_t * ) 0x2000 0002 );
```

32bit read:

```
tempdata = * ( ( uint32_t * ) 0x2000 0004 );
```

6.3.2 Write operation

The write operation supports 3 different bit widths, and data can be written by directly accessing the absolute address, but it should be noted that the written data bit width must be aligned with the corresponding address boundary.

Code example:

8bit write:

```
* ( ( uint8_t * ) 0x2000 0001 ) = 0x12;
```

16bit write:

```
* ( ( uint16_t * ) 0x2000 0002 ) = 0x1234;
```

32bit write:

```
* ( ( uint32_t * ) 0x2000 0004 ) = 0x1234 5678;
```



6.4 Parity check function

The CW32L012 supports the parity check function of RAM. After power-on, the parity check function is enabled by default and cannot be configured by the user.

Each byte of RAM data is actually stored in a 9bit physical space, including 8bit data bits and 1bit parity bits.

When the CPU writes to the RAM, the parity check unit of the RAM calculates the check digit and writes it into the corresponding check digit space. When reading RAM data, the data is read together with the check digit. The CPU performs parity check on each byte of the data, and compares the calculated check digit with the read check digit. If they are consistent, it means that the data in the RAM is correct. If it is inconsistent, the parity error flag `RAM_ISR.PARITY` is set. If the RAM parity error interrupt enables control bit `RAM_IER.PARITY` is set to 1, the CPU will respond to the interrupt service. The user program can set `RAM_ICR.PARITY` to 0 to clear the parity error flag.

The user can obtain the RAM address where the parity error occurs by reading the parity error address register `RAM_ADDR`.



6.5 List of registers

RAM base address: RAM_BASE = 0x4002 2400

Table 6-1 List of RAM registers

Register name	Register address	Register description
RAM_IER	RAM_BASE + 0x00	Interrupt enable control register
RAM_ADDR	RAM_BASE + 0x04	Parity error address register
RAM_ISR	RAM_BASE + 0x08	Interrupt flag register
RAM_ICR	RAM_BASE + 0x0C	Interrupt flag clear register



6.6 Register description

See section 1 [Documentation conventions](#) for abbreviations used in register descriptions.

6.6.1 RAM_IER Interrupt enable control register

Address offset: 0x00 Reset value: 0x0000 0000

Bit field	Name	Permission	Function description
31:1	RFU	-	Reserved bits, please keep the default value.
0	PARITY	RW	RAM parity error interrupt enable control 0: Disabled 1: Enabled

6.6.2 RAM_ADDR Parity error address register

Address offset: 0x04 Reset value: 0x2000 0000

Bit field	Name	Permission	Function description
31:0	ADDR	RO	The address of the RAM where the parity error is located

6.6.3 RAM_ISR Interrupt flag register

Address offset: 0x08 Reset value: 0x0000 0000

Bit field	Name	Permission	Function description
31:1	RFU	-	Reserved bits, please keep the default value.
0	PARITY	RO	Parity error flag 0: No parity error occurred. 1: A parity error has occurred.

6.6.4 RAM_ICR Interrupt flag clear register

Address offset: 0x0C Reset value: 0x0000 0001

Bit field	Name	Permission	Function description
31:1	RFU	-	Reserved bits, please keep the default value.
0	PARITY	R1W0	Parity error flag clear control W0: Clear parity error flag. W1: No function.



7 FLASH Memory

7.1 Overview

The CW32L012 integrates 64KB embedded FLASH for users to use, which can be used to store application programs and user data. The chip supports read, erase and write operations of FLASH memory, and supports erase and write protection and read protection; the chip has a built-in high-voltage BOOST circuit required for FLASH programming, and no additional programming voltage is required.



7.2 Main features

- Support access in 3 widths of byte (8bit), half word (16bit) or full word (32bit)
- Supports read, erase and write operations
- Support erase and write protection, read protection
- Supports security runtime library protection
- Support caching and prefetching



7.3 FLASH memory organization

The CW32L012 integrates 64KB user-accessible FLASH memory, which is managed by paging of 512 bytes per page, with a total of 128 pages. Users can perform full-page erasing and byte-by-byte programming operations on FLASH.

In addition to the user-accessible 64KB FLASH memory, the CW32L012 also integrates a 2KB boot program memory, which solidified with BootLoader and other codes when it leaves the factory, and the user cannot access this memory.



7.4 FLASH memory read wait cycle configuration

The internal FLASH memory of CW32L012 supports the operation clock with the fastest frequency of 24MHz. When the configured HCLK frequency is greater than 24MHz, the number of inserted waiting HCLK cycles needs to be configured through the WAIT bit field of the FLASH control register FLASH_CR2. The configuration rules are shown in the following table:

Table 7-1 FLASH read wait cycle configuration

FLASH_CR2.WAIT	The number of HCLK cycles waiting for the instruction fetch cycle	HCLK frequency
000	0	$HCLK \leq 24MHz$
001	1	$HCLK \leq 48MHz$
010	2	$HCLK \leq 72MHz$
011	3	$HCLK \leq 96MHz$

Attention: When configuring the FLASH read wait period, the FLASH configuration clock must be enabled first.



7.5 FLASH memory operation

User-executable FLASH memory operations include: read operations, erase operations, and write (programming) operations.

The read and write operations of FLASH support 3 widths of 8bit, 16bit and 32bit. The user program can complete the read and write by directly accessing the absolute address, but it should be noted that the data bit width of read and write must be aligned with the corresponding address boundary, otherwise it will cause HardFault hardware error exception.

Attention: FLASH_CR2.CACHE and FLASH_CR2.FETCH must be set to 0 to disable the cache and prefetch functions during page erase or write operations.



7.5.1 Page erase

The minimum unit of the page erase operation of FLASH is 1 page, that is, 512 bytes. After the page erase operation is completed, the data content of all address spaces of the page is 0xFF.

If the page erase operation is performed on the unlocked FLASH page, the operation will fail, and the FLASH_ISR.PAGELOCK flag will be set by hardware. If the FLASH_IER.PAGELOCK is set to 1, the CPU will execute the corresponding interrupt service routine. User can clear the FLASH_ISR.PAGELOCK interrupt flag by setting FLASH_ICR.PAGELOCK to 0.

If the program runs in FLASH, and the page erase operation is performed on the storage space of the page where the PC (program pointer) is located, the operation will fail, and the FLASH_ISR.PC flag will be set by hardware. If FLASH_IER.PC is set to 1, then The CPU will execute the corresponding interrupt service routine. The user can clear the FLASH_ISR.PC interrupt flag by setting FLASH_ICR.PC to 0.

FLASH page erase operation steps are as follows:

- Step 1:** Set FLASH_CR2.CACHE and FLASH_CR2.FETCH to 0 to disable caching and prefetching.
Attention: The FLASH_CR2 register has the KEY protection feature. The upper 16-bit data of the written data must be 0x5A5A, otherwise it cannot be written.
- Step 2:** Set the FLASH operation mode to page erase mode, and write 0x02 to the working mode bit field MODE of the FLASH control register FLASH_CR1.
Attention: The FLASH_CR1 register has the KEY protection feature. The upper 16-bit data of the written data must be 0x5A5A, otherwise it cannot be written.
- Step 3:** To unlock the erasure protection of the page to be erased, see [Table 7-2: FLASH_PAGELOCK Erase lock protection](#), write 1 to the lock bit field LOCKx corresponding to the erasing lock register FLASH_PAGELOCK.
Attention: The FLASH_PAGELOCK register has the KEY protection feature. The upper 16-bit data of the written data must be 0x5A5A, otherwise it cannot be written.
- Step 4:** Write any data to any address in the page to be erased, triggering the erase operation of the page.
Attention: The write data bit width needs to be aligned with the corresponding address boundary.
 Code example,
- ```
*((uint8_t *) 0x0000 0001) = 0x55;
```
- Step 5:** Wait for the erase to complete, and query the FLASH busy flag bit FLASH\_ISR.BUSY becomes 0.
- Step 6:** To erase other pages, repeat step 3 to step 5.
- Step 7:** After the erase operation is completed, if you want to restore the lock protection of FLASH, you need to write 0 to the lock bit field LOCKx corresponding to the erase/write lock register FLASH\_PAGELOCK to activate the lock protection function to protect the FLASH content from being erased by mistake.
- Step 8:** Set the FLASH operation mode to read mode, and write 0x00 to the working mode bit field MODE of the FLASH control register FLASH\_CR1.
- Step 9:** Set FLASH\_CR2.CACHE and FLASH\_CR2.FETCH to 1 to enable caching and prefetching.





### 7.5.2 Write operation

Based on the characteristics of embedded FLASH, the write operation can only rewrite the bit data in the FLASH memory from '1' to '0', but cannot be rewritten from '0' to '1'. Therefore, before writing data, the page where the corresponding address is located must be erased.

The following three principles must be followed for the FLASH write operation:

1. Cannot write to the address whose data bit content is '0'.
2. Cannot write to the address in the locked area.
3. Cannot write to the address of the page where the PC (program pointer) is located.

When the CW32L012 performs a write operation, it will check whether the contents of the storage space are all '1'. FLASH has three write operation bit widths of 8bit, 16bit and 32bit, and the number of bytes to be checked is 1Byte, 2Byte and 4Byte respectively.

If the internal storage space is not all '1', the write operation will fail, and the FLASH\_ISR.PROG flag will be set by hardware. If FLASH\_IER.PROG is set to 1, the CPU will execute the corresponding interrupt service routine. User can clear the FLASH\_ISR.PROG interrupt flag by setting FLASH\_ICR.PROG to 0.

If the storage space of the unlocked FLASH page is written, the operation will fail, and the FLASH\_ISR.PAGELOCK flag will be set by hardware. If FLASH\_IER.PAGELOCK is set to 1, the CPU will execute the corresponding interrupt service routine. User can clear the FLASH\_ISR.PAGELOCK interrupt flag by setting FLASH\_ICR.PAGELOCK to 0.

If the program runs in FLASH, and the write operation is performed on the storage space of the page where the PC (program pointer) is located, the operation will fail, and the FLASH\_ISR.PC flag will be set by hardware. If FLASH\_IER.PC is set to 1, the CPU will Execute the corresponding interrupt service routine. The user can clear the FLASH\_ISR.PC interrupt flag by setting FLASH\_ICR.PC to 0.

The FLASH write operation steps are as follows:

- Step 1:** Set FLASH\_CR2.CACHE and FLASH\_CR2.FETCH to 0 to disable caching and prefetching.  
*Attention: The FLASH\_CR2 register has the KEY protection feature. The upper 16-bit data of the written data must be 0x5A5A, otherwise it cannot be written.*
- Step 2:** Set the FLASH operation mode to write (programming) mode, and write 0x01 to the working mode bit field MODE of the FLASH control register FLASH\_CR1.  
*Attention: The FLASH\_CR1 register has the KEY protection feature. The upper 16-bit data of the written data must be 0x5A5A, otherwise it cannot be written.*
- Step 3:** To unlock the erasure protection of the page to be written, see [Table 7-2: FLASH\\_PAGELOCK Erase lock protection](#), write 1 to the lock bit field LOCKx corresponding to the erasing lock register FLASH\_PAGELOCK.  
*Attention: The FLASH\_PAGELOCK register has the KEY protection feature. The upper 16-bit data of the written data must be 0x5A5A, otherwise it cannot be written.*
- Step 4:** Write the data to be written into the target address and trigger the FLASH write operation. Note that the bit width of the written data needs to be aligned with the corresponding address boundary.  
 8bit width code example:  

```
*((uint8_t *) 0x0000 0001) = 0x55;
```

  
 16bit width code example:  

```
*((uint16_t *) 0x0000 0002) = 0x1234;
```

  
 32bit width code example:  

```
*((uint32_t *) 0x0000 0004) = 0x1234 5678;
```
- Step 5:** Wait for the programming to be completed, query and wait for the FLASH busy flag bit FLASH\_ISR.BUSY to become 0.



- Step 6:** If you need to write more data to the address range of the currently unlocked area, repeat step 4 to step 5; if the write address exceeds the address range of the current unlocked area, repeat step 3 to step 5.
- Step 7:** After the write operation is completed, if you need to restore the lock protection of the FLASH, write 0 to the lock bit field LOCKx corresponding to the erase/write lock register FLASH\_PAGELOCK to activate the lock protection function to protect the FLASH content from being erased by mistake.
- Step 8:** Set the FLASH operation mode to read mode, and write 0x00 to the working mode bit field MODE of the FLASH control register FLASH\_CR1.
- Step 9:** Set FLASH\_CR2.CACHE and FLASH\_CR2.FETCH to 1 to enable caching and prefetching.

### 7.5.3 Read operation

The CW32L012 supports 3 different bit widths for the FLASH read operation, which can be read by direct access to the absolute address. Note that the read data bit width must be aligned with the corresponding address boundary.

8bit width code example:

```
tempdata = *((uint8_t *) 0x0000 0001);
```

16bit width code example:

```
tempdata = *((uint16_t *) 0x0000 0002);
```

32bit width code example:

```
tempdata = *((uint32_t *) 0x0000 0004);
```



## 7.6 FLASH memory protection

FLASH memory has erase and write protection and read protection functions.

The erasure protection includes lock page erasure protection and PC address page erasure protection. Pages in the protected state cannot be erased and written, which can avoid accidental rewriting of FLASH content.

The read protection takes the entire FLASH as the protection object, and does not support single-page protection, which can prevent the user code from being illegally read.



### 7.6.1 Erase and write protection

The CW32L012 realizes the erasing and writing locking of FLASH pages by setting the erasing and writing lock register FLASH\_PAGELOCK. Pages in the locked state cannot be page erased or written.

The internal FLASH memory of CW32L012 is divided into 128 pages, and each 8 pages corresponds to a LOCKx lock bit of the FLASH\_PAGELOCK register. The LOCKx field has a total of 16 bits, which can realize the lock protection of all 128 pages of FLASH memory. The correspondence between the fields of the erasing lock register FLASH\_PAGELOCK and the FLASH lock page is shown in the following table:

Table 7-2 FLASH\_PAGELOCK Erase lock protection

| FLASH_PAGELOCK bit field | Bit field name | Lock page         |
|--------------------------|----------------|-------------------|
| 15                       | LOCK15         | Page120 – Page127 |
| 14                       | LOCK14         | Page112 – Page119 |
| 13                       | LOCK13         | Page104- Page111  |
| 12                       | LOCK12         | Page96 – Page103  |
| 11                       | LOCK11         | Page88 – Page95   |
| 10                       | LOCK10         | Page80 – Page87   |
| 9                        | LOCK9          | Page72 – Page79   |
| 8                        | LOCK8          | Page64- Page71    |
| 7                        | LOCK7          | Page56 - Page63   |
| 6                        | LOCK6          | Page48 - Page55   |
| 5                        | LOCK5          | Page40 - Page47   |
| 4                        | LOCK4          | Page32 - Page39   |
| 3                        | LOCK3          | Page24 - Page31   |
| 2                        | LOCK2          | Page16 - Page23   |
| 1                        | LOCK1          | Page8 - Page15    |
| 0                        | LOCK0          | Page0 - Page7     |

When performing page erasing and writing operations on FLASH, you must first unlock the corresponding page by setting the corresponding FLASH\_PAGELOCK.LOCKx bit field to 1. An example of the unlock operation is as follows:

- Lock Page0-Page7 code example,

```
CW_FLASH->PAGELOCK = 0x5A5A 0000 | (CW_FLASH->PAGELOCK & 0x0000 FFFE);
```

- Unlock Page32-Page39 code example,

```
CW_FLASH->PAGELOCK = 0x5A5A 0000 | (CW_FLASH->PAGELOCK | 0x0000 0010);
```

**Attention:** The FLASH\_PAGELOCK register has the KEY protection feature. The upper 16-bit data of the written data must be 0x5A5A, otherwise it cannot be written.

If the page erase or write operation is performed directly on the unlocked FLASH page, the operation will fail and an interrupt flag will be generated. Please refer to section [7.5.1 Page erase](#) and section [7.5.2 Write operation](#).



### 7.6.2 Erase and write PC page protection

The FLASH memory of CW32L012 supports the function of erasing and writing PC page protection.

When the user program runs FLASH, if the current program pointer PC is just within the range of the FLASH address page to be erased, the erase operation will fail, and the FLASH\_ISR.PC flag will be set by hardware. If FLASH\_IER.PC is set to 1, the CPU will execute the corresponding interrupt service routine. The user can clear the FLASH\_ISR.PC interrupt flag by setting FLASH\_ICR.PC to 0.

### 7.6.3 Read protection

The CW32L012 supports the FLASH read protection function. After the read protection is set, the FLASH cannot be read by ISP or SWD. Read protection only supports whole-chip FLASH protection, and does not support page-by-page protection.

The read protection is divided into 4 protection levels. The current protection level can be obtained by reading the security bit field SECURITY of the FLASH control register FLASH\_CR1. The security bit field is a read-only attribute and cannot be modified. The read protection level can be set by the ISP command, please refer to the ISP programming documentation.

The CPU fetches instructions from FLASH and the program's read operation to FLASH is not affected by the FLASH read protection function.

The read protection function of FLASH is shown in the following table:

Table 7-3 FLASH read protection

| Protection level | FLASH_CR1.SECURITY | Function description                                                                                                                                                                                                                           |
|------------------|--------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Level0           | 00                 | Read protection is not set.<br>The FLASH can be read by SWD or ISP.                                                                                                                                                                            |
| Level1           | 01                 | FLASH content cannot be read by SWD or ISP.<br>The protection level can be reduced to Level 0 through ISP or SWD. After the downgrade, the content in the FLASH is all 0xFF, that is, it is in the empty chip state of the whole chip erasing. |
| Level2           | 10                 | FLASH content cannot be read by SWD or ISP.<br>The protection level can only be lowered by ISP, but after the downgrade, the content in the FLASH is all 0xFF, that is, it is in an empty chip state of whole chip erasing.                    |
| Level3           | 11                 | FLASH content cannot be read by SWD or ISP.<br>Both ISP and SWD downgrade functions are disabled.<br>At this protection level, the chip can only be programmed once.                                                                           |



### 7.6.4 Security runtime library protection

The FLASH memory of CW32L012 supports the security runtime library function, which allows the solution provider to store the core algorithm in the security runtime library area for customers to call during secondary development.

After enabling the secure runtime library protection function, the user code can call the function in the secure runtime library area normally, but cannot read out the content of the secure runtime library area by any means (CPU, DMA, SWD, ISP), and the secure runtime library protection function can be disabled through the ISP protocol and the data in the secure runtime library storage space can be cleared at the same time only when a correct password is provided.

The method to enable the Security Runtime Vault Protection function is as follows: Write a specific control word at address 0xFFFF0~0xFFFF as shown in the table below:

**Table 7-4 Enable security runtime library protection function control word**

|                     |          |          |          |          |          |          |          |          |
|---------------------|----------|----------|----------|----------|----------|----------|----------|----------|
| <b>Address</b>      | 0xFFFF0  | 0xFFFF1  | 0xFFFF2  | 0xFFFF3  | 0xFFFF4  | 0xFFFF5  | 0xFFFF6  | 0xFFFF7  |
| <b>Control word</b> | SlibKey0 | SlibKey1 | SlibKey2 | SlibKey3 | SlibKey4 | SlibKey5 | SlibKey6 | SlibKey7 |
| <b>Address</b>      | 0xFFFF8  | 0xFFFF9  | 0xFFFFA  | 0xFFFFB  | 0xFFFFC  | 0xFFFFD  | 0xFFFFE  | 0xFFFFF  |
| <b>Control word</b> | StartIdx | EndIdx   | 0x5A     | 0x5A     | 0xA5     | 0xA5     | CRCL     | CRCH     |

The meaning of each control word is as follows:

- SlibKey0~SlibKey7 is the password of the security runtime library, which is used to prohibit the protection function of the security runtime library, users should set it according to the actual demand and keep it in a safe place. Do not set the passwords to all FF and all 00.
- StartIdx is the start page number of the security runtime library area, users should set it according to the actual situation, after writing, users can read it out through the START bit field of the FLASH\_SDKCFR register.
- EndIdx is the end page number of the security runtime library area, fixed at 0x7F, after writing, the user can read out through the END bit field of the FLASH\_SDKCFR register.
- CRCL is the lower part of the result of CRC16\_X25 algorithm for 0xFFFF0~0xFFFFD data.
- CRCH is the high byte of the result of the CRC16\_X25 algorithm for 0xFFFF0~0xFFFFD data.

There are the following ways to disable the security runtime library protection function:

1. Execute chip erase operation through ISP protocol: when the chip receives a chip erase instruction and a correct security runtime library password, it will automatically erase all the data of the chip and disable the security runtime library protection function; when the chip receives a chip erase instruction and an incorrect security runtime library password, no operation is performed, the data in all areas are not erased, and the secure runtime library protection function is still enabled. Refer to the ISP programming documentation for instructions.



## 7.7 FLASH memory programming

The CW32L012 internal integrated FLASH memory supports FLASH erasing and programming through SWD or ISP. The SWD method is to use the debugging tool to erase and program the FLASH through the SWD interface. The ISP method is to erase and program the FLASH through the BootLoader code pre-installed before the chip leaves the factory. ISP provides fast and efficient in-circuit erasing and programming methods, please refer to ISP programming documentation.



## 7.8 Precautions

In order to operate FLASH correctly and improve the access efficiency and service life of FLASH, users need to pay attention to the following matters when programming applications:

### 1. Address Alignment Requirements

The address boundary is aligned, that is, the address when accessing FLASH with a 16-bit width must be an even address, and when using a 32-bit width, the address must be an address that is a multiple of 4. Code example for correct address alignment:

- 8bit read:

```
tempdata = *((uint8_t*) 0x0000 0001);
```

- 16bit read:

```
tempdata = *((uint16_t*) 0x0000 0002);
```

- 32bit read:

```
tempdata = *((uint32_t*) 0x0000 0004);
```

Code example with wrong address alignment:

- 16bit read:

```
tempdata = *((uint16_t*) 0x0000 0001);
```

- 32bit read:

```
tempdata = *((uint32_t*) 0x0000 0003);
```

### 2. Operation complete flag query

When the CPU fetches instructions from FLASH and runs, if the page erase/write operation to FLASH is performed, the CPU will automatically stop the next instruction access, and the hardware will automatically wait for the erase and write operation to complete (the FLASH\_ISR.BUSY status bit becomes 0), so the user program does not need to loop through the query operation completion flag to determine whether the operation is completed.

When the CPU fetches instructions from RAM and runs, if a page erase/write operation to FLASH is performed, the CPU will access the next instruction while performing the operation. In order to ensure the correct execution of the next instruction of the program, The user program must query the FLASH\_ISR.BUSY flag cyclically after the FLASH erase and write operations, until the FLASH\_ISR.BUSY flag becomes 0, and then the subsequent tasks can be performed.

Before entering DeepSleep mode, if the FLASH is performing an erase/write operation, you must wait for the FLASH\_ISR.BUSY flag bit to clear to 0, and make sure FLASH\_CR1.MODE is 0 at the same time.

### 3. Service life

Based on the characteristics of embedded FLASH, the number of operations and storage time of FLASH are limited. Users should try to avoid frequent erasing and writing operations on the FLASH memory of a certain page or address in the application program to ensure reliable data storage. Please refer to the datasheet for specific life data.

### 4. Data storage mode

The application of CW32L012 stipulates that data is stored in FLASH in little-endian mode, that is, the least significant byte data of the data is stored in the lowest byte address space.

### 5. Prefetching and caching configuration

The FLASH memory of CW32L012 has prefetch cache function, the cache and prefetch function of FLASH can be enabled by setting FLASH\_CR2.CACHE to 1 and FLASH\_CR2.FETCH to 1, respectively, so as to improve the access efficiency of MCU to FLASH.



## 7.9 List of registers

FLASH base address: FLASH\_BASE = 0x4002 2000

Table 7-5 List of FLASH registers

| Register name  | Register address  | Register description                   |
|----------------|-------------------|----------------------------------------|
| FLASH_CR1      | FLASH_BASE + 0x00 | Control register 1                     |
| FLASH_CR2      | FLASH_BASE + 0x04 | Control register 2                     |
| FLASH_PAGELOCK | FLASH_BASE + 0x08 | Erase lock register                    |
| FLASH_IER      | FLASH_BASE + 0x20 | Interrupt enable register              |
| FLASH_ISR      | FLASH_BASE + 0x24 | Interrupt flag register                |
| FLASH_ICR      | FLASH_BASE + 0x28 | Interrupt flag clear register          |
| FLASH_SDKCFR   | FLASH_BASE + 0x70 | Security runtime library area register |



## 7.10 Register description

See section [1 Documentation conventions](#) for abbreviations used in register descriptions.

### 7.10.1 FLASH\_CR1 Control register 1

Address offset: 0x00      Reset value: 0x0000 0000

| Bit field | Name     | Permission | Function description                                                                                                                                                                                                                                                                                                     |
|-----------|----------|------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:16     | KEY      | WO         | Only when KEY is 0x5A5A, the write operation of this register is valid.                                                                                                                                                                                                                                                  |
| 15:7      | RFU      | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                            |
| 6:5       | SECURITY | RO         | Current protection level<br>00: Level0, ISP can read and write, SWD can read and write.<br>01: Level1, ISP downgradable, SWD downgradable; Data cannot be read.<br>10: Level2, ISP downgradable, SWD has no function; Data cannot be read.<br>11: Level3, ISP has no function, SWD has no function; Data cannot be read. |
| 4:2       | RFU      | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                            |
| 1:0       | MODE     | RW         | Operation mode configuration<br>00: Read mode, Read.<br>01: Write mode, Program.<br>10: Page erase mode, PageErase.<br>11: Chip erase mode, ChipErase.                                                                                                                                                                   |



### 7.10.2 FLASH\_CR2 Control register 2

Address offset: 0x04      Reset value: 0x0000 0000

| Bit field | Name         | Permission | Function description                                                                                                                                                                                                                                                                            |
|-----------|--------------|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:16     | KEY          | WO         | Only when KEY is 0x5A5A, the write operation of this register is valid.                                                                                                                                                                                                                         |
| 15:6      | RFU          | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                   |
| 5         | CACHEINVALID | RW         | 0: No action.<br>1: Clear CACHE cache, CACHE does not work.<br><i>Note: Need software to write 1 and then 0 to clear CACHE cache.</i>                                                                                                                                                           |
| 4         | CACHE        | RW         | Cache enable control<br>0: Disable cache.<br>1: Enable cache.<br><i>Note: When WAIT is not 0, it is recommended to enable the function.</i>                                                                                                                                                     |
| 3         | FETCH        | RW         | Prefetch enable control<br>0: Disable prefetch.<br>1: Enable prefetch.<br><i>Note: When WAIT is not 0, it is recommended to enable the function.</i>                                                                                                                                            |
| 2:0       | WAIT         | RW         | FLASH fetch cycle configuration<br>000: 1 HCLK cycle for HCLK ≤ 24MHz.<br>001: 2 HCLK cycle for 24MHz < HCLK ≤ 48MHz.<br>010: 3 HCLK cycle for 48MHz < HCLK ≤ 72MHz.<br>011: 4 HCLK cycle for 72MHz < HCLK ≤ 96MHz.<br><i>Note: This control bit has the same function as SYSCTRL_CR2[6:4].</i> |



### 7.10.3 FLASH\_PAGELOCK Erase lock register

Address offset: 0x08

Reset value: 0x0000 0000

| Bit field | Name   | Permission | Function description                                                                           |
|-----------|--------|------------|------------------------------------------------------------------------------------------------|
| 31:16     | KEY    | WO         | Only when KEY is 0x5A5A, the write operation of this register is valid.                        |
| 15        | LOCK15 | RW         | Page120–Page127 erase lock configuration<br>0: Locked, non-rewritable.<br>1: Open, rewritable. |
| 14        | LOCK14 | RW         | Page112–Page119 erase lock configuration<br>0: Locked, non-rewritable.<br>1: Open, rewritable. |
| 13        | LOCK13 | RW         | Page104–Page111 erase lock configuration<br>0: Locked, non-rewritable.<br>1: Open, rewritable. |
| 12        | LOCK12 | RW         | Page96–Page103 erase lock configuration<br>0: Locked, non-rewritable.<br>1: Open, rewritable.  |
| 11        | LOCK11 | RW         | Page88–Page95 erase lock configuration<br>0: Locked, non-rewritable.<br>1: Open, rewritable.   |
| 10        | LOCK10 | RW         | Page80–Page87 erase lock configuration<br>0: Locked, non-rewritable.<br>1: Open, rewritable.   |
| 9         | LOCK9  | RW         | Page72–Page79 erase lock configuration<br>0: Locked, non-rewritable.<br>1: Open, rewritable.   |
| 8         | LOCK8  | RW         | Page64–Page71 erase lock configuration<br>0: Locked, non-rewritable.<br>1: Open, rewritable.   |
| 7         | LOCK7  | RW         | Page56–Page63 erase lock configuration<br>0: Locked, non-rewritable.<br>1: Open, rewritable.   |
| 6         | LOCK6  | RW         | Page48–Page55 erase lock configuration<br>0: Locked, non-rewritable.<br>1: Open, rewritable.   |
| 5         | LOCK5  | RW         | Page40–Page47 erase lock configuration<br>0: Locked, non-rewritable.<br>1: Open, rewritable.   |
| 4         | LOCK4  | RW         | Page32–Page39 erase lock configuration<br>0: Locked, non-rewritable.<br>1: Open, rewritable.   |



| Bit field | Name  | Permission | Function description                                                                         |
|-----------|-------|------------|----------------------------------------------------------------------------------------------|
| 3         | LOCK3 | RW         | Page24–Page31 erase lock configuration<br>0: Locked, non-rewritable.<br>1: Open, rewritable. |
| 2         | LOCK2 | RW         | Page16–Page23 erase lock configuration<br>0: Locked, non-rewritable.<br>1: Open, rewritable. |
| 1         | LOCK1 | RW         | Page8–Page15 erase lock configuration<br>0: Locked, non-rewritable.<br>1: Open, rewritable.  |
| 0         | LOCK0 | RW         | Page0–Page7 erase lock configuration<br>0: Locked, non-rewritable.<br>1: Open, rewritable.   |

#### 7.10.4 FLASH\_IER Interrupt enable register

Address offset: 0x20      Reset value: 0x0000 0000

| Bit field | Name     | Permission | Function description                                                                                     |
|-----------|----------|------------|----------------------------------------------------------------------------------------------------------|
| 31:5      | RFU      | -          | Reserved bits, please keep the default value.                                                            |
| 4         | PROG     | RW         | Programming error interrupt enable control<br>0: Disabled<br>1: Enabled                                  |
| 3         | CACHEON  | RW         | Erase and writer mode CACHE and FETCH errors interrupt enable control<br>0: Disabled<br>1: Enabled       |
| 2         | SDKERR   | RW         | Erase and writer secure runtime library area error interrupt enable control<br>0: Disabled<br>1: Enabled |
| 1         | PAGELOCK | RW         | Erase and write PAGELOCK locked page interrupt enable control<br>0: Disabled<br>1: Enabled               |
| 0         | PC       | RW         | Erase and write the page where the PC is located, interrupt enable control<br>0: Disabled<br>1: Enabled  |



### 7.10.5 FLASH\_ISR Interrupt flag register

Address offset: 0x24      Reset value: 0x0000 0000

| Bit field | Name     | Permission | Function description                                                                                                                                                                                                                                      |
|-----------|----------|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:6      | RFU      | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                             |
| 5         | BUSY     | RO         | Erase and write status flag<br>0: Erase and write operation completed.<br>1: Erase and write operation not completed.                                                                                                                                     |
| 4         | PROG     | RO         | Programming error flag<br>0: The content of each byte of the current address to be written is all 0xFF.<br>1: The content of each byte of the current address to be written is not all 0xFF.                                                              |
| 3         | CACHEON  | RO         | Erase and write mode CACHE and FETCH error flag<br>0: CACHE and FETCH have been disabled when the current erase/write operation is being executed.<br>1: CACHE and FETCH have not been disabled when the current erase/write operation is being executed. |
| 2         | SDKERR   | RO         | Erase and write secure runtime library area error interrupt flag<br>0: Current erase area does not include SDK area.<br>1: Current erase area includes SDK area.                                                                                          |
| 1         | PAGELOCK | RO         | Erase and write PAGELOCK locked page interrupt flag<br>0: The current erase address is outside the page locked by PAGELOCK.<br>1: The current erase address is within the page locked by PAGELOCK.                                                        |
| 0         | PC       | RO         | Erase and write the interrupt flag of the page where the PC pointer is located<br>0: The current erase address is outside the page where the PC pointer is located.<br>1: The current erase address is within the page where the PC pointer is located.   |



### 7.10.6 FLASH\_ICR Interrupt flag clear register

Address offset: 0x28      Reset value: 0x0000 001F

| Bit field | Name     | Permission | Function description                                                                                                                                                                              |
|-----------|----------|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:5      | RFU      | -          | Reserved bits, please keep the default value.                                                                                                                                                     |
| 4         | PROG     | R1W0       | Programming error flag cleared<br>W0: Clear programming error flag.<br>W1: No function.                                                                                                           |
| 3         | CACHEON  | R1W0       | Erase and writer mode CACHE and FETCH error flag cleared<br>W0: Clear the erase and writer mode CACHE and FETCH error flag.<br>W1: No function.                                                   |
| 2         | SDKERR   | R1W0       | Erase and writer security runtime library area error interrupt flag cleared<br>W0: Clear the erase and writer security runtime library area error interrupt flag.<br>W1: No function.             |
| 1         | PAGELOCK | R1W0       | Erase and write PAGELOCK locked page interrupt flag cleared<br>W0: Clear the page interrupt flag locked by erasing PAGELOCK.<br>W1: No function.                                                  |
| 0         | PC       | R1W0       | Erase and write the page where the PC pointer is located interrupt flag cleared<br>W0: Clear the erase and writer interrupt flag of the page where the PC pointer is located.<br>W1: No function. |

### 7.10.7 FLASH\_SDKCFR Security runtime library area register

Address offset: 0x70      Reset value: 0x0000 007F

| Bit field | Name  | Permission | Function description                                                                                  |
|-----------|-------|------------|-------------------------------------------------------------------------------------------------------|
| 31:15     | RFU   | -          | Reserved bits, please keep the default value.                                                         |
| 14:8      | END   | RO         | The Flash Page where the end address of the security runtime library area is indexed from 0 to 127.   |
| 7         | RFU   | -          | Reserved bits, please keep the default value.                                                         |
| 6:0       | START | RO         | The Flash Page where the start address of the security runtime library area is indexed from 0 to 127. |



## 8 Direct memory access (DMA)

### 8.1 Overview

CW32L012 supports direct memory access (DMA), which enables high-speed data transmit between peripherals and memory, between peripherals and peripherals, and between memory and memory without CPU intervention. The priority arbiter inside the DMA controller can realize the arbitration of DMA and CPU control of peripheral bus, as well as the scheduling execution between multiple DMA channels.





## 8.2 Main features

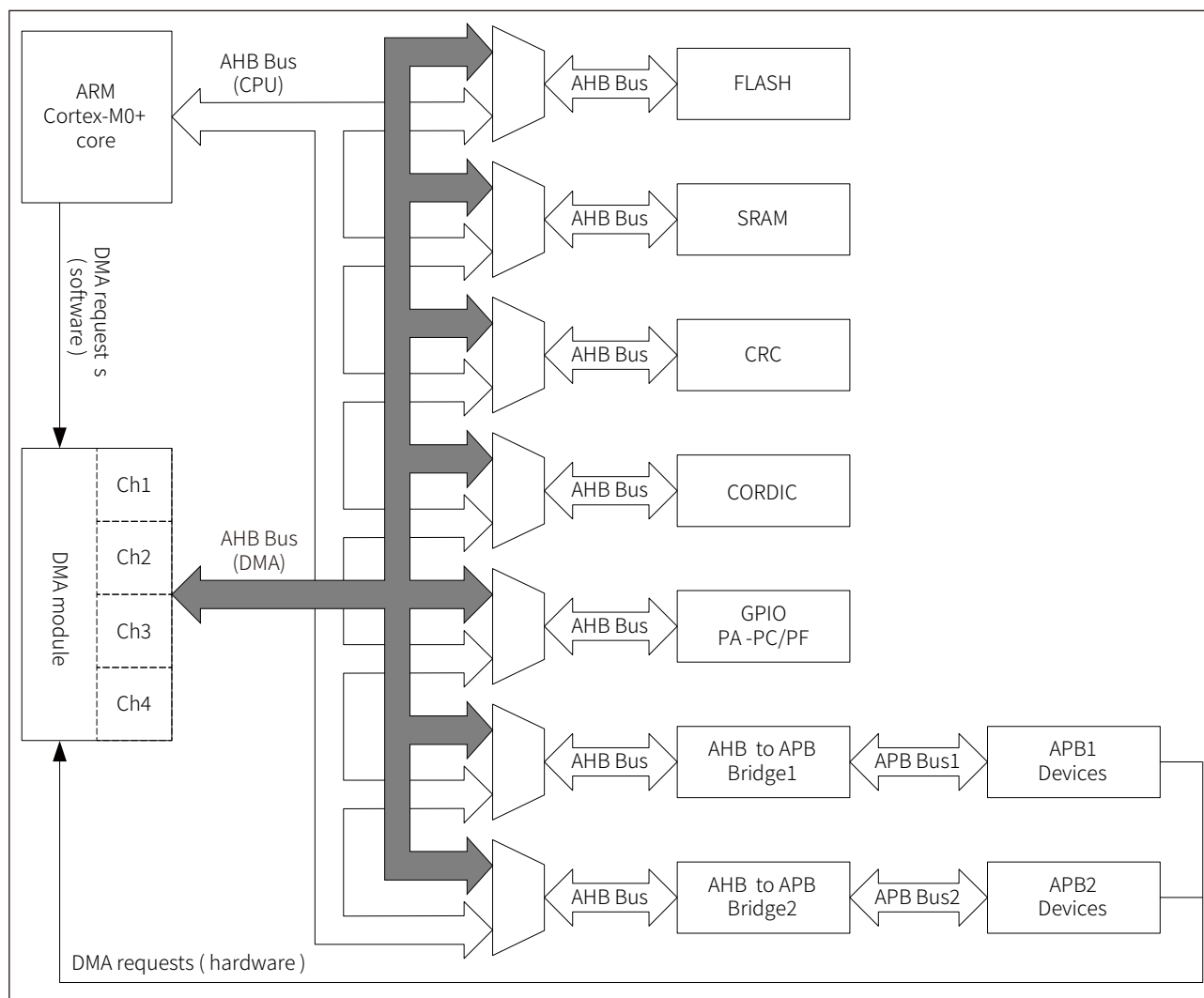
- 4 independent DMA channels.
- Supports automatic repeat.
- 3 data transmit widths: 8bit, 16bit, 32bit.
- 4 transmit modes: Software BLOCK, software BULK, hardware BLOCK, hardware BULK.
- Source address and destination address type: peripherals, memory.
- Address increment method: fixed, auto-increasing.
- Number of data blocks to be transmitted: 1~65535.
- Addressing range: 0x0000 0000~0xFFFF FFFF.



### 8.3 Functional block diagram

The AHB bus of the DMA controller is connected to the AHB bus of the peripheral through the bus matrix to realize the data interconnection between the DMA and the peripheral, including FLASH, SRAM, CRC, CORDIC, GPIO and various APB devices. The functional block diagram of the DMA controller is shown in the following figure:

Figure 8-1 DMA functional block diagram



- DMA controller  
4 independent DMA channels, the channel priority is bound to the channel number, the smaller the channel number, the higher the priority, and the larger the channel number, the lower the priority.
- Bus Matrix  
Both the CPU system bus and the DMA bus interface are connected to the bus matrix, and are switched and connected to peripherals through the bus matrix. When the CPU and DMA access different AHB bus devices or AHB to APB bridges, data transmit can be performed simultaneously; When the CPU and DMA access the same AHB device at the same time (different devices under the AHB to APB bridge are also considered to be the same AHB device), the CPU has a higher priority than the DMA.
- AHB to APB bridge  
DMA can achieve access to APB devices through the bus matrix and AHB to APB bridges. All APB devices under the same AHB-to-APB bridge share the same AHB bus and are therefore regarded as the same AHB device. When the CPU and DMA access the APB device under the same AHB to APB bridge at the same time (whether accessing the same APB device or a different APB device), the priority of the CPU is higher than that of the DMA.



## 8.4 DMA transmit modes

### 8.4.1 Overview of transmit mode

DMA supports 4 transmit modes: software-triggered BLOCK transmit mode, software-triggered BULK transmit mode, hardware-triggered BLOCK transmit mode, and hardware-triggered BULK transmit mode.

- Software and hardware triggering

Some peripherals support hardware DMA. When they are configured as the trigger source of the DMA channel, a DMA request (DMA request) can be generated, and the hardware triggers the DMA transmit without CPU participation. Peripherals that do not support hardware DMA can only be configured as software-triggered DMA transmits.

The trigger mode is controlled by the TYPE bit field of the DMA trigger source control register DMA\_TRIGy (y is the DMA channel number, y=1~4, the same below), TYPE is 0 for software trigger mode, TYPE is 1 for hardware trigger mode.

- Transmit modes of BLOCK and BULK

The difference between BLOCK and BULK transmit modes is whether the transmit is allowed to be interrupted by a higher priority transmit request (high priority DMA or CPU). The TRANS bit field of the DMA control and status register DMA\_CSRy is 1 for the BLOCK transmit mode, and 0 for the BULK transmit mode.

In BULK transmit mode, all data transmit will be completed at one time. No transmit gaps are inserted during data transmit, and the transmit process is not interrupted by CPU or other high-priority DMA channel transmit requests.

In the BLOCK transmit mode, a transmit gap is inserted after each data block is transmitted, and the arbiter performs transmit priority arbitration in the transmit gap. If a CPU or higher priority DMA channel wants to access the peripheral occupied by the current DMA channel, the arbiter will release the control right of the current DMA channel to the device, allowing the CPU or higher priority DMA channel to have priority access, etc. After completion, resume the previously interrupted DMA channel transmit.

The transmit modes of DMA as shown in the following table:

Table 8-1 DMA transmit mode

| Transfer mode  | DMA_TRIGy.TYPE | DMA_CSRy.TRANS | Can it be interrupted by a higher priority DMA channel or CPU? |
|----------------|----------------|----------------|----------------------------------------------------------------|
| Software BLOCK | 0              | 1              | YES                                                            |
| Software BULK  | 0              | 0              | NO                                                             |
| Hardware BLOCK | 1              | 1              | YES                                                            |
| Hardware BULK  | 1              | 0              | NO                                                             |



### DMA transmit process

For a complete DMA transmit process, the user needs to do the following settings:

- Step 1:** Set the transmit mode, DMA\_CSRy.TRANS is set to 1 for the BLOCK transmit mode, and 0 for the BULK transmit mode.
- Step 2:** Set the trigger mode, DMA\_TRIGy.TYPE is set to 0 for software trigger, and 1 is for hardware trigger mode.
- Step 3:** Configure DMA channel data transmit, bit width, DMA\_CSRy.SIZE, select 8, 16, or 32bit.
- Step 4:** Configure the number of DMA channels to transmit data blocks, DMA\_CNTy.CNT, the valid range is 1~65535.
- Step 5:** Configure the DMA channel transmit data block size, DMA\_CNTy.REPEAT, the valid range is 1~15.
- Step 6:** Configure the DMA transmit source address, DMA\_SRCADDRy.
- Step 7:** Configure the DMA transmit destination address, DMA\_DSTADDRy.
- Step 8:** Configure the DMA source address increment method, DMA\_CSRy.SRCINC, fixed or automatically incremented.
- Step 9:** Configure the DMA destination address increment method, DMA\_CSRy.DSTINC, fixed or automatically increased.
- Step 10:** Configures whether the source address is reloaded on transmission completion, DMA\_CSRy.SRCLOAD.
- Step 11:** Configure whether the destination address is reloaded on transmission completion, DMA\_CSRy.DSTLOAD.
- Step 12:** Configure automatic repeat, DMA\_CSRy.RESTART.
- Step 13:** Set DMA\_CSRy.EN to 1 to enable the corresponding DMA channel.
- Step 14:** For software-triggered DMA transmit, set DMA\_TRIGy.SOFTSRC to start DMA transmit. For hardware-triggered DMA transmit, no setting is required, but only the peripherals used to start DMA need to be set correctly.



### 8.4.2 Software trigger BLOCK transmit mode

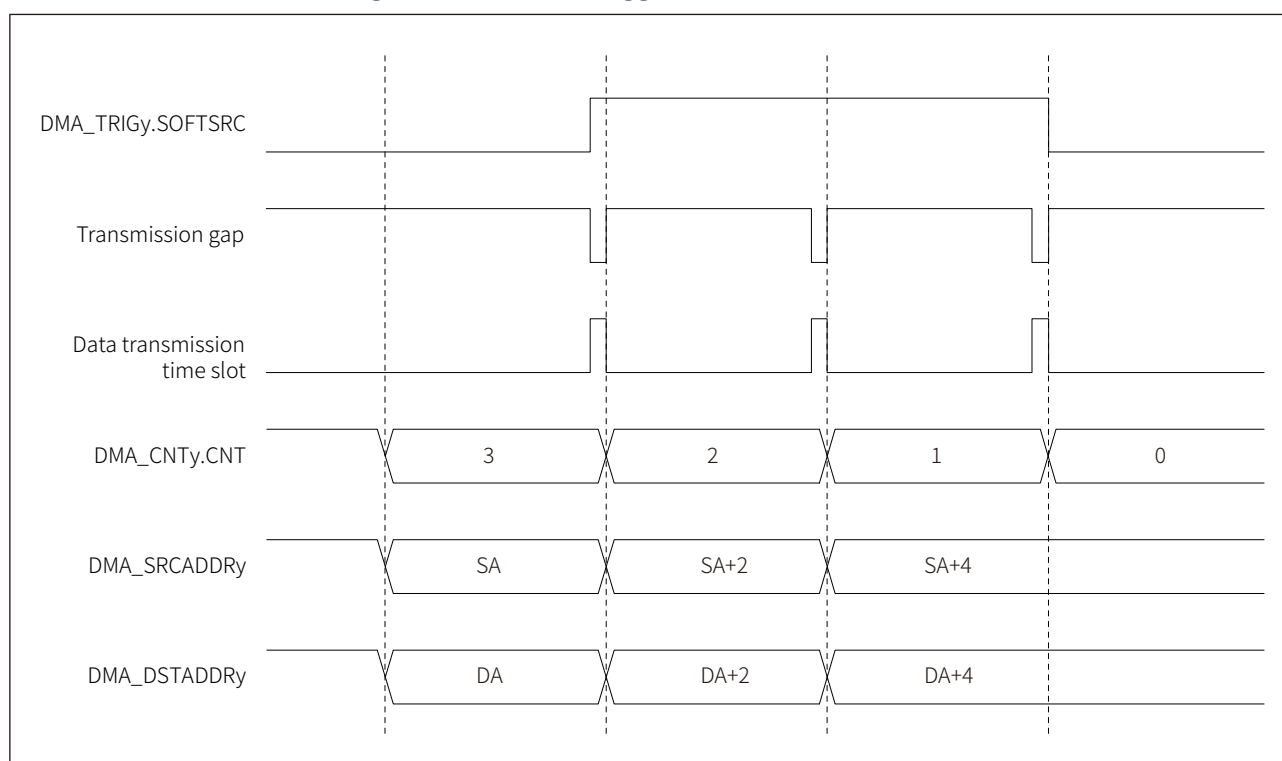
Setting DMA\_TRIGy.SOFTSRC to 1 will immediately trigger the DMA channel for data transfer.

Each time a DMA channel completes a data block transmission, the DMA inserts a transmission gap, and the arbiter performs transmission priority arbitration in the transmission gap. If there is a CPU or higher priority DMA request to access the AHB device occupied by the current DMA channel during the transmission gap, the arbiter will release the control right of the current DMA channel to the device, allowing the CPU or higher priority DMA channel to have priority access; when the CPU or a higher priority DMA channel releases the device access right, the unfinished data transfer of this DMA channel will continue.

DMA\_TRIGy.SOFTSRC is automatically cleared after the transfer is completed.

The software triggers the BLOCK transmission mode, and transmits 3 data with a data width of 16-bit (DMA\_CNTy.REPEAT = 1, DMA\_CNTy.CNT = 3), where SA represents the source address, DA represents the destination address, and both the source address and the destination address are addresses. In auto-increment mode, the transmission process is shown in the following figure:

Figure 8-2 Software trigger BLOCK transmit mode



### 8.4.3 Software trigger BULK transmit mode

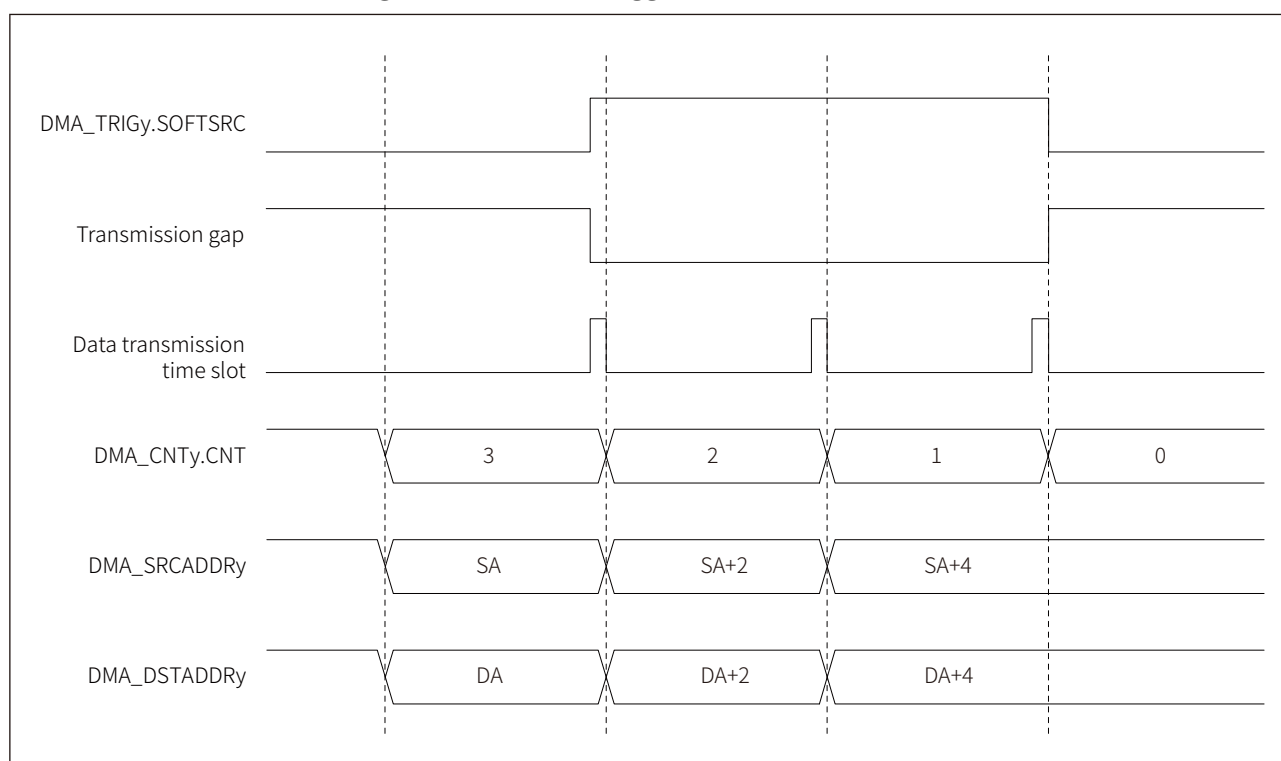
Setting DMA\_TRIGy.SOFTSRC to 1 will immediately trigger DMA for data transfer. After the DMA channel transmits DMA\_CNTy.CNT data blocks, stop the transfer and release the peripheral occupancy rights. After the transfer is completed, DMA\_TRIGy.SOFTSRC is automatically cleared to zero.

Software-triggered BULK transfer mode has no transfer gap. Before the DMA\_CNTy.CNT data block transfer is completed, the DMA channel will always occupy the peripheral. The CPU or higher priority DMA channel can only access the peripheral after the DMA transfer is completed.

In user applications, it is recommended not to transfer too much data at a time, otherwise the CPU may be completely unable to access the peripherals occupied by the DMA channel for a period of time.

The software triggers the BULK transfer mode, and transmits 3 data with a data width of 16-bit (DMA\_CNTy.REPEAT=1, DMA\_CNTy.CNT=3), where SA represents the source address, DA represents the destination address, and both the source address and the destination address are addresses. In auto-increment mode, the transmission process is shown in the following figure:

Figure 8-3 Software trigger BULK transmit mode



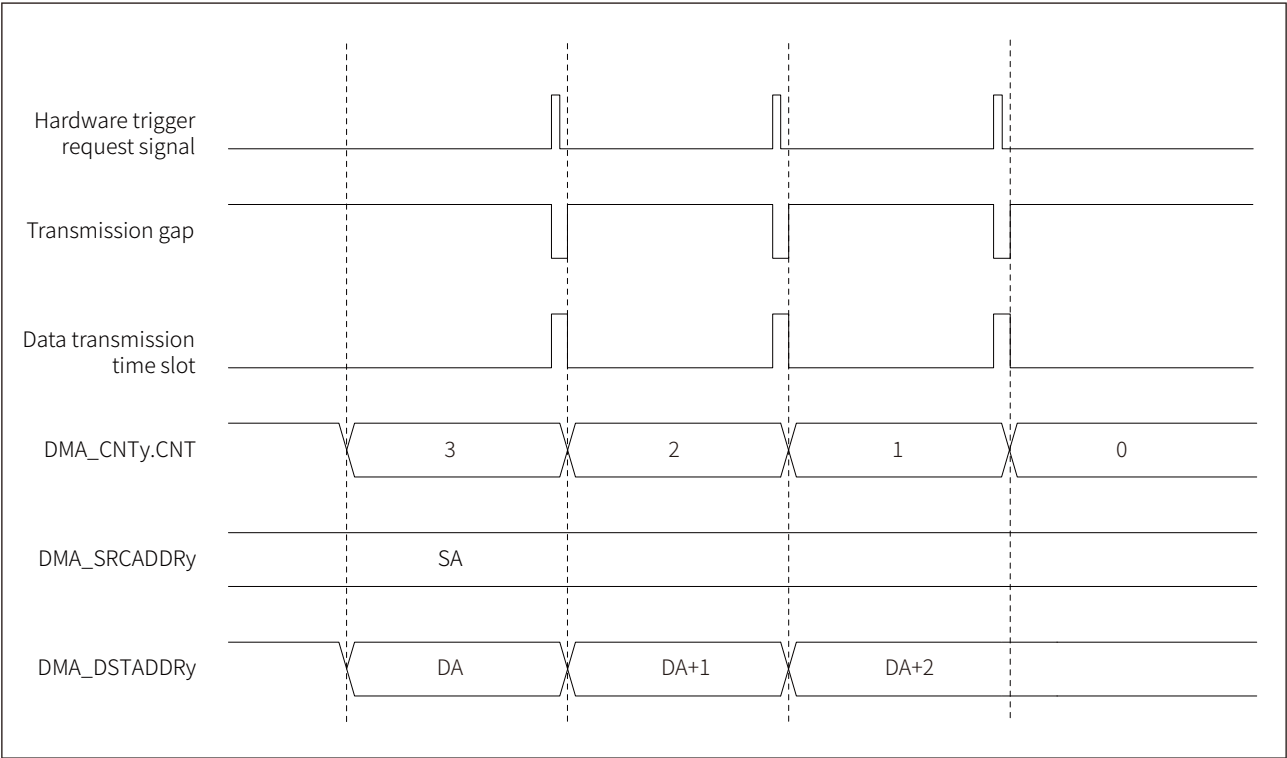
8.4.4 Hardware trigger BLOCK transmit mode

When the peripheral corresponding to the DMA channel generates a hardware trigger request signal, the DMA will transmit one data block, and then wait for the next hardware trigger request signal until all the data volume configured in the DMA controller is transmitted. For the hardware trigger request signal supported by the DMA channel, please refer to the HARDSRC bit field in section 8.8.4 DMA\_TRIGy Channel y trigger source control register.

Like the software BLOCK transmission mode, DMA will insert a transmission gap while the hardware BLOCK transmission mode a data block is completed, allowing the CPU or higher-priority DMA channel transmission requests to preempt peripheral control.

The hardware triggers the BLOCK transmission mode, and transmits 3 data with a data width of 8-bit (DMA\_CNTy.REPEAT=1, DMA\_CNTy.CNT=3), where SA represents the source address, DA represents the destination address, the source address is fixed, and the destination address is incremented automatically, the transmission process is shown in the following figure:

Figure 8-4 Hardware trigger BLOCK transfer mode





### 8.4.5 Hardware trigger BULK transmit mode

When the peripheral corresponding to the DMA channel generates a hardware trigger request signal, it will immediately trigger the DMA for data transmission. After the DMA channel transmits DMA\_CNTy.CNT data blocks, the transmission will be stopped and the peripheral occupancy will be released.

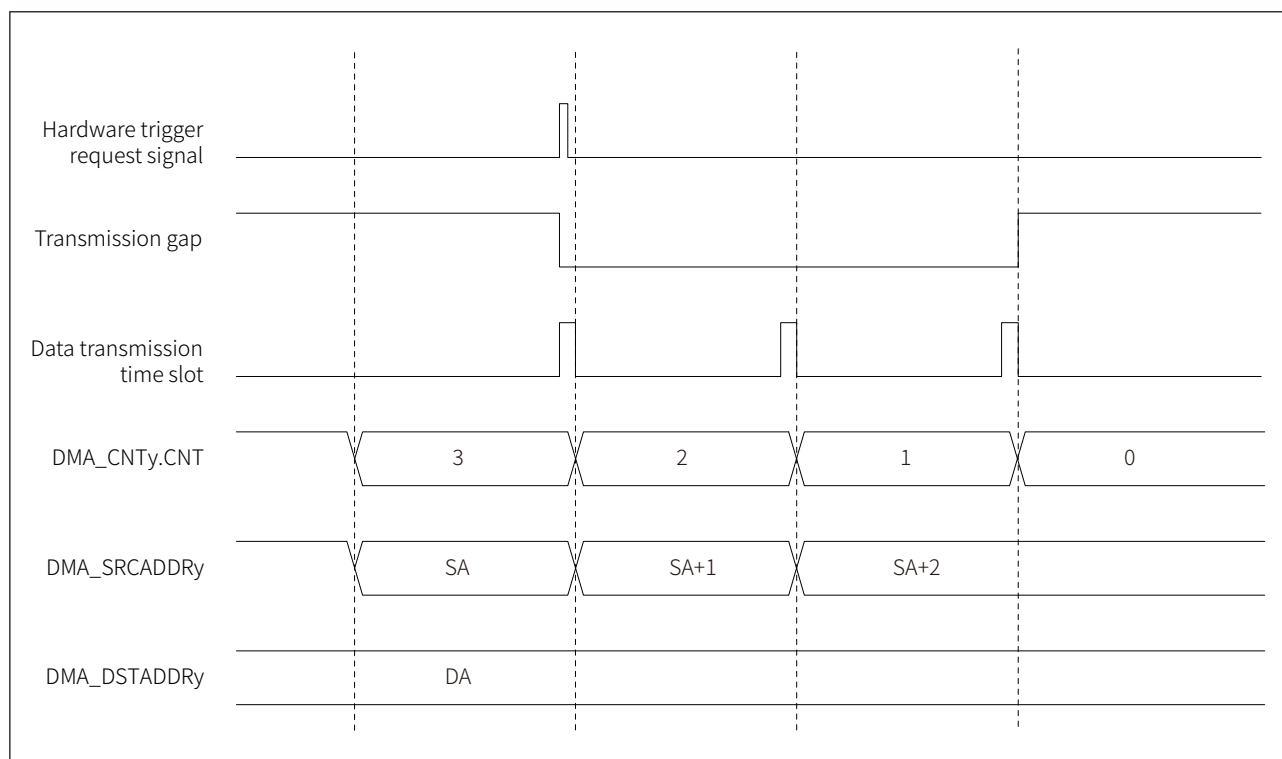
There is no transmission gap in the hardware-triggered BULK transfer mode. Before the transfer of DMA\_CNTy.CNT data block is completed, the DMA will always occupy the peripheral access right. The CPU or higher priority DMA channel can only access the peripheral after the operation is completed.

In user applications, it is recommended not to transfer too much data at a time, otherwise the CPU may be completely unable to access the peripherals occupied by the DMA channel for a period of time.

For the hardware trigger request signal supported by the DMA channel, please refer to HARDSRC bit field in section [8.8.4 DMA\\_TRIGy Channel y trigger source control register](#).

The hardware triggers the BULK transmission mode, and transmits 3 data with a data width of 8-bit (DMA\_CNTy.REPEAT=1, DMA\_CNTy.CNT=3), where SA represents the source address, DA represents the destination address, the source address is auto-increasing, and the destination address is fixed, the transmission process is shown in the following figure:

Figure 8-5 Hardware trigger BULK transmit mode



## 8.5 Other configuration of DMA

### 8.5.1 Data width

The data bit width can be set to 8bit, 16bit, 32bit, which can be set through the SIZE bit field of the DMA\_CSRy register. The data bit width should be exactly the same as the bit width of the source address and destination address of the DMA channel.

### 8.5.2 Data block size

The data block size is configured through the REPEAT bit field of the DMA\_CNTy register, it is recommended to write 1.

*Attention: This bit field value will automatically become 0 after each transmission is completed, and 1 needs to be written again before the next DMA transmission.*

### 8.5.3 Number of data blocks

The number of data blocks to be transferred can be set from 1 to 65535, which is configured through the CNT bit field of the DMA\_CNTy register.

### 8.5.4 Channel priority

The priority of the 4 DMA channels is bound to the channel number. The smaller the channel number, the higher the priority, and the larger the channel number, the lower the priority.

### 8.5.5 Channel auto repeat

The DMA supports auto repeat, i.e., the transmit is started again after the transmission is completed according to the initial configuration, which is configured through the RESTART bit field of the DMA\_CSRy register.

Upon completion of each data block transfer, the source and destination addresses can be reloaded, as set through the SRCLOAD and DSTLOAD bit fields of the DMA\_CSRy register, respectively.



## 8.6 DMA interrupt

The DMA channel can generate 2 interrupt flags during the transfer process: the transfer error interrupt flag and the transfer completion interrupt flag.

The interrupts of different DMA channels are independent, and the interrupt flag of each channel can be obtained through the interrupt flag register DMA\_ISR or the channel y control and status register DMA\_CSRy. The channel transmission error interrupt flag corresponds to multiple possible causes. The specific causes need to query the DMA\_CSRy.STATUS status bit, as shown in the following table:

Table 8-2 DMA interrupt flag status bits

| DMA_CSRy.STATUS status bit | DMA current state of channel y                                            |
|----------------------------|---------------------------------------------------------------------------|
| 000                        | Initial state.                                                            |
| 001                        | Transmission error, transmission address out of addressing range.         |
| 010                        | Transmission error, aborted by transmission stop request.                 |
| 011                        | Transmission error, error accessing the transmission source address.      |
| 100                        | Transmission error, error accessing the transmission destination address. |
| 101                        | Transfer completed.                                                       |

When the transmission error interrupt flag or the transmission completion interrupt flag is set, if the corresponding interrupt request is enabled, the CPU will execute the interrupt service routine. Before exiting the interrupt service routine, the corresponding flag should be cleared to avoid repeatedly entering the interrupt service routine.

The DMA interrupt trigger conditions, interrupt enable and interrupt flag clearing methods are shown in the following table:

Table 8-3 DMA interrupt flag and clear method

| DMA interrupt type     | Triggering conditions                                                                                                                                                                                  | Interrupt enable method | Interrupt flag clearing method |
|------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------|--------------------------------|
| Transmission error     | The transmission address is out of the addressing range;<br>Or, the peripheral requests to abort the DMA transmission;<br>Or, there is an error accessing the transmission source/destination address. | Set DMA_CSRy.TEIE to 1  | Set DMA_ICR.TEy to 0           |
| Transmission completed | The data transmission is all done correctly                                                                                                                                                            | Set DMA_CSRy.TCIE to 1  | Set DMA_ICR.TCy to 0           |



## 8.7 List of registers

DMA base address: DMA\_BASE = 0x4002 0000

Table 8-4 List of DMA registers

| Register name | Register address         | Register description                                             |
|---------------|--------------------------|------------------------------------------------------------------|
| DMA_ISR       | DMA_BASE + 0x00          | Interrupt flag register                                          |
| DMA_ICR       | DMA_BASE + 0x04          | Interrupt flag clear register                                    |
| DMA_CSRy      | DMA_BASE + 0x00 + 32 × y | Channel y control and status register, y = 1~4                   |
| DMA_CNTy      | DMA_BASE + 0x04 + 32 × y | Channel y transmit number register, y = 1~4                      |
| DMA_SRCADDRy  | DMA_BASE + 0x08 + 32 × y | Channel y transmit source address register, y = 1~4              |
| DMA_DSTADDRy  | DMA_BASE + 0x0C + 32 × y | Channel y transmit destination address register, y = 1~4         |
| DMA_TRIGy     | DMA_BASE + 0x10 + 32 × y | Channel y trigger source control register, y = 1~4               |
| DMA_CCNTy     | DMA_BASE + 0x14 + 32 × y | Channel y current remaining transmit count register, y = 1~4     |
| DMA_CSRCADDRy | DMA_BASE + 0x18 + 32 × y | Channel y current transmit source address register, y = 1~4      |
| DMA_CDSTADDRy | DMA_BASE + 0x1C + 32 × y | Channel y current transmit destination address register, y = 1~4 |



## 8.8 Register description

See section [1 Documentation conventions](#) for abbreviations used in register descriptions.

### 8.8.1 DMA\_ISR Interrupt flag register

Address offset: 0x00      Reset value: 0x0000 0000

| Bit field | Name | Permission | Function description                                                                      |
|-----------|------|------------|-------------------------------------------------------------------------------------------|
| 31:14     | RFU  | -          | Reserved bits, please keep the default value.                                             |
| 13        | TE4  | RO         | Channel 4 transmission error flag, see TE1 description.                                   |
| 12        | TC4  | RO         | Channel 4 transmission complete flag, see TC1 description.                                |
| 11:10     | RFU  | -          | Reserved bits, please keep the default value.                                             |
| 9         | TE3  | RO         | Channel 3 transmission error flag, see TE1 description.                                   |
| 8         | TC3  | RO         | Channel 3 transmission complete flag, see TC1 description.                                |
| 7:6       | RFU  | -          | Reserved bits, please keep the default value.                                             |
| 5         | TE2  | RO         | Channel 2 transmission error flag, see TE1 description.                                   |
| 4         | TC2  | RO         | Channel 2 transmission complete flag, see TC1 description.                                |
| 3:2       | RFU  | -          | Reserved bits, please keep the default value.                                             |
| 1         | TE1  | RO         | Channel 1 transmission error flag<br>0: Normal transmission.<br>1: Transmission error.    |
| 0         | TC1  | RO         | Channel 1 transmission complete flag<br>0: Transmit not complete<br>1: Transmit completed |



### 8.8.2 DMA\_ICR Interrupt flag clear register

Address offset: 0x04

Reset value: 0xFFFF FFFF

| Bit field | Name | Permission | Function description                                                                                                 |
|-----------|------|------------|----------------------------------------------------------------------------------------------------------------------|
| 31:14     | RFU  | -          | Reserved bits, please keep the default value.                                                                        |
| 13        | TE4  | R1W0       | Channel 4 transmission error flag clear, see TE1 description.                                                        |
| 12        | TC4  | R1W0       | Channel 4 transmission complete flag cleared, see TC1 description.                                                   |
| 11:10     | RFU  | -          | Reserved bits, please keep the default value.                                                                        |
| 9         | TE3  | R1W0       | Channel 3 transmission error flag clear, see TE1 description.                                                        |
| 8         | TC3  | R1W0       | Channel 3 transmission complete flag cleared, see TC1 description.                                                   |
| 7:6       | RFU  | -          | Reserved bits, please keep the default value.                                                                        |
| 5         | TE2  | R1W0       | Channel 2 transmission error flag clear, see TE1 description.                                                        |
| 4         | TC2  | R1W0       | Channel 2 transmission complete flag cleared, see TC1 description.                                                   |
| 3:2       | RFU  | -          | Reserved bits, please keep the default value.                                                                        |
| 1         | TE1  | R1W0       | Channel 1 transmission error flag clear<br>W0: Clear channel 1 transmission error flag.<br>W1: No function.          |
| 0         | TC1  | R1W0       | Channel 1 transmission complete flag cleared<br>W0: Clear channel 1 transmission completed flag.<br>W1: No function. |



### 8.8.3 DMA\_CSRy Channel y control and status register

Address offset:  $0x00 + 32 \times y$     Reset value:  $0x0000\ 0000$  ( $y = 1 \sim 4$ )

| Bit field | Name    | Permission | Function description                                                                                                                                                                                                                                                                                                                                                                                                      |
|-----------|---------|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:16     | RFU     | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                                                                                                             |
| 15        | DSTLOAD | RW         | Channel y destination address reload enable control at the completion of each data block (Repeat) transmit<br>0: Disabled<br>1: Enabled                                                                                                                                                                                                                                                                                   |
| 14        | SRCLOAD | RW         | Source address reload enable control at the completion of transmit of each data block (Repeat) of channel y<br>0: Disabled<br>1: Enabled                                                                                                                                                                                                                                                                                  |
| 13        | TE      | RO         | Channel y transmit error flag<br>0: Normal transmit<br>1: Transmit error                                                                                                                                                                                                                                                                                                                                                  |
| 12        | TC      | RO         | Channel y transmit completion flag<br>0: Transmit not completed<br>1: Transmit completed                                                                                                                                                                                                                                                                                                                                  |
| 11        | RESTART | RW         | Channel y auto repeat enable control<br>0: Disabled.<br>1: Enabled to start transmit again according to the initial configuration after the transmit is completed.                                                                                                                                                                                                                                                        |
| 10:8      | STATUS  | RW         | Current state of channel y<br>000: Initial state.<br>001: Transmission error, transmission address out of addressing range.<br>010: Transmission error, aborted by transmission stop request.<br>011: Transmission error, error accessing the transmission source address.<br>100: Transmission error, error accessing the transmission destination address.<br>101: Transfer completed.                                  |
| 7:6       | SIZE    | RW         | Channel y transmission data bit width and address from the incremental configuration<br>00: The bit width is 8bit and the auto-increment is 1.<br>01: The bit width is 16bit and the auto-increment is 2.<br>10: The bit width is 32bit and the auto-increment is 4.<br><b>Attention: Bit width configuration is only valid for memory (FLASH, RAM). Since the auto-increment configuration works on all peripherals.</b> |
| 5         | DSTINC  | RW         | Channel y destination address incremental configuration<br>0: Fixed destination address.<br>1: Destination address auto increment.                                                                                                                                                                                                                                                                                        |



| Bit field | Name   | Permission | Function description                                                                                                |
|-----------|--------|------------|---------------------------------------------------------------------------------------------------------------------|
| 4         | SRCINC | RW         | Channel y source address incremental configuration<br>0: Source address fixed.<br>1: Source address auto increment. |
| 3         | TRANS  | RW         | Channel y transfer mode configuration<br>0: Bulk transfer (BULK).<br>1: Block transfer (BLOCK).                     |
| 2         | TEIE   | RW         | Channel y transmission error interrupt enable control<br>0: Disabled<br>1: Enabled                                  |
| 1         | TCIE   | RW         | Channel y transfer completion interrupt enable control<br>0: Disabled<br>1: Enabled                                 |
| 0         | EN     | RW         | Channel y enable control<br>0: Disabled<br>1: Enabled                                                               |





### 8.8.4 DMA\_TRIGy Channel y trigger source control register

Address offset:  $0x10 + 32 \times y$     Reset value:  $0x0000\ 0000$  ( $y = 1 \sim 4$ )

| Bit field | Name    | Permission | Function description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|-----------|---------|------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:8      | RFU     | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 7:2       | HARDSRC | RW         | <p>Channel y hardware trigger signal trigger source configuration:</p> <p>000000: UART1 receive Buf non-null flag</p> <p>000001: UART1 transmit Buf null flag</p> <p>000010: UART2 receive Buf non-null flag</p> <p>000011: UART2 transmit Buf null flag</p> <p>000100: UART3 receive Buf non-null flag</p> <p>000101: UART3 transmit Buf null flag</p> <p>000110: SPI1 receive Buf non-null flag</p> <p>000111: SPI1 transmit Buf null flag</p> <p>001000: SPI2 receive Buf non-null flag</p> <p>001001: SPI2 transmit Buf null flag</p> <p>001010: SPI3 receive Buf non-null flag</p> <p>001011: SPI3 transmit Buf null flag</p> <p>001100: ADC1 sequence conversion complete event</p> <p>001101: ADC1 single conversion complete event</p> <p>001110: ADC2 sequence conversion complete event</p> <p>001111: ADC2 single conversion complete event</p> <p>010000: DAC1 DHR down overflow event</p> <p>010001: DAC2 DHR down overflow event</p> <p>010010: HALLTIM</p> <p>010011: BTIM1 update event</p> <p>010100: BTIM1 trigger event</p> <p>010101: BTIM2 update event</p> <p>010110: BTIM2 trigger event</p> <p>010111: BTIM3 update event</p> <p>011000: BTIM3 trigger event</p> <p>011001: GTIM1 trigger event</p> <p>011010: GTIM1 update event</p> <p>011011: GTIM1 capture/compare 1 event</p> <p>011100: GTIM1 capture/compare 2 event</p> <p>011101: GTIM1 capture/compare 3 event</p> <p>011110: GTIM1 capture/compare 4 event</p> <p>011111: GTIM2 trigger event</p> <p>100000: GTIM2 update event</p> <p>100001: GTIM2 capture/compare 1 event</p> <p>100010: GTIM2 capture/compare 2 event</p> <p>100011: GTIM2 capture/compare 3 event</p> <p>100100: GTIM2 capture/compare 4 event</p> <p>100101: GTIM3 trigger event</p> <p>100110: GTIM3 update event</p> <p>100111: GTIM3 capture/compare 1 event</p> |



| Bit field          | Name    | Permission | Function description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|--------------------|---------|------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 7:2<br>(continued) | HARDSRC | RW         | 101000: GTIM3 capture/compare 2 event<br>101001: GTIM3 capture/compare 3 event<br>101010: GTIM3 capture/compare 4 event<br>101011: GTIM4 trigger event<br>101100: GTIM4 update event<br>101101: GTIM4 capture/compare 1 event<br>101110: GTIM4 capture/compare 2 event<br>101111: GTIM4 capture/compare 3 event<br>110000: GTIM4 capture/compare 4 event<br>110001: ATIM update event<br>110010: ATIM capture/compare 1 event<br>110011: ATIM capture/compare 2 event<br>110100: ATIM capture/compare 3 event<br>110101: ATIM capture/compare 4 event<br>110110: ATIM capture/compare 5 event<br>110111: ATIM capture/compare 6 event<br>111000: ATIM COM event<br>111001: ATIM trigger event<br>111010: cordic_IDLE<br>111011: cordic_EOC<br>111100: I2C1_tx<br>111101: I2C1_rx<br>111110: I2C2_tx<br>111111: I2C2_rx |
| 1                  | SOFTSRC | RW         | Channel y software trigger control<br>R0: Transmit completed.<br>R1: Transmit in progressing.<br>W0: No function.<br>W1: Trigger the channel immediately to start the transmission.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| 0                  | TYPE    | RW         | Channel y trigger type configuration<br>0: In software trigger mode, the transmission is initiated by the SOFTSRC signal.<br>1: In hardware trigger mode, the transmission is initiated by the trigger signal of the trigger source configured by HARDSRC.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |



### 8.8.5 DMA\_CNTy Channel y transmit number register

Address offset:  $0x04 + 32 \times y$     Reset value:  $0x0000\ 0000$  ( $y = 1 \sim 4$ )

| Bit field | Name   | Permission | Function description                                                                                                                                                                                                                                                      |
|-----------|--------|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:20     | RFU    | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                             |
| 19:16     | REPEAT | RW         | Configure the size of the data block to be transmitted on channel y to REPEAT.<br><i>Note: REPEAT can only be set to 1 when the source or destination address can only provide or receive a piece of data every so often.</i><br><i>Note: REPEAT may not be set to 0.</i> |
| 15:0      | CNT    | RW         | Configure the amount of data block to be transmitted by channel y as CNT.<br><i>Note: The total amount of data to be transmitted is <math>CNT \times REPEAT</math>.</i>                                                                                                   |

### 8.8.6 DMA\_SRCADDRy Channel y transmit source address register

Address offset:  $0x08 + 32 \times y$     Reset value:  $0x0000\ 0000$  ( $y = 1 \sim 4$ )

| Bit field | Name    | Permission | Function description                   |
|-----------|---------|------------|----------------------------------------|
| 31:0      | SRCADDR | RW         | Channel y transmission source address. |

*Attention: The source address cannot be a DMA register, a FLASH register, or a RAM register.*

### 8.8.7 DMA\_DSTADDRy Channel y transmit destination address register

Address offset:  $0x0C + 32 \times y$     Reset value:  $0x0000\ 0000$  ( $y = 1 \sim 4$ )

| Bit field | Name    | Permission | Function description                        |
|-----------|---------|------------|---------------------------------------------|
| 31:0      | DSTADDR | RW         | Channel y transmission destination address. |

*Attention: The destination address cannot be a DMA register, a FLASH register, or a RAM register.*



### 8.8.8 DMA\_CCNTy Channel y current remaining transmit count register

Address offset:  $0x14 + 32 \times y$     Reset value:  $0x0000\ 0000$  ( $y = 1 \sim 4$ )

| Bit field | Name    | Permission | Function description                                                                                                                                                                                                                                                                                                                     |
|-----------|---------|------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:20     | RFU     | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                            |
| 19:16     | CREPEAT | RO         | The size of the current data block remaining for transmission on channel y is CREPEAT.                                                                                                                                                                                                                                                   |
| 15:0      | CCNT    | RO         | The number of data blocks currently remaining for transmission on channel y is CCNT.<br><i>Note: During the transmission process, each time a data block is transmitted, the value of this register decreases by 1.</i><br><i>Note: The total amount of data currently remaining to be transmitted is <math>CNT \times REPEAT</math></i> |

### 8.8.9 DMA\_CSRCADDRy Channel y current transmit source address register

Address offset:  $0x18 + 32 \times y$     Reset value:  $0x0000\ 0000$  ( $y = 1 \sim 4$ )

| Bit field | Name     | Permission | Function description                       |
|-----------|----------|------------|--------------------------------------------|
| 31:0      | CSRCADDR | RO         | Channel y current transmit source address. |

### 8.8.10 DMA\_CDSTADDRy Channel y current transmit destination address register

Address offset:  $0x1C + 32 \times y$     Reset value:  $0x0000\ 0000$  ( $y = 1 \sim 4$ )

| Bit field | Name     | Permission | Function description                                |
|-----------|----------|------------|-----------------------------------------------------|
| 31:0      | CDSTADDR | RO         | Channel y current transmission destination address. |



## 9 General-purpose input/output (GPIO)

### 9.1 Overview

The GPIO controller realizes the connection between various digital and analog circuits inside the chip and the physical pins.

GPIO can be configured as digital input and output and analog functions, supports peripheral function multiplexing, supports 2 interrupt sources of rising edge and falling edge, can wake up the MCU return to active mode through external interrupt in DeepSleep mode.



## 9.2 Main features

- All registers are read and written through the AHB bus interface.
- With digital input and output and analog function.
- Digital input and output support common GPIO and function multiplexing.
- The analog function can be used as the input signal of ADC, VC, LVD.
- Support a variety of internal clock signal output.
- Digital input supports internal pull-up and high-impedance modes.
- Digital outputs support push-pull and open-drain modes.
- The digital output supports atomic bit operations of bit setting, bit clearing, and bit flipping.
- Interrupt function supports rising edge, falling edge trigger mode.
- Interrupts with digital filtering, selectable from 7 clock sources.
- Supports wake-up of MCU via external interrupt in DeepSleep mode.

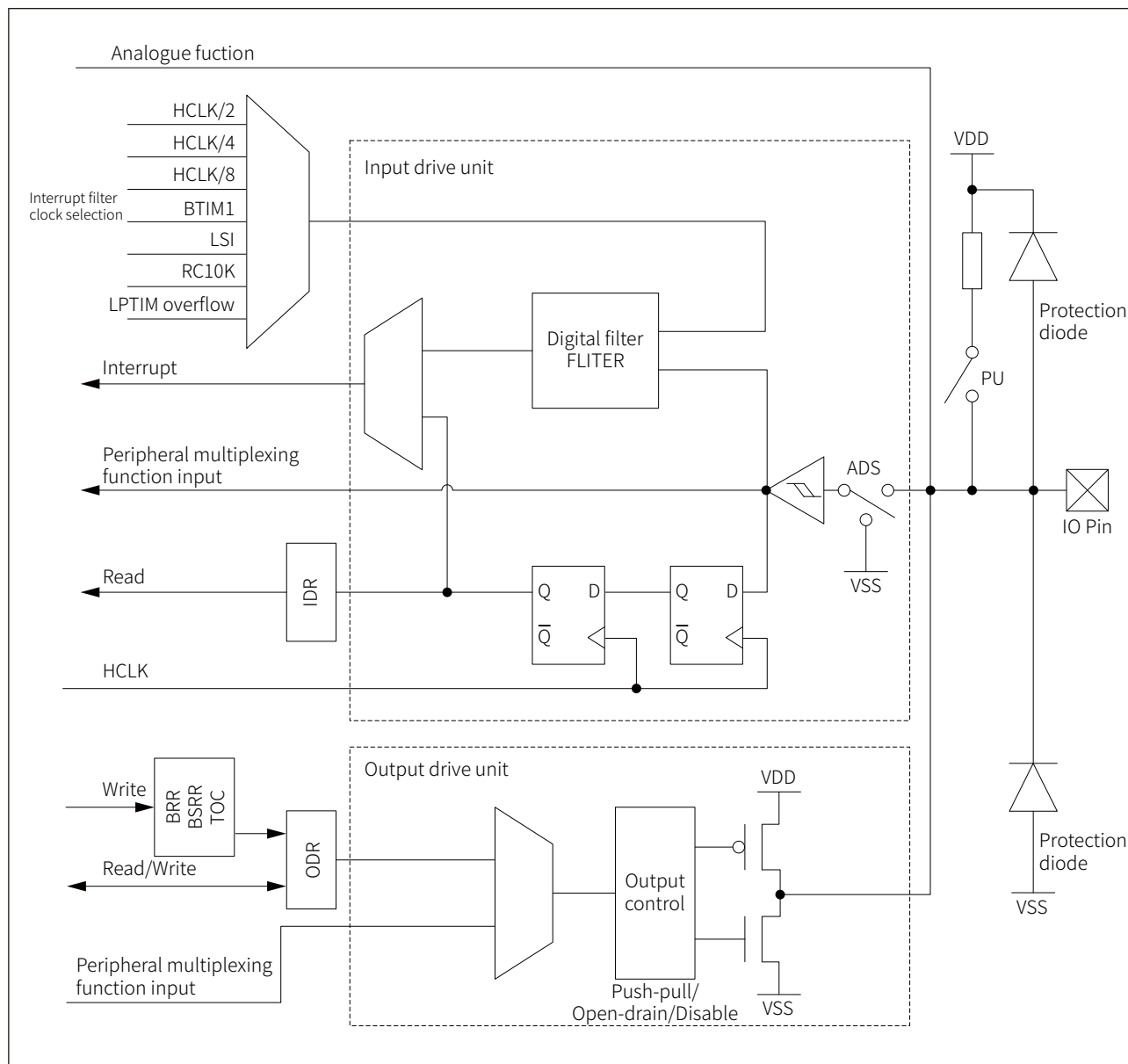


## 9.3 Functional description

### 9.3.1 Functional block diagram

The functional block diagram of the GPIO controller is shown in the following figure:

Figure 9-1 GPIO functional block diagram



### 9.3.2 Digital output

Clear the analog-digital configuration register GPIOx\_ANALOG[y] (x is the GPIO port number, x = A/B/C/F; y is the pin number, y = 0~15; the same below), and configure the corresponding GPIO port as Digital function; clear the input and output direction register GPIOx\_DIR[y] to configure the GPIO port as output mode. Digital output signal sources can be:

- Output data register GPIOx\_ODR
- On-chip digital peripherals

Configure the output mode through the output mode register GPIOx\_OPENDRAIN, which can select push-pull output or open-drain output.





### 9.3.3 Digital input

Clear the analog-digital configuration register `GPIOx_ANALOG[y]` to configure the GPIO port as a digital function; set the input and output direction register `GPIOx_DIR[y]` to configure the corresponding GPIO port as input mode. Digital input signals are configurable:

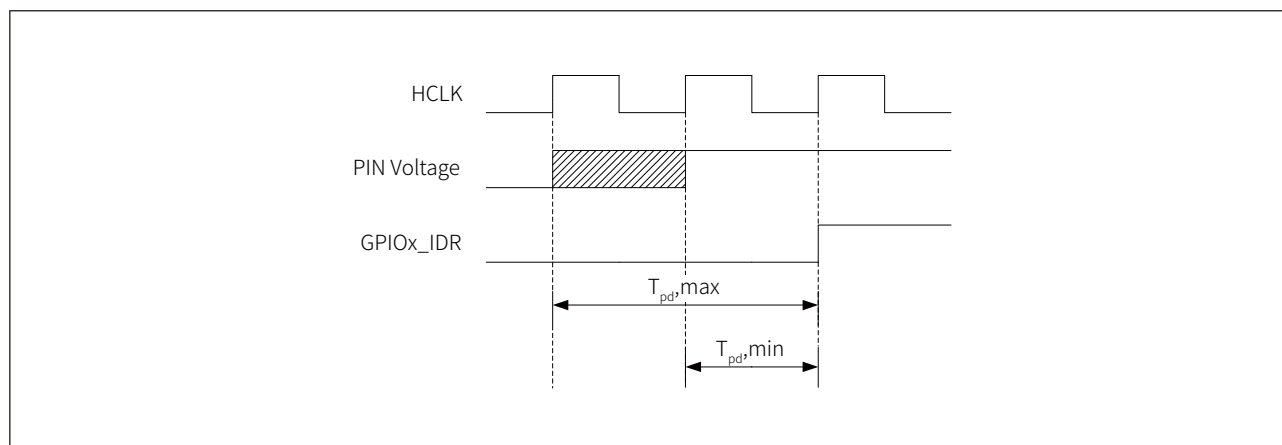
- Reach the input data register `GPIOx_IDR`.
- Reach on-chip digital peripherals.
- Trigger interrupt.

In this mode, the digital input signal is directed to the internal digital input circuit through the ADS switch.

After the level state is confirmed by the Schmitt trigger, it can be directly sent to the input of the digital peripheral pointed to by the on-chip multiplexing function, or through an HCLK-based synchronizer, present on the input data register `GPIOx_IDR[y]`.

The bits of the `GPIOx_IDR` register and the previous latch form a synchronizer, which can avoid the signal instability caused by the pin level jumping during the system clock change, but it will cause a certain read delay. The synchronization timing of the read port pins is shown in the following figure:

Figure 9-2 Read port pin synchronization timing



In the clock cycle after the rising edge of the system clock, the pin level signal will be latched in the internal register, as shown in the shaded part in the figure, after the next rising edge of the system clock, the stable pin level signal is read, and then a on the rising edge of the system clock, the data is latched into the `GPIOx_IDR` register. The signal delay  $T_{pd}$  is 1~2 system clocks.

The built-in hardware filter circuit can also be enabled if this input signal is considered for triggering an interrupt. The filter circuit is implemented based on a double D flip-flop synchronizer, which has 7 clock sources, some of which are specific to the low-power mode. For example, a key debounce operation without software intervention can be easily implemented. For specific clock source options and edge trigger options, see section [9.3.6 Interrupt function](#).

In this mode, the internal pull-up function can be individually selected to be turned on or off through the pull-up resistor register `GPIOx_PUR`.

### 9.3.4 Analogue function

For a GPIO port configured with analog function, you can open the GPIO analog signal channel by setting the analog digital configuration register GPIOx\_ANALOG[y] to 1. When the GPIO analog signal channel is turned on, the digital function of the port is turned off, the internal pull-up is disconnected, the internal digital input signal is shorted to VSS through the ADS switch, and the internal digital output function is disabled.



### 9.3.5 Multiplexing function

The multiplexing function of the input and output ports can be realized through the multiplexing function registers (GPIOx\_AFRH and GPIOx\_AFRL). Each 4-bit bit field in the multiplexing function register corresponds to the multiplexing function selection of a GPIO port, and up to 16 input and output signal targets can logically be selected. The specific definition of GPIO multiplexing function is shown in the following table:

Table 9-1 GPIO multiplexing function settings

| GPIOx_AFRL[4×y] | Multiplexing function | GPIOx_AFRH[4×(z-8)] | Multiplexing function |
|-----------------|-----------------------|---------------------|-----------------------|
| 0000            | GPIO                  | 0000                | GPIO                  |
| 0001            | AF1                   | 0001                | AF1                   |
| 0010            | AF2                   | 0010                | AF2                   |
| 0011            | AF3                   | 0011                | AF3                   |
| 0100            | AF4                   | 0100                | AF4                   |
| 0101            | AF5                   | 0101                | AF5                   |
| 0110            | AF6                   | 0110                | AF6                   |
| 0111            | AF7                   | 0111                | AF7                   |

Note: y, z are pin numbers: y = 0~7 corresponding to GPIOx\_AFRL; z = 8~15 corresponding to GPIOx\_AFRH.

The specific peripheral functions corresponding to each AF are shown in the following table:

Table 9-2 GPIO multiplexing function assignment table

| Pin name | Multiplexing function |           |           |            |            |            |            |           |            |              |
|----------|-----------------------|-----------|-----------|------------|------------|------------|------------|-----------|------------|--------------|
|          | AF0                   | AF1       | AF2       | AF3        | AF4        | AF5        | AF6        | AF7       | AF8        | AF9          |
| PA00     | GPIO                  | UART1_CTS | UART3_RXD | RTC_TAMP   | VC1_OUT    | BTIM3_TOGN | GTIM2_CH1  | ATIM_BK   | SPI2_MISO  | VC3_OUT      |
| PA01     | GPIO                  | UART1_RTS | UART3_TXD | RTC_OUT    | HEXEN      | BTIM3_TOGP | GTIM2_CH2  | ATIM_ETR  | SPI2_MOSI  |              |
| PA02     | GPIO                  | UART1_TXD | UART2_TXD | BTIM2_ETR  | VC2_OUT    | LPTIM_CH1  | GTIM2_CH3  | ATIM_CH4N | SPI2_SCK   |              |
| PA03     | GPIO                  | UART1_RXD | UART2_RXD | RTC_OUT    | PCLK_OUT   | LPTIM_CH2  | GTIM2_CH4  | ATIM_CH4  | SPI2_CS    | HALLTIM_CH1  |
| PA04     | GPIO                  | UART3_TXD | I2C1_SCL  | SPI1_CS    | MCO_OUT    | LPTIM_OUT  | ADC2_SAM   | ATIM_CH2N | GTIM3_CH1  | HALLTIM_CH2  |
| PA05     | GPIO                  | UART3_RXD | I2C1_SDA  | SPI1_SCK   | BTIM2_TOGN | LPTIM_ETR  | GTIM1_CH3  | ATIM_CH1  | GTIM3_CH2  | HALLTIM_CH3  |
| PA06     | GPIO                  | UART2_RXD | BTIM1_ETR | SPI1_MISO  | BTIM2_TOGP | LPTIM_CH1  | GTIM1_CH1  | ATIM_BK   | GTIM3_CH3  |              |
| PA07     | GPIO                  | UART2_TXD | GTIM2_ETR | SPI1_MOSI  | BTIM1_TOGN | LPTIM_CH2  | GTIM1_CH2  | ATIM_CH1N | GTIM3_CH4  |              |
| PA08     | GPIO                  | UART3_TXD | VC4_OUT   | IR_OUT     | MCO_OUT    | BTIM2_TOGP | LVD_OUT    | ATIM_CH1  | SPI3_CS    | GTIM4_ETR    |
| PA09     | GPIO                  | UART1_TXD | UART3_RXD | SPI1_CS    | I2C1_SCL   | LPTIM_ETR  | BTIM1_TOGN | ATIM_CH2  | SPI3_SCK   | GTIM4_CH1    |
| PA10     | GPIO                  | UART1_RXD | BTIM3_ETR | SPI1_SCK   | I2C1_SDA   | LPTIM_OUT  | BTIM1_TOGP | ATIM_CH3  | SPI3_MISO  | GTIM4_CH2    |
| PA11     | GPIO                  | UART1_CTS | VC1_OUT   | SPI1_MISO  | HEXEN      | ATIM_BK2   | GTIM2_CH2  | ATIM_CH4  | SPI3_MOSI  | GTIM4_CH3    |
| PA12     | GPIO                  | UART1_RTS | VC2_OUT   | SPI1_MOSI  | BTIM3_ETR  | ATIM_BK    | GTIM2_CH3  | ATIM_ETR  | GTIM3_ETR  | GTIM4_CH4    |
| PA13     | GPIO                  | UART1_RXD | UART3_TXD | UART2_RTS  | I2C1_SDA   | ATIM_BK    | IR_OUT     | ATIM_CH6  | ATIM_CH5N  |              |
| PA14     | GPIO                  | UART1_TXD | UART3_RXD | UART2_CTS  | I2C1_SCL   | ATIM_BK2   | GTIM1_ETR  | ATIM_CH6N | ATIM_CH5   |              |
| PA15     | GPIO                  | UART1_RXD | UART2_RXD | SPI1_CS    | ATIM_BKOUT | GTIM2_CH1  | GTIM2_ETR  | ATIM_CH1N | GTIM3_CH1  | SPI2_SCK     |
| PB00     | GPIO                  | UART1_RXD | UART2_TXD | HSIOSC_OUT | BTIM1_TOGP | ATIM_ETR   | GTIM1_CH3  | ATIM_CH2N | I2C2_SCL   | HALLTIM_OUT  |
| PB01     | GPIO                  | UART1_TXD | UART2_RXD | RTC_TAMP   | IR_OUT     | BTIM3_TOGN | GTIM1_CH4  | ATIM_CH3N | I2C2_SDA   | HALLTIM_TRGO |
| PB02     | GPIO                  | UART3_TXD | UART2_CTS | ATIM_BKOUT | HSE_OUT    | BTIM3_TOGP | GTIM1_ETR  | ATIM_CH1  | GTIM3_TRGO | HALLTIM_CH1  |
| PB03     | GPIO                  | UART1_TXD | UART2_TXD | SPI1_SCK   | UART3_CTS  | GTIM2_CH2  | GTIM1_ETR  | ATIM_CH2N | GTIM3_CH2  | SPI2_MISO    |
| PB04     | GPIO                  | UART3_TXD | UART2_CTS | SPI1_MISO  | UART3_RTS  | ATIM_ETR   | GTIM1_CH1  | ATIM_CH3N | GTIM3_CH3  | SPI2_MOSI    |
| PB05     | GPIO                  | UART3_RXD | UART2_RTS | SPI1_MOSI  | ADC1_SAM   | ATIM_BK    | GTIM1_CH2  | ATIM_CH1  | GTIM3_CH4  | HALLTIM_CH1  |

Table 9-2 GPIO multiplexing function assignment table (continued)

| Pin name      | Multiplexing function |           |             |           |            |            |            |           |            |             |
|---------------|-----------------------|-----------|-------------|-----------|------------|------------|------------|-----------|------------|-------------|
|               | AF0                   | AF1       | AF2         | AF3       | AF4        | AF5        | AF6        | AF7       | AF8        | AF9         |
| PB06          | GPIO                  | UART1_TXD | UART2_RXD   | I2C1_SCL  | BTIM3_TOGN | GTIM2_CH3  | GTIM1_CH3  | ATIM_CH2  | SPI3_MOSI  | HALLTIM_CH2 |
| PB07          | GPIO                  | UART1_RXD | UART2_TXD   | I2C1_SDA  | BTIM3_TOGP | GTIM2_CH4  | GTIM1_CH4  | ATIM_CH3  | SPI3_MISO  | HALLTIM_CH3 |
| PB08          | GPIO                  | UART2_RXD | UART3_TXD   | I2C1_SCL  | ADC2_SAM   | ATIM_CH4N  | GTIM1_CH3  | ATIM_ETR  | SPI3_SCK   |             |
| PB09          | GPIO                  | UART2_TXD | UART3_RXD   | I2C1_SDA  | IR_OUT     | ATIM_CH4   | GTIM1_CH4  | ATIM_BK   | SPI3_CS    |             |
| PB10          | GPIO                  | UART3_RXD | UART2_RTS   | I2C1_SCL  | ATIM_BK2   | VC4_OUT    | GTIM2_CH3  | ATIM_CH2  | I2C2_SCL   | HALLTIM_CH2 |
| PB11          | GPIO                  | UART1_RXD | ATIM_BK2OUT | I2C1_SDA  | LSI_OUT    | BTIM1_ETR  | GTIM2_CH4  | ATIM_CH3  | I2C2_SDA   | HALLTIM_CH3 |
| PB12          | GPIO                  | UART1_TXD | LPTIM_OUT   | SPI1_CS   | LSE_OUT    | VC1_OUT    | ATIM_TRGO  | ATIM_BK   | SPI2_CS    | HALLTIM_OUT |
| PB13          | GPIO                  | UART2_RXD | LPTIM_ETR   | SPI1_SCK  | ATIM_BK2   | VC2_OUT    | ATIM_TRGO2 | ATIM_CH1N | SPI2_SCK   | HALLTIM_CH1 |
| PB14          | GPIO                  | UART2_TXD | LPTIM_CH2   | SPI1_MISO | RTC_OUT    | VC3_OUT    | GTIM2_CH1  | ATIM_CH2N | SPI2_MISO  | HALLTIM_CH2 |
| PB15          | GPIO                  | UART3_RXD | LPTIM_CH1   | SPI1_MOSI | RTC_OUT    | BTIM2_TOGN | GTIM2_CH2  | ATIM_CH3N | SPI2_MOSI  | HALLTIM_CH3 |
| PC13          | GPIO                  | MCO_OUT   |             | RTC_TAMP  | RTC_OUT    | BTIM2_ETR  | ATIM_BK    | ATIM_ETR  | GTIM3_ETR  | GTIM4_ETR   |
| PC14          | GPIO                  | UART1_RXD | UART2_RXD   | UART3_CTS | I2C1_SDA   | BTIM2_TOGN | ATIM_CH6   | ATIM_CH5N | SPI3_MISO  | GTIM4_CH1   |
| PC15          | GPIO                  | UART1_TXD | UART2_TXD   | UART3_RTS | I2C1_SCL   | BTIM2_TOGP | ATIM_CH6N  | ATIM_CH5  | SPI3_MOSI  | GTIM4_CH2   |
| PF00          | GPIO                  | UART1_RXD |             |           | I2C1_SDA   | BTIM1_TOGN | GTIM1_TRGO | ATIM_CH6N | SPI3_SCK   | GTIM4_CH3   |
| PF01          | GPIO                  | UART1_TXD | HEXEN       | LSE_OUT   | I2C1_SCL   | BTIM1_TOGP | GTIM2_TRGO | ATIM_CH6  | SPI3_CS    | GTIM4_CH4   |
| PF03/<br>BOOT | GPIO                  | LSE_OUT   | LSI_OUT     | ADC1_SAM  | LVD_OUT    | HCLK_OUT   | ATIM_TRGO  | IR_OUT    | GTIM3_TRGO | GTIM4_TRGO  |
| PF06          | GPIO                  | UART1_CTS | UART3_RXD   | UART2_CTS | I2C1_SCL   | BTIM3_TOGN | GTIM2_TRGO | ATIM_CH5  | I2C2_SCL   | GTIM4_TRGO  |
| PF07          | GPIO                  | UART1_RTS | UART3_TXD   | UART2_RTS | I2C1_SDA   | BTIM3_TOGP | GTIM1_TRGO | ATIM_CH5N | I2C2_SDA   | SPI2_CS     |



### 9.3.6 Interrupt function

When each GPIO is set to digital input mode, it can be used as an external interrupt signal source, and the interrupt signal source can be set to two types: rising edge, and falling edge. Interrupt triggering methods can be used in combination, but share the same interrupt flag bit.

After the interrupt is triggered, the corresponding bit of the interrupt flag register GPIOx\_ISR will be set by hardware, and the program can determine which port generated the interrupt by querying GPIOx\_ISR. Through the interrupt flag clear register GPIOx\_ICR[y], the corresponding interrupt flag bit can be cleared.

The internal interrupt digital filter can digitally filter the input signal on the pin, providing 7 filter clock options, as shown in the following table:

Table 9-3 Digital filter clock selection

| GPIOx_FILTER.FLTCLK | Digital filter clock frequency |
|---------------------|--------------------------------|
| 000                 | HCLK/2                         |
| 001                 | HCLK/4                         |
| 010                 | HCLK/8                         |
| 011                 | BTIM1 overflow                 |
| 101                 | LSI (About 32.8kHz)            |
| 110                 | RC10K (About 10KHz)            |
| 111                 | LPTIM overflow                 |

Due to the wide range of selected filter clock cycles, users can easily implement flexible input interrupt debounce functions. Changes in input levels that are not maintained for more than one full filter clock cycle will not be communicated through the hardware filter to the internal interrupt trigger circuitry. Changes in input levels that persist for more than two complete filter clock cycles must pass through the hardware filter.

For the edge trigger type, considering the sensitivity to the time of the trigger edge, it is recommended to disable the hardware filter function in the interrupt digital filter configuration register GPIOx\_FILTER[y], because the hardware filter can improve signal stability while also inserting certain delay.

When CW32L012 works in Sleep mode or DeepSleep mode, the external interrupt function of GPIO can still be used. When an external interrupt is generated, the chip can be woken up from Sleep mode or DeepSleep mode to Active mode.

**Attention:** The same group of GPIOx.PINy shares a hardware filter clock source selection register, so the same group of GPIOs can only filter input signal jitter with the same filter clock.



### 9.3.7 Other functions

#### Atomic bit operation

The GPIO controller supports bit-setting, bit-clearing and bit-flipping functions.

Writing 1 to the GPIO bit clear register GPIOx\_BSRR[y] or the bit clear register GPIOx\_BRR[y] will directly change the state of the corresponding bit of the output data register GPIOx\_ODR, thereby indirectly affecting the final output level without affecting the Status of other bits in this register.

Writing 1 to the GPIO bit toggle register GPIOx\_TOG[y] will toggle the level state of the output port.

#### Port reset state

After power-on or reset, SWCLK (PA14) and SWDIO (PA13) default to digital pull-up, PF03/BOOT default to digital pull-down. Other ports are analog high-resistance input by default, and pull-up is not turned on by default.



## 9.4 Programming example

When configuring the GPIO port, you must first set `SYSCTRL_AHBEN.GPIOx` to 1, enable the corresponding GPIO configuration clock and working clock.

### 9.4.1 Digital output programming example

- Step 1: Set `GPIOx_ANALOG.PINy` to 0 to configure the port as a digital function.
- Step 2: Set `GPIOx_DIR.PINy` to 0, configure the port as output.
- Step 3: Configure the `GPIOx_OPENDRAIN` register to set the port output mode.
- Step 4: Configure the `GPIOx_ODR` register to set the port output level.

### 9.4.2 Digital input programming example

- Step 1: Set `GPIOx_ANALOG.PINy` to 0, configure the port as a digital function.
- Step 2: Set `GPIOx_DIR.PINy` to 1, configure the port as input.
- Step 3: Configure the `GPIOx_PUR` register and select whether to enable the internal pull-up resistor.
- Step 4: Read the `GPIOx_IDR` register and read the port input level.

### 9.4.3 Analog functional programming example

- Step 1: Set `GPIOx_ANALOG.PINy` to 1 to configure the port for analog function.

### 9.4.4 Multiplexed function programming example

- Step 1: Configure the port as digital output or digital input according to application requirements. For specific register configuration steps, please refer to section [9.4.1 Digital output programming example](#), section [9.4.2 Digital input programming example](#).
- Step 2: Configure the `GPIOx_AFRH` or `GPIOx_AFRL` register to set the port multiplexing function, see [Table 9-2: GPIO multiplexing function assignment table](#).

### 9.4.5 Interrupt function programming example

- Step 1: Configure the port as a digital input. For the specific register configuration steps, see section [9.4.2 Digital input programming example](#).
- Step 2: Configure `GPIOx_FILTER.FLTCLK`, and select the port interrupt filter clock.
- Step 3: Set `GPIOx_FILTER.PINy` to 1 to enable the corresponding port filter clock.
- Step 4: Configure the NVIC controller, see chapter [5 Interrupt](#).
- Step 5: Configure the `GPIOx_RISEIE`, `GPIOx_FALLIE` registers according to the application requirements, and select the GPIO interrupt trigger mode.
- Step 6: The port interrupt input signal triggers the GPIO interrupt, execute the interrupt service function.



## 9.5 List of registers

GPIOA base address: GPIOA\_BASE = 0x4800 0000

GPIOB base address: GPIOB\_BASE = 0x4800 0100

GPIOC base address: GPIOC\_BASE = 0x4800 0200

GPIOF base address: GPIOF\_BASE = 0x4800 0300

Table 9-4 List of GPIO registers

| Register name   | Register address  | Register description                                  |
|-----------------|-------------------|-------------------------------------------------------|
| GPIOx_DIR       | GPIOx_BASE + 0x00 | GPIOx I/O direction register                          |
| GPIOx_OPENDRAIN | GPIOx_BASE + 0x04 | GPIOx output mode register                            |
| GPIOx_PDR       | GPIOx_BASE + 0x0C | GPIOx pull-down resistor register                     |
| GPIOx_PUR       | GPIOx_BASE + 0x10 | GPIOx pull-up resistor register                       |
| GPIOx_AFRH      | GPIOx_BASE + 0x14 | GPIOx multiplexing function register high segment     |
| GPIOx_AFRL      | GPIOx_BASE + 0x18 | GPIOx multiplexing function register low segment      |
| GPIOx_ANALOG    | GPIOx_BASE + 0x1C | GPIOx analog digital configuration register           |
| GPIOx_RISEIE    | GPIOx_BASE + 0x24 | GPIOx rising edge interrupt enable register           |
| GPIOx_FALLIE    | GPIOx_BASE + 0x28 | GPIOx falling edge interrupt enable register          |
| GPIOx_ISR       | GPIOx_BASE + 0x34 | GPIOx interrupt flag register                         |
| GPIOx_ICR       | GPIOx_BASE + 0x38 | GPIOx interrupt flag clear register                   |
| GPIOx_FILTER    | GPIOx_BASE + 0x40 | GPIOx interrupt digital filter configuration register |
| GPIOx_IDR       | GPIOx_BASE + 0x50 | GPIOx input data register                             |
| GPIOx_ODR       | GPIOx_BASE + 0x54 | GPIOx output data register                            |
| GPIOx_BRR       | GPIOx_BASE + 0x58 | GPIOx bit clear register                              |
| GPIOx_BSRR      | GPIOx_BASE + 0x5C | GPIOx bit-setting clear register                      |
| GPIOx_TOG       | GPIOx_BASE + 0x60 | GPIOx bit flip register                               |





## 9.6 Register description

See section 1 [Documentation conventions](#) for abbreviations used in register descriptions.

### 9.6.1 GPIOx\_DIR GPIO I/O direction register (x = A/B/C/F)

Address offset: 0x00      Reset value: 0x0000 FFFF(GPIOA)

0x0000 FFFF(GPIOB)

0x0000 E000(GPIOC)

0x0000 00CB(GPIOF)

| Bit field | Name               | Permission | Function description                                                                                           |
|-----------|--------------------|------------|----------------------------------------------------------------------------------------------------------------|
| 31:16     | RFU                | -          | Reserved bits, please keep the default value.                                                                  |
| 15:0      | PINy<br>y = 0 ~ 15 | RW         | Port input and output direction control<br>0: Configure the port as output.<br>1: Configure the port as input. |

### 9.6.2 GPIOx\_OPENDRAIN GPIO output mode register (x = A/B/C/F)

Address offset: 0x04      Reset value: 0x0000 0000

| Bit field | Name               | Permission | Function description                                                          |
|-----------|--------------------|------------|-------------------------------------------------------------------------------|
| 31:16     | RFU                | -          | Reserved bits, please keep the default value.                                 |
| 15:0      | PINy<br>y = 0 ~ 15 | RW         | Port output mode control bit<br>0: Push-pull output.<br>1: Open-drain output. |

### 9.6.3 GPIOx\_PDR GPIO pull-down resistor register (x = A/B/C/F)

Address offset: 0x0C      Reset value: 0x0000 0008

| Bit field | Name               | Permission | Function description                                                                                                                                                                                                                                   |
|-----------|--------------------|------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:16     | RFU                | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                          |
| 15:0      | PINy<br>y = 0 ~ 15 | RW         | Port pull-down resistor enable control<br>0: Disable pull-down resistor.<br>1: Enable pull-down resistor.<br><i>Attention: When GPIOx_PDR.PINy and GPIOx_PUR.PINy are set to 1 at the same time, the port state is to enable the pull-up resistor.</i> |

*Attention: Only the PF03 pin has a pull-down resistor.*

### 9.6.4 GPIOx\_PUR GPIO pull-up resistor register (x = A/B/C/F)

Address offset: 0x10      Reset value: 0x0000 0000

| Bit field | Name               | Permission | Function description                                                                                                                                                                                                                               |
|-----------|--------------------|------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:16     | RFU                | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                      |
| 15:0      | PINy<br>y = 0 ~ 15 | RW         | Port pull-up resistor enable control<br>0: Disable pull-up resistors.<br>1: Enable pull-up resistors.<br><i>Attention: When GPIOx_PDR.PINy and GPIOx_PUR.PINy are set to 1 at the same time, the port state is to enable the pull-up resistor.</i> |

### 9.6.5 GPIOx\_AFRH GPIO multiplexing function register high segment (x = A/B/C/F)

Address offset: 0x14      Reset value: 0x0000 0000

| Bit field                                                       | Name               | Permission | Function description                                                                                                                                |
|-----------------------------------------------------------------|--------------------|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:28<br>27:24<br>23:20<br>19:16<br>15:12<br>11:8<br>7:4<br>3:0 | AFRy<br>y = 8 ~ 15 | RW         | Port digital multiplexing function control<br>0000: GPIO<br>0001: AF1<br>0010: AF2<br>0011: AF3<br>0100: AF4<br>0101: AF5<br>0110: AF6<br>0111: AF7 |

### 9.6.6 GPIOx\_AFRL GPIO multiplexing function register low segment (x = A/B/C/F)

Address offset: 0x18      Reset value: 0x0000 0000

| Bit field                                                       | Name              | Permission | Function description                                                                                                                                |
|-----------------------------------------------------------------|-------------------|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:28<br>27:24<br>23:20<br>19:16<br>15:12<br>11:8<br>7:4<br>3:0 | AFRy<br>y = 0 ~ 7 | RW         | Port digital multiplexing function control<br>0000: GPIO<br>0001: AF1<br>0010: AF2<br>0011: AF3<br>0100: AF4<br>0101: AF5<br>0110: AF6<br>0111: AF7 |



**9.6.7 GPIOx\_ANALOG GPIO analog digital configuration register (x = A/B/C/F)**

Address offset: 0x1C      Reset value: 0x0000 01FF(GPIOA)

0x0000 007F(GPIOB)

0x0000 E000(GPIOC)

0x0000 00C3(GPIOF)

| Bit field | Name               | Permission | Function description                                                                                                                       |
|-----------|--------------------|------------|--------------------------------------------------------------------------------------------------------------------------------------------|
| 31:16     | RFU                | -          | Reserved bits, please keep the default value.                                                                                              |
| 15:0      | PINy<br>y = 0 ~ 15 | RW         | Port analog/digital function configuration<br>0: Configure the port as a digital function.<br>1: Configure the port as an analog function. |

**9.6.8 GPIOx\_RISEIE GPIO rising edge interrupt enable register (x = A/B/C/F)**

Address offset: 0x24      Reset value: 0x0000 0000

| Bit field | Name               | Permission | Function description                                                                                                                                                           |
|-----------|--------------------|------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:16     | RFU                | -          | Reserved bits, please keep the default value.                                                                                                                                  |
| 15:0      | PINy<br>y = 0 ~ 15 | RW         | Port rising edge interrupt enable control<br>0: Disable the rising edge interrupt of the corresponding port.<br>1: Enable the rising edge interrupt of the corresponding port. |

**9.6.9 GPIOx\_FALLIE GPIO falling edge interrupt enable register (x = A/B/C/F)**

Address offset: 0x28      Reset value: 0x0000 0000

| Bit field | Name               | Permission | Function description                                                                                                                                                              |
|-----------|--------------------|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:16     | RFU                | -          | Reserved bits, please keep the default value.                                                                                                                                     |
| 15:0      | PINy<br>y = 0 ~ 15 | RW         | Port falling edge interrupt enable control<br>0: Disable the falling edge interrupt of the corresponding port.<br>1: Enable the falling edge interrupt of the corresponding port. |

Address offset: 0x34      Reset value: 0x0000 0000

| Bit field | Name               | Permission | Function description                                                                                        |
|-----------|--------------------|------------|-------------------------------------------------------------------------------------------------------------|
| 31:16     | RFU                | -          | Reserved bits, please keep the default value.                                                               |
| 15:0      | PINy<br>y = 0 ~ 15 | RO         | Port interrupt status flag<br>0: Enabled interrupt not detected.<br>1: Enabled interrupt has been detected. |

Address offset: 0x38      Reset value: 0x0000 01FF(GPIOA)

0x0000 00CB(GPIOF)

| Bit field | Name               | Permission | Function description                                                                                    |
|-----------|--------------------|------------|---------------------------------------------------------------------------------------------------------|
| 31:16     | RFU                | -          | Reserved bits, please keep the default value.                                                           |
| 15:0      | PINy<br>y = 0 ~ 15 | R1W0       | Port interrupt status flag cleared<br>W0: Clear the corresponding interrupt flag bit<br>W1: No function |

Address offset: 0x40      Reset value: 0x0000 0000

| Bit field | Name               | Permission | Function description                                                                                                                                                                                                                     |
|-----------|--------------------|------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:19     | RFU                | -          | Reserved bits, please keep the default value.                                                                                                                                                                                            |
| 18:16     | FLTCLK             | RW         | Port interrupt filter clock selection<br>000: HCLK/2                      101: LSI (About 32.8kHz)<br>001: HCLK/4                    110: RC10K (About 10KHz)<br>010: HCLK/8                  111: LPTIM overflow<br>011: BTIM1 overflow |
| 15:0      | PINy<br>y = 0 ~ 15 | RW         | Port filter clock enable control<br>0: Disable the corresponding port to filter the clock<br>1: Enable the corresponding port to filter the clock<br><i>Note: Filter out pulses with a width less than the FLTCLK clock width.</i>       |

*Attention: If you need the hardware filter function to work properly in DeepSleep mode, set the filter clock to the overflow signal of LSI, RC10K or LPTIM.*

Address offset: 0x50      Reset value: 0x0000 0180(GPIOA)

0x0000 0080(GPIOB)

0x0000 0000(GPIOC)

0x0000 0008(GPIOF)

| Bit field | Name               | Permission | Function description                                                                    |
|-----------|--------------------|------------|-----------------------------------------------------------------------------------------|
| 31:16     | RFU                | -          | Reserved bits, please keep the default value.                                           |
| 15:0      | PINy<br>y = 0 ~ 15 | RO         | Read port input level status<br>0: The port is low level.<br>1: The port is high level. |

#### 9.6.14 GPIOx\_ODR GPIO output data register (x = A/B/C/F)

Address offset: 0x54      Reset value: 0x---- ----

| Bit field | Name               | Permission | Function description                                                                         |
|-----------|--------------------|------------|----------------------------------------------------------------------------------------------|
| 31:16     | RFU                | -          | Reserved bits, please keep the default value.                                                |
| 15:0      | PINy<br>y = 0 ~ 15 | RW         | Set the port output level<br>0: Set port output low level.<br>1: Set port output high level. |

*Note: Support 8bit read and write.*

#### 9.6.15 GPIOx\_BRR GPIO port bit set clear register (x = A/B/C/F)

Address offset: 0x58      Reset value: 0x0000 0000

| Bit field | Name               | Permission | Function description                                                                                                                                           |
|-----------|--------------------|------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:16     | RFU                | -          | Reserved bits, please keep the default value.                                                                                                                  |
| 15:0      | BRRy<br>y = 0 ~ 15 | R0W1       | Port bit clear control<br>0: Does not affect the corresponding bits of the GPIOx_ODR register.<br>1: Set the corresponding bit of the GPIOx_ODR register to 0. |



### 9.6.16 GPIOx\_BSRR GPIO port bit-setting clear register (x = A/B/C/F)

Address offset: 0x5C      Reset value: 0x0000 0000

| Bit field | Name               | Permission | Function description                                                                                                                                                                                                                          |
|-----------|--------------------|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:16     | BRRy<br>y = 0 ~ 15 | R0W1       | Port bit-setting clear control<br>0: Does not affect the corresponding bits of the GPIOx_ODR register.<br>1: Set the corresponding bit of the GPIOx_ODR register to 0.                                                                        |
| 15:0      | BSSy<br>y = 0 ~ 15 | R0W1       | Port bit-setting control<br>0: Does not affect the corresponding bits of the GPIOx_ODR register.<br>1: Set the corresponding bit of the GPIOx_ODR register to 1.<br><i>Note: When BRRy and BSSy are both set 1, BSSy has higher priority.</i> |

### 9.6.17 GPIOx\_TOG GPIO port bit flip register (x = A/B/C/F)

Address offset: 0x60      Reset value: 0x0000 0000

| Bit field | Name               | Permission | Function description                                                                                                                                        |
|-----------|--------------------|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:16     | RFU                | -          | Reserved bits, please keep the default value.                                                                                                               |
| 15:0      | PINy<br>y = 0 ~ 15 | R0W1       | Port bit flip control<br>0: Does not affect the corresponding bits of the GPIOx_ODR register.<br>1: Invert the corresponding bit of the GPIOx_ODR register. |



## 10 Cyclic redundancy check (CRC)

### 10.1 Overview

Cyclic redundancy check (CRC) is mainly used to verify the correctness and integrity of data transmission or data storage. The CW32L012 integrates a CRC calculation unit, which supports the use of multiple CRC algorithms to perform CRC calculation on the input data.



## 10.2 Main features

- 8bit input data bit width
- 2 kinds of polynomial
  1. CRC-16 polynomial 1:  $x^{16}+x^{15}+x^2+1$
  2. CRC-16 polynomial 2:  $x^{16}+x^{12}+x^5+1$
- 8 kinds of commonly used algorithms  
Combination based on polynomial, initial value, result XOR value, input/output inversion.





### 10.3 Functional description

The CRC unit obtains the CRC calculation result by performing a 'division' operation on the input data (or the inversion of the input data) and the selected polynomial value, and then inverting or not inverting the obtained remainder, as well as XOR processing.

Before the CRC unit is used, it is necessary to set SYSCTRL\_AHBEN.CRC to 1, and turn on the configuration clock and working clock of the CRC unit, which are generally set during system initialization.



### 10.3.1 Algorithmic modes

The CRC unit of CW32L012 supports multiple algorithm modes. Different CRC algorithms have different parameters such as polynomial, initial value, input data inversion, output data inversion, and result XOR value, as shown in the following table:

Table 10-1 CRC algorithm modes

| Algorithm name    | Polynomial value | Initial value | Input inversion | Output inversion | Result XOR value |
|-------------------|------------------|---------------|-----------------|------------------|------------------|
| CRC16_IBM         | 0x8005           | 0x0000        | True            | True             | 0x0000           |
| CRC16_MAXIM       | 0x8005           | 0x0000        | True            | True             | 0xFFFF           |
| CRC16_USB         | 0x8005           | 0xFFFF        | True            | True             | 0xFFFF           |
| CRC16_MODBUS      | 0x8005           | 0xFFFF        | True            | True             | 0x0000           |
| CRC16_CCITT       | 0x1021           | 0x0000        | True            | True             | 0x0000           |
| CRC16_CCITT_False | 0x1021           | 0xFFFF        | False           | False            | 0x0000           |
| CRC16_X25         | 0x1021           | 0xFFFF        | True            | True             | 0xFFFF           |
| CRC16_XMODEM      | 0x1021           | 0x0000        | False           | False            | 0x0000           |

The meaning of each parameter is as follows:

- Polynomial value**  
 The polynomial is the description of the code group, such as CRC-16 polynomial 2:  $x^{16}+x^{12}+x^5+1$ , the corresponding code group is 1 0001 0000 0010 0001. Because the highest bit of the polynomial code group is fixed to 1, and the position of the highest bit is known, the code group after removing the highest bit 1 is generally called a polynomial value. For example, the value of CRC-16 polynomial 2 is 0001 0000 0010 0001, that is 0x1021.
- Initial value:**  
 The initial value of the CRC register before calculating the CRC check value.
- Input data inversion:**  
 That is, before the calculation starts, the high and low order bits of the data that need to calculate the CRC check value are reversed, such as the data bit 1011, which is 1101 after the inversion.
- Output data inversion:**  
 After the CRC calculation is completed, before the XOR value is performed with the result, the high and low order bits of the calculated value are reversed. If the calculation result is 1011, it will be 1101 after inversion.
- Result XOR value:**  
 After the CRC calculation is completed, the obtained CRC calculation value is XORed with the result XOR value to obtain the final CRC check value.

### 10.3.2 Input data bit width

The CRC calculation unit supports 8bit input data bit width. When inputting data, input the low byte first and then the high byte.

**For example**, the user needs to calculate the CRC check value of the group of data 0x00, 0x11, 0x22, 0x33, 0x44, 0x55, 0x66, 0x77, and the writing sequence is 0x00, 0x11, 0x22, 0x33, 0x44, 0x55, 0x66, 0x77.

Code example:

```
CW_CRC -> DR = 0x00 ;
CW_CRC -> DR = 0x11 ;
CW_CRC -> DR = 0x22 ;
CW_CRC -> DR = 0x33 ;
CW_CRC -> DR = 0x44 ;
CW_CRC -> DR = 0x55 ;
CW_CRC -> DR = 0x66 ;
CW_CRC -> DR = 0x77 ;
```



## 10.4 Programming examples

### 10.4.1 CRC16\_CCITT algorithm mode

- Step 1:** Set CRC\_CR.MODE to 0x04, select the CRC16\_CCITT algorithm mode, and the initial value of the CRC register is automatically configured by hardware to 0x0000.
- Step 2:** Write the raw data to be encoded into the data register CRC\_DR in turn, and the input data bit width can be selected from 8bit, 16bit, and 32bit. See the code example in section [10.3.2 Input data bit width](#).
- Step 3:** Read CRC\_RESULT to get the CRC check value.

Code example:

```
tempdata = CW_CRC -> RESULT ;
```



## 10.5 List of registers

CRC base address: CRC\_BASE = 0x4002 3000

Table 10-2 List of CRC registers

| Register name | Register address | Register description |
|---------------|------------------|----------------------|
| CRC_CR        | CRC_BASE + 0x00  | Control register     |
| CRC_DR        | CRC_BASE + 0x08  | Data register        |
| CRC_RESULT    | CRC_BASE + 0x0C  | Result register      |

## 10.6 Register description

See section [1 Documentation conventions](#) for abbreviations used in register descriptions.

### 10.6.1 CRC\_CR Control register

Address offset: 0x00      Reset value: 0x0000 0004

| Bit field | Name | Permission | Function description                                                                                                                                                                                       |
|-----------|------|------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:4      | RFU  | -          | Reserved bits, please keep the default value.                                                                                                                                                              |
| 3:0       | MODE | RW         | CRC algorithm mode configuration<br>0000: CRC16_IBM<br>0001: CRC16_MAXIM<br>0010: CRC16_USB<br>0011: CRC16_MODBUS<br>0100: CRC16_CCITT<br>0101: CRC16_CCITT_False<br>0110: CRC16_X25<br>0111: CRC16_XMODEM |

### 10.6.2 CRC\_DR Data register

Address offset: 0x08      Reset value: 0x0000 0000

| Bit field | Name | Permission | Function description                          |
|-----------|------|------------|-----------------------------------------------|
| 31:8      | RFU  | -          | Reserved bits, please keep the default value. |
| 7:0       | DR   | RW         | Data write register, bit width 8bit.          |

### 10.6.3 CRC\_RESULT Result register

Address offset: 0x0C      Reset value: 0x0000 0000

| Bit field | Name   | Permission | Function description                          |
|-----------|--------|------------|-----------------------------------------------|
| 31:16     | RFU    | -          | Reserved bits, please keep the default value. |
| 15:0      | RESULT | RO         | CRC calculation result.                       |



## 11 Extended arithmetic unit (EAU)

### 11.1 Overview

The extended arithmetic unit (EAU) is a coprocessor specifically designed for arithmetic operations, supporting efficient execution of signed integer division, unsigned integer division, and unsigned integer square root operations to obtain quotients/square roots and remainders. It is an effective supplement and extension to the arithmetic instructions of the M0+ core.



## 11.2 Main features

- Supports 32-bit division by 32-bit signed/unsigned integer division operations to obtain the quotient and remainder.
- Supports 32-bit unsigned integer square root operations to obtain square root and remainder.
- Unsigned division supports the zero divisor flag.
- Signed division supports the zero divisor flag and the overflow flag.





### 11.3 Operation definition

EAU supports three types of operations: division of 32-bit unsigned integers by 32-bit unsigned integers, division of 32-bit signed integers by 32-bit signed integers, and square root of 32-bit unsigned integers. Among them, division operations require 2 to 35 HCLK cycles, and root extraction operations require 17 HCLK cycles. Each operation is defined as follows:

Table 11-1 Definition of operation

| Operation definition                                                                  | Symbol description                                                                                                                                                                                  |
|---------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Unsigned division:<br>$X = A / B$<br>$Y = A \% B$<br>And it satisfies $A = B * X + Y$ | A: Dividend, a 32-bit unsigned integer.<br>B: Divisor, a 32-bit unsigned integer.<br>X: Quotient, a 32-bit unsigned integer.<br>Y: Remainder, a 32-bit unsigned integer.                            |
| Signed division:<br>$X = A / B$<br>$Y = A \% B$<br>And it satisfies $A = B * X + Y$   | A: Dividend, a 32-bit signed integer.<br>B: Divisor, a 32-bit signed integer.<br>X: Quotient, a 32-bit signed integer.<br>Y: Remainder, a 32-bit signed integer with the same sign as the dividend. |
| Square root:<br>$X = A^{1/2}$<br>$Y = A - X^2$                                        | A: Radicand, 32-bit unsigned integer.<br>X: Square root, 32-bit unsigned integer.<br>Y: Remainder, 32-bit unsigned integer.                                                                         |



## 11.4 Status flag

The EAU has 3 status flags to indicate the current operating status of the EAU, including the busy flag bit BUSY, the divisor zero flag bit ZERO and the overflow flag bit OVR.

### Busy flag (EAU\_CSR.BUSY)

The EAU\_CSR.BUSY flag indicates whether the current EAU is in an end-of-operation or idle state.

When EAU\_CSR.BUSY is 0, it indicates that EAU is in the end of operation or idle state, and can read the result or start a new operation; when EAU\_CSR.BUSY is 1, it indicates that EAU is busy, the current data operation is in progress, and the operation result cannot be provided or a new operation cannot be started.

### Divisor zero flag (EAU\_CSR.ZERO)

The EAU\_CSR.ZERO flag indicates whether the division operation that the EAU has just completed is a meaningless operation with a zero divisor. The ZERO flag applies only to division operations (including both unsigned and signed division) and does not apply to square root operations.

When EAU\_CSR.ZERO is 0, it means that the divisor of the division operation just completed is not zero, which is a meaningful operation; when EAU\_CSR.ZERO is 1, it means that the divisor of the division operation just completed is zero, which is a meaningless operation.

Setting the ZERO flag does not affect the continuation of a new round of operations, regardless of the type of operation. When a new round of operation is started, the flag bit is automatically cleared and re-indicated according to the new round of operation.

### Signed division overflow flag (EAU\_CSR.OVR)

The EAU\_CSR.OVR flag indicates whether the signed division operation the EAU just completed overflowed. The OVR flag is only applicable to signed division operations, not root and unsigned division operations.

Due to the limitation of the coding range, for example, when a signed integer division is performed, if the dividend is the smallest negative number (0x8000 0000 in the case of 32-bit) and the divisor is -1 (0xFFFF FFFF in the case of 32-bit), then the quotient obtained is a very large positive number, and its value is 1 larger than the largest integer that can be represented by a signed integer of 32-bit, exceeding the legal coding range. This situation is called overflow of signed division. When EAU\_CSR.OVR is 0, it indicates that the signed division just completed did not overflow; when EAU\_CSR.OVR is 1, it indicates that the signed division just completed did overflow.

Setting the OVR flag does not affect the continuation of a new round of operations, regardless of the type of operation. When a new round of operation is started, the flag bit is automatically cleared and re-indicated according to the new round of operation.



## 11.5 Programming examples

### 11.5.1 Unsigned division programming example

- Step 1:** Set SYSCTRL\_AHBEN.EAU to 1 to enable the configuration clock and working clock of the EAU module.
- Step 2:** Set EAU\_CSR.MODE to 0, select EAU operation mode to unsigned division.
- Step 3:** Write dividend to EAU\_DIVIDEND register.
- Step 4:** Write the divisor to the EAU\_DIVISOR register and the operation starts automatically;
- Step 5:** Query and wait for the busy flag bit EAU\_CSR.BUSY becomes 0 and the operation ends.
- Step 6:** Query divide-by-zero flag EAU\_CSR.ZERO. EAU\_CSR.ZERO = 0 means there is no abnormality in this operation, EAU\_CSR.ZERO = 1 means the divisor of this operation is zero, and the result is meaningless.
- Step 7:** If there is no abnormality in this operation, the quotient can be read from the EAU\_QUOTIENT register, and the remainder can be read from the EAU\_REMAINDER register; if the divisor is 0, it means that the operation result is meaningless and the current result is abandoned.
- Step 8:** If a new round of arithmetic is required, repeat steps 3 to 7.

### 11.5.2 Signed division programming example

- Step 1:** Set SYSCTRL\_AHBEN.EAU to 1 to enable the configuration clock and working clock of the EAU module.
- Step 2:** Set EAU\_CSR.MODE to 1, select signed division as EAU operation mode.
- Step 3:** Write dividend to EAU\_DIVIDEND register.
- Step 4:** Write the divisor to the EAU\_DIVISOR register and the operation starts automatically.
- Step 5:** Query and wait for the busy flag bit EAU\_CSR.BUSY becomes 0 and the operation ends.
- Step 6:** Query divide-by-zero flag EAU\_CSR.ZERO and overflow flag EAU\_CSR.OVR. EAU\_CSR.ZERO is 0 and EAU\_CSR.OVR is 0, which means that there is no abnormality in this operation; EAU\_CSR.ZERO is 1 or EAU\_CSR.OVR is 1, which means that the corresponding abnormality occurs in this operation, and the result is meaningless.
- Step 7:** If there is no abnormality in this operation, the quotient can be read from the EAU\_QUOTIENT register, and the remainder can be read from the EAU\_REMAINDER register; if there is abnormality in this operation, it means that the operation result is meaningless and the current result is abandoned.
- Step 8:** If a new round of arithmetic is required, repeat steps 3 to 7.



### 11.5.3 Square root arithmetic programming example

- Step 1:** Set SYSCTRL\_AHBEN.EAU to 1 to enable the configuration clock and working clock of the EAU module.
- Step 2:** Set EAU\_CSR.MODE to 2, select EAU operation mode as root operation.
- Step 3:** Write the number to be squared into the EAU\_DIVIDEND register, and the operation starts automatically.
- Step 4:** Query and wait for the busy flag bit EAU\_CSR.BUSY becomes 0 and the operation ends.
- Step 5:** Read operation result: read square root from EAU\_QUOTIENT register and remainder from EAU\_REMAINDER register.
- Step 6:** If a new round of arithmetic is required, repeat steps 3 to 5.



## 11.6 List of registers

EAU base address: EAU\_BASE = 0x4002 3200

Table 11-2 List of EAU registers

| Register name | Register address | Register description          |
|---------------|------------------|-------------------------------|
| EAU_CSR       | EAU_BASE + 0x00  | Control and status register   |
| EAU_DIVIDEND  | EAU_BASE + 0x04  | Dividend/radicand register    |
| EAU_DIVISOR   | EAU_BASE + 0x08  | Divisor register              |
| EAU_QUOTIENT  | EAU_BASE + 0x0C  | Quotient/square root register |
| EAU_REMAINDER | EAU_BASE + 0x10  | Remainder register            |



## 11.7 Register descriptions

See section 1 [Documentation conventions](#) for abbreviations used in register descriptions.

EAU only supports 32-bit operations. Any written operation data is treated as 32-bit data to participate in the operation, and the provided results are also stored in the corresponding registers in 32-bit form. EAU only supports 32-bit register access, 8-bit or 16-bit register access will be ignored.

### 11.7.1 EAU\_CSR Control and status register

Address offset: 0x00      Reset value: 0x0000 0000

| Bit field | Name | Permission | Function description                                                                                                                                                                                                                                                                                                                                                       |
|-----------|------|------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:5      | RFU  | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                                                              |
| 4         | BUSY | RO         | Busy flag bit<br>0: Operation completed or idle.<br>1: Operation in progress.                                                                                                                                                                                                                                                                                              |
| 3         | ZERO | RO         | Divisor zero flag<br>0: Divisor is not zero.<br>1: Divisor is zero.<br><i>Note: The setting of this flag bit does not affect the continuation of a new round of arithmetic. When a new round of arithmetic is initiated, this flag bit is automatically cleared to zero and the indication is resumed in accordance with the situation of the new round of arithmetic.</i> |
| 2         | OVR  | RO         | Signed division overflow flag<br>0: No overflow.<br>1: Overflow.<br><i>Note: The setting of this flag bit does not affect the continuation of a new round of arithmetic. When a new round of arithmetic is initiated, this flag bit is automatically cleared to zero and the indication is resumed in accordance with the situation of the new round of arithmetic.</i>    |
| 1:0       | MODE | RW         | Arithmetic mode selection<br>00: Unsigned division.<br>01: Signed Division.<br>10: Square root.<br>11: Reserved.<br><i>Attention: Writable only when BUSY=0, not while the operation is in progress (BUSY=1).</i>                                                                                                                                                          |



### 11.7.2 EAU\_DIVIDEND Dividend/radicand register

Address offset: 0x04      Reset value: 0x0000 0000

| Bit field | Name     | Permission | Function description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|-----------|----------|------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:0      | DIVIDEND | RW         | <p>Dividend/radicand register, which stores input data required for the operation:</p> <p>When performing a square root operation, write the radicand to this register (the write operation triggers the square root operation to start directly).</p> <p>When dividing, write the dividend to this register.</p> <p><i>Attention: Writable only when BUSY=0, not while operation is in progress (BUSY=1). Writes to this register while the operation is in progress are ignored and do not trigger any action.</i></p> |

### 11.7.3 EAU\_DIVISOR Divisor register

Address offset: 0x08      Reset value: 0x0000 0000

| Bit field | Name    | Permission | Function description                                                                                                                                                                                                                                                                                                                                                   |
|-----------|---------|------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:0      | DIVISOR | RW         | <p>Divisor register, storing input data required for the operation.</p> <p>During square root operations, this register has no function and can be ignored.</p> <p><i>Attention: Writable only when BUSY=0, not while operation is in progress (BUSY=1). Writes to this register while the operation is in progress are ignored and do not trigger any action.</i></p> |

### 11.7.4 EAU\_QUOTIENT Quotient/square root register

Address offset: 0x0C      Reset value: 0x0000 0000

| Bit field | Name     | Permission | Function description                                                                                                                                                                                                                                                                                                                                        |
|-----------|----------|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:0      | QUOTIENT | RO         | <p>The quotient/square root register, which stores the result of the operation. The software is readable only after the end of the operation, and reads zero during the operation.</p> <p>For square root operations, this register stores the square root of the result.</p> <p>For division operations, this register stores the calculated quotient.</p> |



### 11.7.5 EAU\_REMAINDER Remainder register

Address offset: 0x10      Reset value: 0x0000 0000

| Bit field | Name      | Permission | Function description                                                                                                                                                                                                                                                                                                                                                      |
|-----------|-----------|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:0      | REMAINDER | RO         | <p>The remainder register, which stores the remainder generated by the operation. The software is readable only after the end of the operation, and reads zero during the operation.</p> <p>For square root operations, this register stores the remainder of the square.</p> <p>For division operations, this register stores the remainder of the undivided number.</p> |





## 12 Coordinate rotation digital computer algorithm (CORDIC)

### 12.1 Overview

Coordinate rotation digital computer algorithm provides hardware acceleration for certain mathematical functions, especially trigonometric functions, which are typically used in motor control, metrology, signal processing, and many other applications. Compared with software implementation, it speeds up the calculation of these functions, allows for lower operating frequencies, or frees up processor cycles to perform other tasks.



## 12.2 Main features

- Supports cosine (cos), sine (sin), phase angle (atan2), modulus (hypot), arctangent (atan), hyperbolic cosine (cosh), hyperbolic sine (sinh), and hyperbolic arctangent (atanh) function operations.
- DMA can be triggered for quick access during idle states or after completion of operations.
- Number of iterations is configurable (range 6 to 66).



## 12.3 Operation definition

CORDIC supports cosine (cos), sine (sin), phase angle (atan2), modulus hypot, arctangent (atan), hyperbolic cosine (cosh), hyperbolic sine (sinh), and hyperbolic arctangent (atanh) function operations, with an execution time of (number of iterations + 2) HCLK cycles.

Each operation is defined as follows:

- Due to the convergence domain limitation of the CORDIC algorithm itself, some functions require the input to be within a certain range.
- Functions with domain requirements are specified in the input data; if not specified, they are considered to be the entire range of [-1,1).

Table 12-1 Definition of operation

| Pre-calculation configuration |       | Pre-calculation input |            |                          | Post-calculation output      |             |                        |
|-------------------------------|-------|-----------------------|------------|--------------------------|------------------------------|-------------|------------------------|
| func                          | scale | X                     | Y          | Z                        | X                            | Y           | Z                      |
| Cosine (cos)                  | 0     | No concern            | No concern | Radians z in units of pi | cosz                         | sinz        | Meaningless            |
| Sine (sin)                    | 0     | No concern            | No concern | Radians z in units of pi | cosz                         | sinz        | Meaningless            |
| Phase angle (atan2)           | 0     | Data x                | Data y     | No concern               | $\frac{\sqrt{x^2 + y^2}}{2}$ | Meaningless | atan(y/x)<br>Unit: pi  |
| Modulus (hypot)               | 0     | Data x                | Data y     | No concern               | $\frac{\sqrt{x^2 + y^2}}{2}$ | Meaningless | atan(y/x)<br>Unit: pi  |
| Arctangent (atan)             | 0     | No concern            | Data y     | No concern               | Meaningless                  | Meaningless | atan(y)<br>Unit: pi    |
|                               | 1     |                       |            |                          |                              |             | atan(2y)<br>Unit: pi   |
|                               | 2     |                       |            |                          |                              |             | atan(4y)<br>Unit: pi   |
|                               | 3     |                       |            |                          |                              |             | atan(8y)<br>Unit: pi   |
|                               | 4     |                       |            |                          |                              |             | atan(16y)<br>Unit: pi  |
|                               | 5     |                       |            |                          |                              |             | atan(32y)<br>Unit: pi  |
|                               | 6     |                       |            |                          |                              |             | atan(64y)<br>Unit: pi  |
|                               | 7     |                       |            |                          |                              |             | atan(128y)<br>Unit: pi |
| Hyperbolic cosine (cosh)      | 1     | No concern            | No concern | Data z<br>[-0.559,0.559] | cosh(2z)/2                   | sinh(2z)/2  | Meaningless            |
| Hyperbolic sine (sinh)        | 1     | No concern            | No concern | Data z<br>[-0.559,0.559] | cosh(2z)/2                   | sinh(2z)/2  | Meaningless            |



Table 12-1 Definition of operation (continued)

| Pre-calculation configuration |   | Pre-calculation input |                          |            | Post-calculation output |             |             |           |
|-------------------------------|---|-----------------------|--------------------------|------------|-------------------------|-------------|-------------|-----------|
| Hyperbolic arctangent (atanh) | 1 | No concern            | Data y<br>[-0.403,0.403] | No concern | Meaningless             | Meaningless | atanh(2y)/2 |           |
| Natural logarithm (ln)        | 1 | Data x                | No concern               | No concern | Meaningless             | Meaningless | ln(2x)/4    |           |
|                               |   | [0.0535,0.5)          |                          |            |                         |             |             |           |
|                               | 2 | Data x                |                          |            |                         |             |             | ln(4x)/2  |
|                               |   | [0.25,0.75)           |                          |            |                         |             |             |           |
|                               | 3 | Data x                |                          |            |                         |             |             | ln(8x)/2  |
|                               |   | [0.375,0.875)         |                          |            |                         |             |             |           |
|                               | 4 | Data x                |                          |            |                         |             |             | ln(16x)/4 |
|                               |   | [0.4375,0.584)        |                          |            |                         |             |             |           |
| Square root (sqrt)            | 0 | Data x                | No concern               | No concern | $\sqrt{x}$              | Meaningless | Meaningless |           |
|                               |   | [0.027,0.75)          |                          |            |                         |             |             |           |
|                               | 1 | Data x                |                          |            |                         |             |             |           |
|                               |   | [0.375,0.875)         |                          |            |                         |             |             |           |
|                               | 2 | Data x                |                          |            |                         |             |             |           |
|                               |   | [0.4375,0.585]        |                          |            |                         |             |             |           |

About q1.15 and q1.31:

- Using sint16/sint32 to represent real numbers in the range  $[-1, 1)$ .
- The sint16 corresponds to  $q1.15 = \text{sint16} / \text{pow}(2, 15)$ .
- The sint32 corresponds to  $q1.31 = \text{sint32} / \text{pow}(2, 31)$ .

About the number of iterations:

- For q1.15, it is recommended to iterate 16 to 20 times.
- For q1.31, it is recommended to iterate 24 to 32 times.

About startup:

- When only one data input (x or y or z) is present, writing that data triggers the operation.
- When two data inputs (x and y) are required, writing y triggers the operation.



## 12.4 Status flag

The CORDIC has 2 status flags to indicate the current operating status of the CORDIC, including the operation status flag BUSY and the operation completion flag EOC.

### Busy flag (CORDIC\_CSR.BUSY)

The CORDIC\_CSR.BUSY flag bit indicates whether the current CORDIC is in the finished operation or idle state. When CORDIC\_CSR.BUSY is 0, it means that CORDIC is in the end of operation or idle state, and can read the result or start a new operation. When CORDIC\_CSR.BUSY is 1, it means that CORDIC is in a busy state, the current data operation is in progress, and cannot provide the operation result or start a new operation.

### Operation completion flag (CORDIC\_CSR.EOC)

The CORDIC\_CSR.EOC flag indicates that the CORDIC operation is complete.

The setting of the EOC flag bit does not affect the continuation of a new round of operations, regardless of the type of the new round of operations. When a new round of operation is initiated, the flag bit is automatically reset to zero and re-indicated based on the situation of the new round of operation.



## 12.5 List of registers

CORDIC base address: CORDIC\_BASE = 0x4002 3400

Table 12-2 List of CORDIC registers

| Register name | Register address   | Register description        |
|---------------|--------------------|-----------------------------|
| CORDIC_CSR    | CORDIC_BASE + 0x00 | Control and status register |
| CORDIC_X      | CORDIC_BASE + 0x04 | Data X register             |
| CORDIC_Y      | CORDIC_BASE + 0x08 | Data Y register             |
| CORDIC_Z      | CORDIC_BASE + 0x0C | Data Z register             |



## 12.6 Register description

See section 1 [Documentation conventions](#) for abbreviations used in register descriptions.

### 12.6.1 CORDIC\_CSR Control and status register

Address offset: 0x00      Reset value: 0x0000 0000

| Bit field | Name    | Permission | Function description                                                                                                                                                                                                                                                                                                             |
|-----------|---------|------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 17        | BUSY    | RO         | Operation status flag<br>0: No operation was performed or the operation was completed.<br>1: In the process of calculation.                                                                                                                                                                                                      |
| 16        | EOC     | RO         | Operation completion flag<br>1: The operation has been completed and the result is ready to be read.<br>0: No result to be read.<br><i>Note: After the operation completes, the bit field is set to 1 by hardware. After reading the result, it is cleared to 0 by hardware.</i>                                                 |
| 15        | DMAIDLE | RW         | Entering idle state triggers DMA enable control<br>0: Disabled<br>1: Enabled<br><i>Note: The idle state refers to both BUSY and EOC being 0.</i>                                                                                                                                                                                 |
| 14        | DMAEOC  | RW         | Operation completion triggers DMA enable control<br>0: Disabled<br>1: Enabled                                                                                                                                                                                                                                                    |
| 13        | IE      | RW         | Operation completion interrupt enable control<br>0: Disable interrupt.<br>1: Enable interrupt.                                                                                                                                                                                                                                   |
| 12        | COMP    | RW         | Compensation strategy for the scaling factor K<br>0: The calculation result has been compensated by the hardware for the scaling factor K.<br>1: The calculation results did not compensate for the scaling factor K.<br><i>Note: It is only applicable to modulus and square root operations performed in the q1.31 format.</i> |
| 11:8      | ITER    | RW         | Iteration number configuration. If the register is configured to n, it will be iterated n times.<br>0000: 2<br>0001: 4<br>...<br>1111: 32                                                                                                                                                                                        |
| 7         | FORMAT  | RW         | Data format configuration<br>0: Use the q1.15 format.<br>1: Use the q1.31 format.                                                                                                                                                                                                                                                |



| Bit field | Name  | Permission | Function description                                                                                                                                                                                                                                                                                                                 |
|-----------|-------|------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 6:4       | SCALE | RW         | The extended range register is used to expand the input range of functions. For configuration details, please refer to the descriptions of each function.                                                                                                                                                                            |
| 3:0       | FUNC  | RW         | Configuration of operation functions<br>0000: cosine (cos)<br>0001: sine (sin)<br>0010: phase angle (atan2)<br>0011: modulus (hypot)<br>0100: arctangent (atan)<br>0101: hyperbolic cosine (cosh)<br>0110: hyperbolic sine (sinh)<br>0111: hyperbolic arctangent (atanh)<br>1000: Natural logarithm (ln)<br>1001: Square roots (qrt) |

### 12.6.2 CORDIC\_X Data X register

Address offset: 0x04      Reset value: 0x0000 0000

| Bit field | Name | Permission | Function description |
|-----------|------|------------|----------------------|
| 31:0      | X    | RW         | Data X register.     |

### 12.6.3 CORDIC\_Y Data Y register

Address offset: 0x08      Reset value: 0x0000 0000

| Bit field | Name | Permission | Function description |
|-----------|------|------------|----------------------|
| 31:0      | Y    | RW         | Data Y register.     |

### 12.6.4 CORDIC\_Z Data Z register

Address offset: 0x0C      Reset value: 0x0000 0000

| Bit field | Name | Permission | Function description |
|-----------|------|------------|----------------------|
| 31:0      | Z    | RW         | Data Z register.     |





## 13 Real time clock (RTC)

### 13.1 Overview

The Real time clock (RTC) is a dedicated counter/timer that provides calendar information including hours, minutes, seconds, subseconds, day, month, year, and week.

RTC has 2 independent alarm clocks, the time and date can be set in combination, and the alarm clock interrupt can be generated and output through the pin; it supports the time stamp function, which can be triggered by the pin, record the current date and time, and generate a time stamp interrupt at the same time; Support periodic interrupt; support automatic wake-up function, which can generate interrupts and output through pins; support 1Hz square wave and RTCOUT output functions; support internal clock calibration compensation.

The CW32L012 has a built-in independently calibrated RC clock source with a frequency of 32KHz to provide the drive clock for the RTC. The RTC can run in DeepSleep mode and is suitable for applications requiring low power consumption.



## 13.2 Main features

- Calendar function (BCD code format):
  - Subseconds, seconds, minutes, hours (support 12/24 time system).
  - Week, day, month and year, support automatic leap year correction.
- Interrupt function:
  - Alarm clock A, B interrupt, can be output through pin.
  - Cycle interruption, wide range: 0.5s to 1 month.
  - Automatic wake-up interrupt, range: 61 $\mu$ s ~ 145h, can be output through pins.
  - Timestamp interruption.
- Calibration compensation function:
  - Minimum clock error compensation step size, 0.119ppm.
  - 1Hz square wave output pin.
- Register lock protection to prevent accidental modification.
- Operates in Active mode, Sleep mode, and DeepSleep mode.

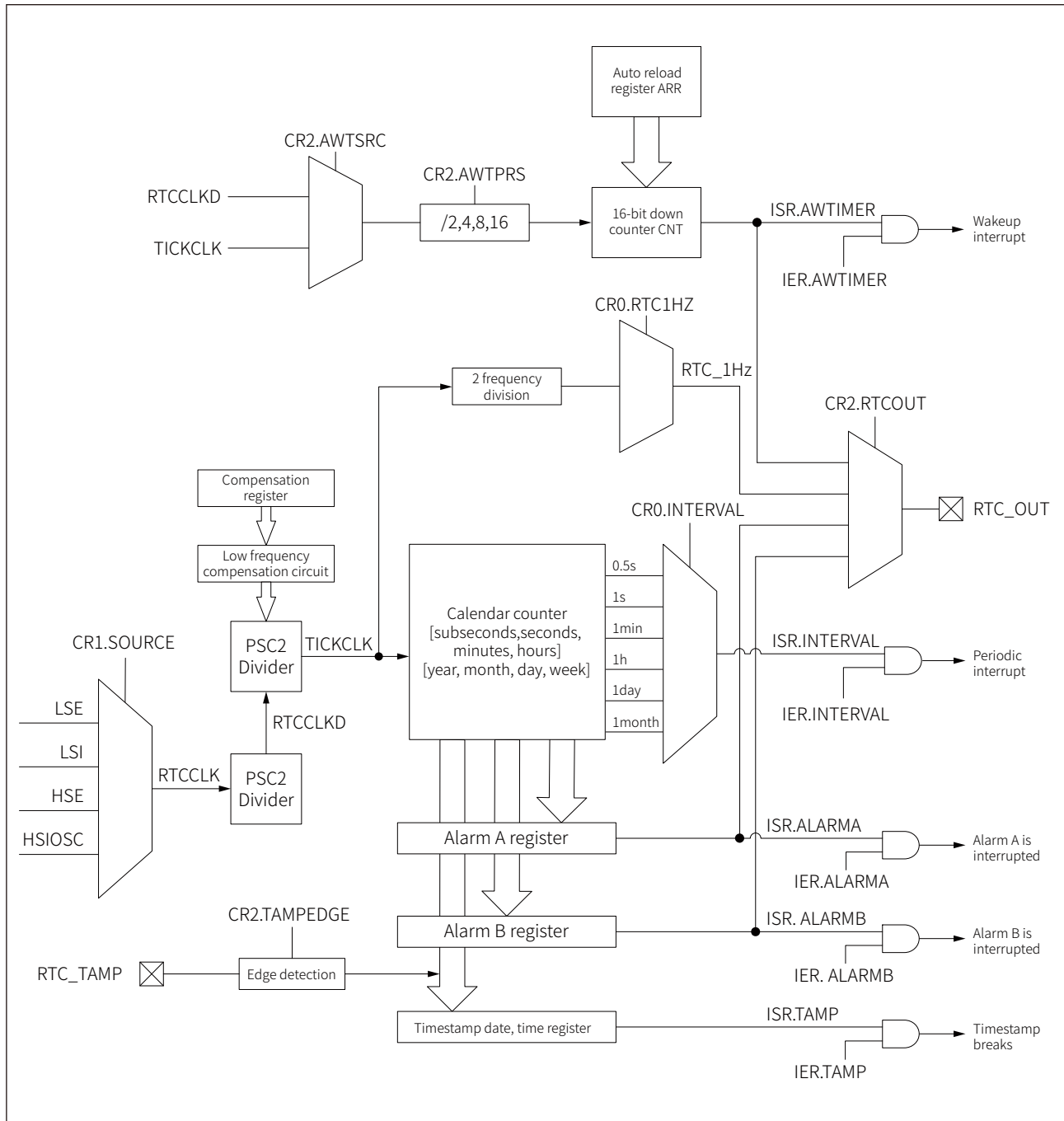


## 13.3 Functional description

### 13.3.1 Functional block diagram

The functional block diagram of the real-time clock (RTC) is shown in the following figure:

Figure 13-1 RTC functional block diagram



### 13.3.2 Clock and prescaler

The real-time clock (RTC) consists mainly of a dedicated high-precision RTC timer with clock source RTCCLK selectable from external low-speed clock LSE, internal low-speed clock LSI, external high-speed clock HSE and internal high-speed clock HSIOSC, selected through the SOURCE bit field of control register RTC\_CR1.

The prescaler is divided into 2 programmable prescalers: PSC1 and PSC2.

The clock after dividing by PSC1 is RTCCLKD, which can be used as the clock source of RTC auto wake-up timer, and the clock frequency after maximum dividing by PSC1 should be less than or equal to 1MHz. the calculation formula is as follows:

$$f_{\text{RTCCLKD}} = f_{\text{RTCCLK}} / (\text{PSC1} + 1)$$

The clock divided by PSC2 is TICKCLK, which is used to update the calendar and also as a clock source for the RTC auto wake-up timer. The calculation formula is as follows:

$$f_{\text{TICKCLK}} = f_{\text{RTCCLKD}} / (\text{PSC2} + 1)$$

The user needs to configure PSC1 and PSC2 appropriately so that  $f_{\text{TICKCLK}} = 2\text{Hz}$ .

The default value of PSC1 is 0 and the default value of PSC2 is 0x3FFF, so when the RTCCLK clock source is LSE, i.e., the clock frequency is 32768Hz, the timing clock is 1s.



### 13.3.3 Real time clock and calendar

The time register RTC\_TIME and the date register RTC\_DATE respectively record the current time and date value in BCD code format, and will automatically check the validity when writing them, any illegal time or date value will not be written, such as the 32 days, 24 hours, 61 seconds, 13 months, etc.

In the date register RTC\_DATE, the YEAR bit field represents the year, the valid value is 0 to 99; the MONTH bit field represents the month, the valid value is 1 to 12; the DAY bit field represents the day, the valid value is 1 to 31; the WEEK bit field represents the week, the valid value is 0 ~6, where 0 means Sunday and 1 to 6 means Monday to Saturday.

In the time register RTC\_TIME, the SECOND bit field represents seconds, with a valid value of 0 to 59; the MINUTE bit field represents minutes, with a valid value of 0 to 59; the HOUR bit field represents hours, with a valid value of 1 to 12 or 0 to 23, and 12 hours can be selected. system or 24-hour format. The H24 bit field of the control register RTC\_CR0 selects 12 or 24 hours:

- When H24 is '1', select 24-hour clock.
- When H24 is '0', 12 o'clock is selected, and the highest bit of the HOUR bit field represents AM/PM:
  - '0' means AM.
  - '1' means PM.

The HOUR bit field value is detailed in the following table:

Table 13-1 HOUR bit field values in 12/24 time system

| 12 time system (RTC_CR0.H24 = 0) |                |              |                | 24 time system (RTC_CR0.H24 = 1) |                |            |                |
|----------------------------------|----------------|--------------|----------------|----------------------------------|----------------|------------|----------------|
| Time                             | HOUR bit field | Time         | HOUR bit field | Time                             | HOUR bit field | Time       | HOUR bit field |
| AM12 o'clock                     | 0x12           | PM12 o'clock | 0x32           | 00 o'clock                       | 0x00           | 12 o'clock | 0x12           |
| AM1 o'clock                      | 0x01           | PM1 o'clock  | 0x21           | 01 o'clock                       | 0x01           | 13 o'clock | 0x13           |
| AM2 o'clock                      | 0x02           | PM2 o'clock  | 0x22           | 02 o'clock                       | 0x02           | 14 o'clock | 0x14           |
| AM3 o'clock                      | 0x03           | PM3 o'clock  | 0x23           | 03 o'clock                       | 0x03           | 15 o'clock | 0x15           |
| AM4 o'clock                      | 0x04           | PM4 o'clock  | 0x24           | 04 o'clock                       | 0x04           | 16 o'clock | 0x16           |
| AM5 o'clock                      | 0x05           | PM5 o'clock  | 0x25           | 05 o'clock                       | 0x05           | 17 o'clock | 0x17           |
| AM6 o'clock                      | 0x06           | PM6 o'clock  | 0x26           | 06 o'clock                       | 0x06           | 18 o'clock | 0x18           |
| AM7 o'clock                      | 0x07           | PM7 o'clock  | 0x27           | 07 o'clock                       | 0x07           | 19 o'clock | 0x19           |
| AM8 o'clock                      | 0x08           | PM8 o'clock  | 0x28           | 08 o'clock                       | 0x08           | 20 o'clock | 0x20           |
| AM9 o'clock                      | 0x09           | PM9 o'clock  | 0x29           | 09 o'clock                       | 0x09           | 21 o'clock | 0x21           |
| AM10 o'clock                     | 0x10           | PM10 o'clock | 0x30           | 10 o'clock                       | 0x10           | 22 o'clock | 0x22           |
| AM11 o'clock                     | 0x11           | PM11 o'clock | 0x31           | 11 o'clock                       | 0x11           | 23 o'clock | 0x23           |



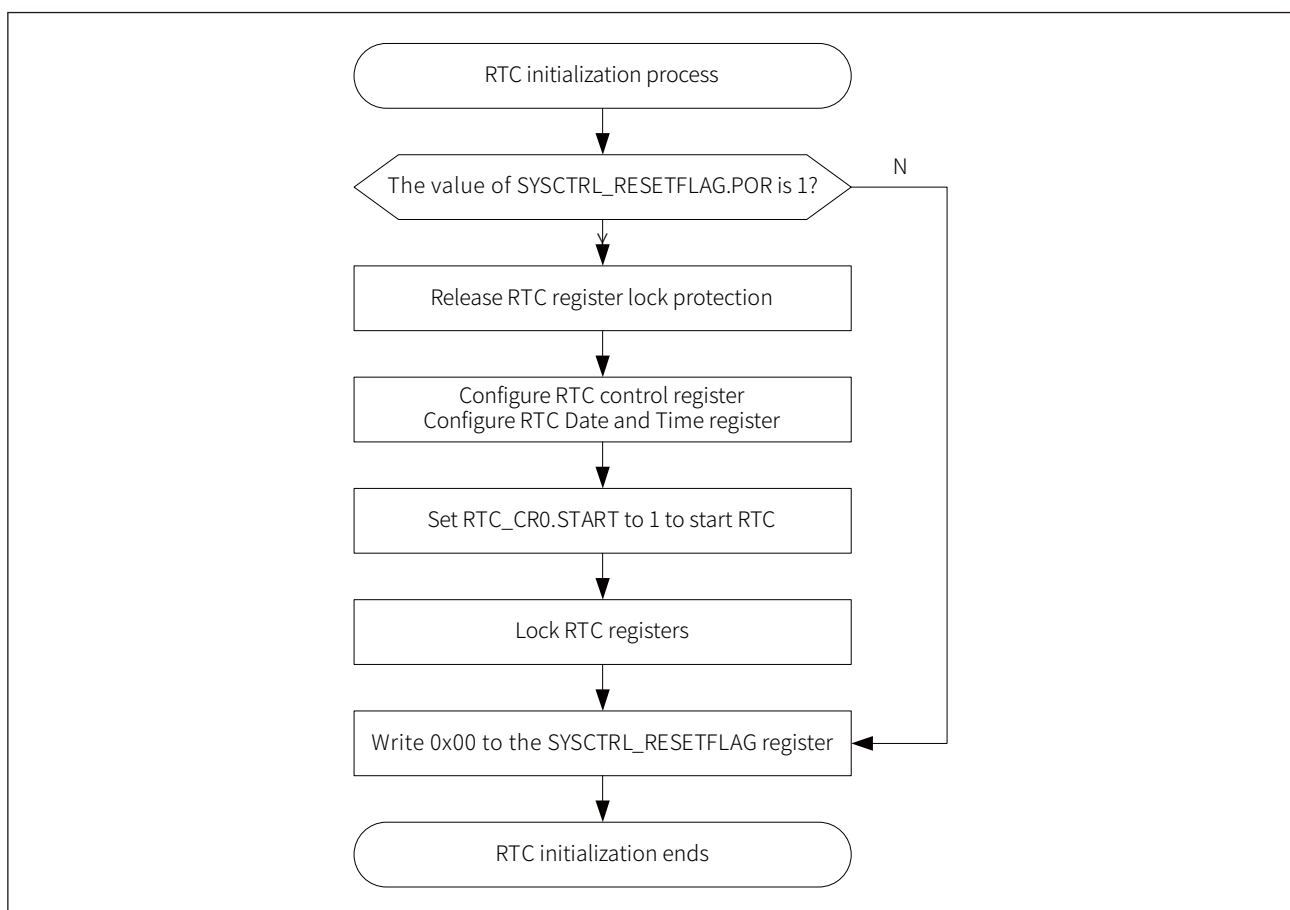
### 13.3.4 RTC initialization settings

The standard RTC module initialization process, shall include the following steps:

- Step 1:** Write 0xCA and 0x53 to the RTC\_KEY register sequentially to unlock the RTC register.
- Step 2:** Configure RTC\_CR0.H24 bit field, select 12/24 time system.
- Step 3:** Configure the RTC\_CR1.SOURCE bit field and select the RTC clock source.  
*Note: If you select LSE or LSI, you need to enable and start LSE or LSI first, and wait for the clock to stabilize.*
- Step 4:** Set the correct time and date values, write the time and date registers.
- Step 5:** Configure the required periodic interrupt unit, automatic wake-up unit, alarm clock A, and alarm clock B.
- Step 6:** Configure the RTC\_IER register, set the periodic interrupt, automatic wake-up interrupt, alarm A, alarm B interrupt.
- Step 7:** Configure the time stamp function and configure the time stamp interruption.
- Step 8:** The RTC\_CR0.START bit field is set to 1 to start the RTC timer.
- Step 9:** Write 0xCA, 0x-- to the RTC\_KEY register sequentially to start the RTC register lock protection.

In user applications, when initializing the RTC, the reset state (SYSCTRL\_RESETFLAG) of the MCU should be checked first. If it is a power-on reset (SYSCTRL\_RESETFLAG.POR = 1), the RTC registers will be reset to their default values. It is necessary to reconfigure the RTC control registers, RTC date and time registers, and restart the RTC; other reset conditions will not reset the RTC registers, the RTC continues to time normally.

Figure 13-2 RTC initialization process



### 13.3.5 Register lock function

After the CW32L012 is powered on and reset, the RTC registers except RTC\_KEY and RTC\_ICR are in lock protection state and cannot be modified. Write 0xCA and 0x53 keywords to the key value register RTC\_KEY in sequence to release the RTC register lock protection; write 0xCA and 0x-- to the RTC\_KEY register in sequence, the RTC register will be locked.

### 13.3.6 Register access operations

When the RTC is not running, after unlocking the registers, all registers can be read and written directly.

When the RTC is running, after unlocking the registers, all registers except timing related registers (RTC\_DATE, RTC\_TIME, RTC\_AWTARR) can be read and written directly.

When the RTC is running, since the running clock of the RTC counter may be out of sync with the clock of the RTC control register, in order to ensure that the contents of the RTC\_DATE, RTC\_TIME and RTC\_AWTARR registers can be accessed correctly without affecting the operation of the RTC, it is necessary to comply with the following operating instructions:

- Step 1:** The query waits for RTC\_CR1.WAIT to be 0, indicating that loading is completed and can be read or written.
- Step 2:** Write 0xCA and 0x53 to the RTC\_KEY register sequentially to unlock the RTC register.
- Step 3:** Execute normal RTC register access operation.
- Step 4:** The query waits for RTC\_CR1.WAIT to be 0, indicating that loading is completed.
- Step 5:** Write 0xCA, 0x-- to the RTC\_KEY register sequentially to enable the write protection of the RTC register.

*Note: If the user needs to read the RTC time and date information more quickly, he can choose to read the time or date register twice in a row. As long as the content of the two reads is the same, the read data can be considered valid.*

When the RTC is running, since the count clock of the RTC module is not synchronized with the read/write clocks of the registers, reading the sub-second count value register RTC\_SSCNT requires following specific operational steps to ensure that the correct sub-second value is read. The following is an example of reading a sub-second moment:

- Step 1:** Read the sub-second count value from the RTC\_SSCNT register and store it into the variable SS1.
- Step 2:** Read the sub-second count value from the RTC\_SSCNT register again and deposit it into the variable SS2.
- Step 3:** If SS1 and SS2 are not equal, jump to step 1.
- Step 4:** If SS1[19:0] is 0, skip to step 1.



### 13.3.7 RTCOUT output

By setting the RTCOUT bit field of the control register RTC\_CR2, the RTC\_OUT output source can be selected, as shown in the following table:

Table 13-2 RTC\_OUT output configuration

| RTC_CR2.RTCOUT | RTC_OUT output               |
|----------------|------------------------------|
| 00             | RTC_1Hz signal.              |
| 01             | Alarm clock A match flag.    |
| 10             | Alarm clock B match flag.    |
| 11             | Wake-up timer overflow flag. |

### 13.3.8 1Hz signal output

Configure the RTC\_1Hz output signal via the RTC1HZ bit field of control register RTC\_CR0 as shown in the following table:

Table 13-3 RTC\_1Hz output configuration

| Clock error compensation status                     | RTC_CR0.RTC1HZ | RTC_1Hz output                                                                                |
|-----------------------------------------------------|----------------|-----------------------------------------------------------------------------------------------|
| Disable error compensation<br>(RTC_COMPCFR1.EN = 0) | 01             | The 1Hz signal obtained by 2-division frequency signal of TICKCLK without clock compensation. |
| Enable error compensation<br>(RTC_COMPCFR1.EN = 1)  | 01             | The 1Hz signal obtained by 2-division frequency signal of TICKCLK with clock compensation.    |

When RTC\_COMPCFR1.EN is set to 0 and RTC\_CR0.RTC1HZ is 0x1, the signal output from RTC\_1Hz is the unlocked compensated 2-division frequency signal of TICKCLK, which is identical to the RTC's second counting clock. The duration of each cycle of this clock signal is identical.

When RTC\_COMPCFR1.EN is set to 1 and RTC\_CR0.RTC1HZ is 0x1, the signal output from RTC\_1Hz is the clock-compensated 2-division frequency signal of TICKCLK, which is identical to the RTC's second counting clock. The number of RTCCLKD clocks contained in each cycle of this clock signal is two integers in the neighbourhood of the value of the expression  $(PSC2+1) \times 2 \pm COMP/N$ , see [13.3.9 Clock error compensation](#).





### 13.3.9 Clock error compensation

The RTC has a built-in clock compensation circuit. When the RTCCLK accuracy cannot meet the requirements, the RTC clock can be compensated to achieve higher precision RTC timing.

The RTC clock error compensation configuration register, RTC\_COMPCFR1, has three bit fields for implementing error compensation:

1. SIGN, compensation direction:
  - When the TICKCLK clock frequency is greater than 2Hz, SIGN is set to 0, increasing the count value and compensating downwards.
  - When the TICKCLK clock frequency is less than 2Hz, SIGN is set to 1, decreasing the count value and compensating upwards.
2. PERIOD, compensation period, 32, 128, 256 seconds can be selected as a compensation period:
  - Set PERIOD to 00, select 32 seconds as one compensation period, which can achieve fast calibration but with lower accuracy, the compensation step is 0.950ppm and the compensation accuracy is 0.475ppm.
  - Set PERIOD to 01, select 128 seconds as one compensation period, the compensation step is 0.238ppm and the compensation accuracy is 0.119ppm.
  - Set PERIOD to 10, select 256 seconds as one compensation period, the adjustment process is the smoothest, the calibration accuracy is the highest, the compensation step is 0.119ppm and the compensation accuracy is 0.060ppm.
3. COMP, compensation value, the clock can be compensated range of  $\pm 488.0$ ppm:
  - When the compensation period is 32 seconds, the valid range of COMP is 1 ~ 511.
  - When the compensation period is 128 seconds, the valid range of COMP is 1 ~ 2047.
  - When the compensation period is 256 seconds, the valid range of COMP is 1 ~ 4095.

The operation steps of clock calibration are as follows:

- Step 1:** Turn off the error compensation function and set RTC\_COMPCFR1.EN to 0.
- Step 2:** According to [Table 9-2: GPIO multiplexing function assignment table](#), configure a GPIO port as RTC\_OUT output, and set RTC\_CR2.RTCOUT to 0 to output uncompensated 1Hz signal.
- Step 3:** According to section [13.3.4 RTC initialization settings](#), configure LSE as the RTC clock source.
- Step 4:** The RTC\_CR0.START bit field is set to 1 to start the RTC.
- Step 5:** Use a high-precision frequency meter to measure the 1Hz signal output by the RTC\_OUT pin, record the actual measurement frequency  $f_0$ .

The calculation method of the compensation value COMP:

Assuming the measured value  $f_0 = 0.9999888$ Hz,  $PSC2 = 0x3FFF$ , the actual frequency of the RTCCLKD clock  $f_z$ :

$$f_z = (PSC2+1) \times 2 \times 0.9999888 \approx 32767.63\text{Hz}$$

$$\text{Frequency error } \Delta f = (PSC2+1) \times 2 - f_z = 32768 - 32767.63 = 0.37$$

Select a compensation period of 256 seconds for the highest calibration accuracy and calculate the compensation value:

$$COMP = \Delta f \times PERIOD = 0.37 \times 256 = 94.72 \approx 95 = 0x5F$$

- Step 6:** According to section [13.3.6 Register access operations](#), configure the RTC\_COMPCFR1 register. In this example, write 1 to the SIGN bit field, write 10 to the PERIOD bit field, and the COMP bit field is 0x5F.
- Step 7:** Write RTC\_COMPCFR1.EN to 1 to start clock error compensation.

*Note: Steps 6 and 7 can also be combined, directly operate by halfword, and write 0xE05F to RTC\_COMPCFR1.EN.*



### 13.3.10 Alarm clock A and alarm clock B

The RTC supports 2 independent alarm clocks (alarm clock A and alarm clock B), which can generate alarm events at any time of the week, and generate alarm interrupts, and output the alarm matching events through the external RTC\_OUT pin. Set the ALARMAEN and ALARMBEN bit fields of the control register RTC\_CR2 to 1 to enable the alarm clock A and the alarm clock B respectively.

By setting the hour, minute, second match control bits HOUREN, MINUTEEN, SECONDEN and the hour, minute, second count value HOUR, MINUTE, SECOND of the alarm clock A and B control registers (RTC\_ALARMA and RTC\_ALARMB), the alarm clock can be set at 'xx time xx Minutes xx seconds', or 'xx minutes xx seconds' or 'xx hours xx minutes' or 'xx hours' and other combinations to generate alarm events; the alarm week enable control bit WEEKMASK can select any day of the week to generate an alarm event, bit0 represents Sunday, and bits 1 to 6 represent Monday to Saturday.

Using 12 or 24 time system, the setting value of the alarm control register RTC\_ALARMx (x = A,B) may be different. The example is as follows:

Table 13-4 Examples of timed alarm clocks

| Timing requirement                                                | Time system | RTC_ALARMx value | Alarm clock time |
|-------------------------------------------------------------------|-------------|------------------|------------------|
| Alarm clock at 6:30 a.m. every morning                            | 12/24 time  | 0x7F06 3000      | 06: 30: 00       |
| Alarm clock at 7:00 every weekday                                 | 12/24 time  | 0x3E07 0000      | 07: 00: 00       |
| Occurs every hour at 50 minutes                                   | 12/24 time  | 0x7F80 5000      | xx: 50: 00       |
| Occurs every 10 seconds per minute                                | 12/24 time  | 0x7F80 8010      | xx: xx: 10       |
| Every Monday, it occurs every five seconds per minute at 12 noon. | 12 time     | 0x0232 8005      | 12PM: xx: 05     |
| Every Monday, it occurs every five seconds per minute at 12 noon. | 24 time     | 0x0212 8005      | 12: xx: 05       |
| Every Sunday, it occurs per second at 12: 30 p.m.                 | 12 time     | 0x0112 3080      | 12AM: 30: xx     |
| Every Sunday, it occurs per second at 00: 30 p.m.                 | 24 time     | 0x0100 3080      | 00: 30: xx       |

When an alarm event occurs, the corresponding alarm match flag bit RTC\_ISR.ALARMx will be set to 1 by hardware. If the alarm interrupt enable bit RTC\_IER.ALARMx is set to 1, an interrupt request will be generated.

The match flag of alarm A or alarm B can be output through RTC\_OUT pin, please refer to section [13.3.7 RTCOUT output](#).



### 13.3.11 Period interrupt function

The RTC has a built-in period interrupt module, which can generate fixed-period interrupt signals. The timing period is configured by the INTERVAL bit field of the control register RTC\_CR0, as shown in the following table:

Table 13-5 Timing period configuration

| RTC_CR0. INTERVAL | Period of period interrupt                             |
|-------------------|--------------------------------------------------------|
| 000               | No period interrupt is generated.                      |
| 001               | 0.5 seconds, generated every 1 second and 0.5 seconds. |
| 010               | 1 second, generated when the seconds counter changes.  |
| 011               | 1 minute, generated when the minute counter changes.   |
| 100               | 1 hour, generated when the hour counter changes.       |
| 101               | 1 day, generated when the day counter changes.         |
| 11x               | 1 month, generated when the month counter changes.     |

Through the INTERVAL bit field of the RTC interrupt enable register RTC\_IER, it is possible to set whether the period interrupt generates an interrupt request.

### 13.3.12 Automatic wake-up function

The automatic wake-up timer is a 16-bit programmable automatic reload subtraction counter, with a selectable clock source, and the timing range is: 61μs ~ 145h. When the counter overflows, it can generate an automatic wake-up interrupt and output the overflow flag through the RTC\_OUT pin.

Set the AWTEN bit field of the control register RTC\_CR2 to 1 to enable the automatic wake-up function, which is specially designed for low-power applications and can work in all working modes of the MCU.

The AWTSRC and AWTPRS bit fields of RTC control register RTC\_CR2 are used to select the count clock source for the automatic wake-up timer. When AWTSRC is set to 0, RTCCLKD is selected as the count clock source; when AWTSRC is set to 1, TICKCLK is selected as the count clock source, at which time the RTC must be enabled. The count clock source can be divided by AWTPRS bit field, and the selectable division coefficients are 2, 4, 8, and 16.

The count period of the automatic wake-up timer is determined by the count clock source and the reload register RTC\_AWTARR. The formula for calculating the timing duration is:

$$\text{Auto wake-up timer period} = (\text{RTC\_AWTARR} + 1) / \text{wake-up timer clock frequency}$$

Shortest timing:  $(0+1)/16384\text{Hz} = 61\mu\text{s}$

Longest timing:  $(65535+1)/0.125\text{Hz} = 524288\text{s} = 8738\text{min} \approx 145.63\text{h}$

Through the AWTIMER bit field of the RTC interrupt enable register RTC\_IER, you can select whether to generate an interrupt request when the automatic wake-up timer overflows. The overflow flag of the automatic wake-up timer can be output through the RTC\_OUT pin, see section [13.3.7 RTCOUT output](#).

The user can read the current count value of the wake-up timer at any time, and reading the RTC\_AWTCNT register should be read multiple times until the same value is read twice.



### 13.3.13 Timestamp function

The RTC supports the timestamp function, that is, triggered by the RTC\_TAMP pin, the current time and date are saved to the timestamp date register RTC\_TAMPDATE and the timestamp time register RTC\_TAMPTIM, and a timestamp interrupt can be generated at the same time.

The TAMPEDGE bit field of the control register RTC\_CR2 is used to select whether the signal triggering the timestamp is valid on the rising edge or the falling edge, and the TAMPEN bit field of the RTC\_CR2 register is used to enable the timestamp function.

Users can flexibly select the trigger pin RTC\_TAMP, and need to configure this pin as a digital input and multiplexing function, please refer to the pin definition in the datasheet for more details.

When a timestamp event occurs, the timestamp event flag bit RTC\_ISR.TAMP will be set to 1. If the timestamp interrupt enable bit RTC\_IER.TAMP is set to 1, an interrupt request will be generated.

If after the first timestamp event occurs, the RTC\_ISR.TAMP flag is not cleared by software, and the second timestamp event occurs, the timestamp overflow flag RTC\_ISR.TAMPOV will be set to 1. If the timestamp overflow interrupt is set, the When the enable bit RTC\_IER.TAMPOV is 1, an interrupt request will be generated.

### 13.3.14 RTC interrupt

The RTC supports 6 interrupt sources: wake-up interrupt, alarm A interrupt, alarm B interrupt, period interrupt, timestamp interrupt and timestamp overflow interrupt.

The opening of the interrupt is controlled by the interrupt enable register RTC\_IER. Write '1' to the corresponding bit field to enable the interrupt, and write '0' to disable the interrupt; when an interrupt occurs, the hardware will automatically set the corresponding bit of the interrupt flag register RTC\_ISR; Write '0' to the corresponding bit of the register RTC\_ICR to clear the corresponding interrupt flag.

In the user application, after the RTC interrupt function is enabled, the same interrupt service routine will be called from the RTC interrupt entry (RTC\_IRQHandler) when all RTC interrupt events occur. The specific RTC interrupt source.

After the interrupt event processing is completed, before exiting the interrupt service routine, you must write '0' to the corresponding bit of the RTC interrupt flag clear register RTC\_ICR to clear the interrupt flag and avoid repeatedly entering the interrupt service routine.



## 13.4 List of registers

RTC base address: RTC\_BASE = 0x4000 4400

Table 13-6 List of RTC registers

| Register name | Register address | Register description                |
|---------------|------------------|-------------------------------------|
| RTC_KEY       | RTC_BASE + 0x00  | Key value register                  |
| RTC_CR0       | RTC_BASE + 0x04  | Control register 0                  |
| RTC_CR1       | RTC_BASE + 0x08  | Control register 1                  |
| RTC_CR2       | RTC_BASE + 0x0C  | Control register 2                  |
| RTC_DATE      | RTC_BASE + 0x14  | Date register                       |
| RTC_TIME      | RTC_BASE + 0x18  | Time register                       |
| RTC_ALARMA    | RTC_BASE + 0x1C  | Alarm clock A control register      |
| RTC_ALARMB    | RTC_BASE + 0x20  | Alarm clock B control register      |
| RTC_TAMPDATE  | RTC_BASE + 0x24  | Timestamp date register             |
| RTC_TAMPTIME  | RTC_BASE + 0x28  | Timestamp time register             |
| RTC_IER       | RTC_BASE + 0x30  | Interrupt enable register           |
| RTC_ISR       | RTC_BASE + 0x34  | Interrupt flag register             |
| RTC_ICR       | RTC_BASE + 0x38  | Interrupt flag clear register       |
| RTC_AWTARR    | RTC_BASE + 0x2C  | Wake-up timer reload value register |
| RTC_AWTCNT    | RTC_BASE + 0x3C  | Wake-up timer count value register  |
| RTC_PSC       | RTC_BASE + 0x40  | Clock prescaler register            |
| RTC_SSCNT     | RTC_BASE + 0x44  | Subsecond count register            |
| RTC_COMPCFR1  | RTC_BASE + 0x10  | Clock error compensation register 1 |



## 13.5 Register description

See section 1 [Documentation conventions](#) for abbreviations used in register descriptions.

### 13.5.1 RTC\_KEY Key value register

Address offset: 0x00      Reset value: 0x0000 0000

| Bit field | Name | Permission | Function description                                                                                                                                                                                                                                                                                                     |
|-----------|------|------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:8      | RFU  | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                            |
| 7:0       | KEY  | WO         | Key value register<br>Write "0x00CA" and "0x0053" to this register in turn to modify the register of the RTC module.<br>Write "0x00CA" and "0x----" to this register in turn, and the registers of the RTC module cannot be modified except for KEY and ICR.<br><i>Attention: "0x----" is any value except "0x0053".</i> |

### 13.5.2 RTC\_CR0 Control register 0

Address offset: 0x04      Reset value: 0x0000 0000

| Bit field | Name     | Permission | Function description                                                                                                                                                                                                                                                                                                                                                                                                                           |
|-----------|----------|------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:8      | RFU      | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                                                                                                                                  |
| 7         | START    | RW         | RTC timer operation control<br>0: Stop RTC timer.<br>1: Start RTC timer.                                                                                                                                                                                                                                                                                                                                                                       |
| 6:5       | RTC1HZ   | RW         | RTC_1Hz output signal configuration<br>01: 1Hz signal compensated by RTCCLK clock.<br>Other: Reserved.                                                                                                                                                                                                                                                                                                                                         |
| 4         | RFU      | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                                                                                                                                  |
| 3         | H24      | RW         | Timing system setting<br>0: 12 time system.<br>1: 24 time system.                                                                                                                                                                                                                                                                                                                                                                              |
| 2:0       | INTERVAL | RW         | Timing period setting<br>000: No timing period is generated.<br>001: 0.5 second, generated when the second timer changes and the 0.5 second moment.<br>010: 1 second, generated when the second timer changes.<br>011: 1 minute, generated when the minute timer changes.<br>100: 1 hour, generated when the hour timer changes.<br>101: 1 day, generated when the day timer changes.<br>11x: 1 month, generated when the month timer changes. |



### 13.5.3 RTC\_CR1 Control register 1

Address offset: 0x08      Reset value: 0x0000 0000

| Bit field | Name   | Permission | Function description                                                                                                                                                                                                 |
|-----------|--------|------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:11     | RFU    | -          | Reserved bits, please keep the default value.                                                                                                                                                                        |
| 10:8      | SOURCE | RW         | Timing clock RTCCLK source configuration<br>000: External low speed clock LSE.<br>001: External high speed clock HSE.<br>010: Internal low-speed clock LSI.<br>011: Internal high speed clock HSIOSC.                |
| 7:3       | RFU    | -          | Reserved bits, please keep the default value.                                                                                                                                                                        |
| 2         | WAIT   | RO         | RTC timing related register 1 loading status<br>0: Load completed, can read and write.<br>1: Load in progress, cannot read or write.<br><i>Note: Including SECOND, MINUTE, HOUR, WEEK, DAY, MONTH, YEAR, AWTARR.</i> |
| 1:0       | RFU    | -          | Reserved bits, please keep the default value.                                                                                                                                                                        |

*Note: The RTC timing related registers are the RTC\_DATE, RTC\_TIME and RTC\_AWTARR registers.*



### 13.5.4 RTC\_CR2 Control register 2

Address offset: 0x0C      Reset value: 0x0000 0000

| Bit field | Name     | Permission | Function description                                                                                                                                                                                 |
|-----------|----------|------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:11     | RFU      | -          | Reserved bits, please keep the default value.                                                                                                                                                        |
| 10        | ALARMBEN | RW         | Alarm clock B enable control<br>0: Disabled<br>1: Enabled                                                                                                                                            |
| 9         | ALARMAEN | RW         | Alarm clock A enable control<br>0: Disabled<br>1: Enabled                                                                                                                                            |
| 8         | RFU      | -          | Reserved bits, please keep the default value.                                                                                                                                                        |
| 7         | AWTEN    | RW         | Wake-up timer enable control<br>0: Disabled<br>1: Enabled                                                                                                                                            |
| 6         | TAMPEN   | RW         | Timestamp enable control<br>0: Disabled<br>1: Enabled                                                                                                                                                |
| 5:4       | RTCOUT   | RW         | RTC_OUT pin output signal configuration<br>00: RTC_1Hz signal, see RTC_CR0.RTC1HZ for details.<br>01: Alarm clock A match flag.<br>10: Alarm clock B match flag.<br>11: Wake-up timer overflow flag. |
| 3         | TAMPEDGE | RW         | Timestamp signal edge configuration<br>0: Record the current time on the rising edge of the RTC_TAMP pin.<br>1: Record the current time on the falling edge of the RTC_TAMP pin.                     |
| 2         | AWTSRC   | RW         | Wake-up timer count clock source configuration<br>0: RTCCLKD<br>1: TICKCLK (need to enable RTC)                                                                                                      |
| 1:0       | AWTPRS   | RW         | Wake-up timer count clock prescaler configuration<br>00: DIV2<br>01: DIV4<br>10: DIV8<br>11: DIV16                                                                                                   |





### 13.5.5 RTC\_PSC Clock prescaler register

Address offset: 0x40      Reset value: 0x0000 3FFF

| Bit field | Name | Permission | Function description                                                                                                                                                                                        |
|-----------|------|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:28     | RFU  | -          | Reserved bits, please keep the default value.                                                                                                                                                               |
| 27:20     | PSC1 | RW         | Timing clock prescaler 1<br>$f_{\text{RTCCCLK}} = f_{\text{RTCCCLK}} / (\text{PSC1} + 1)$<br><b>Attention: The clock frequency after the maximum division of PSC1 should be less than or equal to 1MHz.</b> |
| 19:0      | PSC2 | RW         | Timing clock prescaler 2<br>$f_{\text{TICKCLK}} = f_{\text{RTCCCLK}} / (\text{PSC2} + 1)$                                                                                                                   |

**Note:** User should configure PSC1 and PSC2 appropriately so that  $f_{\text{TICKCLK}} = 2\text{Hz}$ .

### 13.5.6 RTC\_DATE Date register

Address offset: 0x14      Reset value: 0x0000 0000

| Bit field | Name  | Permission | Function description                                                                                                      |
|-----------|-------|------------|---------------------------------------------------------------------------------------------------------------------------|
| 31:27     | RFU   | -          | Reserved bits, please keep the default value.                                                                             |
| 26:24     | WEEK  | RW         | Week count value<br>BCD code format, valid value range is 0~6.<br>0 represents Sunday, 1~6 represents Monday to Saturday. |
| 23:16     | YEAR  | RW         | Year count value<br>BCD code format, valid value range is 00~99.                                                          |
| 15:13     | RFU   | -          | Reserved bits, please keep the default value.                                                                             |
| 12:8      | MONTH | RW         | Month count value<br>BCD code format, valid value range is 1~12.                                                          |
| 7:6       | RFU   | -          | Reserved bits, please keep the default value.                                                                             |
| 5:0       | DAY   | RW         | Day count value<br>BCD code format, the valid value range is 1~28, 1~29, 1~30, 1~31.                                      |



### 13.5.7 RTC\_TIME Time register

Address offset: 0x18      Reset value: 0x0001 0000

| Bit field | Name   | Permission | Function description                                                                                                                                                                                                  |
|-----------|--------|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:22     | RFU    | -          | Reserved bits, please keep the default value.                                                                                                                                                                         |
| 21:16     | HOUR   | RW         | Hour count value<br>BCD code format.<br>24 time system, the valid value range of hour count is 0~23.<br>12 time system, the valid value range of hour count is 1~12 (AM),<br>21~32 (PM), bit21 is 0 for AM, 1 for PM. |
| 15        | RFU    | -          | Reserved bits, please keep the default value.                                                                                                                                                                         |
| 14:8      | MINUTE | RW         | Minute count value<br>BCD code format, valid value range is 0~59.                                                                                                                                                     |
| 7         | RFU    | -          | Reserved bits, please keep the default value.                                                                                                                                                                         |
| 6:0       | SECOND | RW         | Second count value<br>BCD code format, valid value range is 0~59.                                                                                                                                                     |

*Attention: Writing the RTC\_TIME register will clear the current value of the RTC\_SSCNT register.*

### 13.5.8 RTC\_SSCNT Subsecond count register

Address offset: 0x44      Reset value: 0x0000 0000

| Bit field | Name   | Permission | Function description                                                                                                           |
|-----------|--------|------------|--------------------------------------------------------------------------------------------------------------------------------|
| 31:21     | RFU    | -          | Reserved bits, please keep the default value.                                                                                  |
| 20        | SSCNT1 | RO         | Subsecond count 0.5 second flag<br>0: Current count is less than 0.5 seconds.<br>1: Current count is greater than 0.5 seconds. |
| 19:0      | SSCNT0 | RO         | Subsecond count less than 0.5 seconds<br>Subsecond count = (PSC2+1) × SSCNT1 + SSCNT0                                          |



### 13.5.9 RTC\_ALARM Alarm clock A control register

Address offset: 0x1C      Reset value: 0x0212 1000

| Bit field | Name     | Permission | Function description                                                                                                                                                            |
|-----------|----------|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31        | RFU      | -          | Reserved bits, please keep the default value.                                                                                                                                   |
| 30:24     | WEEKMASK | RW         | Alarm clock week enable control<br>Bit0 represents Sunday, bit1 ~ bit6 represent Monday to Saturday respectively.<br>If the corresponding bit is 1, the alarm clock is enabled. |
| 23        | HOUREN   | RW         | Alarm clock hour bit mask control<br>0: The alarm clock needs to check the hour count value.<br>1: The alarm clock has nothing to do with the hour count value.                 |
| 22        | RFU      | -          | Reserved bits, please keep the default value.                                                                                                                                   |
| 21:16     | HOUR     | RW         | Alarm clock hour count value, BCD format.                                                                                                                                       |
| 15        | MINUTEEN | RW         | Alarm clock minute bit mask control<br>0: The alarm clock needs to check the minute count value.<br>1: The alarm clock has nothing to do with the minute count value.           |
| 14:8      | MINUTE   | RW         | Alarm clock minute count value, BCD format.                                                                                                                                     |
| 7         | SECONDEN | RW         | Alarm clock second bit mask control<br>0: The alarm clock needs to check the second count value.<br>1: The alarm has nothing to do with the second count value.                 |
| 6:0       | SECOND   | RW         | Alarm clock second count value, BCD format.                                                                                                                                     |



### 13.5.10 RTC\_ALARM B Alarm clock B control register

Address offset: 0x20      Reset value: 0x0412 0000

| Bit field | Name     | Permission | Function description                                                                                                                                                            |
|-----------|----------|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31        | RFU      | -          | Reserved bits, please keep the default value.                                                                                                                                   |
| 30:24     | WEEKMASK | RW         | Alarm clock week enable control<br>Bit0 represents Sunday, bit1 ~ bit6 represent Monday to Saturday respectively.<br>If the corresponding bit is 1, the alarm clock is enabled. |
| 23        | HOUREN   | RW         | Alarm clock hour bit mask control<br>0: The alarm has nothing to do with the hour count value.<br>1: The alarm clock needs to check the hour count value.                       |
| 22        | RFU      | -          | Reserved bits, please keep the default value.                                                                                                                                   |
| 21:16     | HOUR     | RW         | Alarm hour count value in BCD format.                                                                                                                                           |
| 15        | MINUTEEN | RW         | Alarm clock minute bit mask control<br>0: The alarm has nothing to do with the minute count value.<br>1: The alarm clock needs to check the minute count value.                 |
| 14:8      | MINUTE   | RW         | Alarm clock minute count value, BCD format.                                                                                                                                     |
| 7         | SECONDEN | RW         | Alarm clock second bit mask control<br>0: The alarm has nothing to do with the second count value.<br>1: The alarm clock needs to check the second count value.                 |
| 6:0       | SECOND   | RW         | Alarm clock second count value, BCD format.                                                                                                                                     |

### 13.5.11 RTC\_TAMPDATE Timestamp date register

Address offset: 0x24      Reset value: 0x0000 0000

| Bit field | Name  | Permission | Function description                                        |
|-----------|-------|------------|-------------------------------------------------------------|
| 31:16     | RFU   | -          | Reserved bits, please keep the default value.               |
| 15:13     | WEEK  | RO         | Week count value when time stamp signal occurs, BCD format. |
| 12:8      | MONTH | RO         | Month count value when timestamp signal occurs, BCD format. |
| 7:6       | RFU   | -          | Reserved bits, please keep the default value.               |
| 5:0       | DAY   | RO         | Day count value when timestamp signal occurs, BCD format.   |



### 13.5.12 RTC\_TAMPTIME Timestamp time register

Address offset: 0x28      Reset value: 0x0000 0000

| Bit field | Name   | Permission | Function description                                             |
|-----------|--------|------------|------------------------------------------------------------------|
| 31:22     | RFU    | -          | Reserved bits, please keep the default value.                    |
| 21:16     | HOUR   | RO         | Hour count value when timestamp signal occurs, BCD format.       |
| 15        | RFU    | -          | Reserved bits, please keep the default value.                    |
| 14:8      | MINUTE | RO         | The minute count value when timestamp signal occurs, BCD format. |
| 7         | RFU    | -          | Reserved bits, please keep the default value.                    |
| 6:0       | SECOND | RO         | The second count value when timestamp signal occurs, BCD format. |

### 13.5.13 RTC\_IER Interrupt enable register

Address offset: 0x30      Reset value: 0x0000 0000

| Bit field | Name     | Permission | Function description                                                         |
|-----------|----------|------------|------------------------------------------------------------------------------|
| 31:7      | RFU      | -          | Reserved bits, please keep the default value.                                |
| 6         | INTERVAL | RW         | Timing period interrupt enable control<br>0: Disabled<br>1: Enabled          |
| 5         | RFU      | -          | Reserved bits, please keep the default value.                                |
| 4         | TAMPOV   | RW         | Timestamp overflow interrupt enable control<br>0: Disabled<br>1: Enabled     |
| 3         | TAMP     | RW         | Timestamp interrupt enable control<br>0: Disabled<br>1: Enabled              |
| 2         | AWTIMER  | RW         | Wake up timer overflow interrupt enable control<br>0: Disabled<br>1: Enabled |
| 1         | ALARMB   | RW         | Alarm clock B interrupt enable control<br>0: Disabled<br>1: Enabled          |
| 0         | ALARMA   | RW         | Alarm clock A interrupt enable control<br>0: Disabled<br>1: Enabled          |



### 13.5.14 RTC\_ISR Interrupt flag register

Address offset: 0x34      Reset value: 0x0000 0000

| Bit field | Name     | Permission | Function description                                                                                                                                        |
|-----------|----------|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:7      | RFU      | -          | Reserved bits, please keep the default value.                                                                                                               |
| 6         | INTERVAL | RO         | Periodic timing completion flag<br>0: No periodic timing completion event occurred.<br>1: The periodic timing completion event has occurred.                |
| 5         | RFU      | -          | Reserved bits, please keep the default value.                                                                                                               |
| 4         | TAMPOV   | RO         | Timestamp overflow flag<br>0: No timestamp overflow event occurred.<br>1: The timestamp overflow event has occurred.                                        |
| 3         | TAMP     | RO         | Timestamp event flag<br>0: No timestamp event occurred.<br>1: Timestamp event has occurred.                                                                 |
| 2         | AWTIMER  | RO         | Wake-up timer timing completion flag<br>0: No wake-up timer timing completion event occurred.<br>1: The wake-up timer timing completion event has occurred. |
| 1         | ALARMB   | RO         | Alarm clock B match flag<br>0: No alarm clock B match event occurred.<br>1: The alarm clock B match event has occurred.                                     |
| 0         | ALARMA   | RO         | Alarm clock A match flag<br>0: No alarm clock A match event occurred.<br>1: Alarm clock A match event has occurred.                                         |



### 13.5.15 RTC\_ICR Interrupt flag clear register

Address offset: 0x38      Reset value: 0x0000 007F

| Bit field | Name     | Permission | Function description                                                                                                |
|-----------|----------|------------|---------------------------------------------------------------------------------------------------------------------|
| 31:7      | RFU      | -          | Reserved bits, please keep the default value.                                                                       |
| 6         | INTERVAL | R1W0       | Periodic timing completion flag cleared<br>W0: Clear the periodic timing completion flag.<br>W1: No function.       |
| 5         | RFU      | -          | Reserved bits, please keep the default value.                                                                       |
| 4         | TAMPOV   | R1W0       | Timestamp overflow flag cleared<br>W0: Clear timestamp overflow flag.<br>W1: No function.                           |
| 3         | TAMP     | R1W0       | Timestamp event flag cleared<br>W0: Clear timestamp event flag.<br>W1: No function.                                 |
| 2         | AWTIMER  | R1W0       | Wake-up timer timing completion flag cleared<br>W0: Clear wake-up timer timing completion flag.<br>W1: No function. |
| 1         | ALARMB   | R1W0       | Alarm clock B match flag cleared<br>W0: Clear alarm clock B match flag.<br>W1: No function.                         |
| 0         | ALARMA   | R1W0       | Alarm clock A match flag cleared<br>W0: Clear alarm clock A match flag.<br>W1: No function.                         |

### 13.5.16 RTC\_AWTARR Wake-up timer reload value register

Address offset: 0x2C      Reset value: 0x0000 FFFF

| Bit field | Name | Permission | Function description                          |
|-----------|------|------------|-----------------------------------------------|
| 31:16     | RFU  | -          | Reserved bits, please keep the default value. |
| 15:0      | ARR  | RW         | Wake-up timer reload value.                   |



### 13.5.17 RTC\_AWTCNT Wake-up timer count value register

Address offset: 0x3C      Reset value: 0x0000 FFFF

| Bit field | Name | Permission | Function description                          |
|-----------|------|------------|-----------------------------------------------|
| 31:16     | RFU  | -          | Reserved bits, please keep the default value. |
| 15:0      | CNT  | RO         | Wake-up timer count value.                    |

*Attention: Reading data from this register should be read multiple times until the same value is read twice.*

### 13.5.18 RTC\_COMPCFR1 Wake-up clock error compensation register 1

Address offset: 0x10      Reset value: 0x0000 0000

| Bit field | Name   | Permission | Function description                                                                                                                                                                                                                                                                                                                                      |
|-----------|--------|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:16     | RFU    | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                                             |
| 15        | EN     | RW         | Clock error compensation enable control<br>0: Disable clock error compensation.<br>1: Enable clock error compensation.                                                                                                                                                                                                                                    |
| 14        | SIGN   | RW         | Clock compensation direction<br>0: Increase count value (TICKCLK clock frequency greater than 2Hz).<br>1: Decrease count value (TICKCLK clock frequency less than 2Hz).                                                                                                                                                                                   |
| 13:12     | PERIOD | RW         | Compensation period selection<br>00: 32 seconds, compensation value is COMP[8:0], compensation step 0.950ppm.<br>01: 128 seconds, compensation value is COMP[10:0], compensation step 0.238ppm.<br>10: 256 seconds, compensation value is COMP[11:0], compensation step 0.119ppm.<br><i>Note: Compensation accuracy is half of the compensation step.</i> |
| 11:0      | COMP   | RW         | Compensation Value<br>Compensable range $\pm 488.0$ ppm.                                                                                                                                                                                                                                                                                                  |





## 14 Basic timer (BTIM)

### 14.1 Overview

CW32L012 integrates 3 basic timers (BTIM) internally, each BTIM is completely independent and has the same function, each contains a 16-bit automatic reload counter and is driven by a programmable prescaler. BTIM supports 4 working modes: internal counting mode, external counting mode, triggered start mode and gated counting mode, and supports the interrupt generation and DMA request on the occurrence of update events and trigger events. Counter reset can be controlled by the reset input signal in different working modes.



## 14.2 Main features

- 16-bit auto-reload up counter.
- Programmable prescaler supports 1, 2, 3, 4, ..., 65536 frequency division.
- Supports single counting mode and continuous counting mode.
- Trigger input signal (TRGI) controls the timer for multiple slave modes.
- Trigger output signal (TRGO) for synchronisation between peripherals.
- Reset signal (RSTI) controls counter reset.
- Generates interrupts and DMA request when update and trigger events occur.

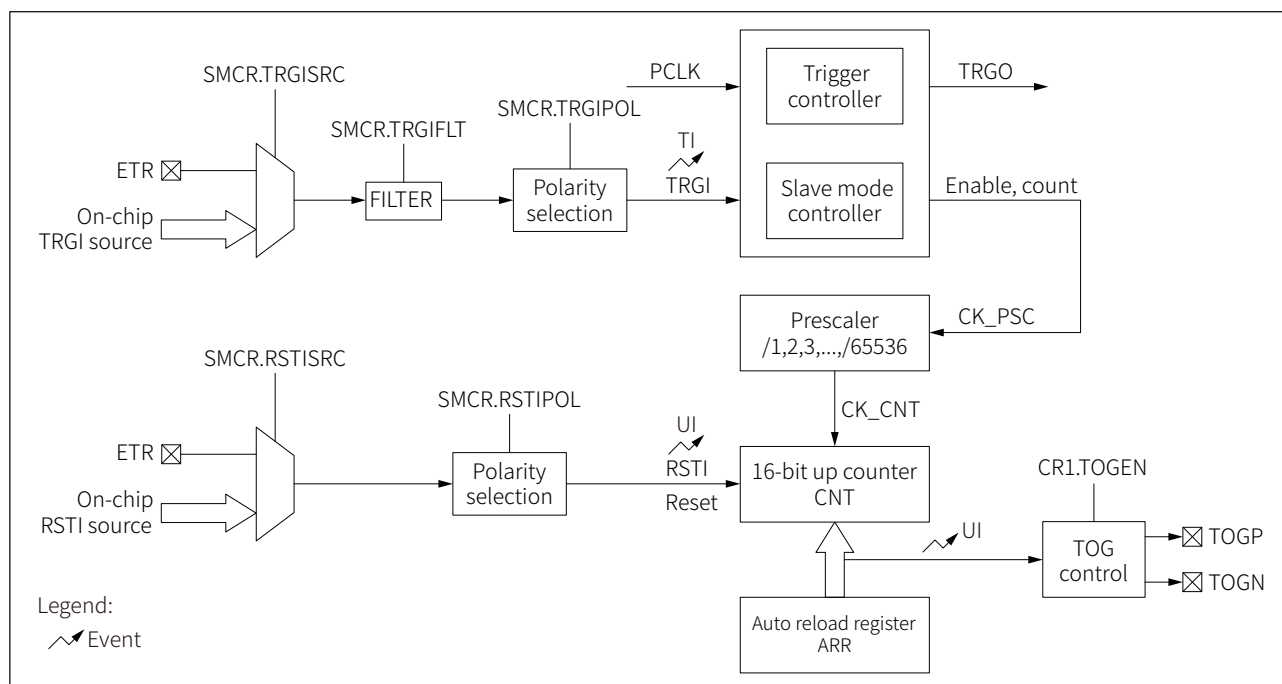


## 14.3 Functional description

### 14.3.1 Functional block diagram

The functional block diagram of BTIM is shown in the following figure:

Figure 14-1 BTIM functional block diagram



#### 14.3.1.1 Counting unit

The core components of the counting unit is a 16bit up-counter CNT and a 16bit auto-reload register ARR, and the counter clock can be divided by a prescaler. The counter register, the auto-reload register and the prescaler register can be read and written by software, and read and write operations can be performed even when the counter is running.

The prescaler divides the CK\_PSC clock to get the count clock CK\_CNT to drive the counter count. The divider coefficient is set through the BTIMx\_PSC register, which supports 1,2,3,4,...,65536 divider.

The counter can operate in either single count or continuous count mode, selected by the ONESHOT bit field of control register BTIMx\_CR1. When the EN bit field of the BTIMx\_CR1 register is set to 1, the counter starts counting up, noting that the actual counter enable signal, CNT\_EN, is set to 1 one clock cycle after EN is set to 1.

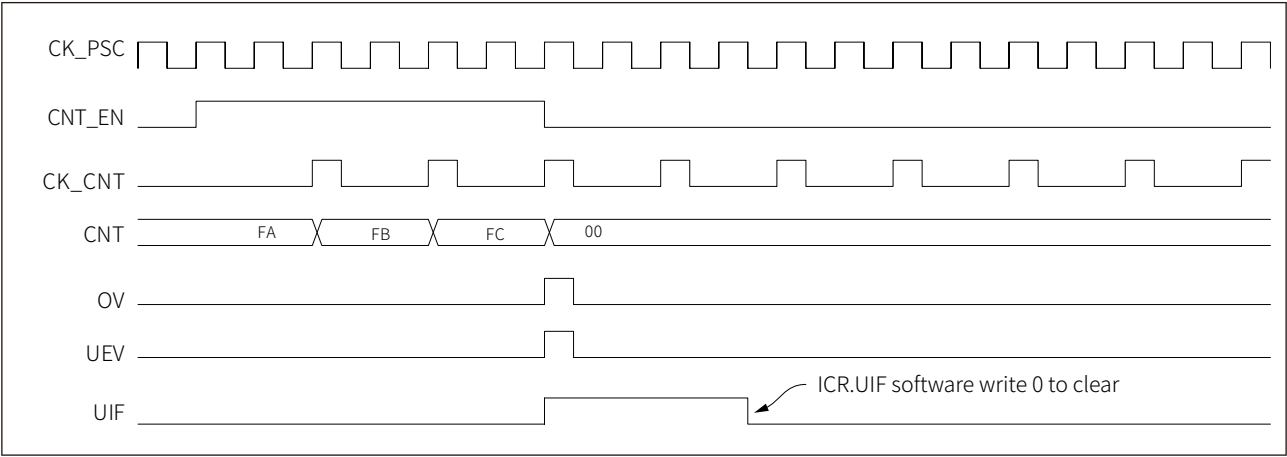
#### Single count mode

Set BTIMx\_CR1.ONESHOT to 1 to enable the timer work in single count mode. Set BTIMx\_CR1.EN to 1 to enable BTIMx, and the counter CNT counts up under the drive of the count clock CK\_CNT. When the count value reaches the reload value ARR, the overflow signal OV and the update event UEV are generated (the OV signal and UEV signal will be cleared automatically), the counter update interrupt flag bit BTIMx\_ISR.UIF is set by the hardware, and at the same time, the counter stops counting, and BTIMx\_CR1.EN is cleared by the hardware automatically.

The following figure shows an example of single count mode, where BTIMx\_PSC = 0x01 and BTIMx\_ARR = 0xFC:



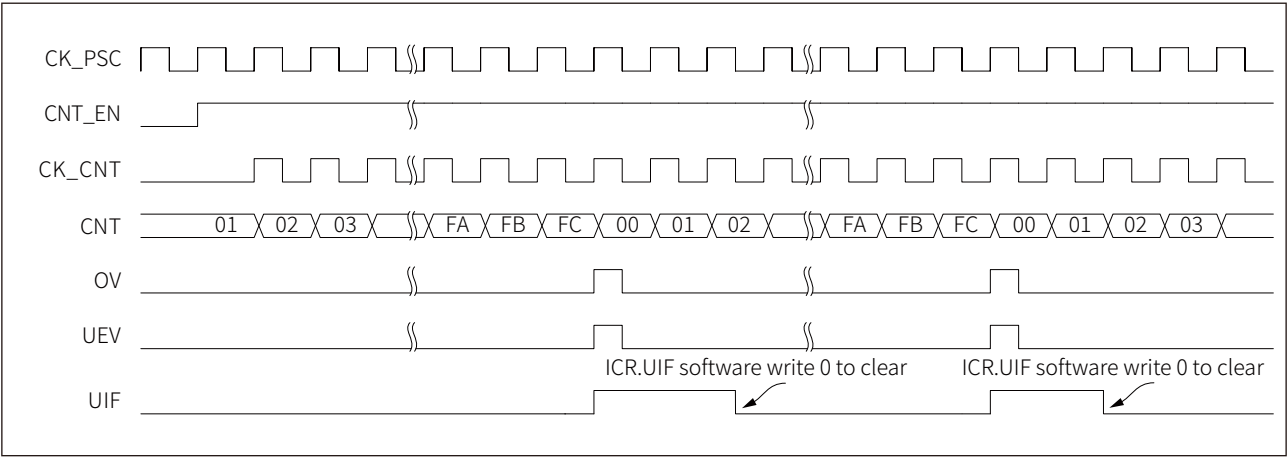
Figure 14-2 Single count mode



Continuous count mode

Set BTIMx\_CR1.ONESHOT to 0 to make the timer work in continuous counting mode. Set BTIMx\_CR.EN to 1 to enable BTIMx, and the counter CNT counts up under the drive of the count clock CK\_CNT. When the count value reaches the reload value ARR, the counter will start to count up again from 0 , and at the same time, the overflow signal OV and the update event UEV will be generated (the OV signal and the UEV signal will be cleared automatically), and the counter update interrupt flag bit, BTIMx\_ISR.UIF, will be set by the hardware. The following figure shows an example of continuous count mode, where BTIMx\_PSC = 0x00, BTIMx\_ARR = 0xFC:

Figure 14-3 Continuous count mode



### 14.3.1.2 Update event (UEV)

Allow update event to be disabled or enabled by the UDIS bit field of the control register BTIMx\_CR1.

#### Enable UEV

Set the UDIS bit field to 0 to enable UEV. At this time, update request sources can be selected according to the URS bit field, as shown in the following table:

Table 14-1 Update source settings

| BTIMx_CR1.URS | Update source                                                                                |
|---------------|----------------------------------------------------------------------------------------------|
| 0             | The counter overflows.<br>UG set.<br>Update events generated from the slave mode controller. |
| 1             | The counter overflows.                                                                       |

When an update event occurs, the following actions occur:

1. Reinitialize the counter, i.e., the counter is cleared to 0.
2. The prescaler counter is cleared to 0, but the prescaler ratio is not affected.
3. The event update interrupt flag bit BTIMx\_ISR.UIF is set by hardware.

When an update event UEV occurs, the event update interrupt flag BTIMx\_ISR.UIF is set by the hardware, and if the interrupt is allowed (BTIMx\_DIER.UIE is set to 1), an update interrupt request is generated, and BTIMx\_ICR.UIF is set to 0 to clear the flag.

#### Disable UEV

Set the UDIS bit field to 1 to disable UEV and no more update events will be generated.

If the UG bit field is set to 1, or a hardware reset is received from the slave mode controller, the counter and the counter of the prescaler are reinitialized.



### 14.3.1.3 Trigger input channel

The trigger input signal (TRGI) can be used as the count clock of the counter, as well as the trigger start signal and gate signal of the counter. For details, see [14.3.2 Operating modes](#). For details on the source of the TRGI signal, see the TRGISRC bit field description of the [14.10.3 BTIMx\\_SMCR Slave mode control register](#). The TRGI signal can be filtered and controlled, and the effective polarity of the TRGI signal can be set.

#### Filter

The filter is a digital filter that samples the trigger input signal at a certain frequency, and the signal is valid when it is continuously sampled to N same levels, otherwise the signal is invalid, so as to filter out the high-frequency clutter signal.

The sampling clock of the filter unit is PCLK or a division of PCLK, and the division ratio of PCLK and the number of sampling points N can be selected from the TRGIFLT bit field of the mode control register BTIMx\_SMCR.

#### Polarity selection

The TRGIPOL bit field of the slave mode control register BTIMx\_SMCR is used to select the effective polarity of the trigger input signal. When TRGIPOL is set to 0, the TRGI signal is not reversed and the high level or rising edge is valid. When TRGIPOL is set to 1, the TRGI signal is reversed and the low level or falling edge is valid.

### 14.3.1.4 Trigger output channel

Through the MMS bit field of the control register BTIMx\_CR2, you can select the trigger output signal (TRGO) to be sent to other peripherals for synchronization in BTIM master mode, as shown in the following table:

Table 14-2 BTIM trigger output (TRGO)

| BTIMx_CR2.MMS | TRGO signal           |
|---------------|-----------------------|
| 000           | Reset signal          |
| 001           | Counter enable signal |
| 010           | Update event          |
| 011           | Overflow signal       |

### 14.3.1.5 Reset input channel

BTIM supports to reset the counter by the valid edge of the reset input signal (RSTI) in different modes of operation, while generating update events.

Refer to the RSTISRC bit field description of the slave mode control register BTIMx\_SMCR for the exact source of the RSTI signal. The reset signal of the BTIMx\_ETR pin can be filtered and controlled by the ETRFLT bit field of the slave mode control register BTIMx\_SMCR, the valid polarity of the RSTI signal is set by the RSTIPOL bit field of the BTIMx\_SMCR register. When RSTIPOL is set to 0, the RSTI signal is not inverted and the rising edge is valid; When RSTIPOL is set to 1, the RSTI signal is inverted and the falling edge is valid.



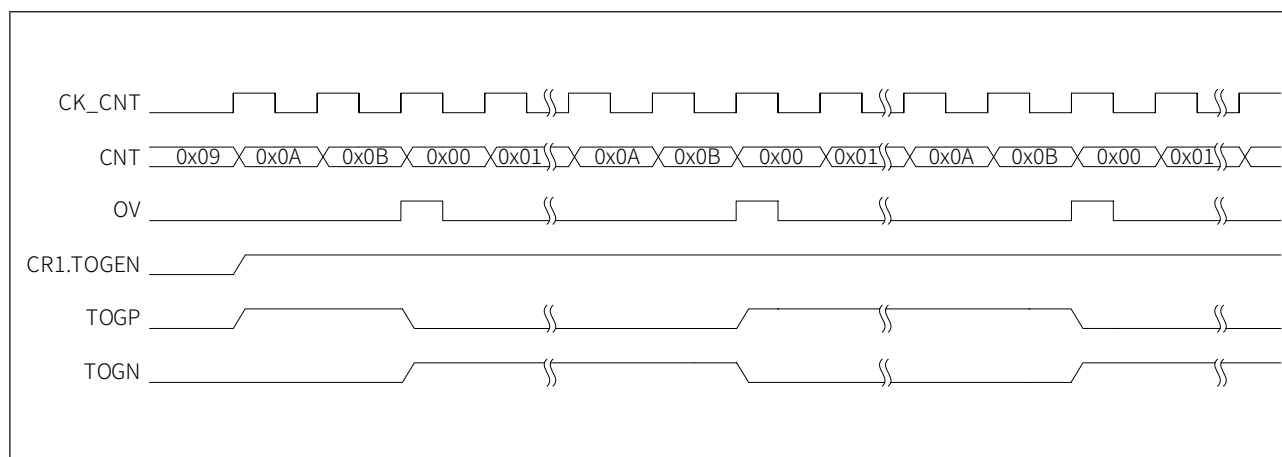
### 14.3.1.6 Flip output unit

The flip output unit can control the external BTIMx\_TOGP and BTIMx\_TOGN pins to output the flip signal through the ARR overflow signal OV.

- When BTIMx\_CR1.TOGEN is set to 0, both BTIMx\_TOGP and BTIMx\_TOGN pins output low level.
- When BTIMx\_CR1.TOGEN is set to 1, the BTIMx\_TOGP and BTIMx\_TOGN pins output signals with opposite levels (BTIMx\_TOGP default level is high); when the counter ARR overflows, the output levels of BTIMx\_TOGP and BTIMx\_TOGN pins will flip.

The following figure shows the schematic diagram of BTIMx\_TOGP and BTIMx\_TOGN pins level flip output in continuous counting mode:

Figure 14-4 Schematic diagram of level flip output (ARR = 0x0B)



### 14.3.2 Operating modes

The BTIM supports 4 operating modes: internal count mode, external count mode, trigger start mode, and gated count mode. Configured through the SMS bit field of the basic control register BTIMx\_SMCR.

Table 14-3 BTIM operating mode

| BTIMx_SMCR.SMS | Operating mode                         | Description                                                      |
|----------------|----------------------------------------|------------------------------------------------------------------|
| 000            | Disable slave mode, use internal clock | Clock source is PCLK.                                            |
| 001            | Trigger start mode                     | Clock source is PCLK, TRGI signal triggers the counter to start. |
| 010            | Gated count mode                       | Clock source is PCLK, TRGI signal as gate signal.                |
| 011            | External count mode                    | Clock source is TRGI signal.                                     |

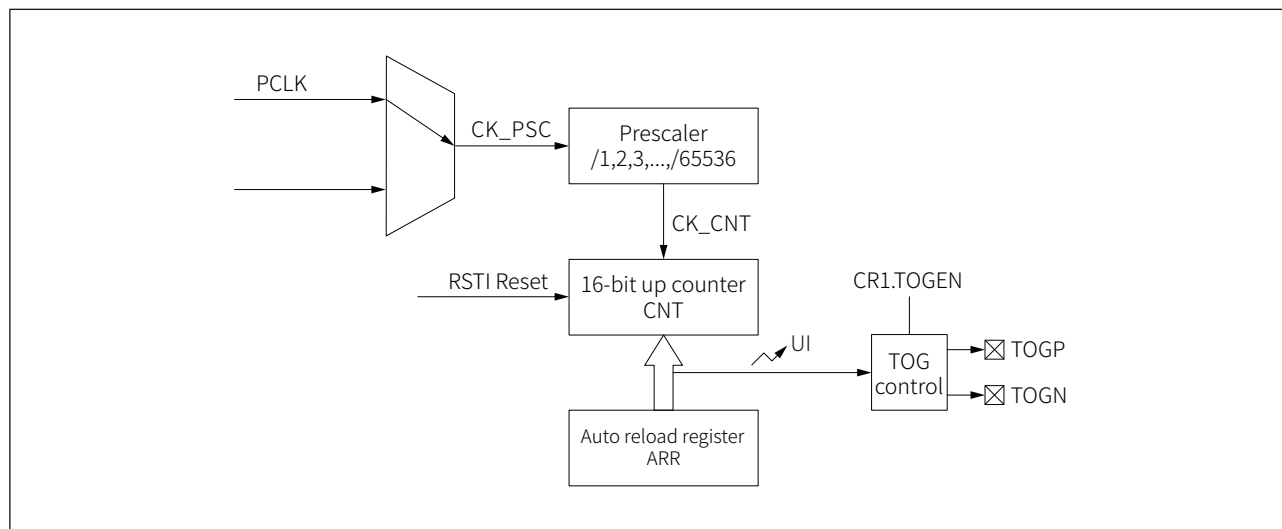




### 14.3.2.1 Internal count mode

When the SMS bit field of the slave mode control register BTIMx\_SMCR is 0x0, the timer slave mode is disabled and the prescaler clock is provided directly by the internal clock PCLK. Setting BTIMx\_CR1.EN to 1 will enable the counter to start counting. The internal count mode block diagram is shown below:

Figure 14-5 Internal count model block diagram



In practice, the frequency of the system clock PCLK is known, through the reasonable set of prescaler coefficients PSC can be counted for a fixed duration clock, with the setup of the reload value ARR and the use of the counter event update interrupt flag bit, you can accurately obtain a particular duration, so as to achieve the purpose of timing.

Timing time T calculation formula:

$$T = ((PSC+1)/PCLK) \times (ARR+1)$$

Where, PCLK is the counter clock source, PSC is the prescaler coefficient, and ARR is the reload value.

**Example,** When the frequency of the counter clock source PCLK is 24MHz, the timing is required to be 100ms. If the prescaler coefficient PSC is set to 0xFF, calculate:

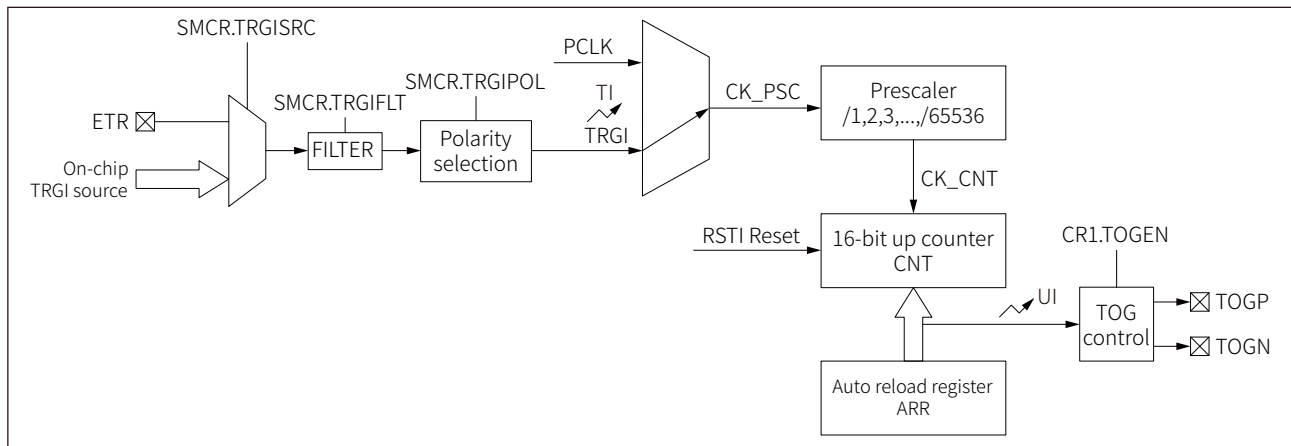
$$T = 100ms = ((255+1)/24MHz) \times (ARR+1)$$

then ARR = 9374 (0x249E). That is, you need to set the reload value ARR to 0x249E.

### 14.3.2.2 External count mode

When the SMS bit field of the slave mode control register BTIMx\_SMCR is 0x3, the counter clock is provided by the rising edge of the selected trigger signal (TRGI). There are various sources of the TRGI signal, which are specifically selected through the TRGISRC bit field of the BTIMx\_SMCR register, and can be filtered for control and polarity selection. The external counter mode block diagram is shown below:

Figure 14-6 External counter mode block diagram



Set BTIMx\_CR1.EN to 1 to enable the counter, which accumulates under the drive of the CK\_CNT clock signal. When TRGI has a valid edge, the BTIMx\_ISR.TIF flag is set to 1. Writing 0 to BTIMx\_ICR.TIF clears the flag.

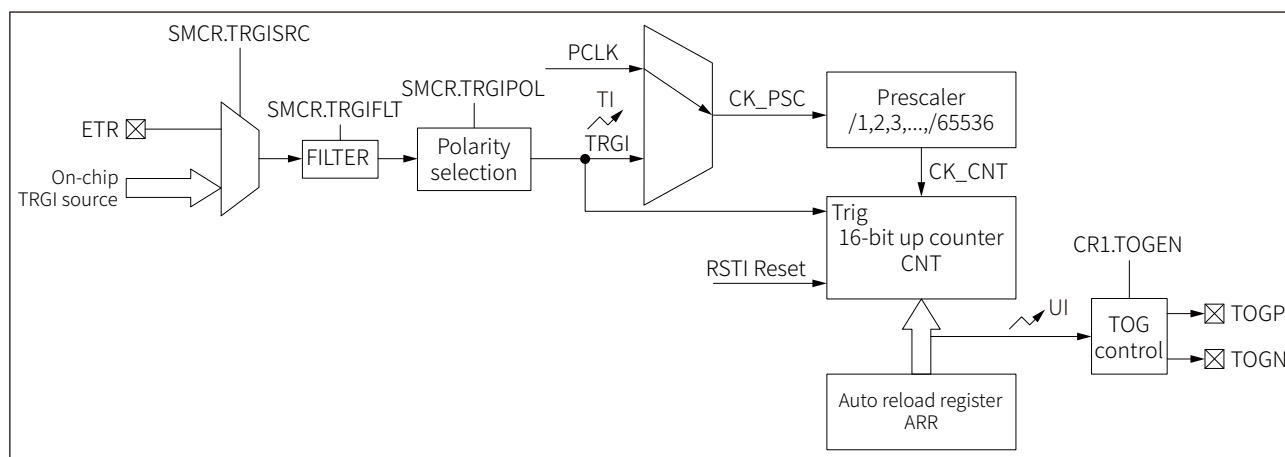
### 14.3.2.3 Trigger start mode

When the SMS bit field of the slave mode control register BTIMx\_SMCR is 0x1, the BTIM is configured to trigger start mode. In this mode, when BTIMx\_CR1.EN is set to 1 or a rising edge of the trigger signal TRGI occurs, the counter CNT will be enabled to count the CK\_CNT signal of the internal clock PCLK divided by the prescaler.

The source of the TRGI signal is controlled by the TRGISRC bit field of the slave mode control register BTIMx\_SMCR, and the TRGI signal filtering can be set by the TRGIFLT bit field, and the effective polarity of the TRGI signal can be set by the TRGIPOL bit field.

The trigger start mode block diagram is shown below:

Figure 14-7 Trigger start mode block diagram

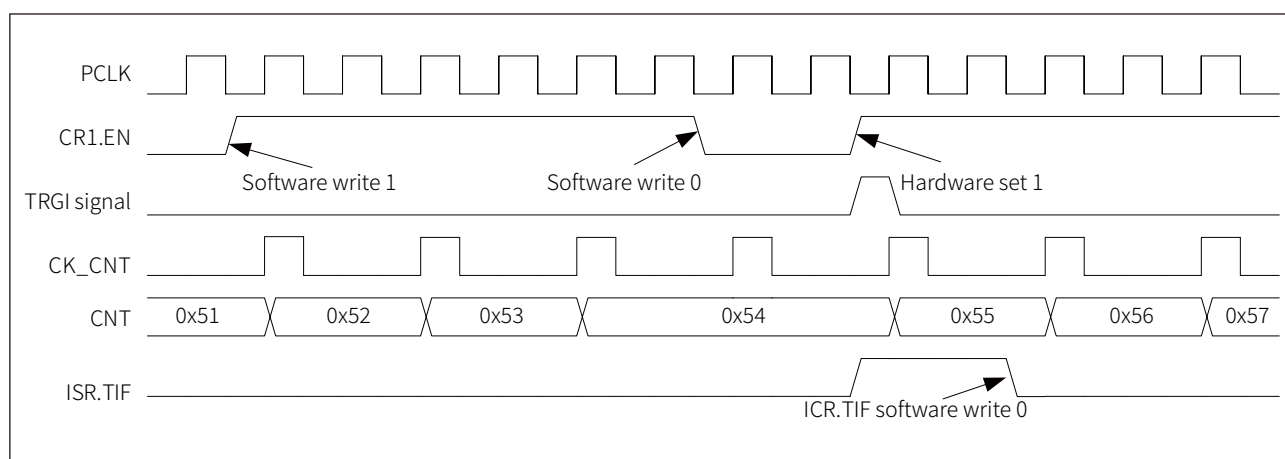


When a valid trigger signal is detected, the following effects occur:

1. BTIMx\_CR1.EN is set by hardware.
2. The trigger interrupt flag bit BTIMx\_ISR.TIF is set to 1, interrupt or DMA request can be generated.
3. The counter starts and starts counting.

After the counter is started, the counter counts upwards from the initial value and generates an overflow when the count value reaches the reload value ARR. When the enabled control bit BTIMx\_CR1.EN is set to 0 at any time, the counter immediately stops counting; When the enabled control bit BTIMx\_CR1.EN is set to 1 again or the trigger signal is valid, the counter continues to count according to the previous setting. The trigger start mode timing diagram is shown below, where BTIMx\_PSC = 0x01:

Figure 14-8 Trigger start mode timing diagram



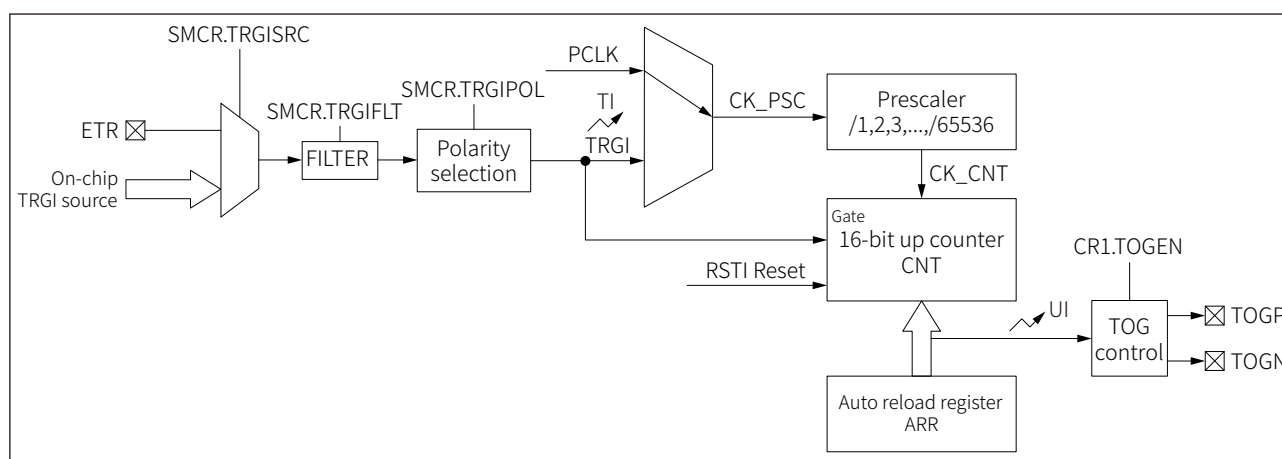
### 14.3.2.4 Gated count mode

When the SMS bit field of the slave mode control register BTIMx\_SMCR is set to 0x2, the BTIM is configured in gated counter mode. In this mode, when the trigger input (TRGI) is high and BTIMx\_CR1.EN is 1, the counter will be activated to count the CK\_CNT signal after the internal clock PCLK is divided by the prescaler; and when the trigger input (TRGI) is low or BTIMx\_CR1.EN is 0, the counter will stop counting immediately.

The source of the TRGI signal is controlled by the TRGISRC bit field of the slave mode control register BTIMx\_SMCR, and the TRGI signal filter can be set via the TRGIFLT bit field, and the effective polarity of the TRGI signal can be set via the TRGIPOL bit field.

The gated count mode block diagram is shown below:

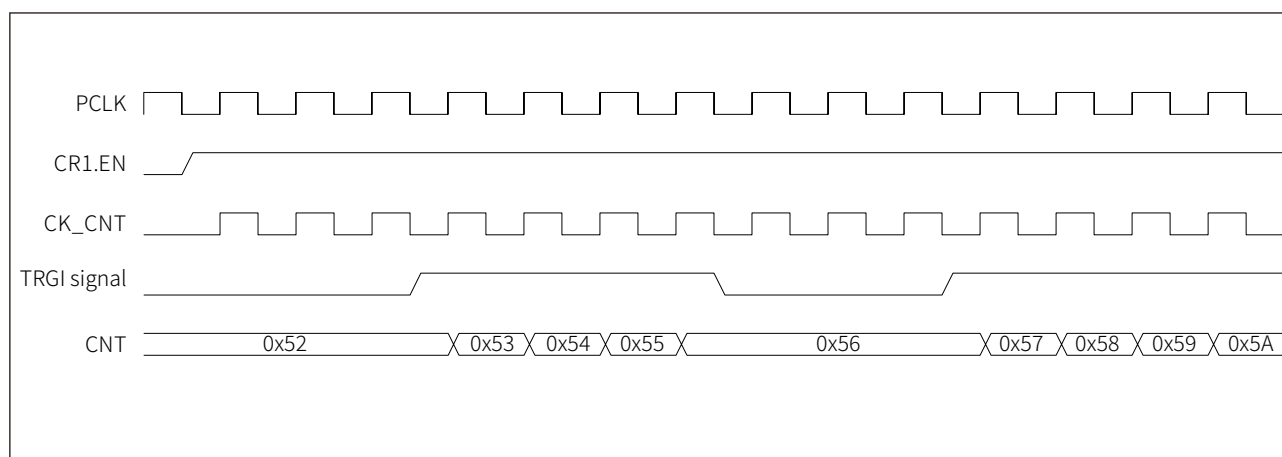
Figure 14-9 Gated count mode block diagram



After the counter is started, the counter starts to count up from the initial value. When the count value reaches the reload value ARR, an overflow occurs. When the running control bit BTIMx\_CR1.EN is set to 0 at any time or the gate control signal is invalid at any time, the counter immediately stops counting. When the running control bit BTIMx\_CR1.EN is set to 1 again and the gate control signal is valid, the counter continues to count according to the previous setting.

The gated count mode timing diagram is shown in the following figure, where BTIMx\_PSC = 0x00:

Figure 14-10 Gated counter mode timing diagram



### 14.3.3 UIF bit remap

Set the UIFREMAP bit in the BTIMx\_CR1 register to 1 to enable the UIF state bit remapping function, which can force the update interrupt flag BTIMx\_ISR.UIF to continuously copy to the UIFCPY bit field of the BTIMx\_CNT register. This allows the counter value and the potential flip-flop condition issued by the UIFCPY flag to be read automatically. In specific cases, this simplifies calculations by avoiding competing conditions when processing is shared between background tasks (counter reads) and interrupts (update interrupts).

There is no delay between UIF and UIFCPY flag enable.

When BTIMx\_CR1.UIFREMAP is 0, the UIFCPY bit field is reserved and read as 0.



## 14.4 BTIM interrupt

The BTIM supports 2 interrupt sources. When a BTIM interrupt event occurs, the interrupt flag bit will be set by hardware, and an interrupt request will be generated if the corresponding interrupt enable control bit is set.

In the user BTIM interrupt service procedure, the relevant BTIM interrupt flag bit should be queried for corresponding processing, and the interrupt flag bit should be cleared before exiting the interrupt service procedure to avoid entering the interrupt service procedure repeatedly.

The flag bit, interrupt enable bit, interrupt flag clear bit or clear method of each BTIM interrupt source are shown in the following table:

Table 14-4 BTIM interrupt control

| Interrupt event   | Interrupt flag bit | Interrupt enable bit | Interrupt flag clear |
|-------------------|--------------------|----------------------|----------------------|
| Update interrupt  | ISR.UIF            | DIER.UIE             | Write 0 to ICR.UIF   |
| Trigger interrupt | ISR.TIF            | DIER.TIE             | Write 0 to ICR.TIF   |



## 14.5 Trigger ADC

The trigger output signal TRGO of the BTIM can be used to trigger the ADC startup, and the trigger output TRGO has a variety of possible events, which are selected by the MMS bit field of the control register BTIMx\_CR2. At the same time, the ADC peripheral needs to configure its external trigger start register to select the corresponding trigger source.

It should be noted that the ADC clock must be enabled before an event can be received from the main timer and that the ADC clock must not be changed in real time when the trigger signal is received from the timer.



## 14.6 Direct memory access (DMA)

BTIM supports generating DMA requests when trigger and update events occur.

The DMA/Interrupt enable register BTIMx\_DIER related bit field is used to enable the DMA request. The user needs to set the corresponding DMA request enable bit in the BTIMx\_DIER register to 1 and set the HARDSRC bit field of the DMA trigger source control register DMA\_TRIGy to the corresponding trigger source.





## 14.7 Debugging support

The BTIM supports to stop or continue counting in debug mode, which is set by the BTIM123 bit field of the debug status timer control register SYSCTRL\_DEBUG:

- Set SYSCTRL\_DEBUG.BTIM123 to 1, then suspend the counter counting of BTIM1, BTIM2 and BTIM3 in the debug state.
- Set SYSCTRL\_DEBUG.BTIM123 to 0, then the counters of BTIM1, BTIM2 and BTIM3 continue to count in the debug state.



## 14.8 Programming examples

### 14.8.1 Internal count mode programming example

- Step 1:** Set `SYSCTRL_APBEN2.BTIM123` to 1, and enable the configuration clock and working clock of BTIM1, BTIM2, and BTIM3.  
*Note: BTIM1, BTIM2 and BTIM3 share the SYSCTRL\_APBEN2.BTIM123 bit.*
- Step 2:** Set `BTIMx_SMCR.SMS` to 0x00 to make BTIMx work in internal count mode.
- Step 3:** Configure `BTIMx_CR1.ONESHOT` to select single or continuous count mode. If it is configured as 0, it is continuous count mode, and if it is configured as 1, it is single count mode.
- Step 4:** Configure `BTIMx_PSC`, set the prescaler division ratio.
- Step 5:** Configure `BTIMx_ARR`, set BTIMx count overflow time.
- Step 6:** Set `BTIMx_CNT` to 0x0000 for counting.
- Step 7:** Set `BTIMx_DIER.UIE` to 1 and configure the corresponding NVIC to enable the update interrupt.
- Step 8:** Set `BTIMx_CR1.EN` to 1, start BTIMx.
- Step 9:** When the counter overflows, the `BTIMx_ISR.UIF` flag bit is 1, enter the interrupt service routine and set `BTIMx_ICR.UIF` to 0 to clear this interrupt flag.

### 14.8.2 External count mode programming example

Configure the counter to count on the rising edge of the `BTIMx_ETR` pin. The operation steps are as follows:

- Step 1:** Set `SYSCTRL_APBEN2.BTIM123` to 1, and enable the configuration clock and working clock of BTIM1, BTIM2, and BTIM3.  
*Note: BTIM1, BTIM2 and BTIM3 share the SYSCTRL\_APBEN2.BTIM123 bit.*
- Step 2:** Configure the GPIO corresponding to the `BTIMx_ETR` pin to multiplexed input mode. For details about register configuration, see [9 General-purpose input/output \(GPIO\)](#).
- Step 3:** Set `BTIMx_SMCR.TRGISRC` to 0xF, and select the source of TRGI signal as `BTIMx_ETR` pin.
- Step 4:** Set `BTIMx_SMCR.TRGIFLT` to configure the TRGI input signal filter bandwidth.
- Step 5:** Set `BTIMx_SMCR.TRGIPO` to 0 and select the rising edge to be valid.
- Step 6:** Set `BTIMx_SMCR.SMS` to 0x3 to make BTIMx work in external count mode.
- Step 7:** Configure `BTIMx_CR1.ONESHOT` to select single or continuous count mode. If configured to 0, it is continuous count mode, and if configured to 1, it is single count mode.
- Step 8:** Configure `BTIMx_PSC` to set the prescaler division ratio.
- Step 9:** Configure `BTIMx_ARR` to set the BTIMx count overflow time.
- Step 10:** Set `BTIMx_DIER.UIE` to 1 and configure the corresponding NVIC to enable the update interrupt.
- Step 11:** Set `BTIMx_CR1.EN` to 1, start BTIMx.
- Step 12:** When the counter overflows, the `BTIMx_ISR.UIF` flag bit is set to 1 to enter the interrupt service routine, and setting `BTIMx_ICR.UIF` to 0 clears this interrupt flag.



### 14.8.3 Trigger start mode programming example

The BTIMx\_ETR input triggers the start of the counter when a rising edge appears. The operation steps are as follows:

- Step 1:** Set SYSCTRL\_APBEN2.BTIM123 to 1, and enable the configuration clock and working clock of BTIM1, BTIM2, and BTIM3.  
*Note: BTIM1, BTIM2 and BTIM3 share the SYSCTRL\_APBEN2.BTIM123 bit.*
- Step 2:** Configure the GPIO corresponding to the BTIMx\_ETR pin to multiplexed input mode. For details about register configuration, see [9 General-purpose input/output \(GPIO\)](#).
- Step 3:** Set BTIMx\_SMCR.TRGISRC to 0xF, and select the source of TRGI signal as BTIMx\_ETR pin.
- Step 4:** Set BTIMx\_SMCR.TRGIFLT to configure the TRGI input signal filter bandwidth.
- Step 5:** Set BTIMx\_SMCR.TRGIPOL to 0 and select the rising edge to be valid.
- Step 6:** Set BTIMx\_SMCR.SMS to 0x1 to make BTIMx work in trigger start mode.
- Step 7:** Configure BTIMx\_CR1.ONESHOT to select single or continuous count mode. If configured to 0, it is continuous count mode, and if configured to 1, it is single count mode.
- Step 8:** Configure BTIMx\_PSC to set the prescaler division ratio.
- Step 9:** Configure BTIMx\_ARR to set the BTIMx count overflow time.
- Step 10:** When a rising edge occurs on the BTIMx\_ETR input, the counter starts counting and the TIF flag is set to 1.

### 14.8.4 Gated count mode programming example

The signal input from the external BTIM\_ETR pin is used as the gated control signal to control the count of the counter. The operation steps are as follows:

- Step 1:** Set SYSCTRL\_APBEN2.BTIM123 to 1, and enable the configuration clock and working clock of BTIM1, BTIM2, and BTIM3.  
*Note: BTIM1, BTIM2 and BTIM3 share the SYSCTRL\_APBEN2.BTIM123 bit.*
- Step 2:** Configure the GPIO corresponding to the BTIMx\_ETR pin to multiplexed input mode. For details about register configuration, see [9 General-purpose input/output \(GPIO\)](#).
- Step 3:** Set BTIMx\_SMCR.TRGISRC to 0xF, and select the source of TRGI signal as BTIMx\_ETR pin.
- Step 4:** Set BTIMx\_SMCR.TRGIFLT to configure the TRGI input signal filter bandwidth.
- Step 5:** Set BTIMx\_SMCR.TRGIPOL to 0 and select high level to be valid.
- Step 6:** Set BTIMx\_SMCR.SMS to 0x2 to make BTIMx work in gated count mode.
- Step 7:** Configure BTIMx\_CR1.ONESHOT to select single or continuous count mode. If configured to 0, it is continuous count mode, and if configured to 1, it is single count mode.
- Step 8:** Configure BTIMx\_PSC to set the prescaler division ratio.
- Step 9:** Configure BTIMx\_ARR to set the BTIMx count overflow time.
- Step 10:** Set BTIMx\_CR1.EN to 1 to enable counter.
- Step 11:** The counter starts counting when the BTIMx\_ETR input is a high level and stops counting when the BTIMx\_ETR input is a low level.



### 14.8.5 Counter reset programming example

The counter is reset when a rising edge occurs on the RSTI input. The operation steps are as follows:

- Step 1:** Configure the BTIM operating mode according to the programming example.
- Step 2:** Configure BTIMx\_SMCR.RSTISRC to select the source of the counter reset signal RSTI.
- Step 3:** Setting BTIMx\_SMCR.RSTIPOL to 0 to select rising edge validity.
- Step 4:** When a rising edge occurs on the RSTI input, the counter is cleared to 0 and starts counting from 0 again, while the UIF flag is set to 1.



## 14.9 List of registers

BTIM1 base address: BTIM1\_BASE = 0x4000 4800

BTIM2 base address: BTIM2\_BASE = 0x4000 4840

BTIM3 base address: BTIM3\_BASE = 0x4000 4880

Table 14-5 List of BTIM registers

| Register name | Register address  | Register description          |
|---------------|-------------------|-------------------------------|
| BTIMx_CR1     | BTIMx_BASE + 0x00 | Control register 1            |
| BTIMx_CR2     | BTIMx_BASE + 0x04 | Control register 2            |
| BTIMx_SMCR    | BTIMx_BASE + 0x08 | Slave mode control register   |
| BTIMx_DIER    | BTIMx_BASE + 0x0C | DMA/Interrupt enable register |
| BTIMx_ISR     | BTIMx_BASE + 0x10 | Interrupt flag register       |
| BTIMx_ICR     | BTIMx_BASE + 0x18 | Interrupt flag clear register |
| BTIMx_EGR     | BTIMx_BASE + 0x14 | Event generation register     |
| BTIMx_CNT     | BTIMx_BASE + 0x24 | Count register                |
| BTIMx_PSC     | BTIMx_BASE + 0x28 | Prescaler register            |
| BTIMx_ARR     | BTIMx_BASE + 0x2C | Auto-reload register          |



## 14.10 Register description

See section 1 [Documentation conventions](#) for abbreviations used in register descriptions.

### 14.10.1 BTIMx\_CR1 Control register 1

Address offset: 0x00      Reset value: 0x0000 0000

| Bit field | Name     | Permission | Function description                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|-----------|----------|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:16     | RFU      | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 15        | TOGEN    | RW         | TOG pin output waveform enable control<br>0: TOGP and TOGN both output low level.<br>1: TOGP and TOGN output opposite level.                                                                                                                                                                                                                                                                                                                                                              |
| 14:12     | RFU      | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 11        | UIFREMAP | RW         | UIF status bit remapping enable control<br>0: Remapping is disabled and BTIMx_CNT[31] remains 0.<br>1: Remapping is enabled. BTIMx_CNT[31] is equivalent to BTIMx_ISR.UIF.                                                                                                                                                                                                                                                                                                                |
| 10:4      | RFU      | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 3         | ONESHOT  | RW         | Single/Continuous count mode control<br>0: Continuous count mode.<br>1: Single count mode.                                                                                                                                                                                                                                                                                                                                                                                                |
| 2         | URS      | RW         | Update the request source configuration<br>0: When UEV is enabled, all of the following events generate update interrupt or DMA request: <ul style="list-style-type: none"> <li>Counter overflow</li> <li>Set the UG bit field to 1</li> <li>Update events generated via the slave mode controller</li> </ul> 1: When UEV is enabled, only overflows on the counter generate update interrupt or DMA request.                                                                             |
| 1         | UDIS     | RW         | Update disable control<br>0: Enable UEV. Update events can be generated by the following events: <ul style="list-style-type: none"> <li>Overflow on the counter.</li> <li>Setting the UG bit field to 1.</li> <li>Update events generated via the slave mode controller.</li> </ul> 1: Disable UEV. No update event is generated. However, if the UG bit field is set to 1 or a hardware reset is received from the slave mode controller, the counters and prescalers are reinitialised. |
| 0         | EN       | RW         | Timer operation control<br>0: Timer stopped.<br>1: Timer running.                                                                                                                                                                                                                                                                                                                                                                                                                         |



### 14.10.2 BTIMx\_CR2 Control register 2

Address offset: 0x04      Reset value: 0x0000 0000

| Bit field | Name | Permission | Function description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|-----------|------|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:7      | RFU  | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 6:4       | MMS  | RW         | <p>Master mode output signal (TRGO) configuration</p> <p>000: Reset signal, EGR.UG is used as trigger output (TRGO). If the reset is generated by the trigger input, the signal on TRGO will be delayed compared to the actual reset.</p> <p>001: Enable signal, CR1.EN used as trigger output (TRGO). The trigger output can be used to start multiple timers at the same time, or to control enabling the slave timer over a period of time. The counter enable signal is a combination of the EN control bit and the logic and operation of the trigger input in gated mode. There is a delay in TRGO when the counter enable signal is controlled by a trigger input, except when master/slave mode is selected (see the description for SMCR.MSM).</p> <p>010: Update signal, select update event as trigger output (TRGO). For example, the master timer can be used as a prescaler for the slave timer.</p> <p>011: Overflow signal, output a pulse whenever BTIM overflows.</p> |
| 3:0       | RFU  | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |

### 14.10.3 BTIMx\_SMCR Slave mode control register

Address offset: 0x08      Reset value: 0x0000 0000

| Bit field | Name    | Permission | Function description                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|-----------|---------|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:18     | RFU     | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 17        | TRGIPOL | RW         | Trigger input signal (TRGI) polarity selection<br>0: Positive phase, high level or rising edge is valid<br>1: Reversed phase, low level or falling edge is valid                                                                                                                                                                                                                                                                                                                          |
| 16        | RSTIPOL | RW         | Count reset signal (RSTI) polarity selection<br>0: Positive phase, rising edge is valid<br>1: Reversed phase, falling edge is valid                                                                                                                                                                                                                                                                                                                                                       |
| 15        | RFU     | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 14:12     | TRGIFLT | RW         | TRGI signal digital filter circuit sampling clock and sampling points configuration<br>000: No filter<br>001: $F_{\text{sample}} = \text{PCLK}$ , $N = 2$<br>010: $F_{\text{sample}} = \text{PCLK}$ , $N = 4$<br>011: $F_{\text{sample}} = \text{PCLK}$ , $N = 6$<br>100: $F_{\text{sample}} = \text{PCLK}/4$ , $N = 4$<br>101: $F_{\text{sample}} = \text{PCLK}/4$ , $N = 6$<br>110: $F_{\text{sample}} = \text{PCLK}/8$ , $N = 4$<br>111: $F_{\text{sample}} = \text{PCLK}/8$ , $N = 6$ |



| Bit field | Name    | Permission | Function description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|-----------|---------|------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 11        | MSM     | RW         | Master/slave mode configuration<br>0: Nothing is done.<br>1: The trigger input event (TRGI) action of the current timer is delayed to perfectly synchronize the current timer with its slave timer (via TRGO).                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 10:7      | TRGISRC | RW         | Trigger input signal (TRGI) source configuration<br>0000: No trigger signal<br>0001: LVD_OUT signal<br>0010: VCx_OUT signal<br>0011: See footnote below table <sup>[1]</sup><br>0100: I2Cx_SCL signal <sup>[2]</sup><br>0101: SPIx_NCS signal<br>0110: UARTx_RXD signal<br>0111: ATIM_TRGO<br>1000: BTIM1_Trgo<br>1001: BTIM2_Trgo<br>1010: BTIM3_Trgo<br>1011: GTIM1_Trgo<br>1100: GTIM2_Trgo<br>1101: GTIM3_Trgo<br>1110: GTIM4_Trgo<br>1111: BTIMx_ETR signal<br><i>Attention: Do not select the Trgo output by BTIMx itself.</i>                                                                                                   |
| 6:3       | RSTISRC | RW         | Count reset signal (RSTI) source configuration<br>0000: No reset signal<br>0001: LVD_OUT signal<br>0010: VCx_OUT signal<br>0011: See footnote below table <sup>[3]</sup><br>0100: I2Cx_TRGO signal <sup>[4]</sup><br>0101: SPIx_NCS signal<br>0110: UARTx_RXD signal<br>0111: ATIM_TRGO<br>1000: BTIM1_Trgo<br>1001: BTIM2_Trgo<br>1010: BTIM3_Trgo<br>1011: GTIM1_Trgo<br>1100: GTIM2_Trgo<br>1101: GTIM3_Trgo<br>1110: GTIM4_Trgo<br>1111: BTIMx_ETR signal<br><i>Note: Reset the counter at the valid edge of the RSTI signal and generate an update event.</i><br><i>Attention: Do not select the Trgo output by BTIMx itself.</i> |
| 2:0       | SMS     | RW         | Slave mode function configuration<br>000: Disable slave mode.<br>001: Trigger start mode: enable counter on rising edge of TRGI.<br>010: Gated count mode: count PCLK at TRGI high level.<br>011: External counting mode: counter incremented by 1 on rising edge of TRGI.                                                                                                                                                                                                                                                                                                                                                             |

[1] When set to 0011, the BTIM1 trigger input signal source is RTC\_OUT; BTIM2 trigger input signal source is LPTIM\_OV; BTIM3 trigger input signal source is VC4\_OUT.

[2] When set to 0100, the BTIM1 trigger input signal source is I2C1\_SCL; BTIM2 trigger input signal source is I2C2\_SCL; BTIM3 trigger input signal source is I2C1\_SCL.

[3] When set to 0011, the BTIM1 reset signal source is RTC\_OUT; BTIM2 reset signal source is LPTIM\_OV; BTIM3 reset signal source is VC4\_OUT.

[4] When set to 0100, the BTIM1 reset signal source is I2C1\_TRGO; BTIM2 reset signal source is I2C2\_TRGO; BTIM3 reset signal source is I2C1\_TRGO.



#### 14.10.4 BTIMx\_DIER DMA/Interrupt enable register

Address offset: 0x0C      Reset value: 0x0000 0000

| Bit field | Name | Permission | Function description                                            |
|-----------|------|------------|-----------------------------------------------------------------|
| 31:15     | RFU  | -          | Reserved bits, please keep the default value.                   |
| 14        | TDE  | RW         | Trigger DMA request enable control<br>0: Disabled<br>1: Enabled |
| 13:9      | RFU  | -          | Reserved bits, please keep the default value.                   |
| 8         | UDE  | RW         | Update DMA request enable control<br>0: Disabled<br>1: Enabled  |
| 7         | RFU  | -          | Reserved bits, please keep the default value.                   |
| 6         | TIE  | RW         | Trigger interrupt enable control<br>0: Disabled<br>1: Enabled   |
| 5:1       | RFU  | -          | Reserved bits, please keep the default value.                   |
| 0         | UIE  | RW         | Update interrupt enable control<br>0: Disabled<br>1: Enabled    |

#### 14.10.5 BTIMx\_ISR Interrupt flag register

Address offset: 0x10      Reset value: 0x0000 0000

| Bit field | Name | Permission | Function description                                                                          |
|-----------|------|------------|-----------------------------------------------------------------------------------------------|
| 31:7      | RFU  | -          | Reserved bits, please keep the default value.                                                 |
| 6         | TIF  | RO         | Trigger interrupt flag<br>0: No trigger event has occurred.<br>1: Trigger event has occurred. |
| 5:1       | RFU  | -          | Reserved bits, please keep the default value.                                                 |
| 0         | UIF  | RO         | Update interrupt flag<br>0: No update event has occurred.<br>1: An update event has occurred. |



### 14.10.6 BTIMx\_ICR Interrupt flag clear register

Address offset: 0x18      Reset value: 0x0000 0041

| Bit field | Name | Permission | Function description                                                      |
|-----------|------|------------|---------------------------------------------------------------------------|
| 31:7      | RFU  | -          | Reserved bits, please keep the default value.                             |
| 6         | TIF  | R1W0       | Trigger flag clear<br>W0: Clear trigger flag.<br>W1: No function.         |
| 5:1       | RFU  | -          | Reserved bits, please keep the default value.                             |
| 0         | UIF  | R1W0       | Update flag clear<br>W0: Counter update flag cleared.<br>W1: No function. |

### 14.10.7 BTIMx\_EGR Event generation register

Address offset: 0x14      Reset value: 0x0000 0000

| Bit field | Name | Permission | Function description                                                                                                                                                                                                                                                                                         |
|-----------|------|------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:7      | RFU  | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                |
| 6         | TG   | WO         | This bit is set to 1 by software to generate an event and is automatically cleared by hardware<br>0: No function.<br>1: Software generated trigger event.                                                                                                                                                    |
| 5:1       | RFU  | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                |
| 0         | UG   | WO         | This bit can be set to 1 by software to generate an event and automatically cleared by hardware<br>0: No function.<br>1: Re-initialises the counter and generates a register update event.<br>Note that the counter for the prescaler will also be cleared to zero, but the prescaler ratio is not affected. |



### 14.10.8 BTIMx\_CNT Count register

Address offset: 0x24      Reset value: 0x0000 0000

| Bit field | Name   | Permission | Function description                                                                                                                                                                                                                                                                                                                           |
|-----------|--------|------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31        | UIFCPY | RO         | According to the value of the UIFREMAP bit field in the BTIMx_CR1 register, this bit field indicates a different meaning: <ul style="list-style-type: none"> <li>UIFREMAP = 0, this bit field is reserved and read as 0.</li> <li>UIFREMAP = 1, this bit field indicates a read-only copy of the UIF bit of the BTIMx_ISR register.</li> </ul> |
| 30:16     | RFU    | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                                  |
| 15:0      | CNT    | RW         | Timer value                                                                                                                                                                                                                                                                                                                                    |

### 14.10.9 BTIMx\_PSC Prescaler register

Address offset: 0x28      Reset value: 0x0000 0000

| Bit field | Name | Permission | Function description                                                                               |
|-----------|------|------------|----------------------------------------------------------------------------------------------------|
| 31:16     | RFU  | -          | Reserved bits, please keep the default value.                                                      |
| 15:0      | PSC  | RW         | Value of the prescaler<br>Counter clock frequency CK_CNT is equal to $f_{CK\_PSC}/(PSC[15:0]+1)$ . |

### 14.10.10 BTIMx\_ARR Auto reload register

Address offset: 0x2C      Reset value: 0x0000 FFFF

| Bit field | Name | Permission | Function description                          |
|-----------|------|------------|-----------------------------------------------|
| 31:16     | RFU  | -          | Reserved bits, please keep the default value. |
| 15:0      | ARR  | RW         | Timer reloads value.                          |



## 15 Low power timer (LPTIM)

### 15.1 Overview

The CW32L012 integrates a 16-bit low-power timer (LPTIM) internally, which can realize the function of timing or counting external pulses with very low power consumption. By selecting an appropriate clock source and trigger signal, the function of waking up the system when it is in low-power sleep mode can be realized. There is a compare register inside the LPTIM, which can realize the compare output and PWM output, and can control the polarity of the output waveform. In addition, LPTIM can also be connected with a quadrature encoder to automatically count up and down.



## 15.2 Main features

- 16bit auto-reload counter.
- Programmable prescaler supports 1, 2, 4, 8, 16, 32, 64, 128 division.
- Working clock optional:
  - Internal clock: PCLK, LSI, LSE.
  - External clock: LPTIM\_CH1.
- Support single mode and continuous mode.
- Rich software and hardware trigger sources.
- Clockless external pulse counting.
- Input and output polarity selection.
- Configurable input port filters.
- Output configurable as single pulse, single set or PWM wave.
- Quadrature code count function.

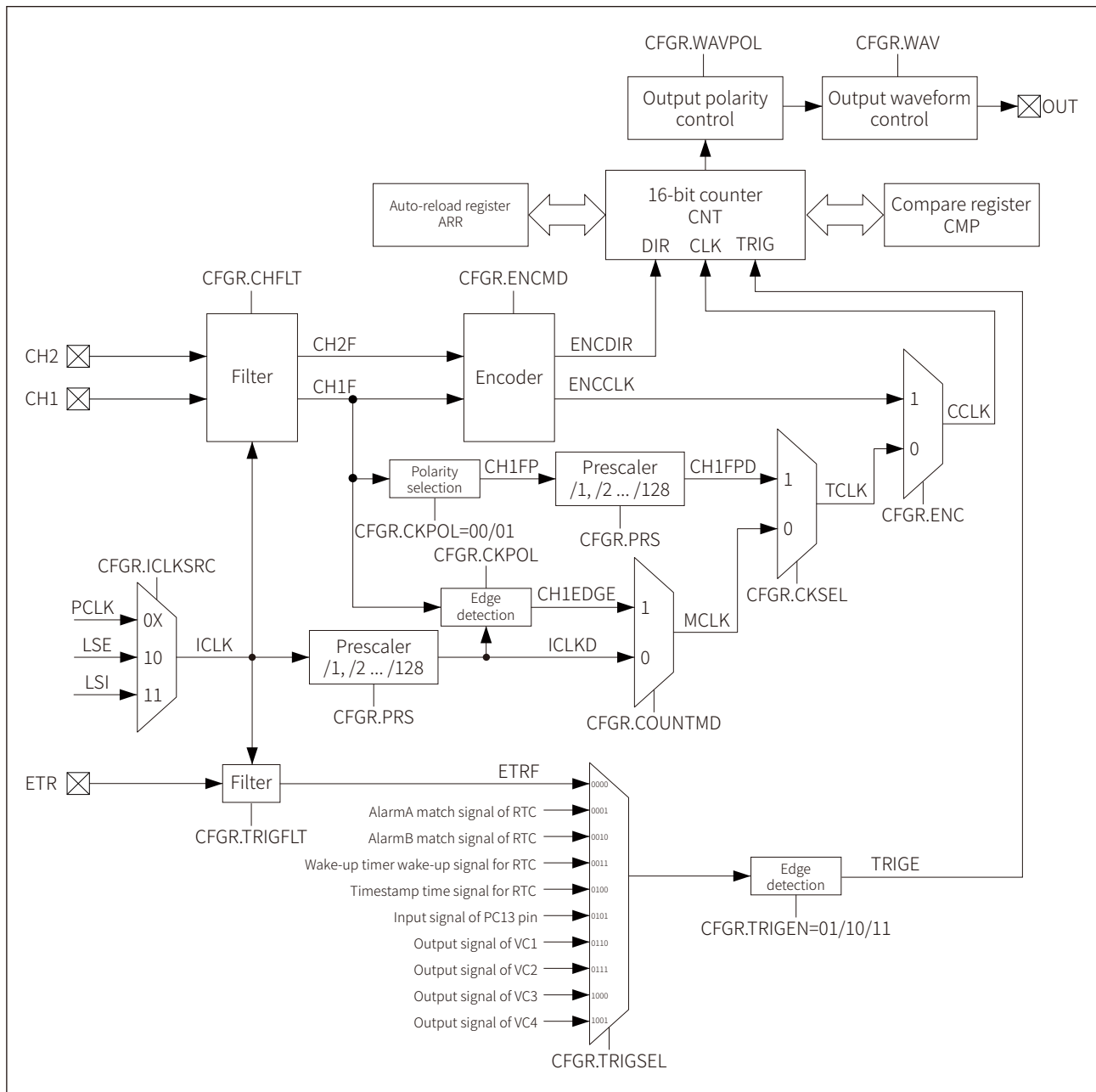


## 15.3 Functional description

### 15.3.1 Functional block diagram

The functional block diagram of LPTIM is shown in the following figure:

Figure 15-1 LPTIM functional block diagram



### 15.3.1.1 Clock source

The counting clock source of LPTIM can be selected from PCLK, LSE, LSI, or an external clock input on the LPTIM\_CH1 pin, which is configured through the ICLKSRC, COUNTMD and CKSEL bit fields of the configuration register LPTIM\_CFGR.

When the external clock input on the LPTIM\_CH1 pin is selected as the counting clock source, the LPTIM can work in the following 2 configuration modes:

1. The first configuration mode (CKSEL = 0, COUNTMD = 1): The external clock input by CH1 is used as the count clock of LPTIM, and the internal clock module also needs to provide clock to LPTIM for filtering and edge detection.
2. The second configuration mode (CKSEL = 1): The external clock input by CH1 is used as the only clock of the LPTIM. This configuration can be used to implement the timeout function or pulse counter function when all the internal oscillators are turned off in the low power mode.

When configured to use an external clock source, the CKPOL bit field is used to set the valid edge of the external clock signal. If configured to be active on both upper and lower edges, the internal clock signal should also be provided (first configuration). In this case, the frequency of the internal clock signal should be at least four times the frequency of the external clock signal.

### 15.3.1.2 Prescaler

The prescaler divides the frequency of the counter clock source by a multiple of 2, and the maximum frequency division factor is 128. The prescaler coefficient is configured by the PRS bit field of the configuration register LPTIM\_CFGR. The frequency division coefficient configuration is shown in the following table:

Table 15-1 Frequency division coefficient configuration table

| PRS | Frequency division factor |
|-----|---------------------------|
| 000 | 1                         |
| 001 | 2                         |
| 010 | 4                         |
| 011 | 8                         |
| 100 | 16                        |
| 101 | 32                        |
| 110 | 64                        |
| 111 | 128                       |



### 15.3.1.3 Filter unit

The LPTIM inputs, whether external (mapped to GPIO) or internal (mapped on-chip to other on-chip peripherals), can be filtered through a digital filter to prevent glitches and noise perturbations from causing false LPTIM counts or triggering.

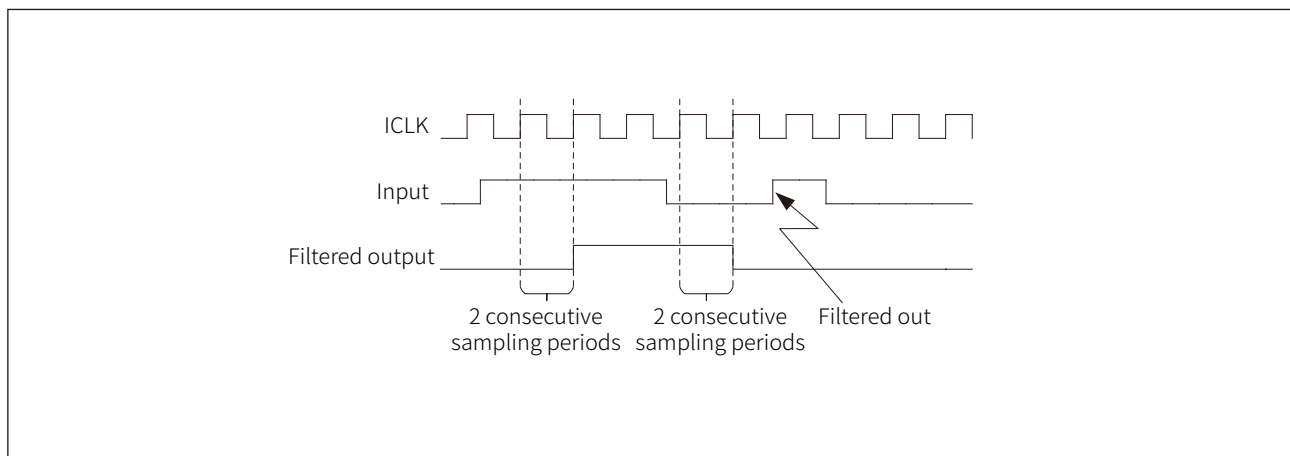
In order for the filter to work properly, an internal clock source should first be provided to the LPTIM. This is a necessary condition to ensure the normal operation of the filter.

Digital filters can be divided into 2 groups:

1. The first group of digital filters is used for the external input of LPTIM (CH1, CH2), and the sensitivity of the digital filters is controlled by the CHFLT bit.
2. The first group of digital filters is used for the external input of LPTIM (CH1, CH2), and the sensitivity of the digital filters is controlled by the CHFLT bit.

The sensitivity of the filter can be changed by changing the number of active level holds that the filter continuously samples the input signal. For example, when the filter is set to an active level for 2 sampling periods, the filtered output waveform is shown in the following figure:

Figure 15-2 Filter timing diagram



*Attention: If no internal clock signal is provided, the digital filter must be disabled by setting the CHFLT and TRGFLT bit fields to '00'. In this case, an external analog filter can be used to shield the LPTIM external input from disturbances.*



### 15.3.2 Start method

The EN bit field of the control register LPTIM\_CR is used to enable/disable the LPTIM core. After setting the EN bit to 1, a delay of two count clock cycles is required before the LPTIM actually starts counting.

The LPTIM counter can be started by software or by a trigger event, set by the TRIGEN bit field of the configuration register LPTIM\_CFGR.

When setting LPTIM\_CFGR.TRIGEN to 00, it is configured for software startup. After starting LPTIM, setting the CNTSTART or SNGSTART bit field to 1 will start the LPTIM counter to count.

When LPTIM\_CFGR.TRIGEN is set to 01/10/11, it is configured to trigger the event to start. After starting LPTIM and configuring the operating mode (single or continuous mode), once a valid edge of the trigger input is detected, the LPTIM counter is started to count. For specific trigger input signals, please refer to section [15.3.3 Trigger source](#).

*Attention: The LPTIM\_CFGR and LPTIM\_IER registers can only be modified when the EN bit field is 0 and LPTIM is disabled.*

*Attention: LPTIM must be enabled (EN = 1) before setting the CNTSTART or SNGSTART bit fields. When LPTIM is disabled, any writes to these bits will be discarded by hardware.*

*Attention: When configured for software startup, after setting the CNTSTART or SNGSTART bit field, the counter does not actually start until after a delay of 3 core clock cycles.*

*Attention: Since the trigger signal and the timing clock of the LPTIM are not synchronized, after the trigger detects the trigger signal, the timer needs to delay two timing clock cycles before it starts running.*



### 15.3.3 Trigger source

The LPTIM counter can be started by software or after a valid edge of the trigger input is detected. LPTIM supports 8 trigger sources, which are set by the TRIGSEL bit field of the configuration register LPTIM\_CFGR, as shown in the following table:

Table 15-2 LPTIM trigger input source table

| TRIGSEL bit field | Trigger source                       |
|-------------------|--------------------------------------|
| 0000              | LPTIM_ETR pin signal                 |
| 0001              | ALARMA match signal for RTC          |
| 0010              | ALARMB match signal for RTC          |
| 0011              | Wake-up timer wake-up signal for RTC |
| 0100              | Timestamp event signal for RTC       |
| 0101              | PC13 pin input signal                |
| 0110              | VC1 output signal                    |
| 0111              | VC2 output signal                    |
| 1000              | VC3 output signal                    |
| 1001              | VC4 output signal                    |

The TRIGEN bit field is used to set the trigger enable and polarity, as shown in the following table:

Table 15-3 LPTIM trigger enable and polarity

| TRIGEN | Trigger mode and polarity                                                      |
|--------|--------------------------------------------------------------------------------|
| 00     | Software start, software writes CNTSTART or SNGSTART to start the counter.     |
| 01     | Hardware trigger, start the counter on the rising edge of the trigger signal.  |
| 10     | Hardware trigger, start the counter on the falling edge of the trigger signal. |
| 11     | Start the counter on the rising or falling edge of the trigger signal.         |

*Note: If a new trigger event occurs when the timer has already started, it will be ignored (unless the timeout function is enabled).*



### 15.3.4 Operating mode

The LPTIM supports 2 modes of operation:

1. Single mode: the counter is started by software or triggered by an event, and stops when the count value reaches the reload value ARR.
2. Continuous mode: The counter is started by software or by a trigger event, the counter counts continuously and stops until the timer is disabled.

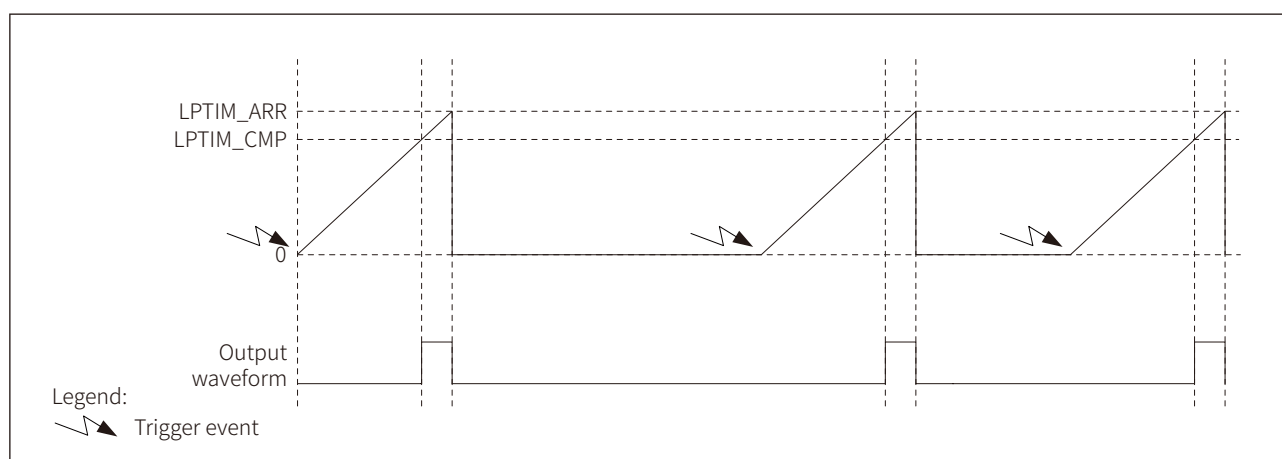
#### Single mode

In the single-counting mode, the counter can be started by software or triggered by an event, and the counter CNT is driven by the TCLK clock to count up. When the count value is equal to the reload value ARR, the automatic reload match flag LPTIM\_ISR.ARRM is set by hardware, and the counter stops counting at the same time.

If it is set to software start, after setting LPTIM\_CR.EN to 1, set LPTIM\_CR.SNGSTART to 1 to start the counter to count.

If set to trigger event start, set LPTIM\_CR.EN to 1 and set LPTIM\_CR.SNGSTART to 1 to make LPTIM run in single shot mode. After that, a trigger event will start the counter to count. Any trigger events after the counter starts and before the counter reaches the ARR will be discarded. After the counter is stopped, a new trigger event will start the counter again and start counting. As shown below:

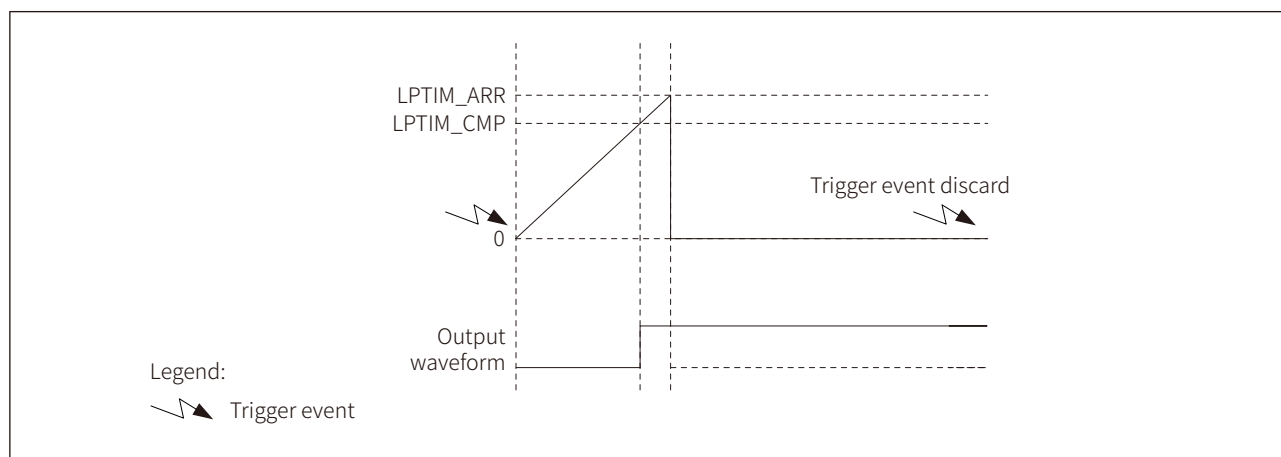
Figure 15-3 LPTIM single mode



#### Single set output

It should be noted that when the WAVE bit field of the configuration register LPTIM\_CFGR is set to 1, the single set mode will be activated. In one-shot mode, the counter is only started once on the first trigger event, and any subsequent trigger events will be discarded, as shown in the following figure:

Figure 15-4 LPTIM single set waveform



If it is set to software start (TRIGEN = '00'), after SNGSTART is set to 1, it will start the counter for one count.

### Continuous mode

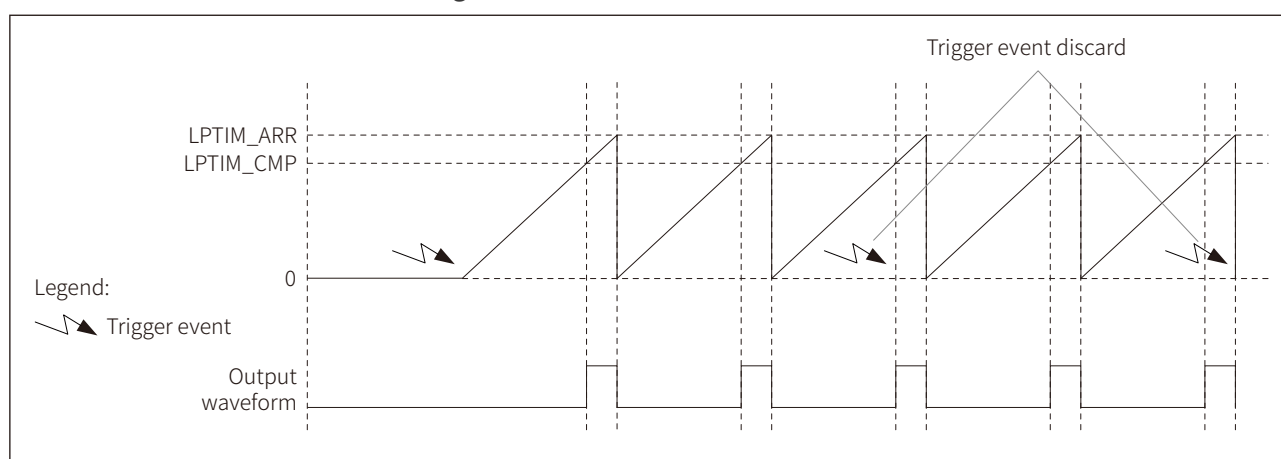
In continuous counting mode, the counter can be started by software or triggered by an event, and the counter CNT counts up under the drive of the TCLK clock. When the count value is equal to the reload value ARR, the automatic reload match flag LPTIM\_ISR.ARRM is set by hardware, the counter overflows and starts to count up in the next cycle, and it stops counting until LPTIM\_CR.EN is set to 0.

If it is set to software start, after setting LPTIM\_CR.EN to 1, set LPTIM\_CR.CNTSTART to 1 to start the counter for continuous counting.

If set to trigger event start, set LPTIM\_CR.EN to 1 and set LPTIM\_CR.CNTSTART to 1 to make LPTIM run in continuous mode. Thereafter, a trigger event will start the counter to count continuously and any subsequent trigger events will be discarded.

The figure below shows an example of count and output in continuous mode:

Figure 15-5 LPTIM continuous mode



Single mode and continuous mode can be switched between each other during operation:

- If previously configured for continuous mode, setting the SNGSTART bit field will cause the LPTIM to switch to single mode. The counter (if activated) will stop as soon as the reload value ARR is reached.
- If previously configured for single mode, setting the CNTSTART bit field will cause the LPTIM to switch to continuous mode. The counter (if activated) will restart as soon as the reload value ARR is reached.

### 15.3.5 Timeout function

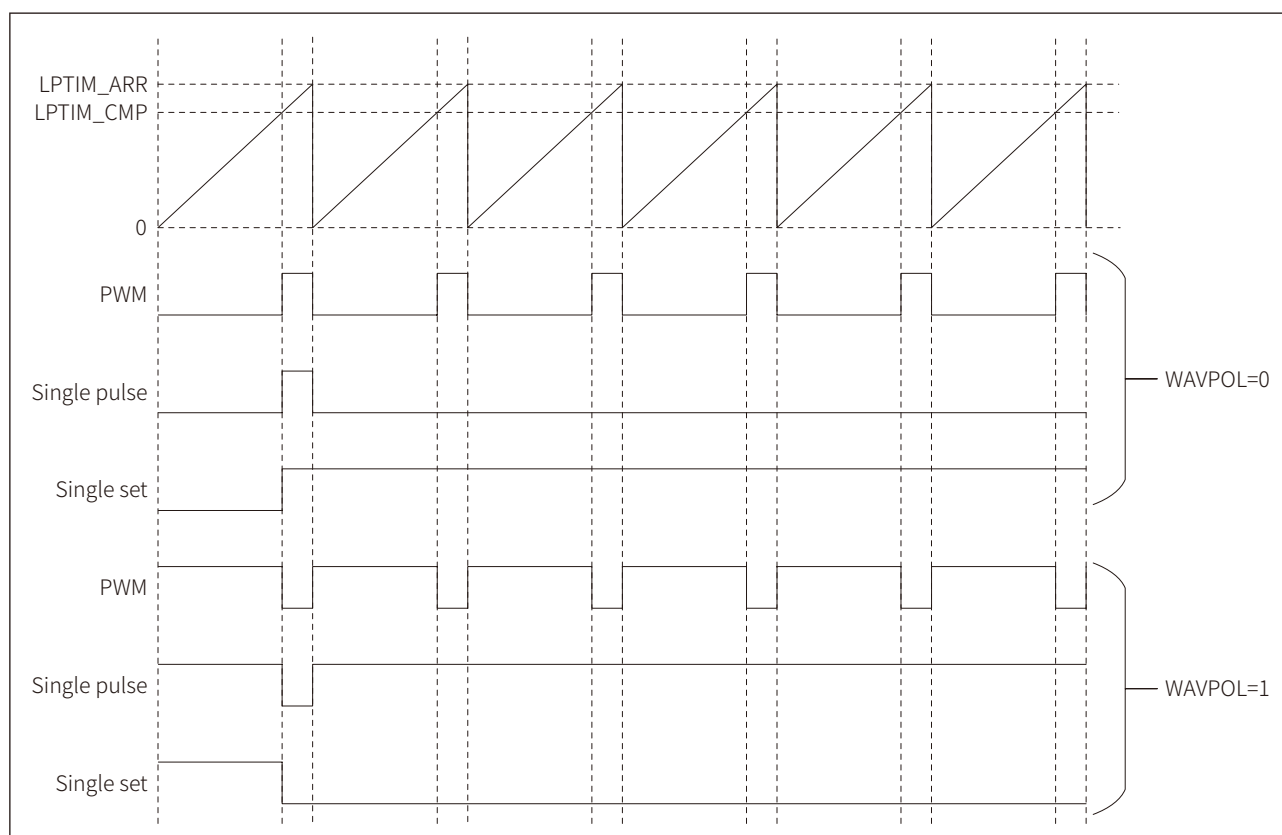
If the TIMOUT bit field is set to 1, when LPTIM is counting, the counter can be reset by a trigger signal, and the counter will start to count again from 0.

This feature can implement a low-power timeout wake-up function, and the timeout threshold is determined by the compare register LPTIM\_CMP. When the trigger signal is not generated within the expected time range, that is, after the trigger signal times out, the counter counts to the comparison value CMP, and LPTIM will generate a compare match interrupt to wake up the MCU.

### 15.3.6 Waveform output control

The LPTIM can output PWM wave in continuous mode, single pulse wave or single set waveform can be output in single mode, and the polarity of output waveform can be controlled by WAVPOL bit. When the WAVPOL bit is 0, it is a forward output; when the WAVPOL bit is 1, it is a reverse output. Change the WAVPOL bit field and the polarity change of the output waveform will take effect immediately. When LPTIM is not enabled, modifying the value of WAVPOL will also change the default level polarity of the output. The figure below shows the 3 waveforms that can be generated on the LPTIM output, in addition, it shows the effect of the WAVPOL bit on the polarity change.

Figure 15-6 Three waveforms of LPTIM output



### 15.3.7 Register update

The PRELOAD bit field of the configuration register LPTIM\_CFGR is used to set the data loading time of the auto-reload register LPTIM\_ARR and the compare register LPTIM\_CMP. When PRELOAD is set to 0, the LPTIM\_ARR and LPTIM\_CMP registers will be updated immediately after the write operation; when PRELOAD is set to 1, the value of the write operation is loaded when the count value reaches the previous ARR value (if the timer has already started).

Since the APB interface of LPTIM and the core of LPTIM use different clocks, when writing the LPTIM\_ARR register and the LPTIM\_CMP register, there is some delay between writing the value to the register and updating the value of the register to the ARR and CMP of the LPTIM. During this delay period, any additional writes to these registers must be avoided. Whether the ARR and CMP are updated can be judged by the ARROK flag and the CMPOK flag in the LPTIM\_ISR register.

The ARROK flag and the CMPOK flag in the LPTIM\_ISR register are 0, indicating that the value of the corresponding register is being updated. Any writing to the LPTIM\_ARR register and the LPTIM\_CMP register at this time will lead to unpredictable results. A new write operation to the same register must wait until the ARROK flag and the CMPOK flag in the LPTIM\_ISR register are 1, that is, when the register has been updated.

### 15.3.8 Count function

The LPTIM can count the signal input on the external LPTIM\_CH1 pin, and can also count the internal clock signal. The count clock source is configured through the CKSEL and COUNTMD bit fields of the configuration register LPTIM\_CFGR.

- When CKSEL is 0, LPTIM operation requires an internal clock ICLK.
  - If COUNTMD is 0, LPTIM counts the pulses of the internal clock source.
  - If COUNTMD is 1, the LPTIM samples the input signal on the LPTIM\_CH1 pin with the internal clock. Therefore, in order not to miss any events, the CH1 signal should not change more frequently than the internal clock supplied to the LPTIM. Also, the internal clock supplied to LPTIM cannot be prescaled (PRS = 000). By setting the value of the CKPOL bit field, you can choose to count on the rising edge, falling edge or upper and lower edges of the CH1 signal.
- When CKSEL is 1, LPTIM operation is provided by an external clock source.
  - In this mode, the value of the COUNTMD bit field has no effect on LPTIM, which does not require an internal clock source (unless the filter is enabled). The external signal input on CH1 of LPTIM is used as the count of LPTIM. For this configuration, the LPTIM counter can count on the rising or falling edge of the clock signal on CH1, but cannot be updated on both the rising and falling edges. Since the signal input to LPTIM's CH1 is also used to clock the LPTIM core logic, after starting LPTIM, the first 3 count clock edges are used for the configuration of the core, and the counter will not start counting until the 4th count clock edge (see [15.5.1 External pulse counting routine](#)).



### 15.3.9 Encoding mode

The LPTIM can process signals from quadrature encoders. The quadrature encoder indicates the position and rotation direction of the rotating element by outputting two pulse signals with a phase difference of 90 degrees. LPTIM in encoder mode is equivalent to connecting an external clock with direction selection, which means that the counter only counts continuously between 0 and the auto-reload value ARR (counting up from 0 to ARR depending on the direction, or counts down from ARR to 0). Therefore, the auto-reload register LPTIM\_ARR must be configured before startup.

The 2 output signals of the encoder generate clock signals through the two external input ports CH1 and CH2 of the LPTIM as the LPTIM counter clock. The phase between these two signals determines the count direction. In encoding mode, the LPTIM must be clocked by the internal clock source, and the frequency of the signals input to CH1 and CH2 must not exceed the 4-frequency division of the LPTIM internal clock frequency, otherwise the LPTIM will not work properly.

Direction changes are indicated by the UP and DOWN flag bits in the LPTIM\_ISR register. In addition, if the corresponding enable bit in the LPTIM\_IER register is set, an interrupt will be triggered when both direction change events occur.

Before using encoder mode, the LPTIM must first be configured for continuous mode, and then set the ENC bit to 1 (**Note that at this time, the COUNTMD bit must be set to 0.**) to enable encoder mode. After enabling, the LPTIM counter will automatically modify the count value of LPTIM\_CNT according to the speed and direction of the encoder. Therefore, the count value of LPTIM\_CNT represents the position of the encoder. The counting direction is indicated by the UP and DOWN signs, corresponding to the direction of rotation of the connected sensor.

LPTIM supports 3 encoding counting modes:

1. Set LPTIM\_CFGR.ENCMD to 0x00, the counter will only count on the edge of CH1.
2. Set LPTIM\_CFGR.ENCMD to 0x01, the counter will only count on the edge of CH2.
3. Set LPTIM\_CFGR.ENCMD to 0x02, the counter will count on the edge of both CH1 and CH2.

The relationship between counting direction and encoder signal is shown in the following table:

Table 15-4 Quadrature encoding mode count and direction

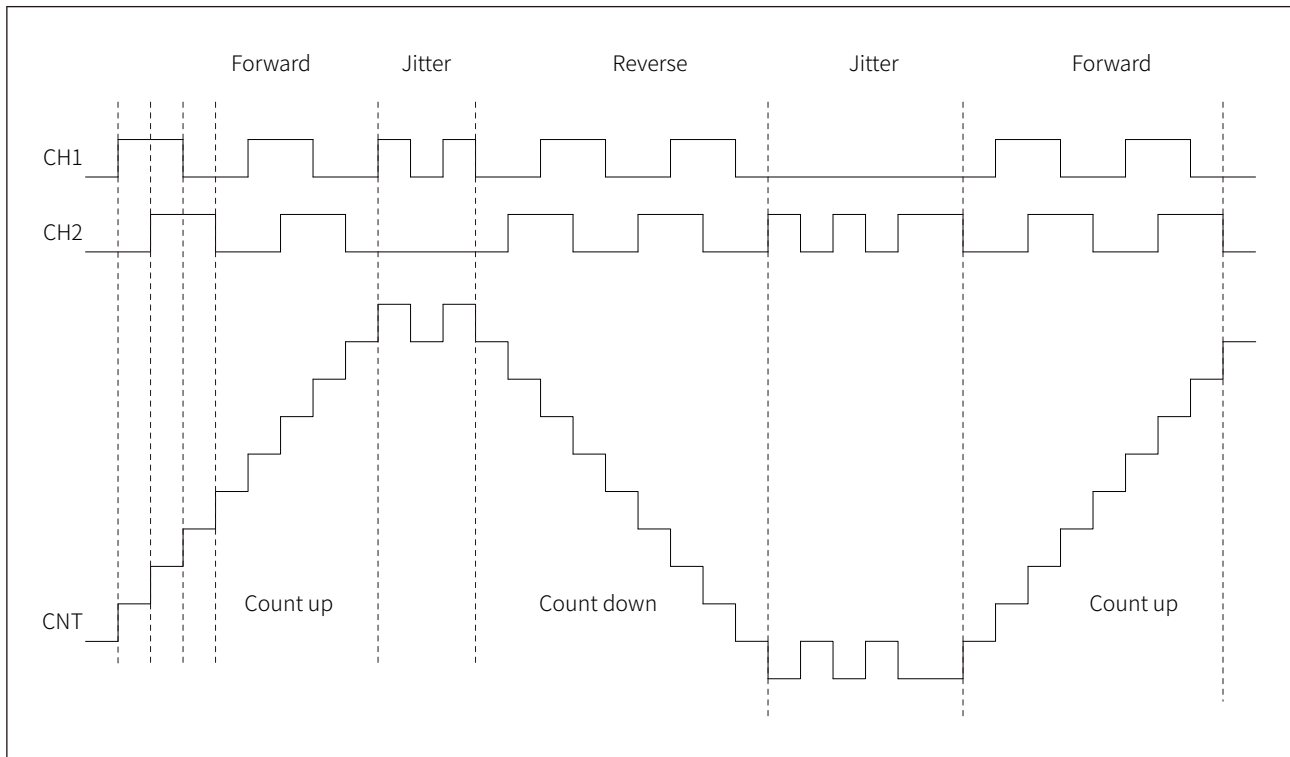
| Mode   | Signal level |      | CH1            |                | CH2            |                |
|--------|--------------|------|----------------|----------------|----------------|----------------|
|        | CH1          | CH2  | Rise           | Decline        | Rise           | Decline        |
| Mode 1 | -            | High | Count down     | Count up       | Does not count | Does not count |
|        | -            | Low  | Count up       | Count down     | Does not count | Does not count |
| Mode 2 | High         | -    | Does not count | Does not count | Count up       | Count down     |
|        | Low          | -    | Does not count | Does not count | Count down     | Count up       |
| Mode 3 | High         | High | Count down     | Count up       | Count up       | Count down     |
|        | Low          | Low  | Count up       | Count down     | Count down     | Count up       |

**Attention:** In this mode, the LPTIM must be clocked by an internal clock source, so the CKSEL bits must remain at their reset value of 0. In addition, the prescaler division ratio must be equal to its reset value of 1 (PRS bit field must be "000").

The figure below is an example of encoder mode counting, showing the generation and direction control of the count signal:



Figure 15-7 Encoder mode count example





## 15.4 Debug support

The LPTIM supports stop or continue counting in debug mode, which is set by the LPTIM bit field of the debug status timer control register SYSCTRL\_DEBUG.

- Set SYSCTRL\_DEBUG.LPTIM to 1 to suspend the counter count of LPTIM in debug state.
- Set SYSCTRL\_DEBUG.LPTIM to 0, the counter of LPTIM continues to count in debug state.



## 15.5 Programming examples

### 15.5.1 External pulse counting routine

- Step 1:** Set `SYSCTRL_APBEN2.LPTIM` to 1 to enable the configuration clock of LPTIM.
- Step 2:** Configure the GPIO corresponding to the `LPTIM_CH1` pin to the multiplexed input mode. For the specific register configuration, please refer to chapter [9 General-purpose input/output \(GPIO\)](#).
- Step 3:** Set `LPTIM_CFGR.COUNTMD`, `LPTIM_CFGR.CKSEL` to 1, configure LPTIM counting mode, the counting clock is provided by CH1.
- Step 4:** Set `LPTIM_IER.ARRM` to 1 to enable ARR auto-reload matching interrupt.
- Step 5:** Set `LPTIM_ARR` to 6, the purpose is to trigger the ARR automatic reload matching interrupt after obtaining 10 external count pulses, because the first 5 pulses, CNT does not count.
- Step 6:** Set `LPTIM_CR.EN` to 1 to enable LPTIM.



## 15.6 List of registers

LPTIM base address: LPTIM\_BASE = 0x4000 6000

Table 15-5 List of LPTIM registers

| Register name | Register address   | Register description          |
|---------------|--------------------|-------------------------------|
| LPTIM_ISR     | LPTIM_BASE + 0x00  | Interrupt and status register |
| LPTIM_ICR     | LPTIM_BASE + 0x04  | Interrupt clear register      |
| LPTIM_IER     | LPTIM_BASE + 0x08  | Interrupt enable register     |
| LPTIM_CFGR    | LPTIM_BASE + 0x0C  | Configuration register        |
| LPTIM_CR0     | LPTIM_BASE + 0x10  | Control register 0            |
| LPTIM_CMP     | LPTIM_BASE + 0x14  | Compare register              |
| LPTIM_ARR     | LPTIM_BASE + 0x18  | Auto-reload register          |
| LPTIM_CNT     | LPTIM_BASE + 0x1C  | Counter register              |
| LPTIM_CR1     | 0x4000 6000 + 0x20 | Control register 1            |



## 15.7 Register description

See section 1 [Documentation conventions](#) for abbreviations used in register descriptions.

### 15.7.1 LPTIM\_CFGR Configuration register

Address offset: 0x0C      Reset value: 0x0000 0000

| Bit field | Name    | Permission | Function description                                                                                                                                                                                                                      |
|-----------|---------|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:27     | RFU     | -          | Reserved bits, please keep the default value.                                                                                                                                                                                             |
| 26:25     | ICLKSRC | RW         | ICLK source configuration<br>00: PCLK<br>01: PCLK<br>10: LSE<br>11: LSI                                                                                                                                                                   |
| 24        | ENC     | RW         | Encoder mode enable<br>0: Disabled<br>1: Enabled<br><i>Attention: Encoding circuits require ICLK to function properly.</i>                                                                                                                |
| 23        | COUNTMD | RW         | Counting mode configuration<br>0: Timing mode, the divided clock of ICLK drives the counter to count.<br>1: Counting mode, the edge of the CH1 pin drives the counter to count.<br><i>Note: In encoder mode, it can only be set to 0.</i> |
| 22        | PRELOAD | RW         | ARR and CMP register update mode<br>0: Update registers immediately.<br>1: Update the registers when the LPTIM cycle ends.                                                                                                                |
| 21        | WAVPOL  | RW         | Polarity configuration of output waveforms<br>0: PWM positive output, $CNT \geq CMP$ output high level.<br>1: PWM reverse output, $CNT < CMP$ output high level.                                                                          |
| 20        | WAVE    | RW         | Waveform control<br>0: Single start SNGSTRT corresponds to single pulse waveform, continuous start CNTSTRT corresponds to PWM waveform.<br>1: Single set waveform.                                                                        |
| 19        | TIMOUT  | RW         | Trigger timeout enable<br>0: Trigger events that occur while counting are ignored.<br>1: A trigger event that occurs while counting is in progress will clear the count and restart counting.                                             |



| Bit field | Name    | Permission | Function description                                                                                                                                                                                                                                                                                                                                                                   |
|-----------|---------|------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 18:17     | TRIGEN  | RW         | Trigger enable and polarity<br>00: Software trigger, start the counter by software writing to the register.<br>01: Hardware trigger, start the counter on the rising edge of the trigger signal.<br>10: Hardware trigger, start the counter on the falling edge of the trigger signal.<br>11: Hardware trigger, start the counter on the rising or falling edge of the trigger signal. |
| 16        | RFU     | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                                                                          |
| 15:12     | TRIGSEL | RW         | Trigger source selection<br>0000: LPTIM_ETR pin signal<br>0001: ALARMA match signal for RTC<br>0010: ALARMB match signal for RTC<br>0011: Wake-up timer wake-up signal for RTC<br>0100: Timestamp event signal for RTC<br>0101: RTC_1Hz signal<br>0110: VC1 output signal<br>0111: VC2 output signal<br>1000: VC3 output signal<br>1001: VC4 output signal                             |
| 11:9      | PRS     | RW         | ICLK/CH1FP signal frequency division configuration<br>000: DIV1<br>001: DIV2<br>010: DIV4<br>011: DIV8<br>100: DIV16<br>101: DIV32<br>110: DIV64<br>111: DIV128<br><i>Attention: When COUNTMD is 1 and CKSEL is 0, only 000 can be configured.</i><br><i>Attention: When operating in encoder mode, only 000 can be configured.</i>                                                    |
| 8         | RFU     | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                                                                          |
| 7:6       | TRIGFLT | RW         | Trigger signal digital filter configuration<br>00: No filter functions<br>01: 2 ICLK clock cycles<br>10: 4 ICLK clock cycles<br>11: 8 ICLK clock cycles<br><i>Attention: When the ICLK clock is stopped, it can only be configured as no filter function.</i><br><i>Attention: When the trigger source selects the RTC related signal please do not set the filter time</i>            |
| 5         | RFU     | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                                                                          |
| 4:3       | CHFLT   | RW         | CH1/CH2 channel digital filter configuration<br>00: No filter functions<br>01: 2 ICLK clock cycles<br>10: 4 ICLK clock cycles<br>11: 8 ICLK clock cycles<br><i>Attention: Configurable only for no filtering when ICLK is stopped.</i>                                                                                                                                                 |



| Bit field | Name  | Permission | Function description                                                                                                                                                                                                                                                                                                          |
|-----------|-------|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 2:1       | ENCMD | RW         | Encoder working mode configuration<br>00: Code counting mode 1, CH1 signal change edge counting.<br>01: Code counting mode 2, CH2 signal change edge counting.<br>10: Code counting mode 3, CH1/CH2 signal change edge counting.<br>11: Reserved.                                                                             |
| 2:1       | CKPOL | RW         | CH1 signal count edge configuration<br>00: The counter increments on the rising edge of the CH1 signal.<br>01: The counter increments on the falling edge of CH1 signal.<br>10: The counter increments when the CH1 signal rises or falls.<br>11: Reserved.<br><i>Attention: When CKSEL is 1, only configurable as 00/01.</i> |
| 0         | CKSEL | RW         | Counter count clock TCLK source configuration<br>0: The count clock comes from MCLK and counts PCLK/LSE/LSI/CH1 edges.<br>1: The count clock comes from CH1FPD, and counts the clock input by CH1.<br><i>Attention: In encoder mode, only 0 can be configured.</i>                                                            |

*Attention: This register can only be set when LPTIM is disabled.*



### 15.7.2 LPTIM\_CR0 Control register 0

Address offset: 0x10      Reset value: 0x0000 0018

| Bit field | Name     | Permission | Function description                                                                                                                                                                                                                                                                 |
|-----------|----------|------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:5      | RFU      | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                        |
| 4         | ARST     | RW         | Asynchronous reset of counter count value .<br>W0: Reset counter count value.<br>W1: No function.<br>R0: The counter count value is being reset.<br>R1: Counter count value reset completed.<br><i>Attention: 0 can be written to reset only when this control bit is read as 1.</i> |
| 3         | SRST     | RW         | Synchronous reset of counter count value<br>W0: Reset counter count value.<br>W1: No function.<br>R0: The counter count value is being reset.<br>R1: Counter count value reset completed.<br><i>Attention: 0 can be written to reset only when this control bit is read as 1.</i>    |
| 2         | CNTSTART | RW         | Continuous Mode Operation Control<br>W0: No function.<br>W1: Start continuous mode.                                                                                                                                                                                                  |
| 1         | SNGSTART | RW         | One-shot mode operation control<br>W0: No function.<br>W1: Start single-pulse mode.                                                                                                                                                                                                  |
| 0         | EN       | RW         | LPTIM enable control<br>0: Disable LPTIM.<br>1: Enable LPTIM.                                                                                                                                                                                                                        |

### 15.7.3 LPTIM\_ARR Auto-reload register

Address offset: 0x18      Reset value: 0x0000 0001

| Bit field | Name | Permission | Function description                          |
|-----------|------|------------|-----------------------------------------------|
| 31:16     | RFU  | -          | Reserved bits, please keep the default value. |
| 15:0      | ARR  | RW         | Auto-reload value                             |



### 15.7.4 LPTIM\_CNT Counter register

Address offset: 0x1C      Reset value: 0x0000 0000

| Bit field | Name | Permission | Function description                                                                         |
|-----------|------|------------|----------------------------------------------------------------------------------------------|
| 31:16     | RFU  | -          | Reserved bits, please keep the default value.                                                |
| 15:0      | CNT  | RO         | Counter value<br><i>Note: Reading the same value twice in a row is considered read data.</i> |

### 15.7.5 LPTIM\_CMP Compare register

Address offset: 0x14      Reset value: 0x0000 0000

| Bit field | Name | Permission | Function description                          |
|-----------|------|------------|-----------------------------------------------|
| 31:16     | RFU  | -          | Reserved bits, please keep the default value. |
| 15:0      | CMP  | RW         | Comparison value                              |





### 15.7.6 LPTIM\_IER Interrupt enable register

Address offset: 0x08

Reset value: 0x0000 0000

| Bit field | Name    | Permission | Function description                                                                         |
|-----------|---------|------------|----------------------------------------------------------------------------------------------|
| 31:7      | RFU     | -          | Reserved bits, please keep the default value.                                                |
| 6         | DOWN    | RW         | Counting direction changes to reverse interrupt enable control<br>0: Disabled<br>1: Enabled  |
| 5         | UP      | RW         | Counting direction changes to positive interrupt enable control<br>0: Disabled<br>1: Enabled |
| 4         | ARROK   | RW         | Auto-reload register update complete interrupt enable control<br>0: Disabled<br>1: Enabled   |
| 3         | CMPOK   | RW         | Compare register update completion interrupt enable control<br>0: Disabled<br>1: Enabled     |
| 2         | EXTTRIG | RW         | Trigger completion interrupt enable control<br>0: Disabled<br>1: Enabled                     |
| 1         | ARRM    | RW         | Auto-reload match interrupt enable control<br>0: Disabled<br>1: Enabled                      |
| 0         | CMPM    | RW         | Compare match interrupt enable control<br>0: Disabled<br>1: Enabled                          |

*Attention: This register can only be set when LPTIM is disabled.*



### 15.7.7 LPTIM\_ISR Interrupt and status register

Address offset: 0x00

Reset value: 0x0000 0000

| Bit field | Name    | Permission | Function description                                                                                                                                                                                                                                                                                                                                                      |
|-----------|---------|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:9      | RFU     | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                                                             |
| 8:7       | DIR     | RO         | Encoder count direction<br>00: The encoder stops counting, and the current counting direction is counting down.<br>01: The encoder stops counting, and the current counting direction is counting up.<br>10: The encoder is counting, the current counting direction is counting down.<br>11: The encoder is counting, and the current counting direction is counting up. |
| 6         | DOWN    | RO         | Encoder counting direction becomes reverse sign<br>0: The counting direction has not changed.<br>1: Counting direction changes from counting up to counting down.                                                                                                                                                                                                         |
| 5         | UP      | RO         | Encoder counting direction changes to positive sign<br>0: The counting direction has not changed.<br>1: The counting direction changes from counting down to counting up.                                                                                                                                                                                                 |
| 4         | ARROK   | RO         | Auto-reload register update completion flag<br>0: Auto-reload registers are being updated.<br>1: Auto-reload registers has been updated.                                                                                                                                                                                                                                  |
| 3         | CMPOK   | RO         | Compare register update completion flag<br>0: Compare register is being updated.<br>1: The compare register has been updated.                                                                                                                                                                                                                                             |
| 2         | EXTTRIG | RO         | Trigger completion flag<br>0: No trigger event occurred.<br>1: Triggering event has occurred.                                                                                                                                                                                                                                                                             |
| 1         | ARRM    | RO         | Auto-reload match flag<br>0: The counter count value is not equal to the ARR register value.<br>1: The counter count value is equal to the ARR register value.                                                                                                                                                                                                            |
| 0         | CMPM    | RO         | Compare match flags<br>0: The count value of the counter is not equal to the value of the CMP register.<br>1: The count value of the counter is equal to the value of the CMP register.                                                                                                                                                                                   |



### 15.7.8 LPTIM\_ICR Interrupt clear register

Address offset: 0x04

Reset value: 0x0000 007F

| Bit field | Name    | Permission | Function description                                                                                                      |
|-----------|---------|------------|---------------------------------------------------------------------------------------------------------------------------|
| 31:7      | RFU     | -          | Reserved bits, please keep the default value.                                                                             |
| 6         | DOWN    | R1W0       | Counting direction changes to reverse flag clear<br>W0: Clear the count direction to reverse flag.<br>W1: No function.    |
| 5         | UP      | R1W0       | Counting direction changes to positive flag clear<br>W0: Clear the count direction to positive flag.<br>W1: No function.  |
| 4         | ARROK   | R1W0       | Auto-reload register update complete flag cleared<br>W0: Clear auto-reload register update done flag.<br>W1: No function. |
| 3         | CMPOK   | R1W0       | Compare register update complete flag is cleared<br>W0: Clear compare register update complete flag.<br>W1: No function.  |
| 2         | EXTTRIG | R1W0       | Trigger completion flag clear<br>W0: Clear trigger completion flag.<br>W1: No function.                                   |
| 1         | ARRM    | R1W0       | Auto-reload match flag clear<br>W0: Clear auto-reload match flag.<br>W1: No function.                                     |
| 0         | CMPM    | R1W0       | Compare match flag clear<br>W0: Clear compare match flag.<br>W1: No function.                                             |



### 15.7.9 LPTIM\_CR1 Control register 1

Address offset: 0x20      Reset value: 0x0000 0000

| Bit field | Name           | Permission | Function description                                                                                                          |
|-----------|----------------|------------|-------------------------------------------------------------------------------------------------------------------------------|
| 31:6      | RFU            | -          | Reserved bits, please keep the default value.                                                                                 |
| 5:3       | CH2INPUTSOURCE | RW         | CH2 channel input signal source configuration<br>000: CH2 pin<br>001: VC1_OUT<br>010: VC2_OUT<br>011: VC3_OUT<br>100: VC4_OUT |
| 2:0       | CH1INPUTSOURCE | RW         | CH1 channel input signal source configuration<br>000: CH1 pin<br>001: VC1_OUT<br>010: VC2_OUT<br>011: VC3_OUT<br>100: VC4_OUT |



## 16 General-purpose timer (GTIM)

### 16.1 Overview

CW32L012 integrates 4 general purpose timer (GTIM) internally. Each GTIM is completely independent and functionally identical, and each GTIM contains a 16-bit auto-reload counter and is driven by a programmable prescaler. The GTIM supports a variety of operating modes such as timing, counting, reset, gating, triggering, and encoder, etc., and each group GTIM with 4 independent capture/compare channels can measure the pulse width of the input signal (input capture) or generate output waveforms (output compare and PWM).



## 16.2 Main features

- 16bit increment, decrement and increment/decrement auto reload counter.
- Programmable prescaler supports 1,2,3,4,...,65536 frequency division.
- Support single count mode and continuous count mode.
- 4 independent input capture and output compare channels.
- Trigger input signal (TRGI) to control timer for multiple slave modes.
- Timer cascade ITR and on-chip peripheral interconnect ETR.
- Support incremental (quadrature) encoders and Hall sensor circuits for positioning.
- Interrupt/DMA request generated when various events occur:
  - Update event.
  - Trigger event.
  - Input capture.
  - Output compare.

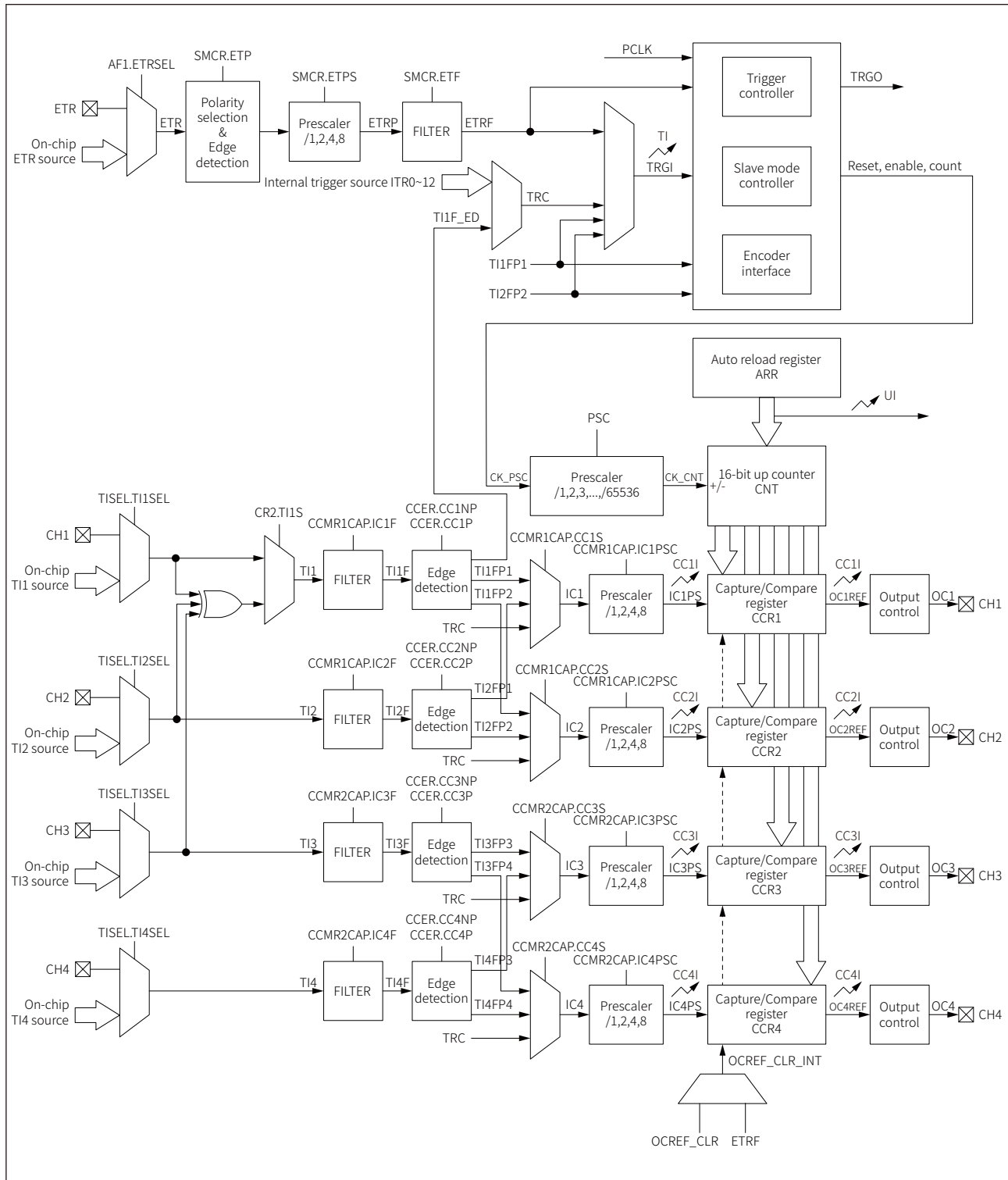


## 16.3 Functional description

### 16.3.1 Functional block diagram

The functional block diagram of GTIM is shown in the following figure:

Figure 16-1 GTIM functional block diagram



16.3.1.1 Clock source

The counter's counting clock source can be provided by the internal clock PCLK, the trigger signal TRGI or the external trigger input signal ETRF, which is divided by the prescaler GTIMx\_PSC to drive the counter for counting. Different operating modes have different clock sources, more details please refer to [16.3.2 Operating modes](#).

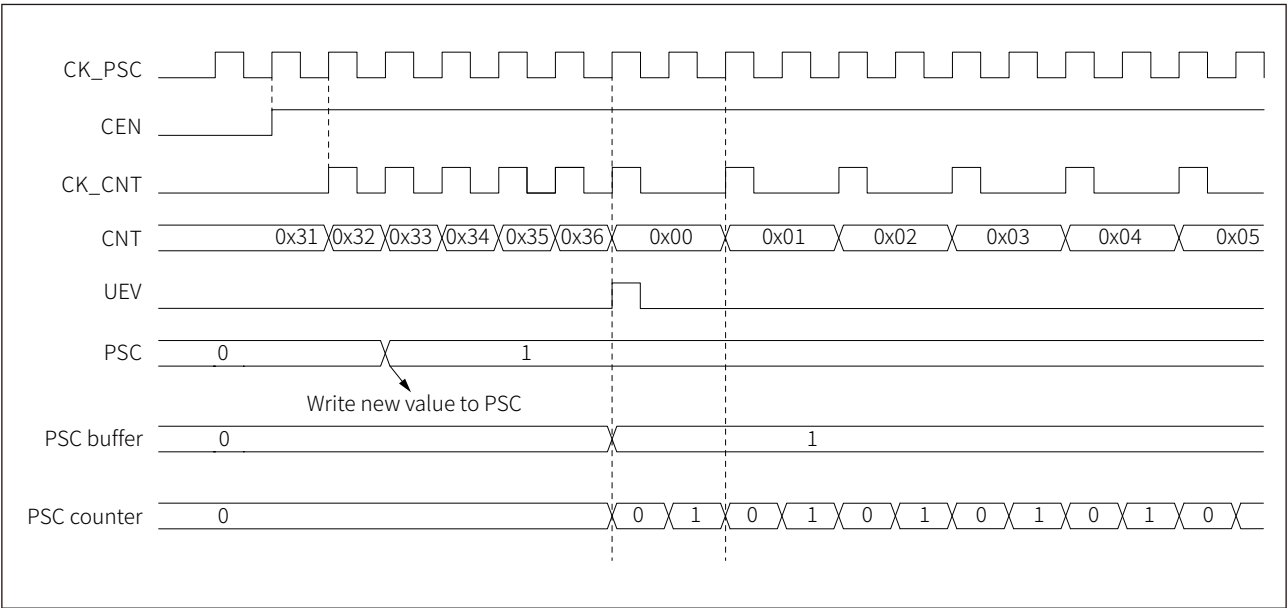
16.3.1.2 Prescaler

The prescaler divides the CK\_PSC clock to get the count clock CK\_CNT to drive the counter to count. The divider coefficient is set through the GTIMx\_PSC register, which supports 1,2,3,4,...,65536 divider.

GTIMx\_PSC register is buffered and can be modified during runtime, and the new prescaler value will take effect when the next update event occurs.

The following figure gives the timing diagram when the prescaler divider frequency changes from 1 to 2 during operation:

Figure 16-2 Timing diagram when the prescaler frequency divider changes from 1 to 2





### 16.3.1.3 Counter and count mode

The counter can be set to count up (edge-aligned mode), count down (edge-aligned mode), or count up/down (center-aligned mode). Specifically, it is configured through the CMS and DIR bit fields of the control register GTIMx\_CR1, as shown in the following table:

Table 16-1 Count mode

| GTIMx_CR1.CMS | GTIMx_CR1.DIR | Count mode                     |
|---------------|---------------|--------------------------------|
| 00            | 0             | Count up (edge-aligned mode)   |
|               | 1             | Count down (edge-aligned mode) |
| 01            | -             | Center-aligned mode 1          |
| 10            | -             | Center-aligned mode 2          |
| 11            | -             | Center-aligned mode 3          |

When the CEN bit field of the GTIMx\_CR1 register is set to 1, the counter starts counting in the set mode, noting that the actual counter enable signal, CNT\_EN, is set to 1 one clock cycle after CEN is set to 1.

#### Count up mode

In count up mode, the counter starts counting up from 0 to the reload value ARR and then starts counting up again from 0 while generating a counter overflow event. When the counter overflows, the overflow signal OV is generated, and the update event UEV is generated (the OV signal and the UEV signal are cleared automatically), and the counter update interrupt flag bit GTIMx\_ISR.UIF is set by the hardware, and if an interrupt is permitted (by setting GTIMx\_DIER.UIE to 1), an interrupt request is generated, and setting GTIMx\_ICR.UIF to 0 to clear this flag bit.

The following is an example of counter operation at different clock frequencies, where GTIMx\_ARR = 0x36:

Figure 16-3 Count up, internal clock frequency division factor is 1

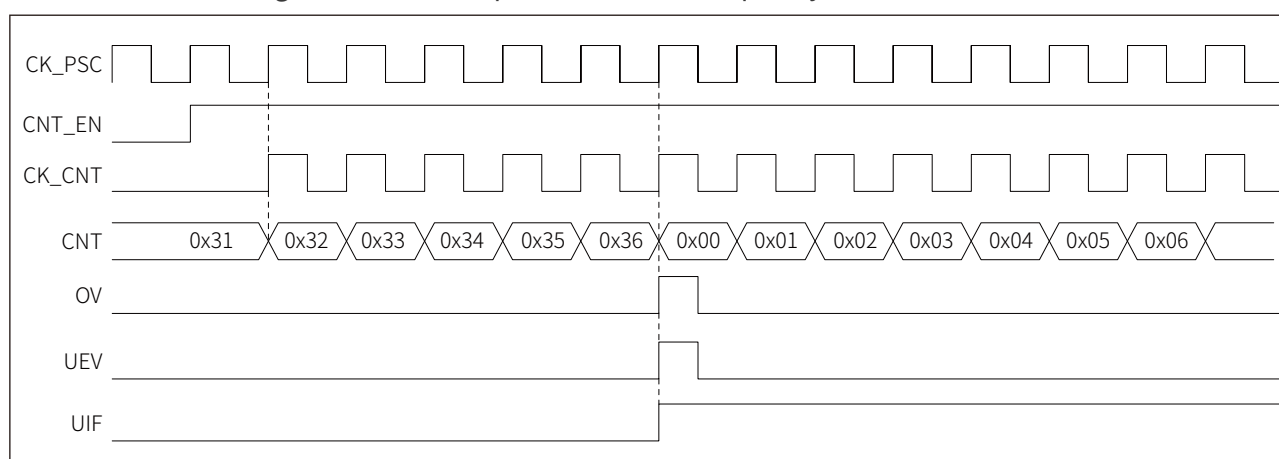


Figure 16-4 Count up, internal clock frequency division factor is 2

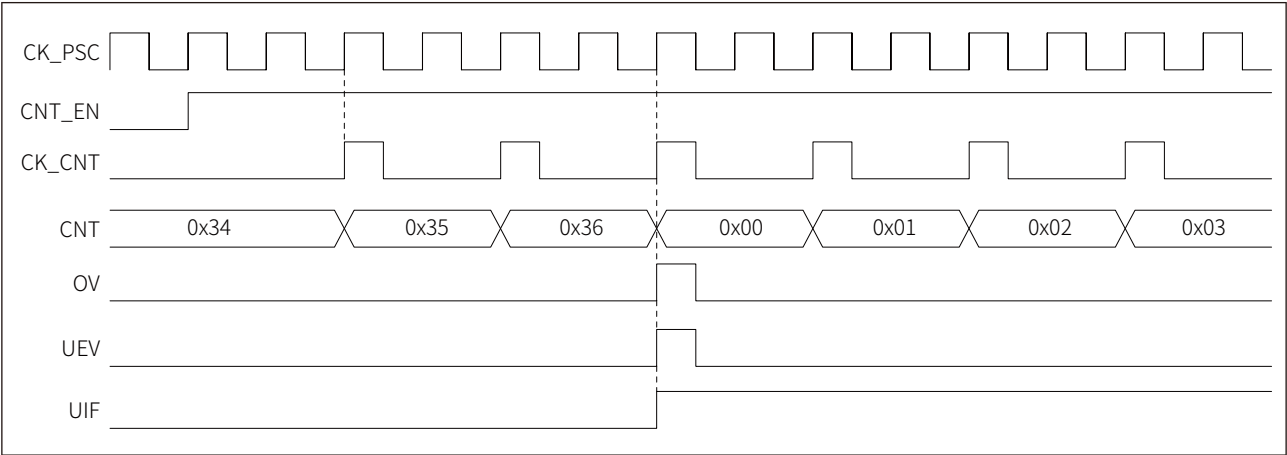


Figure 16-5 Count up, internal clock frequency division factor is 4

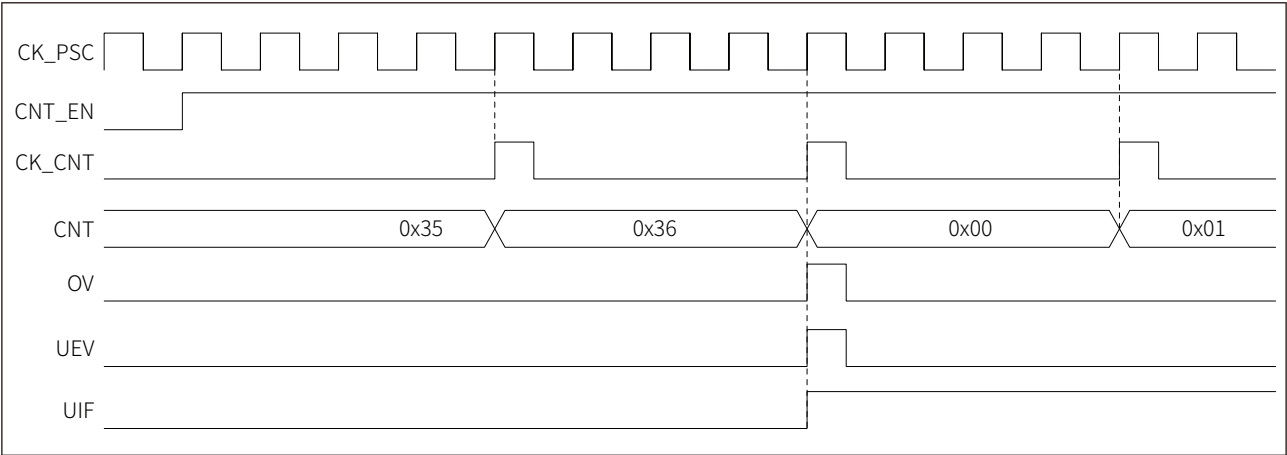


Figure 16-6 Count up, internal clock frequency division factor is N

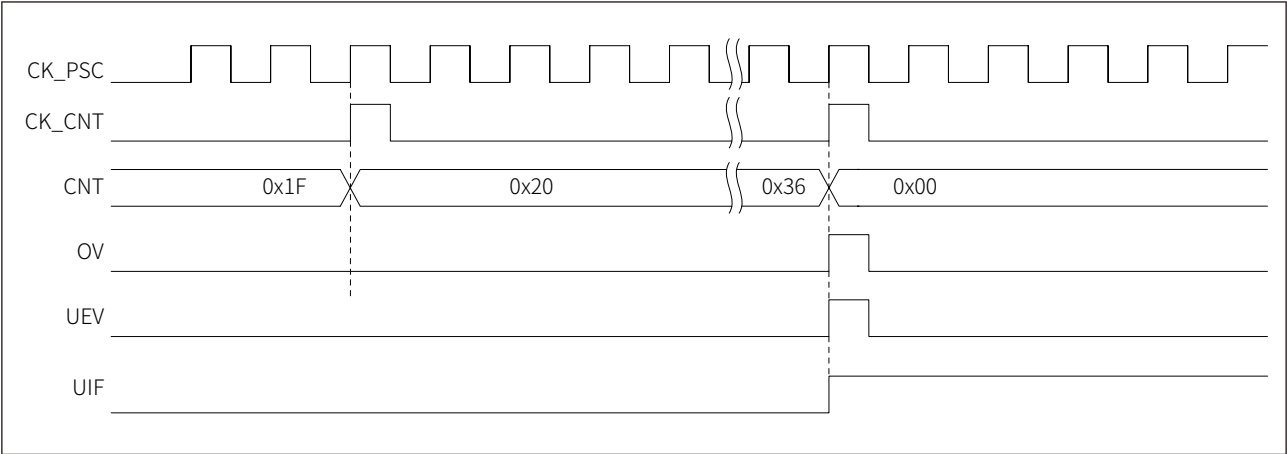


Figure 16-7 Count up, update event when reloading cache is disabled (GTIMx\_CR1.ARPE = 0)

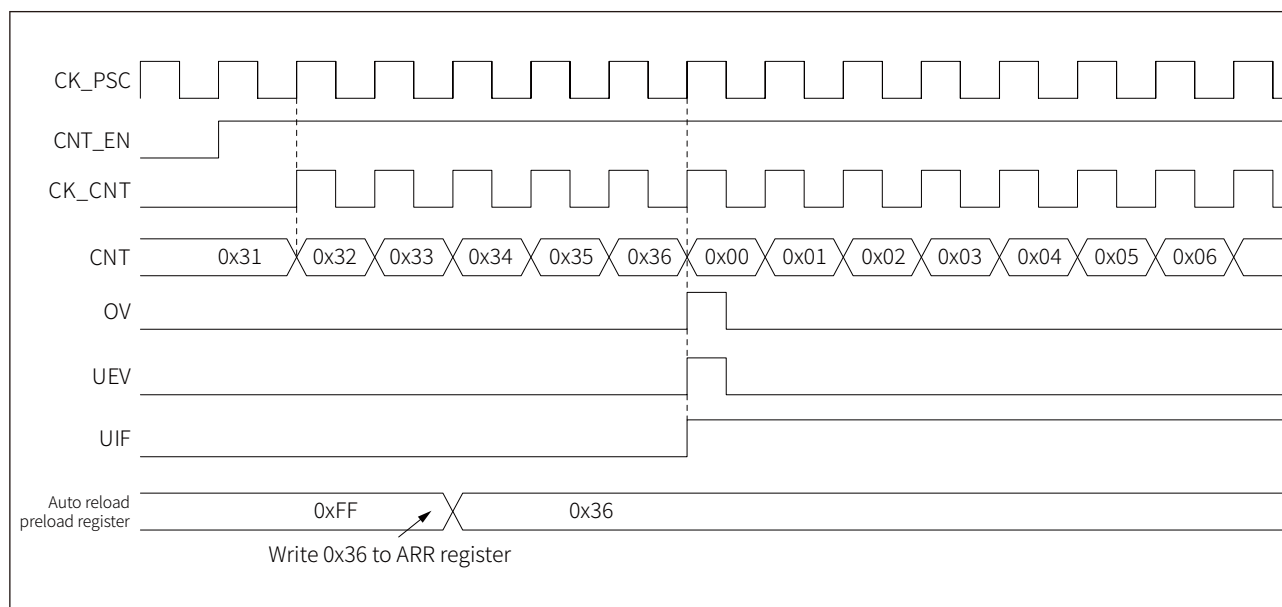
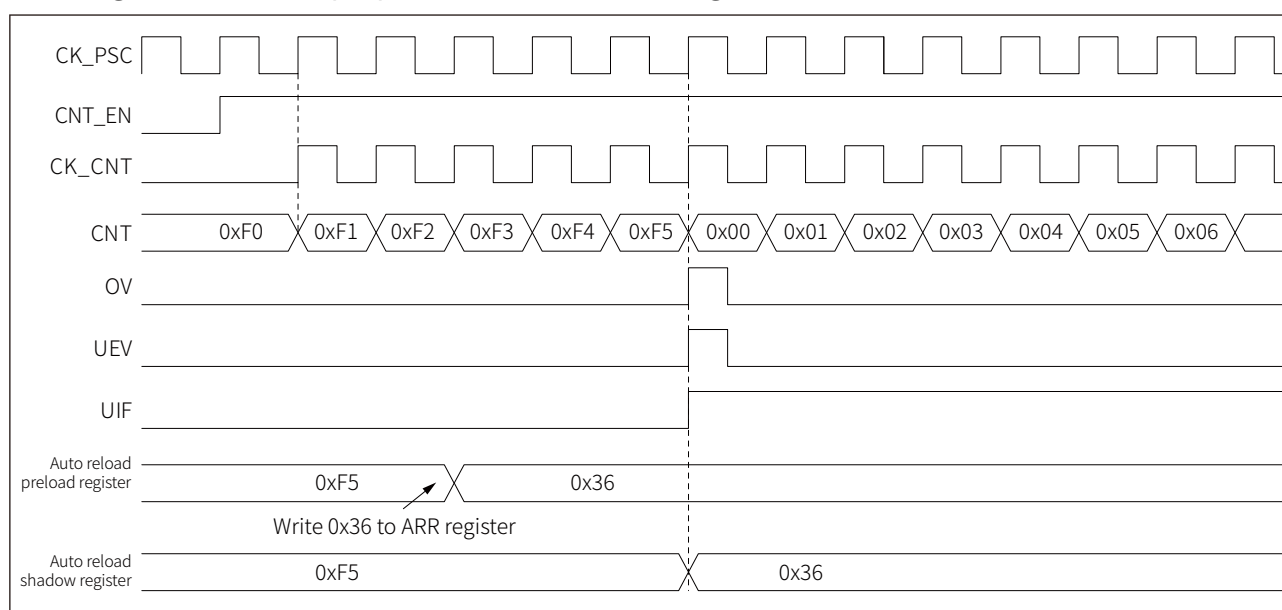


Figure 16-8 Count up, update event when reloading cache is enabled (GTIMx\_CR1.ARPE = 1)



## Count down mode

The counter operates in count down mode when the DIR bit of control register GTIMx\_CR1 is set to 1 in edge-aligned mode.

In count down mode, the hardware automatically loads the ARR value into the counter CNT, the counter starts count down to 0, and then reloads the ARR value to count down while generating a counter underflow event. When the counter underflows, the underflow signal UND is generated, and the update event UEV is generated (the UND signal and the UEV signal are cleared automatically), and the counter update interrupt flag bit GTIMx\_ISR.UIF is set by the hardware, and if an interrupt is permitted (by setting GTIMx\_DIER.UIE to 1), an interrupt request is generated, and setting GTIMx\_ICR.UIF to 0 clears this flag bit.

The following is an example of counter operation at different clock frequencies, where GTIMx\_ARR = 0x36:



Figure 16-9 Count down, internal clock frequency division factor is 1

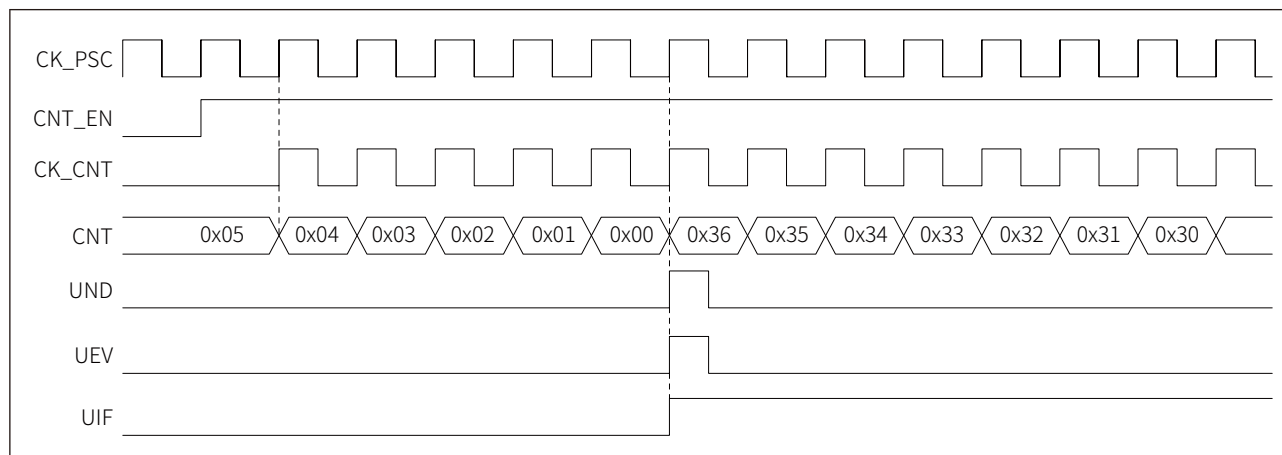


Figure 16-10 Count down, internal clock frequency division factor is 2

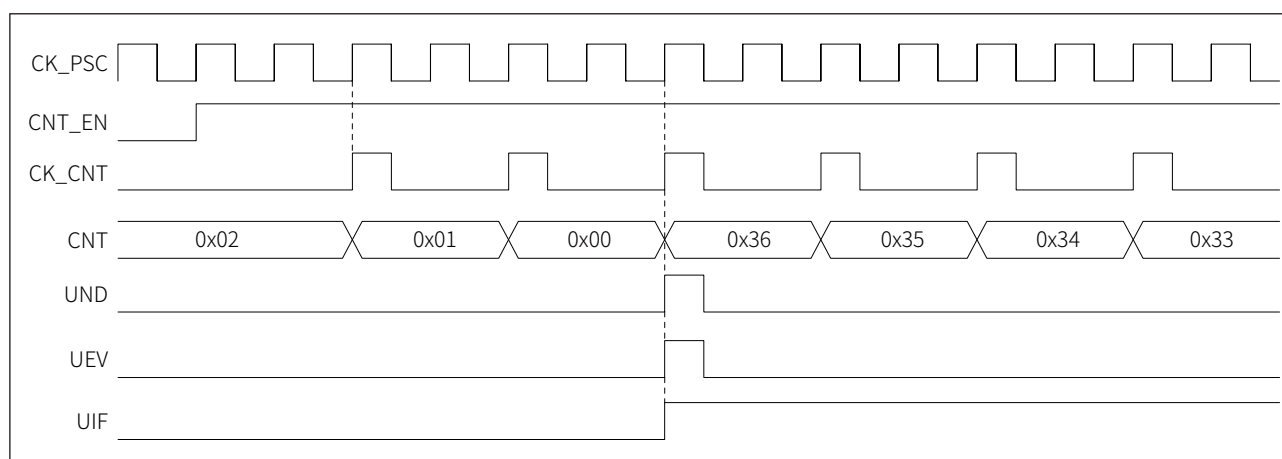


Figure 16-11 Count down, internal clock frequency division factor is 4

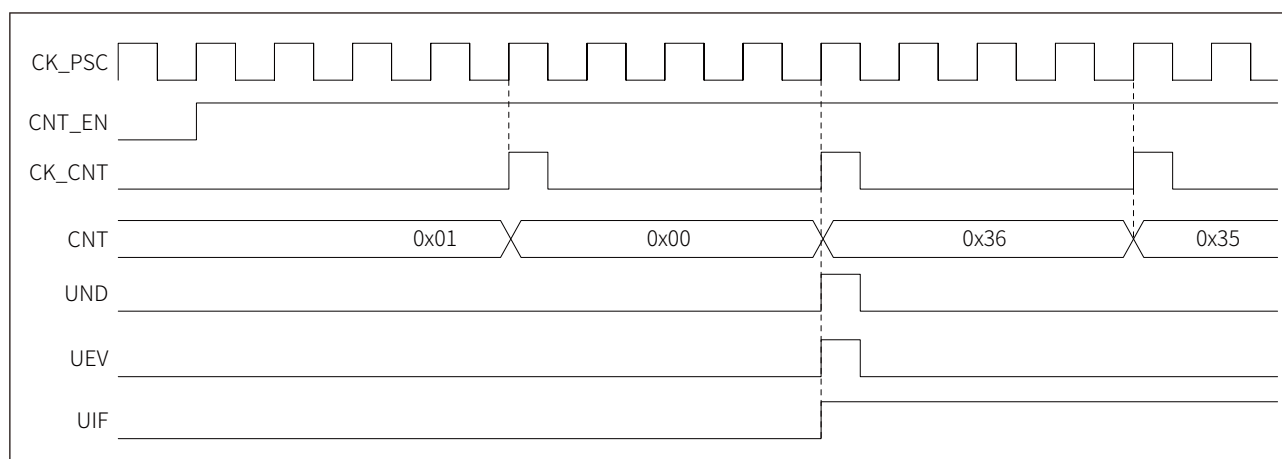
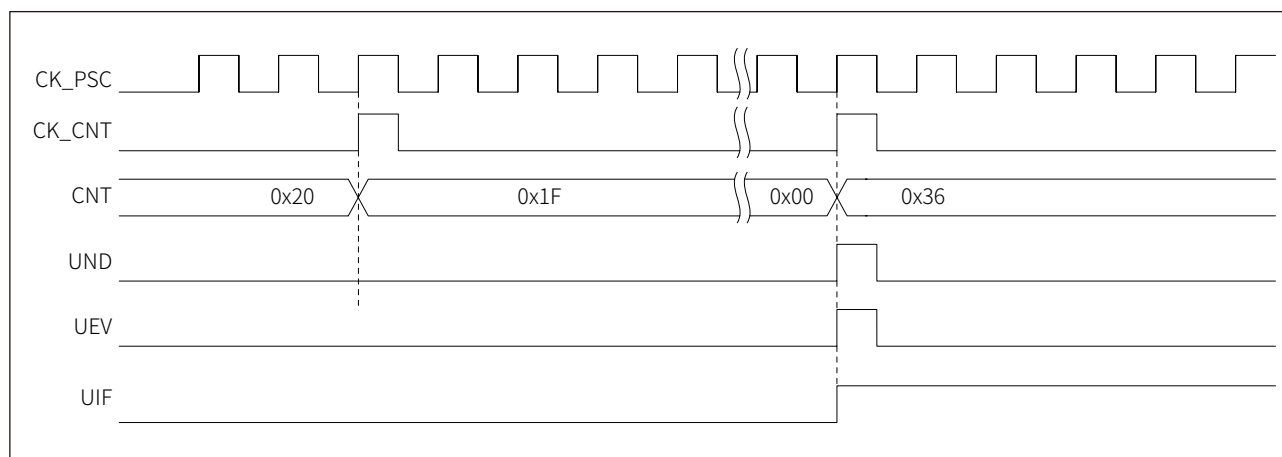


Figure 16-12 Count down, internal clock frequency division factor is N



### Center-aligned mode (count up/down)

In center-aligned mode, the counter counting up from 0 to ARR-1, generates a counter overflow event, then starts counting down from the ARR value to 1 and generates a counter underflow event, after which the counting starts counting up again from 0.

The center-aligned modes include center-aligned mode 1, center-aligned mode 2, and center-aligned mode 3. The 3 modes are identical in counting, except for the timing of the output compare interrupt flag, for details, please refer to the description of the CMS bit field of [16.10.1 GTIMx\\_CR1 Controls register 1](#).

In center-aligned mode, the DIR bit of control register GTIMx\_CR1 cannot be written by software but can be read, and DIR is updated by hardware and indicates the current count direction. When center-aligned mode is activated, the counter will count up or down based on the current DIR bit field value. It is not possible to modify the DIR and CMS bitfields at the same time through software.

It is not recommended to perform a write operation to the counter while running center-aligned mode, as unexpected results will occur. In particular:

- If a value greater than the auto-reload value ARR ( $CNT > ARR$ ) is written to the counter CNT, the direction will not be updated. That is, if the counter was previously counting up, it continues to count up.
- If a value of 0 or ARR is written to the counter CNT, the count direction is updated, but no update event UEV is generated.

The most secure way to use the center-aligned mode is to generate the update event UEV by software before starting the counter (set GTIMx\_EGR.UG to 1) and do not perform a write operation to the counter while it is running.

The following are examples of counter operation at different clock frequencies:

Figure 16-13 Center-aligned mode, internal clock frequency division factor is 1, ARR = 0x06

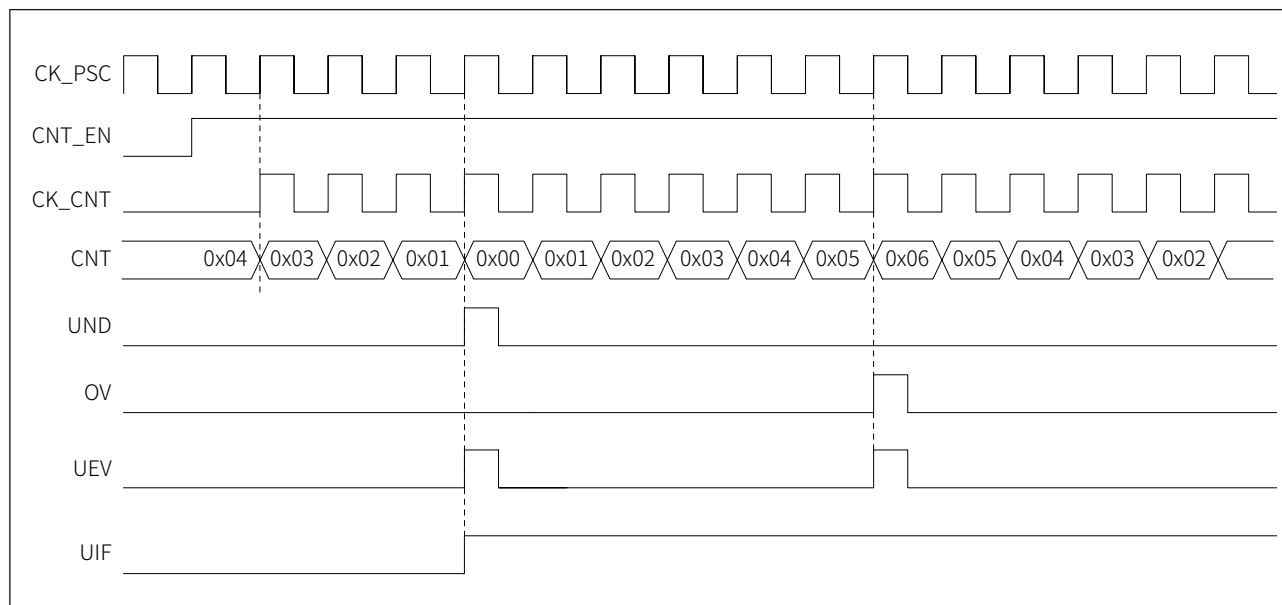


Figure 16-14 Center-aligned mode, internal clock frequency division factor is 2

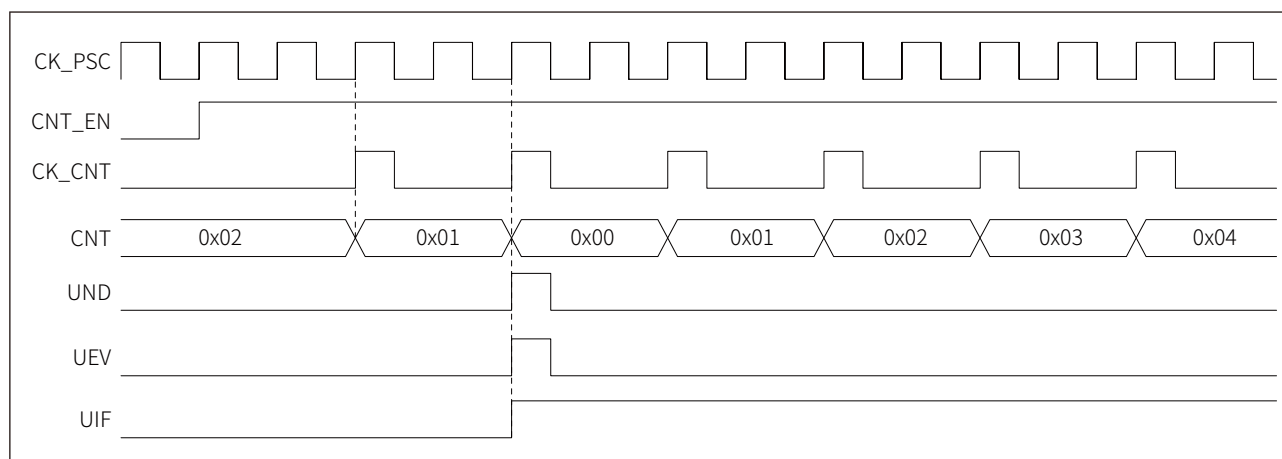


Figure 16-15 Center-aligned mode, internal clock frequency division factor is 4, ARR = 0x36

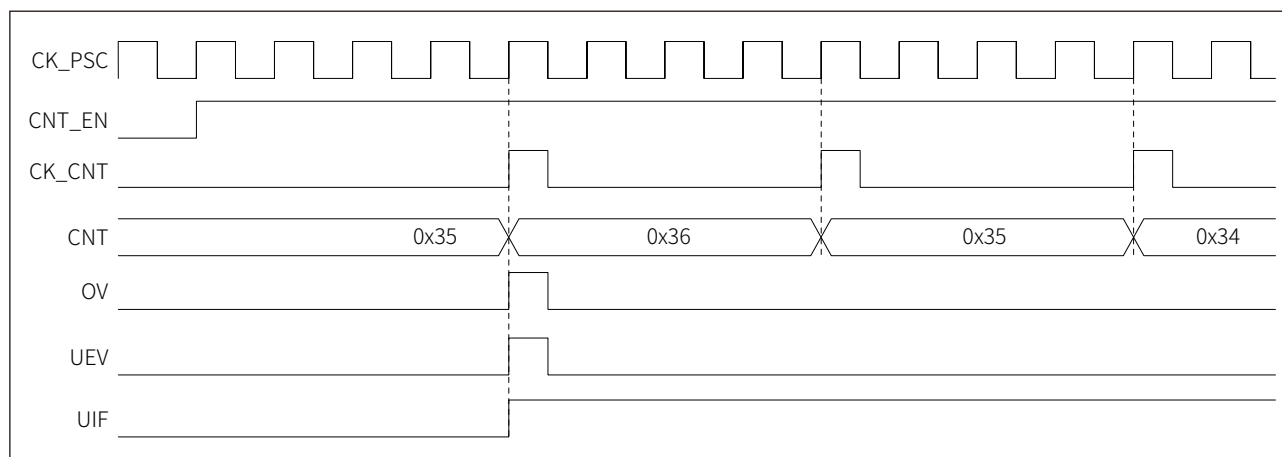


Figure 16-16 Center-aligned mode, internal clock frequency division factor is N

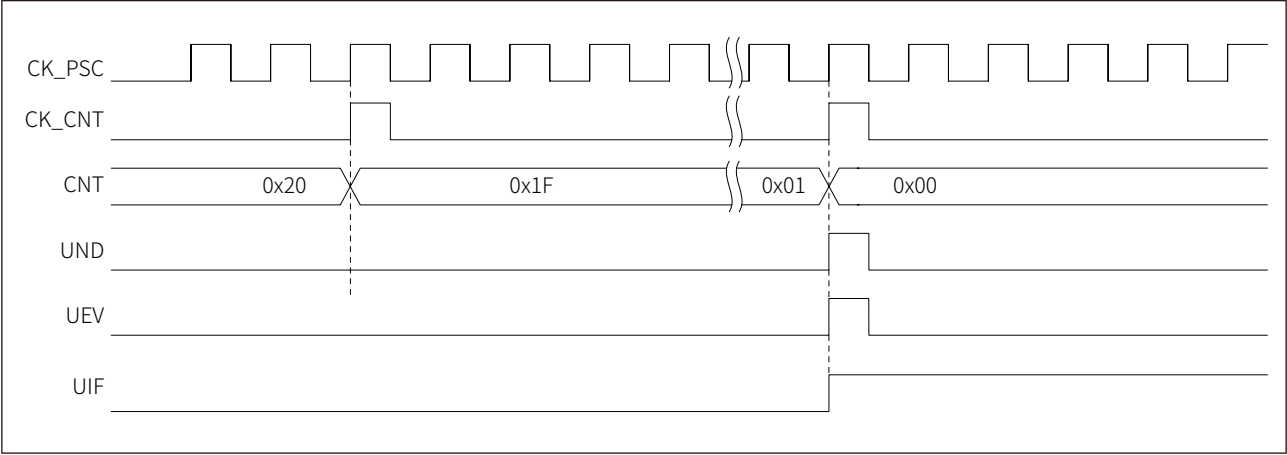


Figure 16-17 Center-aligned mode, update event when reloading cache is disabled (GTIMx\_CR1.ARPE = 0)

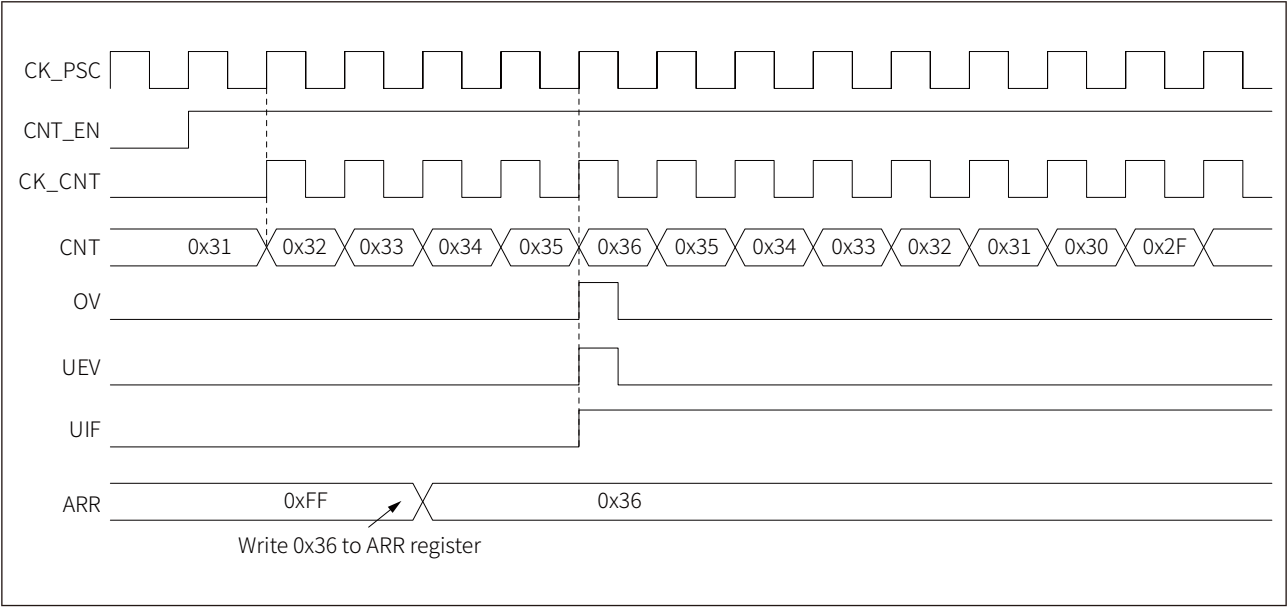
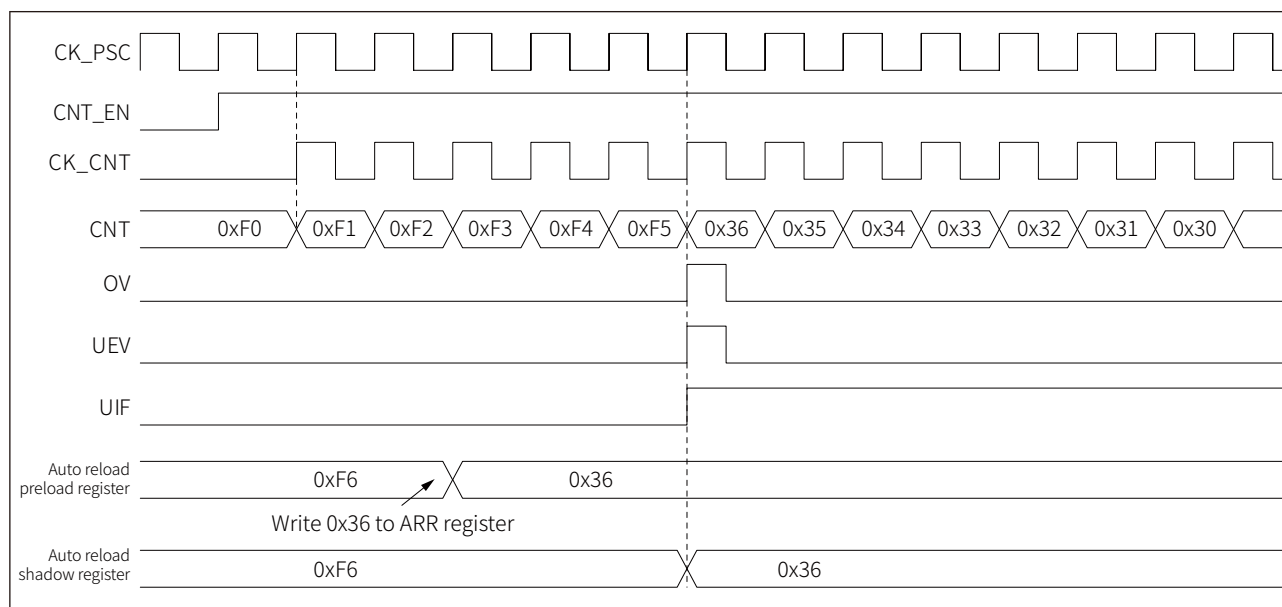


Figure 16-18 Center-aligned mode, update event when reloading cache is enabled (GTIMx\_CR1.ARPE = 1)



#### 16.3.1.4 Reload register

The auto reload register GTIMx\_ARR has a cache function that is enabled or disabled by controlling the ARPE bit field of register GTIMx\_CR1.

When the counter is stopped or the cache function is disabled, updating the reload register ARR will immediately update its shadow register. When the timer is running and the cache function is active, modifying the reload register ARR will not update the shadow register immediately, but will only update the value of the reload register ARR into the shadow register when an update event UEV is generated.



### 16.3.1.5 Update event UEV

Allows updates to be disabled or enabled by the UDIS bit field of the control register GTIMx\_CR1.

#### Enable UEV

Set the UDIS bit field to 0 to enable UEV. At this time, update request sources can be selected according to the URS bit field, as shown in the following table:

Table 16-2 Update source settings

| GTIMx_CR1.URS | Update source                                                                                          |
|---------------|--------------------------------------------------------------------------------------------------------|
| 0             | The counter overflow, underflow.<br>UG set.<br>Update events generated from the slave mode controller. |
| 1             | The counter overflow, underflow.                                                                       |

When an update event occurs, the following actions occur:

1. Reinitialise the counter:
  - Count down (edge-aligned mode) : Reloads the auto-reload value GTIMx\_ARR.
  - Center-aligned mode or count up (edge-aligned mode): the counter is cleared.
2. If the cache register function is used, the corresponding preloaded register is updated to its shadow register.
3. The prescaler counter is cleared and the new prescaler value in GTIMx\_PSC takes effect.
4. The event update interrupt flag bit GTIMx\_ISR.UIF is set by hardware.

When an update event UEV occurs, the event update interrupt flag bit GTIMx\_ISR.UIF is set by hardware. If the interrupt is allowed (set GTIMx\_DIER.UIE to 1), an update interrupt request is generated, and setting GTIMx\_ICR.UIF to 0 clears the flag bit.

#### Disable UEV

Set the UDIS bit field to 1 to disable UEV and no more update events are generated.

If the cache register function is used, the value of each corresponding shadow register remains unchanged. However, if the UG bit is set to 1 or a hardware reset is received from the slave mode controller, the counters of the counter and prescaler are reinitialised.



### 16.3.1.6 Single pulse mode

The counter can work in single count or continuous count mode and is selected by the OPM bit field of the control register GTIMx\_CR1.

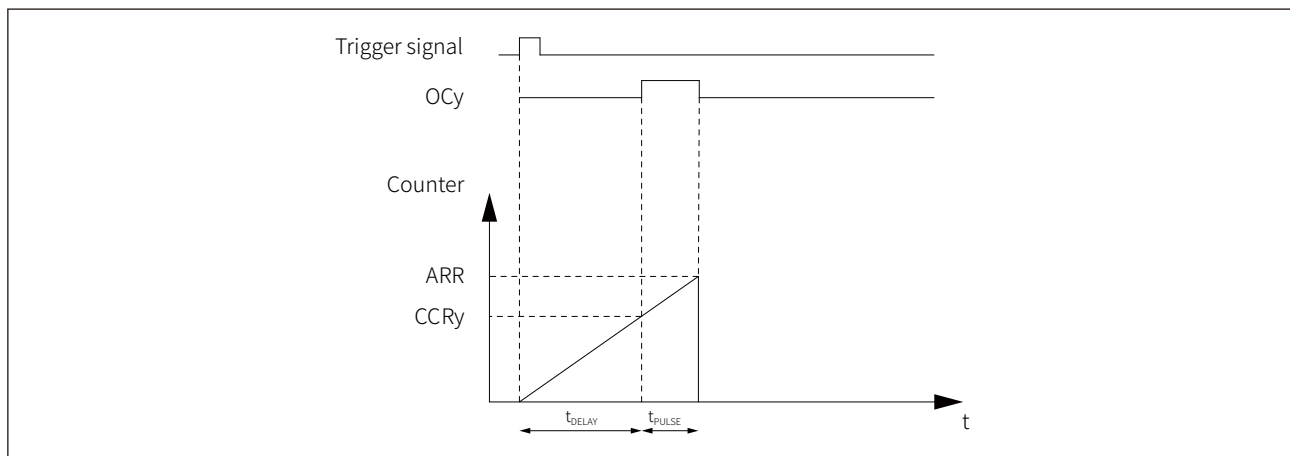
#### Single count mode

Set GTIMx\_CR1.OPM to 1 to enable the timer to work in single count mode.

After starting the counter, the counter CNT counts driven by the CK\_CNT clock, and the counter stops counting when the next update event occurs while GTIMx\_CR1.CEN is automatically cleared by hardware.

The counter can be activated by the slave mode controller, together with the output comparison mode of the timer, so that the counter is activated by a trigger signal and generates a single pulse with a programmable pulse width after a programmable delay. As shown in the figure below, the timer starts counting when it detects a valid edge of the trigger signal, and after a delay of  $t_{\text{DELAY}}$ , it generates a positive pulse of width  $t_{\text{PULSE}}$  on the compare output port. Where  $t_{\text{DELAY}}$  is determined by the difference between the initial value of the count register GTIMx\_CNT and the capture/compare register GTIMx\_CCRy, and  $t_{\text{PULSE}}$  is determined by the difference between the auto-reload register GTIMx\_ARR and the capture/compare register GTIMx\_CCRy.

Figure 16-19 Example of single pulse output (count up mode)



Sampling the trigger input to activate the OC output requires multiple clock cycles, and the OCyFE bit field of the comparison mode registers (GTIMx\_CCMR1CMP and GTIMx\_CCMR2CMP) can be set to 1 to shorten the OC output delay time.

#### Continuous count mode

Set GTIMx\_CR1.OPM to 0 so that the timer works in continuous count mode and the counter does not stop counting when an update event occurs.

### 16.3.1.7 External trigger input channel

The external trigger input signal ETR can be used as a trigger input (TRGI) for the slave mode controller or as a count clock for the counter, see [16.3.2 Operating modes](#). The ETR signal can be input controlled, including polarity selection and edge detection, prescaling and filtering.

#### Polarity selection and edge detection

The ETP bit field of the slave mode control register GTIMx\_SMCR is used to select the trigger polarity of the ETR input signal. When ETP is set to 0, ETR is not inverted and is valid at high level or rising edge; when ETP is set to 1, ETR is inverted and is valid at low level or falling edge.

#### Prescaler

The frequency of external trigger signal ETRP shall not exceed 1/4 of PCLK frequency. when the frequency of ETRP is high, user should divide the external signal by appropriate ETPS prescaler setting to reduce the ETRP frequency, and the dividing coefficient can be set to 1, 2, 4, 8.

#### Filter

The filter uses digital filtering to sample the input signal at a certain frequency, the signal is valid when it is continuously sampled to N same levels, otherwise the signal is invalid, so as to filter out the high-frequency clutter signal.

The filter of the ETR input signal is configured by the ETF bit field of the slave mode control register GTIMx\_SMCR, which can set the sampling frequency and the number of sampling points. The sampling frequency  $f_{\text{SAMPLING}}$  is provided by the clock after  $f_{\text{DTS}}$  dividing.  $f_{\text{DTS}}$  is the frequency obtained after  $f_{\text{PCLK}}$  dividing. The dividing factor is configured by the CKD bit field of the GTIMx\_CR1 register, which can be set to 1, 2, or 4 divisions.



### 16.3.1.8 Input capture channel

The GTIM supports 4 input capture channels TI1/2/3/4 which can be used as capture commands. Among them, TI1 and TI2 can also be used as trigger input (TRGI) and encoder interface input of slave mode controller, see [16.3.2 Operating modes](#).

Supports input control of Tly signals, including filtering, edge detection and prescaling.

#### Filter

In the filtering stage, the corresponding Tly input signal is sampled at a certain frequency, and the filtered signal TlyF is generated.

The filter for the Tly input signal is configured by the ICyF bit field of the capture mode registers GTIMx\_CCMR1CAP and GTIMx\_CCMR2CAP, which can be set for the sampling frequency and the number of sampling points. The sampling frequency  $f_{\text{SAMPLING}}$  is provided by the clock after  $f_{\text{PCLK}}$  or  $f_{\text{DTS}}$  dividing,  $f_{\text{DTS}}$  is the frequency obtained after  $f_{\text{PCLK}}$  dividing, and the dividing factor is configured by the CKD bit field of GTIMx\_CR1 register, and it can be set to 1, 2, or 4 divisions.

#### Edge detection

The edge detector with polarity selection generates a TlxFPx signal, for specific valid polarity please see the description of CCyNP and CCyP bit fields in [16.10.12 GTIMx\\_CCER Capture/compare enable register](#).

#### Prescaler

When the Tly signal is used for input capture, the capture channel signal ICy can be prescaled, which is controlled through the ICyPSC bit field of the capture mode registers GTIMx\_CCMR1CAP and GTIMx\_CCMR2CAP, and supports 1, 2, 4, and 8 frequency division.

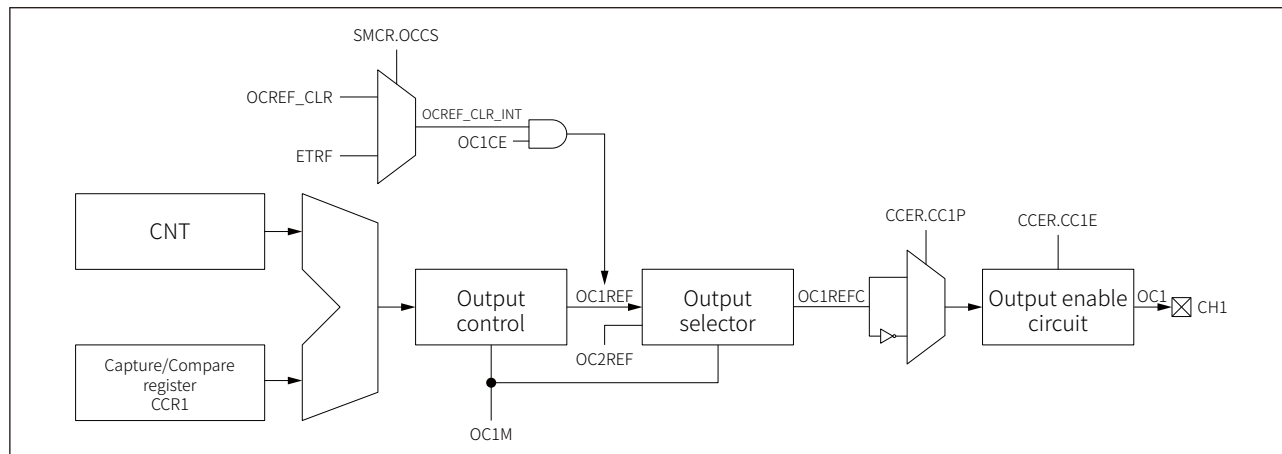


### 16.3.1.9 Output compare channel

The GTIM supports 4 independent output compare channels CH1/2/3/4, each consisting of a capture/compare register (including a shadow register) and an output stage (comparator and output control).

The output control unit is used to control the waveform of the output port when comparing and matching, and can configure the output compare mode, output polarity selection and output enable. Taking compare channel 1 as an example, its block diagram is shown below:

Figure 16-20 Output compare channel 1



### 16.3.2 Operating modes

The GTIM supports multiple operating modes, which are configured by the SMS bit field of the slave mode control register GTIMx\_SMCR, as shown in the following table:

Table 16-3 GTIM operating modes

| GTIMx_SMCR.SMS value | Slave mode selection                            |
|----------------------|-------------------------------------------------|
| 0000                 | Disable slave mode, use internal clock          |
| 0111                 | External clock mode 1                           |
| 0100                 | Reset mode                                      |
| 0101                 | Gated mode                                      |
| 0110                 | Trigger mode                                    |
| 1000                 | Combined reset + Trigger mode                   |
| 1001                 | Combined gated + Reset mode                     |
| 1110                 | Quadrature encoder mode, x1 mode                |
| 1111                 | Quadrature encoder mode, x1 mode                |
| 0001                 | Quadrature encoder mode, x2 mode                |
| 0010                 | Quadrature encoder mode, x2 mode                |
| 0011                 | Quadrature encoder mode, x4 mode                |
| 1010                 | Encoder mode (clock + direction), x2 mode       |
| 1011                 | Encoder mode (clock + direction), x1 mode       |
| 1100                 | Encoding mode ( with direction clock ), x2 mode |
| 1101                 | Encoding mode ( with direction clock ), x1 mode |

*Note: The slave mode select bits include GTIMx\_SMCR[16] and GTIMx\_SMCR[2:0] to be configured in combination.*

*Note: Slave mode select bits SMS[3:0] support the preload function, please refer to the description of SMSPE and SMSPS bit field in [16.10.3 GTIMx\\_SMCR Slave mode control register](#).*

#### 16.3.2.1 Internal clock mode

When the SMS bit field of the slave mode control register GTIMx\_SMCR is 0x0, the timer slave mode is disabled and the prescaler clock is provided directly by the internal clock PCLK. Setting GTIMx\_CR1.CEN to 1 will enable the counter to start counting.



### 16.3.2.2 External clock mode

#### External clock mode 1

When the SMS bit field of the slave mode control register GTIMx\_SMCR is 0x7, the counter clock is supplied by the rising edge of the selected trigger signal (TRGI). There are several trigger options for the TRGI signal, which are selected through the TS bit field of the GTIMx\_SMCR register, as shown in the table below:

Table 16-4 TRGI signal source

| GTIMx_SMCR.TS bit field value | TRGI signal source              |
|-------------------------------|---------------------------------|
| 00111                         | External trigger input (ETRF)   |
| 000xx/01xxx/100xx             | Internal trigger (ITR0~15)      |
| 00100                         | TI1 edge detector (TI1F_ED)     |
| 00101                         | Filtered timer input 1 (TI1FP1) |
| 00110                         | Filtered timer input 2 (TI2FP2) |

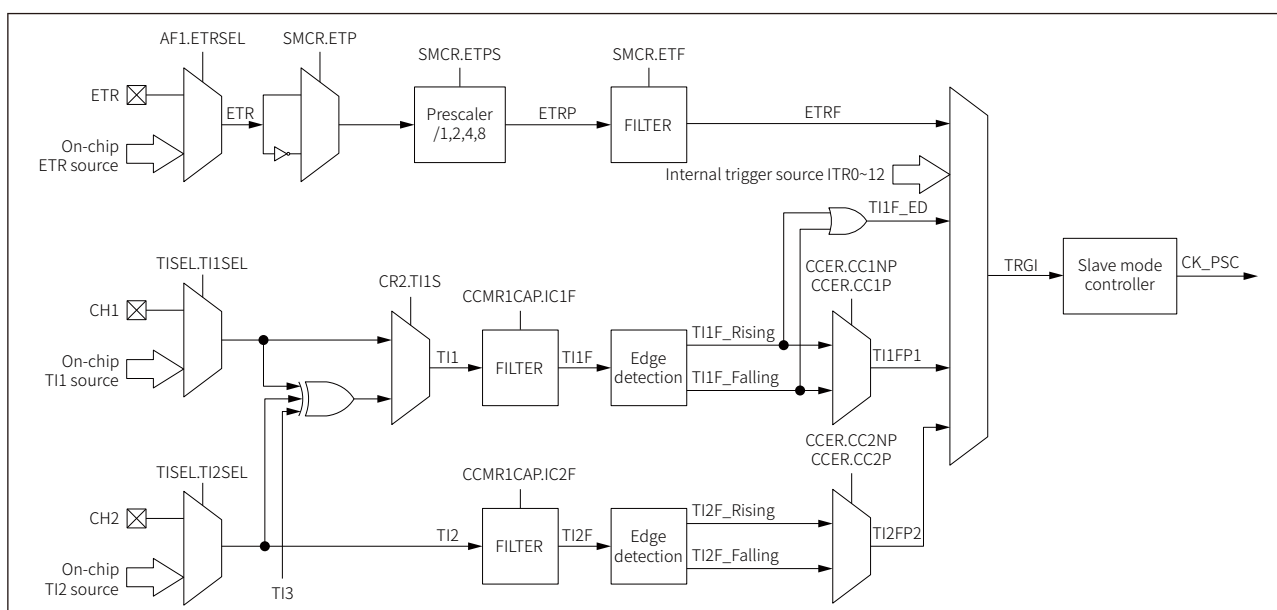
The ETR input signal source can be external GTIMx\_ETR pin or other peripherals on-chip, please refer to [16.3.7 On-chip peripheral interconnect ETR](#). When ETR is selected as the TRGI signal source, the external trigger polarity can be selected via GTIMx\_SMCR.ETP, the pre-scaler frequency can be set via GTIMx\_SMCR.ETPS, and the filtering control can be performed via GTIMx\_SMCR.ETF.

The internal trigger ITR source is the trigger output signal TRGO of BTIM and ATIM, and TXD/RXD signals of UART, please refer to [16.3.6 Timer cascade ITR](#).

Both TI1 and TI2 have filtering and edge detection functions. TI1 and TI2 are controlled by filtering through the IC1F and IC2F bit fields of the capture mode register GTIMx\_CCMR1CAP respectively, and edge detection through the CC1P/CC1NP and CC2P/CC2NP bit fields of the capture/compare enable register GTIMx\_CCER respectively.

The external clock mode 1 connection diagram is shown below:

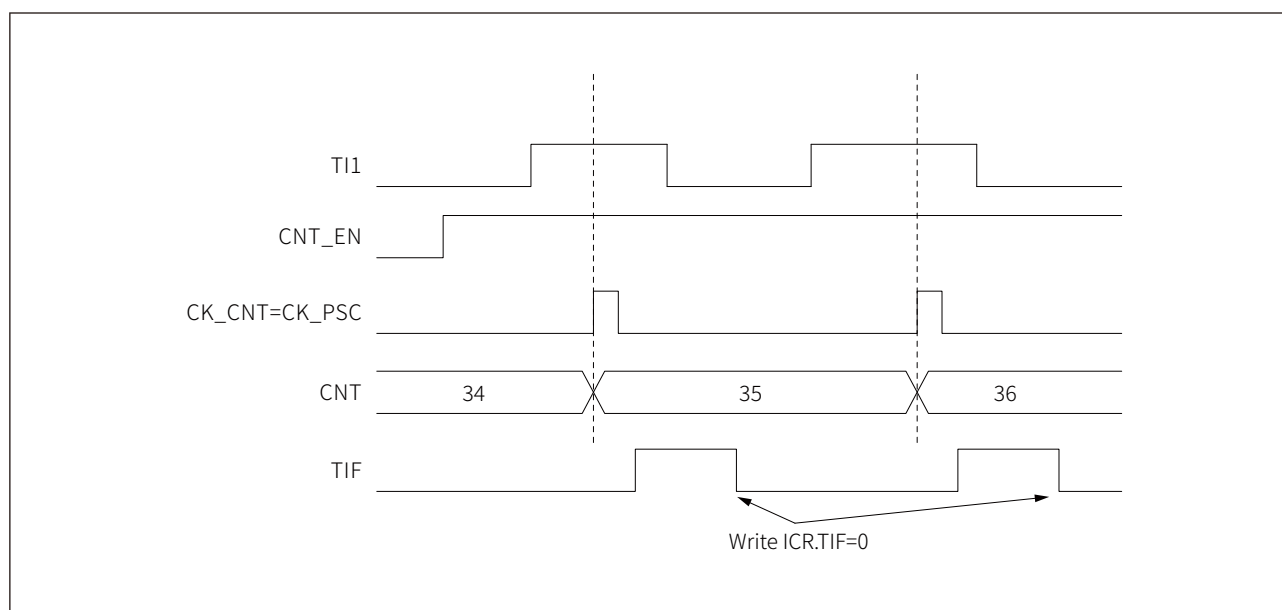
Figure 16-21 External clock mode 1 connection diagram



When a valid edge appears in TRGI, the GTIMx\_ISR.TIF flag will be set to 1, and the flag can be cleared by writing 0 to GTIMx\_ICR.TIF.

The following figure shows the external clock mode 1 timing diagram with TI1 rising edge configured. The delay between the rising edge of TI1 and the actual counter clock is due to the resynchronisation circuitry of the TI1 input.

Figure 16-22 Example of external clock mode 1 timing



### External clock mode 2

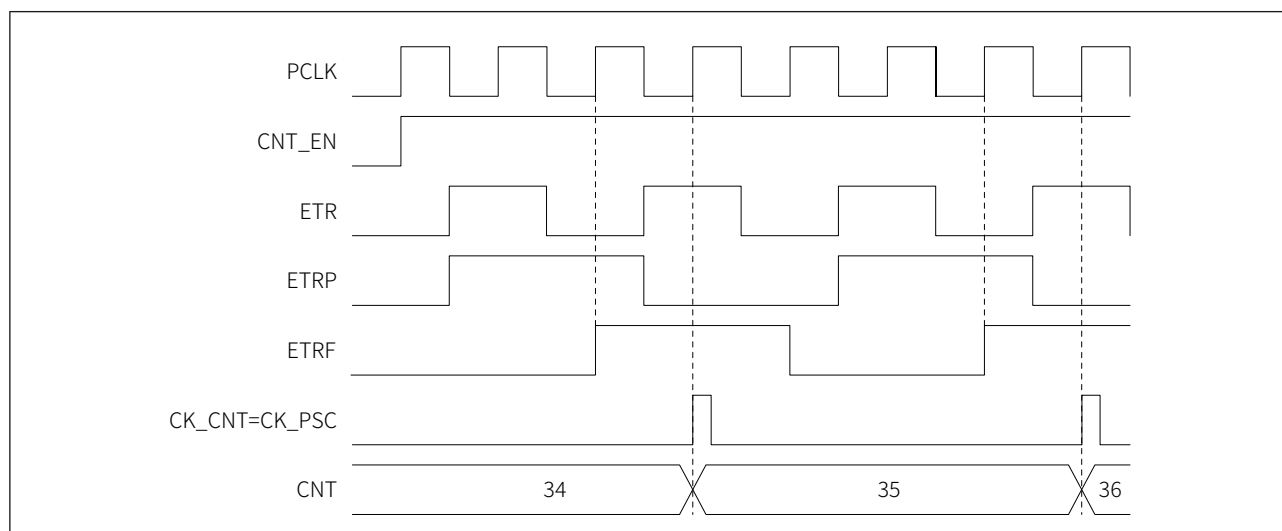
Set the ECE bit field of the slave mode control register GTIMx\_SMCR to 1 to enable external clock mode 2. In this mode, the counter clock is supplied by any valid edge of the ETRF signal, which has the same effect as selecting external clock mode 1 and connecting the TRGI to the ETRF (SMS = 0111 and TS = 00111), refer to [Figure 16-21: External clock mode 1 connection diagram](#).

External clock mode 2 can be used in conjunction with the following slave modes: reset mode, gated mode, and trigger mode, where the TRGI in slave mode must not be connected to the ETRF (i.e., the TS bit field cannot be set to 0x7). If both external clock mode 1 and external clock mode 2 are enabled, the external clock input is ETRF. External clock mode 2 cannot be used in conjunction with encoder mode.

The following figure shows a timing example of external clock mode 2 when ETR is set to 2 divisions, the delay between the rising edge of ETR and the actual counter clock is caused by the resynchronisation circuitry of the ETRP signal, so the maximum frequency that can be correctly captured by the counter is up to 1/4 of the internal clock PCLK frequency, and the user should divide the external signals appropriately when the ETRP signal becomes faster.



Figure 16-23 Example of external clock mode 2 timing



### 16.3.2.3 Reset mode

When the SMS bit field of the slave mode control register GTIMx\_SMCR is 0x4, the GTIM is configured in reset mode and the reset of the counter is controlled by the TRGI signal.

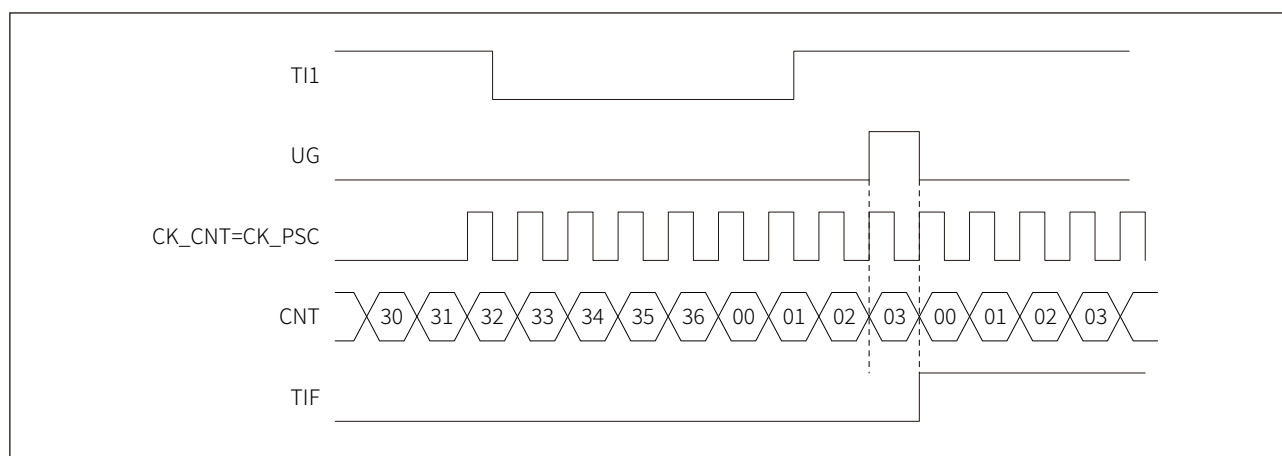
The source of the TRGI signal is controlled by the TS bit field of the slave mode control register GTIMx\_SMCR. See [Table 16-4: TRGI signal source](#).

When a valid trigger input (TRGI) is detected, the following effects occur:

- Reinitialise the counters and prescaler counters.
- If the URS bit field of control register GTIMx\_CR1 is 0, an update event UEV is generated, and the event update interrupt flag GTIMx\_ISR.UIF is set to 1. An interrupt/DMA request can be generated.
- The trigger interrupt flag GTIMx\_ISR.TIF is set to 1, which can generate an interrupt/DMA request.

The following figure shows an example of the reset mode timing diagram. Set GTIMx\_CR1.CEN to 1 to enable the counter, the counter counts normally according to the counting clock CK\_CNT, and when a rising edge of TI1 occurs, the counter is cleared to 0 and starts counting from 0 again, while the GTIMx\_ISR.TIF flag is set to 1. The delay between the rising edge of TI1 and the actual counter reset is caused by the resynchronisation circuitry of the TI1 input.

Figure 16-24 Reset mode timing



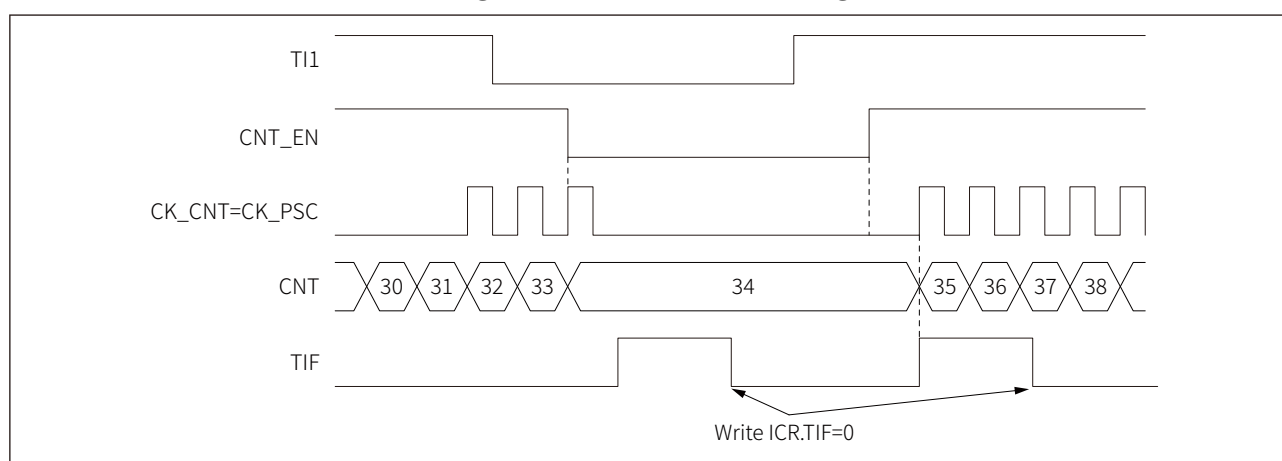
### 16.3.2.4 Gated mode

When the SMS bit field of the slave mode control register GTIMx\_SMCR is 0x5, the GTIM is configured for gated mode. In this mode, the counter is started counting when the trigger input (TRGI) is high and GTIMx\_CR1.CEN is 1. The counter stops counting immediately (but is not reset) when the trigger input (TRGI) is low or GTIMx\_CR1.CEN is 0. Both the start and stop of the counter are controlled.

The source of TRGI signals is controlled by the TS bit field of the slave mode control register GTIMx\_SMCR. For details, see [Table 16-4: TRGI signal source](#), however it should be noted that TI1F\_ED cannot be selected as a trigger input in gated mode.

The following figure shows an example of gated mode timing diagram. When TI1 is high, the counter starts counting; when TI1 is low, the counter pauses counting. The GTIMx\_ISR.TIF flag bit is set to 1 when the counter starts and stops. The delay between the edge of TI1 and the actual counter enable signal CNT\_EN is caused by the resynchronisation circuitry of the TI1 input.

Figure 16-25 Gated mode timing



### 16.3.2.5 Trigger mode

When the SMS bit field of the slave mode control register GTIMx\_SMCR is 0x6, the GTIM is configured for trigger mode. In this mode, when GTIMx\_CR1.CEN is set to 1 or when the rising edge of the trigger signal TRGI appears, the start counter is triggered (but not reset).

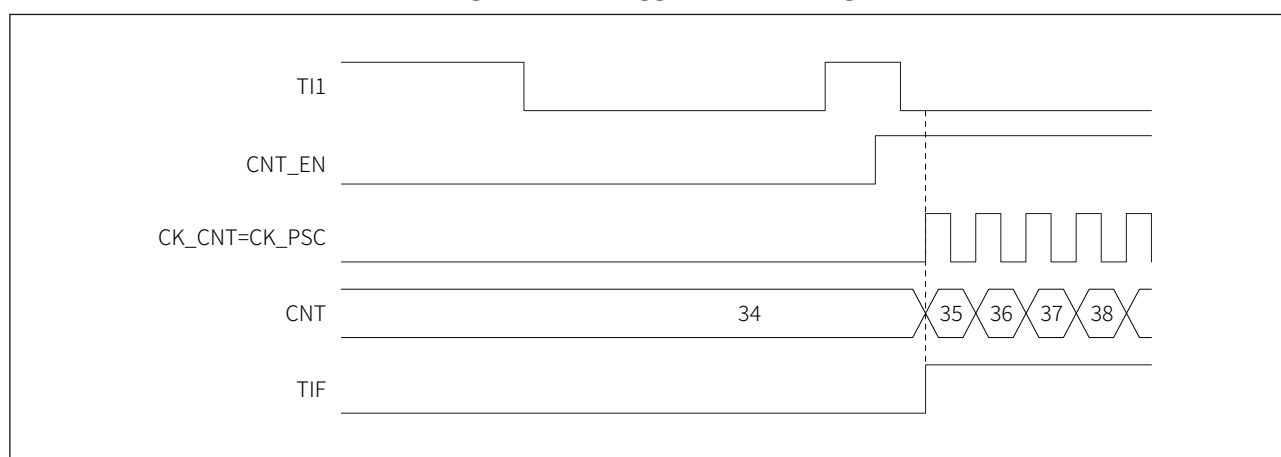
The source of the TRGI signal is controlled by the TS bit field of the slave mode control register GTIMx\_SMCR, see [Table 16-4: TRGI signal source](#).

When a valid trigger signal is detected, the following effects occur:

- GTIMx\_CR1.CEN is set by hardware.
- The trigger interrupt flag bit GTIMx\_ISR.TIF is set to 1, which can generate an interrupt/DMA request.
- The counter is activated and starts counting.

The following figure shows an example of a trigger mode timing diagram. When a rising edge of TI1 occurs, the counter starts counting while the GTIMx\_ISR.TIF flag bit is set to 1. The delay between the rising edge of TI1 and the actual counter startup is caused by the resynchronisation circuitry of the TI1 input.

Figure 16-26 Trigger mode timing



### 16.3.2.6 Combined reset + trigger mode

When the SMS bit field of the slave mode control register GTIMx\_SMCR is 0x8, the GTIM is configured for combined reset + trigger mode. In this mode, when a rising edge of the selected trigger input (TRGI) occurs, the counter will be reinitialised and the counter will be started, and the trigger interrupt flag GTIMx\_ISR.TIF will be set to 1. If the URS bit field of the control register GTIMx\_CR1 is 0, an update event UEV will be generated, and the event update interrupt flag GTIMx\_ISR.UIF will be set by hardware.

The source of the TRGI signal is controlled by the TS bit field of the slave mode control register GTIMx\_SMCR, see [Table 16-4: TRGI signal source](#).

### 16.3.2.7 Combined gated + reset mode

When the SMS bit field of the slave mode control register GTIMx\_SMCR is 0x9, the GTIM is configured for the combined gating + reset mode. In this mode, the start and stop of the counter are controlled. When the trigger input (TRGI) is high and GTIMx\_CR1.CEN is 1, the counter is started to count. When the trigger input (TRGI) is low, the counter immediately stops counting and is reset. When the counter is started and stopped, the GTIMx\_ISR.TIF flag bit is set to 1. When the counter is reset, GTIMx\_ISR.UIF flag bit is set to 1.

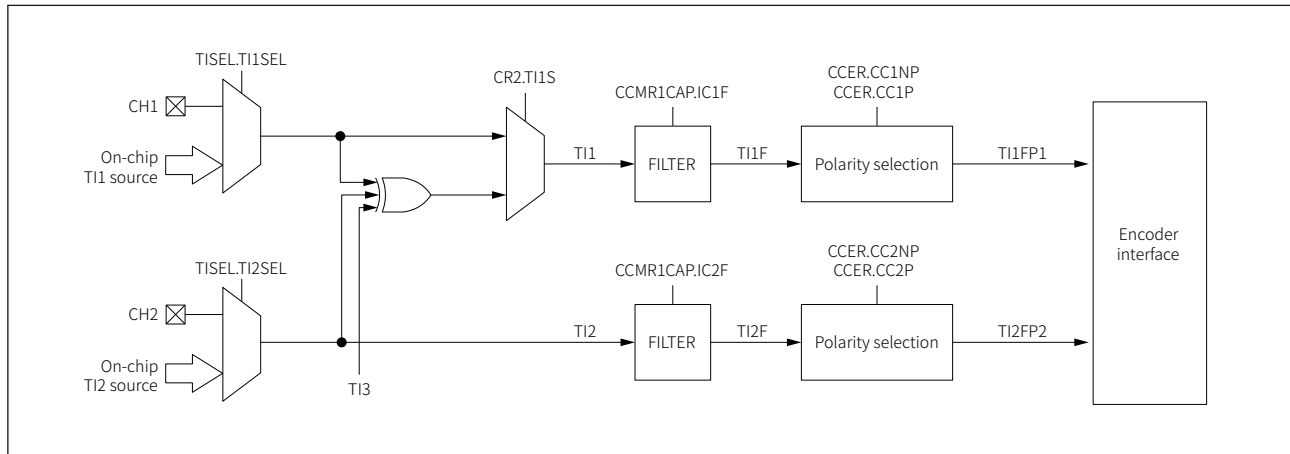
The source of the TRGI signal is controlled by the TS bit field of the slave mode control register GTIMx\_SMCR, see [Table 16-4: TRGI signal source](#).



### 16.3.2.8 Quadrature encoder mode

GTIM supports quadrature encoder mode for receiving and decoding quadrature encoder signals. In this mode, it allows a direct connection to an external quadrature encoder via the CH1 and CH2 pins, which enables the counter to count up or down automatically according to the hopping sequence of the input signals, and the counter value always indicates the position of the encoder. The functional block diagram is shown below:

Figure 16-27 Encoder mode block diagram



The CH1 and CH2 input signals have filtering and polarity selection functions, which are filtering controlled through the IC1F and IC2F bit fields of the GTIMx\_CCMR1CAP register, respectively, and the input polarity is selected through the CC1P and CC2P bit fields of the GTIMx\_CCER register, and the CC1NP and CC2NP bit fields must be kept clear.

Set GTIMx\_CR1.CEN to 1 to enable the counter, the counter will be driven by each valid hop of the signals TI1FP1 and TI2FP2 after the input signals of the channel CH1 and CH2 pins have been filtered and polarity selected, and according to the hopping sequence of the two input signals, the counting pulses and the direction signals are generated, refer to [Table 16-5: Relationship between counting direction and encoder signal](#). The encoder's current count direction flag GTIMx\_CR1.DIR is automatically set and cleared by hardware, and the DIR bit is counted whenever a signal transition occurs on any input (CH1 or CH2), regardless of whether the counter is counting only at the TI1FP1 or TI2FP2 edges, or at both TI1FP1 and TI2FP2.

GTIM supports several quadrature coded counting modes, which are set by the SMS bit field of the slave mode control register GTIMx\_SMCR. The relationship between the counting direction and the encoder signals in different modes is shown in the table below:

Table 16-5 Relationship between counting direction and encoder signal

| Mode                                  | SMS  | Signal level |        | TI1FP1     |          | TI2FP2   |            |
|---------------------------------------|------|--------------|--------|------------|----------|----------|------------|
|                                       |      | TI2FP2       | TI1FP1 | Rising     | Falling  | Rising   | Falling    |
| Counting on the TI1FP1 edge (x1 mode) | 1110 | High         | -      | Down count | Up count | No count | No count   |
|                                       |      | Low          | -      | No count   | No count | No count | No count   |
| Counting on the TI2FP2 edge (x1 mode) | 1111 | -            | High   | No count   | No count | Up count | Down count |
|                                       |      | -            | Low    | No count   | No count | No count | No count   |

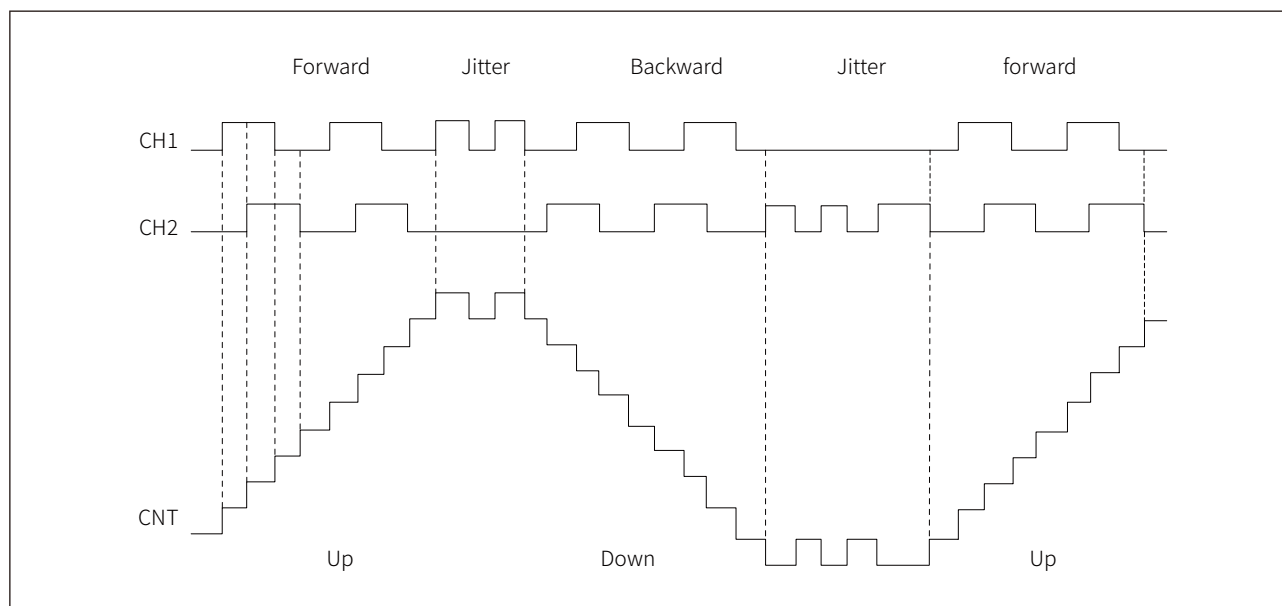
Table 16-5 Relationship between counting direction and encoder signal (continued)

| Mode                                              | SMS  | Signal level |        | TI1FP1     |            | TI2FP2     |            |
|---------------------------------------------------|------|--------------|--------|------------|------------|------------|------------|
|                                                   |      | TI2FP2       | TI1FP1 | Rising     | Falling    | Rising     | Falling    |
| Counting on the TI1FP1 edge (x2 mode)             | 0001 | High         | -      | Down count | Up count   | No count   | No count   |
|                                                   |      | Low          | -      | Up count   | Down count | No count   | No count   |
| Counting on the TI2FP2 edge (x2 mode)             | 0010 | -            | High   | No count   | No count   | Up count   | Down count |
|                                                   |      | -            | Low    | No count   | No count   | Down count | Up count   |
| Counting on the TI1FP1 and TI2FP2 edges (x4 mode) | 0011 | High         | High   | Down count | Up count   | Up count   | Down count |
|                                                   |      | Low          | Low    | Up count   | Down count | Down count | Up count   |

The third signal from the encoder output represents the mechanical zero point and can be connected to an external trigger input pin to trigger a counter reset.

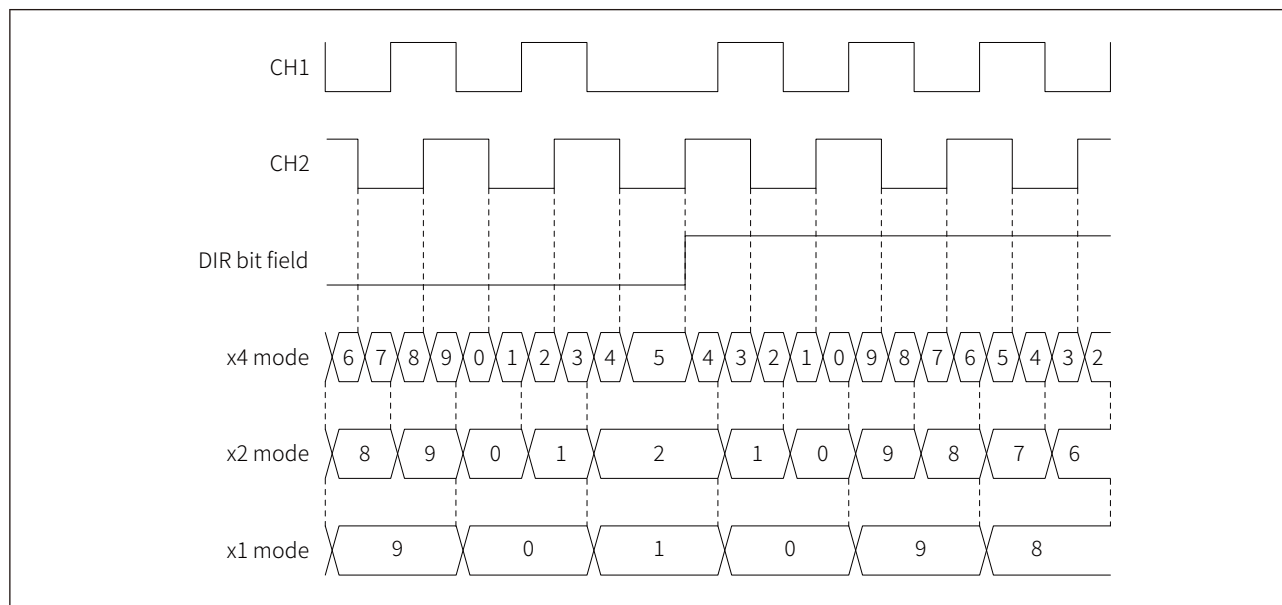
The following figure is an example of quadrature encoded counting mode-x4 mode operation, showing the generation of the counting signal and direction control. It also shows how input jitter is suppressed when a double edge is selected; jitter may be generated when the sensor is positioned close to a transition point.

Figure 16-28 Example of quadrature encoder mode-x4 mode operation



The following figure shows the timer count values during encoder inversion for various quadrature coded counting modes, with neither CH1 nor CH2 inputs inverted in the example.

Figure 16-29 Example of counting for each quadrature encoder mode



The encoder mode provides information about the current position of the sensor. The counter automatically counts up or down based on the pulses output from the encoder, and the current counter value indicates the current position of the encoder. Using another timer configured in capture mode, the rising or falling edge of the encoder signal can be captured and its timestamp recorded. By measuring the time interval between the two events, the corresponding dynamic information such as speed, acceleration and deceleration can be calculated; The encoder output indicating the mechanical zero position can be used for this purpose, and by measuring the time interval from the mechanical zero position to some other position, the change in position and velocity relative to the zero position can be calculated.

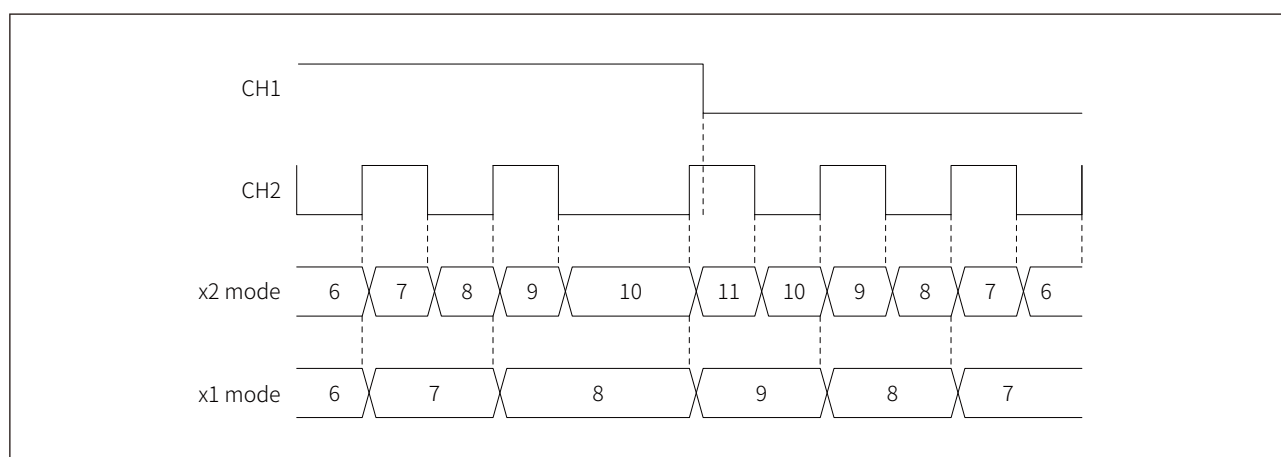
### 16.3.2.9 Clock + direction encoder mode

The timer also supports clock plus direction encoder mode, which requires configuring the SMS bit field of the slave mode control register GTIMx\_SMCR to be 0xA or 0xB, which correspond to x2 mode and x1 mode respectively. The functional block diagram can be referred to [Figure 16-27: Encoder mode block diagram](#), which allows filtering and polarity control of the input signal.

Encoders with direction usually have two output signals: a clock signal and a direction signal. Using the clock signal, the rotational speed and position change of the rotary encoder can be calculated, while the direction signal can be used to determine the direction of rotation. In x2 mode, the counter counts at the rising and falling edge of the clock signal; In x1 mode, the counter counts on a single clock edge based on the value of the CC2P bit field, where CC2P 0 corresponds to rising edge sensitivity and CC2P 1 corresponds to falling edge sensitivity. The polarity of the direction signal on the CH1 channel is set by the CC1P bit: CC1P is 0 corresponding to positive polarity (counts up when CH1 is high, counts down when CH1 is low) and CC1P is 1 corresponding to negative polarity (counts up when CH1 is low, counts down when CH1 is high).

The following figure shows an example of counter counting in clock plus direction encoder mode, with neither CH1 nor CH2 inputs inverted in the example:

Figure 16-30 Example of clock + direction encoder mode counting



### 16.3.2.10 Directional clock encoder mode

The timer also supports directional clock encoder mode, which requires configuring the SMS bit field of the slave mode control register GTIMx\_SMCR to be 0xC or 0xD, corresponding to x2 mode or x1 mode, respectively. The functional block diagram can be referred to [Figure 16-27: Encoder mode block diagram](#), which allows filtering and polarity control of the input signal.

The directional clock encoder provides an up count clock line and a down count clock line, respectively, depending on the direction of rotation. In x2 mode, the counter counts on the rising and falling edges of either of the two clock lines; the CC1P and CC2P bits encode the clock idle state, with CCyP being 0 corresponding to high level idle and CCyP being 1 corresponding to low level idle. In x1 mode, the counter counts on a single clock edge based on the value of the CC1P and CC2P bit fields, with CCyP being 0 corresponding to falling-edge sensitive and high-level idle, and CCyP being 1 corresponding to rising-edge sensitive and low-level idle.

The following table shows the relationship between the counting direction and the encoder signal in different modes:



Table 16-6 Relationship between counting direction and encoder signal

| Mode                | SMS  | Signal level |        | TI1FP1     |            | TI2FP2   |          |
|---------------------|------|--------------|--------|------------|------------|----------|----------|
|                     |      | TI2FP2       | TI1FP1 | Rising     | Falling    | Rising   | Falling  |
| x2 mode<br>CCyP = 0 | 1100 | High         | High   | Down count | Down count | Up count | Up count |
|                     |      | Low          | Low    | No count   | No count   | No count | No count |
| x2 mode<br>CCyP = 1 |      | High         | High   | No count   | No count   | No count | No count |
|                     |      | Low          | Low    | Down count | Down count | Up count | Up count |
| x1 mode<br>CCyP = 0 | 1101 | High         | High   | No count   | Down count | No count | Up count |
|                     |      | Low          | Low    | No count   | No count   | No count | No count |
| x1 mode<br>CCyP = 1 |      | High         | High   | No count   | No count   | No count | No count |
|                     |      | Low          | Low    | Down count | No count   | Up count | No count |

The following figure shows an example of counting in directional clock encoder mode:

Figure 16-31 Example of directional clock encoder mode counting (CC1P = CC2P = 0)

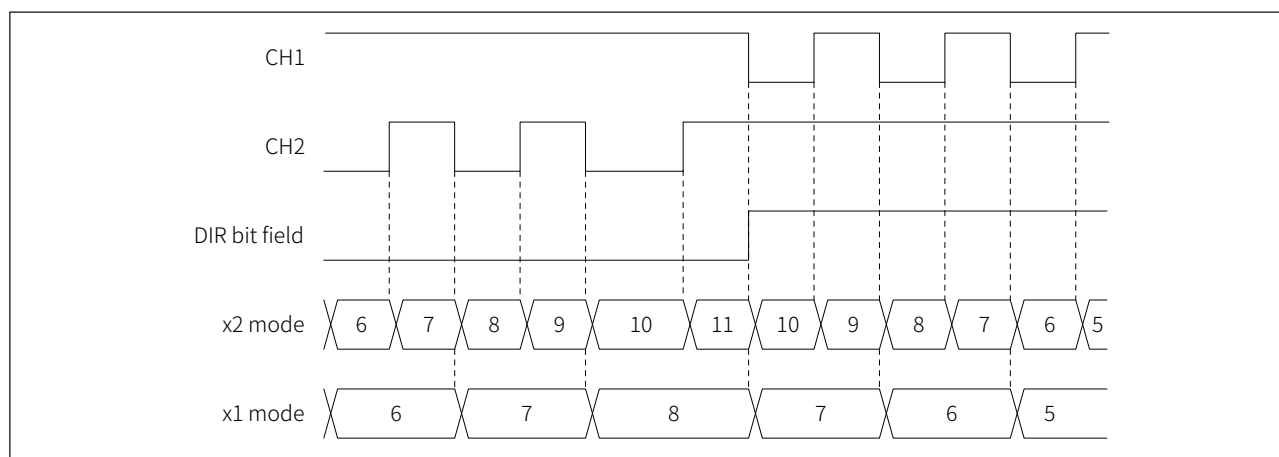
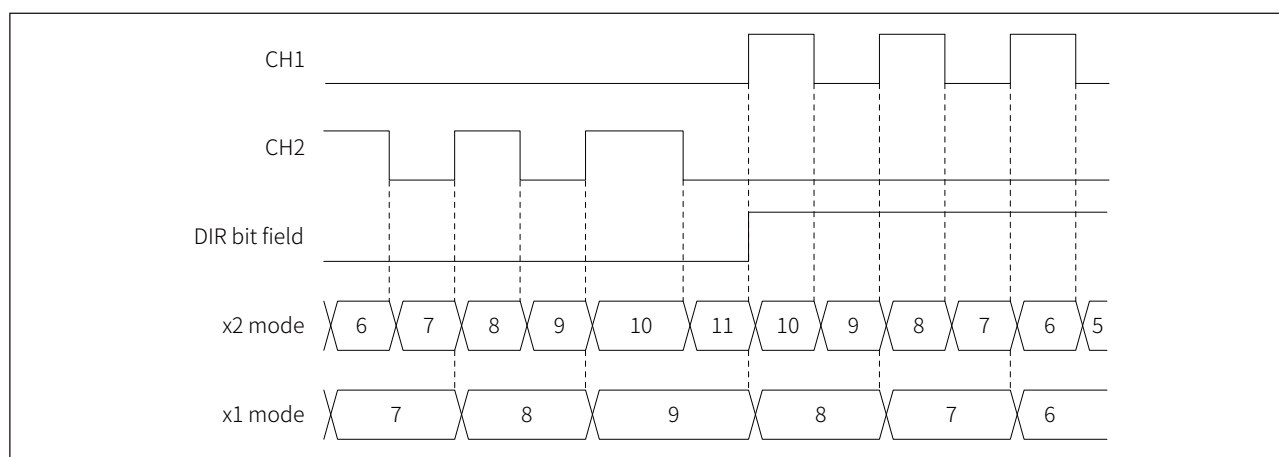


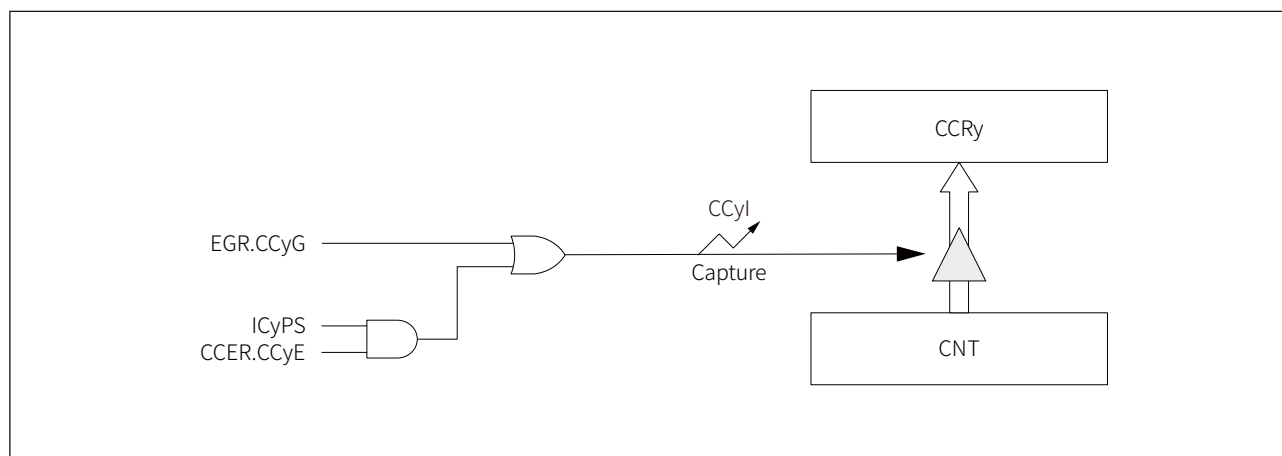
Figure 16-32 Example of directional clock encoder mode counting (CC1P = CC2P = 1)



### 16.3.3 Input capture function

GTIM supports the input capture function. When the CCyS bit field of the capture mode registers (GTIMx\_CCMR1CAP and GTIMx\_CCMR2CAP) is set to a non-zero value, the CCy channel is configured as an input, and at the same time, the input mapping of the corresponding capture channel ICy is specified. Input capture is supported to be triggered by software or hardware, and its function block diagram is shown below:

Figure 16-33 Input capture mode block diagram



**Software trigger:** When GTIMx\_EGR.CCyG is set to 1, a capture will be triggered by software immediately, and the value of the current counter CNT is latched into the capture/comparison register CCRy of the corresponding channel to complete a capture, and the CCyG bit is cleared by the hardware automatically after the completion of the capture.

**Hardware trigger:** When a valid edge is detected on the input channel TIy, the value of the current counter CNT is latched into the capture/comparison register CCRy of the corresponding channel to complete a capture. The valid edge that triggers the capture is configured through the CCyNP and CCyP bit fields of the capture/compare enable register GTIMx\_CCER as shown in the table below:

Table 16-7 Input capture mode configuration

| GTIMx_CCER register   |                      | Capture mode configuration           |
|-----------------------|----------------------|--------------------------------------|
| CCyNP bit field value | CCyP bit field value |                                      |
| 0                     | 0                    | Rising edge capture                  |
| 0                     | 1                    | Falling edge capture                 |
| 1                     | 1                    | Both rising and falling edge capture |

When a capture occurs, the capture/compare interrupt flag GTIMx\_ISR.CCyIF of the corresponding channel is set to 1 by hardware and:

- If interrupt is enabled (setting GTIMx\_DIER.CCyIE to 1), an interrupt request is generated.
- If DMA is enabled (setting GTIMx\_DIER.CCyDE to 1, and GTIMx\_CR2.CCDS to 0), a DMA request is generated.
- If the GTIMx\_ISR.CCyIF flag bit is already high when the capture event occurs, the repeat capture flag bit GTIMx\_ISR.CCyOF will be set to 1 by hardware.
- Setting GTIMx\_ICR.CCyIF to 0 or reading capture register CCRy clears the GTIMx\_ISR.CCyIF flag bit, and setting GTIMx\_ICR.CCyOF to 0 clears the GTIMx\_ISR.CCyOF flag bit.

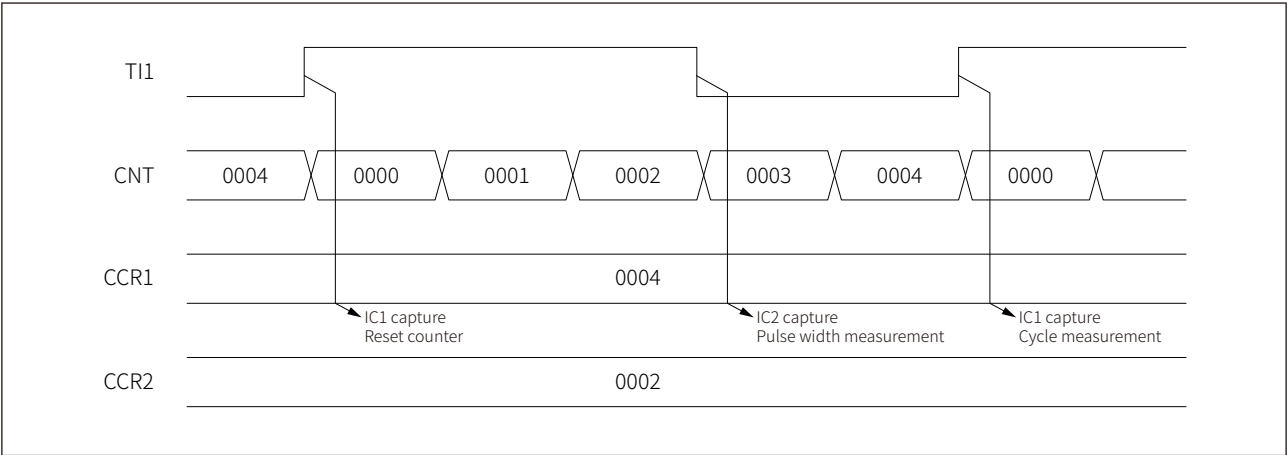
16.3.3.1 PWM input mode

PWM input mode is an application of input capture mode. The input capture function supports the same channel input signal to be captured by two capture registers. Combined with the slave mode reset function, the period and pulse width of the PWM input signal can be easily measured.

The PWM input mode can only be used with the TI1 and TI2 signals because only TI1FP1 and TI2FP2 are connected to the slave mode controller.

The following figure shows an example of the PWM signal timing when it is input from the TI1 channel:

Figure 16-34 PWM input mode timing



### 16.3.3.2 Input capture source

The GTIM input capture source can be the external GTIMx\_CHy pin or other peripherals on-chip, configured through the TlySEL bit field of the TI input select register GTIMx\_TISEL.

When GTIMx\_TISEL.TlySEL is 0x0, the source of the Tly channel input capture signal is the external GTIMx\_CHy pin, in which case the corresponding pin needs to be configured as a multiplexing function through the GPIO multiplexing function registers (GPIOx\_AFRH and GPIOx\_AFLR).

When GTIMx\_TISEL.TlySEL is 0x1~0xF, the Tly channel input capture signals come from other peripherals on-chip, as shown in the following table:

Table 16-8 General-purpose timer input capture source

| TlySEL bit field value | Input capture signal source for each channel |               |               |               |
|------------------------|----------------------------------------------|---------------|---------------|---------------|
|                        | Channel TI1                                  | Channel TI2   | Channel TI3   | Channel TI4   |
| 0000                   | GTIMx_CH1 pin                                | GTIMx_CH2 pin | GTIMx_CH3 pin | GTIMx_CH4 pin |
| 0001                   | VC1_OUT                                      | VC3_OUT       | VC1_OUT       | VC3_OUT       |
| 0010                   | VC2_OUT                                      | VC4_OUT       | VC2_OUT       | VC4_OUT       |
| 0011                   | UART1_RXD                                    | UART1_TXD     | UART1_RXD     | UART1_TXD     |
| 0100                   | UART2_RXD                                    | UART2_TXD     | UART2_RXD     | UART2_TXD     |
| 0101                   | UART3_RXD                                    | UART3_TXD     | UART3_RXD     | UART3_TXD     |
| 0110                   | MCO_OUT                                      | RTC_OUT       | LPTIM_PWM     | LPTIM_PWM     |
| 0111                   | HALLTIM_Trgo                                 | HALLTIM_Trgo  | HALLTIM_Trgo  | HALLTIM_Trgo  |
| 1000                   | BTIM1_Trgo                                   | BTIM1_Trgo    | BTIM1_Trgo    | BTIM1_Trgo    |
| 1001                   | BTIM2_Trgo                                   | BTIM2_Trgo    | BTIM2_Trgo    | BTIM2_Trgo    |
| 1010                   | BTIM3_Trgo                                   | BTIM3_Trgo    | BTIM3_Trgo    | BTIM3_Trgo    |
| 1011                   | GTIM1_Trgo                                   | GTIM1_Trgo    | GTIM1_Trgo    | GTIM1_Trgo    |
| 1100                   | GTIM2_Trgo                                   | GTIM2_Trgo    | GTIM2_Trgo    | GTIM2_Trgo    |
| 1101                   | GTIM3_Trgo                                   | GTIM3_Trgo    | GTIM3_Trgo    | GTIM3_Trgo    |
| 1110                   | GTIM4_Trgo                                   | GTIM4_Trgo    | GTIM4_Trgo    | GTIM4_Trgo    |
| 1111                   | ATIM_Trgo                                    | ATIM_Trgo     | ATIM_Trgo     | ATIM_Trgo     |

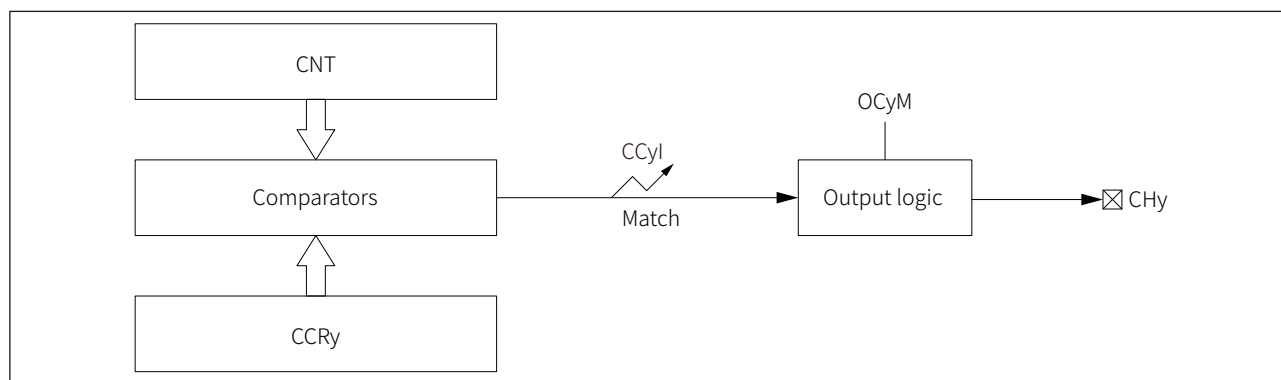
*Attention: Prohibits the selection of Trgo signal output by GTIMx itself.*



### 16.3.4 Output compare function

The GTIM supports the output compare function. Setting the CCyS bit field of the compare mode registers (GTIMx\_CCMR1CMP and GTIMx\_CCMR2CMP) to 0 configures the CHy channel as an output. In the output compare mode, the value of the current counter CNT is compared with the value of the capture/compare register CCRy of the corresponding channel CHy. When the two match, the compare channel CHy outputs to a settable level state, and at the same time generates a compare interrupt. The following diagram shows the functional block diagram:

Figure 16-35 Output compare mode block diagram



The output action of the CHy pin is set by the OCyM bit field of the compare mode registers (GTIMx\_CCMR1CMP and GTIMx\_CCMR2CMP) as shown in the table below:

Table 16-9 Output compare mode configuration

| OCyM bit field value | Compare mode configuration                                    |
|----------------------|---------------------------------------------------------------|
| 0000                 | When comparing and matching, OCyREF keeps the original level. |
| 0001                 | When comparing and matching, OCyREF is set to 1.              |
| 0010                 | When comparing and matching, OCyREF is set to 0.              |
| 0011                 | When comparing and matching, OCyREF flips.                    |
| 0100                 | Force OCyREF to low level.                                    |
| 0101                 | Force OCyREF to high level.                                   |
| 0110                 | PWM mode 1.                                                   |
| 0111                 | PWM mode 2.                                                   |
| 1000                 | Retriggerable OPM mode 1.                                     |
| 1001                 | Retriggerable OPM mode 2.                                     |
| 1011                 | Count direction output.                                       |
| 1100                 | Combined PWM mode 1.                                          |
| 1101                 | Combined PWM mode 2.                                          |
| 1110                 | Asymmetric PWM mode 1.                                        |
| 1111                 | Asymmetric PWM mode 2.                                        |

When a compare match occurs, the capture/compare interrupt flag GTIMx\_ISR.CCyIF of the corresponding channel is set to 1 by hardware, while:



- If interrupt is enabled (setting GTIMx\_DIER.CCylE to 1), an interrupt request will be generated.
- If DMA is enabled (setting GTIMx\_DIER.CCyDE to 1, and GTIMx\_CR2.CCDS to 0), an DMA request will be generated.
- Setting GTIMx\_ICR.CCyIF to 0 clears the GTIMx\_ISR.CCyIF flag bit.

The capture/compare register GTIMx\_CCRy has a cache function and the use of the cache function is selected through the OCyPE bit field. When OCyPE is 0, the comparison cache function of channel CHy is disabled, and the GTIMx\_CCRy register can be updated by software at any time; the updated value takes effect immediately and affects the output waveform; When OCyPE is 1, the comparison cache function of channel CHy is enabled, and updating the GTIMx\_CCRy register will not take effect immediately, and the value of GTIMx\_CCRy register will be updated to the valid register only when the update event UEV occurs.



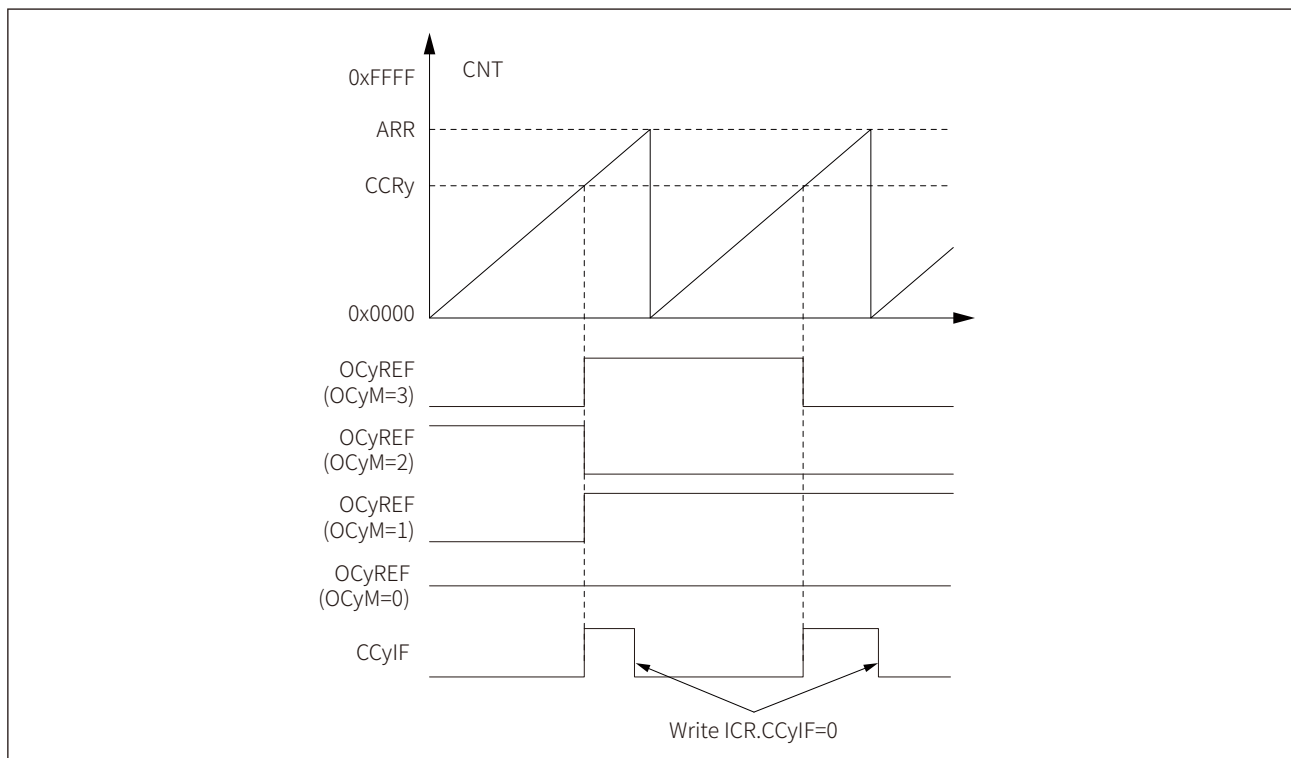
#### 16.3.4.1 Match output

The match output mode is used to control the output waveform or to indicate that a certain time period has passed.

- Set the OCyM bit field to 0x0, when comparing and matching, the reference signal OCyREF keeps its level while the GTIMx\_ISR.CCyIF flag bit field is set to 1.
- Set the OCyM bit field to 0x1, when comparing and matching, the reference signal OCyREF is set to high level while the GTIMx\_ISR.CCyIF flag bit field is set to 1.
- Set the OCyM bit field to 0x2, when comparing and matching, the reference signal OCyREF is set to low level while the GTIMx\_ISR.CCyIF flag bit field is set to 1.
- Set the OCyM bit field to 0x3, when comparing and matching, the reference signal OCyREF is flipped while the GTIMx\_ISR.CCyIF flag bit field is set to 1.

The following figure shows an example of the timing for each compare-match output mode, with the example being the edge-aligned incremental mode:

Figure 16-36 Compare match output mode



### 16.3.4.2 Forced output

In forced output mode, the output comparison signal can be forced to a high or low state directly by software without relying on the comparison result of the capture/compare register GTIMx\_CCRy and the count register GTIMx\_CNT.

Set the OCyM bit field of the compare mode registers (GTIMx\_CCMR1CMP and GTIMx\_CCMR2CMP) to 0x4 to force the reference signal OCyREF to a low level; set the OCyM bit field to 0x5 to force the reference signal OCyREF to a high level.

In forced output mode, the comparison between the GTIMx\_CCRy register and counter GTIMx\_CNT is still in progress, and the corresponding flags are set to 1. The corresponding interrupt/DMA requests are also generated.

### 16.3.4.3 PWM output

Pulse width modulation (PWM) mode can generate a signal with a frequency determined by the reload register GTIMx\_ARR and a duty cycle determined by the capture/compare register GTIMx\_CCRy.

Writing 0x6 (PWM mode 1) or 0x7 (PWM mode 2) to the OCyM bit field of the compare mode registers (GTIMx\_CCMR1CMP and GTIMx\_CCMR2CMP) can independently set each CHy output channel to generate a PWM all the way through the output states as shown in the table below:

Table 16-10 PWM mode 1/2 output state

| Operating mode                           | Count direction | PWM mode 1                  | PWM mode 2                  |
|------------------------------------------|-----------------|-----------------------------|-----------------------------|
| Edge-aligned mode<br>Center-aligned mode | Up count        | CNT < CCRy, OCyREF is high. | CNT < CCRy, OCyREF is low.  |
|                                          | Down count      | CNT > CCRy, OCyREF is low.  | CNT > CCRy, OCyREF is high. |

The following are examples of waveforms for each PWM mode in different count modes, where GTIMx\_ARR is 0x08:





Figure 16-37 PWM mode 1, edge-aligned mode, count up

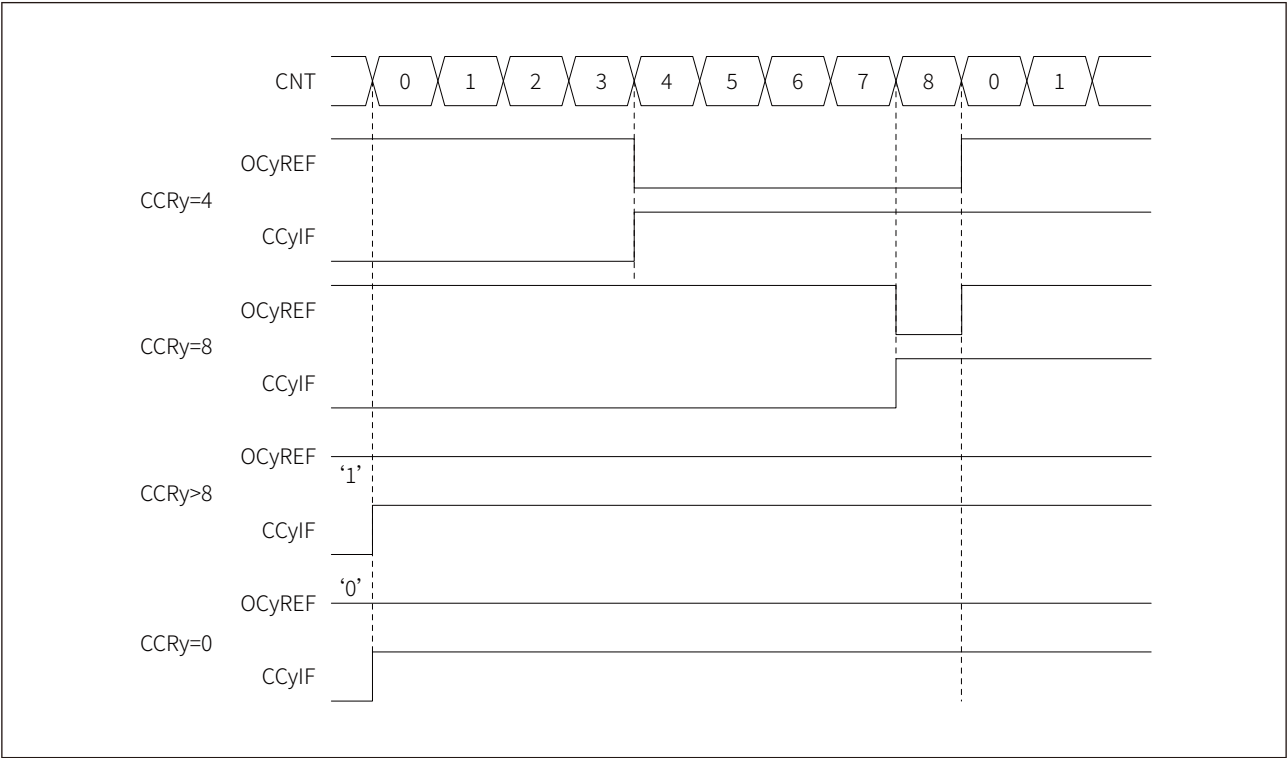


Figure 16-38 PWM mode 1, edge-aligned mode, count down

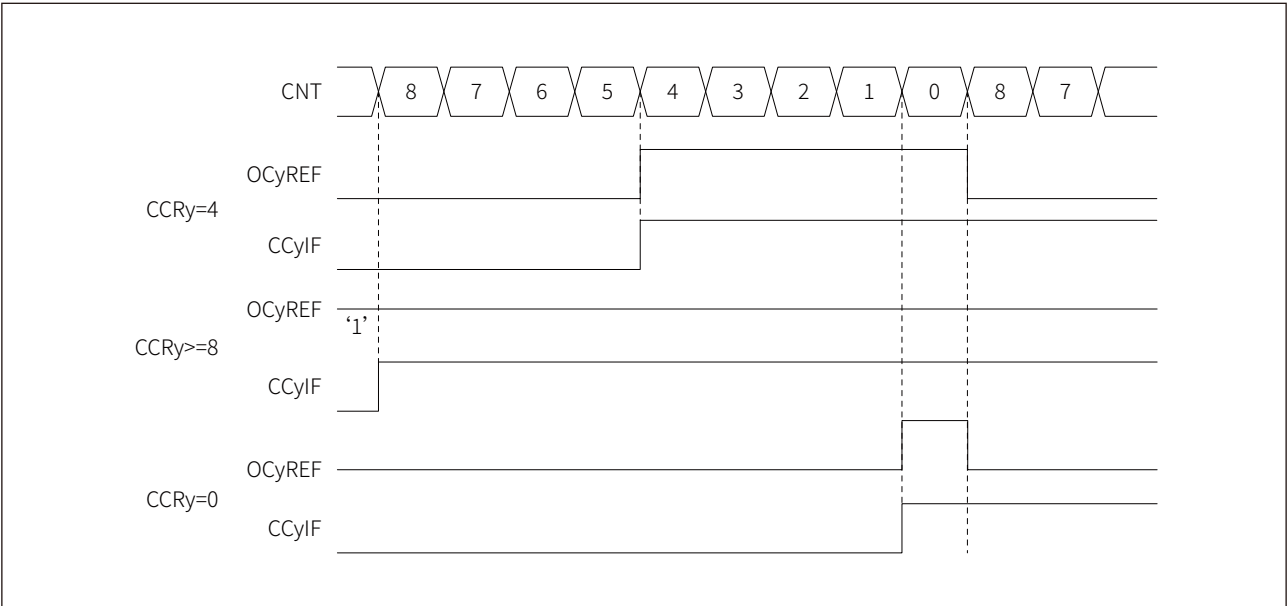
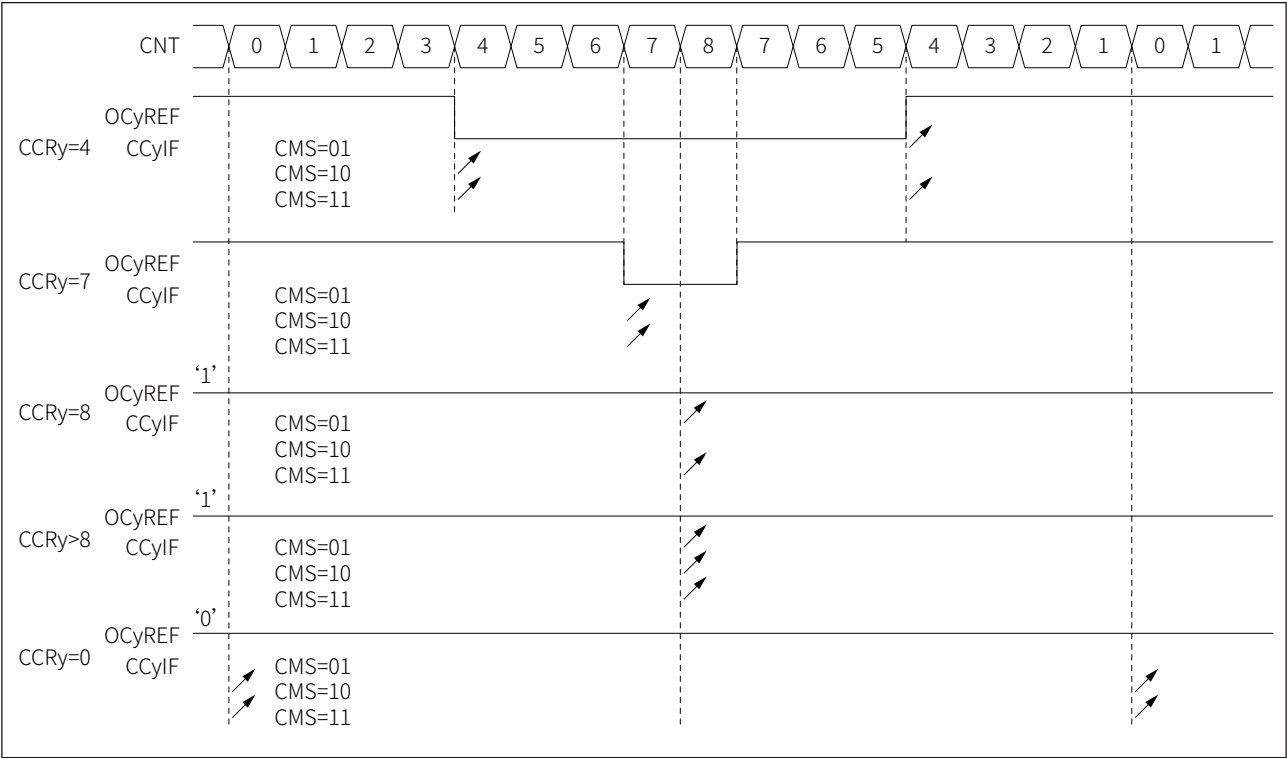


Figure 16-39 PWM mode 1, center-aligned mode



### 16.3.4.4 Asymmetric PWM output

The asymmetric PWM output is relative to the symmetric PWM output when counting in center-aligned mode. In the asymmetric PWM output mode, two PWM signals with programmable phase shifts are allowed to be generated in centre-aligned mode, with the frequency determined by the reload register GTIMx\_ARR, and the duty cycle and phase shifts determined by the two capture/comparison registers GTIMx\_CCRy, where one of the CCRy registers controls the PWM during counting up, and the other CCRy register controls the PWM during counting down, the details are shown below:

- OC1REFC is controlled by CCR1 (count up) and CCR2 (count down).
- OC2REFC is controlled by CCR2 (count up) and CCR1 (count down).
- OC3REFC is controlled by CCR3 (count up) and CCR4 (count down).
- OC4REFC is controlled by CCR4 (count up) and CCR3 (count down).

Writing 0xE (asymmetrical PWM mode 1) or 0xF (asymmetrical PWM mode 2) to the OCyM bit field of the compare mode registers (GTIMx\_CCMR1CMP and GTIMx\_CCMR2CMP) can independently set each CHy output channel to generate asymmetrical PWM.

When the channel is set to be used as an asymmetric PWM output channel, the other auxiliary channel can still be used. For example, when channel CH1 is configured to generate an OC1REFC signal in asymmetric PWM mode 1, channel CH2 can still be configured to output an OC2REF signal in PWM mode 2 or to output an OC2REFC signal in asymmetric PWM mode 2.

The following figure shows an example of a waveform for asymmetric PWM mode 1/2, where ARR = 8, CCR1 = 0, CCR2 = 8, CCR3 = 3, and CCR4 = 5:

Figure 16-40 Example of asymmetric PWM mode 1

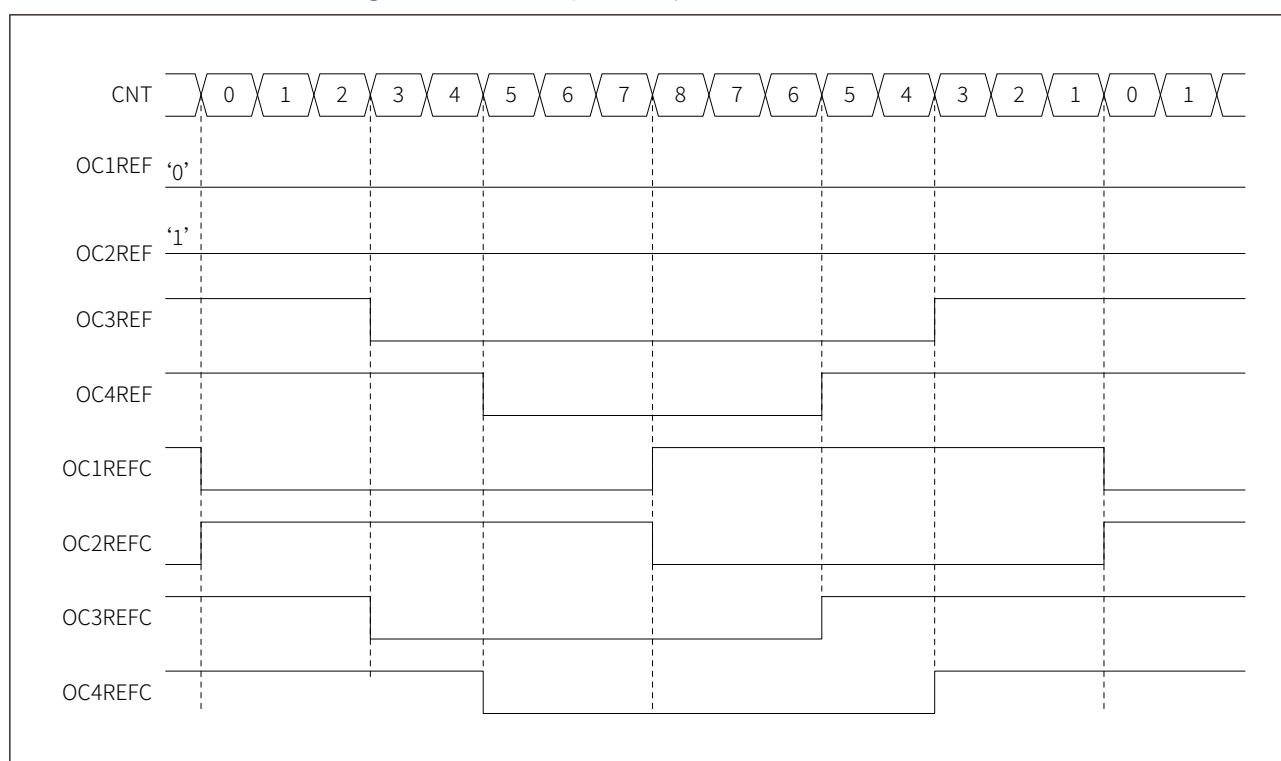
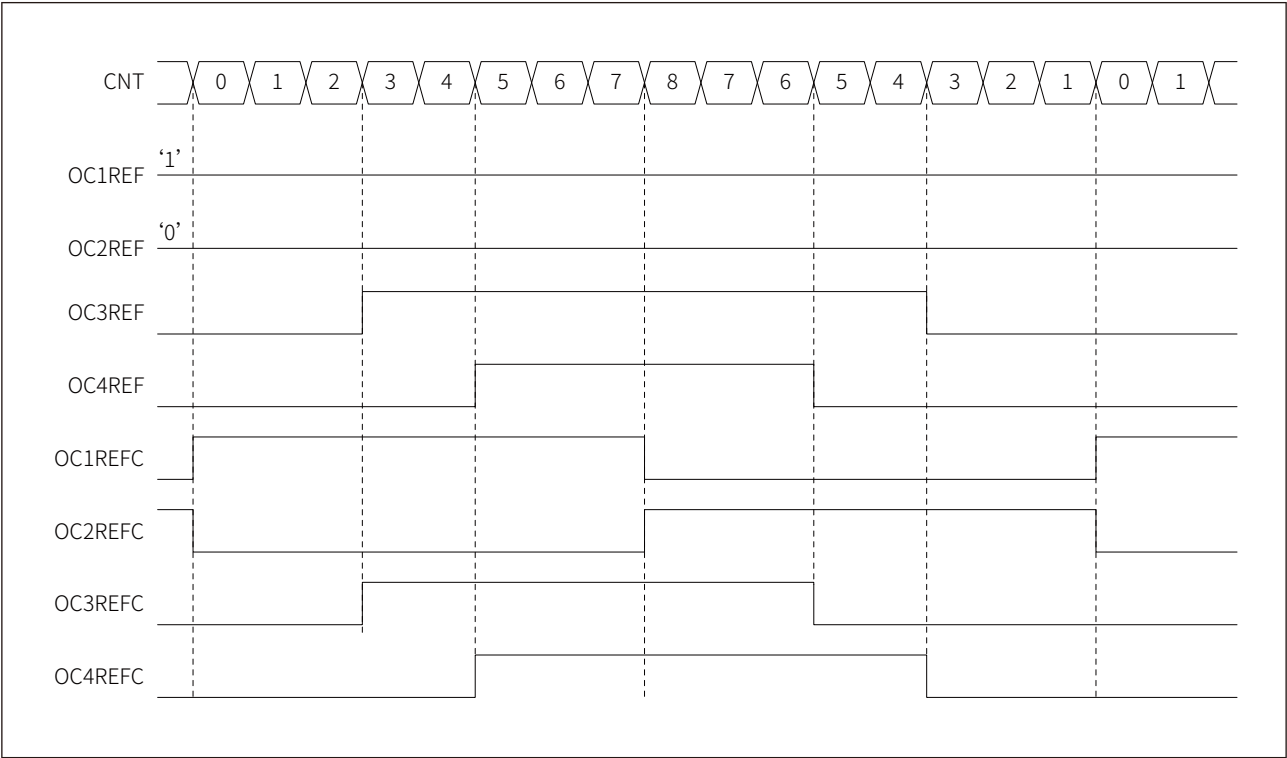


Figure 16-41 Example of asymmetric PWM mode 2



### 16.3.4.5 Combined PWM output

In combined PWM output mode, two PWM pulses with programmable delay and phase shift are allowed to be generated in either edge-aligned or center-aligned mode, with the frequency determined by the reload register GTIMx\_ARR, and the duty cycle and delay determined by the two capture/comparison registers GTIMx\_CCRy.

In this mode, the OCyREFC signal is generated by the logic OR operation or the combination of logic AND operation of the two reference signals OCyREF, and the combined PWM mode can be independently selected through the OCyM bit field of the comparison mode registers (GTIMx\_CCMR1CMP and GTIMx\_CCMR2CMP) as shown in the table below:

**Table 16-11 Combined PWM mode configuration**

| OCyM bit field value         | Output state                                                                                                                                                               |
|------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0xC<br>(Combined PWM mode 1) | OC1REFC (or OC2REFC) is generated by the logical OR operation of OC1REF and OC2REF.<br>OC3REFC (or OC4REFC) is generated by the logical OR operation of OC3REF and OC4REF. |
| 0xD<br>(Combined PWM mode 1) | OC1REFC (or OC2REFC) is generated by the logical AND operation of OC1REF and OC2REF.<br>OC3REFC (or OC4REFC) is generated by the logic AND operation of OC3REF and OC4REF. |

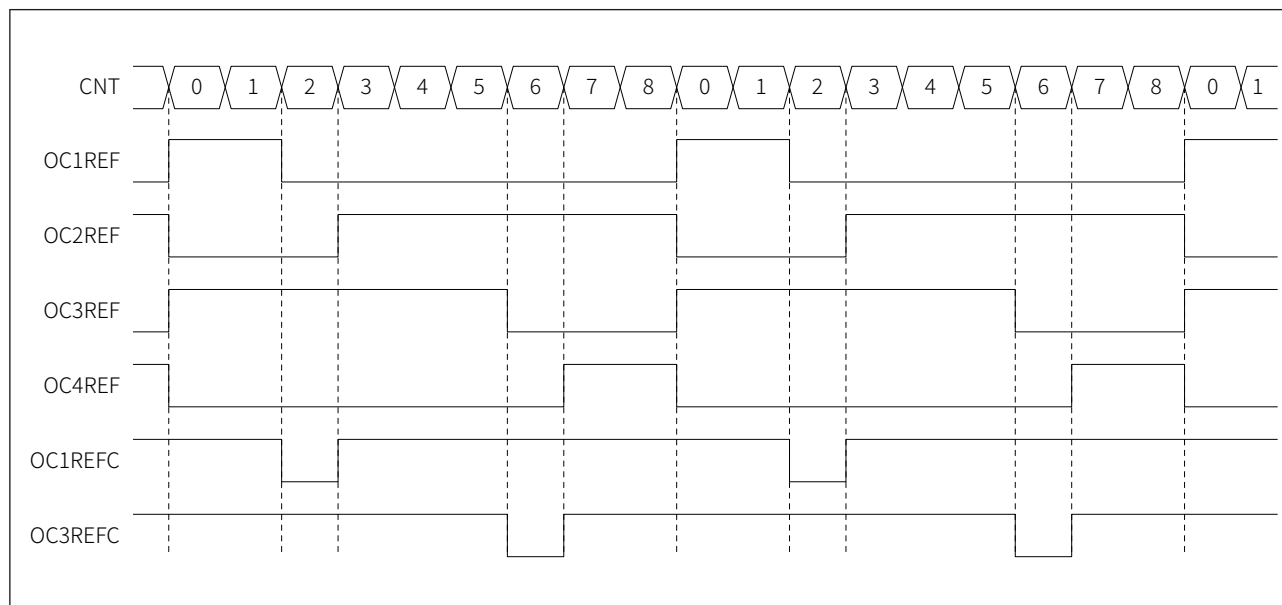
When the set channel is used as a combined PWM output channel, the other auxiliary channel can still be used, but it must be set to the opposite PWM mode. For example, when channel CH1 is configured to generate the OC1REFC signal in combined PWM mode 1, channel CH2 can still be used, and CH2 can be configured to output the OC2REF signal in PWM mode 2.

The following figure shows an example of the signals for the combined PWM mode 1 in the edge-aligned count up mode, which is configured as follows:

- The reload value ARR is 8.
- Channel 1 is configured as combined PWM mode 1, CCR1 = 2.
- Channel 2 is configured as PWM mode 2, CCR2 = 3.
- Channel 3 is configured as combined PWM mode 1, CCR3 = 6.
- Channel 4 is configured as PWM mode 2, CCR4 = 7.



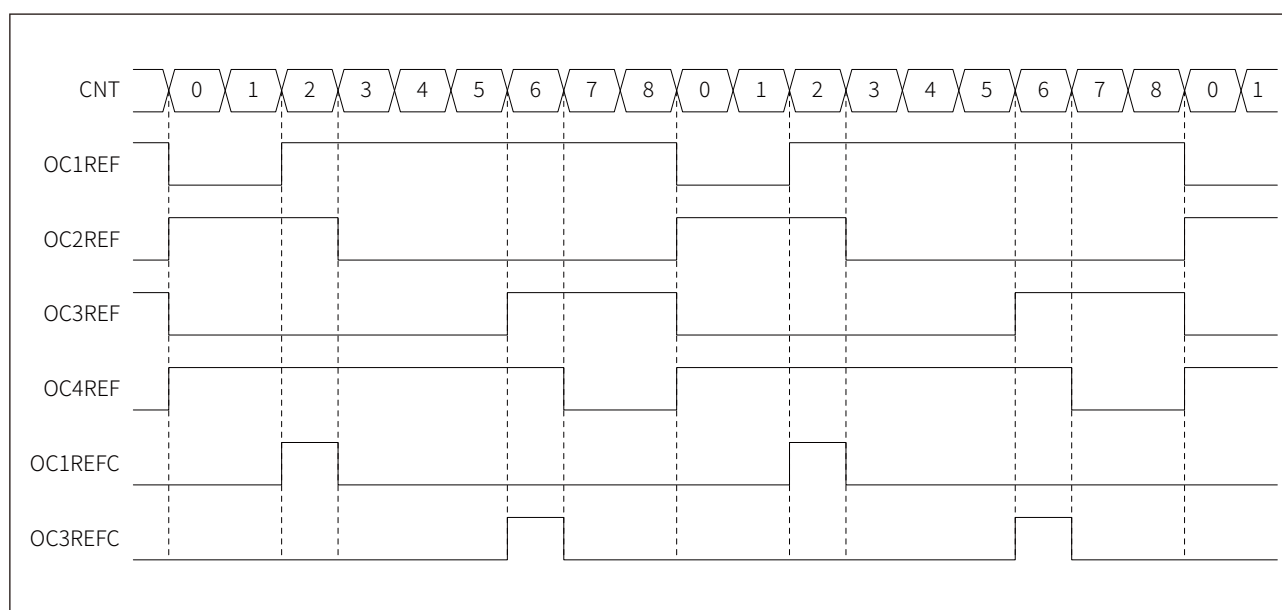
Figure 16-42 Example of combined PWM mode 1



The following figure shows an example of the signals for the combined PWM mode 2 in the edge-aligned count up mode, which is configured as follows:

- The reload value ARR is 8.
- Channel 1 is configured as combined PWM mode 2, CCR1 = 2.
- Channel 2 is configured as PWM mode 1, CCR2 = 3.
- Channel 3 is configured as combined PWM mode 2, CCR3 = 6.
- Channel 4 is configured as PWM mode 1, CCR4 = 7.

Figure 16-43 Example of combined PWM mode 2



### 16.3.4.6 Retriggerable single pulse mode

The retriggerable single pulse mode, in conjunction with the combined reset+trigger mode (SMS = 8), allows the counter to be reset and activated by a trigger signal (TRGI) and generates pulses of programmable length, with the pulse width determined by the ARR.

The differences compared to the single pulse mode are as follows:

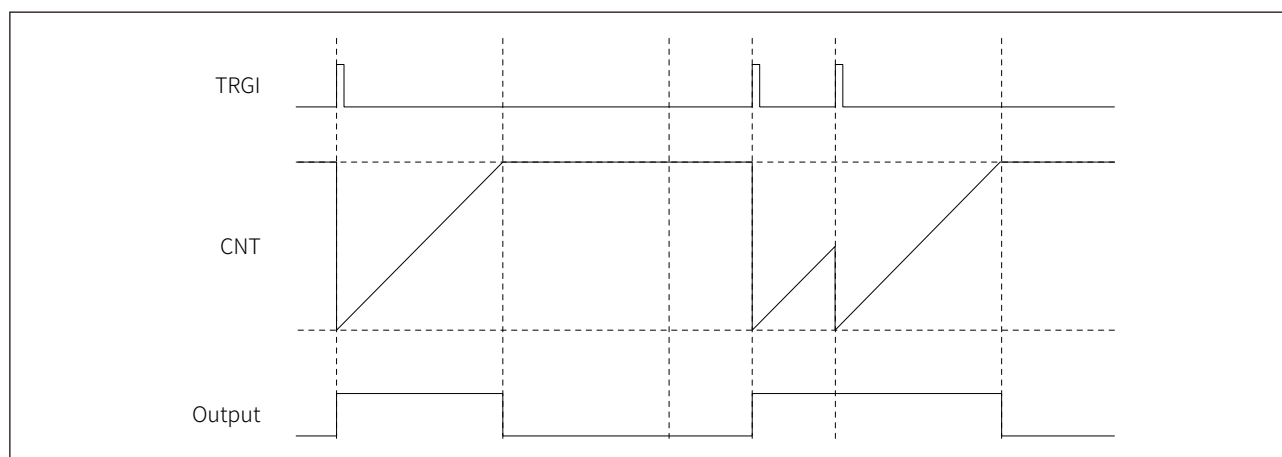
Table 16-12 Retriggerable single pulse mode and single pulse mode

| Single pulse mode                                                                                              | Retriggerable single pulse mode                                                                                        |
|----------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------|
| When a trigger occurs, a single pulse with a programmable pulse width is generated after a programmable delay. | When a trigger occurs, the pulse is generated immediately without programmable delay.                                  |
| The new trigger is invalid.                                                                                    | If the pulse generated by the previous trigger is not completed and a new trigger occurs, the pulse will be prolonged. |

In this mode, the counter can only be set to edge-aligned mode, not center-aligned mode. When configured for counting up mode, the corresponding CCRy must be set to 0. When configured for counting down mode, the corresponding CCRy must be greater than or equal to ARR.

The following figure shows an example of retriggerable single pulse mode 2:

Figure 16-44 Example of retriggerable single pulse mode 2



### 16.3.4.7 Count direction output

Set the OCyM bit field of the GTIMx\_CCMRxCMP register to 0xB, which outputs a count direction signal (a copy of the DIR bit in the GTIMx\_CR1 register) on the corresponding CHy channel.

This function can be used to monitor the counting direction (or direction of rotation) in encoder mode, or to indicate the up/down phase signals in center-aligned PWM mode.

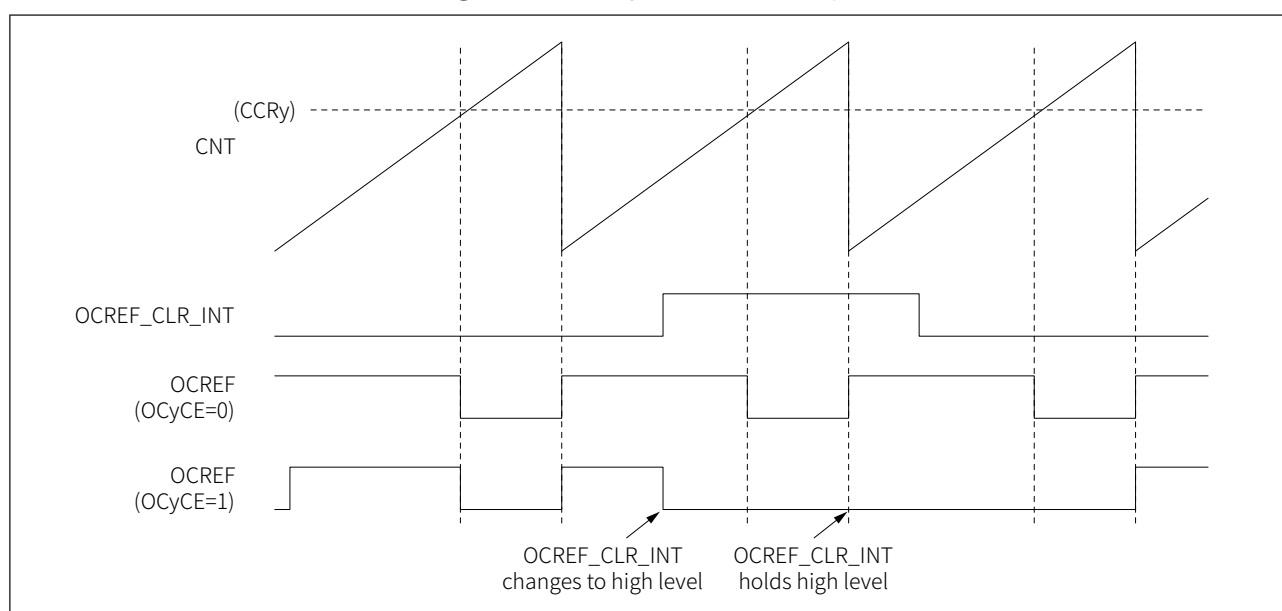
### 16.3.4.8 Reference signal clear function

Set the OCyCE bit field of the GTIMx\_CCMRxCMP register to 1, which enables the OCyREF signal clearing function of the corresponding channel, i.e., when a high level is detected on the OCREf\_CLR\_INT input, the OCyREF signal can be cleared to zero and the OCyREF signal will remain low until the next update event UEV occurs. This function is only available in output compare and PWM modes and is not available in forced output mode.

The clearing source OCREf\_CLR\_INT can be selected as either OCREf\_CLR or ETRF, which is selected by the OCCS bit field of the slave mode control register GTIMx\_SMCR, and there are also multiple input sources for OCREf\_CLR, see the description of the OCRSEL bit field in [16.10.23 GTIMx\\_AF2 Multiplexing function option register 2](#).

The following figure shows the OCyREF signal state when the OCREf\_CLR\_INT input goes high:

Figure 16-45 OCyREF clear example





### 16.3.5 UIF bit remap

Set the UIFREMAP bit in the GTIMx\_CR1 register to 1, which enables the UIF status bit remapping function, forcing the update interrupt flag GTIMx\_ISR.UIF to be copied consecutively into the UIFCPY bit field of the GTIMx\_CNT register. This allows automatically reading the counter value and the potential flip-flop condition issued by the UIFCPY flag. This simplifies the calculation of angular velocity by avoiding competing conditions when processing is shared between background tasks (counter reads) and interrupts (update interrupts).

There is no delay between UIF and UIFCPY flag enable.

When GTIMx\_CR1.UIFREMAP is 0, the UIFCPY bit field is reserved and read as 0.



### 16.3.6 Timer cascade ITR

Timer cascading can be achieved by using the ITR signal, i.e., connecting the trigger output (TRGO) of the master timer to the ITR input of the slave timer, which in turn is connected to the trigger input (TRGI) of the slave timer. The ITR cascade can perform a variety of functions, such as using one timer as a prescaler for another timer, or the master timer performing a reset, enabling, or providing a clock to the slave timer's counter.

The source of the GTIM's cascade input ITR is the trigger output signals (TRGO) of the BTIM, GTIM, HALLTIM and ATIM, and other peripheral signals, which can be selected by the TS bit field of the slave mode control register GTIMx\_SMCR, as configured in the table below:

Table 16-13 ITR signal source

| GTIMx_SMCR.TS<br>bit field value | ITR signal name | ITR signal source |               |               |               |
|----------------------------------|-----------------|-------------------|---------------|---------------|---------------|
|                                  |                 | GTIM1             | GTIM2         | GTIM3         | GTIM4         |
| 00000                            | ITR0            | ATIM_Trigo        | ATIM_Trigo    | ATIM_Trigo    | ATIM_Trigo    |
| 00001                            | ITR1            | GTIM1_Trigo       | GTIM1_Trigo   | GTIM1_Trigo   | GTIM1_Trigo   |
| 00010                            | ITR2            | GTIM2_Trigo       | GTIM2_Trigo   | GTIM2_Trigo   | GTIM2_Trigo   |
| 00011                            | ITR3            | GTIM3_Trigo       | GTIM3_Trigo   | GTIM3_Trigo   | GTIM3_Trigo   |
| 01000                            | ITR4            | GTIM4_Trigo       | GTIM4_Trigo   | GTIM4_Trigo   | GTIM4_Trigo   |
| 01001                            | ITR5            | BTIM1_Trigo       | BTIM1_Trigo   | BTIM1_Trigo   | BTIM1_Trigo   |
| 01010                            | ITR6            | BTIM2_Trigo       | BTIM2_Trigo   | BTIM2_Trigo   | BTIM2_Trigo   |
| 01011                            | ITR7            | BTIM3_Trigo       | BTIM3_Trigo   | BTIM3_Trigo   | BTIM3_Trigo   |
| 01100                            | ITR8            | HALLTIM_Trigo     | HALLTIM_Trigo | HALLTIM_Trigo | HALLTIM_Trigo |
| 01101                            | ITR9            | UART1_TXD         | UART1_TXD     | UART1_TXD     | UART1_TXD     |
| 01110                            | ITR10           | UART2_TXD         | UART2_TXD     | UART2_TXD     | UART2_TXD     |
| 01111                            | ITR11           | UART3_TXD         | UART3_TXD     | UART3_TXD     | UART3_TXD     |
| 10000                            | ITR12           | UART1_RXD         | UART1_RXD     | UART1_RXD     | UART1_RXD     |
| 10001                            | ITR13           | UART2_RXD         | UART2_RXD     | UART2_RXD     | UART2_RXD     |
| 10010                            | ITR14           | UART3_RXD         | UART3_RXD     | UART3_RXD     | UART3_RXD     |
| 10011                            | ITR15           | MCO_OUT           | LPTIM_OV      | HSE_FAULT     | LSE_FAULT     |

**Attention:** Prohibits the selection of Trgo signal output by GTIMx itself.

By controlling the MMS bit field of register GTIMx\_CR2, it is possible to select the message (TRGO) that will be sent to the slave timer for synchronisation in GTIM master mode, connected to the ITR input of the other timer. The following table shows this:



Table 16-14 GTIM trigger outputs (TRGO)

| GTIMx_CR2.MMS | TRGO signal                     |
|---------------|---------------------------------|
| 0000          | Reset                           |
| 0001          | Counter enable signal CNT_EN    |
| 0010          | Update event                    |
| 0011          | Channel 1 compare capture pulse |
| 0100          | Channel 2 compare capture pulse |
| 0101          | Channel 3 compare capture pulse |
| 0110          | Channel 4 compare capture pulse |
| 0111          | OC1REF signal                   |
| 1000          | OC2REF signal                   |
| 1001          | OC3REF signal                   |
| 1010          | OC4REF signal                   |
| 1011          | OC1REFC signal                  |
| 1100          | OC2REFC signal                  |
| 1101          | OC3REFC signal                  |
| 1110          | OC4REFC signal                  |
| 1111          | Encoder clock output            |



### 16.3.7 On-chip peripheral interconnect ETR

The source of the GTIM's ETR signal can be the external GTIMx\_ETR pin or other peripherals on-chip, which can be selected through the ETRSEL bit field of the multiplexing function option register GTIMx\_AF1.

When the ETR signal is input from the GTIMx\_ETR pin, the specific external input port can be referred to [Table 9-2: GPIO multiplexing function assignment table](#), and needs to be multiplexed and configured through the GPIO multiplexing function registers (GPIOx\_AFRH and GPIOx\_AFRL); when the ETR signal is input from other peripheral devices on-chip, it can realise the on-chip and off-chip peripheral interconnection.

The following table shows the sources of ETR signals:

Table 16-15 ETR signal source

| AF1.ETRSEL bit field value | ETR signal source |           |           |           |
|----------------------------|-------------------|-----------|-----------|-----------|
|                            | GTIM1             | GTIM2     | GTIM3     | GTIM4     |
| 0000                       | GTIM1_ETR         | GTIM2_ETR | GTIM3_ETR | GTIM4_ETR |
| 0001                       | VC1_OUT           | VC1_OUT   | VC1_OUT   | VC1_OUT   |
| 0010                       | VC2_OUT           | VC2_OUT   | VC2_OUT   | VC2_OUT   |
| 0011                       | VC3_OUT           | VC3_OUT   | VC3_OUT   | VC3_OUT   |
| 0100                       | VC4_OUT           | VC4_OUT   | VC4_OUT   | VC4_OUT   |
| 0101                       | ADC1_AWD          | ADC1_AWD  | ADC1_AWD  | ADC1_AWD  |
| 0110                       | ADC2_AWD          | ADC2_AWD  | ADC2_AWD  | ADC2_AWD  |
| 0111                       | LVD_OUT           | LVD_OUT   | LVD_OUT   | LVD_OUT   |
| 1000                       | BTIM1_ETR         | BTIM1_ETR | BTIM1_ETR | BTIM1_ETR |
| 1001                       | BTIM2_ETR         | BTIM2_ETR | BTIM2_ETR | BTIM2_ETR |
| 1010                       | BTIM3_ETR         | BTIM3_ETR | BTIM3_ETR | BTIM3_ETR |
| 1011                       | -                 | GTIM1_ETR | GTIM1_ETR | GTIM1_ETR |
| 1100                       | GTIM2_ETR         | -         | GTIM2_ETR | GTIM2_ETR |
| 1101                       | GTIM3_ETR         | GTIM3_ETR | -         | GTIM3_ETR |
| 1110                       | GTIM4_ETR         | GTIM4_ETR | GTIM4_ETR | -         |
| 1111                       | LSE_FAULT         | LSE_FAULT | HSE_FAULT | HSE_FAULT |



## 16.4 GTIM interrupt

GTIM supports 10 interrupt sources. When a GTIM interrupt event occurs, the interrupt flag bit will be set by hardware, and an interrupt request will be generated if the corresponding interrupt enable control bit is set.

In the user GTIM interrupt service procedure, the relevant GTIM interrupt flag bit should be queried for corresponding processing, and the interrupt flag bit should be cleared before exiting the interrupt service procedure to avoid entering the interrupt service procedure repeatedly.

The following table shows the flag bit, interrupt enable bit, interrupt flag clear bit or clear method of each GTIM interrupt source:

Table 16-16 GTIM interrupt control

| Interrupt event                    | Interrupt flag bit | Interrupt enable bit | Interrupt flag clear                        |
|------------------------------------|--------------------|----------------------|---------------------------------------------|
| Update interrupt                   | ISR.UIF            | IER.UIE              | Write 0 to ICR.UIF.                         |
| Capture/compare 1 interrupt        | ISR.CC1IF          | IER.CC1IE            | Write 0 to ICR.CC1IF or read CCR1 register. |
| Capture/compare 2 interrupt        | ISR.CC2IF          | IER.CC2IE            | Write 0 to ICR.CC2IF or read CCR2 register. |
| Capture/compare 3 interrupt        | ISR.CC3IF          | IER.CC3IE            | Write 0 to ICR.CC3IF or read CCR3 register. |
| Capture/compare 4 interrupt        | ISR.CC4IF          | IER.CC4IE            | Write 0 to ICR.CC4IF or read CCR4 register. |
| Trigger interrupt                  | ISR.TIF            | IER.TIE              | Write 0 to ICR.TIF.                         |
| Encoder index interrupt            | ISR.IDXF           | IER.IDXIE            | Write 0 to ICR.IDXF.                        |
| Encoder direction change interrupt | ISR.DIRF           | IER.DIRIE            | Write 0 to ICR.DIRF.                        |
| Encoder index error interrupt      | ISR.IERRF          | IER.IERRIE           | Write 0 to ICR.IERRF.                       |
| Encoder transition error interrupt | ISR.TERRF          | IER.TERRIE           | Write 0 to ICR.TERRF.                       |



## 16.5 Trigger ADC

The timer supports a variety of internal signals to trigger the startup of the ADC, such as capturing the compare signal and triggering the output signal TRGO. There are several possible events for triggering the output TRGO, which are selected by the MMS bit field of the control register GTIMx\_CR2. At the same time, the ADC peripherals need to configure their external trigger start registers to select the corresponding trigger source.

It should be noted that the ADC clock must be enabled before an event can be received from the main timer and that the ADC clock must not be changed in real time when the trigger signal is received from the timer.



## 16.6 Direct memory access (DMA)

GTIM supports the generation of DMA requests when capture/compare, trigger and update events occur. Using the DMA function, it is possible to enable data to be written to the timer from other positions, or read from the timer to be written to other positions, which can be applied, for example, to the auto-carrying of data after a data capture and to changing the period or pulse width of the PWM output.

The DMA/interrupt enable register GTIMx\_DIER related bit field is used to enable the DMA request. The user needs to set the corresponding DMA request enable bit in the GTIMx\_DIER register to 1 and set the HARDSRC bit field of the DMA trigger source control register DMA\_TRIGy to the corresponding trigger source.

With the CCDS bit field of control register GTIMx\_CR2, you can select to send a DMA request when a capture compare event occurs or when an update event occurs.



## 16.7 Debugging support

GTIM supports stopping or continuing counting in debug mode, which is set up by the GTIMx bit field of the debug state timer control register SYSCTRL\_DEBUG:

- Set SYSCTRL\_DEBUG.GTIMx to 1, then the counter of GTIMx is suspended in the debug state.
- Set SYSCTRL\_DEBUG.GTIMx to 0, then the counter of GTIMx continues to count while in the debug state.





## 16.8 Programming examples

### 16.8.1 External clock mode 1 programming example

The following example configures the incremental counter to count on the rising edge of the GTIMx\_CH1 channel, with the following steps:

- Step 1:** Set `SYSCTRL_AHBEN.GPIOx` to 1 and `SYSCTRL_APBEN1.GTIMx` to 1, enable the GPIO clock corresponding to the GTIMx\_CH1 pin and the GTIM configuration clock and operating clock.
- Step 2:** Configure the GPIO corresponding to the GTIMx\_CH1 pin into the multiplexed input mode, for specific register configurations, please refer to [9 General-purpose input/output \(GPIO\)](#).
- Step 3:** Set `GTIMx_TISEL.TI1SEL` to 0 to select the TI1 source as the GTIMx\_CH1 channel.
- Step 4:** Set `GTIMx_CCMR1CAP.IC1F` to configure the input filter bandwidth.
- Step 5:** Set `GTIMx_CCER.CC1NP` to 0 and `GTIMx_CCER.CC1P` to 0 to select the rising edge count to be valid.
- Step 6:** Set `GTIMx_SMCR.TS` to 0x5 to select the TRGI input as TI1FP1.
- Step 7:** Set `GTIMx_SMCR.SMS` to 0x7 to make GTIM to work in external clock mode 1.
- Step 8:** Set `GTIMx_DIER.UIE` to 1 and configure the corresponding NVIC to enable the update interrupt.
- Step 9:** Configure the counter prescaler `GTIMx_PSC`.
- Step 10:** Set the expected reload value `GTIMx_ARR`.
- Step 11:** Set `GTIMx_CR1.CEN` to 1 to enable the counter.
- Step 12:** When the counter overflows, the `GTIMx_ISR.UIF` flag bit is set to 1 to enter the interrupt service routine, and set `GTIMx_ICR.UIF` to 0 to clear this interrupt flag.

### 16.8.2 External clock mode 2 programming example

The following is a programming example for external clock mode 2 + trigger mode, where the GTIMx\_CH1 input triggers to start the counter when a rising edge occurs, and the counter counts on each rising edge of the GTIMx\_ETR input signal, with the following steps:

- Step 1:** Set `SYSCTRL_AHBEN.GPIOx` to 1 and `SYSCTRL_APBEN1.GTIMx` to 1, enable the GPIO clocks corresponding to the GTIMx\_ETR and GTIMx\_CH1 pins and the GTIM configuration clock and operating clock.
- Step 2:** Configure the GPIOs corresponding to the GTIMx\_ETR, GTIMx\_CH1 pins into the multiplexed input mode, for specific register configurations, please refer to [9 General-purpose input/output \(GPIO\)](#).
- Step 3:** Set `GTIMx_AF1.ETRSEL` to 0 to select the ETR source as the GTIMx\_ETR pin.
- Step 4:** Set `GTIMx_SMCR.ETP` to 0 to select ETR rising edge valid.
- Step 5:** Set `GTIMx_SMCR.ETPS` to 0 to disable the prescaler for ETR.
- Step 6:** Set `GTIMx_SMCR.ETF` to 0 to disable ETR input filtering.
- Step 7:** Set `GTIMx_SMCR.ECE` to 1 to enable external clock mode 2.
- Step 8:** Set `GTIMx_TISEL.TI1SEL` to 0 to select the TI1 source as the GTIMx\_CH1 channel.
- Step 9:** Set `GTIMx_CCMR1CAP.IC1F` to 0 to disable TI1 input filtering.
- Step 10:** Set `GTIMx_CCER.CC1NP` to 0 and `GTIMx_CCER.CC1P` to 0 to select detection of TI1 rising edge.
- Step 11:** Set `GTIMx_SMCR.TS` to 0x5 to select the TRGI input as TI1FP1.
- Step 12:** Set `GTIMx_SMCR.SMS` to 0x6 to make GTIM work in trigger mode.
- Step 13:** When enabled on a rising edge of the GTIMx\_CH1 input, the counter counts on each rising edge of the GTIMx\_ETR input signal.



### 16.8.3 Reset mode programming example

In the following example, the counter is reinitialised when a rising edge occurs on the GTIMx\_CH1 input, with the following steps:

- Step 1:** Set `SYSCTRL_AHBEN.GPIOx` to 1 and `SYSCTRL_APBEN1.GTIMx` to 1. Enable the GPIO clock corresponding to the GTIMx\_CH1 pin and the GTIM configuration clock and operating clock.
- Step 2:** Configure the GPIO corresponding to the GTIMx\_CH1 pin into the multiplexed input mode, for specific register configurations, please refer to [9 General-purpose input/output \(GPIO\)](#).
- Step 3:** Set `GTIMx_TISEL.TI1SEL` to 0 to select the TI1 source as the GTIMx\_CH1 channel.
- Step 4:** Set `GTIMx_CCMR1CAP.IC1F` to configure the input filter bandwidth.
- Step 5:** Set `GTIMx_CCER.CC1NP` to 0 and `GTIMx_CCER.CC1P` to 0 to select the detection of the rising edge.
- Step 6:** Set `GTIMx_SMCR.TS` to 0x5 to select the TRGI input as TI1FP1.
- Step 7:** Set `GTIMx_SMCR.SMS` to 0x4 to make GTIM to work in reset mode.
- Step 8:** Set the expected reload value `GTIMx_ARR`.
- Step 9:** Set `GTIMx_CR1.CEN` to 1 to enable the counter and start normal counting.
- Step 10:** When a rising edge occurs on the GTIMx\_CH1 input, the counter is cleared to zero and counting starts again from 0.

### 16.8.4 Gated mode programming example

In the following example, the signal input from the GTIMx\_CH1 pin is used as the gated signal to control the counter counting, with the following steps:

- Step 1:** Set `SYSCTRL_AHBEN.GPIOx` to 1 and `SYSCTRL_APBEN1.GTIMx` to 1. Enable the GPIO clock corresponding to the GTIMx\_CH1 pin and the GTIM configuration clock and operating clock.
- Step 2:** Configure the GPIO corresponding to the GTIMx\_CH1 pin into the multiplexed input mode, for specific register configurations, please refer to [9 General-purpose input/output \(GPIO\)](#).
- Step 3:** Set `GTIMx_TISEL.TI1SEL` to 0 to select the TI1 source as the GTIMx\_CH1 channel.
- Step 4:** Set `GTIMx_CCMR1CAP.IC1F` to configure the input filter bandwidth.
- Step 5:** Set `GTIMx_CCER.CC1NP` to 0 and `GTIMx_CCER.CC1P` to 0 to select detect high level.
- Step 6:** Set `GTIMx_SMCR.TS` to 0x5 to select the TRGI input as TI1FP1.
- Step 7:** Set `GTIMx_SMCR.SMS` to 0x5 to make GTIM to work in gated mode.
- Step 8:** Set the expected reload value `GTIMx_ARR`.
- Step 9:** Set `GTIMx_CR1.CEN` to 1 to enable the counter.
- Step 10:** When GTIMx\_CH1 input is high, the counter starts counting; when GTIMx\_CH1 input is low, the counter stops counting.



### 16.8.5 Trigger mode programming example

In the following example, the counter is triggered to start when a rising edge occurs on the GTIMx\_CH1 input, with the following steps:

- Step 1:** Set `SYSCTRL_AHBEN.GPIOx` to 1 and `SYSCTRL_APBEN1.GTIMx` to 1. Enable the GPIO clock corresponding to the GTIMx\_CH1 pin and the GTIM configuration clock and operating clock.
- Step 2:** Configure the GPIO corresponding to the GTIMx\_CH1 pin into the multiplexed input mode, for specific register configurations, please refer to [9 General-purpose input/output \(GPIO\)](#).
- Step 3:** Set `GTIMx_TISEL.TI1SEL` to 0 to select the TI1 source as the GTIMx\_CH1 channel.
- Step 4:** Set `GTIMx_CCMR1CAP.IC1F` to configure the input filter bandwidth.
- Step 5:** Set `GTIMx_CCER.CC1NP` to 0 and `GTIMx_CCER.CC1P` to 0 to select the detection of the rising edge.
- Step 6:** Set `GTIMx_SMCR.TS` to 0x5 to select the TRGI input as TI1FP1.
- Step 7:** Set `GTIMx_SMCR.SMS` to 0x6 to make GTIM to work in trigger mode.
- Step 8:** Set the expected reload value `GTIMx_ARR`.
- Step 9:** When a rising edge occurs on the GTIMx\_CH1 input, the counter starts counting and the TIF flag is set to 1.

### 16.8.6 Quadrature encoder mode programming example

In the following example, configure GTIM to work in quadrature encoder mode-x4 mode, with the following steps:

- Step 1:** Set `SYSCTRL_AHBEN.GPIOx` to 1, `SYSCTRL_APBEN1.GTIMx` to 1, and enable the GPIO clocks corresponding to the GTIMx\_CH1 and GTIMx\_CH2 pins as well as the GTIM configuration clock and operating clock.
- Step 2:** Configure the GPIO corresponding to the GTIMx\_CH1 and GTIMx\_CH2 pins into the multiplexed input mode, for specific register configurations, please refer to [9 General-purpose input/output \(GPIO\)](#).
- Step 3:** Set `GTIMx_TISEL.TI1SEL` to 0 to select the TI1 source as GTIMx\_CH1 channel.
- Step 4:** Set `GTIMx_TISEL.TI2SEL` to 0 to select the TI2 source as the GTIMx\_CH2 channel.
- Step 5:** Set `GTIMx_CCMR1CAP.IC1F` and `GTIMx_CCMR1CAP.IC2F` to configure the input filter bandwidth.
- Step 6:** Set `GTIMx_CCER.CC1NP` and `GTIMx_CCER.CC1P` to 0, CH1 input is not inverted.
- Step 7:** Set `GTIMx_CCER.CC2NP` and `GTIMx_CCER.CC2P` to 0, CH2 input is not inverted.
- Step 8:** Set `GTIMx_SMCR.SMS` to 0x3 to make GTIM work in quadrature encoder mode-x4 mode.
- Step 9:** Set the appropriate reload value `GTIMx_ARR`.
- Step 10:** Set `GTIMx_CR1.CEN` to 1 to enable the counter.



## 16.8.7 Input capture programming example

### 16.8.7.1 Basic input capture mode

In the following example, execute an input capture when a rising edge occurs on the GTIMx\_CH1 input, with the following steps:

- Step 1:** Set `SYSCTRL_AHBEN.GPIOx` to 1, `SYSCTRL_APBEN1.GTIMx` to 1, and enable the GPIO clock corresponding to the GTIMx\_CH1 pin and the GTIM configuration clock and operating clock.
- Step 2:** Configure the GPIO corresponding to the GTIMx\_CH1 pin into the multiplexed input mode, for specific register configurations, please refer to [9 General-purpose input/output \(GPIO\)](#).
- Step 3:** Set `GTIMx_TISEL.TI1SEL` to 0 to select the TI1 source as the GTIMx\_CH1 channel.
- Step 4:** Set `GTIMx_CCMR1CAP.IC1F` to configure the TI1 input filter bandwidth.
- Step 5:** Set `GTIMx_CCER.CC1NP` to 0 and `GTIMx_CCER.CC1P` to 0 to select detection of IC1 rising edge.
- Step 6:** Set `GTIMx_CCMR1CAP.CC1S` to 1, the CC1 channel is configured as an input and IC1 is mapped to TI1.
- Step 7:** Set `GTIMx_CCMR1CAP.IC1PSC` to 0 to disable the IC1 prescaler.
- Step 8:** Set `GTIMx_CCER.CC1E` to 1 to enable input capture 1 mode.
- Step 9:** Set `GTIMx_DIER.CC1IE` to 1 and configure the corresponding NVIC to enable the input capture 1 interrupt.
- Step 10:** Configure the counter prescaler `GTIMx_PSC`.
- Step 11:** Set the expected reload value `GTIMx_ARR`.
- Step 12:** Set `GTIMx_CR1.CEN` to 1 to enable the counter and start normal counting.
- Step 13:** When a capture occurs, the value of the current counter CNT is latched into the capture/compare register CCR1, completing a capture, while the `GTIMx_ISR.CC1IF` flag bit is set to 1 to enter the interrupt service procedure, and this interrupt flag is cleared by setting `GTIMx_ICR.CC1IF` to 0.



### 16.8.7.2 PWM input mode

The following example measures the period and duty cycle of the PWM signal input from the GTIMx\_CH1 pin, with the following steps:

- Step 1:** Set SYSCTRL\_AHBEN.GPIOx to 1, SYSCTRL\_APBEN1.GTIMx to 1, and turn on the GPIO clock corresponding to the GTIMx\_CH1 pin and the GTIM configuration clock and operating clock.
- Step 2:** Configure the GPIO corresponding to the GTIMx\_CH1 pin into the multiplexed input mode, for specific register configurations, please refer to [9 General-purpose input/output \(GPIO\)](#).
- Step 3:** Set GTIMx\_TISEL.TI1SEL to 0 to select the TI1 source as the GTIMx\_CH1 channel.
- Step 4:** Set GTIMx\_CCMR1CAP.IC1F to configure the TI1 input filter bandwidth.
- Step 5:** Set GTIMx\_CCER.CC1NP to 0 and GTIMx\_CCER.CC1P to 0 to select detection of IC1 rising edge.
- Step 6:** Set GTIMx\_CCMR1CAP.CC1S to 1, the CC1 channel is configured as an input and IC1 is mapped to TI1.
- Step 7:** Set GTIMx\_CCMR1CAP.IC1PSC to 0 to disable the IC1 prescaler.
- Step 8:** Set GTIMx\_CCER.CC1E to 1 to enable input capture 1 mode.
- Step 9:** Set GTIMx\_DIER.CC1IE to 1 and configure the corresponding NVIC to enable input capture 1 interrupt.
- Step 10:** Set GTIMx\_CCER.CC2NP to 0 and GTIMx\_CCER.CC2P to 1 to select detect IC2 falling edge.
- Step 11:** Set GTIMx\_CCMR1CAP.CC2S to 2, CC2 channel configured as input, IC2 mapped to TI1.
- Step 12:** Set GTIMx\_CCMR1CAP.IC2PSC to 0 to disable the IC2 prescaler.
- Step 13:** Set GTIMx\_CCER.CC2E to 1 to enable input capture 2 mode.
- Step 14:** Set GTIMx\_SMCR.TS to 0x5 to select the TRGI input as TI1FP1.
- Step 15:** Set GTIMx\_SMCR.SMS to 0x4 to enable the GTIM to work in reset mode.
- Step 16:** Configure the counter prescaler GTIMx\_PSC.
- Step 17:** Set the expected reload value GTIMx\_ARR.
- Step 18:** Set GTIMx\_CR1.CEN to 1 to enable the counter and start normal counting.



### 16.8.8 Output compare programming example

In the following example, channel CH1 outputs the set waveform externally, with the following steps:

- Step 1:** Set `SYSCTRL_AHBEN.GPIOx` to 1, `SYSCTRL_APBEN1.GTIMx` to 1, enable the GPIO clock corresponding to the `GTIMx_CH1` pin and the GTIM configuration clock and working clock.
- Step 2:** Configure the GPIO corresponding to the `GTIMx_CH1` pin into the multiplexed output mode, for specific register configurations, please refer to [9 General-purpose input/output \(GPIO\)](#).
- Step 3:** Set `GTIMx_CCER.CC1P` to 0 to select the output polarity.
- Step 4:** Set `GTIMx_CCMR1CMP.CC1S` to 0 to configure the CC1 channel as an output.
- Step 5:** Set `GTIMx_CCMR1CMP.OC1M` to select output compare 1 mode.
- Step 6:** Set `GTIMx_CCER.CC1E` to 1 to output the OC1 signal on the corresponding output pin.
- Step 7:** Configure the counter prescaler `GTIMx_PSC`.
- Step 8:** Set the expected reload value `GTIMx_ARR`.
- Step 9:** Set the expected comparison value `GTIMx_CCR1`.
- Step 10:** Set `GTIMx_DIER.CC1IE` to 1 and configure the corresponding NVIC to enable the compare 1 interrupt.
- Step 11:** Set `GTIMx_CR1.CEN` to 1 to enable the counter.
- Step 12:** When a compare match occurs, the `GTIMx_ISR.CC1IF` flag bit is set to 1 to enter the interrupt service procedure, and this interrupt flag is cleared by setting `GTIMx_ICR.CC1IF` to 0.



## 16.9 List of registers

GTIM1 base address: GTIM1\_BASE = 0x4000 1800

GTIM2 base address: GTIM2\_BASE = 0x4000 1C00

GTIM3 base address: GTIM3\_BASE = 0x4000 2400

GTIM4 base address: GTIM4\_BASE = 0x4000 2800

Table 16-17 List of GTIM registers

| Register name  | Register address  | Register description                    |
|----------------|-------------------|-----------------------------------------|
| GTIMx_CR1      | GTIMx_BASE + 0x00 | Control register 1                      |
| GTIMx_CR2      | GTIMx_BASE + 0x04 | Control register 2                      |
| GTIMx_SMCR     | GTIMx_BASE + 0x08 | Slave mode control register             |
| GTIMx_DIER     | GTIMx_BASE + 0x0C | DMA/Interrupt enable register           |
| GTIMx_ISR      | GTIMx_BASE + 0x10 | Interrupt flag register                 |
| GTIMx_ICR      | GTIMx_BASE + 0x70 | Interrupt flag clear register           |
| GTIMx_EGR      | GTIMx_BASE + 0x14 | Event generation register               |
| GTIMx_CCMR1CAP | GTIMx_BASE + 0x18 | Capture mode register 1                 |
| GTIMx_CCMR1CMP | GTIMx_BASE + 0x18 | Compare mode register 1                 |
| GTIMx_CCMR2CAP | GTIMx_BASE + 0x1C | Capture mode register 2                 |
| GTIMx_CCMR2CMP | GTIMx_BASE + 0x1C | Compare mode register 2                 |
| GTIMx_CCER     | GTIMx_BASE + 0x20 | Capture/compare enable register         |
| GTIMx_CNT      | GTIMx_BASE + 0x24 | Count register                          |
| GTIMx_PSC      | GTIMx_BASE + 0x28 | Prescaler register                      |
| GTIMx_ARR      | GTIMx_BASE + 0x2C | Auto reload register                    |
| GTIMx_CCR1     | GTIMx_BASE + 0x34 | Capture/compare register 1              |
| GTIMx_CCR2     | GTIMx_BASE + 0x38 | Capture/compare register 2              |
| GTIMx_CCR3     | GTIMx_BASE + 0x3C | Capture/compare register 3              |
| GTIMx_CCR4     | GTIMx_BASE + 0x40 | Capture/compare register 4              |
| GTIMx_ECR      | GTIMx_BASE + 0x58 | Encoder control register                |
| GTIMx_TISEL    | GTIMx_BASE + 0x5C | TI input selection register             |
| GTIMx_AF1      | GTIMx_BASE + 0x60 | Multiplexing function option register 1 |
| GTIMx_AF2      | GTIMx_BASE + 0x64 | Multiplexing function option register 2 |



## 16.10 Register description

See section 1 [Documentation conventions](#) for abbreviations used in register descriptions.

### 16.10.1 GTIMx\_CR1 Controls register 1

Address offset: 0x00      Reset value: 0x0000 0000

| Bit field | Name     | Permission | Function description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|-----------|----------|------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:12     | RFU      | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 11        | UIFREMAP | RW         | UIF status bit remap<br>0: Remapping is disabled, GTIMx_CNT[31] remains 0.<br>1: Remapping is enabled, GTIMx_CNT[31] is equivalent to GTIMx_ISR.UIF.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| 10        | RFU      | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 9:8       | CKD      | RW         | Configure the PCLK clock to filter clock division ratio.<br>00: $f_{DTS} = f_{PCLK}/1$<br>01: $f_{DTS} = f_{PCLK}/2$<br>10: $f_{DTS} = f_{PCLK}/4$<br><i>Note: The filter clock is used for ETR filtering and TI filtering.</i>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| 7         | ARPE     | RW         | Auto reload preload enable<br>0: Disable ARR register buffering.<br>1: Enable ARR register buffering.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 6:5       | CMS      | RW         | Center-aligned mode selection<br>00: Edge aligned mode. The counter counts up or down according to the direction bit (DIR).<br>01: Center-aligned mode 1. The counter alternates between counting up or down. Only when the counter is counting down, the output compare interrupt flag of the channel configured as an output (CCyS = 00 in the GTIMx_CCMRx register) is set to 1.<br>10: Center-aligned mode 2. the counter alternates between counting up and counting down. Only when the counter is counting up, the output compare interrupt flag of the channel configured as an output (CCyS = 00 in the GTIMx_CCMRx register) is set to 1.<br>11: Center-aligned mode 3. The counter alternates between counting up and counting down. When the counter counts up or down, the output compare interrupt flag of the channel configured as an output (CCyS = 00 in the GTIMx_CCMRx register) is set to 1.<br><i>Attention: As long as the counter is enabled (CEN = 1), it must not be switched from edge-aligned mode to center-aligned mode.</i><br><i>Note: The initial count direction for centre-aligned mode is determined by the DIR bit field.</i> |





| Bit field | Name | Permission | Function description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|-----------|------|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 4         | DIR  | RW/RO      | Count direction<br>0: Counter count up.<br>1: Counter count down.<br><i>Note: When the timer is configured in center-aligned mode or encoder mode, this bit is read-only.</i>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 3         | OPM  | RW         | Single pulse mode<br>0: Counter does not stop counting when an update event occurs.<br>1: Counter stops counting when the next update event occurs (clears the CEN bit to zero).                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 2         | URS  | RW         | Update request source<br>This bit is set to 1 and cleared to 0 by software to select the UEV event source.<br>0: When UEV is enabled, all of the following events generate an update interrupt or DMA request. Including:<br><ul style="list-style-type: none"> <li>Counter overflow/underflow.</li> <li>Set UG bit to 1.</li> <li>Update events generated by the slave mode controller.</li> </ul> 1: When UEV is enabled, only counter overflow/underflow generates an update interrupt or DMA request.                                                                                                                                                                                                                         |
| 1         | UDIS | RW         | Update disabled<br>This bit is set to 1 and cleared to 0 by software to enable/disable UEV event generation.<br>0: UEV enabled. Update events (UEV) can be generated by one of the following events, which then updates the value of the shadow register:<br><ul style="list-style-type: none"> <li>Counter overflow/underflow.</li> <li>Set UG bit to 1.</li> <li>Update events generated by the slave mode controller.</li> </ul> 1: UEV disabled. No update event is generated and the values of the respective shadow registers (ARR, PSC, and CCRy) remain unchanged. However, if the UG bit field is set to 1, or a hardware reset is received from the slave mode controller, the counter and prescaler are reinitialised. |
| 0         | CEN  | RW         | Counter enable<br>0: Counter disabled.<br>1: Counter enabled.<br><i>Note: Only if the CEN bit field is set to 1 in advance by software can external clock, gated mode and encoder mode be used. Trigger modes can be used by automatically setting the CEN bit field to 1 via hardware.</i><br><i>Note: In single pulse mode, the CEN bit is automatically cleared to 0 when an update event occurs.</i>                                                                                                                                                                                                                                                                                                                          |

### 16.10.2 GTIMx\_CR2 Control register 2

Address offset: 0x04      Reset value: 0x0000 0000



| Bit field | Name | Permission | Function description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|-----------|------|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:27     | RFU  | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 26:25     | MMSH | RW         | Master mode selection<br>It is the highest bit of the MMS and must be used with the MMS.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 24:8      | RFU  | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 7         | TI1S | RW         | TI1 input selection<br>0: GTIMx_CH1 pin is connected to the TI1 input.<br>1: XOR combination of GTIMx_CH1, CH2 and CH3 pins are connected to TI1 input.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| 6:4       | MMS  | RW         | Master mode selection<br>These bits select the information that will be sent to the slave timer for synchronisation in master mode (TRGO). Used in conjunction with MMSH, these bits are combined as follows:<br>0000: Reset - The UG bit in the GTIMx_EGR register is used as the trigger output (TRGO). If the reset is generated by a trigger input (slave mode controller configured for reset mode), the signal on TRGO is delayed compared to the actual reset.<br>0001: Enable -<br>The counter enable signal CNT_EN is used as a trigger output (TRGO). This trigger output can be used to start multiple timers at the same time, or to control the enabling of slave timers over a period of time. The counter enable signal consists of a logical sum operation combining the CEN control bit with the trigger input in gated mode. When the counter enable signal is controlled by a trigger input, there is a delay on the TRGO, except when master/slave mode is selected (please refer to the description of the MSM bit in <a href="#">16.10.3 GTIMx_SMCR Slave mode control register</a> ).<br>0010: Update - Selects the update event as the trigger output (TRGO). For example, the master timer can be used as a prescaler for the slave timer.<br>0011: Compare capture pulse - Once an input capture or compare match event occurs on channel 1, the trigger output sends a positive pulse (TRGO), even if the CC1IF flag has been set to 1 (or even if it has been high-level).<br>0100: Compare capture pulse - Once an input capture or compare match event occurs on channel 2, the trigger output sends a positive pulse (TRGO), even if the CC2IF flag has been set to 1 (or even if it has been high-level). |



| Bit field          | Name | Permission | Function description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|--------------------|------|------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 6:4<br>(continued) | MMS  | RW         | <p>0101: Compare capture pulse - Once an input capture or compare match event occurs on channel 3, the trigger output sends a positive pulse (TRGO), even if the CC3IF flag has been set to 1 (or even if it has been high-level).</p> <p>0110: Compare capture pulse - Once an input capture or compare match event occurs on channel 4, the trigger output sends a positive pulse (TRGO), even if the CC4IF flag has been set to 1 (or even if it has been high-level).</p> <p>0111: OC1_REF - OC1REF signal is used as trigger output (TRGO).</p> <p>1000: OC2_REF - OC2REF signal is used as trigger output (TRGO).</p> <p>1001: OC3_REF - OC3REF signal is used as trigger output (TRGO).</p> <p>1010: OC4_REF - OC4REF signal is used as trigger output (TRGO).</p> <p>1011: OC1_REFC - OC1REFC signal is used as trigger output (TRGO).</p> <p>1100: OC2_REFC - OC2REFC signal is used as trigger output (TRGO).</p> <p>1101: OC3_REFC - OC3REFC signal is used as trigger output (TRGO).</p> <p>1110: OC4_REFC - OC4REFC signal is used as trigger output (TRGO).</p> <p>1111: Encoder clock output - The encoder clock signal is used as a trigger output (TRGO). This code applies to the following SMS[3:0] values: 0001, 0010, 0011, 1010, 1011, 1100, 1101, 1110, 1111. No other SMS[3:0] values are permitted, which may result in unexpected behaviour.</p> <p><i>Attention: The clock of the slave timer or ADC must be enabled before an event can be received from the master timer; and the clock of the slave timer or ADC must not be changed in real time when a trigger signal is received from the master timer.</i></p> |
| 3                  | CCDS | RW         | <p>Capture/compare DMA selection</p> <p>0: CCxDMA request is sent when a CCx event occurs.</p> <p>1: CCxDMA request is sent when an update event occurs.</p> <p><i>Attention: When this bit is set to 1, a CCxDMA request is sent whenever an update event occurs, regardless of whether a CCx event is generated.</i></p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 2:0                | RFU  | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |



### 16.10.3 GTIMx\_SMCR Slave mode control register

Address offset: 0x08

Reset value: 0x0000 0000

| Bit field | Name  | Permission | Function description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|-----------|-------|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:26     | RFU   | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 25        | SMSPS | RW         | Preload source<br>This bit selects the trigger source for the transfer of the SMS[3:0] bit fields from the preload register to the active registers.<br>0: Transfer is triggered by a timer update event.<br>1: Transfer is triggered by an index event.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 24        | SMSPE | RW         | Preload enable<br>This bit selects whether the SMS[3:0] bits are preloaded or not.<br>0: Preload disabled.<br>1: Preload enabled.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 23:22     | RFU   | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 21:20     | TSH   | RW         | Trigger selection<br>It is the highest bit of the TS, and needs to be used in conjunction with the TS.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 19:17     | RFU   | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 16        | SMSH  | RW         | Slave mode selection<br>It is the highest bit of SMS, and needs to be used in conjunction with the SMS.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 15        | ETP   | RW         | External trigger polarity<br>This bit selects whether ETR or the inversion of ETR will be used for the trigger operation.<br>0: ETR not inverted, high or rising edge is valid.<br>1: ETR inverted, low level or falling edge is valid.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 14        | ECE   | RW         | External clock enabled<br>This bit enables external clock mode 2.<br>0: Disable external clock mode 2.<br>1: Enables external clock mode 2, the counter clock is provided by any valid edge of the ETRF signal.<br><i>Note: Set the ECE bit field to 1 which has the same effect as selecting External Clock Mode 1 and connecting TRGI to ETRF (SMS = 111 and TS = 00111).</i><br><i>Note: External clock mode 2 can be used in conjunction with the following slave modes: reset mode, gated mode and trigger mode. However, in this case TRGI must not be connected to ETRF (TS bit must not be 00111).</i><br><i>Note: If external clock mode 1 and external clock mode 2 are enabled at the same time, the external clock input is ETRF.</i> |



| Bit field                                       | Name                                     | Permission | Function description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |                                                 |                                         |                                         |                                         |                                         |                                          |                                         |                                          |                                         |                                          |                                         |                                          |                                         |                                          |                                         |                                          |
|-------------------------------------------------|------------------------------------------|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------|-----------------------------------------|-----------------------------------------|-----------------------------------------|-----------------------------------------|------------------------------------------|-----------------------------------------|------------------------------------------|-----------------------------------------|------------------------------------------|-----------------------------------------|------------------------------------------|-----------------------------------------|------------------------------------------|-----------------------------------------|------------------------------------------|
| 13:12                                           | ETPS                                     | RW         | External trigger prescaler<br>The external trigger signal ETRP frequency must not exceed 1/4 of the PCLK frequency. the ETRP frequency can be reduced by enabling the prescaler. This method is useful when a fast external clock is input.<br>00: Prescaler disabled.<br>01: 2-division ETRP frequency.<br>10: 4-division ETRP frequency.<br>11: 8-division ETRP frequency.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |                                                 |                                         |                                         |                                         |                                         |                                          |                                         |                                          |                                         |                                          |                                         |                                          |                                         |                                          |                                         |                                          |
| 11:8                                            | ETF                                      | RW         | External trigger filter<br>This bit field defines the sampling frequency of the ETRP signal and the digital filter bandwidth applicable to the ETRP. The digital filter consists of an event counter, where only every N consecutive events are considered as a valid output edge:<br><table><tr><td>0000: No filter, sampled at frequency <math>f_{DTS}</math></td><td>1000: <math>f_{SAMPLING} = f_{DTS}/8, N = 6</math></td></tr><tr><td>0001: <math>f_{SAMPLING} = f_{DTS}/1, N = 2</math></td><td>1001: <math>f_{SAMPLING} = f_{DTS}/8, N = 8</math></td></tr><tr><td>0010: <math>f_{SAMPLING} = f_{DTS}/1, N = 4</math></td><td>1010: <math>f_{SAMPLING} = f_{DTS}/16, N = 5</math></td></tr><tr><td>0011: <math>f_{SAMPLING} = f_{DTS}/1, N = 8</math></td><td>1011: <math>f_{SAMPLING} = f_{DTS}/16, N = 6</math></td></tr><tr><td>0100: <math>f_{SAMPLING} = f_{DTS}/2, N = 6</math></td><td>1100: <math>f_{SAMPLING} = f_{DTS}/16, N = 8</math></td></tr><tr><td>0101: <math>f_{SAMPLING} = f_{DTS}/2, N = 8</math></td><td>1101: <math>f_{SAMPLING} = f_{DTS}/32, N = 5</math></td></tr><tr><td>0110: <math>f_{SAMPLING} = f_{DTS}/4, N = 6</math></td><td>1110: <math>f_{SAMPLING} = f_{DTS}/32, N = 6</math></td></tr><tr><td>0111: <math>f_{SAMPLING} = f_{DTS}/4, N = 8</math></td><td>1111: <math>f_{SAMPLING} = f_{DTS}/32, N = 8</math></td></tr></table> | 0000: No filter, sampled at frequency $f_{DTS}$ | 1000: $f_{SAMPLING} = f_{DTS}/8, N = 6$ | 0001: $f_{SAMPLING} = f_{DTS}/1, N = 2$ | 1001: $f_{SAMPLING} = f_{DTS}/8, N = 8$ | 0010: $f_{SAMPLING} = f_{DTS}/1, N = 4$ | 1010: $f_{SAMPLING} = f_{DTS}/16, N = 5$ | 0011: $f_{SAMPLING} = f_{DTS}/1, N = 8$ | 1011: $f_{SAMPLING} = f_{DTS}/16, N = 6$ | 0100: $f_{SAMPLING} = f_{DTS}/2, N = 6$ | 1100: $f_{SAMPLING} = f_{DTS}/16, N = 8$ | 0101: $f_{SAMPLING} = f_{DTS}/2, N = 8$ | 1101: $f_{SAMPLING} = f_{DTS}/32, N = 5$ | 0110: $f_{SAMPLING} = f_{DTS}/4, N = 6$ | 1110: $f_{SAMPLING} = f_{DTS}/32, N = 6$ | 0111: $f_{SAMPLING} = f_{DTS}/4, N = 8$ | 1111: $f_{SAMPLING} = f_{DTS}/32, N = 8$ |
| 0000: No filter, sampled at frequency $f_{DTS}$ | 1000: $f_{SAMPLING} = f_{DTS}/8, N = 6$  |            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                                                 |                                         |                                         |                                         |                                         |                                          |                                         |                                          |                                         |                                          |                                         |                                          |                                         |                                          |                                         |                                          |
| 0001: $f_{SAMPLING} = f_{DTS}/1, N = 2$         | 1001: $f_{SAMPLING} = f_{DTS}/8, N = 8$  |            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                                                 |                                         |                                         |                                         |                                         |                                          |                                         |                                          |                                         |                                          |                                         |                                          |                                         |                                          |                                         |                                          |
| 0010: $f_{SAMPLING} = f_{DTS}/1, N = 4$         | 1010: $f_{SAMPLING} = f_{DTS}/16, N = 5$ |            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                                                 |                                         |                                         |                                         |                                         |                                          |                                         |                                          |                                         |                                          |                                         |                                          |                                         |                                          |                                         |                                          |
| 0011: $f_{SAMPLING} = f_{DTS}/1, N = 8$         | 1011: $f_{SAMPLING} = f_{DTS}/16, N = 6$ |            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                                                 |                                         |                                         |                                         |                                         |                                          |                                         |                                          |                                         |                                          |                                         |                                          |                                         |                                          |                                         |                                          |
| 0100: $f_{SAMPLING} = f_{DTS}/2, N = 6$         | 1100: $f_{SAMPLING} = f_{DTS}/16, N = 8$ |            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                                                 |                                         |                                         |                                         |                                         |                                          |                                         |                                          |                                         |                                          |                                         |                                          |                                         |                                          |                                         |                                          |
| 0101: $f_{SAMPLING} = f_{DTS}/2, N = 8$         | 1101: $f_{SAMPLING} = f_{DTS}/32, N = 5$ |            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                                                 |                                         |                                         |                                         |                                         |                                          |                                         |                                          |                                         |                                          |                                         |                                          |                                         |                                          |                                         |                                          |
| 0110: $f_{SAMPLING} = f_{DTS}/4, N = 6$         | 1110: $f_{SAMPLING} = f_{DTS}/32, N = 6$ |            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                                                 |                                         |                                         |                                         |                                         |                                          |                                         |                                          |                                         |                                          |                                         |                                          |                                         |                                          |                                         |                                          |
| 0111: $f_{SAMPLING} = f_{DTS}/4, N = 8$         | 1111: $f_{SAMPLING} = f_{DTS}/32, N = 8$ |            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                                                 |                                         |                                         |                                         |                                         |                                          |                                         |                                          |                                         |                                          |                                         |                                          |                                         |                                          |                                         |                                          |
| 7                                               | MSM                                      | RW         | Master/ slave mode<br>0: No action is performed.<br>1: The action of the trigger input event (TRGI) of the current timer is postponed to allow perfect synchronisation (via TRGO) of the current timer with its slave timers. This setting applies when multiple timers are synchronised by a single external event.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |                                                 |                                         |                                         |                                         |                                         |                                          |                                         |                                          |                                         |                                          |                                         |                                          |                                         |                                          |                                         |                                          |
| 6:4                                             | TS                                       | RW         | Trigger input TRGI selection<br>This bit field selects the trigger input that will be used to synchronise the counter.<br>00000: Internal trigger 0 (ITR0)<br>00001: Internal trigger 1 (ITR1)<br>00010: Internal trigger 2 (ITR2)<br>00011: Internal trigger 3 (ITR3)<br>00100: TI1 edge detector (TI1F_ED)<br>00101: Filtered timer input 1 (TI1FP1)<br>00110: Filtered timer input 2 (TI2FP2)<br>00111: External trigger input (ETRF)<br>01000: Internal trigger 4 (ITR4)<br>01001: Internal trigger 5 (ITR5)<br>01010: Internal trigger 6 (ITR6)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |                                                 |                                         |                                         |                                         |                                         |                                          |                                         |                                          |                                         |                                          |                                         |                                          |                                         |                                          |                                         |                                          |



| Bit field          | Name | Permission | Function description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|--------------------|------|------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 6:4<br>(continued) | TS   | RW         | <p>01011: Internal trigger 7 (ITR7)<br/> 01100: Internal trigger 8 (ITR8)<br/> 01101: Internal trigger 9 (ITR9)<br/> 01110: Internal trigger 10 (ITR10)<br/> 01111: Internal trigger 11 (ITR11)<br/> 10000: Internal trigger 12 (ITR12)<br/> 10001: Internal trigger 13 (ITR13)<br/> 10010: Internal trigger 14 (ITR14)<br/> 10011: Internal trigger 15 (ITR15)<br/> Other values: Reserved<br/> <i>Note: These bits can only be changed if they are not used (e.g. SMS = 0000) to avoid incorrect edge detection during transition.</i><br/> <i>Note: The other bit fields are in bits 21 and 20 of the same register and need to be used in conjunction with the TS.</i><br/> <i>Note: For specific ITRx sources, please see <a href="#">Table 16-13: ITR signal source</a>.</i><br/> <i>Attention: Prohibits the selection of Trgo signal output by GTIMx itself.</i></p>                                                                                                                                                                                                                                                                                     |
| 3                  | OCCS | RW         | <p>OCREF cleared selection<br/> This bit is used to select the OCREF clear source.<br/> 0: OCREF_CLR_INT connected to OCREF_CLR input.<br/> 1: OCREF_CLR_INT connected to ETRF.<br/> <i>Note: OCREF_CLR input source please refer to <a href="#">16.10.23 GTIMx_AF2 Multiplexing function option register 2</a> for the description of the OCRSEL bit field.</i></p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 2:0                | SMS  | RW         | <p>Slave mode selection<br/> When an external signal is selected, the valid edge of the trigger signal (TRGI) is related to the polarity selected on the external input (see the input related control registers).<br/> 0000: Slave mode disabled - If CEN = '1', the prescaler clock is supplied directly from the internal clock.<br/> 0001: Quadrature encoder mode - x2 mode , the counter counts up/down on the TI1FP1 edge according to the TI2FP2 level.<br/> 0010: Quadrature encoder mode - x2 mode, the counter counts up/down according to the TI1FP1 level at the TI2FP2 edge.<br/> 0011: Quadrature encoder mode - x4 mode, the counter counts on the TI1FP1 and TI2FP2 edges, the direction of the count depends on the level of the other input.<br/> 0100: Reset mode - The counter is reinitialised and a register update event is generated on the occurrence of a rising edge of the selected trigger input (TRGI).<br/> 0101: Gated mode - Enables the counter clock when the trigger input (TRGI) is high. The counter stops counting (but not resetting) as soon as the trigger input goes low. Counter start and stop are controlled.</p> |



| Bit field          | Name | Permission | Function description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|--------------------|------|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 2:0<br>(continued) | SMS  | RW         | <p>0110: Trigger mode - Starts the counter (but does not reset) on a rising edge of the trigger signal (TRGI). Only the start of the counter is controlled.</p> <p>0111: External clock mode 1 - Counter clock is provided by the rising edge of the selected trigger signal (TRGI).</p> <p>1000: Combined reset + trigger mode - Reinitialises the counter on the rising edge of the selected trigger input (TRGI), generates a register update event and starts the counter.</p> <p>1001: Combined gated + reset mode - The counter is enabled and starts counting when the trigger input (TRGI) goes high, the counter stops counting and is reset when the trigger input goes low, in this mode the start and stop of the counter are controlled.</p> <p>1010: Encoder mode - Clock + direction, x2 mode.</p> <p>1011: Encoder mode - Clock+direction, x1 mode, TI2FP2's edge polarity is set by CC2P.</p> <p>1100: Encoder mode - Clock with direction, x2 mode.</p> <p>1101: Encoder mode - Clock with direction, x1 mode, TI1FP1 and TI2FP2 edge polarity is set by CC1P and CC2P.</p> <p>1110: Quadrature encoder mode - x1 mode, counts only TI1FP1 edges, whose edge polarity is set by CC1P.</p> <p>1111: Quadrature encoder mode - x1 mode, counts only TI2FP2 edges, whose polarity is set by CC2P.</p> <p><i>Note: If TI1F_ED is selected as the trigger input (TS = 00100), the gated mode must not be used. In fact, TI1F_ED outputs one pulse for each TI1F transition, while the gated mode checks the level of the trigger signal.</i></p> <p><i>Attention: The clock of the slave peripheral (timer, ADC, etc.) that receives the TRGO signal must be enabled before the event can be received from the master timer; and the clock frequency must not be changed in real time (prescaler) when the trigger signal is received from the master timer.</i></p> <p><i>Attention: When the slave mode is configured to 0100, 0101, 0110, 1000, and 1001, there is a TIE interrupt flag; otherwise, there is no TIE interrupt flag in the slave mode.</i></p> |



### 16.10.4 GTIMx\_DIER DMA/Interrupt enable register

Address offset: 0x0C

Reset value: 0x0000 0000

| Bit field | Name   | Permission | Function description                                                                                                                 |
|-----------|--------|------------|--------------------------------------------------------------------------------------------------------------------------------------|
| 31:24     | RFU    | -          | Reserved bits, please keep the default value.                                                                                        |
| 23        | TERRIE | RW         | Encoder transition error interrupt enable control<br>0: Disable transition error interrupt.<br>1: Enable transition error interrupt. |
| 22        | IERRIE | RW         | Encoder index error interrupt enable control<br>0: Disable index error interrupt.<br>1: Enable index error interrupt.                |
| 21        | DIRIE  | RW         | Encoder direction change interrupt enabled<br>0: Disable direction change interrupt.<br>1: Enable direction change interrupt.        |
| 20        | IDXIE  | RW         | Encoder index interrupt enable control<br>0: Disable index interrupt.<br>1: Enable index interrupt.                                  |
| 19:15     | RFU    | -          | Reserved bits, please keep the default value.                                                                                        |
| 14        | TDE    | RW         | Trigger DMA request enabled<br>0: Disable trigger DMA request.<br>1: Enable trigger DMA request.                                     |
| 13        | RFU    | -          | Reserved bits, please keep the default value.                                                                                        |
| 12        | CC4DE  | RW         | Capture/compare 4 DMA request enabled<br>0: Disable capture/compare 4 DMA request.<br>1: Enable capture/compare 4 DMA request.       |
| 11        | CC3DE  | RW         | Capture/compare 3 DMA request enabled<br>0: Disable capture/compare 3 DMA request.<br>1: Enable capture/compare 3 DMA request.       |
| 10        | CC2DE  | RW         | Capture/compare 2 DMA request enabled<br>0: Disable capture/compare 2 DMA request.<br>1: Enable capture/compare 2 DMA request.       |
| 9         | CC1DE  | RW         | Capture/compare1 DMA request enabled<br>0: Disable capture/compare 1 DMA request.<br>1: Enable Capture/compare 1 DMA request.        |
| 8         | UDE    | RW         | Update DMA request enabled<br>0: Disable update DMA request.<br>1: Enable update DMA request.                                        |
| 7         | RFU    | -          | Reserved bits, please keep the default value.                                                                                        |





| Bit field | Name  | Permission | Function description                                                                                                                                                                                                                   |
|-----------|-------|------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 6         | TIE   | RW         | Trigger interrupt enabled<br>0: Disable trigger interrupt<br>1: Enable trigger interrupt<br><i>Attention: When working in slave mode, this bit is only valid when the slave mode configuration is 0100, 0101, 0110, 1000, or 1001.</i> |
| 5         | RFU   | -          | Reserved bits, please keep the default value.                                                                                                                                                                                          |
| 4         | CC4IE | RW         | Capture/compare 4 interrupt enabled<br>0: Disable capture/compare 4 interrupt.<br>1: Enable capture/compare 4 interrupt.                                                                                                               |
| 3         | CC3IE | RW         | Capture/compare 3 interrupt enabled<br>0: Disable capture/compare 3 interrupt.<br>1: Enable capture/compare 3 interrupt.                                                                                                               |
| 2         | CC2IE | RW         | Capture/compare 2 interrupt enabled<br>0: Disable capture/compare 2 interrupt.<br>1: Enable capture/compare 2 interrupt.                                                                                                               |
| 1         | CC1IE | RW         | Capture/compare 1 interrupt enabled<br>0: Disable capture/compare 1 interrupt.<br>1: Enable capture/compare 1 interrupt.                                                                                                               |
| 0         | UIE   | RW         | Update interrupt enabled<br>0: Disable update interrupt.<br>1: Enable update interrupt.                                                                                                                                                |



### 16.10.5 GTIMx\_ISR Interrupt flag register

Address offset: 0x10

Reset value: 0x0000 0000

| Bit field | Name  | Permission | Function description                                                                                                                                                                                                                                                                                                                                                                                                                   |
|-----------|-------|------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:24     | RFU   | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                                                                                                                          |
| 23        | TERRF | RW         | Encoder transition error interrupt flag<br>0: No transition error occurred.<br>1: A transition error has occurred.                                                                                                                                                                                                                                                                                                                     |
| 22        | IERRF | RW         | Encoder index error interrupt flag<br>0: No index error occurred.<br>1: An indexing error has occurred.                                                                                                                                                                                                                                                                                                                                |
| 21        | DIRF  | RW         | Encoder direction change interrupt flag<br>0: No encoding direction change has occurred.<br>1: Encoding direction change has occurred.                                                                                                                                                                                                                                                                                                 |
| 20        | IDXF  | RW         | Encoder index interrupt flag<br>0: Index flag has not occurred.<br>1: Index flag has occurred.                                                                                                                                                                                                                                                                                                                                         |
| 19:13     | RFU   | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                                                                                                                          |
| 12        | CC4OF | RW         | Capture/compare 4 repeat capture flag, see CC1OF description.                                                                                                                                                                                                                                                                                                                                                                          |
| 11        | CC3OF | RW         | Capture/compare 3 repeat capture flag, see CC1OF description.                                                                                                                                                                                                                                                                                                                                                                          |
| 10        | CC2OF | RW         | Capture/compare 2 repeat capture flag, see CC1OF description.                                                                                                                                                                                                                                                                                                                                                                          |
| 9         | CC1OF | RW         | Capture/compare 1 repeat capture flag<br>This flag bit is set to 1 by hardware only when the corresponding channel is configured in input capture mode.<br>0: No repeat capture detected.<br>1: Counter value has been captured in the GTIMx_CCR1 register and the CC1IF flag has been set to 1.                                                                                                                                       |
| 8:7       | RFU   | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                                                                                                                          |
| 6         | TIF   | RW         | Trigger interrupt flag<br>In all modes except gated mode, this flag will be set by hardware to 1 when a TRGI trigger event occurs after enabling the slave mode controller (a valid edge is detected on the TRGI input.) When gated mode is selected, this flag will be set to 1 when the counter is started and stopped. This flag needs to be cleared by software.<br>0: No trigger event occurred.<br>1: Trigger interrupt pending. |
| 5         | RFU   | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                                                                                                                          |
| 4         | CC4IF | RW         | Capture/compare 4 interrupt flag, see CC1IF description.                                                                                                                                                                                                                                                                                                                                                                               |
| 3         | CC3IF | RW         | Capture/compare 3 interrupt flag, see CC1IF description.                                                                                                                                                                                                                                                                                                                                                                               |
| 2         | CC2IF | RW         | Capture/compare 2 interrupt flag, see CC1IF description.                                                                                                                                                                                                                                                                                                                                                                               |



| Bit field | Name  | Permission | Function description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|-----------|-------|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1         | CC1IF | RW         | <p>Capture/compare 1 interrupt flag</p> <p>This flag is set to 1 by hardware. It is cleared by software (Input capture or output compare modes), or by reading the GTIMx_CCR1 register (Input capture mode only).</p> <p>0: No compare match/input capture event occurs.<br/>1: Compare match/input capture event occurs.</p> <ul style="list-style-type: none"> <li>● If channel CC1 is configured as an output:<br/>This flag is set to 1 when the value of the counter GTIMx_CNT matches the value of the GTIMx_CCR1 register. When the value of GTIMx_CCR1 is greater than the value of GTIMx_ARR, the CC1IF bit will become high-level when the counter overflows (in count up mode and center-aligned mode) or underflows (in count down mode) occur. There are 3 possible flag setting options in center-aligned mode, see the CMS bit in the GTIMx_CR1 register for a complete description.</li> <li>● If channel CC1 is configured as input:<br/>This bit is set to 1 if: the counter value is captured in the GTIMx_CCR1 register (according to the edge sensitivity defined by the CC1P and CC1NP bit settings in GTIMx_CCER and an edge is detected on IC1).</li> </ul> |
| 0         | UIF   | RW         | <p>Update interrupt flag</p> <p>This bit is set to 1 by hardware when an update event occurs, but needs to be cleared by software.</p> <p>0: No update occurred.<br/>1: Update interrupt pending.</p> <p>This bit is set to 1 by hardware when the register is updated under the following conditions:</p> <ul style="list-style-type: none"> <li>● Overflow or underflow and when UDIS = 0 in the GTIMx_CR1 register.</li> <li>● URS = 0 and UDIS = 0 in the GTIMx_CR1 register and when the CNT is reinitialised by software using the UG bit in the GTIMx_EGR register.</li> <li>● URS = 0 and UDIS = 0 in the GTIMx_CR1 register and when the CNT is reinitialised by a trigger event (see slave mode control register description).</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                 |



### 16.10.6 GTIMx\_ICR Interrupt flag clear register

Address offset: 0x70

Reset value: 0x00F0 1E5F

| Bit field | Name  | Permission | Function description                                                                                                    |
|-----------|-------|------------|-------------------------------------------------------------------------------------------------------------------------|
| 31:24     | RFU   | -          | Reserved bits, please keep the default value.                                                                           |
| 23        | TERRF | R1W0       | Encoder transition error interrupt flag cleared<br>0: Clear encoder transition error interrupt flag.<br>1: No function. |
| 22        | IERRF | R1W0       | Encoder index error interrupt flag cleared<br>0: Clear encoder index error interrupt flag.<br>1: No function.           |
| 21        | DIRF  | R1W0       | Encoder direction change interrupt flag cleared<br>0: Clear encoder direction change interrupt flag.<br>1: No function. |
| 20        | IDXF  | R1W0       | Encoder index interrupt flag cleared<br>0: Clear encoder index interrupt flag.<br>1: No function.                       |
| 19:13     | RFU   | -          | Reserved bits, please keep the default value.                                                                           |
| 12        | CC4OF | R1W0       | Capture/compare 4 repeat capture flag cleared, see CC1OF description.                                                   |
| 11        | CC3OF | R1W0       | Capture/compare 3 repeat capture flag cleared, see CC1OF description.                                                   |
| 10        | CC2OF | R1W0       | Capture/compare 2 repeat capture flag cleared, see CC1OF description.                                                   |
| 9         | CC1OF | R1W0       | Capture/compare 1 repeat capture flag cleared<br>0: Clear capture/compare 1 repeat capture flag.<br>1: No function.     |
| 8:7       | RFU   | -          | Reserved bits, please keep the default value.                                                                           |
| 6         | TIF   | R1W0       | Trigger interrupt flag cleared<br>0: Clear trigger interrupt flag.<br>1: No function.                                   |
| 5         | RFU   | -          | Reserved bits, please keep the default value.                                                                           |
| 4         | CC4IF | R1W0       | Capture/compare 4 interrupt flag cleared, see CC1IF description.                                                        |
| 3         | CC3IF | R1W0       | Capture/compare 3 interrupt flag cleared, see CC1IF description.                                                        |
| 2         | CC2IF | R1W0       | Capture/compare 2 interrupt flag cleared, see CC1IF description.                                                        |
| 1         | CC1IF | R1W0       | Capture/compare 1 interrupt flag cleared<br>0: Clear capture/compare 1 interrupt flag.<br>1: No function.               |
| 0         | UIF   | R1W0       | Update interrupt flag cleared<br>0: Clear update interrupt flag.<br>1: No function.                                     |



### 16.10.7 GTIMx\_EGR Event generation register

Address offset: 0x14

Reset value: 0x0000 0000

| Bit field | Name | Permission | Function description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|-----------|------|------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:7      | RFU  | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 6         | TG   | RW         | <p>Trigger generated</p> <p>This bit is set to 1 by software to generate an event and is automatically cleared by hardware.</p> <p>0: No operation is performed.</p> <p>1: The TIF flag in the GTIMx_ISR register is set to 1. When enabled, the associated interrupt or DMA transfer event can occur.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 5         | RFU  | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 4         | CC4G | RW         | Capture/compare 4 generation, see CC1G description.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 3         | CC3G | RW         | Capture/compare 3 generation, see CC1G description.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 2         | CC2G | RW         | Capture/compare 2 generation, see CC1G description.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 1         | CC1G | RW         | <p>Capture/compare 1 generated</p> <p>This bit is set to 1 by software to generate an event and is automatically cleared by hardware.</p> <p>0: No operation is performed.</p> <p>1: Capture/compare events are generated on channel 1 and are classified as follows depending on the configuration of CC1:</p> <ul style="list-style-type: none"> <li>• If channel CC1 is configured as output: When enabled, the CC1IF flag is set to 1 and the corresponding interrupt or DMA request is sent.</li> <li>• If channel CC1 is configured as an input: The current value of the counter will be captured in the GTIMx_CCR1 register. When enabled, the CC1IF flag is set to 1 and the corresponding interrupt or DMA request is sent. If the CC1IF flag is already high, the CC1OF flag will be set to 1.</li> </ul> <p><i>Attention: CC1E must be enabled, otherwise CCR1 register will capture the current value of the counter, but no interrupt flag will be generated, and no interrupt can be entered.</i></p> |
| 0         | UG   | RW         | <p>Update generated</p> <p>This bit can be set to 1 by software and automatically cleared by hardware.</p> <p>0: No operation is performed.</p> <p>1: Reinitialise the counter and generate a register update event. Note that the prescaler counter will also be cleared (but the prescaler ratio is not affected). The counter will be cleared if center-aligned mode or up-count mode is selected; if down-count mode is selected, the counter will use the auto-reload value GTIMx_ARR.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |



### 16.10.8 GTIMx\_CCMR1CAP Capture mode register 1

Address offset: 0x18      Reset value: 0x0000 0000

This register is the same register as GTIMx\_CCMR1CMP and can be used in input capture mode or output compare mode. The channel direction is defined by configuring the corresponding CCyS bit. All other bits of this register function differently in input capture mode and output compare mode.

| Bit field                                       | Name                                     | Permission | Function description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                                                 |                                         |                                        |                                         |                                        |                                          |                                        |                                          |                                         |                                          |                                         |                                          |                                         |                                          |                                         |                                          |
|-------------------------------------------------|------------------------------------------|------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------|-----------------------------------------|----------------------------------------|-----------------------------------------|----------------------------------------|------------------------------------------|----------------------------------------|------------------------------------------|-----------------------------------------|------------------------------------------|-----------------------------------------|------------------------------------------|-----------------------------------------|------------------------------------------|-----------------------------------------|------------------------------------------|
| 31:16                                           | RFU                                      | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |                                                 |                                         |                                        |                                         |                                        |                                          |                                        |                                          |                                         |                                          |                                         |                                          |                                         |                                          |                                         |                                          |
| 15:12                                           | IC2F                                     | RW         | Input capture 2 filter, see IC1F description.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |                                                 |                                         |                                        |                                         |                                        |                                          |                                        |                                          |                                         |                                          |                                         |                                          |                                         |                                          |                                         |                                          |
| 11:10                                           | IC2PSC                                   | RW         | Input capture 2 prescaler, see IC1PSC description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |                                                 |                                         |                                        |                                         |                                        |                                          |                                        |                                          |                                         |                                          |                                         |                                          |                                         |                                          |                                         |                                          |
| 9:8                                             | CC2S                                     | RW         | Capture/compare 2 selection<br>This bit field defines the channel direction (input/output) and the input mapping used.<br>00: CC2 channel configured as output.<br>01: CC2 channel configured as input, IC2 mapped to TI2.<br>10: CC2 channel configured as input, IC2 mapped to TI1.<br>11: CC2 channel configured as input, IC2 mapped to TRC, this mode is only valid when the internal trigger input is selected via the TS bit ( GTIMx_SMCR register).<br><i>Attention: Data can be written to the CC2S bit only when the channel is closed (CC2E = 0 in GTIMx_CCER).</i>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |                                                 |                                         |                                        |                                         |                                        |                                          |                                        |                                          |                                         |                                          |                                         |                                          |                                         |                                          |                                         |                                          |
| 7:4                                             | IC1F                                     | RW         | Input capture 1 filter<br>This bit field defines the sampling frequency of the TI1 input and the sample length of the digital filter applicable to TI1. The digital filter consists of an event counter, where every N consecutive events are considered to be only one valid output edge.<br><table><tr><td>0000: No filter, sampled at frequency <math>f_{DTS}</math></td><td>1000: <math>f_{SAMPLING} = f_{DTS}/8, N = 6</math></td></tr><tr><td>0001: <math>f_{SAMPLING} = f_{PCLK}, N = 2</math></td><td>1001: <math>f_{SAMPLING} = f_{DTS}/8, N = 8</math></td></tr><tr><td>0010: <math>f_{SAMPLING} = f_{PCLK}, N = 4</math></td><td>1010: <math>f_{SAMPLING} = f_{DTS}/16, N = 5</math></td></tr><tr><td>0011: <math>f_{SAMPLING} = f_{PCLK}, N = 8</math></td><td>1011: <math>f_{SAMPLING} = f_{DTS}/16, N = 6</math></td></tr><tr><td>0100: <math>f_{SAMPLING} = f_{DTS}/2, N = 6</math></td><td>1100: <math>f_{SAMPLING} = f_{DTS}/16, N = 8</math></td></tr><tr><td>0101: <math>f_{SAMPLING} = f_{DTS}/2, N = 8</math></td><td>1101: <math>f_{SAMPLING} = f_{DTS}/32, N = 5</math></td></tr><tr><td>0110: <math>f_{SAMPLING} = f_{DTS}/4, N = 6</math></td><td>1110: <math>f_{SAMPLING} = f_{DTS}/32, N = 6</math></td></tr><tr><td>0111: <math>f_{SAMPLING} = f_{DTS}/4, N = 8</math></td><td>1111: <math>f_{SAMPLING} = f_{DTS}/32, N = 8</math></td></tr></table> | 0000: No filter, sampled at frequency $f_{DTS}$ | 1000: $f_{SAMPLING} = f_{DTS}/8, N = 6$ | 0001: $f_{SAMPLING} = f_{PCLK}, N = 2$ | 1001: $f_{SAMPLING} = f_{DTS}/8, N = 8$ | 0010: $f_{SAMPLING} = f_{PCLK}, N = 4$ | 1010: $f_{SAMPLING} = f_{DTS}/16, N = 5$ | 0011: $f_{SAMPLING} = f_{PCLK}, N = 8$ | 1011: $f_{SAMPLING} = f_{DTS}/16, N = 6$ | 0100: $f_{SAMPLING} = f_{DTS}/2, N = 6$ | 1100: $f_{SAMPLING} = f_{DTS}/16, N = 8$ | 0101: $f_{SAMPLING} = f_{DTS}/2, N = 8$ | 1101: $f_{SAMPLING} = f_{DTS}/32, N = 5$ | 0110: $f_{SAMPLING} = f_{DTS}/4, N = 6$ | 1110: $f_{SAMPLING} = f_{DTS}/32, N = 6$ | 0111: $f_{SAMPLING} = f_{DTS}/4, N = 8$ | 1111: $f_{SAMPLING} = f_{DTS}/32, N = 8$ |
| 0000: No filter, sampled at frequency $f_{DTS}$ | 1000: $f_{SAMPLING} = f_{DTS}/8, N = 6$  |            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |                                                 |                                         |                                        |                                         |                                        |                                          |                                        |                                          |                                         |                                          |                                         |                                          |                                         |                                          |                                         |                                          |
| 0001: $f_{SAMPLING} = f_{PCLK}, N = 2$          | 1001: $f_{SAMPLING} = f_{DTS}/8, N = 8$  |            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |                                                 |                                         |                                        |                                         |                                        |                                          |                                        |                                          |                                         |                                          |                                         |                                          |                                         |                                          |                                         |                                          |
| 0010: $f_{SAMPLING} = f_{PCLK}, N = 4$          | 1010: $f_{SAMPLING} = f_{DTS}/16, N = 5$ |            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |                                                 |                                         |                                        |                                         |                                        |                                          |                                        |                                          |                                         |                                          |                                         |                                          |                                         |                                          |                                         |                                          |
| 0011: $f_{SAMPLING} = f_{PCLK}, N = 8$          | 1011: $f_{SAMPLING} = f_{DTS}/16, N = 6$ |            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |                                                 |                                         |                                        |                                         |                                        |                                          |                                        |                                          |                                         |                                          |                                         |                                          |                                         |                                          |                                         |                                          |
| 0100: $f_{SAMPLING} = f_{DTS}/2, N = 6$         | 1100: $f_{SAMPLING} = f_{DTS}/16, N = 8$ |            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |                                                 |                                         |                                        |                                         |                                        |                                          |                                        |                                          |                                         |                                          |                                         |                                          |                                         |                                          |                                         |                                          |
| 0101: $f_{SAMPLING} = f_{DTS}/2, N = 8$         | 1101: $f_{SAMPLING} = f_{DTS}/32, N = 5$ |            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |                                                 |                                         |                                        |                                         |                                        |                                          |                                        |                                          |                                         |                                          |                                         |                                          |                                         |                                          |                                         |                                          |
| 0110: $f_{SAMPLING} = f_{DTS}/4, N = 6$         | 1110: $f_{SAMPLING} = f_{DTS}/32, N = 6$ |            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |                                                 |                                         |                                        |                                         |                                        |                                          |                                        |                                          |                                         |                                          |                                         |                                          |                                         |                                          |                                         |                                          |
| 0111: $f_{SAMPLING} = f_{DTS}/4, N = 8$         | 1111: $f_{SAMPLING} = f_{DTS}/32, N = 8$ |            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |                                                 |                                         |                                        |                                         |                                        |                                          |                                        |                                          |                                         |                                          |                                         |                                          |                                         |                                          |                                         |                                          |
| 3:2                                             | IC1PSC                                   | RW         | Input capture 1 prescaler<br>This bit field defines the prescaler ratio for the CC1 input (IC1). The prescaler is reset as soon as CC1E = 0 (GTIMx_CCER register).<br>00: No prescaler, capture is performed every time an edge is detected on the capture input.<br>01: Capture is performed every 2 events.<br>10: Capture is performed every 4 events.<br>11: Capture is performed every 8 events.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |                                                 |                                         |                                        |                                         |                                        |                                          |                                        |                                          |                                         |                                          |                                         |                                          |                                         |                                          |                                         |                                          |



| Bit field | Name | Permission | Function description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|-----------|------|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1:0       | CC1S | RW         | <p>Capture/compare 1 selection</p> <p>This bit field defines the channel direction (input/output) and the input mapping used.</p> <p>00: CC1 channel configured as output.</p> <p>01: CC1 channel configured as input, IC1 mapped to TI1.</p> <p>10: CC1 channel configured as input, IC1 mapped to TI2.</p> <p>11: CC1 channel configured as input, IC1 mapped to TRC, this mode is only valid when internal trigger input is selected via the TS bit (GTIMx_SMCR register).</p> <p><i>Attention: Data can be written to the CC1S bit only when the channel is off (CC1E = 0 in GTIMx_CCER).</i></p> |



### 16.10.9 GTIMx\_CCMR1CMP Compare mode register 1

Address offset: 0x18      Reset value: 0x0000 0000

This register is the same register as GTIMx\_CCMR1CAP and can be used in input capture mode or output compare mode. The channel direction is defined by configuring the corresponding CCyS bit. All other bits of this register function differently in input capture mode and output compare mode.

| Bit field | Name  | Permission | Function description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|-----------|-------|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:25     | RFU   | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| 24        | OC2MH | RW         | Used with OC2M, it is the highest bit of OC2M.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 23:17     | RFU   | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| 16        | OC1MH | RW         | Used with OC1M, it is the highest bit of OC1M.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 15        | OC2CE | RW         | Output compare 2 clear enabled, see OC1CE description.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 14:12     | OC2M  | RW         | Output compare 2 mode, see OC1M description.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| 11        | OC2PE | RW         | Output compare 2 preload enabled, see OC1PE description.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 10        | OC2FE | RW         | Output compare 2 fast enabled, see OC1FE description.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 9:8       | CC2S  | RW         | <p>Capture/compare 2 selection<br/>This bit field defines the channel direction (input/output) and the input mapping used.</p> <p>00: CC2 channel configured as output.<br/>01: CC2 channel configured as input, IC2 mapped to TI2.<br/>10: CC2 channel configured as input, IC2 mapped to TI1.<br/>11: CC2 channel configured as input, IC2 mapped to TRC, this mode is only valid when internal trigger input is selected via the TS bit ( GTIMx_SMCR register).</p> <p><b>Attention: Data can be written to the CC2S bit only when the channel is disabled (CC2E = 0 in GTIMx_CCER).</b></p> |
| 7         | OC1CE | RW         | <p>Output compare 1 clear enabled<br/>0: OC1REF is not affected by the OCREF_CLR_INT input.<br/>1: OC1REF is immediately cleared when a high level is detected on the OCREF_CLR_INT input.</p>                                                                                                                                                                                                                                                                                                                                                                                                  |
| 6:4       | OC1M  | RW         | <p>Output compare 1 mode<br/>These bits define the behaviour of the output reference signal OC1REF which provides OC1. OC1REF is valid at high level, while the valid level of OC1 depends on the CC1P bit.</p> <p>0000: Freeze - Comparison of output compare register GTIMx_CCR1 with counter GTIMx_CNT will not have any effect on the output. (This mode is used for time base generation).</p> <p>0001: Set channel 1 to output a valid level when matched. The OC1REF signal is forced to high level when counter GTIMx_CNT is matched with capture/compare register 1 (GTIMx_CCR1).</p>  |





| Bit field          | Name | Permission | Function description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|--------------------|------|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 6:4<br>(continued) | OC1M | RW         | <p>0010: Set channel 1 to output an invalid level when matched. When counter GTIMx_CNT matches capture/compare register 1 (GTIMx_CCR1), the OC1REF signal is forced to low level.</p> <p>0011: Flipped - OC1REF is flipped when GTIMx_CNT = GTIMx_CCR1.</p> <p>0100: Forced to invalid level - OC1REF is forced to low level.</p> <p>0101: Forced to valid level - OC1REF is forced to high level.</p> <p>0110: PWM mode 1 - In count up mode, channel 1 is valid (OC1REF = 1) as long as GTIMx_CNT &lt; GTIMx_CCR1, otherwise it is invalid (OC1REF = 0). In count down mode, channel 1 is invalid (OC1REF = 0) as long as GTIMx_CNT &gt; GTIMx_CCR1, otherwise it is valid (OC1REF = 1).</p> <p>0111: PWM mode 2 - In count up mode, channel 1 is invalid (OC1REF = 0) whenever GTIMx_CNT &lt; GTIMx_CCR1, otherwise it is valid (OC1REF = 1). In count down mode, channel 1 is valid (OC1REF = 1) as long as GTIMx_CNT &gt; GTIMx_CCR1, otherwise it is invalid (OC1REF = 0).</p> <p>1000: Retriggerable OPM mode 1 - In count up mode, the channel is in valid state until a trigger event is detected (on the TRGI signal). Then, a comparison is made in PWM mode 1 and the channel will become valid again at the next update. In count down mode, the channel is invalid until a trigger event is detected (on the TRGI signal). Then, a comparison is made in PWM mode 1 and the channel becomes invalid again at the next update.</p> <p>1001: Retriggerable OPM mode 2 - In count up mode, the channel is invalid until a trigger event is detected (on the TRGI signal). Then, a comparison is made in PWM mode 2 and the channel becomes invalid again at the next update. In count down mode, the channel is valid until a trigger event is detected (on the TRGI signal). Then, the comparison is performed in PWM mode 2 and the channel becomes valid again at the next update.</p> <p>1010: Reserved.</p> <p>1011: Count direction output, output the count direction signal of the timer on OC3 and OC4.</p> <p>1100: Combined PWM mode 1 - OC1REFC behaves the same as in PWM mode 1. OC1REFC is the logical OR of OC1REF and OC2REF.</p> <p>1101: Combined PWM mode 2 - OC1REFC behaves the same as in PWM mode 2. OC1REFC is the logical AND of OC1REF and OC2REF.</p> <p>1110: Asymmetric PWM mode 1 - OC1REFC outputs OC1REF when the counter counts up and OC2REF when the counter counts down.</p> <p>1111: Asymmetric PWM mode 2 - OC1REFC outputs OC1REF when the counter counts up and OC2REF when the counter counts down.</p> <p><i>Note: In PWM mode, OCyREF level will be changed only when the comparison result is changed or the output comparison mode is switched from 'Freeze' mode to 'PWM' mode.</i></p> |



| Bit field | Name  | Permission | Function description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|-----------|-------|------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 3         | OC1PE | RW         | <p>Output compare 1 preload enabled</p> <p>0: Disable the preload register related to GTIMx_CCR1. Data can be written to GTIMx_CCR1 at any time and the new value will be used immediately after writing.</p> <p>1: Enable the preload register related to GTIMx_CCR1. Read/write access to the preloaded registers. The GTIMx_CCR1 preloaded value is loaded into the valid registers each time that an update event is generated.</p> <p><i>Attention: PWM mode without validation of the preload register is only possible in single pulse mode (OPM bit field in the GTIMx_CR1 register is set to 1). The behaviour is not guaranteed in other cases.</i></p>                                                                          |
| 2         | OC1FE | RW         | <p>Output compare 1 fast enabled</p> <p>This bit is used to speed up the effect of a trigger input event on the CC output.</p> <p>0: Even if the trigger is on, CC1 will work properly according to the counter and CCR1 values. When an edge is present on the trigger input, the minimum delay to activate the CC1 output is 5 clock cycles.</p> <p>1: The presence of a valid edge on the trigger input corresponds to a comparison match on the CC1 output. Subsequently, OC is set to the compare level regardless of the comparison result. The delay time between sampling the trigger input and activating the CC1 output is reduced to 3 clock cycles. OCyFE works only when the channel is configured for PWM1 or PWM2 mode.</p> |
| 1:0       | CC1S  | RW         | <p>Capture/compare 1 selection</p> <p>This bit field defines the channel direction (input/output) and the input used.</p> <p>00: CC1 channel configured as output.</p> <p>01: CC1 channel configured as input, IC1 mapped to TI1.</p> <p>10: CC1 channel configured as input, IC1 mapped to TI2.</p> <p>11: CC1 channel configured as input, IC1 mapped to TRC, this mode is only valid when the internal trigger input is selected via the TS bit (GTIMx_SMCR register).</p> <p><i>Attention: Data can be written to the CC1S bit only when the channel is off (CC1E = 0 in GTIMx_CCER).</i></p>                                                                                                                                          |



### 16.10.10 GTIMx\_CCMR2CAP Capture mode register 2

Address offset: 0x1C      Reset value: 0x0000 0000

This register is the same register as GTIMx\_CCMR2CMP and can be used in input capture mode or output compare mode. The channel direction is defined by configuring the corresponding CCyS bit. All other bits of this register function differently in input capture mode and output compare mode.

| Bit field | Name   | Permission | Function description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|-----------|--------|------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:16     | RFU    | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 15:12     | IC4F   | RW         | Input capture 4 filter, see IC1F description.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 11:10     | IC4PSC | RW         | Input capture 4 prescaler, see IC1PSC description.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 9:8       | CC4S   | RW         | Capture/compare 4 selection<br>This bit field defines the channel direction (input/output) and the input mapping used.<br>00: CC4 channel configured as output.<br>01: CC4 channel configured as input, IC4 mapped to TI4.<br>10: CC4 channel configured as input, IC4 mapped to TI3.<br>11: CC4 channel configured as input, IC4 mapped to TRC, this mode is only valid when internal trigger input is selected via the TS bit (GTIMx_SMCR register).<br><i>Attention: Data can be written to the CC4S bit only when the channel is off (CC4E = 0 in GTIMx_CCER).</i> |
| 7:4       | IC3F   | RW         | Input capture 3 filter, see IC1F description.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 3:2       | IC3PSC | RW         | Input capture 3 prescaler, see IC1PSC description.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 1:0       | CC3S   | RW         | Capture/compare 3 selection<br>This bit field defines the channel direction (input/output) and the input mapping used.<br>00: CC3 channel configured as output.<br>01: CC3 channel configured as input, IC3 mapped to TI3.<br>10: CC3 channel configured as input, IC3 mapped to TI4.<br>11: CC3 channel configured as input, IC3 mapped to TRC, this mode is only valid when internal trigger input is selected via the TS bit (GTIMx_SMCR register).<br><i>Attention: Data can be written to the CC3S bit only when the channel is off (CC3E = 0 in GTIMx_CCER).</i> |



### 16.10.11 GTIMx\_CCMR2CMP Compare mode register 2

Address offset: 0x1C      Reset value: 0x0000 0000

This register is the same register as GTIMx\_CCMR2CAP and can be used in input capture mode or output compare mode. The channel direction is defined by configuring the corresponding CCyS bit. All other bits of this register function differently in input capture mode and output compare mode.

| Bit field | Name  | Permission | Function description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|-----------|-------|------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:25     | RFU   | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 24        | OC4MH | RW         | Used with OC4M, it is the highest bit of the OC4M.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 23:17     | RFU   | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 16        | OC3MH | RW         | Used with OC3M, it is the highest bit of the OC3M.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 15        | OC4CE | RW         | Output compare 4 clear enabled, see OC1CE description.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 14:12     | OC4M  | RW         | Output compare 4 mode, see OC1M description.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 11        | OC4PE | RW         | Output compare 4 preload enabled, see OC1PE description.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 10        | OC4FE | RW         | Output compare 4 fast enabled, see OC1FE description.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 9:8       | CC4S  | RW         | Capture/compare 4 selection<br>This bit field defines the channel direction (input/output) and the input used.<br>00: CC4 channel configured as output.<br>01: CC4 channel configured as input, IC4 mapped to TI4.<br>10: CC4 channel configured as input, IC4 mapped to TI3.<br>11: CC4 channel configured as input, IC4 mapped to TRC, this mode is only valid when the internal trigger input is selected via the TS bit (GTIMx_SMCR register).<br><i>Attention: Data can be written to the CC4S bit only when the channel is off (CC4E = 0 in GTIMx_CCER).</i> |
| 7         | OC3CE | RW         | Output compare 3 clear to enabled, see OC1CE description.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 6:4       | OC3M  | RW         | Output compare 3 mode, see OC1M description.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 3         | OC3PE | RW         | Output compare 3 preload enabled, see OC1PE description.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 2         | OC3FE | RW         | Output compare 3 fast enabled, see OC1FE description.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 1:0       | CC3S  | RW         | Capture/compare 3 selection<br>This bit field defines the channel direction (input/output) and the input used.<br>00: CC3 channel configured as output.<br>01: CC3 channel configured as input, IC3 mapped to TI3.<br>10: CC3 channel configured as input, IC3 mapped to TI4.<br>11: CC3 channel configured as input, IC3 mapped to TRC. This mode is only valid when the internal trigger input is selected via the TS bit (GTIMx_SMCR register).<br><i>Attention: Data can be written to the CC3S bit only when the channel is off (CC3E = 0 in GTIMx_CCER).</i> |



### 16.10.12 GTIMx\_CCER Capture/compare enable register

Address offset: 0x20

Reset value: 0x0000 0000

| Bit field | Name  | Permission | Function description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|-----------|-------|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:16     | RFU   | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 15        | CC4NP | RW         | Capture/compare 4 complementary output polarity, see CC1NP.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 14        | RFU   | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 13        | CC4P  | RW         | Capture/compare 4 output polarity, see CC1P description.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 12        | CC4E  | RW         | Capture/compare 4 output enabled, see CC1E description.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| 11        | CC3NP | RW         | Capture/compare 3 complementary output polarity, see CC1NP.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 10        | RFU   | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 9         | CC3P  | RW         | Capture/compare 3 output polarity, see CC1P description.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 8         | CC3E  | RW         | Capture/compare 3 output enabled, see CC1E description.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| 7         | CC2NP | RW         | Capture/compare 2 complementary output polarity, see CC1NP description.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| 6         | RFU   | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 5         | CC2P  | RW         | Capture/compare 2 output polarity, see CC1P description.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 4         | CC2E  | RW         | Capture/compare 2 output enabled, see CC1E description.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| 3         | CC1NP | RW         | Capture/compare 1 complementary output polarity<br>CC1 channel configured as output: in this case CC1NP must be held clear.<br>CC1 channel configured as input: this bit is used with CC1P to define TI1FP1/TI2FP1 polarity. See CC1P description.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| 2         | RFU   | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 1         | CC1P  | RW         | Capture/compare 1 output polarity<br>0: OC1 is valid at high level (output mode)/edge sensitivity selection (input mode).<br>1: OC1 is valid at low level (output mode)/edge sensitivity selection (input mode).<br>When the CC1 channel is configured as an input, the CC1NP/CC1P bits select the valid polarity of TI1FP1 and TI2FP1 for trigger or capture operations. <ul style="list-style-type: none"> <li>CC1NP = 0, CC1P = 0: Non-inverting/rising edge trigger. The circuit is sensitive to the rising edge of TIxFP1 (performing a capture or trigger operation in reset mode, external clock mode, or trigger mode), and TIxFP1 is not inverted (performing a trigger operation in gated mode or encoder mode).</li> </ul> |



| Bit field        | Name | Permission | Function description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|------------------|------|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1<br>(continued) | CC1P | RW         | <ul style="list-style-type: none"> <li>● CC1NP = 0, CC1P = 1: Inverted/falling edge trigger. Circuit is sensitive to TlxFP1 falling edge (performs capture or trigger operation in reset mode, external clock mode, or trigger mode) and TlxFP1 is inverted (performs trigger operation in gated mode or encoder mode).</li> <li>● CC1NP = 1, CC1P = 1: Non-inverting/dual edge trigger. The circuit is sensitive to both the rising and falling edges of TlxFP1 (performs a capture or trigger operation in reset mode, external clock mode, or trigger mode), and TlxFP1 is not inverted (performs a trigger operation in gated mode).<br/><i>Attention: This configuration must not be used in encoder mode.</i></li> <li>● CC1NP = 1, CC1P = 0: This configuration is reserved and must not be used.</li> </ul> |
| 0                | CC1E | RW         | <p>Capture/compare 1 output enabled</p> <p>0: Capture mode disabled/OC1 not active.</p> <p>1: Capture mode enabled/OC1 signal output on corresponding output pin.</p> <p><i>Note: This bit must be configured to 1 when using the input capture feature.</i></p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |



### 16.10.13 GTIMx\_CNT Count register

Address offset: 0x24      Reset value: 0x0000 0000

| Bit field | Name   | Permission | Function description                                                                                                                                                                                                                                                                                                                          |
|-----------|--------|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31        | UIFCPY | RO         | According to the value of the UIFREMAP bit field in the GTIMx_CR1 register, this bit field indicates different meanings: <ul style="list-style-type: none"> <li>UIFREMAP = 0, this bit field is reserved and read as 0.</li> <li>UIFREMAP = 1, this bit field indicates a read-only copy of the UIF bit of the GTIMx_ISR register.</li> </ul> |
| 30:16     | RFU    | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                                 |
| 15:0      | CNT    | RW         | Count value.                                                                                                                                                                                                                                                                                                                                  |

### 16.10.14 GTIMx\_PSC Prescaler register

Address offset: 0x28      Reset value: 0x0000 0000

| Bit field | Name | Permission | Function description                                                                                                                                                                                                                                                                                                                                             |
|-----------|------|------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:16     | RFU  | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                                                    |
| 15:0      | PSC  | RW         | Prescaler value<br>The counter clock frequency CK_CNT is equal to:<br>$f_{CK\_PSC} / (PSC[15:0] + 1).$ The PSC contains the value to be loaded into the valid prescaler register each time an update event occurs, including when the counter is cleared by the UG bit in the GTIMx_EGR register, or by the trigger controller when configured for 'reset mode'. |



### 16.10.15 GTIMx\_ARR Auto-reload register

Address offset: 0x2C      Reset value: 0x0000 FFFF

| Bit field | Name | Permission | Function description                                                                                                                                  |
|-----------|------|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:16     | RFU  | -          | Reserved bits, please keep the default value.                                                                                                         |
| 15:0      | ARR  | RW         | Auto reload value<br>ARR is the value to be loaded into the real auto reload register, the counter does not work when the auto reload value is empty. |

### 16.10.16 GTIMx\_CCR1 Capture/compare register 1

Address offset: 0x34      Reset value: 0x0000 0000

| Bit field | Name | Permission | Function description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|-----------|------|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:16     | RFU  | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 15:0      | CCR1 | RW         | <p>Capture/compare 1 value<br/>If channel CC1 is configured to output: CCR1 is the preload value for capture/compare register 1.</p> <ul style="list-style-type: none"> <li>This value takes effect immediately if the preload function is not enabled via the OC1PE bit in the GTIMx_CCMR1CMP register; otherwise it takes effect only when an update event occurs (copied to the valid capture/compare register 1). The effective capture/compare register contains the value to be compared with counter GTIMx_CNT and signalled on the OC1 output.</li> </ul> <p>If channel CC1 is configured as input: CCR1 is the counter value at the time of the last input capture 1 event (IC1). Only the GTIMx_CCR1 register can be read, it cannot be programmed.</p> |





### 16.10.17 GTIMx\_CCR2 Capture/compare register 2

Address offset: 0x38

Reset value: 0x0000 0000

| Bit field | Name | Permission | Function description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|-----------|------|------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:16     | RFU  | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 15:0      | CCR2 | RW         | <p>Capture/compare 2 value</p> <p>If channel CC2 is configured to output: CCR2 is the preload value for capture/compare register 2.</p> <ul style="list-style-type: none"> <li>This value takes effect immediately if the preload function is not enabled via the OC2PE bit in the GTIMx_CCMR1CMP register; otherwise it takes effect only when an update event occurs (copied to the valid capture/compare register 2). The effective capture/compare register contains the value to be compared with counter GTIMx_CNT and signalled on the OC2 output.</li> </ul> <p>If channel CC2 is configured as input: CCR2 is the counter value at the time of the last input capture 2 event (IC2). Only the GTIMx_CCR2 register can be read, it cannot be programmed.</p> |

### 16.10.18 GTIMx\_CCR3 Capture/compare register 3

Address offset: 0x3C

Reset value: 0x0000 0000

| Bit field | Name | Permission | Function description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|-----------|------|------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:16     | RFU  | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| 15:0      | CCR3 | RW         | <p>Capture/compare 3 value</p> <p>If channel CC3 is configured to output: CCR3 is the preload value for capture/compare register 3.</p> <ul style="list-style-type: none"> <li>This value takes effect immediately if the preload function is not enabled via the OC3PE bit in the GTIMx_CCMR2CMP register; otherwise it takes effect only when an update event occurs (copied to valid capture/compare register 3). The effective capture/compare register contains the value to be compared with counter GTIMx_CNT and signalled on the OC3 output.</li> </ul> <p>If channel CC3 is configured as input: CCR3 is the counter value at the time of the last Input Capture 3 event (IC3). Only the GTIMx_CCR3 register can be read, it cannot be programmed.</p> |



### 16.10.19 GTIMx\_CCR4 Capture/compare register 4

Address offset: 0x40

Reset value: 0x0000 0000

| Bit field | Name | Permission | Function description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|-----------|------|------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:16     | RFU  | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 15:0      | CCR4 | RW         | <p>Capture/compare 4 value</p> <p>If channel CC4 is configured to output: CCR4 is the preload value for capture/compare register 4.</p> <ul style="list-style-type: none"> <li>This value takes effect immediately if the preload function is not enabled via the OC4PE bit in the GTIMx_CCMR2CMP register; otherwise it takes effect only when an update event occurs (copied to the valid capture/compare register 4). The effective capture/compare register contains the value to be compared with counter GTIMx_CNT and signalled on the OC4 output.</li> </ul> <p>If channel CC4 is configured as input: CCR4 is the counter value at the time of the last input capture 4 event (IC4). Only the GTIMx_CCR4 register can be read, it cannot be programmed.</p> |



### 16.10.20 GTIMx\_ECR Encoder control register

Address offset: 0x58

Reset value: 0x0000 0000

| Bit field | Name | Permission | Function description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|-----------|------|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:8      | RFU  | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| 7:6       | IPOS | RW         | <p>Coding index position</p> <p>In quadrature encoder mode (SMS[3:0] = 0001, 0010, 0011, 1110, 1111), this bit indicates in which AB input configuration the index event resets the counter.</p> <p>00: When AB = 00, the index resets the counter.</p> <p>01: When AB = 01, the index resets the counter.</p> <p>10: When AB = 10, the index resets the counter.</p> <p>11: When AB = 11, the index resets the counter.</p> <p>In directional clock encoder mode or clock plus direction encoder mode (SMS[3:0] = 1010, 1011, 1100, 1101), these bits indicate at which level the index event resets the counter. In directional clock encoder mode, this applies to both clock inputs.</p> <p>x0: when the clock signal is low, the index resets the counter.</p> <p>x1: When the clock signal is high, the index resets the counter.</p> |
| 5         | FIDX | RW         | <p>This bit indicates whether only the first index is considered</p> <p>0: Index is always active.</p> <p>1: Only the first index resets the counter.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| 4:3       | RFU  | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| 2:1       | IDIR | RW         | <p>Index direction</p> <p>This bit field indicates in which count direction the index resets the counter.</p> <p>00: Reset counter in any counting direction.</p> <p>01: Reset counter only when counting up.</p> <p>10: Reset counter only when counting down.</p> <p>11: Reserved.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| 0         | IE   | RW         | <p>Index enabled</p> <p>0: Index disabled.</p> <p>1: Index enabled.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |



### 16.10.21 GTIMx\_TISEL TI input selection register

Address offset: 0x5C

Reset value: 0x0000 0000

| Bit field | Name   | Permission | Function description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|-----------|--------|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:28     | RFU    | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 27:24     | TI4SEL | RW         | TI4 channel input capture signal source selection<br>0000: GTIMx_CH4                      1000: BTIM1_Trigo<br>0001: VC3_OUT                      1001: BTIM2_Trigo<br>0010: VC4_OUT                      1010: BTIM3_Trigo<br>0011: UART1_TXD                    1011: GTIM1_Trigo<br>0100: UART2_TXD                    1100: GTIM2_Trigo<br>0101: UART3_TXD                    1101: GTIM3_Trigo<br>0110: LPTIM_PWM                   1110: GTIM4_Trigo<br>0111: HALLTIM_Trigo                1111: ATIM_Trigo<br><i>Attention: Prohibits the selection of Trigo signal output by GTIMx itself.</i>  |
| 23:20     | RFU    | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 19:16     | TI3SEL | RW         | TI3 channel input capture signal source selection<br>0000: GTIMx_CH3                      1000: BTIM1_Trigo<br>0001: VC1_OUT                      1001: BTIM2_Trigo<br>0010: VC2_OUT                      1010: BTIM3_Trigo<br>0011: UART1_RXD                    1011: GTIM1_Trigo<br>0100: UART2_RXD                    1100: GTIM2_Trigo<br>0101: UART3_RXD                    1101: GTIM3_Trigo<br>0110: LPTIM_PWM                   1110: GTIM4_Trigo<br>0111: HALLTIM_Trigo                1111: ATIM_Trigo<br><i>Attention: Prohibits the selection of Trigo signal output by GTIMx itself.</i>  |
| 15:12     | RFU    | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 11:8      | TI2SEL | RW         | TI2 channel input capture signal source selection<br>0000: GTIMx_CH2                      1000: BTIM1_Trigo<br>0001: VC3_OUT                      1001: BTIM2_Trigo<br>0010: VC4_OUT                      1010: BTIM3_Trigo<br>0011: UART1_TXD                    1011: GTIM1_Trigo<br>0100: UART2_TXD                    1100: GTIM2_Trigo<br>0101: UART3_TXD                    1101: GTIM3_Trigo<br>0110: RTC_OUT                      1110: GTIM4_Trigo<br>0111: HALLTIM_Trigo                1111: ATIM_Trigo<br><i>Attention: Prohibits the selection of Trigo signal output by GTIMx itself.</i> |
| 7:4       | RFU    | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |



| Bit field          | Name             | Permission | Function description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |                 |                  |               |                  |               |                  |                 |                  |                 |                  |                 |                  |               |                  |                    |                 |
|--------------------|------------------|------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------|------------------|---------------|------------------|---------------|------------------|-----------------|------------------|-----------------|------------------|-----------------|------------------|---------------|------------------|--------------------|-----------------|
| 3:0                | TI1SEL           | RW         | <div>TI1 channel input capture signal source selection</div> <table><tr><td>0000: GTIMx_CH1</td><td>1000: BTIM1_Trgo</td></tr><tr><td>0001: VC1_OUT</td><td>1001: BTIM2_Trgo</td></tr><tr><td>0010: VC2_OUT</td><td>1010: BTIM3_Trgo</td></tr><tr><td>0011: UART1_RXD</td><td>1011: GTIM1_Trgo</td></tr><tr><td>0100: UART2_RXD</td><td>1100: GTIM2_Trgo</td></tr><tr><td>0101: UART3_RXD</td><td>1101: GTIM3_Trgo</td></tr><tr><td>0110: MCO_OUT</td><td>1110: GTIM4_Trgo</td></tr><tr><td>0111: HALLTIM_Trgo</td><td>1111: ATIM_Trgo</td></tr></table> <div><i>Attention: Prohibits the selection of Trgo signal output by GTIMx itself.</i></div> | 0000: GTIMx_CH1 | 1000: BTIM1_Trgo | 0001: VC1_OUT | 1001: BTIM2_Trgo | 0010: VC2_OUT | 1010: BTIM3_Trgo | 0011: UART1_RXD | 1011: GTIM1_Trgo | 0100: UART2_RXD | 1100: GTIM2_Trgo | 0101: UART3_RXD | 1101: GTIM3_Trgo | 0110: MCO_OUT | 1110: GTIM4_Trgo | 0111: HALLTIM_Trgo | 1111: ATIM_Trgo |
| 0000: GTIMx_CH1    | 1000: BTIM1_Trgo |            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |                 |                  |               |                  |               |                  |                 |                  |                 |                  |                 |                  |               |                  |                    |                 |
| 0001: VC1_OUT      | 1001: BTIM2_Trgo |            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |                 |                  |               |                  |               |                  |                 |                  |                 |                  |                 |                  |               |                  |                    |                 |
| 0010: VC2_OUT      | 1010: BTIM3_Trgo |            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |                 |                  |               |                  |               |                  |                 |                  |                 |                  |                 |                  |               |                  |                    |                 |
| 0011: UART1_RXD    | 1011: GTIM1_Trgo |            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |                 |                  |               |                  |               |                  |                 |                  |                 |                  |                 |                  |               |                  |                    |                 |
| 0100: UART2_RXD    | 1100: GTIM2_Trgo |            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |                 |                  |               |                  |               |                  |                 |                  |                 |                  |                 |                  |               |                  |                    |                 |
| 0101: UART3_RXD    | 1101: GTIM3_Trgo |            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |                 |                  |               |                  |               |                  |                 |                  |                 |                  |                 |                  |               |                  |                    |                 |
| 0110: MCO_OUT      | 1110: GTIM4_Trgo |            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |                 |                  |               |                  |               |                  |                 |                  |                 |                  |                 |                  |               |                  |                    |                 |
| 0111: HALLTIM_Trgo | 1111: ATIM_Trgo  |            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |                 |                  |               |                  |               |                  |                 |                  |                 |                  |                 |                  |               |                  |                    |                 |



## 16.10.22 GTIMx\_AF1 Multiplexing function option register 1

Address offset: 0x60

Reset value: 0x0000 0000

| Bit field | Name   | Permission | Function description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|-----------|--------|------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:18     | RFU    | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 17:14     | ETRSEL | RW         | <b>ETR source selection for GTIM1</b><br>0000: GTIM1_ETR                      1000: BTIM1_ETR<br>0001: VC1_OUT                      1001: BTIM2_ETR<br>0010: VC2_OUT                      1010: BTIM3_ETR<br>0011: VC3_OUT                      1011: Reserved<br>0100: VC4_OUT                      1100: GTIM2_ETR<br>0101: ADC1_AWD                    1101: GTIM3_ETR<br>0110: ADC2_AWD                    1110: GTIM4_ETR<br>0111: LVD_OUT                      1111: LSE_FAULT<br><b>ETR source selection for GTIM2</b><br>0000: GTIM2_ETR                    1000: BTIM1_ETR<br>0001: VC1_OUT                      1001: BTIM2_ETR<br>0010: VC2_OUT                      1010: BTIM3_ETR<br>0011: VC3_OUT                      1011: GTIM1_ETR<br>0100: VC4_OUT                      1100: Reserved<br>0101: ADC1_AWD                    1101: GTIM3_ETR<br>0110: ADC2_AWD                    1110: GTIM4_ETR<br>0111: LVD_OUT                      1111: LSE_FAULT<br><b>ETR source selection for GTIM3</b><br>0000: GTIM3_ETR                    1000: BTIM1_ETR<br>0001: VC1_OUT                      1001: BTIM2_ETR<br>0010: VC2_OUT                      1010: BTIM3_ETR<br>0011: VC3_OUT                      1011: GTIM1_ETR<br>0100: VC4_OUT                      1100: GTIM2_ETR<br>0101: ADC1_AWD                    1101: Reserved<br>0110: ADC2_AWD                    1110: GTIM4_ETR<br>0111: LVD_OUT                      1111: HSE_FAULT<br><b>ETR source selection for GTIM4</b><br>0000: GTIM4_ETR                    1000: BTIM1_ETR<br>0001: VC1_OUT                      1001: BTIM2_ETR<br>0010: VC2_OUT                      1010: BTIM3_ETR<br>0011: VC3_OUT                      1011: GTIM1_ETR<br>0100: VC4_OUT                      1100: GTIM2_ETR<br>0101: ADC1_AWD                    1101: GTIM3_ETR<br>0110: ADC2_AWD                    1110: Reserved<br>0111: LVD_OUT                      1111: HSE_FAULT |
| 13:0      | RFU    | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |



### 16.10.23 GTIMx\_AF2 Multiplexing function option register 2

Address offset: 0x64

Reset value: 0x0000 0000

| Bit field | Name   | Permission | Function description                                                                                                                                      |
|-----------|--------|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:19     | RFU    | -          | Reserved bits, please keep the default value.                                                                                                             |
| 18:16     | OCRSEL | RW         | OCREF_CLR source selection<br>000: VC1_OUT<br>001: VC2_OUT<br>010: VC3_OUT<br>011: VC4_OUT<br>100: ADC1_AWD<br>101: ADC2_AWD<br>110: LVD_OUT<br>111: ETRF |
| 15:0      | RFU    | -          | Reserved bits, please keep the default value.                                                                                                             |



## 17 Advanced-control timer (ATIM)

### 17.1 Overview

The advanced timer (ATIM) consists of a 16bit auto-reload counter driven by a programmable prescaler. ATIM supports a variety of operating modes such as timing, counting, reset, gating, triggering, and encoder with 6 independent capture/compare channels for 6 independent PWM outputs or 6 pairs of complementary PWM outputs with dead time, or capture of 6 inputs. It can be used for basic timing/counting, measuring the pulse width and period of input signals, and generating output waveforms (PWM, single pulse, complementary PWM with dead time insertion, etc.).





## 17.2 Main features

- 16bit counting up, down and up/down auto-reload counter.
- Programmable prescaler supports 1, 2, 3, 4, ..., 65536 divider frequencies.
- Support single count mode and continuous count mode.
- 6 independent input capture and output compare channels.
- Dead time programmable complementary PWM output.
- Supports double-point comparison phase shift and logic and phase shift.
- Supports 2 brake inputs.
- Supports 1 brake output.
- Trigger input signal (TRGI) control timer for multiple slave modes.
- Timer cascade ITR and on-chip peripheral interconnect ETR.
- Supports incremental (quadrature) encoder and Hall sensor circuits for positioning.
- Generates interrupt/DMA requests when multiple events occur:
  - Update event
  - Trigger event
  - Input capture
  - Output compare
  - Reversal event
  - Brake event
  - Encoder event

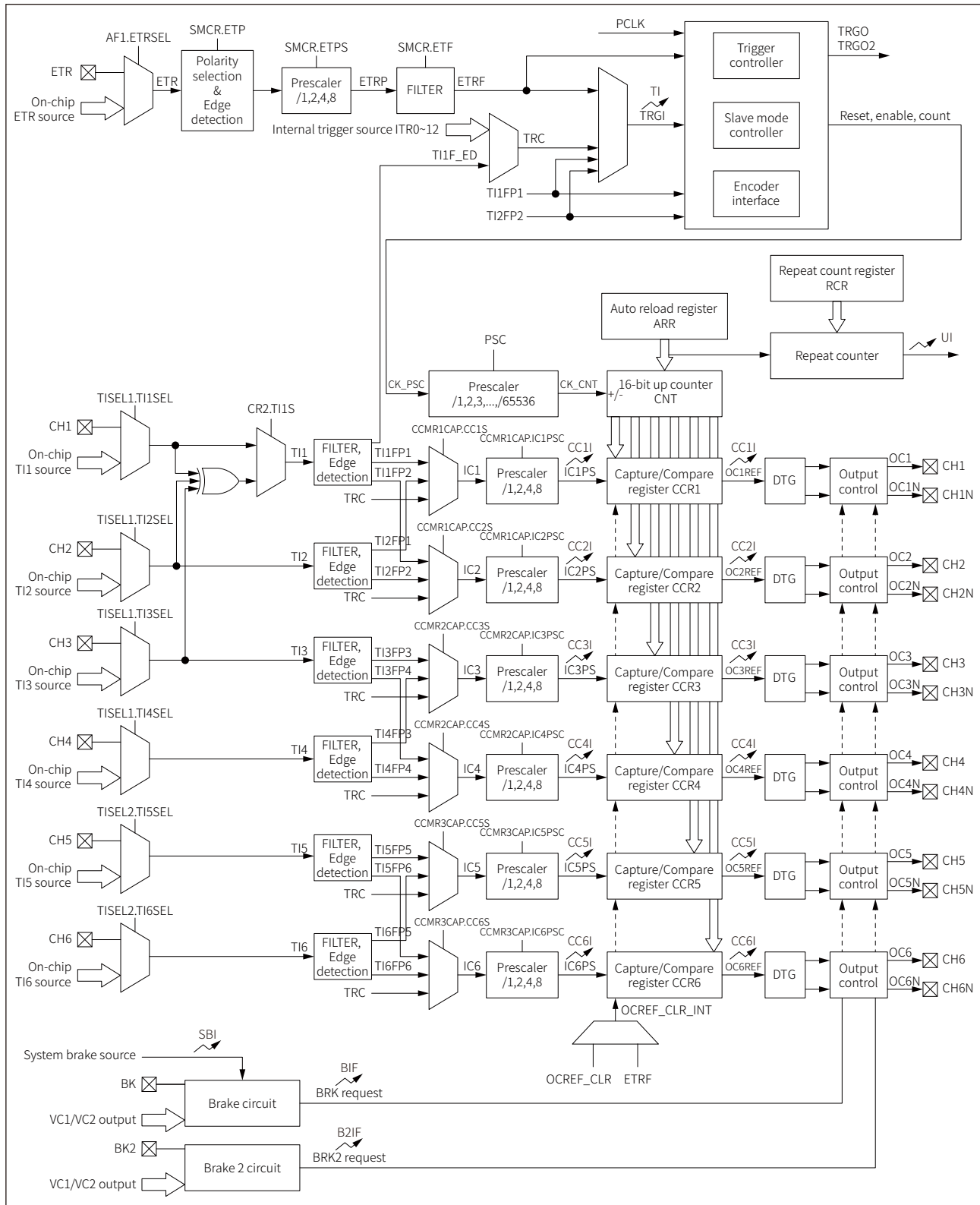


## 17.3 Functional description

### 17.3.1 Functional block diagram

The functional block diagram of ATIM is shown in the following figure:

Figure 17-1 ATIM functional block diagram



### 17.3.1.1 Clock source

The counter's counting clock source can be provided by the internal clock PCLK, the trigger signal TRGI or the external trigger input signal ETRF, which is divided by the prescaler ATIM\_PSC to drive the counter for counting. Different clock sources are available in different operating modes, specific please refer to [17.3.2 Operating modes](#).

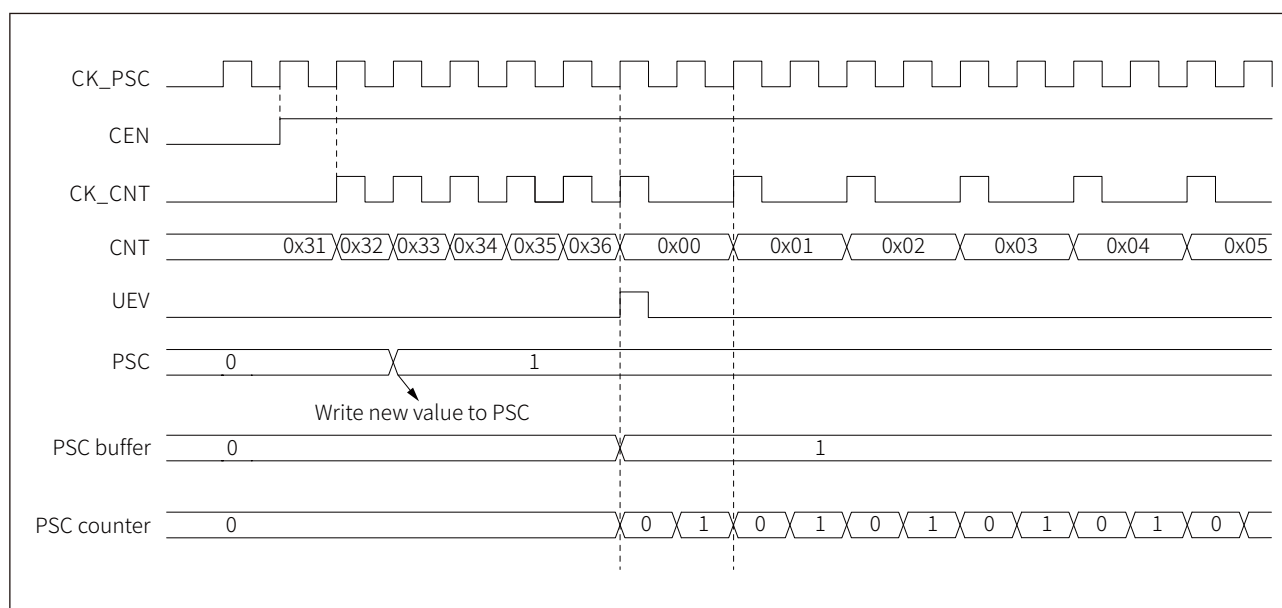
### 17.3.1.2 Prescaler

The prescaler divides the CK\_PSC clock to obtain the count clock CK\_CNT to drive the counter count. The divider coefficient is set through the ATIM\_PSC register, which supports 1, 2, 3, 4, ..., 65536 divider.

The ATIM\_PSC register is buffered and can be modified during runtime, and the new prescaler value will take effect at the next update event.

The following figure shows the timing diagram when the division frequency of the prescaler is changed from 1 to 2 during operation:

Figure 17-2 Timing diagram when the prescaler division frequency changes from 1 to 2



### 17.3.1.3 Counter and count mode

The counter can be set to count up (edge-aligned mode), count down (edge-aligned mode), or count up/down (center-aligned mode). Specifically, it is configured through the CMS and DIR bit fields of the control register ATIM\_CR1, as shown in the following table:

Table 17-1 Count mode

| ATIM_CR1.CMS | ATIM_CR1.DIR | Count mode                     |
|--------------|--------------|--------------------------------|
| 00           | 0            | Count up (edge-aligned mode)   |
|              | 1            | Count down (edge-aligned mode) |
| 01           | -            | Center-aligned mode 1          |
| 10           | -            | Center-aligned mode 2          |
| 11           | -            | Center-aligned mode 3          |

When the CEN bit field of the ATIM\_CR1 register is set to 1, the counter starts counting in the set mode, noting that the counter will start counting one clock cycle after the CEN bit is set to 1.

#### Count up mode

In count up mode, the counter starts counting up from 0 to the reload value ARR, and then starts counting up again from 0, generating an overflow event on the counter and generating an overflow signal OV (the OV signal is cleared automatically).

If a repeat counter is used, an update event UEV will be generated when the number of overflows reaches the repeat value plus 1 (REP+1) (the UEV signal will be cleared automatically), otherwise an update event UEV will be generated each time the counter overflows, and the counter update interrupt flag bit, ATIM\_ISR.UIF, is set by hardware, so that if an interrupt is permitted (setting ATIM\_DIER.UIE to 1), an interrupt request will be generated, the flag bit is cleared by setting ATIM\_ICR.UIF to 0.

The following is an example of counter operation at different clock frequencies, where ATIM\_ARR = 0x36, ATIM\_RCR = 0x00:

Figure 17-3 Count up, internal clock frequency division factor is 1

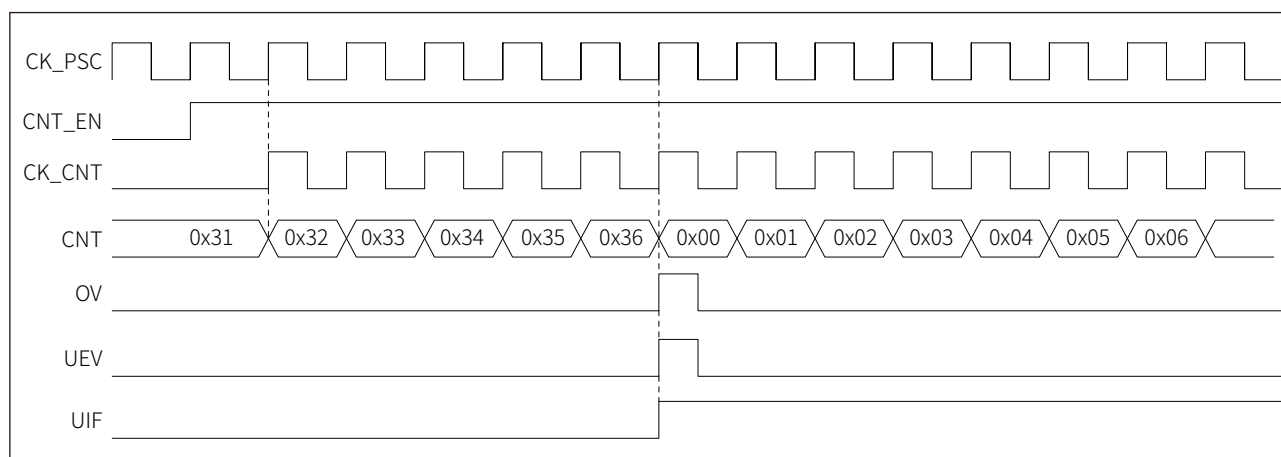


Figure 17-4 Count up, internal clock frequency division factor is 2

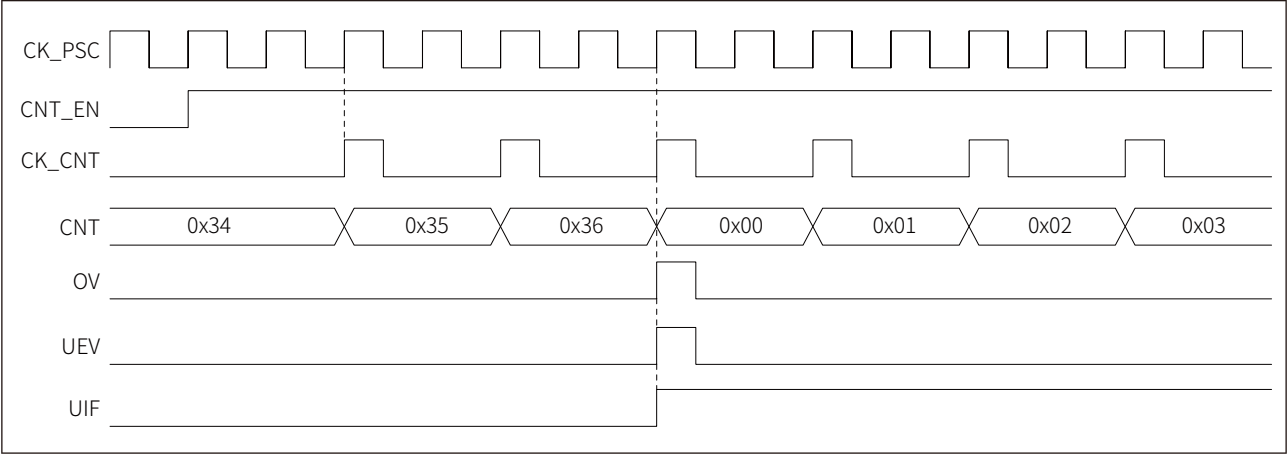


Figure 17-5 Count up, internal clock frequency division factor is 4

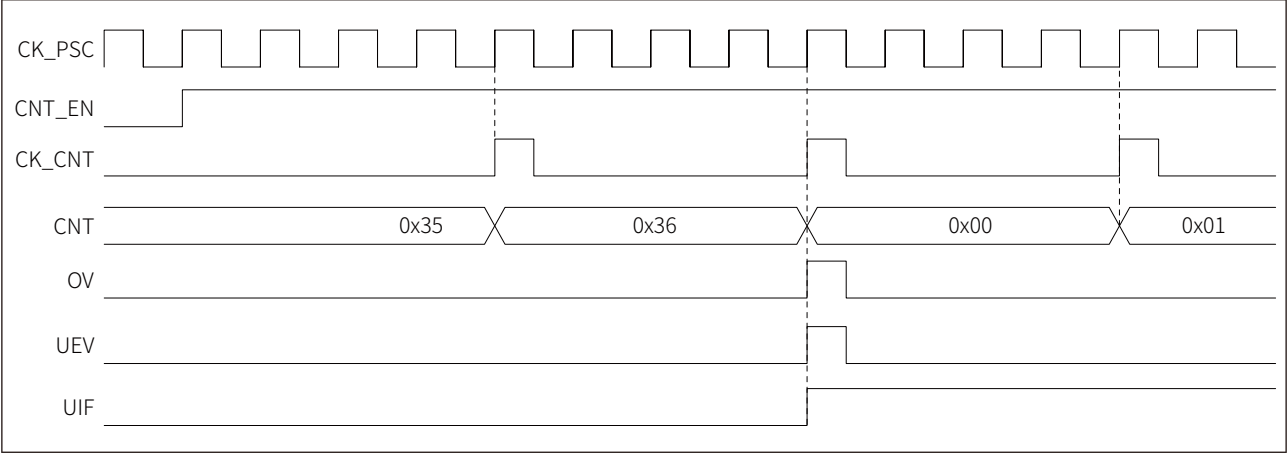


Figure 17-6 Count up, internal clock frequency division factor is N

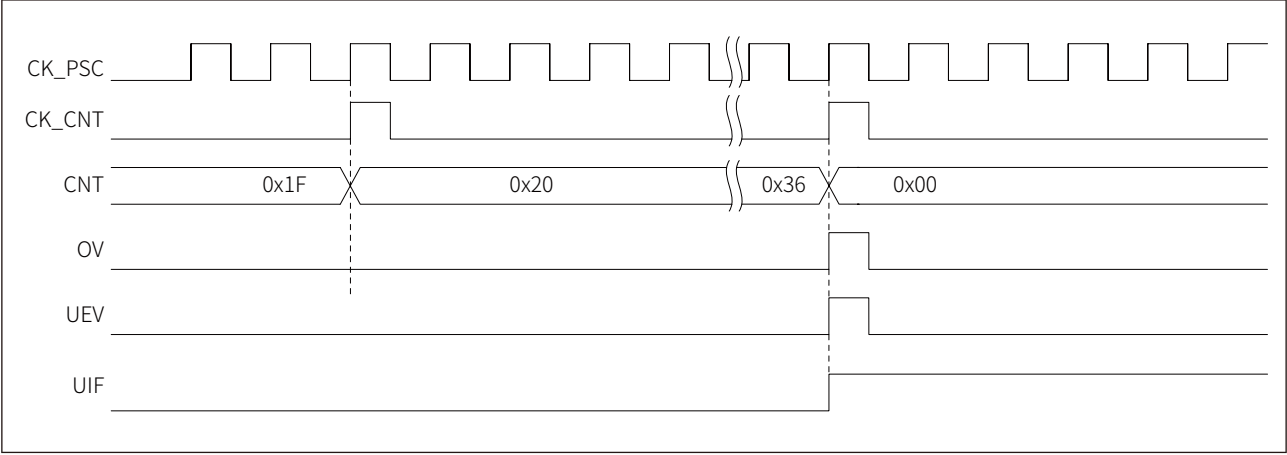


Figure 17-7 Count up, update event when reloading cache is disabled (ATIM\_CR1.ARPE = 0)

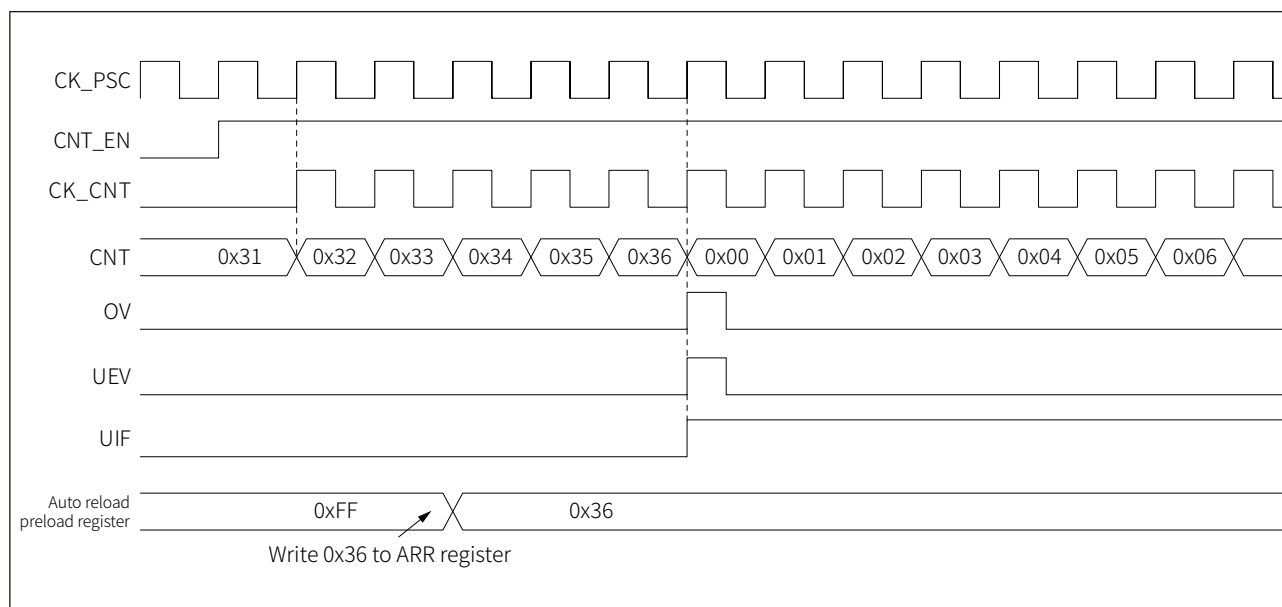
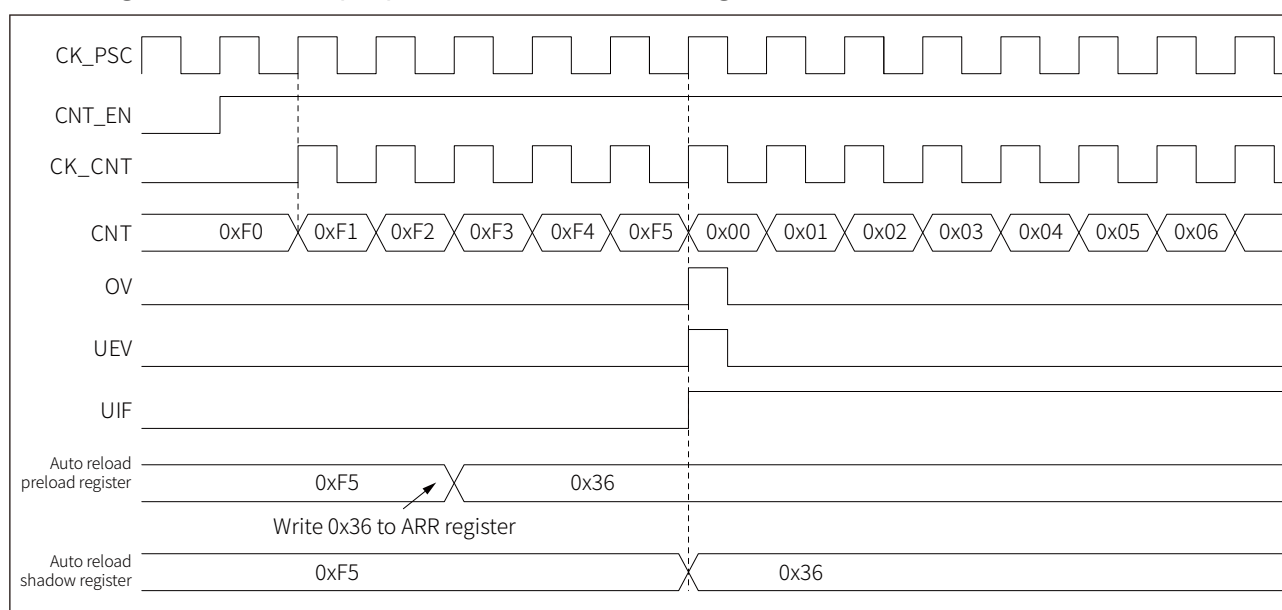


Figure 17-8 Count up, update event when reloading cache is enabled (ATIM\_CR1.ARPE = 1)



## Count down mode

The counter operates in count down mode when the DIR bit of control register ATIM\_CR1 is set to 1 in edgealigned mode.

In count down mode, the hardware automatically loads the ARR value into the counter CNT, the counter starts counting down to 0, and then reloads the ARR value to count down while generating a counter underflow event and generating the underflow signal UND (the UND signal is cleared automatically).

If a repeat counter is used, an update event UEV will be generated when the number of overflows reaches the repeat value plus 1 (REP+1) (the UEV signal will be cleared automatically), otherwise an update event UEV will be generated each time the counter underflows, and the counter update interrupt flag bit, ATIM\_ISR.UIF, is set by hardware, so that if an interrupt is permitted (setting ATIM\_DIER.UIE to 1), an interrupt request will be generated, the flag bit is cleared by setting ATIM\_ICR.UIF to 0.



The following is an example of counter operation at different clock frequencies, where ATIM\_ARR = 0x36, ATIM\_RCR = 0x00:

Figure 17-9 Count down, internal clock frequency division factor is 1

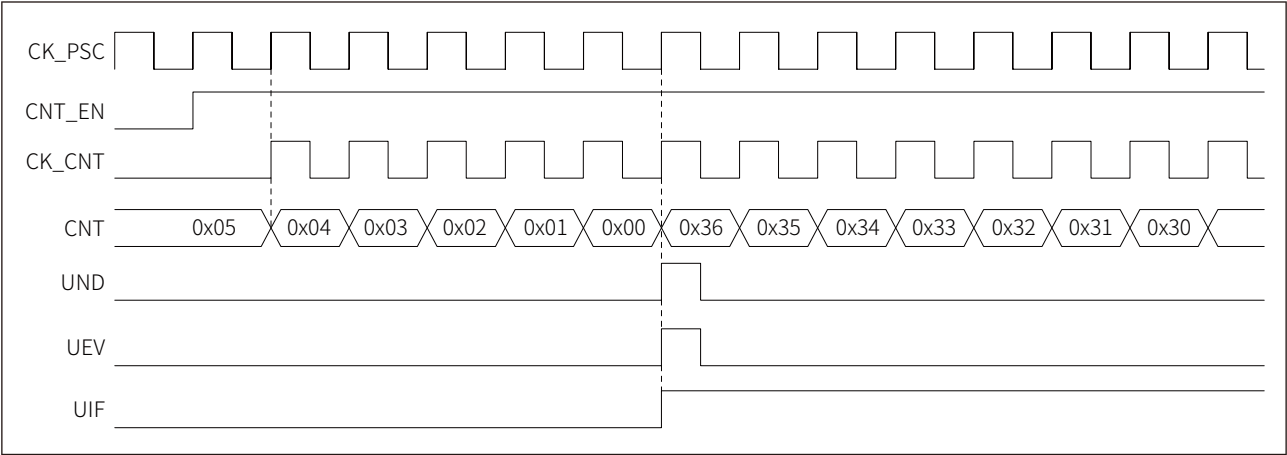


Figure 17-10 Count down, internal clock frequency division factor is 2

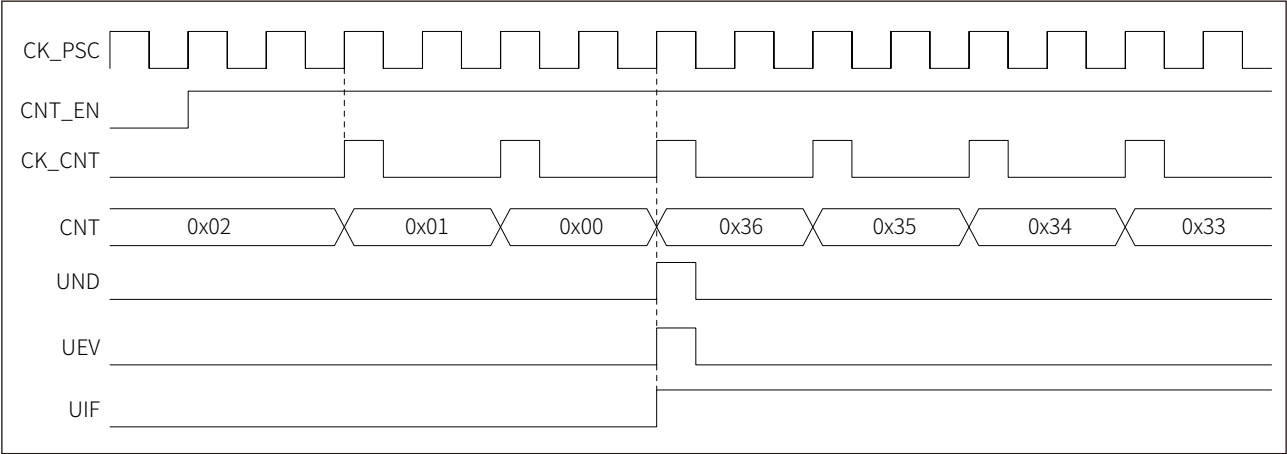


Figure 17-11 Count down, internal clock frequency division factor is 4

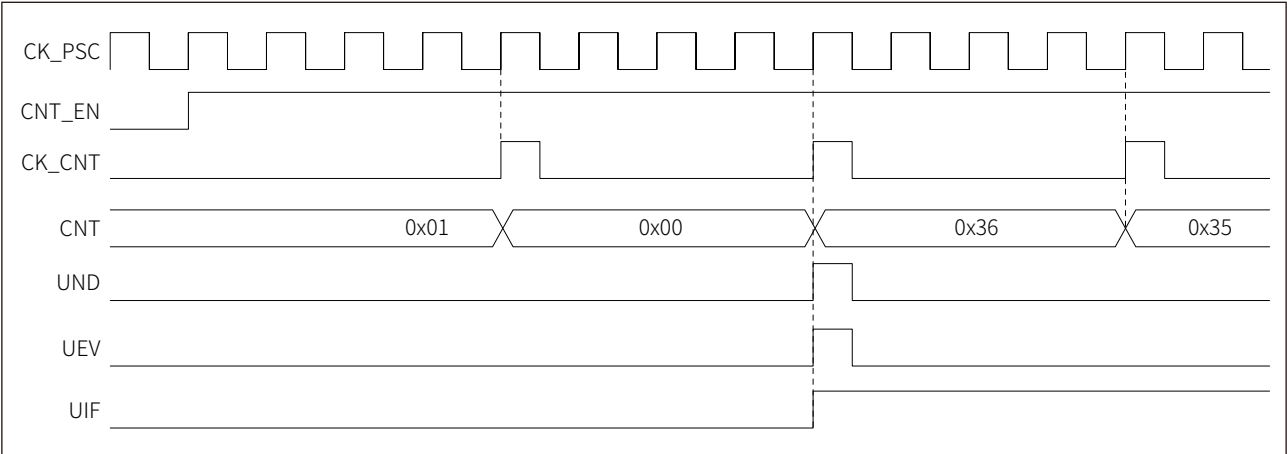
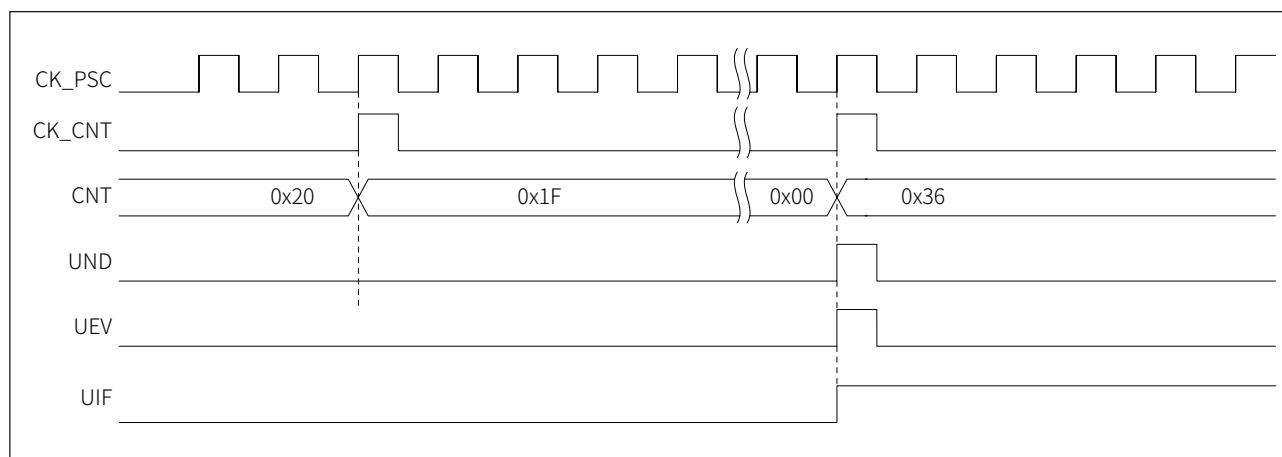


Figure 17-12 Count down, internal clock frequency division factor is N



### Center-aligned mode (count up/down)

In center-aligned mode, the counter counting up from 0 to ARR-1, generates a counter overflow event, then starts counting down from the ARR value to 1 and generates a counter underflow event, after which the counting starts counting up again from 0.

The center-aligned modes include center-aligned mode 1, center-aligned mode 2, and center-aligned mode 3. The three modes are identical in counting, except for the timing of the output compare interrupt flag, for details, please refer to the description of the CMS bit field of [17.10.1 ATIM\\_CR1 Control register 1](#).

In center-aligned mode, the DIR bit of control register ATIM\_CR1 cannot be written by software but can be read, and DIR is updated by hardware and indicates the current count direction. When center-aligned mode is activated, the counter will count up or down based on the current DIR bit field value. It is not possible to modify the DIR and CMS bit fields at the same time through software.

It is not recommended to perform a write operation to the counter while running center-aligned mode, as unexpected results will occur. In particular:

- If a value greater than the auto-reload value ARR ( $CNT > ARR$ ) is written to the counter CNT, the direction will not be updated. That is, if the counter was previously counting up, it continues to count up.
- If a value of 0 or ARR is written to the counter CNT, the count direction is updated, but no update event UEV is generated.

The most secure way to use the center-aligned mode is to generate the update event UEV by software before starting the counter (set ATIM\_EGR.UG to 1) and do not perform a write operation to the counter while it is running.

The following are examples of counter operation at different clock frequencies, without the use of a repeat counter:



Figure 17-13 Center-aligned mode, internal clock frequency division factor is 1, ARR = 0x06

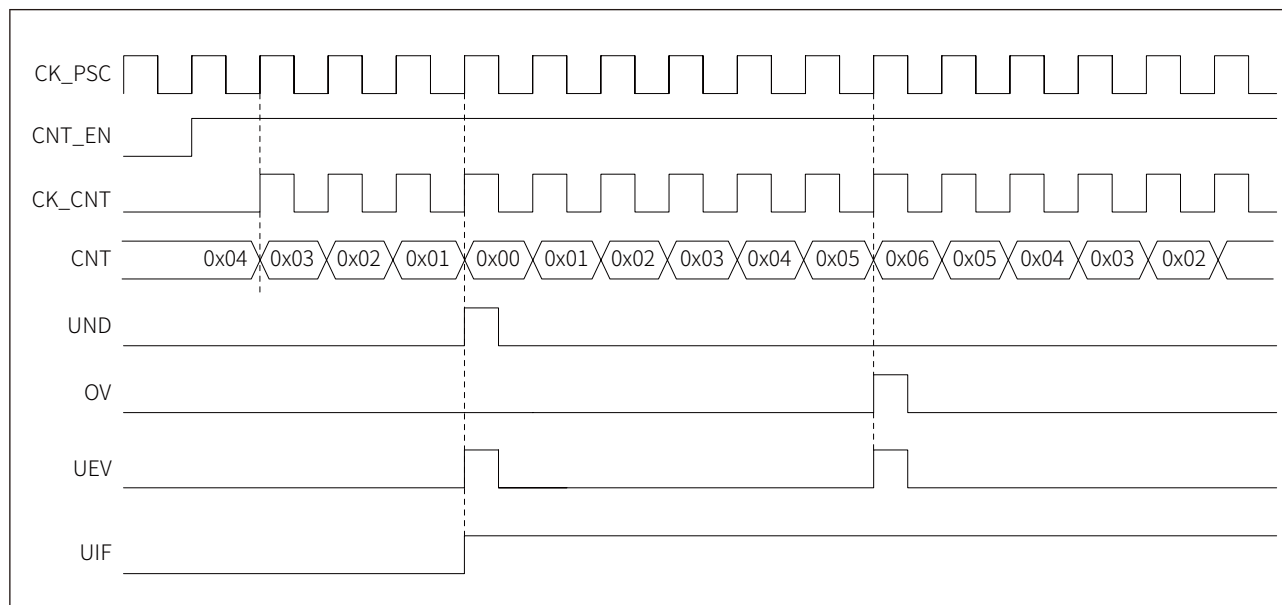


Figure 17-14 Center-aligned mode, internal clock frequency division factor is 2

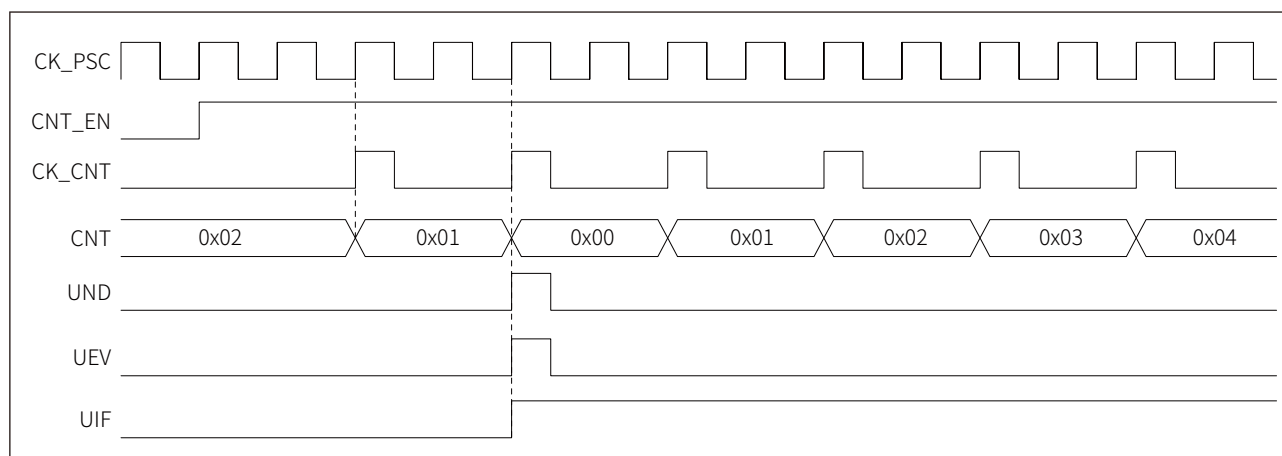


Figure 17-15 Center-aligned mode, internal clock frequency division factor is 4, ARR = 0x36

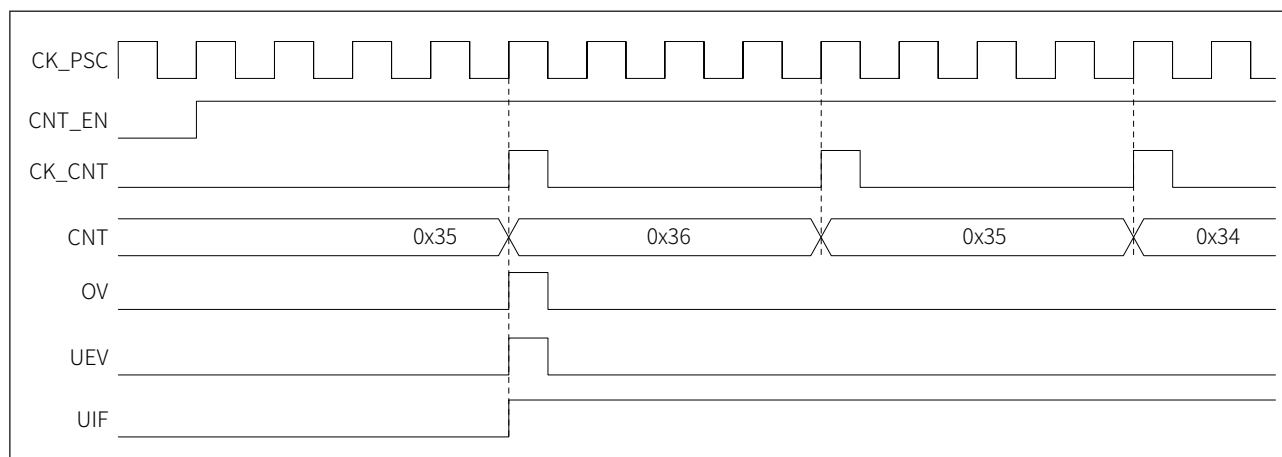


Figure 17-16 Center-aligned mode, internal clock frequency division factor is N

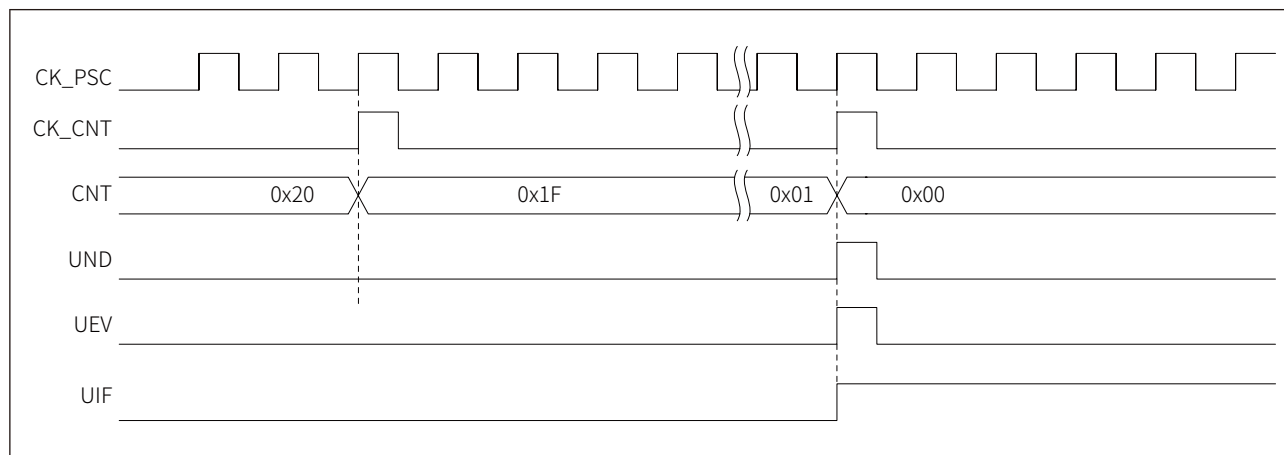


Figure 17-17 Center-aligned mode, update event when reloading cache is disabled (ATIM\_CR1.ARPE = 0)

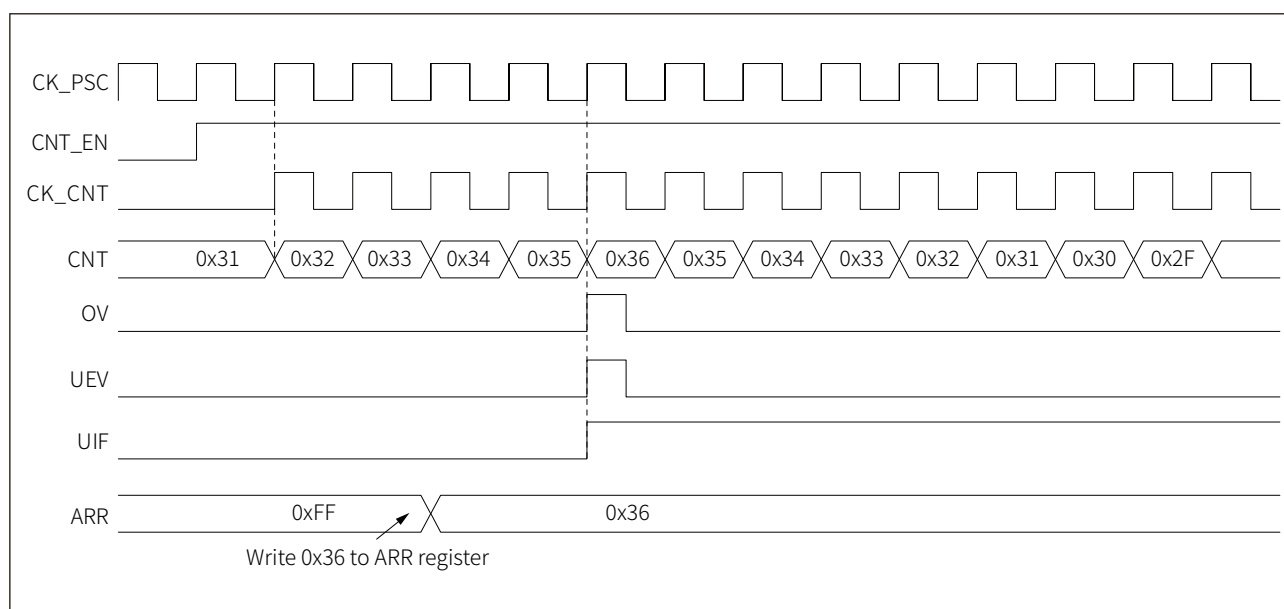
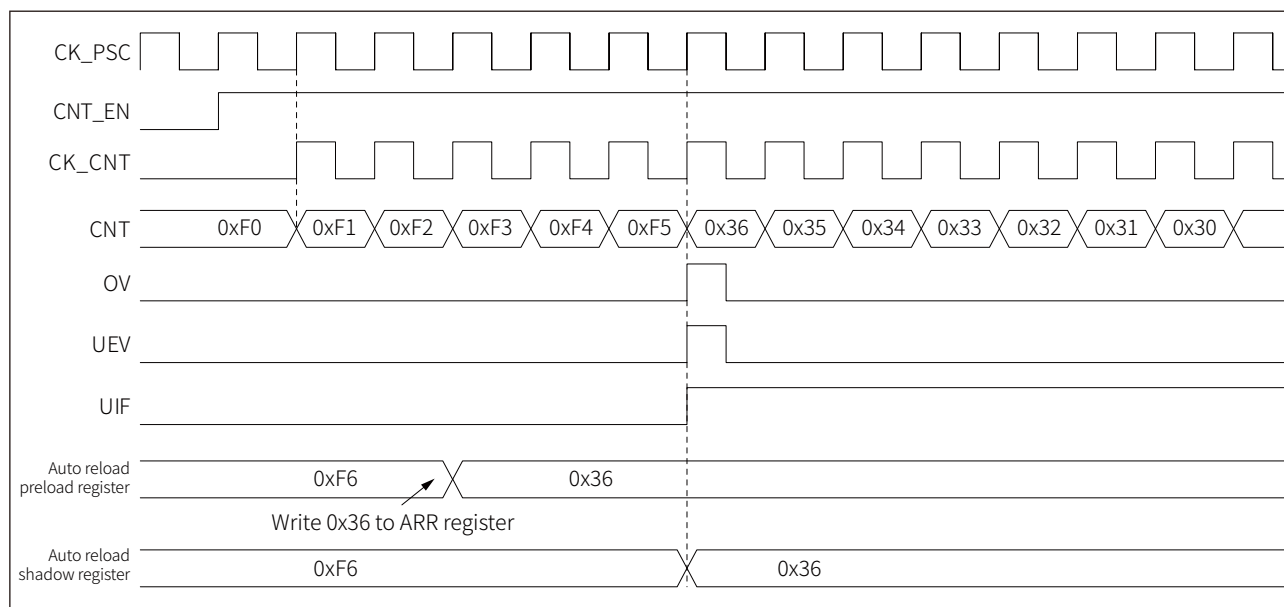


Figure 17-18 Center-aligned mode, update event when reloading cache is enabled (ATIM\_CR1.ARPE = 1)



#### 17.3.1.4 Reload register

The auto reload register ATIM\_ARR has a cache function that is enabled or disabled by controlling the ARPE bit field of register ATIM\_CR1.

When the counter is stopped or the cache function is disabled, updating the reload register ARR will immediately update its shadow register. When the timer is running and the cache function is active, modifying the reload register ARR will not update the shadow register immediately, but will only update the value of the reload register ARR into the shadow register when an update event UEV is generated.



### 17.3.1.5 Repeat counter

Starting ATIM automatically loads the REP value of the repeat counter register ATIM\_RCR into the repeat counter and automatically decrements it by 1 when the main counter overflows.

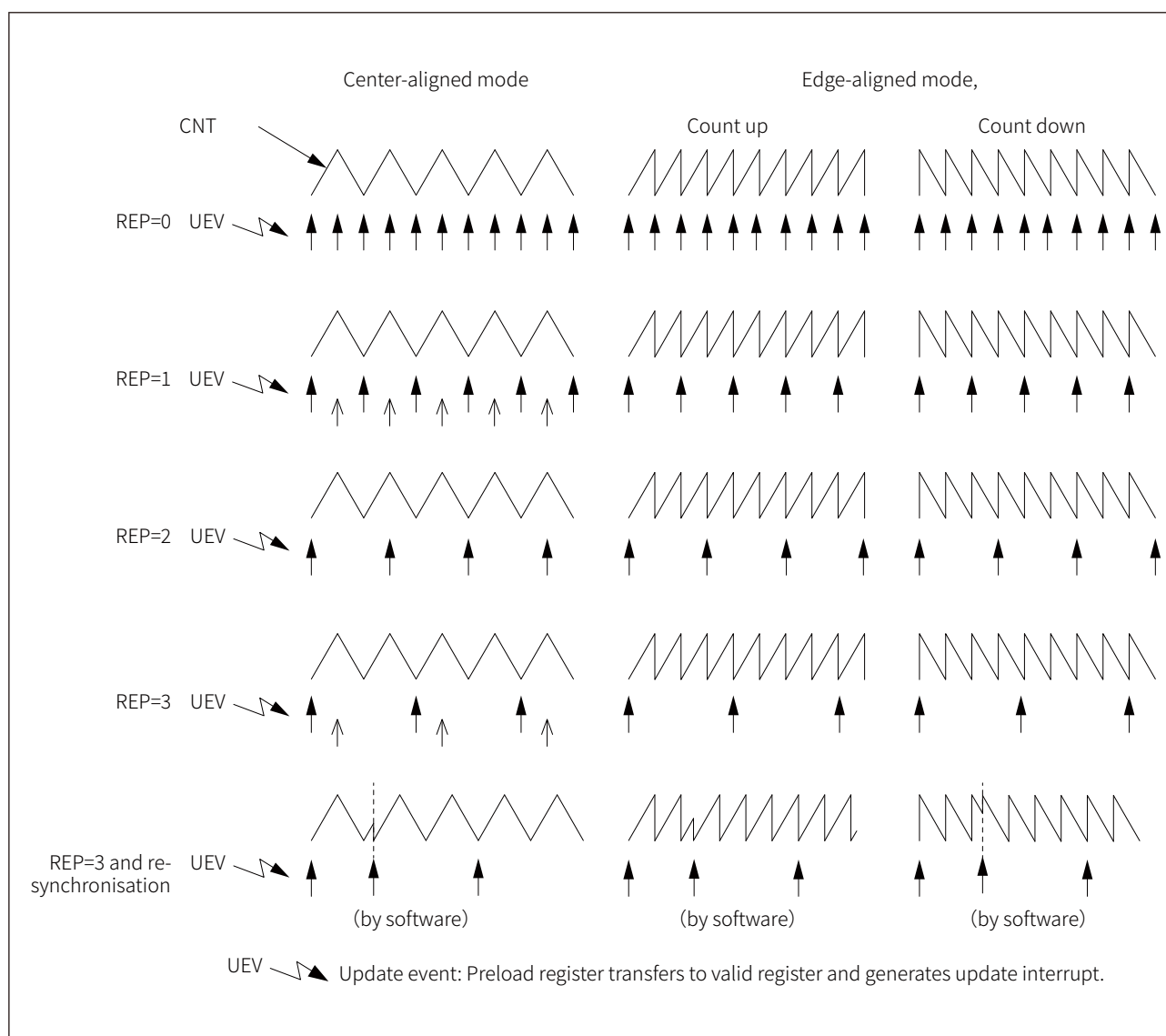
When using the repeat counter function (REP setting for ATIM\_RCR is not 0), overflow of the master counter triggers an update event UEV only if the repeat counter is decremented to 0.

When the UG is updated by software or reset by slave mode, the update event UEV is generated immediately, regardless of the current repeat counter value, and, the content of REP in the ATIM\_RCR register will be loaded into the repeat counter immediately.

In center-aligned mode, if the repeat value REP is odd, an update event occurs on overflow or underflow depending on when the ATIM\_RCR register is written and when the start counter is started: update event UEV occurs on overflow if the RCR register is written before the start counter; update event UEV occurs on underflow if the RCR register is written after the start counter.

The following figure shows an example of generating update event UEV in different counting modes with different repeat count values:

Figure 17-19 Repeat counting example



### 17.3.1.6 Update event UEV

Allows updates to be disabled or enabled by the UDIS bit field of the control register ATIM\_CR1.

#### Enable UEV

Set the UDIS bit field to 0 to enable UEV. At this time, update request sources can be selected according to the URS bit field, as shown in the following table:

Table 17-2 Update source settings

| ATIM_CR1.URS | Update source                                                                                          |
|--------------|--------------------------------------------------------------------------------------------------------|
| 0            | The counter overflow, underflow.<br>UG set.<br>Update events generated from the slave mode controller. |
| 1            | The counter overflow, underflow.                                                                       |

When an update event occurs, the following actions occur:

1. Reinitialise the counter:
  - Count down (edge-aligned mode) : Reloads the auto-reload value ATIM\_ARR.
  - Center-aligned mode or count up (edge-aligned mode): the counter is cleared.
2. If the repeat counter is enabled, the repeat counter is reloaded with the contents of the ATIM\_RCR register.
3. If the cache register function is used, the corresponding preloaded register is updated to its shadow register.
4. The prescaler counter is cleared and the new prescaler value in ATIM\_PSC takes effect.
5. The event update interrupt flag bit ATIM\_ISR.UIF is set by hardware.

When an update event UEV occurs, the event update interrupt flag bit ATIM\_ISR.UIF is set by hardware. If the interrupt is allowed (set ATIM\_IER.UIE to 1), an update interrupt request is generated, and setting ATIM\_ICR.UIF to 0 clears the flag bit.

*Attention: If a repeat counter is used (ATIM\_RCR.REP is not 0), an update event UEV is generated only if the count reaches the repeat count (ATIM\_RCR.REP reaches 0).*

#### Disable UEV

Set the UDIS bit field to 1 to disable UEV and no more update events are generated.

If the cache register function is used, the value of each corresponding shadow register remains unchanged. However, if the UG bit is set to 1 or a hardware reset is received from the slave mode controller, the counters of the counter and prescaler are reinitialised.



### 17.3.1.7 Single pulse mode

The counter can work in single count or continuous count mode and is selected by the OPM bit field of the control register ATIM\_CR1.

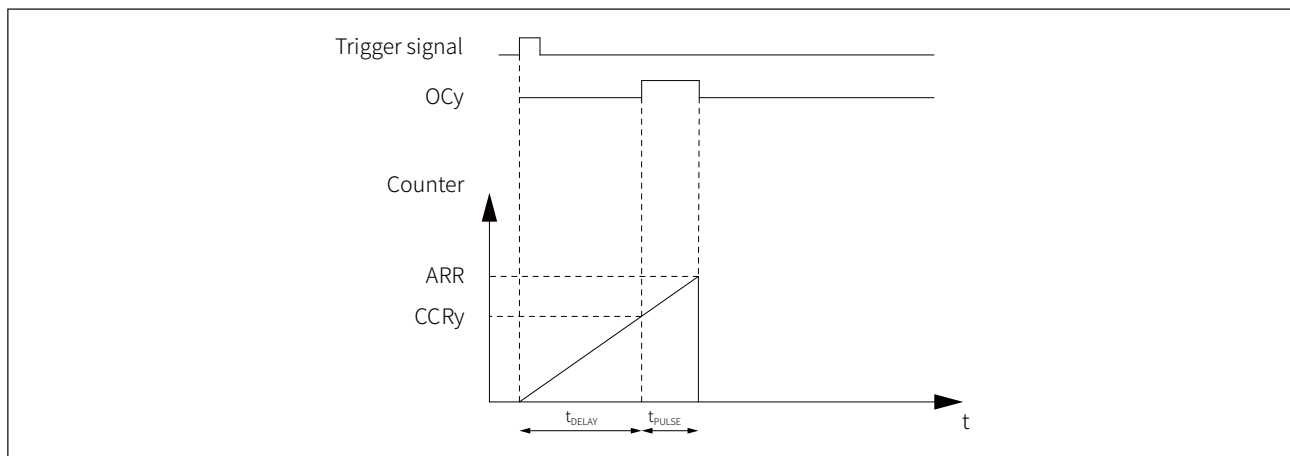
#### Single count mode

Set ATIM\_CR1.OPM to 1 to enable the timer to work in single count mode.

After starting the counter, the counter CNT counts driven by the CK\_CNT clock, and the counter stops counting when the next update event occurs while ATIM\_CR1.CEN is automatically cleared by hardware.

The counter can be activated by the slave mode controller, together with the output comparison mode of the timer, so that the counter is activated by a trigger signal and generates a single pulse with a programmable pulse width after a programmable delay. As shown in the figure below, the timer starts counting when it detects a valid edge of the trigger signal, and after a delay of  $t_{\text{DELAY}}$ , it generates a positive pulse of width  $t_{\text{PULSE}}$  on the compare output port. Where  $t_{\text{DELAY}}$  is determined by the difference between the initial value of the count register ATIM\_CNT and the capture/compare register ATIM\_CCRy, and  $t_{\text{PULSE}}$  is determined by the difference between the auto-reload register ATIM\_ARR and the capture/compare register ATIM\_CCRy.

Figure 17-20 Example of single pulse output (count up mode)



Sampling the trigger input to activate the OC output requires multiple clock cycles, and the OCyFE bit field of the comparison mode register ATIM\_CCMRxCMP can be set to 1 to shorten the OC output delay time.

#### Continuous count mode

Set ATIM\_CR1.OPM to 0 so that the timer works in continuous count mode and the counter does not stop counting when an update event occurs.

### 17.3.1.8 External trigger input channel

The external trigger input signal ETR can be used as a trigger input (TRGI) for the slave mode controller or as a count clock for the counter, see [17.3.2 Operating modes](#). The ETR signal can be input controlled, including polarity selection and edge detection, prescaling and filtering.

#### Polarity selection and edge detection

The ETP bit field of the slave mode control register ATIM\_SMCR is used to select the trigger polarity of the ETR input signal. When ETP is set to 0, ETR is not inverted and is valid at high level or rising edge; when ETP is set to 1, ETR is inverted and is valid at low level or falling edge.

#### Prescaler

The frequency of external trigger signal ETRP shall not exceed 1/4 of PCLK frequency. when the frequency of ETRP is high, user should divide the external signal by appropriate ETPS prescaler setting to reduce the ETRP frequency, and the dividing coefficient can be set to 1, 2, 4, 8.

#### Filter

The filter uses digital filtering to sample the input signal at a certain frequency, the signal is valid when it is continuously sampled to N same levels, otherwise the signal is invalid, so as to filter out the high-frequency clutter signal.

The filter of the ETR input signal is configured by the ETF bit field of the slave mode control register ATIM\_SMCR, which can set the sampling frequency and the number of sampling points. The sampling frequency  $f_{\text{SAMPLING}}$  is provided by the clock after  $f_{\text{DTS}}$  dividing.  $f_{\text{DTS}}$  is the frequency obtained after  $f_{\text{PCLK}}$  dividing. The dividing factor is configured by the CKD bit field of the ATIM\_CR1 register, which can be set to 1, 2, or 4 divisions.



### 17.3.1.9 Input capture channel

The ATIM supports 6 input capture channels TI1/2/3/4/5/6 which can be used as capture commands. Among them, TI1 and TI2 can also be used as trigger input (TRGI) and encoder interface input of slave mode controller, see [17.3.2 Operating modes](#).

Supports input control of Tly signals, including filtering, edge detection and prescaling.

#### Filter

In the filtering stage, the corresponding Tly input signal is sampled at a certain frequency, and the filtered signal TlyF is generated.

The filter for the Tly input signal is configured by the ICyF bit field of the capture mode register ATIM\_CCMRxCAP, which can be set for the sampling frequency and the number of sampling points. The sampling frequency  $f_{\text{SAMPLING}}$  is provided by the clock after  $f_{\text{PCLK}}$  or  $f_{\text{DTS}}$  dividing,  $f_{\text{DTS}}$  is the frequency obtained after  $f_{\text{PCLK}}$  dividing, and the dividing factor is configured by the CKD bit field of ATIM\_CR1 register, and it can be set to 1, 2, or 4 divisions.

#### Edge detection

The edge detector with polarity selection generates a TlxFPx signal, for specific valid polarity please see the description of CCyNP and CCyP bit fields in [17.10.14 ATIM\\_CCER Capture/compare enable register](#).

#### Prescaler

When the Tly signal is used for input capture, the capture channel signal ICy can be prescaled, which is controlled through the ICyPSC bit field of the capture mode register ATIM\_CCMRxCAP, and supports 1, 2, 4, and 8 frequency division.



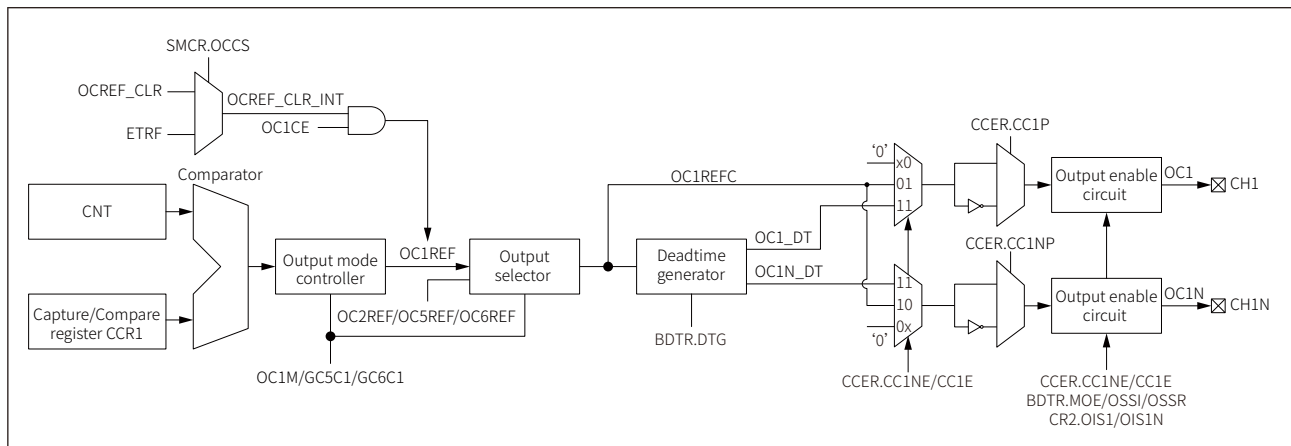


### 17.3.1.10 Output compare channel

ATIM supports 6 independent output compare channels CH1/2/3/4/5/6 (all channels support complementary output CHyN), each consisting of a capture/compare register (including a shadow register) and an output stage (comparator and output control).

The output control unit is used to control the waveform of the output port when comparing and matching, and can configure the output compare mode, output polarity selection and output enable. Taking compare channel 1 as an example, its block diagram is shown below:

Figure 17-21 Output compare channel 1



### 17.3.2 Operating modes

The ATIM supports multiple operating modes, which are configured by the SMS bit field of the slave mode control register ATIM\_SMCR, as shown in the following table:

Table 17-3 ATIM operating modes

| ATIM_SMCR.SMS | Slave mode selection                            |
|---------------|-------------------------------------------------|
| 0000          | Disable slave mode, use internal clock          |
| 0111          | External clock mode 1                           |
| 0100          | Reset mode                                      |
| 0101          | Gated mode                                      |
| 0110          | Trigger mode                                    |
| 1000          | Combined reset + Trigger mode                   |
| 1001          | Combined gated + Reset mode                     |
| 1110          | Quadrature encoder mode, x1 mode                |
| 1111          | Quadrature encoder mode, x1 mode                |
| 0001          | Quadrature encoder mode, x2 mode                |
| 0010          | Quadrature encoder mode, x2 mode                |
| 0011          | Quadrature encoder mode, x4 mode                |
| 1010          | Encoder mode (clock + direction), x2 mode       |
| 1011          | Encoder mode (clock + direction), x1 mode       |
| 1100          | Encoding mode ( with direction clock ), x2 mode |
| 1101          | Encoding mode ( with direction clock ), x1 mode |

*Note: The slave mode select bits include ATIM\_SMCR[16] and ATIM\_SMCR[2:0] to be configured in combination.*

*Note: Slave mode select bits SMS[3:0] support the preload function, please refer to the description of SMSPE and SMSPS bit field.*

#### 17.3.2.1 Internal clock mode

When the SMS bit field of the slave mode control register ATIM\_SMCR is 0x0, the timer slave mode is disabled and the prescaler clock is provided directly by the internal clock PCLK. Setting ATIM\_CR1.CEN to 1 will enable the counter to start counting.



### 17.3.2.2 External clock mode

#### External clock mode 1

When the SMS bit field of the slave mode control register ATIM\_SMCR is 0x7, the counter clock is supplied by the rising edge of the selected trigger signal (TRGI). There are several trigger options for the TRGI signal, which are selected through the TS bit field of the ATIM\_SMCR register, as shown in the table below:

Table 17-4 TRGI signal source

| ATIM_SMCR.TS bit field value | TRGI signal source              |
|------------------------------|---------------------------------|
| 00111                        | External trigger input (ETRF)   |
| 000xx/01xxx/10000            | Internal trigger (ITR0~12)      |
| 00100                        | TI1 edge detector (TI1F_ED)     |
| 00101                        | Filtered timer input 1 (TI1FP1) |
| 00110                        | Filtered timer input 2 (TI2FP2) |

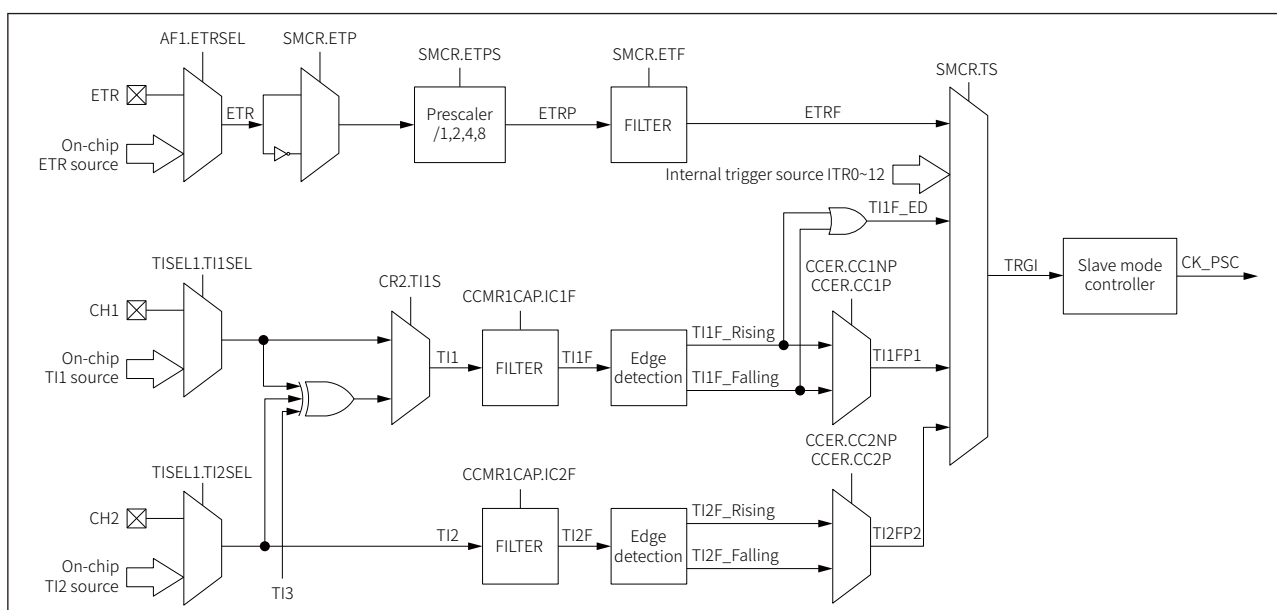
The ETR input signal source can be external ATIM\_ETR/GTIMx\_ETR pins or other peripherals on-chip, please refer to [17.3.7 On-chip peripheral interconnect ETR](#). When ATIM\_ETR is selected as the TRGI signal source, the external trigger polarity can be selected via ATIM\_SMCR.ETP, the pre-scaler frequency can be set via ATIM\_SMCR.ETPS, and the filtering control can be performed via ATIM\_SMCR.ETF.

The internal trigger ITR sources are the trigger output signal TRGO of BTIM and GTIM, and the TXD/RXD signals of UART, please refer to [17.3.6 Timer cascade ITR](#).

Both TI1 and TI2 have filtering and edge detection functions. TI1 and TI2 are controlled by filtering through the IC1F and IC2F bit fields of the capture mode register ATIM\_CCMR1CAP, respectively, and edge detection through the CC1P/CC1NP and CC2P/CC2NP bit fields of the capture/compare enable register ATIM\_CCER, respectively.

The external clock mode 1 connection diagram is shown below:

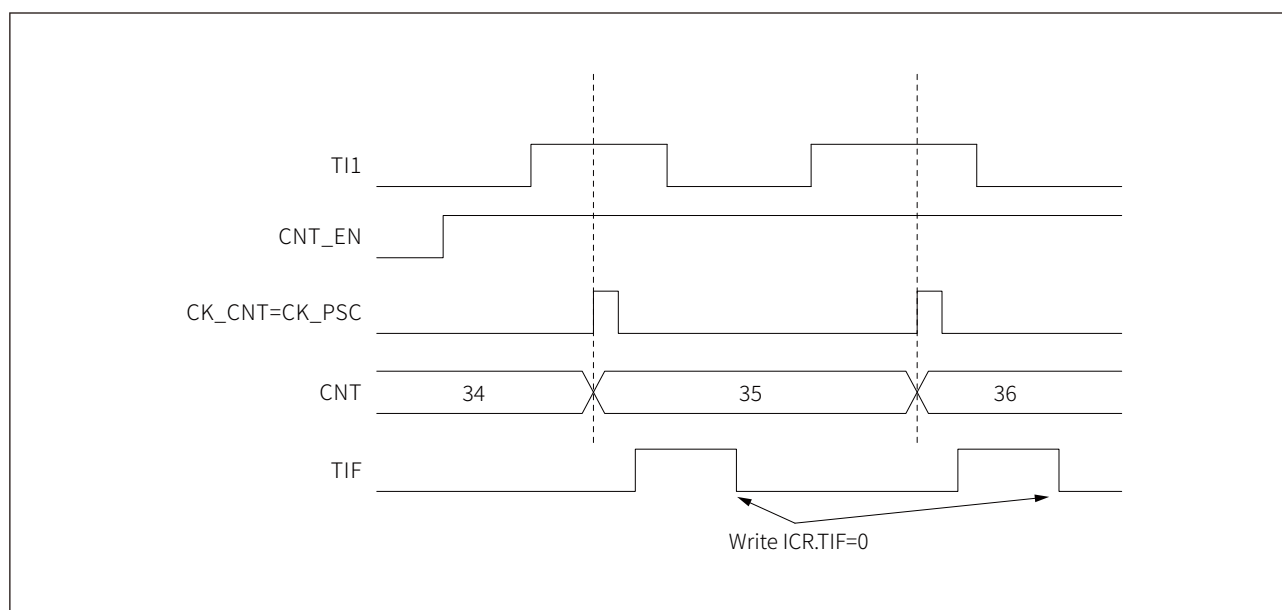
Figure 17-22 External clock mode 1 connection diagram



When a valid edge appears in TRGI, the ATIM\_ISR.TIF flag will be set to 1, and the flag can be cleared by writing 0 to ATIM\_ICR.TIF.

The following figure shows the external clock mode 1 timing diagram with TI1 rising edge configured. The delay between the rising edge of TI1 and the actual counter clock is due to the resynchronisation circuitry of the TI1 input.

Figure 17-23 Example of external clock mode 1 timing



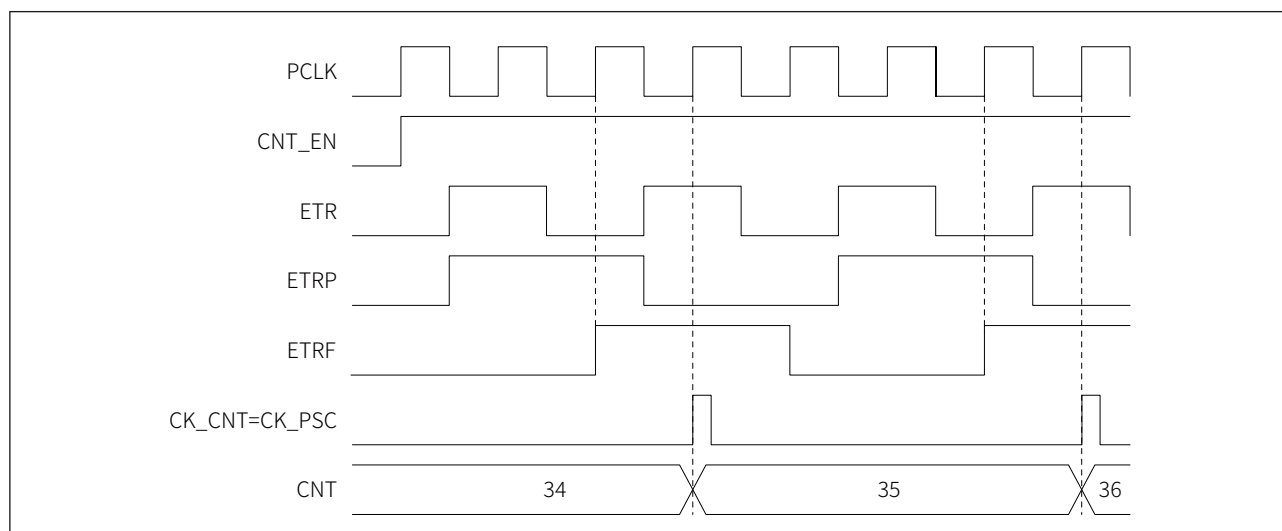
### External clock mode 2

Set the ECE bit field of the slave mode control register ATIM\_SMCR to 1 to enable external clock mode 2. In this mode, the counter clock is supplied by any valid edge of the ETRF signal, which has the same effect as selecting external clock mode 1 and connecting the TRGI to the ETRF (SMS = 0111 and TS = 00111), refer to [Figure 17-22: External clock mode 1 connection diagram](#).

External clock mode 2 can be used in conjunction with the following slave modes: reset mode, gated mode, and trigger mode, where the TRGI in slave mode must not be connected to the ETRF (i.e., the TS bit field cannot be set to 0x7). If both external clock mode 1 and external clock mode 2 are enabled, the external clock input is ETRF. External clock mode 2 cannot be used in conjunction with encoder mode.

The following figure shows a timing example of external clock mode 2 when ETR is set to 2 divisions, the delay between the rising edge of ETR and the actual counter clock is caused by the resynchronisation circuitry of the ETRP signal, so the maximum frequency that can be correctly captured by the counter is up to 1/4 of the internal clock PCLK frequency, and the user should divide the external signals appropriately when the ETRP signal becomes faster.

Figure 17-24 Example of external clock mode 2 timing



### 17.3.2.3 Reset mode

When the SMS bit field of the slave mode control register ATIM\_SMCR is 0x4, the ATIM is configured in reset mode and the reset of the counter is controlled by the TRGI signal.

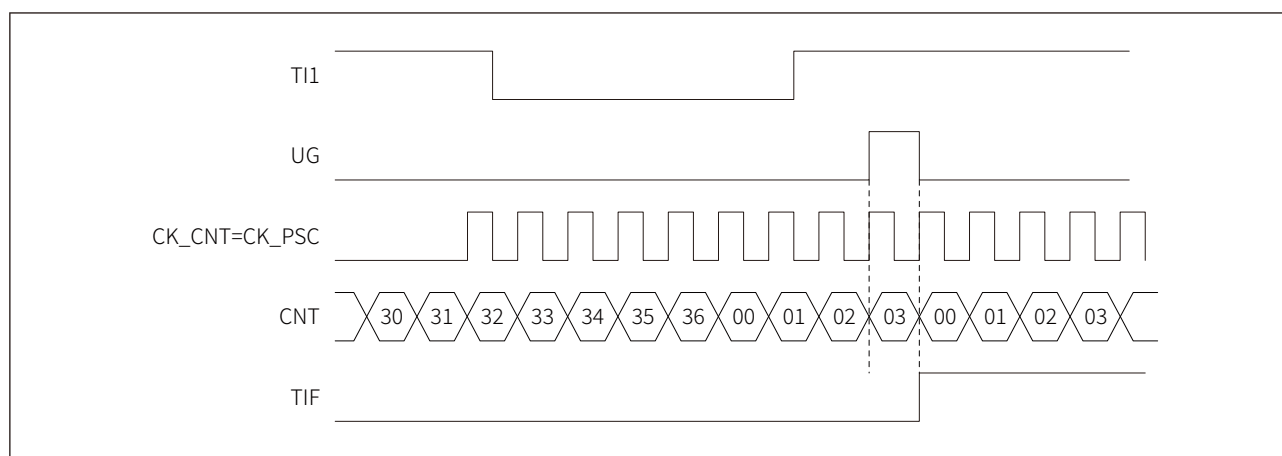
The source of the TRGI signal is controlled by the TS bit field of the slave mode control register ATIM\_SMCR. See [Table 17-4: TRGI signal source](#).

When a valid trigger input (TRGI) is detected, the following effects occur:

- Reinitialise the counters and prescaler counters.
- If the URS bit field of control register ATIM\_CR1 is 0, an update event UEV is generated, and the event update interrupt flag ATIM\_ISR.UIF is set to 1. An interrupt/DMA request can be generated.
- The trigger interrupt flag ATIM\_ISR.TIF is set to 1, which can generate an interrupt/DMA request.

The following figure shows an example of the reset mode timing diagram. Set ATIM\_CR1.CEN to 1 to enable the counter, the counter counts normally according to the counting clock CK\_CNT, and when a rising edge of TI1 occurs, the counter is cleared to 0 and starts counting from 0 again, while the ATIM\_ISR.TIF flag is set to 1. The delay between the rising edge of TI1 and the actual counter reset is caused by the resynchronisation circuitry of the TI1 input.

Figure 17-25 Reset mode timing



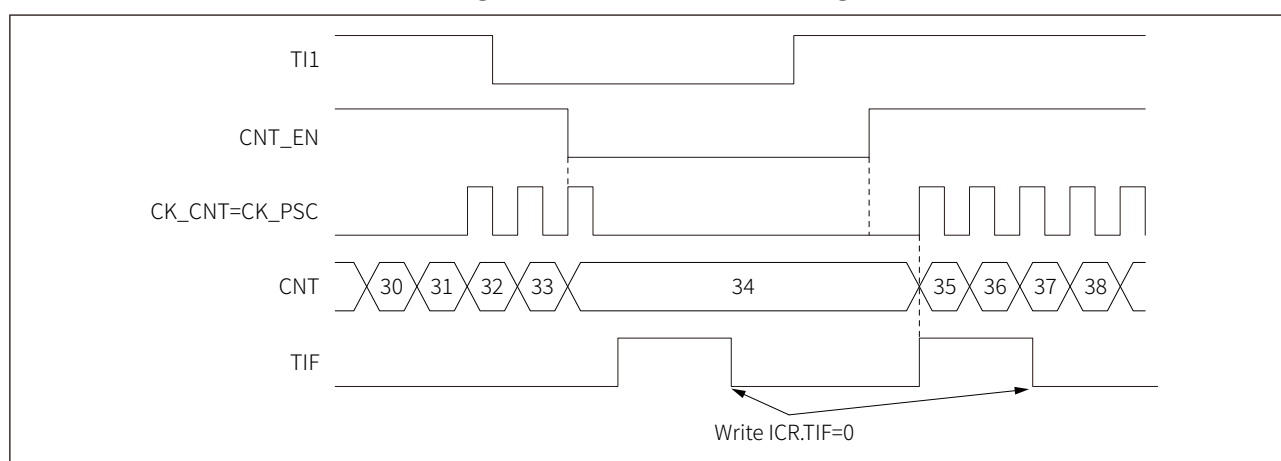
### 17.3.2.4 Gated mode

When the SMS bit field of the slave mode control register ATIM\_SMCR is 0x5, the ATIM is configured for gated mode. In this mode, the counter is started counting when the trigger input (TRGI) is high and ATIM\_CR1.CEN is 1. The counter stops counting immediately (but is not reset) when the trigger input (TRGI) is low or ATIM\_CR1.CEN is 0. Both the start and stop of the counter are controlled.

The source of TRGI signals is controlled by the TS bit field of the slave mode control register ATIM\_SMCR. For details, see [Table 17-4: TRGI signal source](#), however it should be noted that TI1F\_ED cannot be selected as a trigger input in gated mode.

The following figure shows an example of gated mode timing diagram. When TI1 is high, the counter starts counting; when TI1 is low, the counter pauses counting. The ATIM\_ISR.TIF flag bit is set to 1 when the counter starts and stops. The delay between the edge of TI1 and the actual counter enable signal CNT\_EN is caused by the resynchronisation circuitry of the TI1 input.

Figure 17-26 Gated mode timing



### 17.3.2.5 Trigger mode

When the SMS bit field of the slave mode control register ATIM\_SMCR is 0x6, the ATIM is configured for trigger mode. In this mode, when ATIM\_CR1.CEN is set to 1 or when the rising edge of the trigger signal TRGI appears, the start counter is triggered (but not reset).

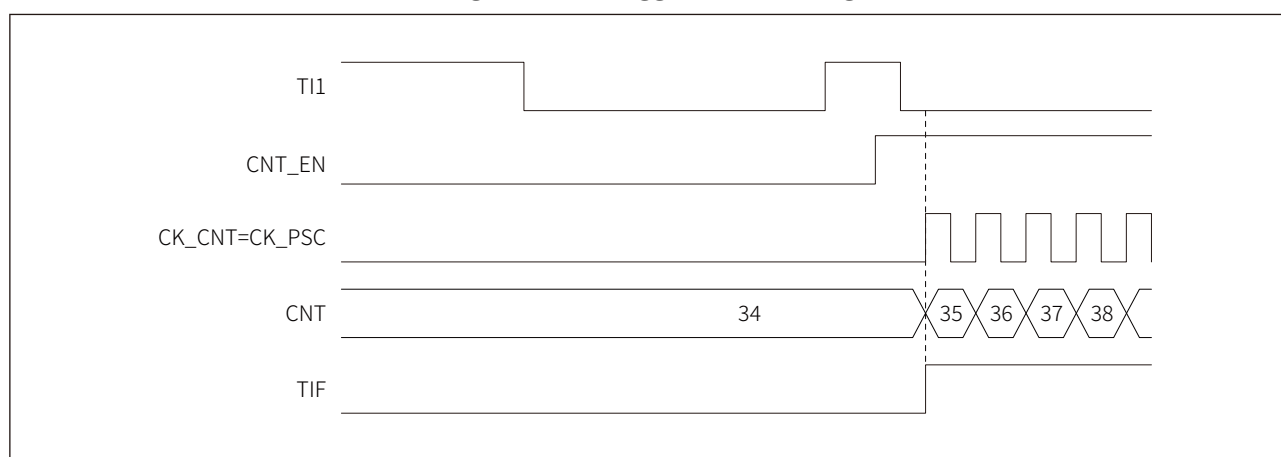
The source of the TRGI signal is controlled by the TS bit field of the slave mode control register ATIM\_SMCR, see [Table 17-4: TRGI signal source](#).

When a valid trigger signal is detected, the following effects occur:

- ATIM\_CR1.CEN is set by hardware.
- The trigger interrupt flag bit ATIM\_ISR.TIF is set to 1, which can generate an interrupt/DMA request.
- The counter is activated and starts counting.

The following figure shows an example of a trigger mode timing diagram. When a rising edge of TI1 occurs, the counter starts counting while the ATIM\_ISR.TIF flag bit is set to 1. The delay between the rising edge of TI1 and the actual counter startup is caused by the resynchronisation circuitry of the TI1 input.

Figure 17-27 Trigger mode timing



### 17.3.2.6 Combined reset + trigger mode

When the SMS bit field of the slave mode control register ATIM\_SMCR is 0x8, the ATIM is configured for combined reset + trigger mode. In this mode, when a rising edge of the selected trigger input (TRGI) occurs, the counter will be reinitialised and the counter will be started, and the trigger interrupt flag ATIM\_ISR.TIF will be set to 1. If the URS bit field of the control register ATIM\_CR1 is 0, an update event UEV will be generated, and the event update interrupt flag ATIM\_ISR.UIF will be set by hardware.

The source of the TRGI signal is controlled by the TS bit field of the slave mode control register ATIM\_SMCR, see [Table 17-4: TRGI signal source](#).

### 17.3.2.7 Combined gated + reset mode

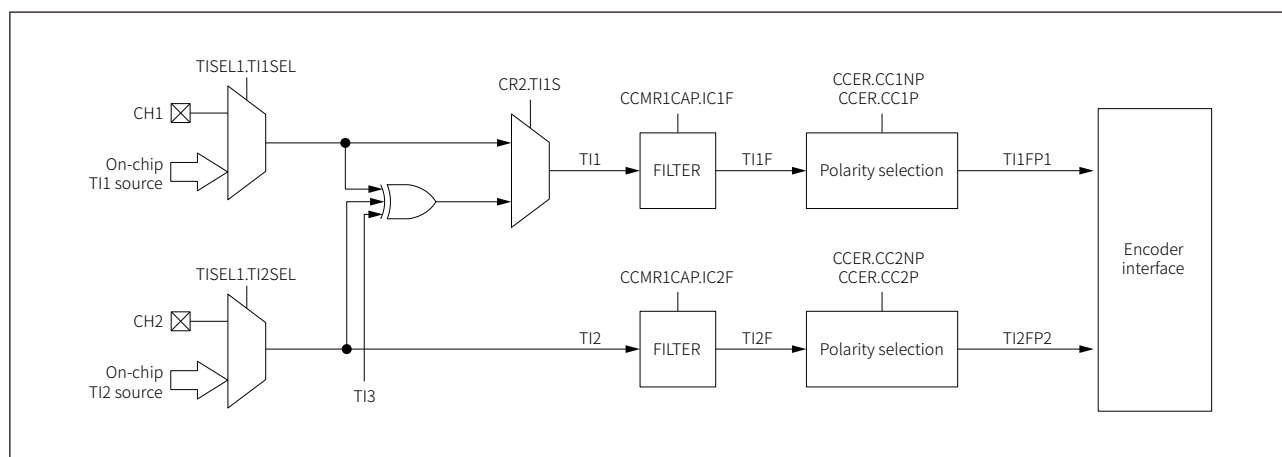
When the SMS bit field of the slave mode control register ATIM\_SMCR is 0x9, the ATIM is configured for the combined gating + reset mode. In this mode, the start and stop of the counter are controlled. When the trigger input (TRGI) is high and ATIM\_CR1.CEN is 1, the counter is started to count. When the trigger input (TRGI) is low, the counter immediately stops counting and is reset. When the counter is started and stopped, the ATIM\_ISR.TIF flag bit is set to 1. When the counter is reset, ATIM\_ISR.UIF flag bit is set to 1.

The source of the TRGI signal is controlled by the TS bit field of the slave mode control register ATIM\_SMCR, see [Table 17-4: TRGI signal source](#).

### 17.3.2.8 Quadrature encoder mode

ATIM supports quadrature encoder mode for receiving and decoding quadrature encoder signals. In this mode, it allows a direct connection to an external quadrature encoder via the CH1 and CH2 pins, which enables the counter to count up or down automatically according to the hopping sequence of the input signals, and the counter value always indicates the position of the encoder. The functional block diagram is shown below:

Figure 17-28 Encoder mode block diagram



The CH1 and CH2 input signals have filtering and polarity selection functions, which are filtering controlled through the IC1F and IC2F bit fields of the ATIM\_CCMR1CAP register, respectively, and the input polarity is selected through the CC1P and CC2P bit fields of the ATIM\_CCER register, and the CC1NP and CC2NP bit fields must be kept clear.

Set ATIM\_CR1.CEN to 1 to enable the counter, the counter will be driven by each valid hop of the signals TI1FP1 and TI2FP2 after the input signals of the channel CH1 and CH2 pins have been filtered and polarity selected, and according to the hopping sequence of the two input signals, the counting pulses and the direction signals are generated, refer to [Table 17-5: Relationship between counting direction and encoder signal](#). The encoder's current count direction flag ATIM\_CR1.DIR is automatically set and cleared by hardware, and the DIR bit is counted whenever a signal transition occurs on any input (CH1 or CH2), regardless of whether the counter is counting only at the TI1FP1 or TI2FP2 edges, or at both TI1FP1 and TI2FP2.

ATIM supports several quadrature coded counting modes, which are set by the SMS bit field of the slave mode control register ATIM\_SMCR. The relationship between the counting direction and the encoder signals in different modes is shown in the table below:



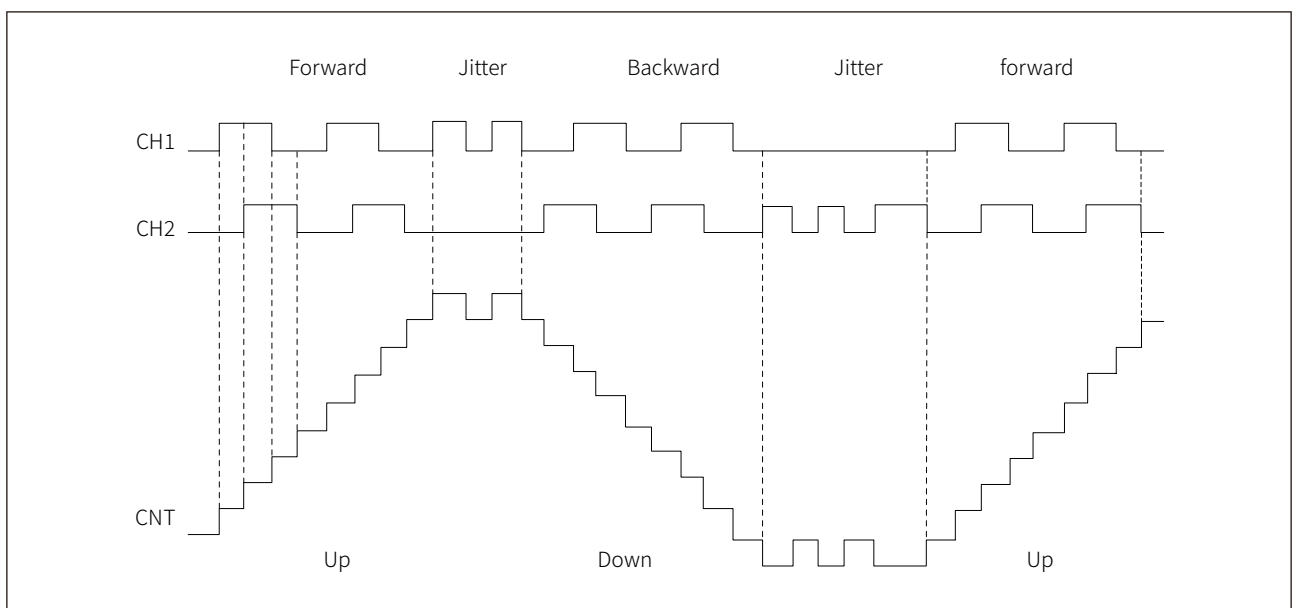
Table 17-5 Relationship between counting direction and encoder signal

| Mode                                             | SMS  | Signal level |        | TI1FP1     |            | TI2FP2     |            |
|--------------------------------------------------|------|--------------|--------|------------|------------|------------|------------|
|                                                  |      | TI2FP2       | TI1FP1 | Rising     | Falling    | Rising     | Falling    |
| Counting on the TI1FP1 edge (x1 mode)            | 1110 | High         | -      | Down count | Up count   | No count   | No count   |
|                                                  |      | Low          | -      | No count   | No count   | No count   | No count   |
| Counting on the TI2FP2 edge (x1 mode)            | 1111 | -            | High   | No count   | No count   | Up count   | Down count |
|                                                  |      | -            | Low    | No count   | No count   | No count   | No count   |
| Counting on the TI1FP1 edge (x2 mode)            | 0001 | High         | -      | Down count | Up count   | No count   | No count   |
|                                                  |      | Low          | -      | Up count   | Down count | No count   | No count   |
| Counting on the TI2FP2 edge (x2 mode)            | 0010 | -            | High   | No count   | No count   | Up count   | Down count |
|                                                  |      | -            | Low    | No count   | No count   | Down count | Up count   |
| Counting on the TI1FP1 and TI2FP2 edge (x4 mode) | 0011 | High         | High   | Down count | Up count   | Up count   | Down count |
|                                                  |      | Low          | Low    | Up count   | Down count | Down count | Up count   |

The third signal from the encoder output represents the mechanical zero point and can be connected to an external trigger input pin to trigger a counter reset.

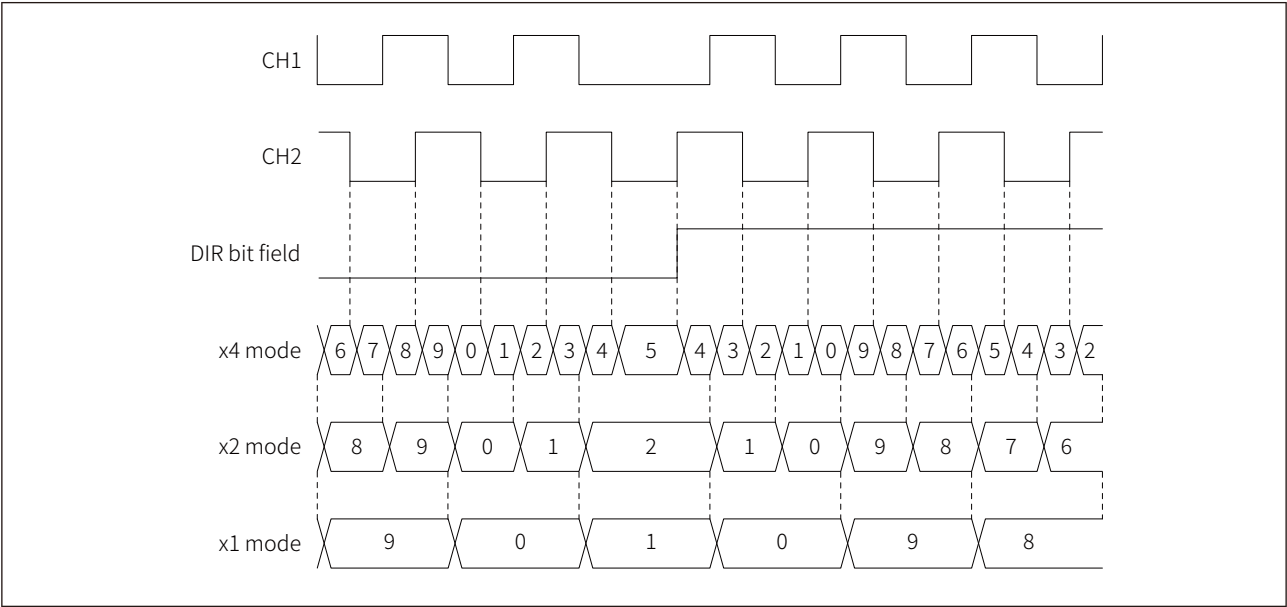
The following figure is an example of quadrature encoded counting mode-x4 mode operation, showing the generation of the counting signal and direction control. It also shows how input jitter is suppressed when a double edge is selected; jitter may be generated when the sensor is positioned close to a transition point.

Figure 17-29 Example of quadrature encoder mode-x4 mode operation



The following figure shows the timer count values during encoder inversion for various quadrature coded counting modes, with neither CH1 nor CH2 inputs inverted in the example.

Figure 17-30 Example of counting for each quadrature encoder mode



The encoder mode provides information about the current position of the sensor. The counter automatically counts up or down based on the pulses output from the encoder, and the current counter value indicates the current position of the encoder. Using another timer configured in capture mode, the rising or falling edge of the encoder signal can be captured and its timestamp recorded. By measuring the time interval between the two events, the corresponding dynamic information such as speed, acceleration and deceleration can be calculated; The encoder output indicating the mechanical zero position can be used for this purpose, and by measuring the time interval from the mechanical zero position to some other position, the change in position and velocity relative to the zero position can be calculated.

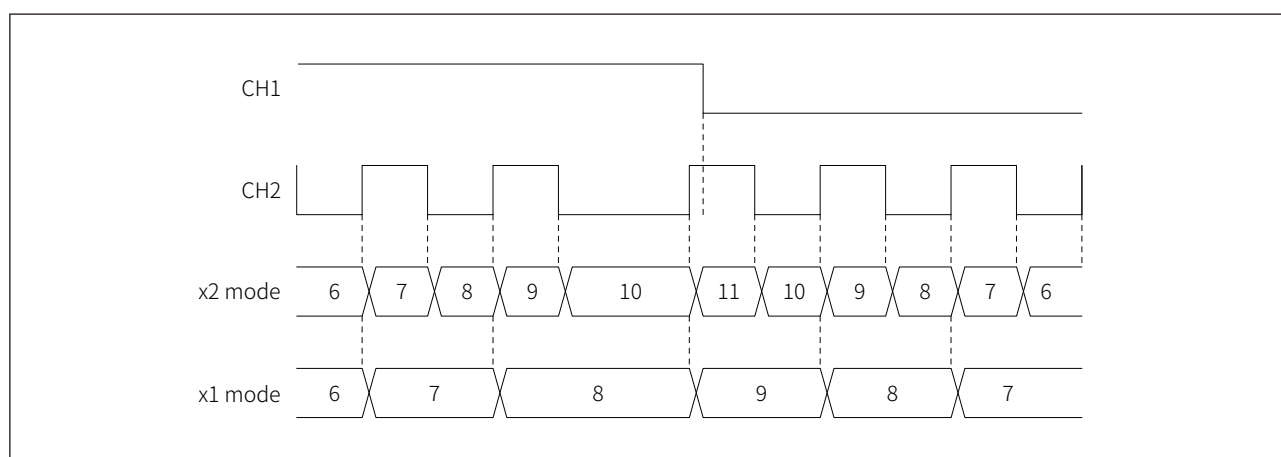
### 17.3.2.9 Clock + direction encoder mode

The timer also supports clock plus direction encoder mode, which requires configuring the SMS bit field of the slave mode control register ATIM\_SMCR to be 0xA or 0xB, which correspond to x2 mode and x1 mode respectively. The functional block diagram can be referred to [Figure 17-28: Encoder mode block diagram](#), which allows filtering and polarity control of the input signal.

Encoders with direction usually have two output signals: a clock signal and a direction signal. Using the clock signal, the rotational speed and position change of the rotary encoder can be calculated, while the direction signal can be used to determine the direction of rotation. In x2 mode, the counter counts at the rising and falling edge of the clock signal; In x1 mode, the counter counts on a single clock edge based on the value of the CC2P bit field, where CC2P 0 corresponds to rising edge sensitivity and CC2P 1 corresponds to falling edge sensitivity. The polarity of the direction signal on the CH1 channel is set by the CC1P bit: CC1P is 0 corresponding to positive polarity (counts up when CH1 is high, counts down when CH1 is low) and CC1P is 1 corresponding to negative polarity (counts up when CH1 is low, counts down when CH1 is high).

The following figure shows an example of counter counting in clock plus direction encoder mode, with neither CH1 nor CH2 inputs inverted in the example:

Figure 17-31 Example of clock + direction encoder mode counting



### 17.3.2.10 Directional clock encoder mode

The timer also supports directional clock encoder mode, which requires configuring the SMS bit field of the slave mode control register ATIM\_SMCR to be 0xC or 0xD, corresponding to x2 mode or x1 mode, respectively. The functional block diagram can be referred to [Figure 17-28: Encoder mode block diagram](#), which allows filtering and polarity control of the input signal.

The directional clock encoder provides an up count clock line and a down count clock line, respectively, depending on the direction of rotation. In x2 mode, the counter counts on the rising and falling edges of either of the two clock lines; the CC1P and CC2P bits encode the clock idle state, with CCyP being 0 corresponding to high level idle and CCyP being 1 corresponding to low level idle. In x1 mode, the counter counts on a single clock edge based on the value of the CC1P and CC2P bit fields, with CCyP being 0 corresponding to falling-edge sensitive and high-level idle, and CCyP being 1 corresponding to rising-edge sensitive and low-level idle.

The following table shows the relationship between the counting direction and the encoder signal in different modes:

Table 17-6 Relationship between counting direction and encoder signal

| Mode                | SMS  | Signal level |        | TI1FP1     |            | TI2FP2   |          |
|---------------------|------|--------------|--------|------------|------------|----------|----------|
|                     |      | TI2FP2       | TI1FP1 | Rising     | Falling    | Rising   | Falling  |
| x2 mode<br>CCyP = 0 | 1100 | High         | High   | Down count | Down count | Up count | Up count |
|                     |      | Low          | Low    | No count   | No count   | No count | No count |
| x2 mode<br>CCyP = 1 |      | High         | High   | No count   | No count   | No count | No count |
|                     |      | Low          | Low    | Down count | Down count | Up count | Up count |
| x1 mode<br>CCyP = 0 | 1101 | High         | High   | No count   | Down count | No count | Up count |
|                     |      | Low          | Low    | No count   | No count   | No count | No count |
| x1 mode<br>CCyP = 1 |      | High         | High   | No count   | No count   | No count | No count |
|                     |      | Low          | Low    | Down count | No count   | Up count | No count |

The following figure shows an example of counting in directional clock encoder mode:

Figure 17-32 Example of directional clock encoder mode counting (CC1P = CC2P = 0)

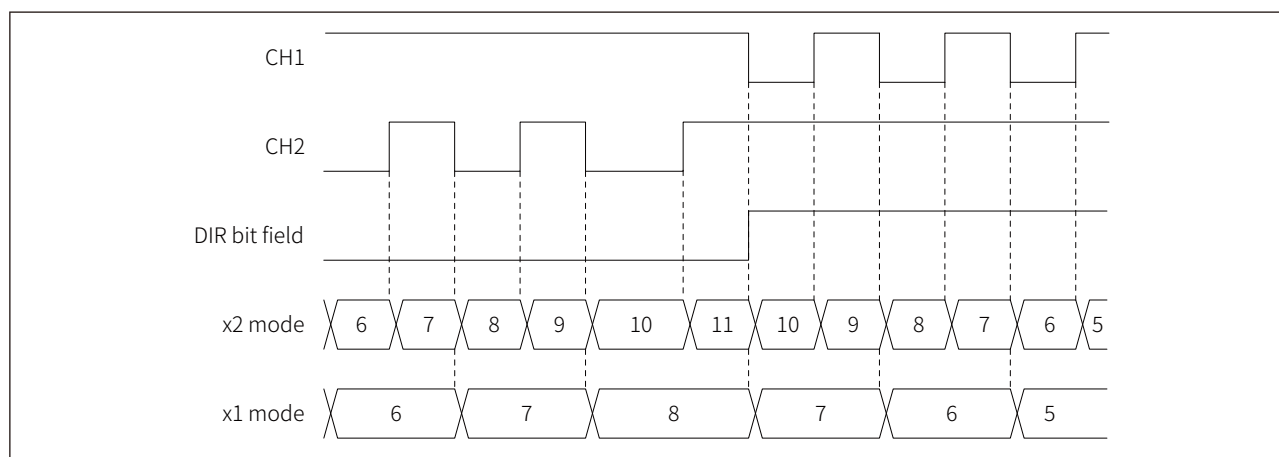
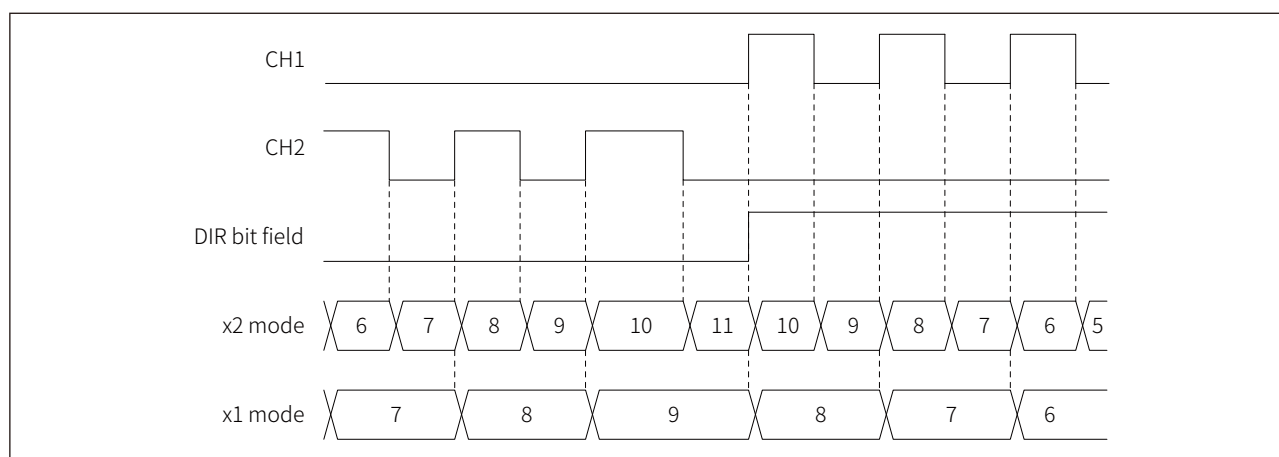


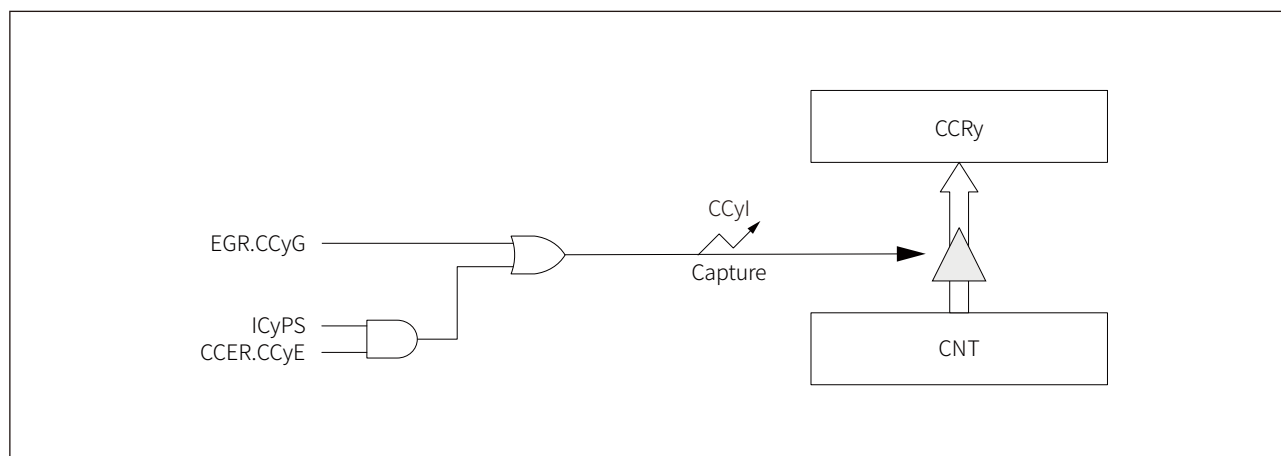
Figure 17-33 Example of directional clock encoder mode counting (CC1P = CC2P = 1)



### 17.3.3 Input capture function

ATIM supports the input capture function. When the CCyS bit field of the capture mode registers ATIM\_CCMRxCAP is set to a non-zero value, the CCy channel is configured as an input, and at the same time, the input mapping of the corresponding capture channel ICy is specified. Input capture is supported to be triggered by software or hardware, and its function block diagram is shown below:

Figure 17-34 Input capture mode block diagram



**Software trigger:** When ATIM\_EGR.CCyG is set to 1, a capture will be triggered by software immediately, and the value of the current counter CNT is latched into the capture/compare register CCRy of the corresponding channel to complete a capture, and the CCyG bit is cleared by the hardware automatically after the completion of the capture.

**Hardware trigger:** When a valid edge is detected on the input channel TIy, the value of the current counter CNT is latched into the capture/compare register CCRy of the corresponding channel to complete a capture. The valid edge that triggers the capture is configured through the CCyNP and CCyP bit fields of the capture/compare enable register ATIM\_CCER as shown in the table below:

Table 17-7 Input capture mode configuration

| ATIM_CCER register    |                      | Capture mode configuration           |
|-----------------------|----------------------|--------------------------------------|
| CCyNP bit field value | CCyP bit field value |                                      |
| 0                     | 0                    | Rising edge capture                  |
| 0                     | 1                    | Falling edge capture                 |
| 1                     | 1                    | Both rising and falling edge capture |

When a capture occurs, the capture/compare interrupt flag ATIM\_ISR.CCyIF of the corresponding channel is set to 1 by hardware and:

- If interrupt is enabled (setting ATIM\_DIER.CCyIE to 1), an interrupt request is generated.
- If DMA is enabled (setting ATIM\_DIER.CCyDE to 1, and ATIM\_CR2.CCDS to 0), a DMA request is generated.
- If the ATIM\_ISR.CCyIF flag bit is already high when the capture event occurs, the repeat capture flag bit ATIM\_ISR.CCyOF will be set to 1 by hardware.
- Setting ATIM\_ICR.CCyIF to 0 or reading capture register CCRy clears the ATIM\_ISR.CCyIF flag bit, and setting ATIM\_ICR.CCyOF to 0 clears the ATIM\_ISR.CCyOF flag bit.

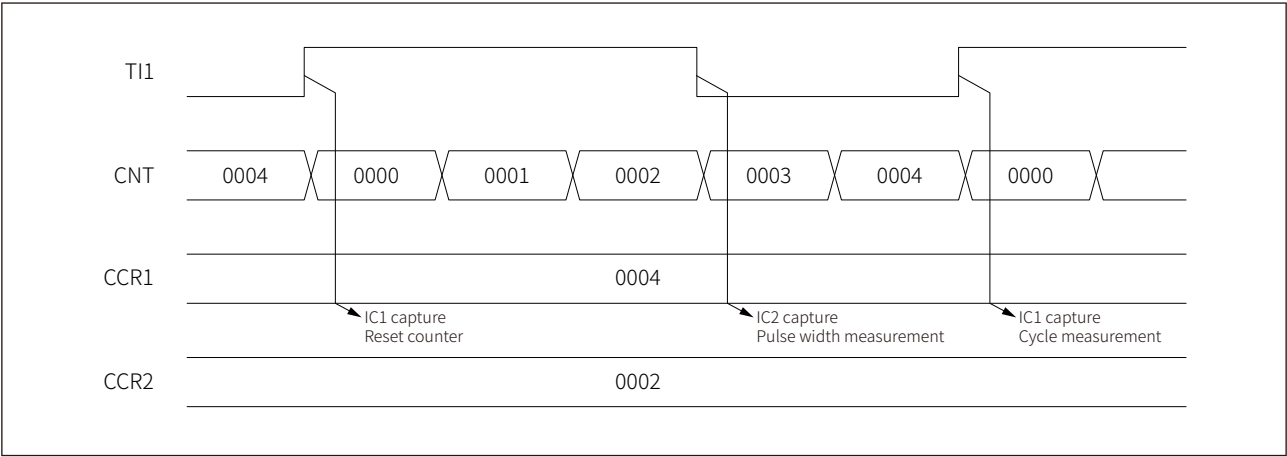
17.3.3.1 PWM input mode

PWM input mode is an application of input capture mode. The input capture function supports the same channel input signal to be captured by two capture registers. Combined with the slave mode reset function, the period and pulse width of the PWM input signal can be easily measured.

The PWM input mode can only be used with the TI1 and TI2 signals because only TI1FP1 and TI2FP2 are connected to the slave mode controller.

The following figure shows an example of the PWM signal timing when it is input from the TI1 channel:

Figure 17-35 PWM input mode timing



### 17.3.3.2 Input capture source

The ATIM input capture source can be the external ATIM\_CHy pin or other peripherals on-chip, configured through the TlySEL bit field of the TI input select registers (ATIM\_TISEL1 and ATIM\_TISEL2).

When ATIM\_TISELx.TlySEL is 0x0, the source of the Tly channel input capture signal is the external ATIM\_CHy pin, in which case the corresponding pin needs to be configured as a multiplexing function through the GPIO multiplexing function registers (GPIOx\_AFRH and GPIOx\_AFRL).

When ATIM\_TISELx.TlySEL is 0x1~0xD, the Tly channel input capture signals come from other peripherals on-chip, as shown in the following table:

Table 17-8 Advanced timer input capture source

| TlySEL bit field value | Input capture source for the Tly channel |
|------------------------|------------------------------------------|
| 0000                   | ATIM_CHy pin                             |
| 0001                   | VC1_OUT/VC3_OUT                          |
| 0010                   | VC2_OUT/VC4_OUT                          |
| 0011                   | UART1_RXD                                |
| 0100                   | UART2_RXD                                |
| 0101                   | UART3_RXD                                |
| 0110                   | HSE_FAULT                                |
| 0111                   | LSE_FAULT                                |
| 1000                   | RTC_OUT                                  |
| 1001                   | LSI_OUT                                  |
| 1010                   | BTIM1_Trgo                               |
| 1011                   | BTIM2_Trgo                               |
| 1100                   | BTIM3_Trgo                               |
| 1101                   | GTIM1_Trgo                               |
| 1110                   | GTIM2_Trgo                               |
| 1111                   | MCO_OUT                                  |



### 17.3.4 Output compare function

The ATIM supports the output compare function. Setting the CCyS bit field of the compare mode register ATIM\_CCMRxCMP to 0 configures the CCy channel as an output. In the output compare mode, the value of the current counter CNT is compared with the value of the capture/compare register CCRy of the corresponding channel CCy. When the two match, the reference signal OCyREF is output to a settable level state, and a compare interrupt is generated at the same time. The reference signal is output by the CHy and CHyN pins after passing through the output control unit, and its output logic can be found in [17.3.1.10 Output compare channel](#).

The output action of the reference signal OCyREF is set by the OCyM bit field of the compare mode register ATIM\_CCMRxCMP, as shown in the following table:

Table 17-9 Output compare mode configuration

| OCyM bit field value | Compare mode configuration                                   |
|----------------------|--------------------------------------------------------------|
| 0000                 | When comparing and matching, OCyREF keeps the original level |
| 0001                 | When comparing and matching, OCyREF is set to 1              |
| 0010                 | When comparing and matching, OCyREF is set to 0              |
| 0011                 | When comparing and matching, OCyREF flips                    |
| 0100                 | Force OCyREF to low level                                    |
| 0101                 | Force OCyREF to high level                                   |
| 0110                 | PWM mode 1                                                   |
| 0111                 | PWM mode 2                                                   |
| 1000                 | Retriggerable OPM mode 1                                     |
| 1001                 | Retriggerable OPM mode 2                                     |
| 1011                 | Count direction output                                       |
| 1100                 | Combined PWM mode 1                                          |
| 1101                 | Combined PWM mode 2                                          |
| 1110                 | Asymmetric PWM mode 1                                        |
| 1111                 | Asymmetric PWM mode 2                                        |

When a compare match occurs, the capture/compare interrupt flag ATIM\_ISR.CCyIF of the corresponding channel is set to 1 by hardware while:

- If interrupt is enabled (setting ATIM\_DIER.CCyIE to 1), an interrupt request will be generated.
- Setting ATIM\_ICR.CCyIF to 0 clears the ATIM\_ISR.CCyIF flag bit.

The capture/compare register ATIM\_CCRy has a cache function and the use of the cache function is selected through the OCyPE bit field. When OCyPE is 0, the comparison cache function of channel CHy is disabled, and the ATIM\_CCRy register can be updated by software at any time; the updated value takes effect immediately and affects the output waveform; When OCyPE is 1, the comparison cache function of channel CHy is enabled, and updating the ATIM\_CCRy register will not take effect immediately, and the value of ATIM\_CCRy register will be updated to the valid register only when the update event UEV occurs.





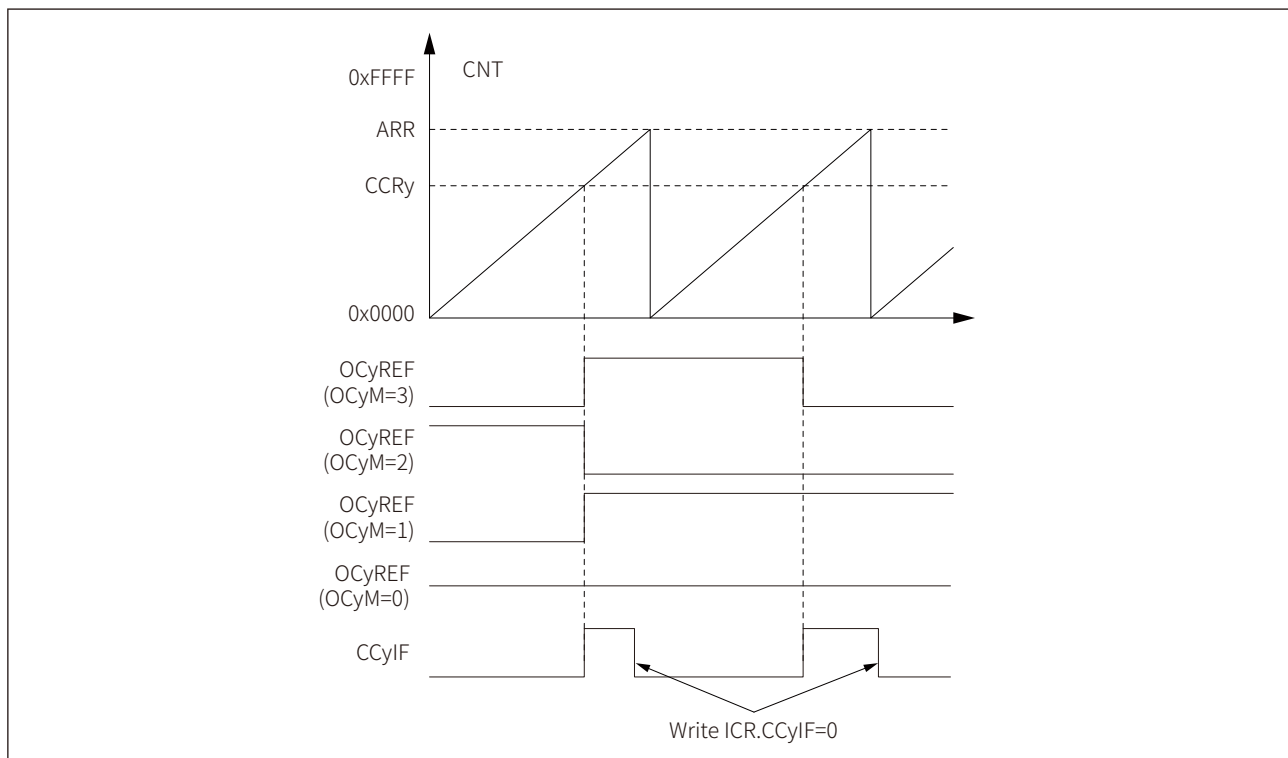
### 17.3.4.1 Match output

The match output mode is used to control the output waveform or to indicate that a certain time period has passed.

- Set the OCyM bit field to 0x0, when comparing and matching, the reference signal OCyREF keeps its level while the ATIM\_ISR.CCyIF flag bit field is set to 1.
- Set the OCyM bit field to 0x1, when comparing and matching, the reference signal OCyREF is set to high level while the ATIM\_ISR.CCyIF flag bit field is set to 1.
- Set the OCyM bit field to 0x2, when comparing and matching, the reference signal OCyREF is set to low level while the ATIM\_ISR.CCyIF flag bit field is set to 1.
- Set the OCyM bit field to 0x3, when comparing and matching, the reference signal OCyREF is flipped while the ATIM\_ISR.CCyIF flag bit field is set to 1.

The following figure shows an example of the timing for each compare-match output mode, with the example being the edge-aligned incremental mode:

Figure 17-36 Compare match output mode



### 17.3.4.2 Forced output

In forced output mode, the output comparison signal can be forced to a high or low state directly by software without relying on the comparison result of the capture/compare register ATIM\_CCRy and the count register ATIM\_CNT.

Set the OCyM bit field of the compare mode registers ATIM\_CCMRxCMP to 0x4 to force the reference signal OCyREF to a low level; set the OCyM bit field to 0x5 to force the reference signal OCyREF to a high level.

In forced output mode, the comparison between the ATIM\_CCRy register and counter ATIM\_CNT is still in progress, and the corresponding flags are set to 1. The corresponding interrupt/DMA requests are also generated.

### 17.3.4.3 PWM output

Pulse width modulation (PWM) mode can generate a signal with a frequency determined by the reload register ATIM\_ARR and a duty cycle determined by the capture/comparison register ATIM\_CCRy.

In order to prevent the output PWM signal from being incorrect when modifying the ARR register, it is necessary to set ATIM\_CR1.ARPE to 1 to enable the cache function of the ARR register; in order to prevent the output PWM signal from being incorrect when modifying the CCRy register, it is necessary to set ATIM\_CCMRxCMP.OCyPE to 1 to enable the cache function of the CCRy register. Only when the update event UEV occurs the value of the preloaded register will be transferred to the shadow register; before starting the timer after configuring various parameters, it is necessary to write 1 to the ATIM\_EGR.UG bit to initialise all registers.

The PWM mode can be selected independently for each channel by simply writing 0x6 (PWM mode 1) or 0x7 (PWM mode 2) to the OCyM bit field of the compare mode register, ATIM\_CCMRxCMP, with the output states shown in the following table:

Table 17-10 PWM mode 1/2 output state

| Operating mode                           | Counting direction | PWM mode 1                 | PWM mode 2                 |
|------------------------------------------|--------------------|----------------------------|----------------------------|
| Edge aligned mode<br>Center aligned mode | Up count           | CNT < CCRy, OCyREF is high | CNT < CCRy, OCyREF is low  |
|                                          | Down count         | CNT > CCRy, OCyREF is low  | CNT > CCRy, OCyREF is high |

The polarity of OCy can be set through the CCyP bit field of the ATIM\_CCER register, which can be set to active high or active low. The OCy output is enabled by a combination of the CCyE, CCyNE, MOE, OSS1, and OSSR bits (ATIM\_CCER and ATIM\_BDTR registers).

The following are examples of waveforms for each PWM mode in different count modes, where ATIM\_ARR is 0x08:

Figure 17-37 PWM mode 1, edge-aligned mode, count up

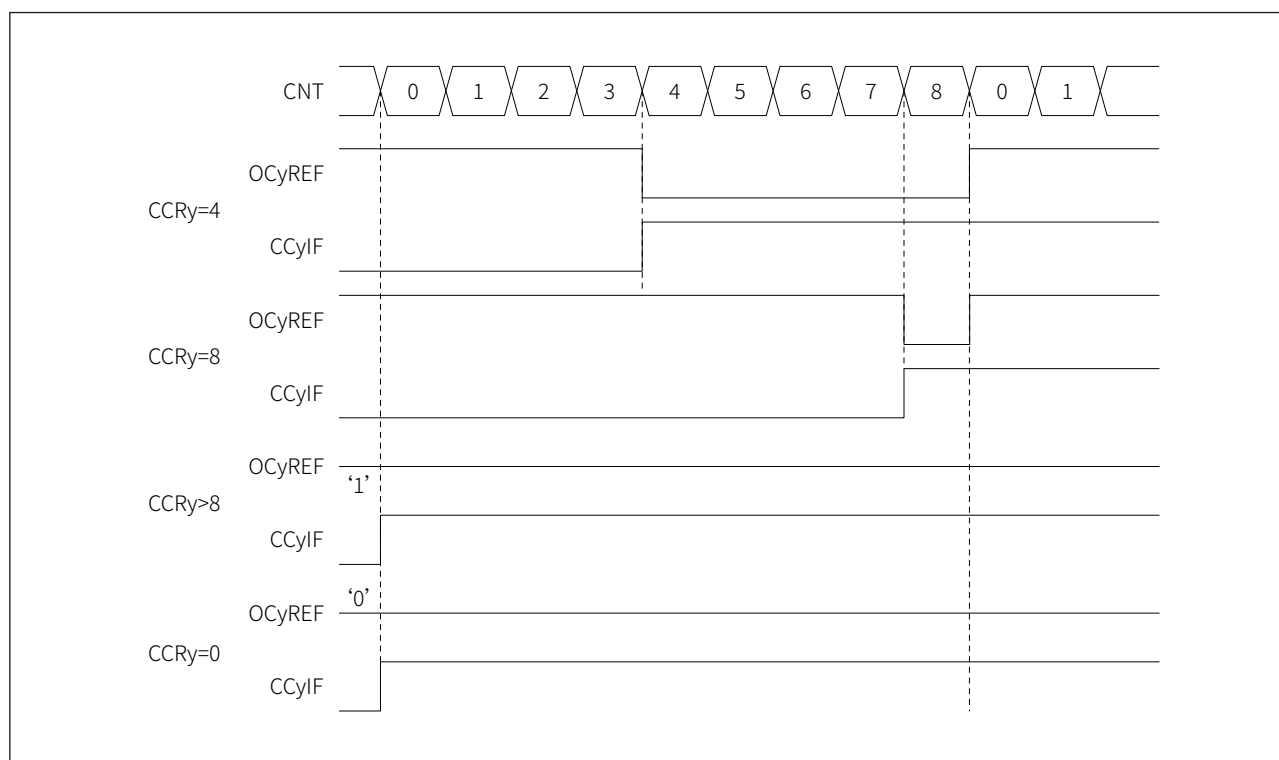


Figure 17-38 PWM mode 1, edge-aligned mode, count down

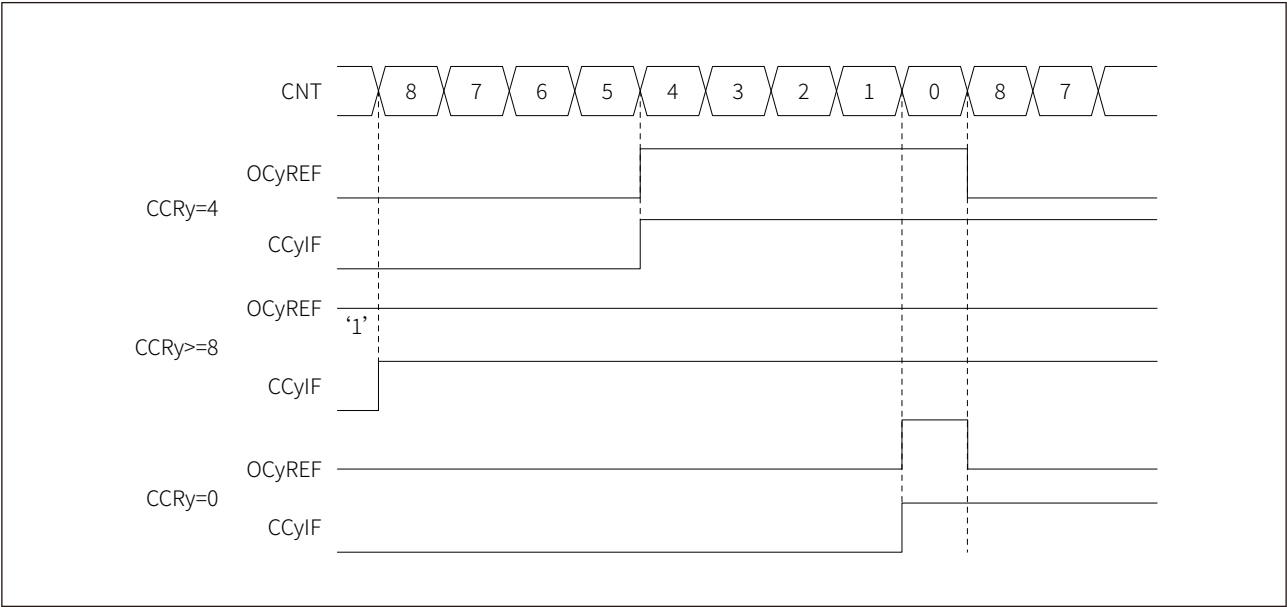
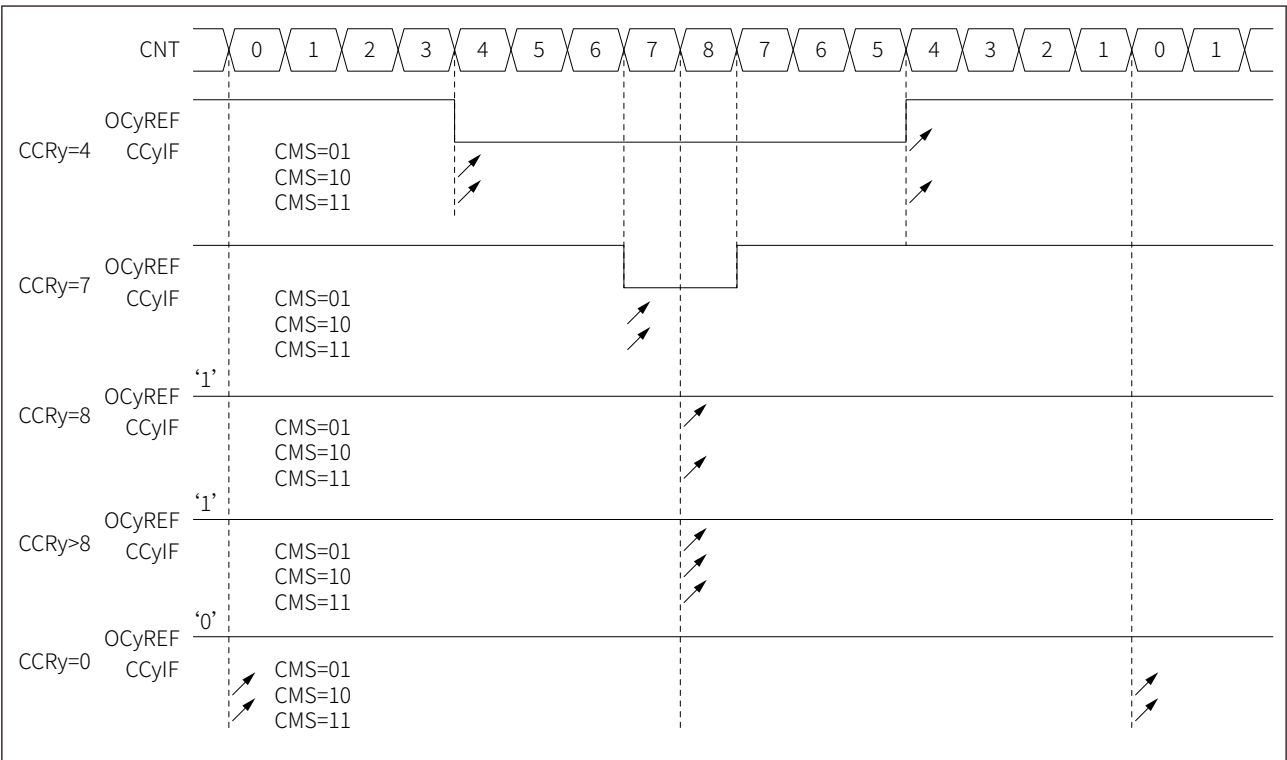


Figure 17-39 PWM mode 1, center-aligned mode



### 17.3.4.4 Asymmetric PWM output

The asymmetric PWM output is relative to the symmetric PWM output when counting in center-aligned mode. In the asymmetric PWM output mode, two PWM signals with programmable phase shifts are allowed to be generated in centre-aligned mode, with the frequency determined by the reload register ATIM\_ARR, and the duty cycle and phase shifts determined by the two capture/comparison registers ATIM\_CCRy, where one of the CCRy registers controls the PWM during counting up, and the other CCRy register controls the PWM during counting down, the details are shown below:

- OC1REFC is controlled by CCR1 (count up) and CCR2 (count down).
- OC2REFC is controlled by CCR2 (count up) and CCR1 (count down).
- OC3REFC is controlled by CCR3 (count up) and CCR4 (count down).
- OC4REFC is controlled by CCR4 (count up) and CCR3 (count down).
- OC5REFC is controlled by CCR5 (count up) and CCR6 (count down).
- OC6REFC is controlled by CCR6 (count up) and CCR5 (count down).

Each channel can independently select the asymmetric PWM mode by simply writing 0xE (asymmetric PWM mode 1) or 0xF (asymmetric PWM mode 2) to the OCyM bit field of the compare mode register ATIM\_CCMRxCMP.

When the channel is set to be used as an asymmetric PWM output channel, the other auxiliary channel can still be used. For example, when channel 1 is configured to generate an OC1REFC signal in asymmetric PWM mode 1, channel 2 can still be configured to output an OC2REF signal in PWM mode 2 or to output an OC2REFC signal in asymmetric PWM mode 2.

The following figure shows an example of a waveform for asymmetric PWM mode 1/2, where ARR = 8, CCR1 = 0, CCR2 = 8, CCR3 = 3, and CCR4 = 5:

Figure 17-40 Example of asymmetric PWM mode 1

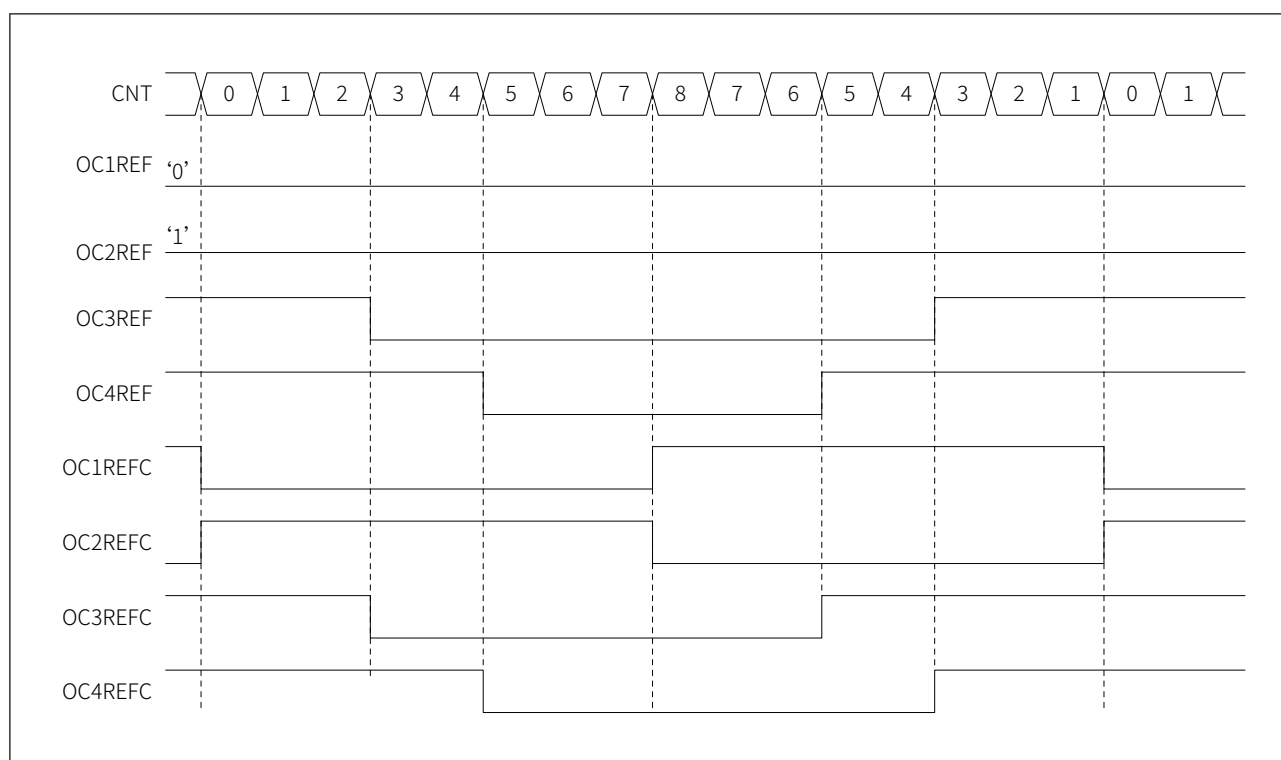
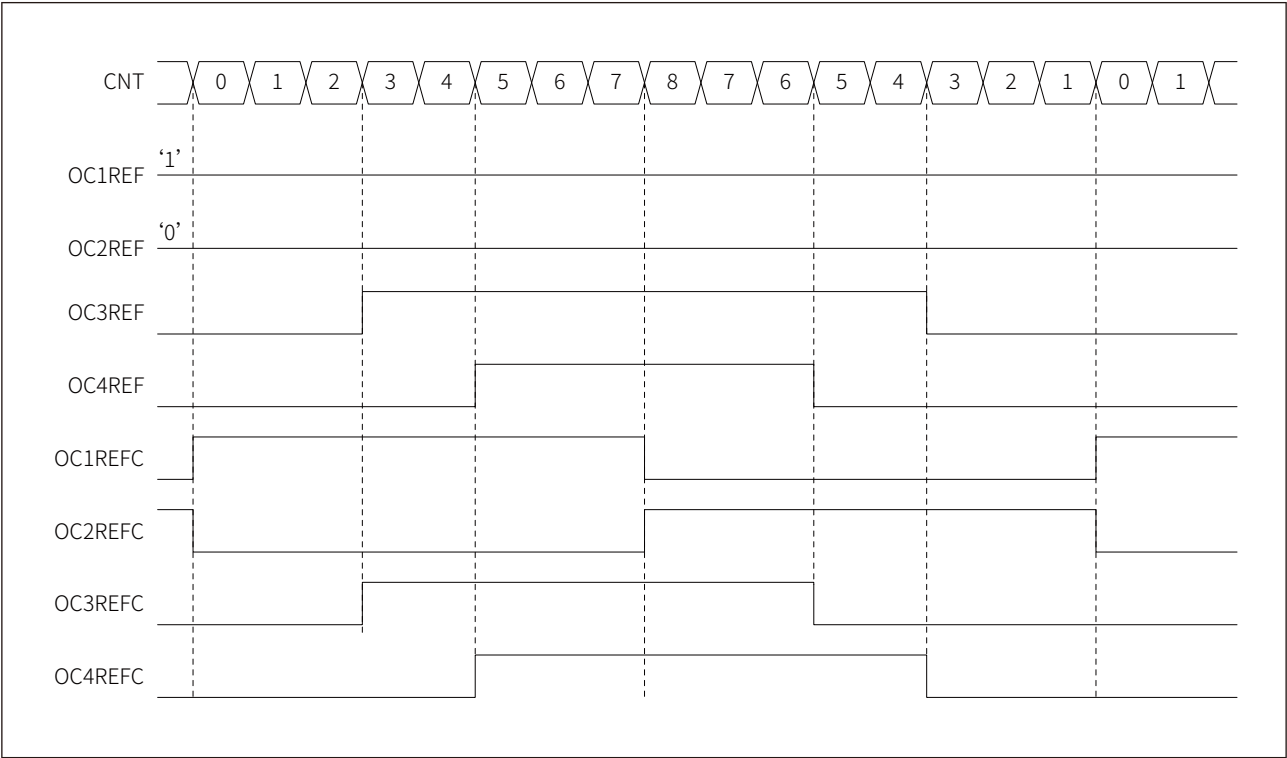


Figure 17-41 Example of asymmetric PWM mode 2



### 17.3.4.5 Combined PWM output

In combined PWM output mode, two PWM signals with programmable delay and phase shift are allowed to be generated in either edge-aligned or center-aligned mode, with the frequency determined by the reload register ATIM\_ARR, and the duty cycle and delay determined by the two capture/comparison registers ATIM\_CCRy.

In this mode, the OCyREFC signal is generated by the logic or operation or the combination of logic and operation of the two reference signals OCyREF, and the combined PWM mode can be independently selected through the OCyM bit field of the comparison mode register ATIM\_CCMRxCMP as shown in the following table:

**Table 17-11 Combined PWM mode configuration**

| OCyM bit field value         | Output state                                                                                                                                                                                                                                                       |
|------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0xC<br>(Combined PWM mode 1) | OC1REFC (or OC2REFC) is generated by the logical OR operation of OC1REF and OC2REF.<br>OC3REFC (or OC4REFC) is generated by the logical OR operation of OC3REF and OC4REF.<br>OC5REFC (or OC6REFC) is generated by the logical OR operation of OC5REF and OC6REF.  |
| 0xD<br>(Combined PWM mode 2) | OC1REFC (or OC2REFC) is generated by the logical AND operation of OC1REF and OC2REF.<br>OC3REFC (or OC4REFC) is generated by the logic AND operation of OC3REF and OC4REF.<br>OC5REFC (or OC6REFC) is generated by the logical AND operation of OC5REF and OC6REF. |

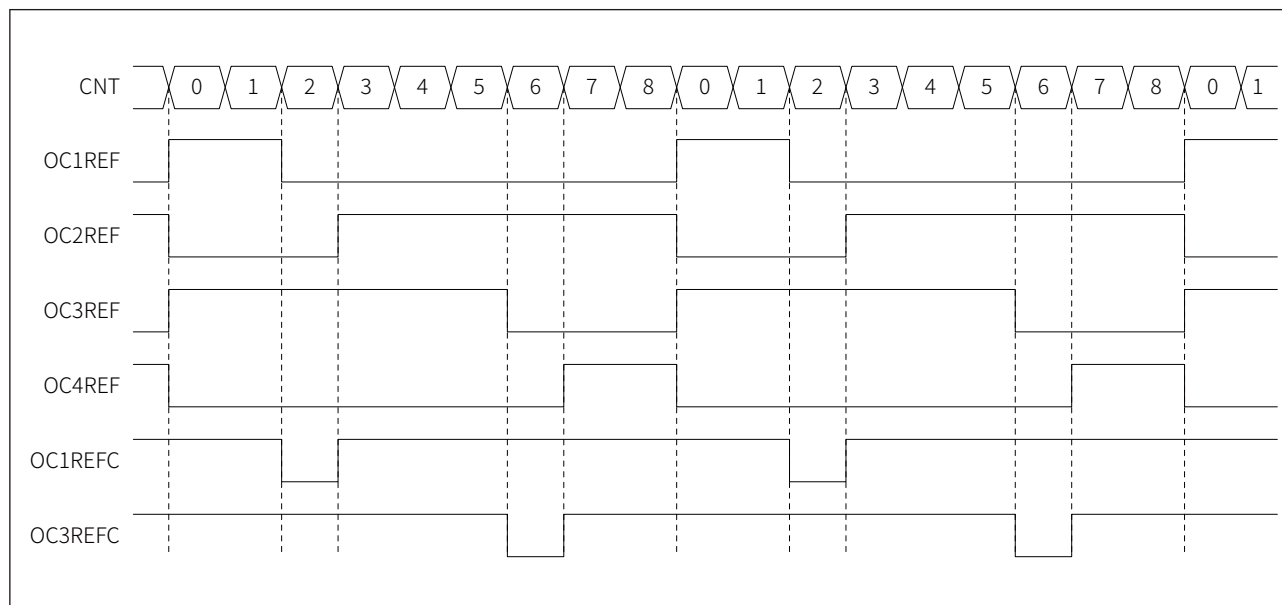
When the channel is set to be used as a combination PWM output channel, the other auxiliary channel can still be used, but must be set to the opposite PWM mode. For example, channel 2 can still be used when channel 1 is configured to generate the OC1REFC signal in combined PWM mode 1, and channel 2 can be configured to output the OC2REF signal in PWM mode 2.

The following figure shows an example of the signals for the combined PWM mode 1 in the edge-aligned incremental mode, which is configured as follows:

- The reload value ARR is 8.
- Channel 1 is configured as combined PWM mode 1, CCR1 = 2.
- Channel 2 is configured as PWM mode 2, CCR2 = 3.
- Channel 3 is configured as combined PWM mode 1, CCR3 = 6.
- Channel 4 is configured as PWM mode 2, CCR4 = 7.



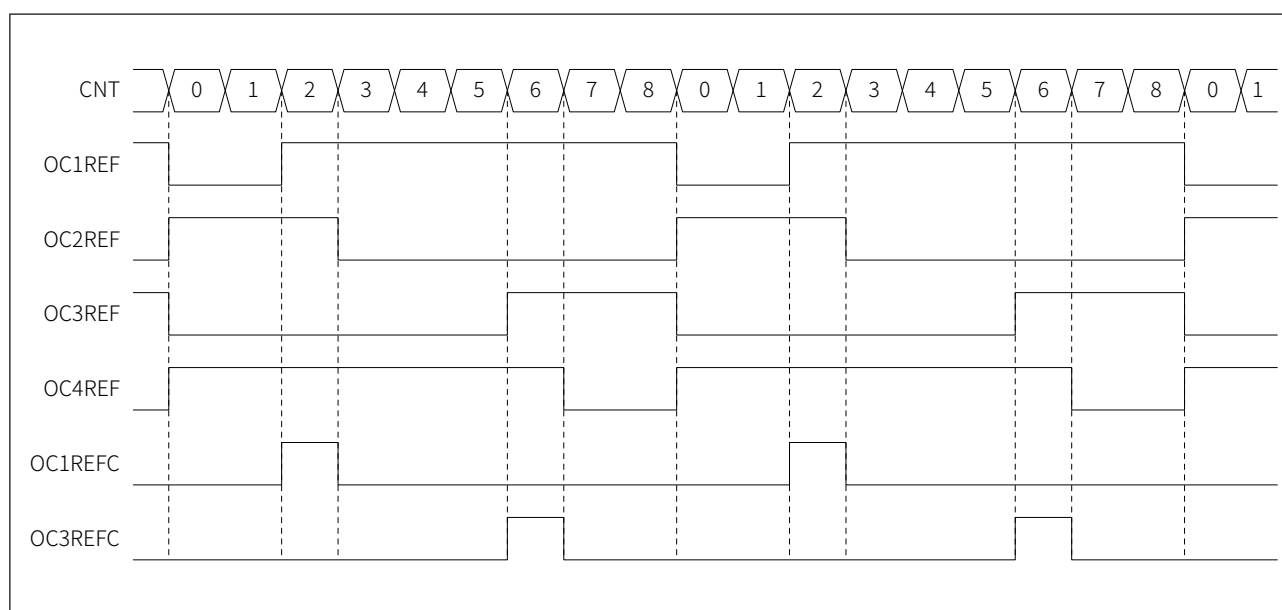
Figure 17-42 Example of combined PWM mode 1



The following figure shows an example of the signals for the combined PWM mode 2 in the edge-aligned incremental mode, which is configured as follows:

- The reload value ARR is 8.
- Channel 1 is configured as combined PWM mode 2, CCR1 = 2.
- Channel 2 is configured as PWM mode 1, CCR2 = 3.
- Channel 3 is configured as combined PWM mode 2, CCR3 = 6.
- Channel 4 is configured as PWM mode 1, CCR4 = 7.

Figure 17-43 Example of combined PWM mode 2



### 17.3.4.6 Phase shift PWM mode

In motor applications, it is necessary to sample the current through the ADC at a specific moment between the rising edges of the two PWM waveforms. When the interval between the rising edges of the two PWM waveforms is too short to complete the current sampling, the edge of one of the PWM waveforms can be adjusted through the phase-shift PWM mode to increase the interval between the rising edges of the two PWM waveforms under the condition of keeping the duty cycle unchanged.

Set the GC5Cy bit field of the ATIM\_CCR5 register to 1 to enable phase-shift PWM mode 5, and the OCyREFC output states are as follows:

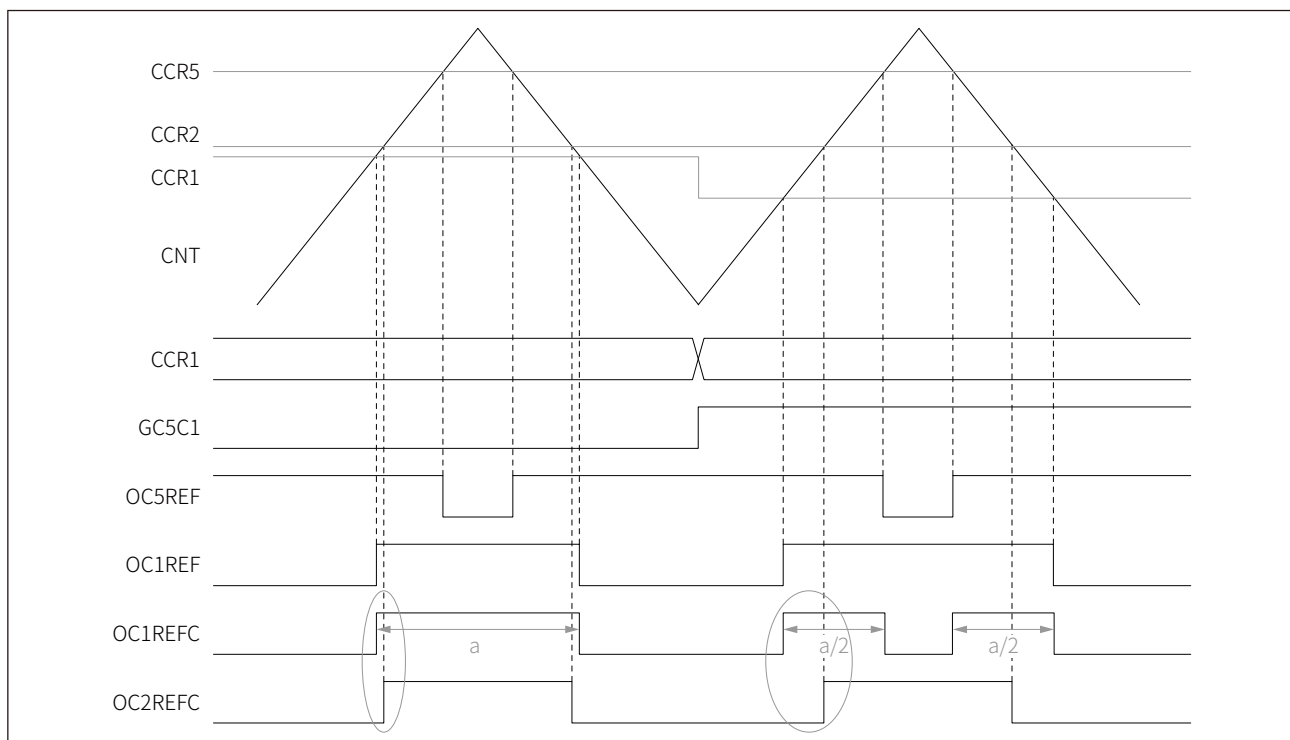
- Set GC5C1 to 1, OC1REFC outputs the logical AND of OC1REF and OC5REF.
- Set GC5C2 to 1, OC2REFC outputs the logical AND of OC2REF and OC5REF.
- Set GC5C3 to 1, OC3REFC outputs the logical AND of OC3REF and OC5REF.
- Set GC5C4 to 1, OC4REFC outputs the logical AND of OC4REF and OC5REF.
- Set GC5C6 to 1, OC6REFC outputs the logical AND of OC6REF and OC5REF.

Set the GC6Cy bit field of the ATIM\_CCR6 register to 1 to enable phase-shift PWM mode 6, and the OCyREFC output states are as follows:

- Set GC6C1 to 1, OC1REFC outputs OC1REF (count up) and OC6REF (count down).
- Set GC6C2 to 1, OC2REFC outputs OC2REF (count up) and OC6REF (count down).
- Set GC6C3 to 1, OC3REFC outputs OC3REF (count up) and OC6REF (count down).
- Set GC6C4 to 1, OC4REFC outputs OC4REF (count up) and OC6REF (count down).
- Set GC6C5 to 1, OC5REFC outputs OC5REF (count up) and OC6REF (count down).

The following figure shows the timing diagram for increasing the interval between the rising edge of OC1REFC and the rising edge of OC2REFC in phase-shifted PWM mode 5 by adjusting CCR1 and GC5C1:

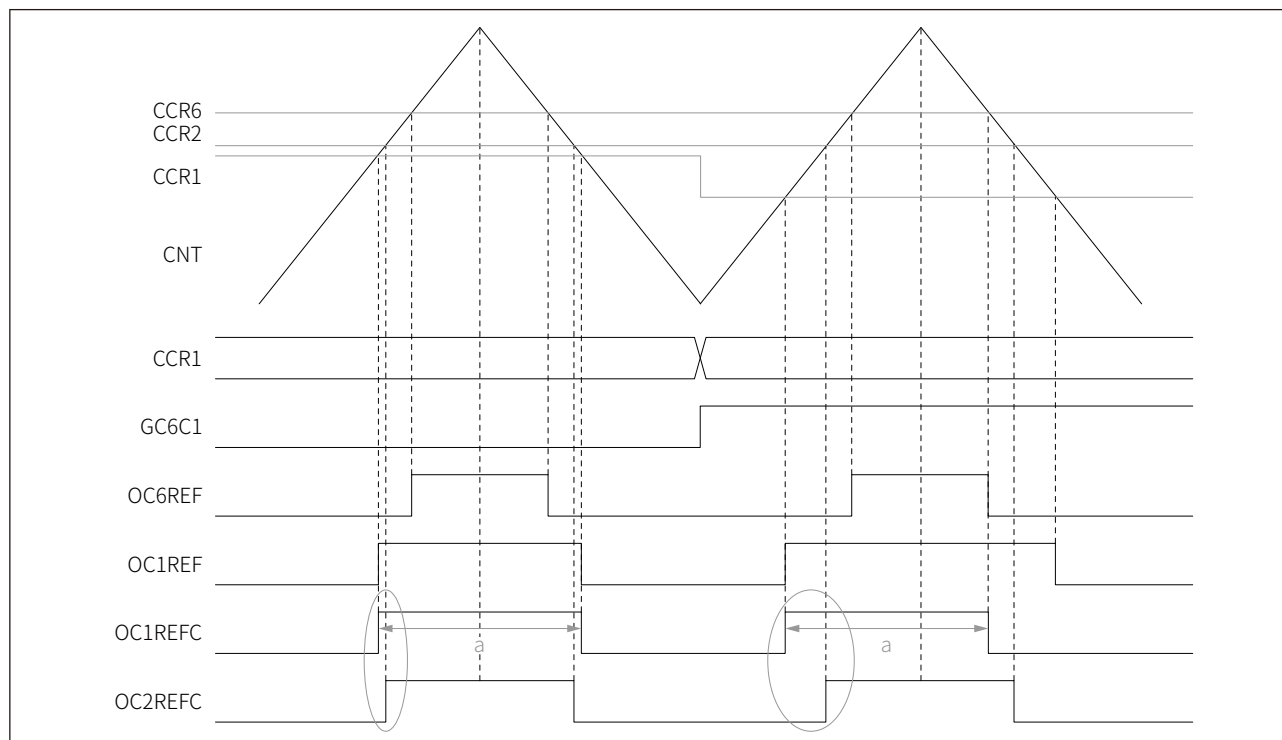
Figure 17-44 Example of phase-shifted PWM mode 5



The following figure shows the timing diagram for increasing the interval between the rising edge of OC1REFC and the rising edge of OC2REFC in phase-shifted PWM mode 6, achieved by adjusting CCR1 and GC6C1:



Figure 17-45 Example of phase-shifted PWM mode 6



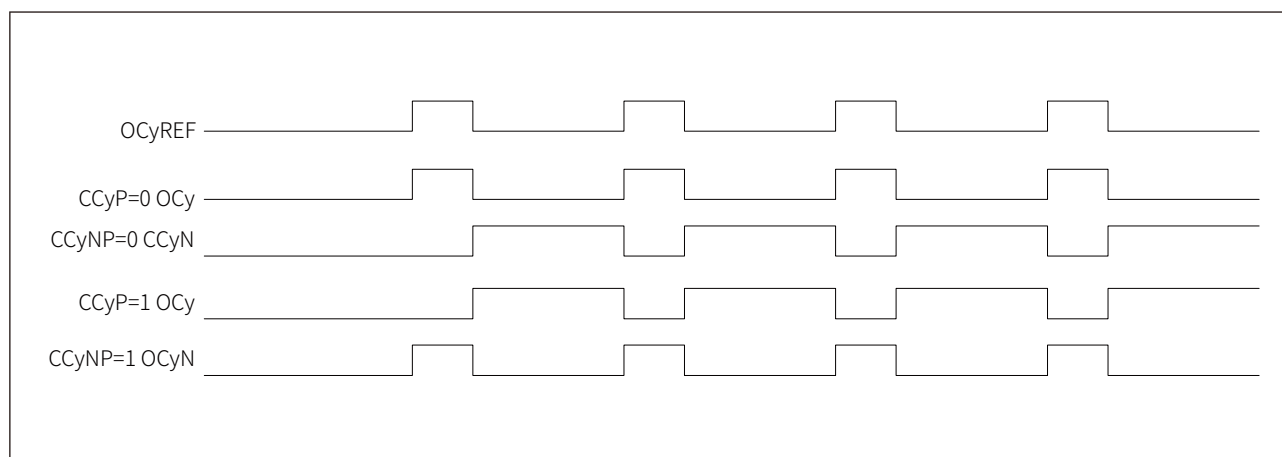
#### 17.3.4.7 Complementary output and dead-time insertion

The ATIM can output 6 pairs of complementary PWM signals and supports symmetric or asymmetric dead time insertion.

The main output OCy and the complementary output OCyN can be independently set in output polarity and are controlled via the CCyP and CCyNP bit fields of the ATIM\_CCER register, respectively.

The main output OCy and complementary output OCyN are activated by a combination of several control bits: the CCyE and CCyNE bits in the ATIM\_CCER register, the MOE, OSSl, and OSSR bits in the ATIM\_BDTR register, and the OISy and OISyN bits in the ATIM\_CR2 register. For more details, see [Table 17-13: Output control bits for complementary channels OCy and OCyN with brake function](#). It should be noted that at the moment of switching to the idle state (MOE becomes 0), the deadband is still active.

Figure 17-46 Complementary PWM output waveform



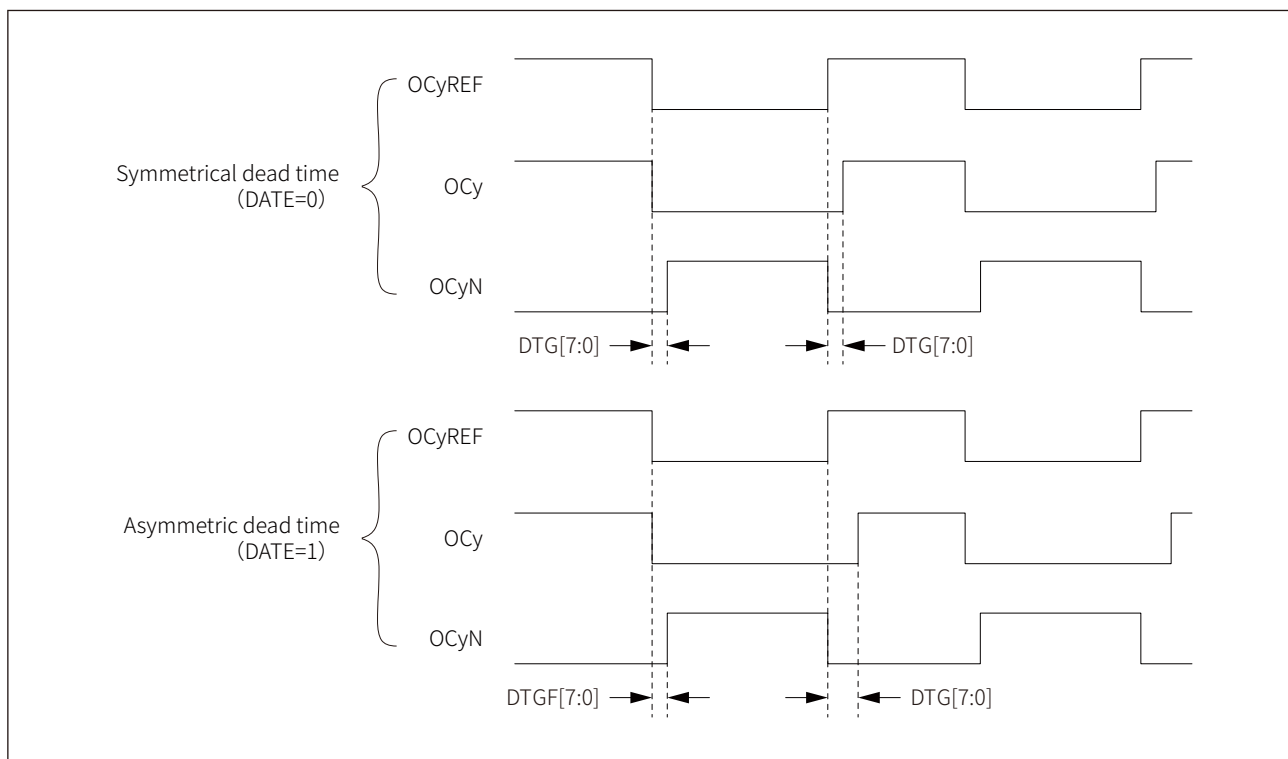
The ATIM supports symmetric or asymmetric dead insertion times, which are the same for all channels. When ATIM\_DTR2.DTAE is 0, the dead time of the rising edge and the falling edge is the same and is defined by DTG[7:0] in the ATIM\_BDTR register; when ATIM\_DTR2.DTAE is 1, the dead time of the rising edge is defined by DTG[7:0] and the dead time of the falling edge is defined by DTGF[7:0]. The DTAE bit must be written before enabling the counter and must not be modified when CEN is 1. DTG[7:0] is defined as shown in the table below and DTGF[7:0] is defined as DTG[7:0]:

Table 17-12 Dead-time settings

| ATIM_BDTR.DTG  | Step length | Dead-time                                         | When $t_{DTS} = 125\text{ns}$ , dead-time range |
|----------------|-------------|---------------------------------------------------|-------------------------------------------------|
| DTG[7:5] = 0xx | $1t_{DTS}$  | $\text{DTG}[7:0] \times t_{DTS}$                  | 0~15.875 $\mu\text{s}$                          |
| DTG[7:5] = 10x | $2t_{DTS}$  | $(64 + \text{DTG}[5:0]) \times 2 \times t_{DTS}$  | 16~31.750 $\mu\text{s}$                         |
| DTG[7:5] = 110 | $8t_{DTS}$  | $(32 + \text{DTG}[4:0]) \times 8 \times t_{DTS}$  | 32~63 $\mu\text{s}$                             |
| DTG[7:5] = 111 | $16t_{DTS}$ | $(32 + \text{DTG}[4:0]) \times 16 \times t_{DTS}$ | 64~126 $\mu\text{s}$                            |

The following figure shows the relationship between the complementary PWM output signal and the reference signal OCyREF for symmetric and asymmetric deadband insertion times, with CCyP = 0 and CCyNP = 0 in the example:

Figure 17-47 Complementary PWM output waveform with dead-time insertion



A preload mechanism may be used to update the dead time value on the fly during PWM operation. When the DTPE bit is set to 1, the dead time value preload is enabled and the preload value is loaded into the valid register at the next update event. If the DTPE bit is enabled while the counter is enabled, any new values written since the last update are discarded and the previous values are used.

### 17.3.4.8 Brake function

ATIM supports two brake inputs BRK and BRK2, and when the brake signal is active, the brake circuit turns off the PWM output and forces it to a predefined safe state.

The brake state supports output from GPIO port, which requires users to configure the corresponding GPIO port as digital output and select function multiplexing at the same time. When ATIM is in brake state, it outputs high level; when ATIM is in non-brake state, it outputs low level. Users can utilize this feature to achieve more board-level protection.

Brake channel sources include system-level faults (clock failures and parity errors, etc.) and application faults (from input pins and built-in comparators) that force the output to a predefined level (active or inactive) after a dead time duration. Brake 2 channel sources include application faults that can only force the output to an invalid state. BRK inputs have higher priority than BRK2 inputs.

1. Brake (BRK) trigger source:

- Software braking, set BG bit field of ATIM\_EGR register to 1 to generate braking event, BG bit is automatically cleared by hardware.
- The compare output of voltage comparator VC1/2, the relevant bit field of the ATIM\_AF1 register, sets the active polarity and enables the input.
- System brake request, enabled via the appropriate bit field of SYSCTRL\_CR2 register:
  - Generate HardFault or Cortex-M0+ LockUp flag
  - LVD output
  - RAM parity error
  - DeepSleep
  - LSE runtime failure
  - HSE runtime failure
- External ATIM\_BK pin input (GPIO function multiplex setting), the relevant bit field of the ATIM\_AF1 register sets active polarity and input enable.

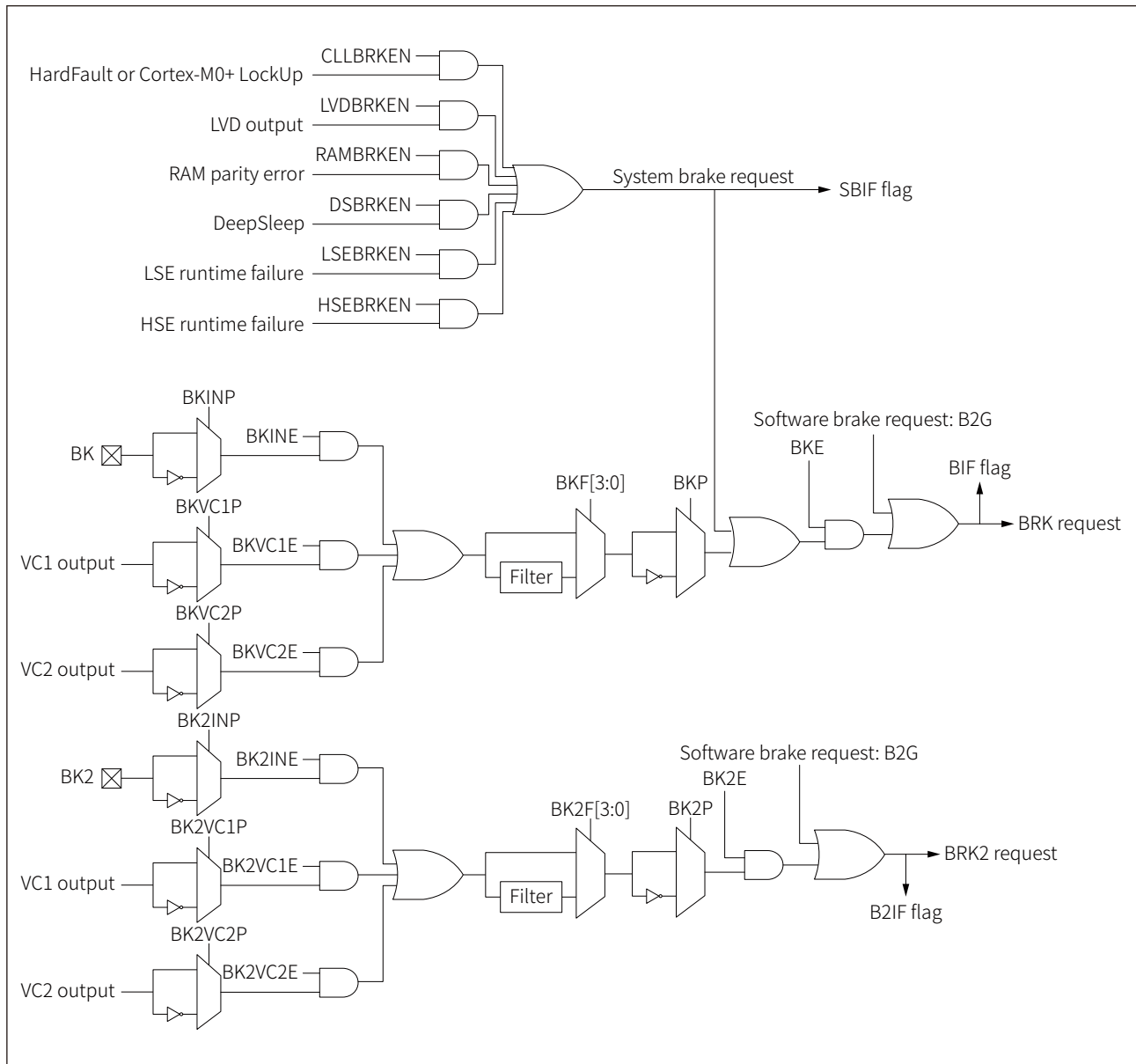
2. Brake 2 (BRK2) trigger source:

- Software braking, set the B2G bit field of ATIM\_EGR register to 1 to generate braking event, B2G bit is automatically cleared by hardware.
- The compare output of voltage comparator VC1/2, the relevant bit field of the ATIM\_AF2 register, sets the active polarity and enables the input.
- External ATIM\_BK2 pin input (GPIO function multiplex setting), the relevant bit field of the ATIM\_AF2 register sets active polarity and input enable.

Before all sources enter the timer BRK or BRK2 input, they are logically OR, as shown in the following figure:



Figure 17-48 Brake and brake 2 circuit



**Attention:** Asynchronous (no clock) operation is only guaranteed when the programmable filter is disabled. If the programmable filter is enabled, fail-safe clock mode must be used to ensure that brake events can be handled.

When a braking event occurs, the MOE bit is cleared asynchronously, leaving the output in an inactive state, idle state or even releasing control to the GPIO controller (selected by the OSS1 bit, see [Table 17-13: Output control bits for complementary channels OCy and OCyN with brake function](#) for details). This feature is enabled even if the MCU oscillator is turned off. If AOE is set, the MOE bit is automatically set again on the next update event (UEV), otherwise MOE remains low until the application sets it again. Because the brake input is active, the MOE bit cannot be set (automatically or by software) and the status flags BIF and B2IF cannot be cleared when the brake input is active.

The following figure shows an example of the output behavior of OCy and OCyN when braking event BRK occurs:

Figure 17-49 Output behavior in response to braking event BRK (OSS1 = 1)

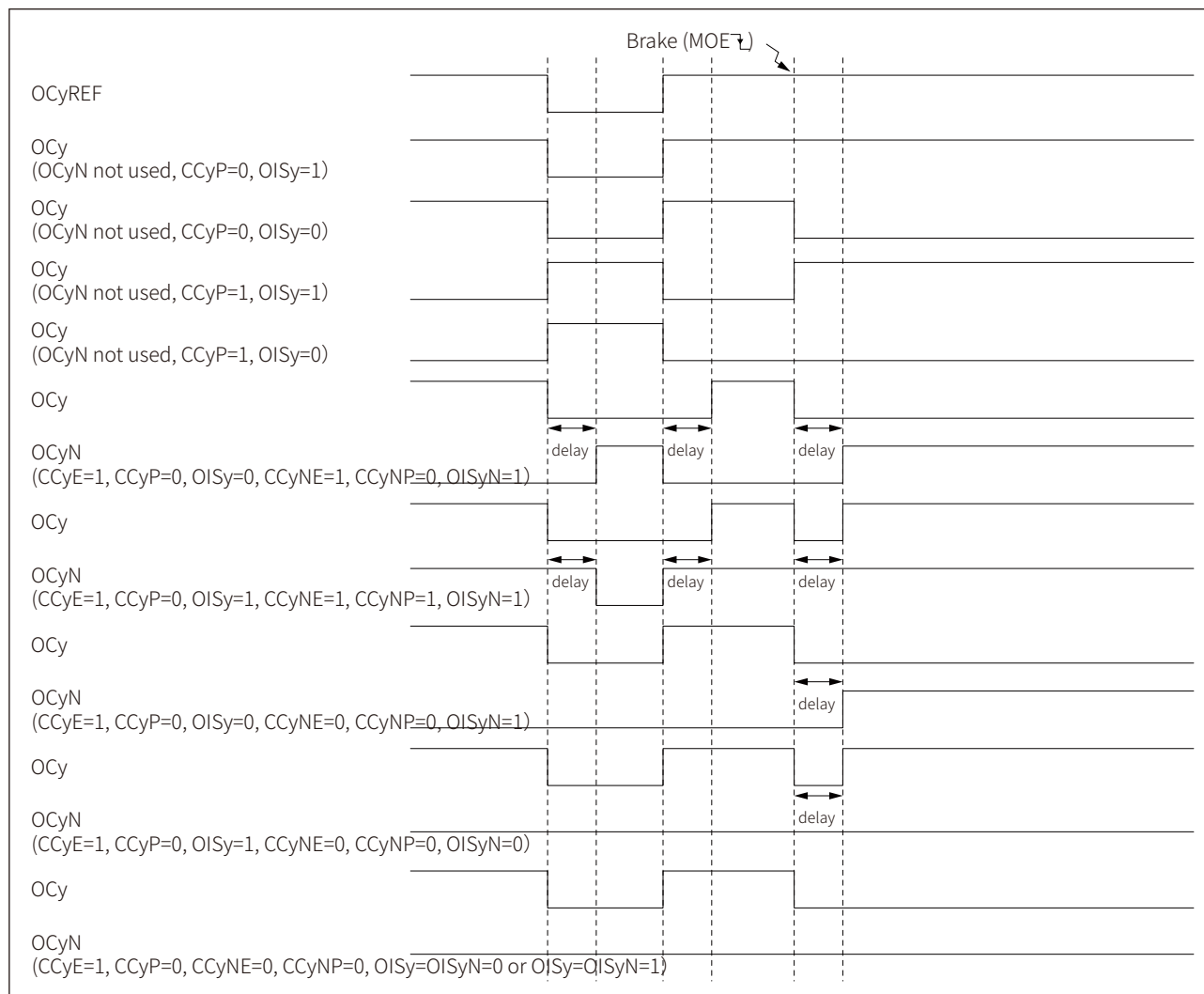
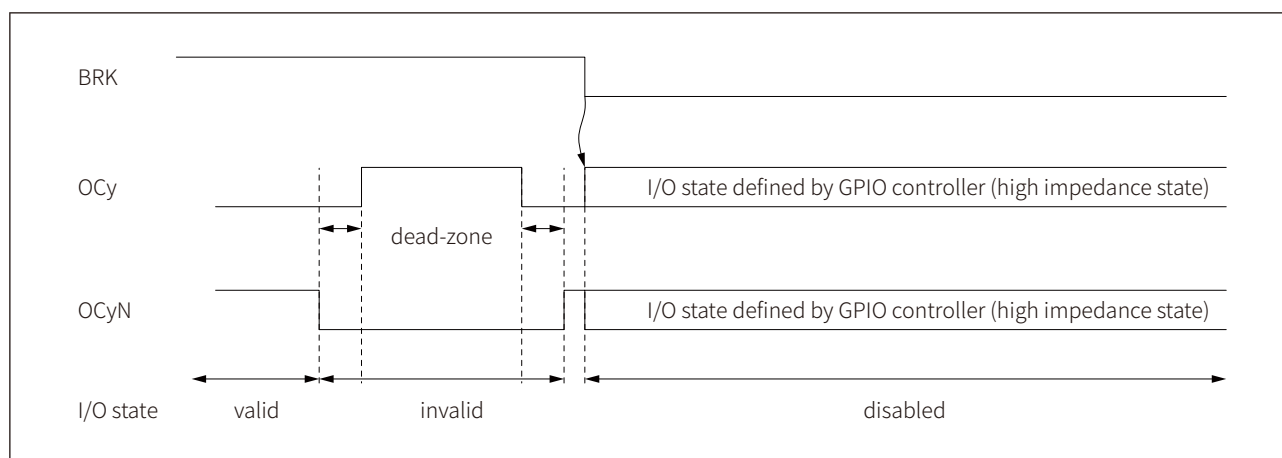


Figure 17-50 Output behavior in response to braking event BRK (OSS1 = 0)



The following figure shows an example of the output behavior of OCy and OCyN for braking events BRK and BRK2, where CCyP = CCyNP = 0:

Figure 17-51 Output behavior in response to braking BRK and BRK2 (OSS1 = 1)

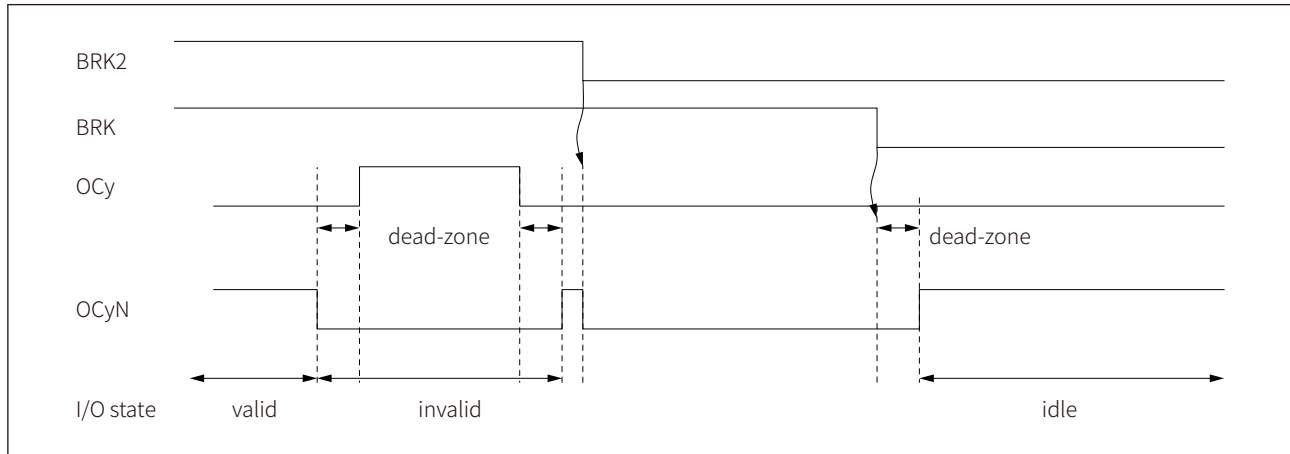


Table 17-13 Output control bits for complementary channels OCy and OCyN with brake function

| Output control bit |      |      |      |       | Output state                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |                                                                                                                  |
|--------------------|------|------|------|-------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------|
| MOE                | OSSI | OSSR | CCyE | CCyNE | OCy output state                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | OCyN output state                                                                                                |
| 1                  | x    | x    | 0    | 0     | Disable output (no longer driven by timer). The output state is forced high impedance by GPIO logic.<br>OCy = 0, OCyN = 0                                                                                                                                                                                                                                                                                                                                                            |                                                                                                                  |
|                    |      | 0    | 0    | 1     | Disable output (no longer driven by timer). The output state is forced high impedance by GPIO logic.<br>OCy = 0                                                                                                                                                                                                                                                                                                                                                                      | OCyREF + polarity<br>OCyN = OCyREF XOR CCyNP                                                                     |
|                    |      | 0    | 1    | 0     | OCyREF + polarity<br>OCy = OCyREF XOR CCyP                                                                                                                                                                                                                                                                                                                                                                                                                                           | Disable output (no longer driven by timer). The output state is forced high impedance by GPIO logic.<br>OCyN = 0 |
|                    |      | x    | 1    | 1     | OCyREF + polarity + deadband                                                                                                                                                                                                                                                                                                                                                                                                                                                         | OCyREF complement + polarity + deadband                                                                          |
|                    |      | 1    | 0    | 1     | Off state (output enable is invalid).<br>OCy = CCyP                                                                                                                                                                                                                                                                                                                                                                                                                                  | OCyREF + polarity<br>OCyN = OCyREF XOR CCyNP                                                                     |
|                    |      | 1    | 1    | 0     | OCyREF + polarity<br>OCy = OCyREF XOR CCyP                                                                                                                                                                                                                                                                                                                                                                                                                                           | Off state (output enable is invalid).<br>OCyN = CCyNP                                                            |
| 0                  | 1    | x    | x    | x     | Disable output (no longer driven by timer). The output state is forced high impedance by GPIO logic.                                                                                                                                                                                                                                                                                                                                                                                 |                                                                                                                  |
|                    |      |      | 0    | 0     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |                                                                                                                  |
|                    |      |      | 0    | 1     | Off state (output enable is invalid). Asynchronous: OCy = CCyP, OCyN = CCyNP (if BRK or BRK2 is triggered). Subsequently (valid only when BRK is triggered), if a clock is present, OCy = OISy and OCyN = OISyN after the dead time, assuming that OISy and OISyN are not both set to a valid level for OCy and OCyN (otherwise a short circuit may result when driving the switch in a half-bridge configuration).<br><b>Attention: BRK2 can only be used when OSS1 = OSSR = 1.</b> |                                                                                                                  |
|                    |      |      | 1    | 0     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |                                                                                                                  |
|                    |      |      | 1    | 1     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |                                                                                                                  |

**Attention:** If both outputs of a channel are unused (controlled by GPIO), the OISy, OISyN, CCyP, and CCyNP bits must remain a clear state.

17.3.4.9 Retriggerable single pulse mode

The retriggerable single pulse mode, in conjunction with the combined reset+trigger mode (SMS = 8), allows the counter to be reset and activated by a trigger signal (TRGI) and generates pulses of programmable length, with the pulse width determined by the ARR.

The differences compared to the single pulse mode are as follows:

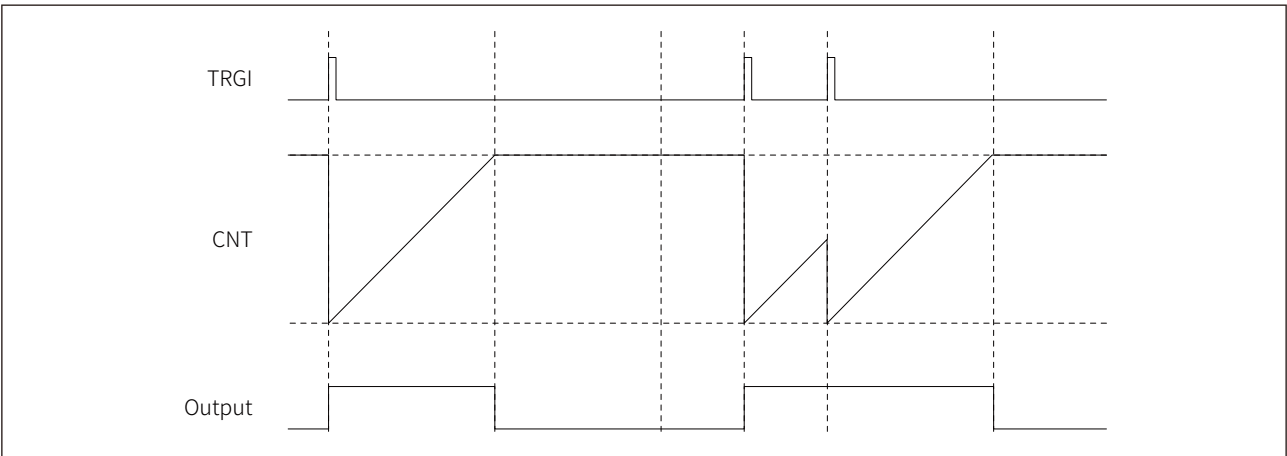
Table 17-14 Retriggerable single pulse mode and single pulse mode

| Single pulse mode                                                                                              | Retriggerable single pulse mode                                                                                        |
|----------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------|
| When a trigger occurs, a single pulse with a programmable pulse width is generated after a programmable delay. | When a trigger occurs, the pulse is generated immediately without programmable delay.                                  |
| The new trigger is invalid.                                                                                    | If the pulse generated by the previous trigger is not completed and a new trigger occurs, the pulse will be prolonged. |

In this mode, the counter can only be set to edge-aligned mode, not center-aligned mode. When configured for counting up mode, the corresponding CCRy must be set to 0. When configured for counting down mode, the corresponding CCRy must be greater than or equal to ARR.

The following figure shows an example of retriggerable single pulse mode 2:

Figure 17-52 Example of retriggerable single pulse mode 2



17.3.4.10 Count direction output

Set the OCyM bit field of the ATIM\_CCMRxCMP register to 0xB, which can output the count direction signal (a copy of the DIR bit in the ATIM\_CR1 register) on the corresponding reference signal OCyREF.

This function can be used to monitor the count direction (or direction of rotation) in encoder mode, or to indicate up/down phase signals in center-aligned PWM mode.

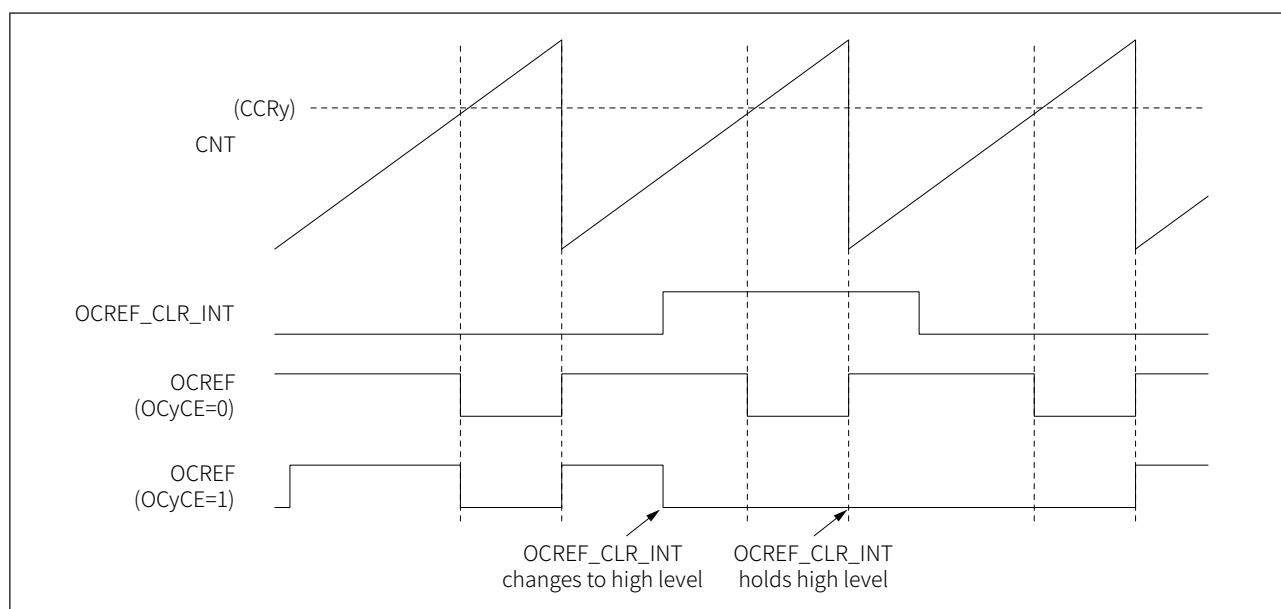
### 17.3.4.11 Reference signal clear function

Set the OCyCE bit field of the ATIM\_CCMRxCMP register to 1, which enables the OCyREF signal clearing function of the corresponding channel, i.e., when a high level is detected on the OCREf\_CLR\_INT input, the OCyREF signal can be cleared to zero and the OCyREF signal will remain low until the next update event UEV occurs. This function is only available in output compare and PWM modes and is not available in forced output mode.

The clearing source OCREf\_CLR\_INT can be selected as either OCREf\_CLR or ETRF, which is selected by the OCCS bit field of the slave mode control register ATIM\_SMCR, and there are also multiple input sources for OCREf\_CLR, see the description of the OCRSEL bit field in [17.10.31 ATIM\\_AF2 Multiplexing function option register 2](#).

The following figure shows the OCyREF signal state when the OCREf\_CLR\_INT input goes high:

Figure 17-53 OCyREF clear example





### 17.3.5 UIF bit remap

Set the UIFREMAP bit in the ATIM\_CR1 register to 1, which enables the UIF status bit remapping function, forcing the update interrupt flag ATIM\_ISR.UIF to be copied consecutively into the UIFCPY bit field of the ATIM\_CNT register. This allows automatically reading the counter value and the potential flip-flop condition issued by the UIFCPY flag. This simplifies the calculation by avoiding competing conditions when processing is shared between background tasks (counter reads) and interrupts (update interrupts).

There is no delay between UIF and UIFCPY flag enable.

When ATIM\_CR1.UIFREMAP is 0, the UIFCPY bit field is reserved and read as 0.

### 17.3.6 Timer cascade ITR

Timer cascading can be achieved by using the ITR signal, i.e., connecting the trigger output (TRGO) of the master timer to the ITR input of the slave timer, which in turn is connected to the trigger input (TRGI) of the slave timer. The ITR cascade can perform a variety of functions, such as using one timer as a prescaler for another timer, or the master timer performing a reset, enabling, or providing a clock to the slave timer's counter.

The source of the ATIM's cascade input ITR is the trigger output signals (TRGO) of the BTIM and GTIM, and the TXD/RXD signals of the UART, which can be selected by the TS bit field of the slave mode control register ATIM\_SMCR, as configured in the table below:

Table 17-15 ITR signal source

| ATIM_SMCR.TS bit field value | ITR signal name | ITR signal source |
|------------------------------|-----------------|-------------------|
| 00000                        | ITR0            | GTIM1_Trgo        |
| 00001                        | ITR1            | GTIM2_Trgo        |
| 00010                        | ITR2            | GTIM3_Trgo        |
| 00011                        | ITR3            | GTIM4_Trgo        |
| 01000                        | ITR4            | BTIM1_Trgo        |
| 01001                        | ITR5            | BTIM2_Trgo        |
| 01010                        | ITR6            | BTIM3_Trgo        |
| 01011                        | ITR7            | HALLTIM_Trgo      |
| 01100                        | ITR8            | UART1_RXD         |
| 01101                        | ITR9            | UART2_RXD         |
| 01110                        | ITR10           | UART3_RXD         |
| 01111                        | ITR11           | UART1_TXD         |
| 10000                        | ITR12           | UART2_TXD         |
| 10001                        | ITR13           | UART3_TXD         |
| 10010                        | ITR14           | HSE_FAULT         |
| 10011                        | ITR15           | LSE_FAULT         |



### 17.3.7 On-chip peripheral interconnect ETR

The source of the ATIM's ETR signal can be the external ATIM\_ETR/GTIMx\_ETR pins or other peripherals on-chip, which can be selected through the ETRSEL bit field of the multiplexing function option register ATIM\_AF1.

When the ETR signal is input from the ETR pin, the specific external input port can be referred to [Table 9-2: GPIO multiplexing function assignment table](#), and needs to be multiplexed and configured through the GPIO multiplexing function registers (GPIOx\_AFRH and GPIOx\_AFRL); when the ETR signal is input from other peripheral devices on-chip, it can realise the on-chip and off-chip peripheral interconnection.

The following table shows the sources of ETR signals:

Table 17-16 ETR signal sources

| AF1.ETRSEL bit field value | ETR signal source |
|----------------------------|-------------------|
| 0000                       | ATIM_ETR pin      |
| 0001                       | VC1_OUT           |
| 0010                       | VC2_OUT           |
| 0011                       | VC3_OUT           |
| 0100                       | VC4_OUT           |
| 0101                       | ADC1_AWD          |
| 0110                       | ADC2_AWD          |
| 0111                       | LVD_OUT           |
| 1000                       | BTIM1_ETR pin     |
| 1001                       | BTIM2_ETR pin     |
| 1010                       | BTIM3_ETR pin     |
| 1011                       | GTIM1_ETR pin     |
| 1100                       | GTIM2_ETR pin     |
| 1101                       | GTIM3_ETR pin     |
| 1110                       | GTIM4_ETR pin     |
| 1111                       | -                 |



## 17.4 ATIM interrupt

ATIM supports 16 interrupt sources. When a ATIM interrupt event occurs, the interrupt flag bit will be set by hardware, and an interrupt request will be generated if the corresponding interrupt enable control bit is set.

In the user ATIM interrupt service procedure, the relevant ATIM interrupt flag bit should be queried for corresponding processing, and the interrupt flag bit should be cleared before exiting the interrupt service procedure to avoid entering the interrupt service procedure repeatedly.

The following table shows the flag bit, interrupt enable bit, interrupt flag clear bit or clear method of each ATIM interrupt source:

Table 17-17 ATIM interrupt control

| Interrupt event                    | Interrupt flag bit | Interrupt enable bit | Interrupt flag clear                        |
|------------------------------------|--------------------|----------------------|---------------------------------------------|
| Update interrupt                   | ISR.UIF            | IER.UIE              | Write 0 to ICR.UIF.                         |
| Capture/compare 1 interrupt        | ISR.CC1IF          | IER.CC1IE            | Write 0 to ICR.CC1IF or read CCR1 register. |
| Capture/compare 2 interrupt        | ISR.CC2IF          | IER.CC2IE            | Write 0 to ICR.CC2IF or read CCR2 register. |
| Capture/compare 3 interrupt        | ISR.CC3IF          | IER.CC3IE            | Write 0 to ICR.CC3IF or read CCR3 register. |
| Capture/compare 4 interrupt        | ISR.CC4IF          | IER.CC4IE            | Write 0 to ICR.CC4IF or read CCR4 register. |
| Capture/compare 5 interrupt        | ISR.CC5IF          | IER.CC5IE            | Write 0 to ICR.CC5IF or read CCR5 register. |
| Capture/compare 6 interrupt        | ISR.CC6IF          | IER.CC6IE            | Write 0 to ICR.CC6IF or read CCR6 register. |
| COM interrupt                      | ISR.COMIF          | IER.COMIE            | Write 0 to ICR.COMIF.                       |
| Trigger interrupt                  | ISR.TIF            | IER.TIE              | Write 0 to ICR.TIF.                         |
| Brake interrupt                    | ISR.BIF            | IER.BIE              | Write 0 to ICR.BIF.                         |
| Brake 2 interrupt                  | ISR.B2IF           | IER.BIE              | Write 0 to ICR.B2IF.                        |
| System brake interrupt             | ISR.SBIF           | IER.BIE              | Write 0 to ICR.SBIF.                        |
| Encoder index interrupt            | ISR.IDXF           | IER.IDXIE            | Write 0 to ICR.IDXF.                        |
| Encoder direction change interrupt | ISR.DIRF           | IER.DIRIE            | Write 0 to ICR.DIRF.                        |
| Encoder index error interrupt      | ISR.IERRF          | IER.IERRIE           | Write 0 to ICR.IERRF.                       |
| Encoder transition error interrupt | ISR.TERRF          | IER.TERRIE           | Write 0 to ICR.TERRF.                       |



## 17.5 Trigger ADC

The timer supports a variety of internal signals to trigger the startup of the ADC, such as capturing the compare signal and triggering the output signal TRGO and TRGO2. There are several possible events for triggering the output TRGO and TRGO2, which are selected by the MMS and MMS2 bit field of the control register ATIM\_CR2. At the same time, the ADC peripherals need to configure their external trigger start registers to select the corresponding trigger source.

It should be noted that the ADC clock must be enabled before an event can be received from the main timer and that the ADC clock must not be changed in real time when the trigger signal is received from the timer.



## 17.6 Direct memory access (DMA)

ATIM supports the generation of DMA requests when capture/compare, trigger, update, and COM events occur. Using the DMA function, it is possible to enable data to be written to the timer from other positions, or read from the timer to be written to other positions, which can be applied, for example, to the auto-carrying of data after a data capture and to changing the period or pulse width of the PWM output.

The DMA/interrupt enable register ATIM\_DIER related bit field is used to enable the DMA request. The user needs to set the corresponding DMA request enable bit in the ATIM\_DIER register to 1 and set the HARDSRC bit field of the DMA trigger source control register DMA\_TRIGy to the corresponding trigger source.

With the CCDS bit field of control register ATIM\_CR2, you can select to send a DMA request when a capture compare event occurs or when an update event occurs.



## 17.7 Debugging support

ATIM supports stopping or continuing counting in debug mode, which is set up by the ATIM bit field of the debug state timer control register `SYSCTRL_DEBUG`:

- Set `SYSCTRL_DEBUG.ATIM` to 1, then the counter of ATIM is suspended in the debug state.
- Set `SYSCTRL_DEBUG.ATIM` to 0, then the counter of ATIM continues to count while in the debug state.



## 17.8 Programming examples

### 17.8.1 External clock mode 1 programming example

The following example configures the incremental counter to count on the rising edge of the ATIM\_CH1 channel, with the following steps:

- Step 1:** Set `SYSCTRL_AHBEN.GPIOx` to 1 and `SYSCTRL_APBEN1.ATIM` to 1, enable the GPIO clock corresponding to the ATIM\_CH1 pin and the ATIM configuration clock and operating clock.
- Step 2:** Configure the GPIO corresponding to the ATIM\_CH1 pin into the multiplexed input mode, for specific register configurations, please refer to [9 General-purpose input/output \(GPIO\)](#).
- Step 3:** Set `ATIM_TISEL1.TI1SEL` to 0 to select the TI1 source as the ATIM\_CH1 channel.
- Step 4:** Set `ATIM_CCMR1CAP.IC1F` to configure the input filter bandwidth.
- Step 5:** Set `ATIM_CCER.CC1NP` to 0 and `ATIM_CCER.CC1P` to 0 to select the rising edge count to be valid.
- Step 6:** Set `ATIM_SMCR.TS` to 0x5 to select the TRGI input as TI1FP1.
- Step 7:** Set `ATIM_SMCR.SMS` to 0x7 to make ATIM to work in external clock mode 1.
- Step 8:** Set `ATIM_DIER.UIE` to 1 and configure the corresponding NVIC to enable the update interrupt.
- Step 9:** Configure the counter prescaler `ATIM_PSC`.
- Step 10:** Set the expected reload value `ATIM_ARR`.
- Step 11:** Set `ATIM_CR1.CEN` to 1 to enable the counter.
- Step 12:** When the counter overflows, the `ATIM_ISR.UIF` flag bit is set to 1 to enter the interrupt service routine, and set `ATIM_ICR.UIF` to 0 to clear this interrupt flag.

### 17.8.2 External clock mode 2 programming example

The following is a programming example for external clock mode 2 + trigger mode, where the ATIM\_CH1 input triggers to start the counter when a rising edge occurs, and the counter counts on each rising edge of the ATIM\_ETR input signal, with the following steps:

- Step 1:** Set `SYSCTRL_AHBEN.GPIOx` to 1 and `SYSCTRL_APBEN1.ATIM` to 1, enable the GPIO clocks corresponding to the ATIM\_ETR and ATIM\_CH1 pins and the ATIM configuration clock and operating clock.
- Step 2:** Configure the GPIOs corresponding to the ATIM\_ETR, ATIM\_CH1 pins into the multiplexed input mode, for specific register configurations, please refer to [9 General-purpose input/output \(GPIO\)](#).
- Step 3:** Set `ATIM_AF1.ETRSEL` to 0 to select the ETR source as the ATIM\_ETR pin.
- Step 4:** Set `ATIM_SMCR.ETP` to 0 to select ETR rising edge valid.
- Step 5:** Set `ATIM_SMCR.ETPS` to 0 to disable the prescaler for ETR.
- Step 6:** Set `ATIM_SMCR.ETF` to 0 to disable ETR input filtering.
- Step 7:** Set `ATIM_SMCR.ECE` to 1 to enable external clock mode 2.
- Step 8:** Set `ATIM_TISEL1.TI1SEL` to 0 to select the TI1 source as the ATIM\_CH1 channel.
- Step 9:** Set `ATIM_CCMR1CAP.IC1F` to 0 to disable TI1 input filtering.
- Step 10:** Set `ATIM_CCER.CC1NP` to 0 and `ATIM_CCER.CC1P` to 0 to select detection of TI1 rising edge.
- Step 11:** Set `ATIM_SMCR.TS` to 0x5 to select the TRGI input as TI1FP1.
- Step 12:** Set `ATIM_SMCR.SMS` to 0x6 to make ATIM work in trigger mode.
- Step 13:** When enabled on a rising edge of the ATIM\_CH1 input, the counter counts on each rising edge of the ATIM\_ETR input signal.



### 17.8.3 Reset mode programming example

In the following example, the counter is reinitialised when a rising edge occurs on the ATIM\_CH1 input, with the following steps:

- Step 1:** Set `SYSCTRL_AHBEN.GPIOx` to 1 and `SYSCTRL_APBEN1.ATIM` to 1. Enable the GPIO clock corresponding to the ATIM\_CH1 pin and the ATIM configuration clock and operating clock.
- Step 2:** Configure the GPIO corresponding to the ATIM\_CH1 pin into the multiplexed input mode, for specific register configurations, please refer to [9 General-purpose input/output \(GPIO\)](#).
- Step 3:** Set `ATIM_TISEL1.TI1SEL` to 0 to select the TI1 source as the ATIM\_CH1 channel.
- Step 4:** Set `ATIM_CCMR1CAP.IC1F` to configure the input filter bandwidth.
- Step 5:** Set `ATIM_CCER.CC1NP` to 0 and `ATIM_CCER.CC1P` to 0 to select the detection of the rising edge.
- Step 6:** Set `ATIM_SMCR.TS` to 0x5 to select the TRGI input as TI1FP1.
- Step 7:** Set `ATIM_SMCR.SMS` to 0x4 to make ATIM to work in reset mode.
- Step 8:** Set the expected reload value `ATIM_ARR`.
- Step 9:** Set `ATIM_CR1.CEN` to 1 to enable the counter and start normal counting.
- Step 10:** When a rising edge occurs on the ATIM\_CH1 input, the counter is cleared to zero and counting starts again from 0.

### 17.8.4 Gated mode programming example

In the following example, the signal input from the ATIM\_CH1 pin is used as the gated signal to control the counter counting, with the following steps:

- Step 1:** Set `SYSCTRL_AHBEN.GPIOx` to 1 and `SYSCTRL_APBEN1.ATIM` to 1. Enable the GPIO clock corresponding to the ATIM\_CH1 pin and the ATIM configuration clock and operating clock.
- Step 2:** Configure the GPIO corresponding to the ATIM\_CH1 pin into the multiplexed input mode, for specific register configurations, please refer to [9 General-purpose input/output \(GPIO\)](#).
- Step 3:** Set `ATIM_TISEL1.TI1SEL` to 0 to select the TI1 source as the ATIM\_CH1 channel.
- Step 4:** Set `ATIM_CCMR1CAP.IC1F` to configure the input filter bandwidth.
- Step 5:** Set `ATIM_CCER.CC1NP` to 0 and `ATIM_CCER.CC1P` to 0 to select detect high level.
- Step 6:** Set `ATIM_SMCR.TS` to 0x5 to select the TRGI input as TI1FP1.
- Step 7:** Set `ATIM_SMCR.SMS` to 0x5 to make ATIM to work in gated mode.
- Step 8:** Set the expected reload value `ATIM_ARR`.
- Step 9:** Set `ATIM_CR1.CEN` to 1 to enable the counter.
- Step 10:** When ATIM\_CH1 input is high, the counter starts counting; when ATIM\_CH1 input is low, the counter stops counting.





### 17.8.5 Trigger mode programming example

In the following example, the counter is triggered to start when a rising edge occurs on the ATIM\_CH1 input, with the following steps:

- Step 1:** Set `SYSCTRL_AHBEN.GPIOx` to 1 and `SYSCTRL_APBEN1.ATIM` to 1. Enable the GPIO clock corresponding to the ATIM\_CH1 pin and the ATIM configuration clock and operating clock.
- Step 2:** Configure the GPIO corresponding to the ATIM\_CH1 pin into the multiplexed input mode, for specific register configurations, please refer to [9 General-purpose input/output \(GPIO\)](#).
- Step 3:** Set `ATIM_TISEL1.TI1SEL` to 0 to select the TI1 source as the ATIM\_CH1 channel.
- Step 4:** Set `ATIM_CCMR1CAP.IC1F` to configure the input filter bandwidth.
- Step 5:** Set `ATIM_CCER.CC1NP` to 0 and `ATIM_CCER.CC1P` to 0 to select the detection of the rising edge.
- Step 6:** Set `ATIM_SMCR.TS` to 0x5 to select the TRGI input as TI1FP1.
- Step 7:** Set `ATIM_SMCR.SMS` to 0x6 to make ATIM to work in trigger mode.
- Step 8:** Set the expected reload value `ATIM_ARR`.
- Step 9:** When a rising edge occurs on the ATIM\_CH1 input, the counter starts counting and the TIF flag is set to 1.

### 17.8.6 Quadrature encoder mode programming example

In the following example, configure ATIM to work in quadrature encoder mode -x4 mode, with the following steps:

- Step 1:** Set `SYSCTRL_AHBEN.GPIOx` to 1, `SYSCTRL_APBEN1.ATIM` to 1, and enable the GPIO clocks corresponding to the ATIM\_CH1 and ATIM\_CH2 pins as well as the ATIM configuration clock and operating clock.
- Step 2:** Configure the GPIO corresponding to the ATIM\_CH1 and ATIM\_CH2 pins into the multiplexed input mode, for specific register configurations, please refer to [9 General-purpose input/output \(GPIO\)](#).
- Step 3:** Set `ATIM_TISEL1.TI1SEL` to 0 to select the TI1 source as ATIM\_CH1 channel.
- Step 4:** Set `ATIM_TISEL1.TI2SEL` to 0 to select the TI2 source as the ATIM\_CH2 channel.
- Step 5:** Set `ATIM_CCMR1CAP.IC1F` and `ATIM_CCMR1CAP.IC2F` to configure the input filter bandwidth.
- Step 6:** Set `ATIM_CCER.CC1NP` and `ATIM_CCER.CC1P` to 0, CH1 input is not inverted.
- Step 7:** Set `ATIM_CCER.CC2NP` and `ATIM_CCER.CC2P` to 0, CH2 input is not inverted.
- Step 8:** Set `ATIM_SMCR.SMS` to 0x3 to make ATIM work in quadrature encoder mode-x4 mode.
- Step 9:** Set the appropriate reload value `ATIM_ARR`.
- Step 10:** Set `ATIM_CR1.CEN` to 1 to enable the counter.



## 17.8.7 Input capture programming example

### 17.8.7.1 Basic input capture mode

In the following example, execute an input capture when a rising edge occurs on the ATIM\_CH1 input, with the following steps:

- Step 1:** Set `SYSCTRL_AHBEN.GPIOx` to 1, `SYSCTRL_APBEN1.ATIM` to 1, and enable the GPIO clock corresponding to the ATIM\_CH1 pin and the ATIM configuration clock and operating clock.
- Step 2:** Configure the GPIO corresponding to the ATIM\_CH1 pin into the multiplexed input mode, for specific register configurations, please refer to [9 General-purpose input/output \(GPIO\)](#).
- Step 3:** Set `ATIM_TISEL1.TI1SEL` to 0 to select the TI1 source as the ATIM\_CH1 channel.
- Step 4:** Set `ATIM_CCMR1CAP.IC1F` to configure the TI1 input filter bandwidth.
- Step 5:** Set `ATIM_CCER.CC1NP` to 0 and `ATIM_CCER.CC1P` to 0 to select detection of IC1 rising edge.
- Step 6:** Set `ATIM_CCMR1CAP.CC1S` to 1, the CC1 channel is configured as an input and IC1 is mapped to TI1.
- Step 7:** Set `ATIM_CCMR1CAP.IC1PSC` to 0 to disable the IC1 prescaler.
- Step 8:** Set `ATIM_CCER.CC1E` to 1 to enable input capture 1 mode.
- Step 9:** Set `ATIM_DIER.CC1IE` to 1 and configure the corresponding NVIC to enable the input capture 1 interrupt.
- Step 10:** Configure the counter prescaler `ATIM_PSC`.
- Step 11:** Set the expected reload value `ATIM_ARR`.
- Step 12:** Set `ATIM_CR1.CEN` to 1 to enable the counter and start normal counting.
- Step 13:** When a capture occurs, the value of the current counter CNT is latched into the capture/compare register CCR1, completing a capture, while the `ATIM_ISR.CC1IF` flag bit is set to 1 to enter the interrupt service procedure, and this interrupt flag is cleared by setting `ATIM_ICR.CC1IF` to 0.



### 17.8.7.2 PWM input mode

The following example measures the period and duty cycle of the PWM signal input from the ATIM\_CH1 pin, with the following steps:

- Step 1:** Set `SYSCTRL_AHBEN.GPIOx` to 1, `SYSCTRL_APBEN1.ATIM` to 1, and turn on the GPIO clock corresponding to the ATIM\_CH1 pin and the ATIM configuration clock and operating clock.
- Step 2:** Configure the GPIO corresponding to the ATIM\_CH1 pin into the multiplexed input mode, for specific register configurations, please refer to [9 General-purpose input/output \(GPIO\)](#).
- Step 3:** Set `ATIM_TISEL1.TI1SEL` to 0 to select the TI1 source as the ATIM\_CH1 channel.
- Step 4:** Set `ATIM_CCMR1CAP.IC1F` to configure the TI1 input filter bandwidth.
- Step 5:** Set `ATIM_CCER.CC1NP` to 0 and `ATIM_CCER.CC1P` to 0 to select detection of IC1 rising edge.
- Step 6:** Set `ATIM_CCMR1CAP.CC1S` to 1, the CC1 channel is configured as an input and IC1 is mapped to TI1.
- Step 7:** Set `ATIM_CCMR1CAP.IC1PSC` to 0 to disable the IC1 prescaler.
- Step 8:** Set `ATIM_CCER.CC1E` to 1 to enable input capture 1 mode.
- Step 9:** Set `ATIM_DIER.CC1IE` to 1 and configure the corresponding NVIC to enable input capture 1 interrupt.
- Step 10:** Set `ATIM_CCER.CC2NP` to 0 and `ATIM_CCER.CC2P` to 1 to select detect IC2 falling edge.
- Step 11:** Set `ATIM_CCMR1CAP.CC2S` to 2, CC2 channel configured as input, IC2 mapped to TI1.
- Step 12:** Set `ATIM_CCMR1CAP.IC2PSC` to 0 to disable the IC2 prescaler.
- Step 13:** Set `ATIM_CCER.CC2E` to 1 to enable input capture 2 mode.
- Step 14:** Set `ATIM_SMCR.TS` to 0x5 to select the TRGI input as TI1FP1.
- Step 15:** Set `ATIM_SMCR.SMS` to 0x4 to enable the ATIM to work in reset mode.
- Step 16:** Configure the counter prescaler `ATIM_PSC`.
- Step 17:** Set the expected reload value `ATIM_ARR`.
- Step 18:** Set `ATIM_CR1.CEN` to 1 to enable the counter and start normal counting.



### 17.8.8 Output compare programming example

In the following example, channel CH1 outputs the set waveform externally, with the following steps:

- Step 1:** Set `SYSCTRL_AHBEN.GPIOx` to 1, `SYSCTRL_APBEN1.ATIM` to 1, enable the GPIO clock corresponding to the `ATIM_CH1` pin and the ATIM configuration clock and working clock.
- Step 2:** Configure the GPIO corresponding to the `ATIM_CH1` pin into the multiplexed output mode, for specific register configurations, please refer to [9 General-purpose input/output \(GPIO\)](#).
- Step 3:** Set `ATIM_CCER.CC1P` to 0 to select the `OC1` output valid polarity.
- Step 4:** Set `ATIM_CCMR1CMP.CC1S` to 0 to configure the `CC1` channel as an output.
- Step 5:** Set `ATIM_CCMR1CMP.OC1M` to select output compare 1 mode.
- Step 6:** Set `ATIM_CCER.CC1E` to 1 to output the `OC1` signal on the corresponding output pin.
- Step 7:** Set `ATIM_BDTR.MOE` to 1 to enable main output.
- Step 8:** Configure the counter prescaler `ATIM_PSC`.
- Step 9:** Set the expected reload value `ATIM_ARR`.
- Step 10:** Set the expected comparison value `ATIM_CCR1`.
- Step 11:** Set `ATIM_DIER.CC1IE` to 1 and configure the corresponding NVIC to enable the compare 1 interrupt.
- Step 12:** Set `ATIM_CR1.CEN` to 1 to enable the counter.
- Step 13:** When a compare match occurs, the `ATIM_ISR.CC1IF` flag bit is set to 1 to enter the interrupt service procedure, and this interrupt flag is cleared by setting `ATIM_ICR.CC1IF` to 0.



## 17.9 List of registers

ATIM base address: ATIM\_BASE = 0x4000 1400

Table 17-18 List of ATIM registers

| Register name | Register address | Register description                    |
|---------------|------------------|-----------------------------------------|
| ATIM_CR1      | ATIM_BASE + 0x00 | Control register 1                      |
| ATIM_CR2      | ATIM_BASE + 0x04 | Control register 2                      |
| ATIM_SMCR     | ATIM_BASE + 0x08 | Slave mode control register             |
| ATIM_DIER     | ATIM_BASE + 0x0C | DMA/Interrupt enable register           |
| ATIM_ISR      | ATIM_BASE + 0x10 | Interrupt flag register                 |
| ATIM_ICR      | ATIM_BASE + 0x70 | Interrupt flag clear register           |
| ATIM_EGR      | ATIM_BASE + 0x14 | Event generation register               |
| ATIM_CCMR1CAP | ATIM_BASE + 0x18 | Capture mode register 1                 |
| ATIM_CCMR1CMP | ATIM_BASE + 0x18 | Compare mode register 1                 |
| ATIM_CCMR2CAP | ATIM_BASE + 0x1C | Capture mode register 2                 |
| ATIM_CCMR2CMP | ATIM_BASE + 0x1C | Compare mode register 2                 |
| ATIM_CCMR3CAP | ATIM_BASE + 0x50 | Capture mode register 3                 |
| ATIM_CCMR3CMP | ATIM_BASE + 0x50 | Compare mode register 3                 |
| ATIM_CCER     | ATIM_BASE + 0x20 | Capture/compare enable register         |
| ATIM_CNT      | ATIM_BASE + 0x24 | Count register                          |
| ATIM_PSC      | ATIM_BASE + 0x28 | Prescaler register                      |
| ATIM_ARR      | ATIM_BASE + 0x2C | Auto reload register                    |
| ATIM_RCR      | ATIM_BASE + 0x30 | Repeat count register                   |
| ATIM_CCR1     | ATIM_BASE + 0x34 | Capture/compare register 1              |
| ATIM_CCR2     | ATIM_BASE + 0x38 | Capture/compare register 2              |
| ATIM_CCR3     | ATIM_BASE + 0x3C | Capture/compare register 3              |
| ATIM_CCR4     | ATIM_BASE + 0x40 | Capture/compare register 4              |
| ATIM_CCR5     | ATIM_BASE + 0x48 | Capture/compare register 5              |
| ATIM_CCR6     | ATIM_BASE + 0x4C | Capture/compare register 6              |
| ATIM_BDTR     | ATIM_BASE + 0x44 | Brake and deadband register             |
| ATIM_DTR2     | ATIM_BASE + 0x54 | Dead time register 2                    |
| ATIM_ECR      | ATIM_BASE + 0x58 | Encoder control register                |
| ATIM_TISEL1   | ATIM_BASE + 0x5C | TI input selection register 1           |
| ATIM_TISEL2   | ATIM_BASE + 0x6C | TI input selection register 2           |
| ATIM_AF1      | ATIM_BASE + 0x60 | Multiplexing function option register 1 |

Table 17-18 List of ATIM registers (continued)

| Register name | Register address | Register description                    |
|---------------|------------------|-----------------------------------------|
| ATIM_AF2      | ATIM_BASE + 0x64 | Multiplexing function option register 2 |



## 17.10 Register description

See section 1 [Documentation conventions](#) for abbreviations used in register descriptions.

### 17.10.1 ATIM\_CR1 Control register 1

Address offset: 0x00      Reset value: 0x0000 0000

| Bit field | Name     | Permission | Function description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|-----------|----------|------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:12     | RFU      | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 11        | UIFREMAP | RW         | UIF status bit remap<br>0: Remapping is disabled, ATIM_CNT[31] remains 0.<br>1: Remapping is enabled, ATIM_CNT[31] is equivalent to ATIM_ISR.UIF.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| 10        | RFU      | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 9:8       | CKD      | RW         | Configure the division ratio between PCLK clock and deadband and sampling clock ( $f_{DTS}$ ) used by deadband generator and digital filter (ETR, TIy):<br>00: $f_{DTS} = f_{PCLK}/1$<br>01: $f_{DTS} = f_{PCLK}/2$<br>10: $f_{DTS} = f_{PCLK}/4$<br><i>Note: The filter clock is used for ETR filtering and input capture filtering.</i>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 7         | ARPE     | RW         | Auto reload preload enable<br>0: Disable ARR register buffering.<br>1: Enable ARR register buffering.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 6:5       | CMS      | RW         | Counter alignment mode configuration<br>00: Edge-aligned mode: The counter is controlled by the CR1.DIR bit to count up or down.<br>01: Center-aligned mode 1: Counter alternately counts up and down. The output compare interrupt flag is set to 1 only when the counter counts down.<br>10: Center-aligned mode 2: Counter alternately counts up and down. The output compare interrupt flag is set to 1 only when the counter counts up.<br>11: Center-aligned mode 3: Counter alternately counts up and down. The output compare interrupt flag is set to 1 when either the counter counts up or down.<br><i>Attention: Do not switch from edge-aligned mode to center-aligned mode as long as the counter is enabled (CEN = 1).</i><br><i>Note: The initial count direction of the center-aligned mode is determined by the DIR bit field.</i> |
| 4         | DIR      | RW/RO      | Count direction<br>0: Counter count up.<br>1: Counter count down.<br><i>Note: When the timer is configured in center aligned mode or encoder mode, this bit is read-only.</i>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |



| Bit field | Name | Permission | Function description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|-----------|------|------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 3         | OPM  | RW         | Single pulse mode enable control<br>0: Disable single pulse mode.<br>1: Enable single pulse mode, counter automatically stops when update event is received during running (set CEN to 0).                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 2         | URS  | RW         | Update request source<br>This bit is set to 1 and cleared to 0 by software to select the UEV event source.<br>0: When UEV is enabled, all of the following events generate an update interrupt or DMA request. Including: <ul style="list-style-type: none"> <li>Counter overflow/underflow.</li> <li>Set UG bit to 1.</li> <li>Update events generated by the slave mode controller.</li> </ul> 1: When UEV is enabled, only counter overflow/underflow generates an update interrupt or DMA request.                                                                                                                                                                                                                         |
| 1         | UDIS | RW         | Update disabled<br>This bit is set to 1 and cleared to 0 by software to enable/disable UEV event generation.<br>0: UEV enabled. Update events (UEV) can be generated by one of the following events, which then updates the value of the shadow register: <ul style="list-style-type: none"> <li>Counter overflow/underflow.</li> <li>Set UG bit to 1.</li> <li>Update events generated by the slave mode controller.</li> </ul> 1: UEV disabled. No update event is generated and the values of the respective shadow registers (ARR, PSC, and CCRy) remain unchanged. However, if the UG bit field is set to 1, or a hardware reset is received from the slave mode controller, the counter and prescaler are reinitialised. |
| 0         | CEN  | RW         | Counter enable<br>0: Counter disabled<br>1: Counter enabled<br><i>Attention: External clock, gated mode and encoder mode can only be used if the CEN bit field has been previously set to 1 by software. Trigger mode, on the other hand, automatically sets CEN position 1 via hardware.</i><br><i>Note: In single pulse mode, the CEN bit is automatically cleared to 0 when an update event occurs.</i>                                                                                                                                                                                                                                                                                                                     |





### 17.10.2 ATIM\_CR2 Control register 2

Address offset: 0x04

Reset value: 0x0000 0000

| Bit field | Name  | Permission | Function description                                                                                                                                                                                                                                                                              |
|-----------|-------|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:27     | RFU   | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                     |
| 26:25     | MMSH  | RW         | Master mode selection<br>It is the highest 2 bits of the MMS and must be used with the MMS bit field.                                                                                                                                                                                             |
| 24:20     | MMS2  | RW         | Master mode selection 2<br>These bits select information to be sent to the ADC for synchronization (TRGO2). The combination of these bits is described in MMS bit field.                                                                                                                          |
| 19        | OIS6N | RW         | Output idle state 6 (OC6N output), see OIS1N description.                                                                                                                                                                                                                                         |
| 18        | OIS6  | RW         | Output idle state 6 (OC6 output), see OIS1 description.                                                                                                                                                                                                                                           |
| 17        | OIS5N | RW         | Output idle state 5 (OC5N output), see OIS1N description.                                                                                                                                                                                                                                         |
| 16        | OIS5  | RW         | Output idle state 5 (OC5 output), see OIS1 description.                                                                                                                                                                                                                                           |
| 15        | OIS4N | RW         | Output idle state 4 (OC4N output), see OIS1N description.                                                                                                                                                                                                                                         |
| 14        | OIS4  | RW         | Output idle state 4 (OC4 output), see OIS1 description.                                                                                                                                                                                                                                           |
| 13        | OIS3N | RW         | Output idle state 3 (OC3N output), see OIS1N description.                                                                                                                                                                                                                                         |
| 12        | OIS3  | RW         | Output idle state 3 (OC3 output), see OIS1 description.                                                                                                                                                                                                                                           |
| 11        | OIS2N | RW         | Output idle state 2 (OC2N output), see OIS1N description.                                                                                                                                                                                                                                         |
| 10        | OIS2  | RW         | Output idle state 2 (OC2 output), see OIS1 description.                                                                                                                                                                                                                                           |
| 9         | OIS1N | RW         | Output idle state 1 (OC1N output)<br>0: When MOE = 0, OC1N = 0 after dead time.<br>1: When MOE = 0, OC1N = 1 after dead time.<br><i>Attention: This bit cannot be modified as long as LOCK (LOCK bit in ATIM_BDTR register) level 1, 2 or 3 is programmed.</i>                                    |
| 8         | OIS1  | RW         | Output idle state 1 (OC1 output)<br>0: When MOE = 0, (after dead time if OC1N is valid) OC1 = 0.<br>1: When MOE = 0, (after dead time if OC1N is valid) OC1 = 1.<br><i>Attention: This bit cannot be modified as long as LOCK (LOCK bit in ATIM_BDTR register) level 1, 2 or 3 is programmed.</i> |
| 7         | TI1S  | RW         | TI1 input selection<br>0: ATIM_CH1 pin is connected to the TI1 input.<br>1: XOR combination of ATIM_CH1, CH2 and CH3 pins are connected to TI1 input.                                                                                                                                             |
| 6:4       | MMS   | RW         | Master mode selection<br>These bits select the information that will be sent to the slave timer for synchronisation in master mode (TRGO). Used in conjunction with MMSH, these bits are combined as follows:                                                                                     |



| Bit field          | Name | Permission | Function description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|--------------------|------|------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 6:4<br>(continued) | MMS  | RW         | <p>00000: Reset - The UG bit in the ATIM_EGR register is used as the trigger output (TRGO). If the reset is generated by a trigger input ( slave mode controller configured for reset mode), the signal on TRGO is delayed compared to the actual reset.</p> <p>00001: Enable - The counter enable signal CNT_EN is used as a trigger output (TRGO). This trigger output can be used to start multiple timers at the same time, or to control the enabling of slave timers over a period of time. The counter enable signal consists of a logical sum operation combining the CEN control bit with the trigger input in gated mode. When the counter enable signal is controlled by a trigger input, there is a delay on the TRGO, except when master/slave mode is selected ( please refer to the description of the MSM bit in <a href="#">17.10.3 ATIM_SMCR Slave mode control register</a>).</p> <p>00010: Update - Select the update event as the trigger output (TRGO). For example, the master timer can be used as a prescaler for the slave timer.</p> <p>00011: Encoder count clock output.</p> <p>00100: CH1 output compare or input capture pulse.</p> <p>00101: CH2 output compare or input capture pulse.</p> <p>00110: CH3 output compare or input capture pulse.</p> <p>00111: CH4 output compare or input capture pulse.</p> <p>01000: CH5 output compare or input capture pulse.</p> <p>01001: CH6 output compare or input capture pulse.</p> <p>01010: OC1REF signal.</p> <p>01011: OC2REF signal.</p> <p>01100: OC3REF signal.</p> <p>01101: OC4REF signal.</p> <p>01110: OC5REF signal.</p> <p>01111: OC6REF signal.</p> <p>10000: OC1REFC signal.</p> <p>10001: OC2REFC signal.</p> <p>10010: OC3REFC signal.</p> <p>10011: OC4REFC signal.</p> <p>10100: OC5REFC signal.</p> <p>10101: OC6REFC signal.</p> <p>10110: OC3REFC rising edge or falling edge.</p> <p>10111: OC4REFC rising edge or falling edge.</p> <p>11000: OC5REFC rising edge or falling edge.</p> <p>11001: OC6REFC rising edge or falling edge.</p> <p>11010: OC4REFC rising edge or OC5REFC rising edge.</p> <p>11011: OC4REFC rising edge or OC5REFC falling edge.</p> <p>11100: OC4REFC rising edge or OC6REFC rising edge.</p> <p>11101: OC4REFC rising edge or OC6REFC falling edge.</p> <p>11110: OC5REFC rising edge or OC6REFC rising edge.</p> <p>11111: OC5REFC rising edge or OC6REFC falling edge.</p> <p><b>Attention: The clock of the slave timer or ADC must be enabled before an event can be received from the master timer; and the clock of the slave timer or ADC must not be changed in real time when a trigger signal is received from the master timer.</b></p> |
| 3                  | CCDS | RW         | <p>Capture/compare DMA selection</p> <p>0: Send CCx DMA request when CCx event occurs.</p> <p>1: Send CCx DMA request when update event occurs.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |



| Bit field | Name | Permission | Function description                                                                                                                                                                                                                                                                                                                                                                                                                            |
|-----------|------|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 2         | CCUS | RW         | <p>Capture/compare control update selection</p> <p>0: If the capture/compare control bits are preloaded (CCPC = 1), these bits can be updated only by setting the COMG bit field to 1.</p> <p>1: If the capture/compare control bits are preloaded (CCPC = 1), these bits can be updated by setting the COMG bit field to 1 or the rising edge of TRGI.</p> <p><i>Note: This bit is only valid for channels with complementary outputs.</i></p> |
| 1         | RFU  | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                                                                                                                                   |
| 0         | CCPC | RW         | <p>Capture/compare preload control</p> <p>0: CCyE, CCyNE and OCyM bits are not preloaded.</p> <p>1: The CCyE, CCyNE, and OCyM bits are preloaded. After these bits are written, they are updated only when a commutation event (COM) occurs (COMG bit field is set to 1 or a rising edge is detected on TRGI, depending on the CCUS bit).</p> <p><i>Note: This bit is only valid for channels with complementary outputs.</i></p>               |



### 17.10.3 ATIM\_SMCR Slave mode control register

Address offset: 0x08

Reset value: 0x0000 0000

| Bit field | Name  | Permission | Function description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|-----------|-------|------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:26     | RFU   | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 25        | SMSPS | RW         | Preload source<br>This bit selects the trigger source for the transfer of the SMS[3:0] bit fields from the preload register to the valid registers.<br>0: Transmit is triggered by a timer update event.<br>1: Transmit is triggered by an index event.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 24        | SMSPE | RW         | Preload enable<br>This bit selects whether the SMS[3:0] bits are preloaded or not.<br>0: Preload disabled.<br>1: Preload enabled.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| 23:22     | RFU   | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 21:20     | TSH   | RW         | Trigger selection<br>It is the highest bit of the TS, and needs to be used in conjunction with the TS.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| 19:17     | RFU   | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 16        | SMSH  | RW         | Slave mode selection<br>It is the highest bit of SMS, and needs to be used in conjunction with SMS.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 15        | ETP   | RW         | External trigger polarity<br>This bit selects whether ETR or the inversion of ETR will be used for the trigger operation.<br>0: ETR not inverted, high or rising edge is valid.<br>1: ETR inverted, low level or falling edge is valid.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 14        | ECE   | RW         | External clock enabled<br>This bit can enable external clock mode 2.<br>0: Disable external clock mode 2<br>1: Enable external clock mode 2, the counter clock is provided by any valid edge of the ETRF signal.<br><i>Note: Set the ECE bit field to 1 which has the same effect as selecting external clock mode 1 and connecting TRGI to ETRF (SMS = 111 and TS = 00111).</i><br><i>Note: External clock mode 2 can be used in conjunction with the following slave modes: reset mode, gated mode and trigger mode. However, in these cases TRGI must not be connected to ETRF (TS bit must not be 00111).</i><br><i>Note: If external clock mode 1 and external clock mode 2 are enabled at the same time, the external clock input is ETRF.</i> |



| Bit field | Name | Permission | Function description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|-----------|------|------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 13:12     | ETPS | RW         | <p>External trigger prescaler</p> <p>The external trigger signal ETRP frequency must not exceed 1/4 of the PCLK frequency. The ETRP frequency can be reduced by enabling the prescaler, and this method is useful when a fast external clock is input.</p> <p>00: Prescaler disabled.<br/> 01: 2-division ETRP frequency.<br/> 10: 4-division ETRP frequency.<br/> 11: 8-division ETRP frequency.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 11:8      | ETF  | RW         | <p>External trigger filter</p> <p>This bit field defines the sampling frequency of the ETRP signal and the digital filter bandwidth applicable to the ETRP. The digital filter consists of an event counter, where only every N consecutive events are considered as a valid output edge:</p> <p>0000: No filter, sampled at frequency <math>f_{DTS}</math><br/> 0001: <math>f_{SAMPLING} = f_{DTS}/1</math>, <math>N = 2</math><br/> 0010: <math>f_{SAMPLING} = f_{DTS}/1</math>, <math>N = 4</math><br/> 0011: <math>f_{SAMPLING} = f_{DTS}/1</math>, <math>N = 8</math><br/> 0100: <math>f_{SAMPLING} = f_{DTS}/2</math>, <math>N = 6</math><br/> 0101: <math>f_{SAMPLING} = f_{DTS}/2</math>, <math>N = 8</math><br/> 0110: <math>f_{SAMPLING} = f_{DTS}/4</math>, <math>N = 6</math><br/> 0111: <math>f_{SAMPLING} = f_{DTS}/4</math>, <math>N = 8</math><br/> 1000: <math>f_{SAMPLING} = f_{DTS}/8</math>, <math>N = 6</math><br/> 1001: <math>f_{SAMPLING} = f_{DTS}/8</math>, <math>N = 8</math><br/> 1010: <math>f_{SAMPLING} = f_{DTS}/16</math>, <math>N = 5</math><br/> 1011: <math>f_{SAMPLING} = f_{DTS}/16</math>, <math>N = 6</math><br/> 1100: <math>f_{SAMPLING} = f_{DTS}/16</math>, <math>N = 8</math><br/> 1101: <math>f_{SAMPLING} = f_{DTS}/32</math>, <math>N = 5</math><br/> 1110: <math>f_{SAMPLING} = f_{DTS}/32</math>, <math>N = 6</math><br/> 1111: <math>f_{SAMPLING} = f_{DTS}/32</math>, <math>N = 8</math></p> |
| 7         | MSM  | RW         | <p>Master/slave mode</p> <p>0: No action is performed.<br/> 1: The action of the trigger input event (TRGI) of the current timer is postponed to allow perfect synchronisation (via TRGO) of the current timer with its slave timers. This setting applies when multiple timers are synchronised by a single external event.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |



| Bit field | Name | Permission | Function description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|-----------|------|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 6:4       | TS   | RW         | <p>Trigger input TRGI selection</p> <p>This bit field selects the trigger input that will be used to synchronise the counter.</p> <p>00000: Internal trigger 0 (ITR0).<br/> 00001: Internal trigger 1 (ITR1).<br/> 00010: Internal trigger 2 (ITR2).<br/> 00011: Internal trigger 3 (ITR3).<br/> 00100: TI1 edge detector (TI1F_ED).<br/> 00101: Filtered timer input 1 (TI1FP1).<br/> 00110: Filtered timer input 2 (TI2FP2).<br/> 00111: External trigger input (ETRF).<br/> 01000: Internal trigger 4 (ITR4).<br/> 01001: Internal trigger 5 (ITR5).<br/> 01010: Internal trigger 6 (ITR6).<br/> 01011: Internal trigger 7 (ITR7).<br/> 01100: Internal trigger 8 (ITR8).<br/> 01101: Internal trigger 9 (ITR9).<br/> 01110: Internal trigger 10 (ITR10).<br/> 01111: Internal trigger 11 (ITR11).<br/> 10000: Internal trigger 12 (ITR12).<br/> 10001: Internal trigger13 (ITR13).<br/> 10010: Internal trigger14 (ITR14).<br/> 10011: Internal trigger15 (ITR15).<br/> Other values: Reserved.</p> <p><i>Note: These bits can only be changed if they are not used (e.g. SMS = 0000) to avoid incorrect edge detection during transition.</i></p> <p><i>Note: The other bit fields are in bits 21 and 20 of the same register and need to be used in conjunction with the TS.</i></p> <p><i>Note: For specific ITRx sources, please refer to <a href="#">Table 17-15: ITR signal source</a>.</i></p> |
| 3         | OCCS | RW         | <p>OCREF cleared selection</p> <p>This bit is used to select the OCREF clear source.</p> <p>0: OCREF_CLR_INT connected to OCREF_CLR input.<br/> 1: OCREF_CLR_INT connected to ETRF.</p> <p><i>Note: OCREF_CLR input source please refer to <a href="#">17.10.31 ATIM_AF2 Multiplexing function option register 2</a> for the description of the OCRSEL bit field.</i></p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 2:0       | SMS  | RW         | <p>Slave mode selection</p> <p>When an external signal is selected, the valid edge of the trigger signal (TRGI) is related to the polarity selected on the external input (see the input related control registers).</p> <p>0000: Slave mode disabled - If CEN = '1', the prescaler clock is supplied directly from the internal clock.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |



| Bit field          | Name | Permission | Function description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|--------------------|------|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 2:0<br>(continued) | SMS  | RW         | <p>0001: Quadrature encoder mode - x2 mode , the counter counts up/down on the TI1FP1 edge according to the TI2FP2 level.</p> <p>0010: Quadrature encoder mode - x2 mode, the counter counts up/down according to the TI1FP1 level at the TI2FP2 edge.</p> <p>0011: Quadrature encoder mode - x4 mode, the counter counts on the TI1FP1 and TI2FP2 edges, the direction of the count depends on the level of the other input.</p> <p>0100: Reset mode - The counter is reinitialised and a register update event is generated on the occurrence of a rising edge of the selected trigger input (TRGI).</p> <p>0101: Gated mode - Enables the counter clock when the trigger input (TRGI) is high. The counter stops counting (but not resetting) as soon as the trigger input goes low. Counter start and stop are controlled.</p> <p>0110: Trigger mode - Starts the counter (but does not reset) on a rising edge of the trigger signal (TRGI). Only the start of the counter is controlled.</p> <p>0111: External clock mode 1 - Counter clock is provided by the rising edge of the selected trigger signal (TRGI).</p> <p>1000: Combined reset + trigger mode - Reinitialises the counter on the rising edge of the selected trigger input (TRGI), generates a register update event and starts the counter.</p> <p>1001: Combined gated + reset mode - The counter is enabled and starts counting when the trigger input (TRGI) goes high, the counter stops counting and is reset when the trigger input goes low, in this mode the start and stop of the counter are controlled.</p> <p>1010: Encoder mode - Clock + direction, x2 mode.</p> <p>1011: Encoder mode - Clock+direction, x1 mode, TI2FP2's edge polarity is set by CC2P.</p> <p>1100: Encoder mode - Clock with direction, x2 mode.</p> <p>1101: Encoder mode - Clock with direction, x1 mode, TI1FP1 and TI2FP2 edge polarity is set by CC1P and CC2P.</p> <p>1110: Quadrature encoder mode - x1 mode, counts only TI1FP1 edges, whose edge polarity is set by CC1P.</p> <p>1111: Quadrature encoder mode - x1 mode, counts only TI2FP2 edges, whose polarity is set by CC2P.</p> <p><i>Attention: If TI1F_ED is selected as the trigger input (TS = 00100), the gated mode must not be used. In fact, TI1F_ED outputs one pulse for each TI1F transition, while the gated mode checks the level of the trigger signal.</i></p> <p><i>Attention: The clock of the slave peripheral (timer, ADC, etc.) that receives the TRGO signal must be enabled before the event can be received from the master timer; and the clock frequency must not be changed in real time (prescaler) when the trigger signal is received from the master timer.</i></p> <p><i>Attention: When the slave mode is configured to 0100, 0101, 0110, 1000, and 1001, there is a TIE interrupt flag; otherwise, there is no TIE interrupt flag in the slave mode.</i></p> |



### 17.10.4 ATIM\_DIER DMA/Interrupt enable register

Address offset: 0x0C

Reset value: 0x0000 0000

| Bit field | Name   | Permission | Function description                                                                                                                 |
|-----------|--------|------------|--------------------------------------------------------------------------------------------------------------------------------------|
| 31:24     | RFU    | -          | Reserved bits, please keep the default value.                                                                                        |
| 23        | TERRIE | RW         | Encoder transition error interrupt enable control<br>0: Disable transition error interrupt.<br>1: Enable transition error interrupt. |
| 22        | IERRIE | RW         | Encoder index error interrupt enable control<br>0: Disable index error interrupt.<br>1: Enable index error interrupt.                |
| 21        | DIRIE  | RW         | Encoder direction change interrupt enabled<br>0: Disable direction change interrupt.<br>1: Enable direction change interrupt.        |
| 20        | IDXIE  | RW         | Encoder index interrupt enable control<br>0: Disable index interrupt.<br>1: Enable index interrupt.                                  |
| 19        | CC6DE  | RW         | Capture/compare 6 DMA request enabled<br>0: Disable capture/compare 6 DMA request.<br>1: Enable capture/compare 6 DMA request.       |
| 18        | CC5DE  | RW         | Capture/compare 5 DMA request enabled<br>0: Disable capture/compare 5 DMA request.<br>1: Enable capture/compare 5 DMA request.       |
| 17        | CC6IE  | RW         | Capture/compare 6 interrupt enabled<br>0: Disable capture/compare 6 interrupt.<br>1: Enable capture/compare 6 interrupt.             |
| 16        | CC5IE  | RW         | Capture/compare 5 interrupt enabled<br>0: Disable capture/compare 5 interrupt.<br>1: Enable capture/compare 5 interrupt.             |
| 15        | RFU    | -          | Reserved bits, please keep the default value.                                                                                        |
| 14        | TDE    | RW         | Trigger DMA request enabled<br>0: Disable trigger DMA request.<br>1: Enable trigger DMA request.                                     |
| 13        | COMDE  | RW         | COM DMA request enabled<br>0: Disable COM DMA request.<br>1: Enable COM DMA request.                                                 |
| 12        | CC4DE  | RW         | Capture/compare 4 DMA request enabled<br>0: Disable capture/compare 4 DMA request.<br>1: Enable capture/compare 4 DMA request.       |
| 11        | CC3DE  | RW         | Capture/compare 3 DMA request enabled<br>0: Disable capture/compare 3 DMA request.<br>1: Enable capture/compare 3 DMA request.       |





| Bit field | Name  | Permission | Function description                                                                                                                                                                                                                     |
|-----------|-------|------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 10        | CC2DE | RW         | Capture/compare 2 DMA request enabled<br>0: Disable capture/compare 2 DMA request.<br>1: Enable capture/compare 2 DMA request.                                                                                                           |
| 9         | CC1DE | RW         | Capture/compare 1 DMA request enabled<br>0: Disable capture/compare 1 DMA request.<br>1: Enable capture/compare 1 DMA request.                                                                                                           |
| 8         | UDE   | RW         | Update DMA request enabled<br>0: Disable update DMA request.<br>1: Enable update DMA request.                                                                                                                                            |
| 7         | BIE   | RW         | Brake interrupt enabled<br>0: Disable brake interrupt.<br>1: Enable brake interrupt.                                                                                                                                                     |
| 6         | TIE   | RW         | Trigger interrupt enabled<br>0: Disable trigger interrupt.<br>1: Enable trigger interrupt.<br><i>Attention: When working in slave mode, this bit is only valid when the slave mode configuration is 0100, 0101, 0110, 1000, or 1001.</i> |
| 5         | COMIE | RW         | COM interrupt enabled<br>0: Disable COM interrupt.<br>1: Enable COM interrupt.                                                                                                                                                           |
| 4         | CC4IE | RW         | Capture/compare 4 interrupt enabled<br>0: Disable capture/compare 4 interrupt.<br>1: Enable capture/compare 4 interrupt.                                                                                                                 |
| 3         | CC3IE | RW         | Capture/compare 3 interrupt enabled<br>0: Disable capture/compare 3 interrupt.<br>1: Enable capture/compare 3 interrupt.                                                                                                                 |
| 2         | CC2IE | RW         | Capture/compare 2 interrupt enabled<br>0: Disable capture/compare 2 interrupt.<br>1: Enable capture/compare 2 interrupt.                                                                                                                 |
| 1         | CC1IE | RW         | Capture/compare 1 interrupt enabled<br>0: Disable capture/compare 1 interrupt.<br>1: Enable capture/compare 1 interrupt.                                                                                                                 |
| 0         | UIE   | RW         | Update interrupt enabled<br>0: Disable update interrupt.<br>1: Enable update interrupt.                                                                                                                                                  |



### 17.10.5 ATIM\_ISR Interrupt flag register

Address offset: 0x10

Reset value: 0x0000 0000

| Bit field | Name  | Permission | Function description                                                                                                                                                                                                                                                                                                                                                                                                                    |
|-----------|-------|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:24     | RFU   | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                                                                                                                           |
| 23        | TERRF | RO         | Encoder transition error interrupt flag<br>0: No transition error occurred.<br>1: A transition error has occurred.                                                                                                                                                                                                                                                                                                                      |
| 22        | IERRF | RO         | Encoder index error interrupt flag<br>0: No index error occurred.<br>1: An indexing error has occurred.                                                                                                                                                                                                                                                                                                                                 |
| 21        | DIRF  | RO         | Encoder direction change interrupt flag<br>0: No encoding direction change has occurred.<br>1: Encoding direction change has occurred.                                                                                                                                                                                                                                                                                                  |
| 20        | IDXF  | RO         | Encoder index interrupt flag<br>0: Index flag has not occurred.<br>1: Index flag has occurred.                                                                                                                                                                                                                                                                                                                                          |
| 19        | CC6OF | RO         | Capture/compare 6 repeat capture flag, see CC1OF description.                                                                                                                                                                                                                                                                                                                                                                           |
| 18        | CC5OF | RO         | Capture/compare 5 repeat capture flag, see CC1OF description.                                                                                                                                                                                                                                                                                                                                                                           |
| 17        | CC6IF | RO         | Capture/compare 6 interrupt flag, see CC1IF description.                                                                                                                                                                                                                                                                                                                                                                                |
| 16        | CC5IF | RO         | Capture/compare 5 interrupt flag, see CC1IF description.                                                                                                                                                                                                                                                                                                                                                                                |
| 15:14     | RFU   | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                                                                                                                           |
| 13        | SBIF  | RO         | System brake interrupt flag<br>This flag is set by hardware to 1 whenever the system brake input becomes valid, and can be cleared by software when the system brake input becomes invalid.<br>This flag must be reset in order for the PWM to restart operation.<br>0: No system brake event has occurred.<br>1: A valid level is detected on the system brake input. If BIE = 1 in the ATIM_DIER register, an interrupt is generated. |
| 12        | CC4OF | RO         | Capture/compare 4 repeat capture flag, see CC1OF description.                                                                                                                                                                                                                                                                                                                                                                           |
| 11        | CC3OF | RO         | Capture/compare 3 repeat capture flag, see CC1OF description.                                                                                                                                                                                                                                                                                                                                                                           |
| 10        | CC2OF | RO         | Capture/compare 2 repeat capture flag, see CC1OF description.                                                                                                                                                                                                                                                                                                                                                                           |
| 9         | CC1OF | RO         | Capture/compare 1 repeat capture flag<br>This flag bit is set to 1 by hardware only when the corresponding channel is configured in input capture mode.<br>0: No repeat capture detected.<br>1: Counter value has been captured in the ATIM_CCR1 register and the CC1IF flag has been set to 1.                                                                                                                                         |



| Bit field | Name  | Permission | Function description                                                                                                                                                                                                                                                                                                                                                                                     |
|-----------|-------|------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 8         | B2IF  | RO         | Brake 2 interrupt flag<br>This flag is set by hardware to 1 whenever the brake 2 input becomes valid, and can be cleared by software after the brake 2 input becomes invalid.<br>0: No brake 2 event has occurred.<br>1: A valid level is detected on the brake 2 input. If BIE = 1 in the ATIM_DIER register, an interrupt is generated.                                                                |
| 7         | BIF   | RO         | Brake interrupt flag<br>This flag is set by hardware to 1 whenever the brake input becomes valid, and can be cleared by software after the brake input becomes invalid.<br>0: No brake event has occurred.<br>1: A valid level is detected on the brake input. If BIE = 1 in the ATIM_DIER register, an interrupt is generated.                                                                          |
| 6         | TIF   | RO         | Trigger interrupt flag<br>In all modes except gated mode, this flag will be set by hardware to 1 when a valid edge is detected on the TRGI input after enabling the slave mode controller. When gated mode is selected, this flag will be set to 1 when the counter is started and stopped. This flag needs to be cleared by software.<br>0: No trigger event occurred.<br>1: Trigger interrupt pending. |
| 5         | COMIF | RO         | COM interrupt flag<br>This flag is set to 1 by hardware when a COM event occurs (when the capture/compare control bits CCyE, CCyNE, and OCyM have been updated), but needs to be cleared by software.<br>0: COM event has not occurred.<br>1: COM interrupt pending.                                                                                                                                     |
| 4         | CC4IF | RO         | Capture/compare 4 interrupt flag, see CC1IF description.                                                                                                                                                                                                                                                                                                                                                 |
| 3         | CC3IF | RO         | Capture/compare 3 interrupt flag, see CC1IF description.                                                                                                                                                                                                                                                                                                                                                 |
| 2         | CC2IF | RO         | Capture/compare 2 interrupt flag, see CC1IF description.                                                                                                                                                                                                                                                                                                                                                 |



| Bit field | Name  | Permission | Function description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|-----------|-------|------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1         | CC1IF | RO         | <p>Capture/compare 1 interrupt flag</p> <p>If channel CC1 is configured to output:</p> <ul style="list-style-type: none"> <li>This flag is set by hardware when the counter matches the compare value, except in center-aligned mode (see <a href="#">17.10.1 ATIM_CR1 Control register 1</a> for a description of the CMS bit). But it needs to be cleared by software.</li> </ul> <p>0: No match.<br/>1: The value of the ATIM_CNT counter matches the value of the ATIM_CCR1 register.</p> <ul style="list-style-type: none"> <li>When the value of ATIM_CCR1 is greater than the value of ATIM_ARR, the CC1IF bit will go high when the counter overflows (in count up mode and center-aligned mode) or underflows (in count down mode).</li> </ul> <p>If channel CC1 is configured as an input:</p> <ul style="list-style-type: none"> <li>This bit is set by hardware when a capture event occurs. Clear this bit by software or by reading the ATIM_CCR1 register.</li> </ul> <p>0: No input capture event occurred<br/>1: Counter value captured in ATIM_CCR1 register (edge matching selected polarity detected on IC1)</p> |
| 0         | UIF   | RO         | <p>Update interrupt flag</p> <p>This bit is set to 1 by hardware when an update event occurs. But needs to be cleared by software.</p> <p>0: No updates occurred.<br/>1: Update interrupt pending.</p> <p>This bit is set to 1 by hardware when the register is updated when:</p> <ul style="list-style-type: none"> <li>UDIS = 0 in the ATIM_CR1 register and the repeat counter value underflows (updated when repeat counter = 0).</li> <li>URS = 0 and UDIS = 0 in the ATIM_CR1 register and CNT is reinitialized by software using the UG bit in the ATIM_EGR register.</li> <li>URS = 0 and UDIS = 0 in the ATIM_CR1 register and CNT is reinitialized by a trigger event.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                          |



### 17.10.6 ATIM\_ICR Interrupt flag clear register

Address offset: 0x70

Reset value: 0x00FF 3FFF

| Bit field | Name  | Permission | Function description                                                                                                    |
|-----------|-------|------------|-------------------------------------------------------------------------------------------------------------------------|
| 31:24     | RFU   | -          | Reserved bits, please keep the default value.                                                                           |
| 23        | TERRF | R1W0       | Encoder transition error interrupt flag cleared<br>0: Clear encoder transition error interrupt flag.<br>1: No function. |
| 22        | IERRF | R1W0       | Encoder index error interrupt flag cleared<br>0: Clear encoder index error interrupt flag.<br>1: No function.           |
| 21        | DIRF  | R1W0       | Encoder direction change interrupt flag cleared<br>0: Clear encoder direction change interrupt flag.<br>1: No function. |
| 20        | IDXF  | R1W0       | Encoder index interrupt flag cleared<br>0: Clear encoder index interrupt flag.<br>1: No function.                       |
| 19        | CC6OF | R1W0       | Capture/compare 6 repeat capture flag cleared, see CC1OF description.                                                   |
| 18        | CC5OF | R1W0       | Capture/compare 5 repeat capture flag cleared, see CC1OF description.                                                   |
| 17        | CC6IF | R1W0       | Capture/compare 6 interrupt flag cleared, see CC1IF description.                                                        |
| 16        | CC5IF | R1W0       | Capture/compare 5 interrupt flag cleared, see CC1IF description.                                                        |
| 15:14     | RFU   | -          | Reserved bits, please keep the default value.                                                                           |
| 13        | SBIF  | R1W0       | System brake interrupt flag cleared<br>0: Clear system brake interrupt flag.<br>1: No function.                         |
| 12        | CC4OF | R1W0       | Capture/compare 4 repeat capture flag cleared, see CC1OF description.                                                   |
| 11        | CC3OF | R1W0       | Capture/compare 3 repeat capture flag cleared, see CC1OF description.                                                   |
| 10        | CC2OF | R1W0       | Capture/compare 2 repeat capture flag cleared, see CC1OF description.                                                   |
| 9         | CC1OF | R1W0       | Capture/compare 1 repeat capture flag cleared<br>0: Clear capture/compare 1 repeat capture flag.<br>1: No function.     |
| 8         | B2IF  | R1W0       | Brake 2 interrupt flag cleared<br>0: Clear brake 2 interrupt flag.<br>1: No function.                                   |
| 7         | BIF   | R1W0       | Brake break flag cleared<br>0: Clear brake interrupt flag.<br>1: No function.                                           |
| 6         | TIF   | R1W0       | Trigger interrupt flag cleared<br>0: Clear trigger interrupt flag.<br>1: No function.                                   |



| Bit field | Name  | Permission | Function description                                                                                      |
|-----------|-------|------------|-----------------------------------------------------------------------------------------------------------|
| 5         | COMIF | R1W0       | COM interrupt flag cleared<br>0: Clear COM interrupt flag.<br>1: No function.                             |
| 4         | CC4IF | R1W0       | Capture/compare 4 interrupt flag cleared, see CC1IF description.                                          |
| 3         | CC3IF | R1W0       | Capture/compare 3 interrupt flag cleared, see CC1IF description.                                          |
| 2         | CC2IF | R1W0       | Capture/compare 2 interrupt flag cleared, see CC1IF description.                                          |
| 1         | CC1IF | R1W0       | Capture/compare 1 interrupt flag cleared<br>0: Clear capture/compare 1 interrupt flag.<br>1: No function. |
| 0         | UIF   | R1W0       | Update interrupt flag cleared<br>0: Clear update interrupt flag.<br>1: No function.                       |



### 17.10.7 ATIM\_EGR Event generation register

Address offset: 0x14

Reset value: 0x0000 0000

| Bit field | Name | Permission | Function description                                                                                                                                                                                                                                                                                                         |
|-----------|------|------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:18     | RFU  | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                |
| 17        | CC6G | RW         | Capture/compare 6 generation, see CC1G description.                                                                                                                                                                                                                                                                          |
| 16        | CC5G | RW         | Capture/compare 5 generation, see CC1G description.                                                                                                                                                                                                                                                                          |
| 15:9      | RFU  | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                |
| 8         | B2G  | RW         | Brake 2 generated.<br>This bit is set to 1 by software to generate an event and is automatically cleared by hardware.<br>0: No operation is performed.<br>1: Brake 2 event is generated. The MOE bit is cleared and the B2IF flag is set to 1. Correlation interrupts can occur when enabled.                                |
| 7         | BG   | RW         | Brake generated<br>This bit is set to 1 by software to generate an event and is automatically cleared by hardware.<br>0: No operation is performed.<br>1: Brake event is generated. The MOE bit is cleared and the BIF flag is set to 1. Correlation interrupts can occur when enabled.                                      |
| 6         | TG   | RW         | Trigger generated<br>This bit is set to 1 by software to generate an event and is automatically cleared by hardware.<br>0: No operation is performed.<br>1: The TIF flag in ATIM_ISR register is set to 1. When enabled, the associated interrupt or DMA transfer event can occur.                                           |
| 5         | COMG | RW         | Capture/compare control update generated<br>This bit can be set to 1 by software and automatically cleared by hardware.<br>0: No operation is performed.<br>1: CCyE, CCyNE and OCyM bits can be updated when the CCPC bit field is set to 1.<br><i>Note: This bit is only valid for channels with complementary outputs.</i> |
| 4         | CC4G | RW         | Capture/compare 4 generation, see CC1G description.                                                                                                                                                                                                                                                                          |
| 3         | CC3G | RW         | Capture/compare 3 generation, see CC1G description.                                                                                                                                                                                                                                                                          |
| 2         | CC2G | RW         | Capture/compare 2 generation, see CC1G description.                                                                                                                                                                                                                                                                          |



| Bit field | Name | Permission | Function description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|-----------|------|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1         | CC1G | RW         | <p>Capture/compare 1 generated</p> <p>This bit is set to 1 by software to generate an event and is automatically cleared by hardware.</p> <p>0: No operation is performed.</p> <p>1: Capture/compare events are generated on channel 1.</p> <p>If channel CC1 is configured as output, when enabled, the CC1IF flag is set to 1 and the corresponding interrupt or DMA request is sent.</p> <p>If channel CC1 is configured as input, the current value of the counter will be captured in the ATIM_CCR1 register. When enabled, the CC1IF flag is set to 1 and the appropriate interrupt or DMA request is sent. If the CC1IF flag is already high, the CC1OF flag will be set to 1.</p> <p><i>Attention: CC1E must be enabled, otherwise CCR1 register will capture the current value of the counter, but no interrupt flag will be generated, and no interrupt can be entered.</i></p> |
| 0         | UG   | RW         | <p>Update generated</p> <p>This bit can be set to 1 by software and automatically cleared by hardware.</p> <p>0: No operation is performed.</p> <p>1: Reinitialise the counter and generate a register update event. Note that the prescaler counter will also be cleared (but the prescaler ratio is not affected). The counter will be cleared if center-aligned mode or count up mode is selected; if down-count mode is selected, the counter will use the auto-reload value ATIM_ARR.</p>                                                                                                                                                                                                                                                                                                                                                                                            |





### 17.10.8 ATIM\_CCMR1CAP Capture mode register 1

Address offset: 0x18      Reset value: 0x0000 0000

This register is the same register as ATIM\_CCMR1CMP and can be used in input capture mode or output compare mode. The channel direction is defined by configuring the corresponding CCyS bit. All other bits of this register function differently in input capture mode and output compare mode.

| Bit field                                         | Name                                     | Permission | Function description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |                                                   |                                         |                                        |                                         |                                        |                                          |                                        |                                          |                                         |                                          |                                         |                                          |                                         |                                          |                                         |                                          |
|---------------------------------------------------|------------------------------------------|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------|-----------------------------------------|----------------------------------------|-----------------------------------------|----------------------------------------|------------------------------------------|----------------------------------------|------------------------------------------|-----------------------------------------|------------------------------------------|-----------------------------------------|------------------------------------------|-----------------------------------------|------------------------------------------|-----------------------------------------|------------------------------------------|
| 31:16                                             | RFU                                      | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |                                                   |                                         |                                        |                                         |                                        |                                          |                                        |                                          |                                         |                                          |                                         |                                          |                                         |                                          |                                         |                                          |
| 15:12                                             | IC2F                                     | RW         | Input capture 2 filter, see IC1F description.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |                                                   |                                         |                                        |                                         |                                        |                                          |                                        |                                          |                                         |                                          |                                         |                                          |                                         |                                          |                                         |                                          |
| 11:10                                             | IC2PSC                                   | RW         | Input capture 2 prescaler, see IC1PSC description.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |                                                   |                                         |                                        |                                         |                                        |                                          |                                        |                                          |                                         |                                          |                                         |                                          |                                         |                                          |                                         |                                          |
| 9:8                                               | CC2S                                     | RW         | Capture/compare 2 selection<br>This bit field defines the channel direction (input/output) and the input mapping used.<br>00: CC2 channel configured as output.<br>01: CC2 channel configured as input, IC2 mapped to TI2.<br>10: CC2 channel configured as input, IC2 mapped to TI1.<br>11: CC2 channel configured as input, IC2 mapped to TRC, this mode is only valid when the internal trigger input is selected via the TS bit (ATIM_SMCR register).<br><i>Attention: Data can be written to the CC2S bit only when the channel is closed (CC2E = 0 in ATIM_CCER).</i>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |                                                   |                                         |                                        |                                         |                                        |                                          |                                        |                                          |                                         |                                          |                                         |                                          |                                         |                                          |                                         |                                          |
| 7:4                                               | IC1F                                     | RW         | Input capture 1 filter<br>This bit field defines the sampling frequency of the TI1 input and the sample length of the digital filter applicable to TI1. The digital filter consists of an event counter, where every N consecutive events are considered to be only one valid output edge.<br><table><tr><td>0000: No filter, sampled at frequency <math>f_{DTS}</math>.</td><td>1000: <math>f_{SAMPLING} = f_{DTS}/8, N = 6</math></td></tr><tr><td>0001: <math>f_{SAMPLING} = f_{PCLK}, N = 2</math></td><td>1001: <math>f_{SAMPLING} = f_{DTS}/8, N = 8</math></td></tr><tr><td>0010: <math>f_{SAMPLING} = f_{PCLK}, N = 4</math></td><td>1010: <math>f_{SAMPLING} = f_{DTS}/16, N = 5</math></td></tr><tr><td>0011: <math>f_{SAMPLING} = f_{PCLK}, N = 8</math></td><td>1011: <math>f_{SAMPLING} = f_{DTS}/16, N = 6</math></td></tr><tr><td>0100: <math>f_{SAMPLING} = f_{DTS}/2, N = 6</math></td><td>1100: <math>f_{SAMPLING} = f_{DTS}/16, N = 8</math></td></tr><tr><td>0101: <math>f_{SAMPLING} = f_{DTS}/2, N = 8</math></td><td>1101: <math>f_{SAMPLING} = f_{DTS}/32, N = 5</math></td></tr><tr><td>0110: <math>f_{SAMPLING} = f_{DTS}/4, N = 6</math></td><td>1110: <math>f_{SAMPLING} = f_{DTS}/32, N = 6</math></td></tr><tr><td>0111: <math>f_{SAMPLING} = f_{DTS}/4, N = 8</math></td><td>1111: <math>f_{SAMPLING} = f_{DTS}/32, N = 8</math></td></tr></table> | 0000: No filter, sampled at frequency $f_{DTS}$ . | 1000: $f_{SAMPLING} = f_{DTS}/8, N = 6$ | 0001: $f_{SAMPLING} = f_{PCLK}, N = 2$ | 1001: $f_{SAMPLING} = f_{DTS}/8, N = 8$ | 0010: $f_{SAMPLING} = f_{PCLK}, N = 4$ | 1010: $f_{SAMPLING} = f_{DTS}/16, N = 5$ | 0011: $f_{SAMPLING} = f_{PCLK}, N = 8$ | 1011: $f_{SAMPLING} = f_{DTS}/16, N = 6$ | 0100: $f_{SAMPLING} = f_{DTS}/2, N = 6$ | 1100: $f_{SAMPLING} = f_{DTS}/16, N = 8$ | 0101: $f_{SAMPLING} = f_{DTS}/2, N = 8$ | 1101: $f_{SAMPLING} = f_{DTS}/32, N = 5$ | 0110: $f_{SAMPLING} = f_{DTS}/4, N = 6$ | 1110: $f_{SAMPLING} = f_{DTS}/32, N = 6$ | 0111: $f_{SAMPLING} = f_{DTS}/4, N = 8$ | 1111: $f_{SAMPLING} = f_{DTS}/32, N = 8$ |
| 0000: No filter, sampled at frequency $f_{DTS}$ . | 1000: $f_{SAMPLING} = f_{DTS}/8, N = 6$  |            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |                                                   |                                         |                                        |                                         |                                        |                                          |                                        |                                          |                                         |                                          |                                         |                                          |                                         |                                          |                                         |                                          |
| 0001: $f_{SAMPLING} = f_{PCLK}, N = 2$            | 1001: $f_{SAMPLING} = f_{DTS}/8, N = 8$  |            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |                                                   |                                         |                                        |                                         |                                        |                                          |                                        |                                          |                                         |                                          |                                         |                                          |                                         |                                          |                                         |                                          |
| 0010: $f_{SAMPLING} = f_{PCLK}, N = 4$            | 1010: $f_{SAMPLING} = f_{DTS}/16, N = 5$ |            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |                                                   |                                         |                                        |                                         |                                        |                                          |                                        |                                          |                                         |                                          |                                         |                                          |                                         |                                          |                                         |                                          |
| 0011: $f_{SAMPLING} = f_{PCLK}, N = 8$            | 1011: $f_{SAMPLING} = f_{DTS}/16, N = 6$ |            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |                                                   |                                         |                                        |                                         |                                        |                                          |                                        |                                          |                                         |                                          |                                         |                                          |                                         |                                          |                                         |                                          |
| 0100: $f_{SAMPLING} = f_{DTS}/2, N = 6$           | 1100: $f_{SAMPLING} = f_{DTS}/16, N = 8$ |            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |                                                   |                                         |                                        |                                         |                                        |                                          |                                        |                                          |                                         |                                          |                                         |                                          |                                         |                                          |                                         |                                          |
| 0101: $f_{SAMPLING} = f_{DTS}/2, N = 8$           | 1101: $f_{SAMPLING} = f_{DTS}/32, N = 5$ |            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |                                                   |                                         |                                        |                                         |                                        |                                          |                                        |                                          |                                         |                                          |                                         |                                          |                                         |                                          |                                         |                                          |
| 0110: $f_{SAMPLING} = f_{DTS}/4, N = 6$           | 1110: $f_{SAMPLING} = f_{DTS}/32, N = 6$ |            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |                                                   |                                         |                                        |                                         |                                        |                                          |                                        |                                          |                                         |                                          |                                         |                                          |                                         |                                          |                                         |                                          |
| 0111: $f_{SAMPLING} = f_{DTS}/4, N = 8$           | 1111: $f_{SAMPLING} = f_{DTS}/32, N = 8$ |            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |                                                   |                                         |                                        |                                         |                                        |                                          |                                        |                                          |                                         |                                          |                                         |                                          |                                         |                                          |                                         |                                          |
| 3:2                                               | IC1PSC                                   | RW         | Input capture 1 prescaler<br>This bit field defines the prescaler ratio for the CC1 input (IC1). The prescaler is reset as soon as CC1E = 0 (ATIM_CCER register).<br>00: No prescaler, capture is performed every time an edge is detected on the capture input.<br>01: Capture is performed every 2 events.<br>10: Capture is performed every 4 events.<br>11: Capture is performed every 8 events.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |                                                   |                                         |                                        |                                         |                                        |                                          |                                        |                                          |                                         |                                          |                                         |                                          |                                         |                                          |                                         |                                          |

| Bit field | Name | Permission | Function description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|-----------|------|------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1:0       | CC1S | RW         | <p>Capture/compare 1 selection</p> <p>This bit field defines the channel direction (input/output) and the input mapping used.</p> <p>00: CC1 channel configured as output.</p> <p>01: CC1 channel configured as input, IC1 mapped to TI1.</p> <p>10: CC1 channel configured as input, IC1 mapped to TI2.</p> <p>11: CC1 channel configured as input, IC1 mapped to TRC, this mode is only valid when the internal trigger input is selected via the TS bit (ATIM_SMCR register).</p> <p><i>Attention: Data can be written to the CC1S bit only when the channel is closed (CC1E = 0 in ATIM_CCER).</i></p> |



### 17.10.9 ATIM\_CCMR1CMP Compare mode register 1

Address offset: 0x18      Reset value: 0x0000 0000

This register is the same register as ATIM\_CCMR1CAP and can be used in input capture mode or output compare mode. The channel direction is defined by configuring the corresponding CCyS bit. All other bits of this register function differently in input capture mode and output compare mode.

| Bit field | Name  | Permission | Function description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|-----------|-------|------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:25     | RFU   | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 24        | OC2MH | RW         | Used in conjunction with OC2M, it is the highest bit of OC2M.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 23:17     | RFU   | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 16        | OC1MH | RW         | Used in conjunction with OC1M, it is the highest bit of OC1M.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 15        | OC2CE | RW         | Output compare 2 clear enabled, see OC1CE description.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 14:12     | OC2M  | RW         | Output compare 2 mode, see OC1M description.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 11        | OC2PE | RW         | Output compare 2 preload enabled, see OC1PE description.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| 10        | OC2FE | RW         | Output compare 2 fast enabled, see OC1FE description.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 9:8       | CC2S  | RW         | <p>Capture/compare 2 selection<br/>This bit field defines the channel direction (input/output) and the input mapping used.</p> <p>00: CC2 channel configured as output.<br/>01: CC2 channel configured as input, IC2 mapped to TI2.<br/>10: CC2 channel configured as input, IC2 mapped to TI1.<br/>11: CC2 channel configured as input, IC2 mapped to TRC, this mode is only valid when the internal trigger input is selected via the TS bit (ATIM_SMCR register).</p> <p><b>Attention: Data can be written to the CC2S bit only when the channel is closed (CC2E = 0 in ATIM_CCER).</b></p>                         |
| 7         | OC1CE | RW         | <p>Output compare 1 clear enabled<br/>0: OC1REF is not affected by the OCREF_CLR_INT input.<br/>1: OC1REF is immediately cleared when a high level is detected on the OCREF_CLR_INT input.</p>                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 6:4       | OC1M  | RW         | <p>Output compare 1 mode<br/>These bits define the behaviour of the output reference signal OC1REF which provides OC1 and OC1N. OC1REF is valid at high level, while the valid level of OC1 and OC1N depends on the CC1P and CC1NP bit.</p> <p>0000: Freeze - Comparison of output compare register ATIM_CCR1 with counter ATIM_CNT will not have any effect on the output. (This mode is used for time base generation).</p> <p>0001: Set channel 1 to output a valid level when matched. The OC1REF signal is forced to high level when counter ATIM_CNT is matched with capture/compare register 1 (ATIM_CCR1).</p> |



| Bit field          | Name | Permission | Function description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|--------------------|------|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 6:4<br>(continued) | OC1M | RW         | <p>0010: Set channel 1 to output an invalid level when matched. When counter ATIM_CNT matches capture/ compare register 1 (ATIM_CCR1), the OC1REF signal is forced to low level.</p> <p>0011: Flipped - OC1REF is flipped when ATIM_CNT = ATIM_CCR1.</p> <p>0100: Forced to invalid level - OC1REF is forced to low level.</p> <p>0101: Forced to valid level - OC1REF is forced to high level.</p> <p>0110: PWM mode 1 - In count up mode, channel 1 is valid (OC1REF = 1) as long as ATIM_CNT &lt; ATIM_CCR1, otherwise it is invalid (OC1REF = 0). In count down mode, channel 1 is invalid (OC1REF = 0) as long as ATIM_CNT &gt; ATIM_CCR1, otherwise it is valid (OC1REF = 1).</p> <p>0111: PWM mode 2 - In count up mode, channel 1 is invalid (OC1REF = 0) whenever ATIM_CNT &lt; ATIM_CCR1, otherwise it is valid (OC1REF = 1). In count down mode, channel 1 is valid (OC1REF = 1) as long as ATIM_CNT &gt; ATIM_CCR1, otherwise it is invalid (OC1REF = 0).</p> <p>1000: Retriggerable OPM mode 1 - In count up mode, the channel is in valid state until a trigger event is detected (on the TRGI signal). Then, a comparison is made in PWM mode 1 and the channel will become valid again at the next update. In count down mode, the channel is invalid until a trigger event is detected (on the TRGI signal). Then, a comparison is made in PWM mode 1 and the channel becomes invalid again at the next update.</p> <p>1001: Retriggerable OPM mode 2 - In count up mode, the channel is invalid until a trigger event is detected (on the TRGI signal). Then, a comparison is made in PWM mode 2 and the channel becomes invalid again at the next update. In count down mode, the channel is valid until a trigger event is detected (on the TRGI signal). Then, the comparison is performed in PWM mode 2 and the channel becomes valid again at the next update.</p> <p>1010: Reserved.</p> <p>1011: Count direction output, OC1REF signal is the count direction signal for the DIR bit.</p> <p>1100: Combined PWM Mode 1 - OC1REF behaves the same as in PWM Mode 1. OC1REFC is the logical OR of OC1REF and OC2REF.</p> <p>1101: Combined PWM Mode 2 - OC1REF behaves the same as in PWM Mode 2. OC1REFC is the logical AND of OC1REF and OC2REF.</p> <p>1110: Asymmetric PWM mode 1 - OC1REF behaves the same as in PWM Mode 1. OC1REFC outputs OC1REF when the counter counts up and OC2REF when the counter counts down.</p> <p>1111: Asymmetric PWM mode 2 - OC1REF behaves the same as in PWM Mode 2. OC1REFC outputs OC1REF when the counter counts up and OC2REF when the counter counts down.</p> <p><i>Attention: These bits cannot be modified as long as LOCK (LOCK bit in ATIM_BDTR register) level 3 is programmed and CC1S = "00" (channel configured as output).</i></p> <p><i>Attention: In PWM mode, the OCyREF level changes only when the comparison result changes or when the output comparison mode is switched from Freeze mode to PWM mode.</i></p> <p><i>Attention: This bit field will be preloaded on channels with complementary outputs. If the CCPC bit in the ATIM_CR2 register is 1, the OC1M valid bit will get a new value from the preload bit only when a COM event is generated.</i></p> |



| Bit field | Name  | Permission | Function description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|-----------|-------|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 3         | OC1PE | RW         | <p>Output compare 1 preload enabled</p> <p>0: Disable the preload register related to ATIM_CCR1. Data can be written to ATIM_CCR1 at any time and the new value will be used immediately after writing.</p> <p>1: Enable the preload register related to ATIM_CCR1. Read/write access to the preloaded registers. The ATIM_CCR1 preloaded value is loaded into the valid registers each time that an update event is generated.</p> <p><i>Attention: These bits cannot be modified as long as LOCK (LOCK bit in ATIM_BDTR register) level 3 is programmed and CC1S = "00"(channel configured as output).</i></p> <p><i>Attention: PWM mode without validation of the preload register is only possible in single pulse mode (OPM bit field in the ATIM_CR1 register is set to 1). The behaviour is not guaranteed in other cases.</i></p> |
| 2         | OC1FE | RW         | <p>Output compare 1 fast enabled</p> <p>This bit is used to speed up the effect of a trigger input event on the CC output.</p> <p>0: Even if the trigger is on, CC1 will work properly according to the counter and CCR1 values. When an edge is present on the trigger input, the minimum delay to activate the CC1 output is 5 clock cycles.</p> <p>1: The presence of a valid edge on the trigger input corresponds to a comparison match on the CC1 output. Subsequently, OC is set to the compare level regardless of the comparison result. The delay time between sampling the trigger input and activating the CC1 output is reduced to 3 clock cycles. OCyFE works only when the channel is configured for PWM1 or PWM2 mode.</p>                                                                                                |
| 1:0       | CC1S  | RW         | <p>Capture/compare 1 selection</p> <p>This bit field defines the channel direction (input/output) and the input mapping used.</p> <p>00: CC1 channel configured as output.</p> <p>01: CC1 channel configured as input, IC1 mapped to TI1.</p> <p>10: CC1 channel configured as input, IC1 mapped to TI2.</p> <p>11: CC1 channel configured as input, IC1 mapped to TRC, this mode is only valid when the internal trigger input is selected via the TS bit (ATIM_SMCR register).</p> <p><i>Attention: Data can be written to the CC1S bit only when the channel is closed (CC1E = 0 in ATIM_CCER).</i></p>                                                                                                                                                                                                                                |



### 17.10.10 ATIM\_CCMR2CAP Capture mode register 2

Address offset: 0x1C      Reset value: 0x0000 0000

This register is the same register as ATIM\_CCMR2CMP and can be used in input capture mode or output compare mode. The channel direction is defined by configuring the corresponding CCyS bit. All other bits of this register function differently in input capture mode and output compare mode.

| Bit field | Name   | Permission | Function description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|-----------|--------|------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:16     | RFU    | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 15:12     | IC4F   | RW         | Input capture 4 filter, see IC1F description.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 11:10     | IC4PSC | RW         | Input capture 4 prescaler, see IC1PSC description.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| 9:8       | CC4S   | RW         | Capture/compare 4 selection<br>This bit field defines the channel direction (input/output) and the input mapping used.<br>00: CC4 channel configured as output.<br>01: CC4 channel configured as input, IC4 mapped to TI4.<br>10: CC4 channel configured as input, IC4 mapped to TI3.<br>11: CC4 channel configured as input, IC4 mapped to TRC, this mode is only valid when internal trigger input is selected via the TS bit (ATIM_SMCR register).<br><i>Attention: Data can be written to the CC4S bit only when the channel is off (CC4E = 0 in ATIM_CCER).</i> |
| 7:4       | IC3F   | RW         | Input capture 3 filter, see IC1F description.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 3:2       | IC3PSC | RW         | Input capture 3 prescaler, see IC1PSC description.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| 1:0       | CC3S   | RW         | Capture/compare 3 selection<br>This bit field defines the channel direction (input/output) and the input mapping used.<br>00: CC3 channel configured as output.<br>01: CC3 channel configured as input, IC3 mapped to TI3.<br>10: CC3 channel configured as input, IC3 mapped to TI4.<br>11: CC3 channel configured as input, IC3 mapped to TRC, this mode is only valid when internal trigger input is selected via the TS bit (ATIM_SMCR register).<br><i>Attention: Data can be written to the CC3S bit only when the channel is off (CC3E = 0 in ATIM_CCER).</i> |



### 17.10.11 ATIM\_CCMR2CMP Compare mode register 2

Address offset: 0x1C      Reset value: 0x0000 0000

This register is the same register as ATIM\_CCMR2CAP and can be used in input capture mode or output compare mode. The channel direction is defined by configuring the corresponding CCyS bit. All other bits of this register function differently in input capture mode and output compare mode.

| Bit field | Name  | Permission | Function description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|-----------|-------|------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:25     | RFU   | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 24        | OC4MH | RW         | Used in conjunction with OC4M, it is the highest bit of the OC4M.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| 23:17     | RFU   | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 16        | OC3MH | RW         | Used in conjunction with OC3M, it is the highest bit of the OC3M.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| 15        | OC4CE | RW         | Output compare 4 clear enabled, see OC1CE description.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| 14:12     | OC4M  | RW         | Output compare 4 mode, see OC1M description.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 11        | OC4PE | RW         | Output compare 4 preload enabled, see OC1PE description.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 10        | OC4FE | RW         | Output compare 4 fast enabled, see OC1FE description.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 9:8       | CC4S  | RW         | Capture/compare 4 selection<br>This bit field defines the channel direction (input/output) and the input mapping used.<br>00: CC4 channel configured as output.<br>01: CC4 channel configured as input, IC4 mapped to TI4.<br>10: CC4 channel configured as input, IC4 mapped to TI3.<br>11: CC4 channel configured as input, IC4 mapped to TRC, this mode is only valid when internal trigger input is selected via the TS bit (ATIM_SMCR register).<br><i>Attention: Data can be written to the CC4S bit only when the channel is off (CC4E = 0 in ATIM_CCER).</i> |
| 7         | OC3CE | RW         | Output compare 3 clear to enabled, see OC1CE description.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 6:4       | OC3M  | RW         | Output compare 3 mode, see OC1M description.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 3         | OC3PE | RW         | Output compare 3 preload enabled, see OC1PE description.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 2         | OC3FE | RW         | Output compare 3 fast enabled, see OC1FE description.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 1:0       | CC3S  | RW         | Capture/compare 3 selection<br>This bit field defines the channel direction (input/output) and the input used.<br>00: CC3 channel configured as output.<br>01: CC3 channel configured as input, IC3 mapped to TI3.<br>10: CC3 channel configured as input, IC3 mapped to TI4.<br>11: CC3 channel configured as input, IC3 mapped to TRC. This mode is only valid when the internal trigger input is selected via the TS bit (ATIM_SMCR register).<br><i>Attention: Data can be written to the CC3S bit only when the channel is off (CC3E = 0 in ATIM_CCER).</i>     |



### 17.10.12 ATIM\_CCMR3CAP Capture mode register 3

Address offset: 0x50      Reset value: 0x0000 0000

This register is the same register as ATIM\_CCMR3CMP and can be used in input capture mode or output compare mode. The channel direction is defined by configuring the corresponding CCyS bit. All other bits of this register function differently in input capture mode and output compare mode.

| Bit field | Name   | Permission | Function description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|-----------|--------|------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:16     | RFU    | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 15:12     | IC6F   | RW         | Input capture 6 filter, see IC1F description.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 11:10     | IC6PSC | RW         | Input capture 6 prescaler, see IC1PSC description.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 9:8       | CC6S   | RW         | Capture/compare 6 selection<br>This bit field defines the channel direction (input/output) and the input mapping used.<br>00: CC6 channel configured as output.<br>01: CC6 channel configured as input, IC6 mapped to TI6.<br>10: CC6 channel configured as input, IC6 mapped to TI5.<br>11: CC6 channel configured as input, IC6 mapped to TRC. This mode is only valid when the internal trigger input is selected via the TS bit (ATIM_SMCR register).<br><i>Attention: Data can be written to the CC6S bit only when the channel is off (CC6E = 0 in ATIM_CCER).</i> |
| 7:4       | IC5F   | RW         | Input capture 5 filter, see IC1F description.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 3:2       | IC5PSC | RW         | Input capture 5 prescaler, see IC1PSC description.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 1:0       | CC5S   | RW         | Capture/compare 5 selection<br>This bit field defines the channel direction (input/output) and the input mapping used.<br>00: CC5 channel configured as output.<br>01: CC5 channel configured as input, IC5 mapped to TI5.<br>10: CC5 channel configured as input, IC5 mapped to TI6.<br>11: CC5 channel configured as input, IC5 mapped to TRC. This mode is only valid when the internal trigger input is selected via the TS bit (ATIM_SMCR register).<br><i>Attention: Data can be written to the CC5S bit only when the channel is off (CC5E = 0 in ATIM_CCER).</i> |





### 17.10.13 ATIM\_CCMR3CMP Compare mode register 3

Address offset: 0x50      Reset value: 0x0000 0000

This register is the same register as ATIM\_CCMR3CAP and can be used in input capture mode or output compare mode. The channel direction is defined by configuring the corresponding CCyS bit. All other bits of this register function differently in input capture mode and output compare mode.

| Bit field | Name  | Permission | Function description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|-----------|-------|------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:25     | RFU   | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| 24        | OC6MH | RW         | Used in conjunction with OC6M, it is the highest bit of the OC6M.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 23:17     | RFU   | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| 16        | OC5MH | RW         | Used in conjunction with OC5M, it is the highest bit of the OC5M.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 15        | OC6CE | RW         | Output compare 6 clear enabled, see OC1CE description.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 14:12     | OC6M  | RW         | Output compare 6 mode, see OC1M description.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 11        | OC6PE | RW         | Output compare 6 preload enabled, see OC1PE description.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 10        | OC6FE | RW         | Output compare 6 fast enabled, see OC1FE description.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 9:8       | CC6S  | RW         | Capture/compare 6 selection<br>This bit field defines the channel direction (input/output) and the input used.<br>00: CC6 channel configured as output.<br>01: CC6 channel configured as input, IC6 mapped to TI6.<br>10: CC6 channel configured as input, IC6 mapped to TI5.<br>11: CC6 channel configured as input, IC6 mapped to TRC. This mode is only valid when the internal trigger input is selected via the TS bit (ATIM_SMCR register).<br><i>Attention: Data can be written to the CC6S bit only when the channel is off (CC6E = 0 in ATIM_CCER).</i> |
| 7         | OC5CE | RW         | Output compare 5 clear enabled, see OC1CE description.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 6:4       | OC5M  | RW         | Output compare 5 mode, see OC1M description.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 3         | OC5PE | RW         | Output compare 5 preload enabled, see OC1PE description.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 2         | OC5FE | RW         | Output compare 5 fast enabled, see OC1FE description.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 1:0       | CC5S  | RW         | Capture/compare 5 selection<br>This bit field defines the channel direction (input/output) and the input used.<br>00: CC5 channel configured as output.<br>01: CC5 channel configured as input, IC5 mapped to TI5.<br>10: CC5 channel configured as input, IC5 mapped to TI6.<br>11: CC5 channel configured as input, IC5 mapped to TRC. This mode is only valid when the internal trigger input is selected via the TS bit (ATIM_SMCR register).<br><i>Attention: Data can be written to the CC5S bit only when the channel is off (CC5E = 0 in ATIM_CCER).</i> |



### 17.10.14 ATIM\_CCER Capture/compare enable register

Address offset: 0x20

Reset value: 0x0000 0000

| Bit field | Name  | Permission | Function description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|-----------|-------|------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:24     | RFU   | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 23        | CC6NP | RW         | Capture/compare 6 complementary output polarity, see CC1NP.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 22        | CC6NE | RW         | Capture/compare 6 complementary output enabled, see CC1NE.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 21        | CC6P  | RW         | Capture/compare 6 output polarity, see CC1P.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 20        | CC6E  | RW         | Capture/compare 6 output enabled, see CC1E.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 19        | CC5NP | RW         | Capture/compare 5 complementary output polarity, see CC1NP.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 18        | CC5NE | RW         | Capture/compare 5 complementary output enabled, see CC1NE.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 17        | CC5P  | RW         | Capture/compare 5 output polarity, see CC1P.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 16        | CC5E  | RW         | Capture/compare 5 output enabled, see CC1E.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 15        | CC4NP | RW         | Capture/compare 4 complementary output polarity, see CC1NP.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 14        | CC4NE | RW         | Capture/compare 4 complementary output enabled, see CC1NE.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 13        | CC4P  | RW         | Capture/compare 4 output polarity, see CC1P.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 12        | CC4E  | RW         | Capture/compare 4 output enabled, see CC1E.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 11        | CC3NP | RW         | Capture/compare 3 complementary output polarity, see CC1NP.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 10        | CC3NE | EW         | Capture/compare 3 complementary output enabled, see CC1NE.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 9         | CC3P  | RW         | Capture/compare 3 output polarity, see CC1P.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 8         | CC3E  | RW         | Capture/compare 3 output enabled, see CC1E.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 7         | CC2NP | RW         | Capture/compare 2 complementary output polarity, see CC1NP.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 6         | CC2NE | RW         | Capture/compare 2 complementary output enabled, see CC1NE.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 5         | CC2P  | RW         | Capture/compare 2 output polarity, see CC1P.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 4         | CC2E  | RW         | Capture/compare 2 output enabled, see CC1E.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 3         | CC1NP | RW         | <p>Capture/compare 1 complementary output polarity</p> <p><b>CC1 channel configured as output:</b></p> <p>0: OC1N is valid at high level.</p> <p>1: OC1N is valid at low level.</p> <p><b>CC1 channel configured as input:</b></p> <p>This bit is used in conjunction with CC1P to define TI1FP1/TI2FP1 polarity. See CC1P description.</p> <p><i>Note: This bit becomes unwritable as soon as LOCK (LOCK bit in ATIM_BDTR register) level 2 or 3 is programmed and CC1S = "00" (channel configured as output).</i></p> <p><i>Note: This bit will preload on channels with complementary outputs. If the CCPC bit in the ATIM_CR2 register is set to 1, the CC1NP valid bit takes a new value from the preload bit only when a commutation event is generated.</i></p> |



| Bit field | Name  | Permission | Function description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|-----------|-------|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 2         | CC1NE | RW         | <p>Capture/compare 1 complementary output enabled</p> <p>0: OFF - OC1N is inactive. The OC1N level depends on the MOE, OSS1, OSSR, OIS1, OIS1N, and CC1E bits.</p> <p>1: ON - OC1N signal is output on the corresponding output pin, depending on the MOE, OSS1, OSSR, OIS1, OIS1N and CC1E bits.</p> <p><i>Note: This bit will preload on channels with complementary outputs. If the CCPC bit in the ATIM_CR2 register is set to 1, the CC1NE valid bit takes a new value from the preload bit only when a commutation event is generated.</i></p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 1         | CC1P  | RW         | <p>Capture/compare 1 output polarity</p> <p><b>CC1 channel configured as output:</b></p> <p>0: OC1 is valid at high level.</p> <p>1: OC1 is valid at low level.</p> <p><b>CC1 channel configured as input:</b></p> <p>The CC1NP/CC1P bits select the active polarity of TI1FP1 and TI2FP1 for trigger or capture operations.</p> <p>00: Non-inverted/rising edge trigger. The circuit acts on the rising edge of TlxFP1 (capture or trigger operation in reset mode, external clock mode or trigger mode), TlxFP1 is not inverted (trigger operation in gate mode or encoder mode).</p> <p>01: Inverted/falling edge trigger. The circuit acts on the falling edge of TlxFP1 (performs capture or trigger operation in reset mode, external clock mode, or trigger mode) and TlxFP1 inverts (performs trigger operation in gate mode or encoder mode).</p> <p>10: Reserved, this configuration is not used.</p> <p>11: Non-inverted/triggered on both rising and falling edges. The circuit acts on the rising and falling edges of TlxFP1 (capture or trigger operation in reset mode, external clock mode or trigger mode), TlxFP1 is not inverted (trigger operation in gated mode). This configuration should not be used in encoder mode.</p> <p><i>Note: This bit becomes unwritable as soon as LOCK (LOCK bit in the ATIM_BDTR register) level 2 or 3 is programmed.</i></p> <p><i>Note: This bit will preload on channels with complementary outputs. If the CCPC bit in the ATIM_CR2 register is set to 1, the CC1P valid bit takes a new value from the preload bit only when a commutation event is generated.</i></p> |



| Bit field | Name | Permission | Function description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|-----------|------|------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0         | CC1E | RW         | <p>Capture/compare 1 output enabled</p> <p><b>CC1 channel configured as output:</b></p> <p>0: OFF - OC1 is inactive. The OC1 level depends on the MOE, OSSI, OSSR, OIS1, OIS1N and CC1NE bits.</p> <p>1: ON - OC1 signal is output on the corresponding output pin, depending on the MOE, OSSI, OSSR, OIS1, OIS1 and CC1NE bits.</p> <p><b>CC1 channel configured as input:</b></p> <p>This bit determines whether the counter value can actually be captured in Input Capture/Compare Register 1 (ATIM_CCR1).</p> <p>0: Capture disabled.</p> <p>1: Capture enabled.</p> <p><i>Note: This bit will preload on channels with complementary outputs. If the CCPC bit in the ATIM_CR2 register is set to 1, the CC1E valid bit takes a new value from the preload bit only when a commutation event is generated.</i></p> <p><i>Note: This bit must be configured to 1 when using the input capture feature.</i></p> |



### 17.10.15 ATIM\_CNT Count register

Address offset: 0x24      Reset value: 0x0000 0000

| Bit field | Name   | Permission | Function description                                                                                                                                                                                                                                                                                                                        |
|-----------|--------|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31        | UIFCPY | RO         | According to the value of the UIFREMAP bit field in the ATIM_CR1 register, this bit field indicates different meanings: <ul style="list-style-type: none"> <li>UIFREMAP = 0, this bit field is reserved and read as 0.</li> <li>UIFREMAP = 1, this bit field indicates a read-only copy of the UIF bit of the ATIM_ISR register.</li> </ul> |
| 30:16     | RFU    | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                               |
| 15:0      | CNT    | RW         | Count value                                                                                                                                                                                                                                                                                                                                 |

### 17.10.16 ATIM\_PSC Prescaler register

Address offset: 0x28      Reset value: 0x0000 0000

| Bit field | Name | Permission | Function description                                                                                                                                                                                                                                                                                                                                        |
|-----------|------|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:16     | RFU  | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                                               |
| 15:0      | PSC  | RW         | Prescaler value<br>The counter clock frequency CK_CNT is equal to $f_{CK\_PSC}/(PSC[15:0]+1)$ .<br>The PSC contains the value to be loaded into the valid prescaler register each time an update event occurs, including when the counter is cleared by the UG bit in the ATIM_EGR register, or by the trigger controller when configured for "reset mode". |

### 17.10.17 ATIM\_ARR Auto reload register

Address offset: 0x2C      Reset value: 0x0000 FFFF

| Bit field | Name | Permission | Function description                                                                                                                                  |
|-----------|------|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:16     | RFU  | -          | Reserved bits, please keep the default value.                                                                                                         |
| 15:0      | ARR  | RW         | Auto reload value<br>ARR is the value to be loaded into the real auto reload register, the counter does not work when the auto reload value is empty. |



### 17.10.18 ATIM\_RCR Repeat count register

Address offset: 0x30      Reset value: 0x0000 0000

| Bit field | Name | Permission | Function description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|-----------|------|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:16     | RFU  | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 15:0      | REP  | RW         | <p>Repeat counter value</p> <p>These bits allow the user to set the update frequency of the compare register (i.e., the periodic transfer of data from the preload register to the active register) when the preload register is enabled, or the generation rate of update interrupts when update interrupts are enabled.</p> <p>Each time the down counter associated with REP_CNT counts to 0, an update event is generated and the counter restarts counting from the REP value. Because REP_CNT overloads the REP value only when a repeat update event U_RC is generated, whatever value is written to the ATIM_RCR register has no effect until the next repeat update event is generated.</p> <p>This means that (REP+1) in PWM mode is equivalent to:</p> <ul style="list-style-type: none"> <li>• The number of PWM cycles in edge-aligned mode.</li> <li>• The number of PWM half-cycles in center-aligned mode.</li> </ul> |

### 17.10.19 ATIM\_CCR1 Capture/compare register 1

Address offset: 0x34      Reset value: 0x0000 0000

| Bit field | Name | Permission | Function description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|-----------|------|------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:16     | RFU  | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 15:0      | CCR1 | RW         | <p>Capture/compare 1 value</p> <ul style="list-style-type: none"> <li>• If Channel CC1 is configured to output, CCR1 is the value to be loaded into the valid capture/compare 1 register (preload value). This value takes effect immediately if the preload function is not enabled via the OC1PE bit in the ATIM_CCMR1CMP register; otherwise, it takes effect only on an update event (copied to the capture/compare register 1 that is actually active).</li> <li>• The actual capture/compare register contains the value to be compared to counter ATIM_CNT and signaled on the OC1 output.</li> <li>• If channel CC1 is configured as input, CCR1 is the counter value when the last Input Capture 1 event (IC1) occurred. The ATIM_CCR1 register can only be read and cannot be programmed.</li> </ul> |



**17.10.20 ATIM\_CCR2 Capture/compare register 2**

Address offset: 0x38      Reset value: 0x0000 0000

| Bit field | Name | Permission | Function description                           |
|-----------|------|------------|------------------------------------------------|
| 31:16     | RFU  | -          | Reserved bits, please keep the default value.  |
| 15:0      | CCR2 | RW         | Capture/compare 2 value, see CCR1 description. |

**17.10.21 ATIM\_CCR3 Capture/compare register 3**

Address offset: 0x3C      Reset value: 0x0000 0000

| Bit field | Name | Permission | Function description                           |
|-----------|------|------------|------------------------------------------------|
| 31:16     | RFU  | -          | Reserved bits, please keep the default value.  |
| 15:0      | CCR3 | RW         | Capture/compare 3 value, see CCR1 description. |

**17.10.22 ATIM\_CCR4 Capture/compare register 4**

Address offset: 0x40      Reset value: 0x0000 0000

| Bit field | Name | Permission | Function description                           |
|-----------|------|------------|------------------------------------------------|
| 31:16     | RFU  | -          | Reserved bits, please keep the default value.  |
| 15:0      | CCR4 | RW         | Capture/compare 4 value, see CCR1 description. |

**17.10.23 ATIM\_CCR5 Capture/compare register 5**

Address offset: 0x48      Reset value: 0x0000 0000

| Bit field | Name  | Permission | Function description                                                                                                                        |
|-----------|-------|------------|---------------------------------------------------------------------------------------------------------------------------------------------|
| 31        | GC5C6 | RW         | CH6/CH5 combined waveform configuration, see GC5C1 description.                                                                             |
| 30        | GC5C5 | RW         | CH5/CH5 combined waveform configuration, see GC5C1 description.                                                                             |
| 29        | GC5C4 | RW         | CH4/CH5 combined waveform configuration, see GC5C1 description.                                                                             |
| 28        | GC5C3 | RW         | CH3/CH5 combined waveform configuration, see GC5C1 description.                                                                             |
| 27        | GC5C2 | RW         | CH2/CH5 combined waveform configuration, see GC5C1 description.                                                                             |
| 26        | GC5C1 | RW         | CH1/CH5 combined waveform configuration<br>0: OC1REFC is independent of OC5REF.<br>1: Logic AND of OC1REF and OC5REF from OC1REFC outputs . |
| 25:16     | RFU   | -          | Reserved bits, please keep the default value.                                                                                               |
| 15:0      | CCR5  | RW         | Capture/compare 5 value, see CCR1 description.                                                                                              |



### 17.10.24 ATIM\_CCR6 Capture/compare register 6

Address offset: 0x4C

Reset value: 0x0000 0000

| Bit field | Name  | Permission | Function description                                                                                                                                                    |
|-----------|-------|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31        | GC6C6 | RW         | CH6/CH6 combined waveform configuration, see GC6C1 description.                                                                                                         |
| 30        | GC6C5 | RW         | CH5/CH6 combined waveform configuration, see GC6C1 description.                                                                                                         |
| 29        | GC6C4 | RW         | CH4/CH6 combined waveform configuration, see GC6C1 description.                                                                                                         |
| 28        | GC6C3 | RW         | CH3/CH6 combined waveform configuration, see GC6C1 description.                                                                                                         |
| 27        | GC6C2 | RW         | CH2/CH6 combined waveform configuration, see GC6C1 description.                                                                                                         |
| 26        | GC6C1 | RW         | CH1/CH6 combined waveform configuration<br>0: OC1REFC is independent of OC6REF.<br>1: OC1REFC output OC1REF when counting up; OC1REFC output OC6REF when counting down. |
| 25:16     | RFU   | -          | Reserved bits, please keep the default value.                                                                                                                           |
| 15:0      | CCR6  | RW         | Capture/compare 6 value, see CCR1 descriptionv.                                                                                                                         |





## 17.10.25 ATIM\_BDTR Brake and deadband register

Address offset: 0x44

Reset value: 0x0000 0000

| Bit field | Name | Permission | Function description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|-----------|------|------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:26     | RFU  | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 25        | BK2P | RW         | Brake 2 polarity<br>0: Brake input BRK2 is valid at low level.<br>1: Brake input BRK2 is valid at high level.<br><i>Attention: This bit cannot be modified as long as LOCK (LOCK bit in the ATIM_BDTR register) level 1 is programmed.</i><br><i>Attention: Any write to this bit requires a delay of 1 APB clock cycle to take effect.</i>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 24        | BK2E | RW         | Brake 2 enabled<br>0: Disable brake input BRK2.<br>1: Enable brake input BRK2.<br><i>Attention: BRK2 must only be used when OSSR = OSS1 = 1.</i><br><i>Attention: This bit cannot be modified as long as LOCK (LOCK bit in the ATIM_BDTR register) level 1 is programmed.</i><br><i>Attention: Any write to this bit requires a delay of 1 APB clock cycle to take effect.</i>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 23:20     | BK2F | RW         | Brake 2 filter, this bit field defines the sampling frequency of the BRK2 input and the digital filter bandwidth applicable to the BRK2. The digital filter consists of event counters, and every N consecutive events are regarded as a valid output edge.<br>0000: No filter, BRK2 operates asynchronously 1000: $f_{\text{SAMPLING}} = f_{\text{DTS}} / 8$ , N = 6<br>0001: $f_{\text{SAMPLING}} = f_{\text{PCLK}}$ , N = 2 1001: $f_{\text{SAMPLING}} = f_{\text{DTS}} / 8$ , N = 8<br>0010: $f_{\text{SAMPLING}} = f_{\text{PCLK}}$ , N = 4 1010: $f_{\text{SAMPLING}} = f_{\text{DTS}} / 16$ , N = 5<br>0011: $f_{\text{SAMPLING}} = f_{\text{PCLK}}$ , N = 8 1011: $f_{\text{SAMPLING}} = f_{\text{DTS}} / 16$ , N = 6<br>0100: $f_{\text{SAMPLING}} = f_{\text{DTS}} / 2$ , N = 6 1100: $f_{\text{SAMPLING}} = f_{\text{DTS}} / 16$ , N = 8<br>0101: $f_{\text{SAMPLING}} = f_{\text{DTS}} / 2$ , N = 8 1101: $f_{\text{SAMPLING}} = f_{\text{DTS}} / 32$ , N = 5<br>0110: $f_{\text{SAMPLING}} = f_{\text{DTS}} / 4$ , N = 6 1110: $f_{\text{SAMPLING}} = f_{\text{DTS}} / 32$ , N = 6<br>0111: $f_{\text{SAMPLING}} = f_{\text{DTS}} / 4$ , N = 8 1111: $f_{\text{SAMPLING}} = f_{\text{DTS}} / 32$ , N = 8<br><i>Attention: This bit cannot be modified as long as LOCK (LOCK bit in the ATIM_BDTR register) level 1 is programmed.</i> |
| 19:16     | BKF  | RW         | Brake filter, this bit field defines the sampling frequency of the BRK input and the digital filter bandwidth applicable to the BRK. The digital filter consists of event counters, and every N consecutive events are regarded as a valid output edge.<br>0000: No filter, BRK operates asynchronously. 1000: $f_{\text{SAMPLING}} = f_{\text{DTS}} / 8$ , N = 6<br>0001: $f_{\text{SAMPLING}} = f_{\text{PCLK}}$ , N = 2 1001: $f_{\text{SAMPLING}} = f_{\text{DTS}} / 8$ , N = 8<br>0010: $f_{\text{SAMPLING}} = f_{\text{PCLK}}$ , N = 4 1010: $f_{\text{SAMPLING}} = f_{\text{DTS}} / 16$ , N = 5<br>0011: $f_{\text{SAMPLING}} = f_{\text{PCLK}}$ , N = 8 1011: $f_{\text{SAMPLING}} = f_{\text{DTS}} / 16$ , N = 6<br>0100: $f_{\text{SAMPLING}} = f_{\text{DTS}} / 2$ , N = 6 1100: $f_{\text{SAMPLING}} = f_{\text{DTS}} / 16$ , N = 8<br>0101: $f_{\text{SAMPLING}} = f_{\text{DTS}} / 2$ , N = 8 1101: $f_{\text{SAMPLING}} = f_{\text{DTS}} / 32$ , N = 5<br>0110: $f_{\text{SAMPLING}} = f_{\text{DTS}} / 4$ , N = 6 1110: $f_{\text{SAMPLING}} = f_{\text{DTS}} / 32$ , N = 6<br>0111: $f_{\text{SAMPLING}} = f_{\text{DTS}} / 4$ , N = 8 1111: $f_{\text{SAMPLING}} = f_{\text{DTS}} / 32$ , N = 8<br><i>Attention: This bit cannot be modified as long as LOCK (LOCK bit in the ATIM_BDTR register) level 1 is programmed.</i>     |



| Bit field | Name | Permission | Function description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|-----------|------|------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 15        | MOE  | RW         | <p>Master output enabled</p> <p>This bit is cleared asynchronously by hardware whenever the brake input (BRK or BRK2) is valid. This bit is set to 1 by software or can be set automatically based on the state of the AOE bit. This bit is only valid for channels configured as outputs.</p> <p>0: Responding to braking events (2). OC and OCN outputs disabled. Responding to a braking event or writing a 0 to MOE: OC and OCN outputs are disabled or forced to idle, depending on the OSSR bit.</p> <p>1: The OC and OCN outputs are enabled if their respective enable bits (CCyE and CCyNE bits in the ATIM_CCER register) are both set to 1.</p>   |
| 14        | AOE  | RW         | <p>Auto output enabled</p> <p>0: MOE can only be set to 1 by software</p> <p>1: MOE can be set to 1 by software or automatically when the next update event occurs (this bit is invalid if the brake inputs BRK and BRK2 are valid).</p> <p><i>Attention: This bit cannot be modified as long as LOCK (LOCK bit in the ATIM_BDTR register) level 1 is programmed.</i></p>                                                                                                                                                                                                                                                                                    |
| 13        | BKP  | RW         | <p>Brake polarity</p> <p>0: Brake input BRK is valid at low level.</p> <p>1: Brake input BRK is valid at high level.</p> <p><i>Attention: This bit cannot be modified as long as LOCK (LOCK bit in the ATIM_BDTR register) level 1 is programmed.</i></p> <p><i>Attention: Any write to this bit requires a delay of 1 APB clock cycle to take effect.</i></p>                                                                                                                                                                                                                                                                                               |
| 12        | BKE  | RW         | <p>Brake enabled</p> <p>This bit enables complete brake protection (including all internal sources and corresponding BKIN sources).</p> <p>0: Disable brake function.</p> <p>1: Enable brake function.</p> <p><i>Attention: This bit cannot be modified as long as LOCK (LOCK bit in the ATIM_BDTR register) level 1 is programmed.</i></p> <p><i>Attention: Any write to this bit requires a delay of 1 APB clock cycle to take effect.</i></p>                                                                                                                                                                                                             |
| 11        | OSSR | RW         | <p>Off state selection in operating mode</p> <p>This bit works on channels configured in output mode with complementary outputs when MOE = 1. If there is no complementary output in the timer, then there is no OSSR.</p> <p>0: When inactive, OC/OCN output is disabled (timer releases output control, taken over by GPIO logic forcing high impedance state).</p> <p>1: In the inactive state, once CCyE = 1 or CCyNE = 1, enable OC/OCN output and set it to inactive level (output is still controlled by timer).</p> <p><i>Attention: This bit cannot be modified as long as LOCK (LOCK bit in the ATIM_BDTR register) level 2 is programmed.</i></p> |



| Bit field | Name | Permission | Function description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|-----------|------|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 10        | OSSI | RW         | <p>Off state selection in idle mode</p> <p>When MOE = 0 due to a braking event or software write operation, this bit acts on channels configured as outputs.</p> <p>0: When inactive, OC/OCN output is disabled (timer releases output control, taken over by GPIO logic forcing high impedance state).</p> <p>1: In the inactive state, force the OC/OCN output to its inactive level first and then to its idle level after the dead time. The timer always controls the output.</p> <p><b>Attention: This bit cannot be modified as long as LOCK (LOCK bit in the ATIM_BDTR register) level 2 is programmed.</b></p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| 9:8       | LOCK | RW         | <p>Locked configuration</p> <p>These bits are used to provide write protection against software errors.</p> <p>00: Off lock - Does not write protect any bits.</p> <p>01: Lock level 1, where no write operation can be performed to the OISy and OISyN bits in the ATIM_CR2 register, DTGF, DATE and DTPE bits in the ATIM_DTR2 register, BK2P, BK2E, BK2F[3:0], BKF[3:0], AOE, BKP, BKE, LOCK and DTG bits in the ATIM_BDTR register, all bit fields in the ATIM_AF1 register, and all bit fields in the ATIM_AF2 register.</p> <p>10: Lock level 2, where no write operation can be performed to the applicable bits in lock level 1, CC polarity bits (CCyP/CCyNP bits in the ATIM_CCER register, provided the associated channel is configured as output via the CCyS bit), and OSSR and OSSI bits in the ATIM_BDTR register.</p> <p>11: Lock level 3, where no write operation can be performed to the applicable bits in lock level 2, CC control bits (OCyM and OCyPE bits in the ATIM_CCMRxCMP register, as long as the associated channel is configured as output via the CCyS bit).</p> <p><b>Attention: Only one write to the LOCK bit can be performed after reset. A write to the ATIM_BDTR register freezes its contents until the next reset.</b></p> |
| 7:0       | DTG  | RW         | <p>Configure deadband generator</p> <p>This bit field defines the dead time duration inserted between complementary outputs. DT corresponds to this duration.</p> <p><math>DTG[7:5] = 0xx \Rightarrow DT = DTG[7:0] \times t_{dtg}</math>, where <math>t_{dtg} = t_{dTS}</math>.</p> <p><math>DTG[7:5] = 10x \Rightarrow DT = (64 + DTG[5:0]) \times t_{dtg}</math>, where <math>t_{dtg} = 2 \times t_{dTS}</math>.</p> <p><math>DTG[7:5] = 110 \Rightarrow DT = (32 + DTG[4:0]) \times t_{dtg}</math>, where <math>t_{dtg} = 8 \times t_{dTS}</math>.</p> <p><math>DTG[7:5] = 111 \Rightarrow DT = (32 + DTG[4:0]) \times t_{dtg}</math>, where <math>t_{dtg} = 16 \times t_{dTS}</math>.</p> <p>Example: If <math>t_{dTS} = 125ns</math> (8MHz), the possible deadband values are:</p> <ul style="list-style-type: none"> <li>0 to 15875ns (in 125ns steps).</li> <li>16<math>\mu</math>s to 31750ns (in 250ns steps).</li> <li>32<math>\mu</math>s to 63<math>\mu</math>s (in 1<math>\mu</math>s steps).</li> <li>64<math>\mu</math>s to 126<math>\mu</math>s (in 2 <math>\mu</math>s steps).</li> </ul> <p><b>Attention: This bit field cannot be modified as long as LOCK (LOCK bit in the ATIM_BDTR register) level 1, 2, or 3 is programmed.</b></p>           |

**Attention:** Since the write to the lock bits BK2P, BK2E, BK2F[3:0], BKF[3:0], AOE, BKP, BKE, OSSI, OSSR, and DTG[7:0] can be configured according to LOCK, these bits must be configured the first time a write access is performed to the ATIM\_BDTR register.



### 17.10.26 ATIM\_DTR2 Dead time register 2

Address offset: 0x54

Reset value: 0x0000 0000

| Bit field | Name | Permission | Function description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|-----------|------|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:18     | RFU  | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 17        | DTPE | RW         | Dead time value preload enabled<br>0: Disable dead time value preload.<br>1: Enable dead time value preload.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 16        | DTAE | RW         | Dead time asymmetry enabled<br>0: The dead time of the rising and falling edges is the same, defined by DTG[7:0].<br>1: Dead time of rising edge is defined by DTG[7:0] and dead time of falling edge is defined by DTGF[7:0].<br><i>Attention: This bit field cannot be modified as long as LOCK (LOCK bit in the ATIM_BDTR register) level 1, 2, or 3 is programmed.</i>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| 15:8      | RFU  | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 7:0       | DTGF | RW         | Dead time generator for falling edge<br>This bit field defines the dead time duration inserted between falling edges of complementary outputs.<br>DTGF[7:5] = 0xx => DTF = DTGF[7:0] × t <sub>dtg</sub> , where t <sub>dtg</sub> = t <sub>DTS</sub> .<br>DTGF[7:5] = 10x => DTF = (64+DTG[5:0]) × t <sub>dtg</sub> , where t <sub>dtg</sub> = 2 × t <sub>DTS</sub> .<br>DTGF[7:5] = 110 => DTF = (32+DTG[4:0]) × t <sub>dtg</sub> , where t <sub>dtg</sub> = 8 × t <sub>DTS</sub> .<br>DTGF[7:5] = 111 => DTF = (32+DTG[4:0]) × t <sub>dtg</sub> , where t <sub>dtg</sub> = 16 × t <sub>DTS</sub> .<br>Example: If t <sub>DTS</sub> = 125ns (8MHz), the possible deadband values are: <ul style="list-style-type: none"> <li>• 0 to 15875ns (in 125ns steps)</li> <li>• 16μs to 31750ns (in 250ns steps)</li> <li>• 32 μs to 63μs (in 1μs steps)</li> <li>• 64μs to 126μs (in 2 μs steps)</li> </ul> <i>Attention: This bit field cannot be modified as long as LOCK (LOCK bit in the ATIM_BDTR register) level 1, 2, or 3 is programmed.</i> |



### 17.10.27 ATIM\_ECR Encoder control register

Address offset: 0x58

Reset value: 0x0000 0000

| Bit field | Name | Permission | Function description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|-----------|------|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:8      | RFU  | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 7:6       | IPOS | RW         | <p>Coding index position</p> <p>In quadrature encoder mode (SMS[3:0] = 0001, 0010, 0011, 1110, 1111), this bit indicates in which AB input configuration the index event resets the counter.</p> <p>00: When AB = 00, the index resets the counter.</p> <p>01: When AB = 01, the index resets the counter.</p> <p>10: When AB = 10, the index resets the counter.</p> <p>11: When AB = 11, the index resets the counter.</p> <p>In directional clock encoder mode or clock plus direction encoder mode (SMS[3:0] = 1010, 1011, 1100, 1101), these bits indicate at which level the index event resets the counter. In bidirectional clock encoder mode, this applies to both clock inputs.</p> <p>x0: when the clock signal is low, the index resets the counter.</p> <p>x1: When the clock signal is high, the index resets the counter.</p> |
| 5         | FIDX | RW         | <p>This bit indicates whether only the first index is considered</p> <p>0: Index is always active.</p> <p>1: Only the first index resets the counter.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 4:3       | RFU  | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 2:1       | IDIR | RW         | <p>Index direction</p> <p>This bit field indicates in which count direction the index resets the counter</p> <p>00: Reset counter in any counting direction.</p> <p>01: Reset counter only when counting up.</p> <p>10: Reset counter only when counting down.</p> <p>11: Reserved.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 0         | IE   | RW         | <p>Index enabled</p> <p>0: Index disabled.</p> <p>1: Index enabled.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |



### 17.10.28 ATIM\_TISEL1 TI input selection register 1

Address offset: 0x5C

Reset value: 0x0000 0000

| Bit field | Name   | Permission | Function description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|-----------|--------|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:28     | RFU    | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 27:24     | TI4SEL | RW         | TI4 channel input capture signal source selection<br>0000: ATIM_CH4                      1000: BTIM1_Trigo<br>0001: VC3_OUT                      1001: BTIM2_Trigo<br>0010: VC4_OUT                      1010: BTIM3_Trigo<br>0011: UART1_TXD                   1011: GTIM1_Trigo<br>0100: UART2_TXD                   1100: GTIM2_Trigo<br>0101: UART3_TXD                   1101: GTIM3_Trigo<br>0110: MCO_OUT                      1110: GTIM4_Trigo<br>0111: HALLTIM_Trigo                Other: Reserved |
| 23:20     | RFU    | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 19:16     | TI3SEL | RW         | TI3 channel input capture signal source selection<br>0000: ATIM_CH3                      1000: BTIM1_Trigo<br>0001: VC1_OUT                      1001: BTIM2_Trigo<br>0010: VC2_OUT                      1010: BTIM3_Trigo<br>0011: UART1_RXD                   1011: GTIM1_Trigo<br>0100: UART2_RXD                   1100: GTIM2_Trigo<br>0101: UART3_RXD                   1101: GTIM3_Trigo<br>0110: LPTIM_PWM                   1110: GTIM4_Trigo<br>0111: HALLTIM_Trigo                Other: Reserved  |
| 15:12     | RFU    | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 11:8      | TI2SEL | RW         | TI2 channel input capture signal source selection<br>0000: ATIM_CH2                      1000: BTIM1_Trigo<br>0001: VC3_OUT                      1001: BTIM2_Trigo<br>0010: VC4_OUT                      1010: BTIM3_Trigo<br>0011: UART1_TXD                   1011: GTIM1_Trigo<br>0100: UART2_TXD                   1100: GTIM2_Trigo<br>0101: UART3_TXD                   1101: GTIM3_Trigo<br>0110: RTC_OUT                      1110: GTIM4_Trigo<br>0111: HALLTIM_Trigo                Other: Reserved |
| 7:4       | RFU    | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 3:0       | TI1SEL | RW         | TI1 channel input capture signal source selection<br>0000: ATIM_CH1                      1000: BTIM1_Trigo<br>0001: VC1_OUT                      1001: BTIM2_Trigo<br>0010: VC2_OUT                      1010: BTIM3_Trigo<br>0011: UART1_RXD                   1011: GTIM1_Trigo<br>0100: UART2_RXD                   1100: GTIM2_Trigo<br>0101: UART3_RXD                   1101: GTIM3_Trigo<br>0110: MCO_OUT                      1110: GTIM4_Trigo<br>0111: HALLTIM_Trigo                Other: Reserved |



### 17.10.29 ATIM\_TISEL2 TI input selection register 2

Address offset: 0x6C

Reset value: 0x0000 0000

| Bit field | Name   | Permission | Function description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|-----------|--------|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:12     | RFU    | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 11:8      | TI6SEL | RW         | TI6 channel input capture signal source selection<br>0000: ATIM_CH6                      1000: BTIM1_Trgo<br>0001: VC3_OUT                      1001: BTIM2_Trgo<br>0010: VC4_OUT                      1010: BTIM3_Trgo<br>0011: UART1_TXD                   1011: GTIM1_Trgo<br>0100: UART2_TXD                   1100: GTIM2_Trgo<br>0101: UART3_TXD                   1101: GTIM3_Trgo<br>0110: LPTIM_PWM                   1110: GTIM4_Trgo<br>0111: HALLTIM_Trgo                Other: Reserved  |
| 7:4       | RFU    | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 3:0       | TI5SEL | RW         | TI5 channel input capture signal source selection<br>0000: ATIM_CH5                      1000: BTIM1_Trgo<br>0001: VC1_OUT                      1001: BTIM2_Trgo<br>0010: VC2_OUT                      1010: BTIM3_Trgo<br>0011: UART1_RXD                   1011: GTIM1_Trgo<br>0100: UART2_RXD                   1100: GTIM2_Trgo<br>0101: UART3_RXD                   1101: GTIM3_Trgo<br>0110: RTC_OUT                      1110: GTIM4_Trgo<br>0111: HALLTIM_Trgo                Other: Reserved |



### 17.10.30 ATIM\_AF1 Multiplexing function option register 1

Address offset: 0x60

Reset value: 0x0000 0001

| Bit field | Name   | Permission | Function description                          |
|-----------|--------|------------|-----------------------------------------------|
| 31:20     | RFU    | -          | Reserved bits, please keep the default value. |
| 19:16     | ETRSEL | RW         | ETR source selection<br>0000: ATIM_ETR pin    |





| Bit field | Name   | Permission | Function description                                                                                                                                                                                                                                                                                                                                            |
|-----------|--------|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 9         | BKINP  | RW         | BRK BKIN input polarity<br>This bit selects the BKIN multiplex function input level sensitivity and must be programmed with the BKP polarity bit.<br>0: BKIN input is valid at high level.<br>1: BKIN input is valid at low level.<br><i>Attention: This bit cannot be modified as long as LOCK (LOCK bit in the ATIM_BDTR register) level 1 is programmed.</i> |
| 8:5       | RFU    | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                                                   |
| 4         | BKVC4E | RW         | BRK VC4 enabled<br>This bit enables VC4 input by the timer BRK, and the VC4 output performs an "OR" operation with other BRK sources.<br>0: Disable VC4 input.<br>1: Enable VC4 input.<br><i>Attention: This bit cannot be modified as long as LOCK (LOCK bit in the ATIM_BDTR register) level 1 is programmed.</i>                                             |
| 3         | BKVC3E | RW         | BRK VC3 enabled<br>This bit enables VC3 input by the timer BRK, and the VC3 output performs an "OR" operation with other BRK sources.<br>0: Disable VC3 input.<br>1: Enable VC3 input.<br><i>Attention: This bit cannot be modified as long as LOCK (LOCK bit in the ATIM_BDTR register) level 1 is programmed.</i>                                             |
| 2         | BKVC2E | RW         | BRK VC2 enabled<br>This bit enables VC2 input by the timer BRK, and the VC2 output performs an "OR" operation with other BRK sources.<br>0: Disable VC2 input.<br>1: Enable VC2 input.<br><i>Attention: This bit cannot be modified as long as LOCK (LOCK bit in the ATIM_BDTR register) level 1 is programmed.</i>                                             |
| 1         | BKVC1E | RW         | BRK VC1 enabled<br>This bit enables VC1 input by the timer BRK, and the VC1 output performs an "OR" operation with other BRK sources.<br>0: Disable VC1 input.<br>1: Enable VC1 input.<br><i>Attention: This bit cannot be modified as long as LOCK (LOCK bit in the ATIM_BDTR register) level 1 is programmed.</i>                                             |
| 0         | BKINE  | RW         | BRK BKIN input enabled<br>This bit enables the BKIN multiplexing function input by the timer BRK, and the BKIN input performs an "OR" operation with other BRK sources.<br>0: Disable BKIN input.<br>1: Enable BKIN input.<br><i>Attention: This bit cannot be modified as long as LOCK (LOCK bit in the ATIM_BDTR register) level 1 is programmed.</i>         |



### 17.10.31 ATIM\_AF2 Multiplexing function option register 2

Address offset: 0x64      Reset value: 0x0000 0001

| Bit field | Name    | Permission | Function description                                                                                                                                                                                                                                                                                                                      |
|-----------|---------|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:20     | RFU     | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                             |
| 19:16     | OCRSEL  | RW         | OCREF_CLR source selection<br>0000: VC1_OUT                      0111: ADC2_AWD<br>0001: VC2_OUT                      1000: LVD_OUT<br>0010: VC3_OUT                      1010: ETRF<br>0011: VC4_OUT                      Other: Reserved<br>0110: ADC1_AWD                                                                              |
| 15:14     | RFU     | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                             |
| 13        | BK2VC4P | RW         | BRK2 VC4 input polarity<br>This bit selects the VC4 input level sensitivity and must be programmed with the BK2P polarity bit.<br>0: VC4 input is valid at low level.<br>1: VC4 input is valid at highlevel.<br><i>Attention: This bit cannot be modified as long as LOCK (LOCK bit in the ATIM_BDTR register) level 1 is programmed.</i> |
| 12        | BK2VC3P | RW         | BRK2 VC3 input polarity<br>This bit selects the VC3 input level sensitivity and must be programmed with the BK2P polarity bit.<br>0: VC3 input is valid at low level.<br>1: VC3 input is valid at highlevel.<br><i>Attention: This bit cannot be modified as long as LOCK (LOCK bit in the ATIM_BDTR register) level 1 is programmed.</i> |
| 11        | BK2VC2P | RW         | BRK2 VC2 input polarity<br>This bit selects the VC2 input level sensitivity and must be programmed with the BK2P polarity bit.<br>0: VC2 input is valid at low level.<br>1: VC2 input is valid at highlevel.<br><i>Attention: This bit cannot be modified as long as LOCK (LOCK bit in the ATIM_BDTR register) level 1 is programmed.</i> |
| 10        | BK2VC1P | RW         | BRK2 VC1 input polarity<br>This bit selects the VC1 input level sensitivity and must be programmed with the BK2P polarity bit.<br>0: VC1 input is valid at low level.<br>1: VC1 input is valid at highlevel.<br><i>Attention: This bit cannot be modified as long as LOCK (LOCK bit in the ATIM_BDTR register) level 1 is programmed.</i> |



| Bit field | Name    | Permission | Function description                                                                                                                                                                                                                                                                                                                                                  |
|-----------|---------|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 9         | BK2INP  | RW         | BRK2 BKIN2 input polarity<br>This bit selects the BKIN2 multiplex function input level sensitivity and must be programmed with the BK2P polarity bit.<br>0: BKIN2 input is valid at low level.<br>1: BKIN2 input is valid at high level.<br><i>Attention: This bit cannot be modified as long as LOCK (LOCK bit in the ATIM_BDTR register) level 1 is programmed.</i> |
| 8:5       | RFU     | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                                                         |
| 4         | BK2VC4E | RW         | BRK2 VC4 enabled<br>This bit enables VC4 input by the timer BRK2, and the VC4 output performs an "OR" operation with other BRK2 sources.<br>0: Disable VC4 input.<br>1: Enable VC4 input.<br><i>Attention: This bit cannot be modified as long as LOCK (LOCK bit in the ATIM_BDTR register) level 1 is programmed.</i>                                                |
| 3         | BK2VC3E | RW         | BRK2 VC3 enabled<br>This bit enables VC3 input by the timer BRK2, and the VC3 output performs an "OR" operation with other BRK2 sources.<br>0: Disable VC3 input.<br>1: Enable VC3 input.<br><i>Attention: This bit cannot be modified as long as LOCK (LOCK bit in the ATIM_BDTR register) level 1 is programmed.</i>                                                |
| 2         | BK2VC2E | RW         | BRK2 VC2 enabled<br>This bit enables VC2 input by the timer BRK2, and the VC2 output performs an "OR" operation with other BRK2 sources.<br>0: Disable VC2 input.<br>1: Enable VC2 input.<br><i>Attention: This bit cannot be modified as long as LOCK (LOCK bit in the ATIM_BDTR register) level 1 is programmed.</i>                                                |
| 1         | BK2VC1E | RW         | BRK2 VC1 enabled<br>This bit enables VC1 input by the timer BRK2, and the VC1 output performs an "OR" operation with other BRK2 sources.<br>0: Disable VC1 input.<br>1: Enable VC1 input.<br><i>Attention: This bit cannot be modified as long as LOCK (LOCK bit in the ATIM_BDTR register) level 1 is programmed.</i>                                                |
| 0         | BK2INE  | RW         | BRK2 BKIN2 input enabled<br>This bit enables the BKIN2 multiplexing function input by the timer BRK2, and the BKIN2 input performs an "OR" operation with other BRK2 sources.<br>0: Disable BKIN2 input.<br>1: Enable BKIN2 input.<br><i>Attention: This bit cannot be modified as long as LOCK (LOCK bit in the ATIM_BDTR register) level 1 is programmed.</i>       |



## 18 Hall sensor dedicated timer (HALLTIM)

### 18.1 Overview

CW32L012 integrates 1 24-bit up-counter Hall sensor-specific timer (HALLTIM) internally. It is a circuit integrated with a Hall effect sensor, used to detect magnetic fields and convert them into electrical signals. It is commonly used in applications such as motor control and position detection. It has 3 independent capture channels and 1 output, capable of outperforming PWM, output comparison, HALL signal variation pulses, and 3-channel filtered signal XOR, etc. In addition, HALLTIM also supports DMA requests.



## 18.2 Main features

- 24-bit incremental auto-reload counter.
- Programmable prescaler supporting 1, 2, 4, 8 division.
- 3 independent input capture channels.
- PWM waveform output through the HALLTIM\_OUT pin.
- Trigger output signal (TRGO) for synchronization between peripherals:
  - Overflow pulse.
  - PWM signal.
  - CCR matching pulse.
  - HALL signal variation pulse.
  - CH1/CH2/CH3 filtered waveform.
  - 3-channel filtered signal XOR.
- Supports direct memory access (DMA).

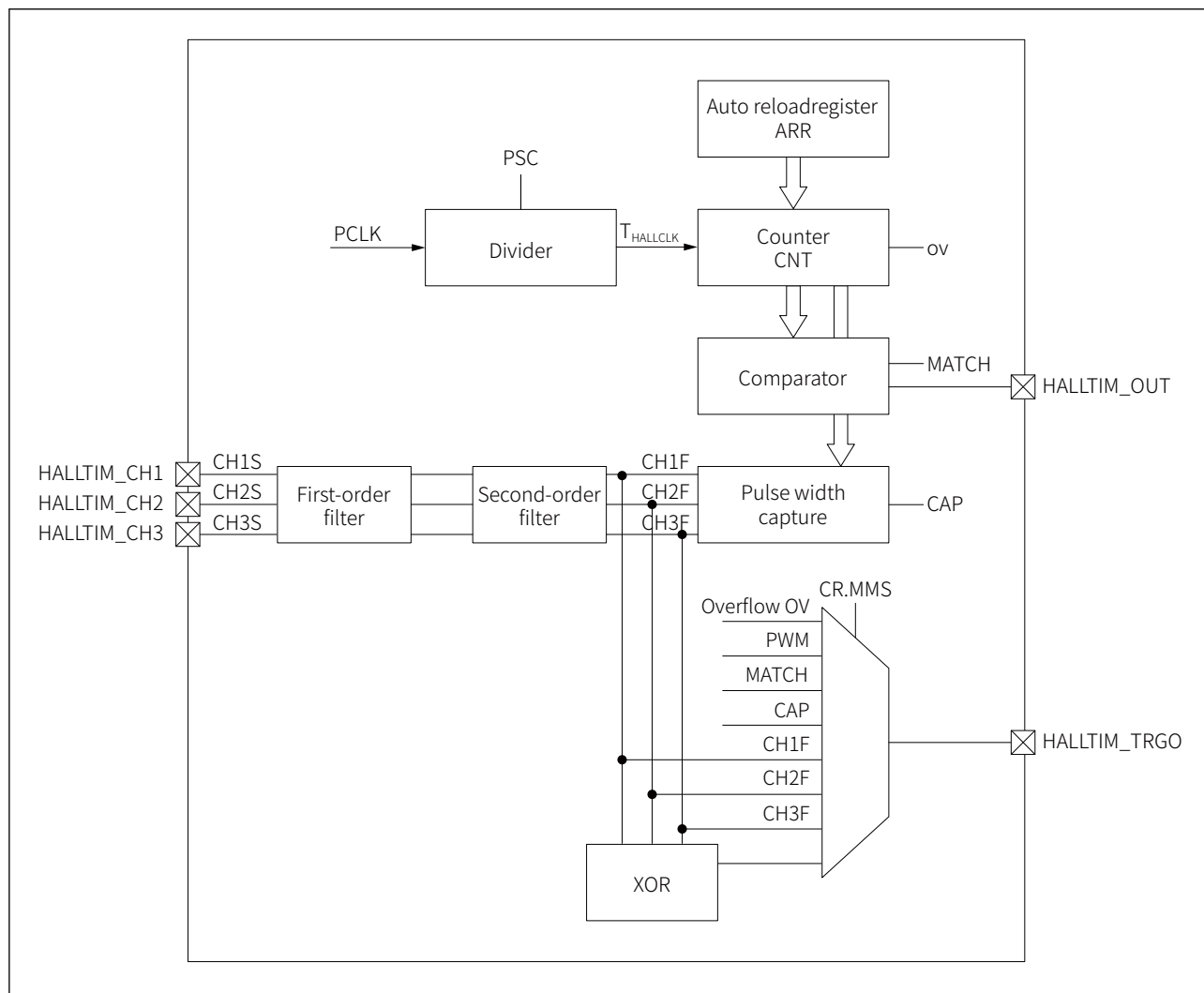


## 18.3 Functional description

### 18.3.1 Functional block diagram

The functional block diagram of HALLTIM is shown in the following figure:

Figure 18-1 HALLTIM functional block diagram



### 18.3.1.1 Operation clock

The counting clock source of HALLTIM is PCLK. By setting the DIV bit field of CR register, PCLK can be divided into 1, 2, 4 and 8 frequencies to obtain the counting clock  $T_{HALLCLK}$  of HALLTIM.

Table 18-1 Frequency division factor configuration

| DIV | Frequency division factor |
|-----|---------------------------|
| 00  | 1                         |
| 01  | 2                         |
| 10  | 4                         |
| 11  | 8                         |

### 18.3.1.2 Filter unit

HALLTIM provides 3 input channels, each with two-order filtering. The first-order filter is a simple low-pass filter that filters out some high-frequency interference. It can be enabled or disabled by the FLT1EN bit field in the CR register. The second-order filter is a digital filter that can be configured by the FLT2LEN bit field in the CR register to filter out pulses with pulse widths shorter than the set duration.



## 18.3.2 Operation description

### 18.3.2.1 Overflow signal

The HALLTIM counter features an auto-reload function, with the reload value configurable by the ARR register. When the counter value CNT, driven by the  $T_{HALLCLK}$  clock, counts up to the value set in the ARR register, an overflow signal OV is generated with a width equal to one  $T_{HALLCLK}$  clock cycle. The OVF bit field in the ISR register of the interrupt status register is set, and CNT is reset to zero before restarting the count.

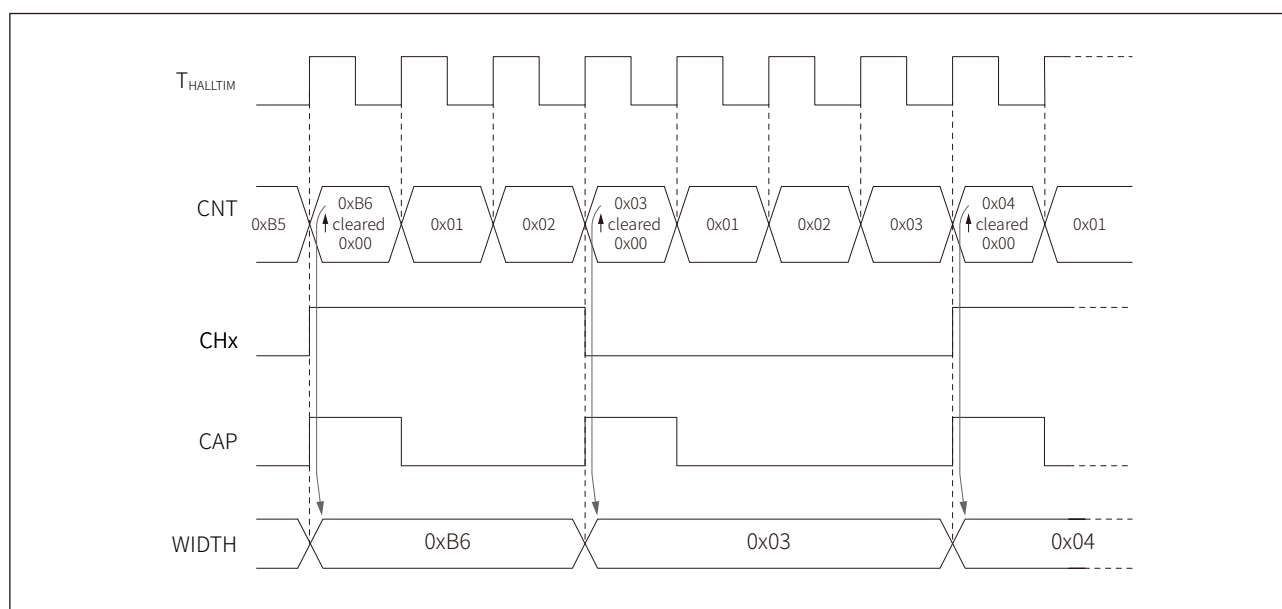
### 18.3.2.2 PWM waveform output

HALLTIM contains a comparator that can be used to generate PWM waveforms, with the generated waveform output through the HALLTIM\_OUT pin. When the value in the counter register CNT is less than the value in the compare register CCR, the comparator outputs a high level; otherwise, it outputs a low level. When CNT equals CCR, the comparator generates a MATCH comparison pulse signal with a width equal to one  $T_{HALLCLK}$  cycle, simultaneously setting the MATCHF bit in the interrupt status register (ISR).

### 18.3.2.3 Input capture

The HALLTIM contains an edge capture circuit. When any of HALLTIM's three input channels detects a level change, it triggers the capture circuit to store the current CNT value in the pulse width register (WIDTH). At the same time, the hardware clears CNT and generates a CAP input capture pulse signal with a duration equal to one  $T_{HALLCLK}$  clock cycle, and the CAPF bit field in the ISR register of the interrupt status register is set. As shown in the following diagram:

Figure 18-2 Input capture mode block diagram





## 18.4 HALLTIM interrupt

HALLTIM supports 3 interrupt sources. When a HALLTIM interrupt event occurs, the interrupt flag bit is set by hardware. If the corresponding interrupt enable control bit is set, an interrupt request is generated.

Within the user HALLTIM interrupt service routine, the relevant HALLTIM interrupt flag should be checked to perform the appropriate handling. Before exiting the interrupt service routine, the interrupt flag must be cleared to prevent repeated entry into the interrupt service routine.

The flag bits, interrupt enable bits, interrupt flag clear bits, or clearing methods for each HALLTIM interrupt source are shown in the following table:

Table 18-2 HALLTIM interrupt control

| Interrupt event    | Interrupt flag bit | Interrupt enable bit | Interrupt flag clear  |
|--------------------|--------------------|----------------------|-----------------------|
| Input capture flag | ISR.CAPF           | DIER.CAPF            | Write 0 to ICR.CAPF   |
| Overflow flag      | ISR.OVF            | DIER.OVF             | Write 0 to ICR.OVF    |
| Match flag         | ISR.MATCHF         | DIER.MATCHF          | Write 0 to ICR.MATCHF |



## 18.5 Trigger output signal TRGO

HALLTIM provides multiple trigger output options via the HALLTIM\_TRGO pin. These options can be selected through the MMS bit field in the CR register. The correspondence between the output signal and the register bit field value is shown in the following table:

Table 18-3 Trigger output signal source

| CR.MMS | Trigger signal source          |
|--------|--------------------------------|
| 000    | Overflow pulse.                |
| 001    | PWM signal.                    |
| 010    | CCR match pulse.               |
| 011    | HALL signal change pulse.      |
| 100    | Filtered CH3 waveform.         |
| 101    | Filtered CH2 waveform.         |
| 110    | Filtered CH1 waveform.         |
| 111    | 3-channel filtered signal XOR. |



## 18.6 Direct memory access (DMA)

HALLTIM supports generating DMA requests when input capture, counter overflow, counter matching, and software-triggered events occur. After enabling the DMA function, it is possible to write data from other locations to the timer or read data from the timer to other locations. For example, it can be applied to the automatic transfer of data after data capture and to change the period or pulse width of the PWM output.

The relevant bit field of the DMA/ interrupt enable register HALLTIM\_DIER is used to enable DMA requests. Users need to set the corresponding DMA request enable bit in the HALLTIM\_DIER register to 1, and set the HARDSRC bit field of the DMA trigger source control register DMA\_TRIGy to HALLTIM.



## 18.7 List of registers

HALLTIM base address: HALLTIM\_BASE = 0x4000 6400

Table 18-4 List of HALLTIM registers

| Register name | Register address   | Register description                  |
|---------------|--------------------|---------------------------------------|
| HALLTIM_CR    | HALLTIM_BASE +0x00 | Control register                      |
| HALLTIM_DIER  | HALLTIM_BASE +0x04 | DMA/Interrupt enable control register |
| HALLTIM_CNT   | HALLTIM_BASE +0x08 | Count value register                  |
| HALLTIM_ARR   | HALLTIM_BASE +0x0C | Auto reload register                  |
| HALLTIM_WIDTH | HALLTIM_BASE +0x10 | HALL pulse width register             |
| HALLTIM_CCR   | HALLTIM_BASE +0x14 | Campare register                      |
| HALLTIM_ISR   | HALLTIM_BASE +0x18 | Interrupt flag register               |
| HALLTIM_ICR   | HALLTIM_BASE +0x1C | Interrupt flag clear register         |
| HALLTIM_STATE | HALLTIM_BASE +0x20 | Status register                       |



## 18.8 Register description

See section 1 [Documentation conventions](#) for abbreviations used in register descriptions.

### 18.8.1 HALLTIM\_CR Control register

Address offset: 0x00      Reset value: 0x0000 0004

| Bit field | Name    | Permission | Function description                                                                                                                                                                                                                                         |
|-----------|---------|------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:24     | RFU     | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                |
| 23        | SOFTCAP | R0W1       | The software captures signals to generate control<br>W0: No function.<br>W1: Generate software trigger signal.                                                                                                                                               |
| 22:8      | FLT2LEN | RW         | Second-order filter bandwidth configuration<br>Signals with pulse widths less than $FLT2LEN \times T_{HALLCLK}$ will be filtered out.<br>Signals with pulse widths no less than $(FLT2LEN+2) \times T_{HALLCLK}$ will pass through.                          |
| 7         | RFU     | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                |
| 6:4       | MMS     | RW         | Trigger output selection<br>000: Overflow pulse<br>001: PWM signal<br>010: CCR match pulse<br>011: HALL signal change pulse<br>100: Filtered CH3 waveform<br>101: Filtered CH2 waveform<br>110: Filtered CH1 waveform<br>111: 3-channel filtered signal XOR. |
| 3:2       | DIV     | RW         | HALL module counting clock source frequency division configuration<br>00: PCLK<br>01: PCLK/2<br>10: PCLK/4<br>11: PCLK/8                                                                                                                                     |
| 1         | FLT1EN  | RW         | First-order filter (5/7 filter) enable control<br>0: Disabled<br>1: Enabled                                                                                                                                                                                  |
| 0         | EN      | RW         | HALL module enable control<br>0: Disabled<br>1: Enabled                                                                                                                                                                                                      |



### 18.8.2 HALLTIM\_DIER DMA/Interrupt enable control register

Address offset: 0x04      Reset value: 0x0000 0004

| Bit field | Name     | Permission | Function description                                                            |
|-----------|----------|------------|---------------------------------------------------------------------------------|
| 31:8      | RFU      | -          | Reserved bits, please keep the default value.                                   |
| 7         | SOFTDMA  | RW         | Software-triggered DMA enable control<br>0: Disabled<br>1: Enabled              |
| 6         | MATCHDMA | RW         | Counter match DMA enable control<br>0: Disabled<br>1: Enabled                   |
| 5         | OVDMA    | RW         | Counter overflow DMA enable control<br>0: Disabled<br>1: Enabled                |
| 4         | CAPDMA   | RW         | Input capture trigger DMA interrupt enable control<br>0: Disabled<br>1: Enabled |
| 3         | RFU      | -          | Reserved bits, please keep the default value.                                   |
| 2         | MATCHIE  | RW         | Counter match interrupt enable control<br>0: Disabled<br>1: Enabled             |
| 1         | OVIE     | RW         | Counter overflow interrupt enable control<br>0: Disabled<br>1: Enabled          |
| 0         | CAPIE    | RW         | Input capture interrupt enable control<br>0: Disabled<br>1: Enabled             |

### 18.8.3 HALLTIM\_CNT Count value register

Address offset: 0x08      Reset value: 0x0000 FFFF

| Bit field | Name | Permission | Function description                          |
|-----------|------|------------|-----------------------------------------------|
| 31:24     | RFU  | -          | Reserved bits, please keep the default value. |
| 23:0      | DATA | RW0        | HALL counter count value.                     |



### 18.8.4 HALLTIM\_ARR Auto reload register

Address offset: 0x0C      Reset value: 0x00FF FFFF

| Bit field | Name | Permission | Function description                          |
|-----------|------|------------|-----------------------------------------------|
| 31:24     | RFU  | -          | Reserved bits, please keep the default value. |
| 23:0      | DATA | RW         | Auto reload value.                            |

### 18.8.5 HALLTIM\_WIDTH HALL pulse width register

Address offset: 0x10      Reset value: 0x0000 0000

| Bit field | Name | Permission | Function description                          |
|-----------|------|------------|-----------------------------------------------|
| 31:24     | RFU  | -          | Reserved bits, please keep the default value. |
| 23:0      | DATA | RO         | HALL pulse width count value.                 |

### 18.8.6 HALLTIM\_CCR Compare register

Address offset: 0x14      Reset value: 0x0000 FFFF

| Bit field | Name | Permission | Function description                          |
|-----------|------|------------|-----------------------------------------------|
| 31:24     | RFU  | -          | Reserved bits, please keep the default value. |
| 23:0      | DATA | RW         | Compare register                              |

### 18.8.7 HALLTIM\_ISR Interrupt flag register

Address offset: 0x18      Reset value: 0x0000 FFFF

| Bit field | Name   | Permission | Function description                          |
|-----------|--------|------------|-----------------------------------------------|
| 31:3      | RFU    | -          | Reserved bits, please keep the default value. |
| 2         | MATCHF | RO         | Match flag                                    |
| 1         | OVF    | RO         | Overflow flag                                 |
| 0         | CAPF   | RO         | Input capture flag                            |



### 18.8.8 HALLTIM\_ICR Interrupt flag clear register

Address offset: 0x1C      Reset value: 0x0000 FFFF

| Bit field | Name   | Permission | Function description                          |
|-----------|--------|------------|-----------------------------------------------|
| 31:3      | RFU    | -          | Reserved bits, please keep the default value. |
| 2         | MATCHF | R1W0       | Match flag cleared.                           |
| 1         | OVF    | R1W0       | Overflow flag cleared.                        |
| 0         | CAPF   | R1W0       | Input capture flag cleared.                   |

### 18.8.9 HALLTIM\_STATE Status register

Address offset: 0x20      Reset value: 0x0000 0000

| Bit field | Name | Permission | Function description                                                                                                   |
|-----------|------|------------|------------------------------------------------------------------------------------------------------------------------|
| 31:7      | RFU  | -          | Reserved bits, please keep the default value.                                                                          |
| 6         | CH3S | RO         | CH3 input signal pre-filter level<br>0: CH3 pre-filter signal is low level.<br>1: CH3 pre-filter signal is high level. |
| 5         | CH2S | RO         | CH2 input signal pre-filter level<br>0: CH2 pre-filter signal is low level.<br>1: CH2 pre-filter signal is high level. |
| 4         | CH1S | RO         | CH1 input signal pre-filter level<br>0: CH1 pre-filter signal is low level.<br>1: CH1 pre-filter signal is high level. |
| 3         | RFU  | -          | Reserved bits, please keep the default value.                                                                          |
| 2         | CH3F | RO         | CH3 input signal filtered level<br>0: CH3 filtered signal is low level.<br>1: CH3 filtered signal is high level.       |
| 1         | CH2F | RO         | CH2 input signal filtered level<br>0: CH2 filtered signal is low level.<br>1: CH2 filtered signal is high level.       |
| 0         | CH1F | RO         | CH1 input signal filtered level<br>0: CH1 filtered signal is low level.<br>1: CH1 filtered signal is high level.       |





## 19 Independent watchdog timer (IWDT)

### 19.1 Overview

The CW32L012 integrates an independent watchdog timer (IWDT), and uses a dedicated internal RC clock source RC10K, which can avoid being affected by external factors during operation. Once the IWDT is started, the user needs to reload the IWDT counter within the specified time interval, otherwise the counter overflow will trigger a reset or generate an interrupt signal. After the IWDT is started, the counting can be stopped. The IWDT can be selected to keep running or suspend counting during DeepSleep mode.

The specially set key-value register can lock the key registers of the IWDT to prevent the registers from being accidentally modified.



## 19.2 Main features

- 12-bit down counter.
- Independent internal low-speed RC oscillator.
- Programmable clock prescale period.
- Overflow can trigger interrupt or reset.
- Register protection lock function.
- Counting can be suspended in DeepSleep mode.

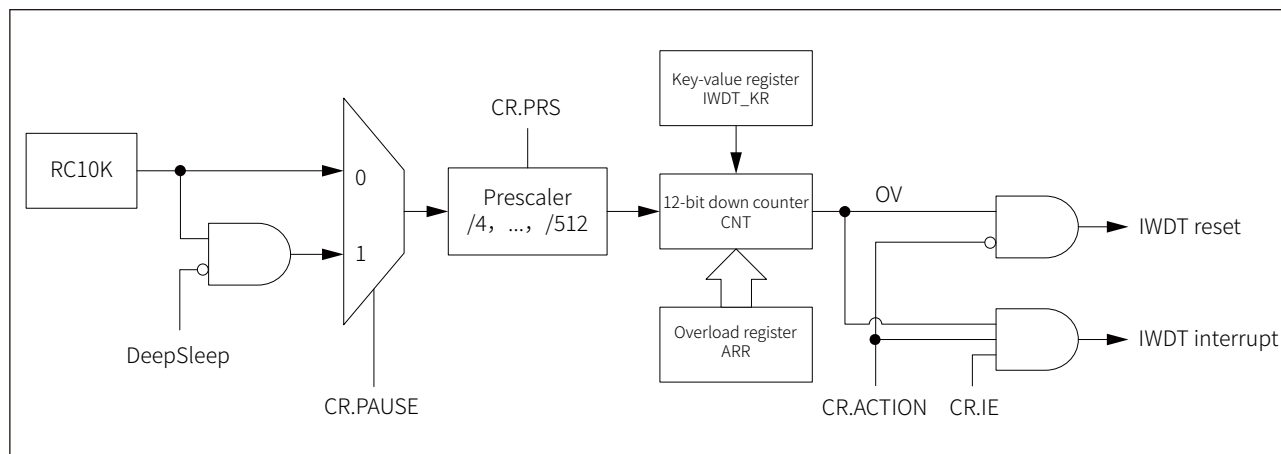


## 19.3 Functional description

### 19.3.1 Functional block diagram

The functional block diagram of IWDT is shown in the following figure:

Figure 19-1 IWDT functional block diagram



The IWDT is implemented by a 12-bit reloadable down-counter, and its counting clock source is the internal dedicated low-speed RC oscillator RC10K. The clock source RC10K signal can be prescaled by 4 to 512 through the PRS bit field of the control register IWDT\_CR. When the IWDT counter overflows, it can optionally generate interrupt and reset signals.

### 19.3.2 Operating modes

To start the IWDT counter, you need to write 0xCCCC to the key value register IWDT\_KR, and the counter starts to count down from 0xFFFF.

Before the counter reduces to 0, writing 0xAAAA to the IWDT\_KR register will trigger a counter reload, loading the ARR register value into the counter. When the IWDT counter value is reduced to 0, an overflow event will be generated. The overflow event can trigger the MCU reset or generate the IWDT interrupt signal, and trigger the counter reload at the same time. The ACTION and IE bit fields of the control register IWDT\_CR are used to control whether interrupts and resets are generated when the watchdog overflows, as shown in the following table:

Table 19-1 Operating modes

| IWDT_CR.ACTION | IWDT_CR.IE | Action after IWDT overflows |
|----------------|------------|-----------------------------|
| 0              | X          | Reset                       |
| 1              | 1          | No reset, only interrupt.   |
| 1              | 0          | No reset, no interrupt.     |

When the MCU enters the DeepSleep mode, the IWDT can choose to suspend the IWDT counting, so as to achieve lower overall system power consumption: the PAUSE bit field of the control register IWDT\_CR is 0, and the IWDT timer is kept running in the DeepSleep mode; when it is 1, the counting is suspended, the IWDT automatically resumes counting when the MCU exits the DeepSleep mode.

### 19.3.3 Window options

Write a value smaller than the reload register IWDT\_ARR to the window register IWDT\_WINR to make the IWDT work in the window watchdog mode. The default value of the IWDT\_WINR register is 0x0FFF, that is, the window option is closed by default.

Modify the value of the IWDT\_WINR register will trigger a reload operation to load the ARR register value into the counter.

When using the IWDT window function, the user should perform a reload operation before the counter value is less than or equal to the IWDT\_WINR window value and decrements to 0 to avoid reset or watchdog timer overflow. A reload operation when the watchdog counter value is greater than the window value will trigger a system reset.

### 19.3.4 Register lock function

Through the key value register IWDT\_KR of the IWDT, the important registers IWDT\_CR, IWDT\_ARR, and IWDT\_WINR of the IWDT can be configured to be locked or unlocked. The registers cannot be modified in the locked state.

Write 0x5555 to the IWDT\_KR register to unlock the IWDT\_CR, IWDT\_ARR, and IWDT\_WINR registers; write any other value to the IWDT\_KR register to enable lock protection.

After the CW32L012 is powered on, the IWDT\_CR, IWDT\_ARR, and IWDT\_WINR registers are locked by default, and the user needs to unlock them before modifying them.

### 19.3.5 Start refresh and stop

Configure the IWDT\_KR register to start, refresh and stop the IWDT:

- Write 0xCCCC to start IWDT.
- Write 0xAAAA, reload the counter, that is, refresh the IWDT.
- Write 0x5A5A, 0xA5A5 sequentially, stop IWDT.

*Attention: The above operations will simultaneously activate the IWDT register lock protection.*



### 19.3.6 Status register

The status register IWDT\_SR indicates the current running status or register update status of the IWDT, as shown in the following table:

Table 19-2 IWDT status indication

| IWDT_SR bit | Name   | Description               | 1                 | 0                |
|-------------|--------|---------------------------|-------------------|------------------|
| 4           | RUN    | Run flag                  | Running           | Not running      |
| 3           | OV     | Overflow flag             | Generate overflow | Not overflow     |
| 5           | RELOAD | Counter reload flag       | Reloading         | Reload completed |
| 2           | WINRF  | WINR register update flag | Updating          | Update completed |
| 1           | ARRF   | ARR register update flag  | Updating          | Update completed |
| 0           | CRF    | CR register update flag   | Updating          | Update completed |

To ensure the correct operation of the IWDT register, after updating IWDT\_WINR, IWDT\_ARR, and IWDT\_CR, the user needs to check whether the WINRF, ARRF, and CRF flags are 0 to confirm whether the operation is completed.

To ensure the reload operation of the IWDT, the user should check whether the RELOAD flag is 0 after the reload operation.



### 19.3.7 Timing duration setting

The counting clock source of the IWDT is dedicated low-speed clock RC10K (clock frequency is approximately 10 kHz, please refer to the data sheet for details). By controlling the PRS bit field of the IWDT\_CR register, the frequency of the clock source RC10K signal can be divided, as shown in the following table:

Table 19-3 IWDT prescaler divider table

| IWDT_CR.PRS | Prescaler value |
|-------------|-----------------|
| 000         | 4               |
| 001         | 8               |
| 010         | 16              |
| 011         | 32              |
| 100         | 64              |
| 101         | 128             |
| 110         | 256             |
| 111         | 512             |

Watchdog timer duration calculation formula:

$$T = (4 \times 2^{\text{PRS}} / f) \times (\text{ARR} + 1)$$

Where, f is the frequency of the clock source RC10K, PRS is the prescaler divider, and ARR is the reload value.

Therefore, when the frequency of the clock source RC10K is 10000Hz, the longest and shortest timing ranges of the IWDT are:

$$\text{IWDT shortest timing} = (4 \times 2^0 / 10000) \times (0x000 + 1) \approx 400\mu\text{s}$$

$$\text{IWDT longest timing} = (4 \times 2^7 / 10000) \times (0xFF + 1) \approx 209.7\text{s}$$

**Example**, when the frequency of the clock source RC10K is 10000Hz, set the prescale value to 64 and the reload value to 512, then:

$$\text{IWDT timing duration} = (4 \times 2^4 / 10000) \times (512 + 1) = 3.28\text{s}.$$



## 19.4 Programming examples

### 19.4.1 Configure IWDT as independent watchdog

- Step 1: Set SYSCTRL\_APBEN2.IWDT to 1 to enable the configuration clock of IWDT.
- Step 2: Write 0xCCCC to the IWDT\_KR register to start the IWDT.  
*Attention: It is necessary to start IWDT before modifying the relevant registers.*
- Step 3: Write 0x5555 to the IWDT\_KR register to release the lock function of the IWDT register.
- Step 4: Configure IWDT\_CR, configure the prescale value of watchdog count clock and RC10K oscillator, action after overflow, whether to automatically suspend in DeepSleep mode.
- Step 5: Configure IWDT\_ARR and configure the overflow period of the watchdog.
- Step 6: Wait for IWDT\_SR.ARRF and IWDT\_SR.CRF to become 0, wait for the reload value and CR register update to complete.
- Step 7: Write 0xAAAA to the IWDT\_KR register to load the ARR into the IWDT counter.

### 19.4.2 Configure IWDT as window watchdog

- Step 1: Set SYSCTRL\_APBEN2.IWDT to 1 to enable the configuration clock of IWDT.
- Step 2: Write 0xCCCC to the IWDT\_KR register to start the IWDT.  
*Attention: It is necessary to start IWDT before modifying the relevant registers.*
- Step 3: Write 0x5555 to the IWDT\_KR register to release the lock function of the IWDT register.
- Step 4: Configure IWDT\_CR, configure the prescale value of watchdog count clock and RC10K oscillator, action after overflow, whether to automatically suspend in DeepSleep mode.
- Step 5: Configure IWDT\_ARR, configure the reload value of the watchdog.
- Step 6: Configure IWDT\_WINR, configure the window size. Caution that IWDT\_WINR must be less than the overloaded value of IWDT\_ARR.
- Step 7: Wait for IWDT\_SR.ARRF, IWDT\_SR.WINRF and IWDT\_SR.CRF to become 0, wait for the reload value, window register and CR register update to complete.
- Step 8: Write 0xAAAA to the IWDT\_KR register to load the ARR into the IWDT counter.

### 19.4.3 Refresh IWDT (feed dog operation)

- Step 1: Write 0xAAAA to the IWDT\_KR register to load ARR into the counter.
- Step 2: Wait for IWDT\_SR.RELOAD to become 0, and wait for the reload operation to complete.



## 19.5 List of registers

IWDT base address: IWDT\_BASE = 0x4000 5000

Table 19-4 List of IWDT registers

| Register name | Register address | Register description |
|---------------|------------------|----------------------|
| IWDT_KR       | IWDT_BASE + 0x00 | Key-value register   |
| IWDT_CR       | IWDT_BASE + 0x04 | Control register     |
| IWDT_ARR      | IWDT_BASE + 0x08 | Reload register      |
| IWDT_SR       | IWDT_BASE + 0x0C | Status register      |
| IWDT_WINR     | IWDT_BASE + 0x10 | Window register      |
| IWDT_CNT      | IWDT_BASE + 0x24 | Count value register |





## 19.6 Register description

See section 1 [Documentation conventions](#) for abbreviations used in register descriptions.

### 19.6.1 IWDT\_KR Key-value register

Address offset: 0x00      Reset value: 0x0000 0000

| Bit field | Name | Permission | Function description                                                                                                                                                                                                                                                                                 |
|-----------|------|------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:16     | RFU  | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                        |
| 15:0      | KR   | WO         | Write 0xCCCC: start the IWDT counter.<br>Write 0xAAAA: reload the IWDT count value.<br>Write 0x5A5A, 0xA5A5 in sequence: stop watchdog.<br>Write 0x5555: release the write protection of IWDT_ARR, IWDT_WINR, IWDT_CR.<br>Write non-0x5555: enable write protection of IWDT_ARR, IWDT_WINR, IWDT_CR. |

### 19.6.2 IWDT\_CR Control register

Address offset: 0x04      Reset value: 0x0000 0000

| Bit field | Name   | Permission | Function description                                                                                                                                                                                                                                                                                                          |
|-----------|--------|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:6      | RFU    | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                 |
| 5         | PAUSE  | RW         | IWDT suspension control in DeepSleep mode<br>0: IWDT continues to run in DeepSleep mode.<br>1: IWDT automatically suspends in DeepSleep mode.                                                                                                                                                                                 |
| 4         | IE     | RW         | IWDT interrupt enable control<br>0: Watchdog interrupt disabled.<br>1: Watchdog interrupt enabled.                                                                                                                                                                                                                            |
| 3         | ACTION | RW         | Action configuration after IWDT overflow<br>0: Reset after watchdog overflow.<br>1: Interrupt after watchdog overflow.                                                                                                                                                                                                        |
| 2:0       | PRS    | RW         | Configure the divider ratio of the watchdog count clock to the RC10K<br>000: 4 frequency division      100: 64 frequency division<br>001: 8 frequency division      101: 128 frequency division<br>010: 16 frequency division      110: 256 frequency division<br>011: 32 frequency division      111: 512 frequency division |

*Attention: This register can be modified only after writing 0x5555 to the IWDT\_KR register; IWDT\_CR can only be modified when IWDT\_SR.CRF is 0.*



### 19.6.3 IWDT\_ARR Reload register

Address offset: 0x08      Reset value: 0x0000 0FFF

| Bit field | Name | Permission | Function description                          |
|-----------|------|------------|-----------------------------------------------|
| 31:12     | RFU  | -          | Reserved bits, please keep the default value. |
| 11:0      | ARR  | RW         | IWDT reload value                             |

*Attention: This register can be modified only after writing 0x5555 to the IWDT\_KR register; IWDT\_ARR can only be modified when IWDT\_SR.CRF is 0.*

### 19.6.4 IWDT\_CNT Count value register

Address offset: 0x24      Reset value: 0x0000 0FFF

| Bit field | Name | Permission | Function description                                                                                                                 |
|-----------|------|------------|--------------------------------------------------------------------------------------------------------------------------------------|
| 31:12     | RFU  | -          | Reserved bits, please keep the default value.                                                                                        |
| 11:0      | CNT  | RO         | Count value register<br><i>Note: The count value can only be considered to be read if the value read twice in a row is the same.</i> |

### 19.6.5 IWDT\_WINR Window register

Address offset: 0x10      Reset value: 0x0000 0FFF

| Bit field | Name | Permission | Function description                          |
|-----------|------|------------|-----------------------------------------------|
| 31:12     | RFU  | -          | Reserved bits, please keep the default value. |
| 11:0      | WINR | RW         | Window comparator compare value.              |



### 19.6.6 IWDT\_SR Status register

Address offset: 0x0C      Reset value: 0x0000 0000

| Bit field | Name   | Permission | Function description                                                                                                                                                                                  |
|-----------|--------|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:6      | RFU    | -          | Reserved bits, please keep the default value.                                                                                                                                                         |
| 5         | RELOAD | RO         | WDT counter reload flag<br>0: WDT counter has completed reload operation.<br>1: WDT counter is reloading.<br><i>Note: In window mode, you need to wait for this flag to become 0 after reloading.</i> |
| 4         | RUN    | RO         | WDT running flag<br>0: WDT is not running.<br>1: WDT is running.                                                                                                                                      |
| 3         | OV     | RW0        | WDT overflow flag<br>R0: WDT does not overflow.<br>R1: WDT overflow.<br>W0: Clear the WDT overflow flag.                                                                                              |
| 2         | WINRF  | RO         | WINR register update flag<br>0: WINR register update completed.<br>1: WINR register is updating.<br><i>Note: After writing the WINR register, you need to wait for this flag to become 0.</i>         |
| 1         | ARRF   | RO         | ARR register update flag<br>0: ARR register update completed.<br>1: ARR register is updating.<br><i>Note: After writing the ARR register, you need to wait for this flag to become 0.</i>             |
| 0         | CRF    | RO         | CR register update flag<br>0: CR register update completed.<br>1: CR register is updating.<br><i>Note: After writing the CR register, you need to wait for the flag to become 0.</i>                  |



## 20 Window watchdog timer (WWDT)

### 20.1 Overview

CW32L012 integrates a window watchdog timer (WWDT), the user needs to refresh within the set time window, otherwise it will trigger a system reset. WWDT is usually used to monitor the program execution flow with strict time requirements to prevent the abnormal execution of the application program caused by external interference or unknown conditions, resulting in system failure.



## 20.2 Main features

- 7-bit down counter.
- PCLK clock driven, 8-level prescaler, maximum frequency division 524288.
- Supports pre-overflow interrupt and count overflow, load count value error reset.
- Can not be closed after opening, unless the system is reset.

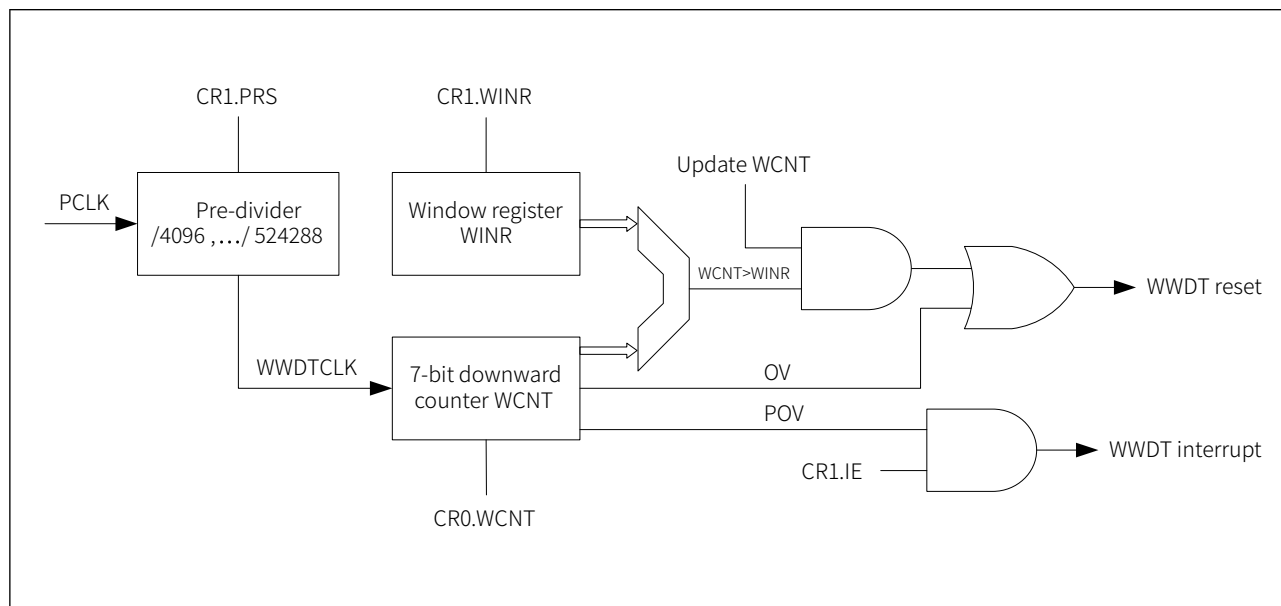


## 20.3 Functional description

### 20.3.1 Functional block diagram

The functional block diagram of WWDT is shown in the following figure:

Figure 20-1 WWDT functional block diagram



WWDT contains a 7-bit down counter. The counting clock source is the internal system clock PCLK. The clock source PCLK can be divided by the PRS bit field of the control register WWDT\_CR1. After frequency division, the counting clock WWDTCCLK is used to drive the counter to count.

The WWDT will stop counting in DeepSleep mode, and resume normal work after the CPU is woken up.

### 20.3.2 Operating mode

After the system is reset, the window watchdog WWDT is in a closed state, and the EN bit field of the control register WWDT\_CR0 is set to 1 to start the WWDT. Once the WWDT is opened, it cannot be closed unless a reset occurs.

Before starting the WWDT, the user must preset the watchdog overflow time and window time. Setting the value of WWDT\_CR0.WCNT can update the initial value of the counter, and setting the value of WWDT\_CR1.WINR can configure the window value of the watchdog. The window value must be less than the initial value of the watchdog counter. After starting the WWDT, the counter counts down from the initial value.

- When the counter decrements to 0x40, the pre-overflow signal POV will be generated, and setting WWDT\_CR1.IE to 1 will generate a pre-overflow interrupt.
- When the counter decrements to 0x3F, an overflow signal OV will be generated, which can trigger a system reset.

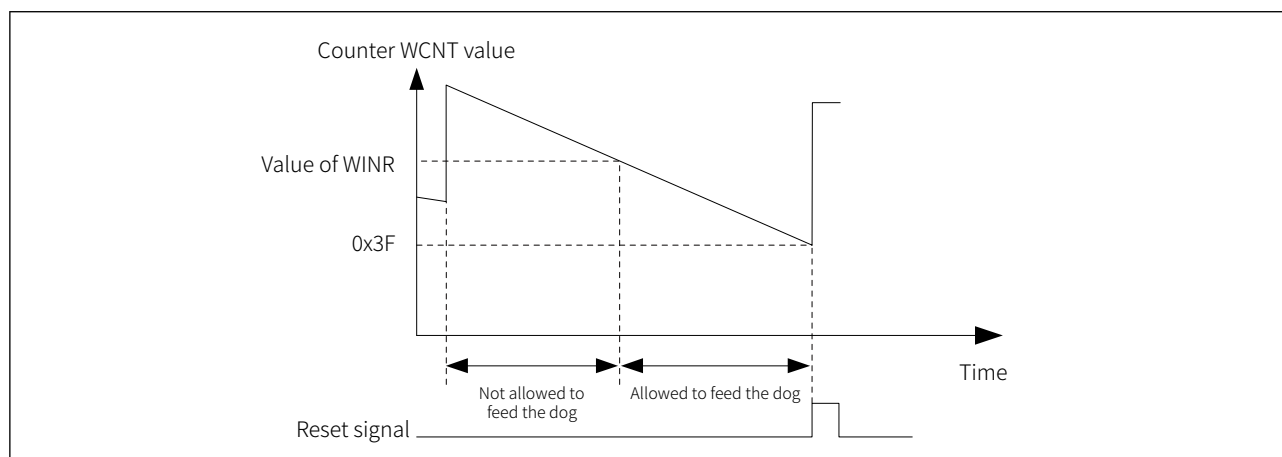
When the current count value WCNT is greater than the window value WINR, updating the watchdog counter will also trigger a system reset.

### 20.3.3 Refresh counter

When WWDT is running, the counter can be refreshed only when the count value is less than or equal to the window value and the count value is decremented to 0x3F, that is, the dog feeding operation, otherwise a system reset will occur.

Set the value of WWDT\_CR1.WINR to configure the upper limit of the delay time window before reset, and the lower limit of the delay time window is fixed at 0x3F. The timing of feeding dog of the window watchdog is shown in the following figure:

Figure 20-2 Refresh (feed dog) timing of the window watchdog



### 20.3.4 Dog-feeding time

The calculation formula of the dog feeding time of the window watchdog:

Minimum time for dog-feeding:

$$T_{WWD\_MIN} = T_{PCLK} \times 4096 \times 2^{PRS} \times (WCNT - WINR)$$

Maximum time for dog-feeding:

$$T_{WWD\_MAX} = T_{PCLK} \times 4096 \times 2^{PRS} \times (WCNT - 0x3F)$$

Where:

$T_{PCLK}$ : PCLK clock period.

PRS: Prescale factor.

WCNT: Counter current count value.

WINR: Watchdog window value.

**Example**, when the PCLK frequency is 24MHz, the prescaler factor PRS is set to 0x01, WCNT is set to 0x6F, and the window value WINR is set to 0x4F, then the dog-feeding time is as follows:

$$T_{WWD\_MIN} = 1/24\text{MHz} \times 4096 \times 21 \times (0x6F - 0x4F) \approx 10.922\text{ms}$$

$$T_{WWD\_MAX} = 1/24\text{MHz} \times 4096 \times 21 \times (0x6F - 0x3F) = 16.384\text{ms}$$

**That is**, the dog-feeding time cannot be earlier than 10.922ms at the earliest, otherwise an error in the loading count value will occur and the system will reset; the dog-feeding time cannot be later than 16.384ms at the latest, otherwise an underflow will occur and the system will reset. The time window size for properly dog-feeding is about 5ms.



### 20.3.5 Reset and interrupt

When the value of the counter WCNT is decremented to 0x40, the pre-overflow interrupt flag bit WWDT\_SR.POV is set by hardware. If the IE bit field of the control register WWDT\_CR1 is set to 1 (**Caution: this bit cannot be cleared to 0 after being set to 1.**), a pre-overflow interrupt flag will be generated. Overflow interrupt request. The user can update the counter WCNT in the interrupt service routine to avoid the reset of the WWDT.

A system reset can be triggered when one of the following conditions is met:

1. The value of the counter WCNT is decremented to 0x3F.
2. When updating the counter WCNT, the current count value is greater than the window value.
3. Write a value less than or equal to 0x3F to WWDT\_CR0.WCNT.





## 20.4 Programming examples

### WWDT basic configuration process

- Step 1: Set `SYSCTRL_APBEN2.WWDT` to 1, enable the configuration clock and working clock of WWDT.
- Step 2: Configure the prescaler of the watchdog counter clock in the window through `WWDT_CR1.PRS`.
- Step 3: Configure the compare value of the window watchdog count through `WWDT_CR1.WINR`.
- Step 4: Configure the initial value of the counter through `WWDT_CR0.WCNT`.
- Step 5: Configure `WWDT_CR1.IE` according to whether the pre-overflow interrupt needs to be enabled.
- Step 6: Set `WWDT_CR0.EN` to 1 to start the window watchdog.

### WWDT dog-feeding

When the value of the counter `WCNT` is decremented to less than or equal to `WWDT_CR1.WINR` and greater than `0x3F`, rewrite the initial value of the counter to `WWDT_CR0.WCNT`.



## 20.5 List of registers

WWDT base address: WWDT\_BASE = 0x4000 5400

Table 20-1 List of WWDT registers

| Register name | Register address | Register description |
|---------------|------------------|----------------------|
| WWDT_CR0      | WWDT_BASE + 0x00 | Control register 0   |
| WWDT_CR1      | WWDT_BASE + 0x04 | Control register 1   |
| WWDT_SR       | WWDT_BASE + 0x08 | Status register      |



## 20.6 Register description

See section 1 [Documentation conventions](#) for abbreviations used in register descriptions.

### 20.6.1 WWDT\_CR0 Control register 0

Address offset: 0x00      Reset value: 0x0000 007F

| Bit field | Name | Permission | Function description                                                                                  |
|-----------|------|------------|-------------------------------------------------------------------------------------------------------|
| 31:8      | RFU  | -          | Reserved bits, please keep the default value.                                                         |
| 7         | EN   | RW1        | WWDT enable control, cannot be disabled after enabling<br>0: WWDT disabled.<br>1: WWDT enabled.       |
| 6:0       | WCNT | RW         | Read: WWDT counter current count value.<br>Write: Update the current count value of the WWDT counter. |

### 20.6.2 WWDT\_CR1 Control register 1

Address offset: 0x04      Reset value: 0x0000 007F

| Bit field | Name | Permission | Function description                                                                                                                                                                     |
|-----------|------|------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:11     | RFU  | -          | Reserved bits, please keep the default value.                                                                                                                                            |
| 10        | IE   | RW         | WWDT pre-overflow interrupt enable control<br>0: Pre-overflow interrupt disabled.<br>1: Pre-overflow interrupt enabled.<br><i>Attention: It cannot be cleared after being set to 1.</i>  |
| 9:7       | PRS  | RW         | WWDT count clock prescale value<br>000: PCLK/4096<br>001: PCLK/8192<br>010: PCLK/16384<br>011: PCLK/32768<br>100: PCLK/65536<br>101: PCLK/131072<br>110: PCLK/262144<br>111: PCLK/524288 |
| 6:0       | WINR | RW         | Watchdog window value.                                                                                                                                                                   |



### 20.6.3 WWDT\_SR Status register

Address offset: 0x08      Reset value: 0x0000 0000

| Bit field | Name | Permission | Function description                                                                                                                            |
|-----------|------|------------|-------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:1      | RFU  | -          | Reserved bits, please keep the default value.                                                                                                   |
| 0         | POV  | RW0        | WWDT pre-overflow flag<br>R0: No pre-overflow occurred in WWDT.<br>R1: WWDT pre-overflow has occurred.<br>W0: Clear the WWDT pre-overflow flag. |



## 21 Universal asynchronous receiver/transmitter (UART)

### 21.1 Overview

CW32L012 integrates 3 universal asynchronous receiver/transmitter (UART), supports asynchronous full-duplex, synchronous half-duplex and single-wire half-duplex modes, supports hardware data flow control and multi-computer communication, also supports LIN (Local Interconnect Network); programmable data frame structure can provide wide range of baud rate selection through wavelet baud rate generator. Built-in timer module supporting receive timeout detection, receive idle detection, automatic baud rate detection and general timing functions.

The UART controller works in a dual clock domain, allowing data reception in DeepSleep mode, and the reception completion interrupt can wake the MCU back to Active mode.



## 21.2 Main features

- Supports dual clock domain drive:
  - Configure clock PCLK
  - Transmit clock UCLK
- Programmable data frame structure:
  - Data word length: 8/9 bits, LSB/MSB first.
  - Parity bit: no parity/odd parity/even parity.
  - Stop bit length: 1/1.5/2 bits .
- 16-bit integer, 4-bit fractional baud rate generator.
- Support asynchronous full-duplex, synchronous half-duplex, single-wire half-duplex.
- Separate transmitter and receiver enable bits.
- Separate transmitted and received signal polarity control.
- TXD/RXD pin configurations are interchangeable.
- Support LoopBack mode.
- Support received data match detection.
- Interval frame and idle character transmission and reception functions.
- Hardware flow control RTS, CTS and RS485 driver enable.
- Transmit buffer empty/receive data complete trigger start ADC.
- Supports multi-machine communication, automatic address recognition.
- 13 interrupt sources with interrupt flags.
- Auto baud rate detection mode 1/2.
- Receive timeout detection/idle detection.
- Built-in timer module supports universal timer function.
- Error detection: parity error, frame structure error, overflow error, noise error.
- Transmit and receive data in low power mode, interrupt wake-up MCU.
- Support communication with devices operating at lower voltages than the MCU (via VC).
- LIN master synchronous break send capability and LIN slave break detection capability:
  - Support length-configurable Break Field transmission.
  - Support 10/11 bit Break Field detection.



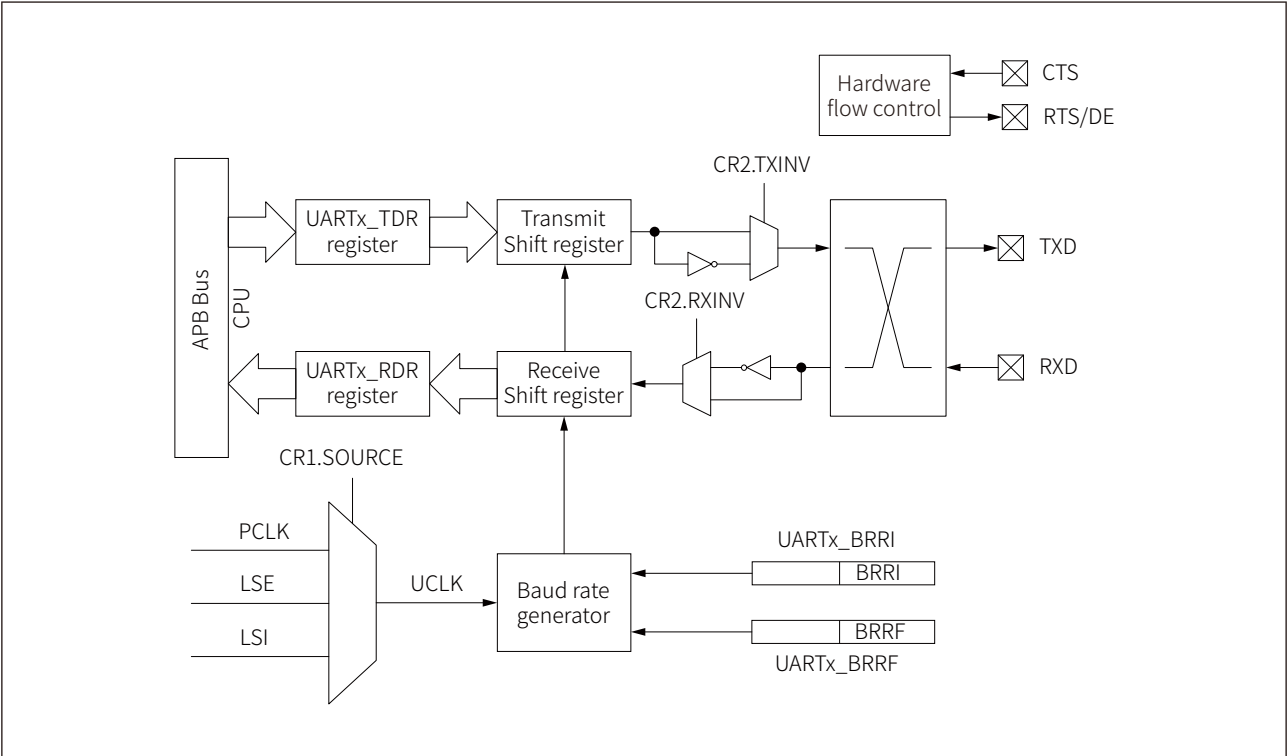
## 21.3 Functional description

### 21.3.1 Functional block diagram

The UART controller is mounted on the APB bus, and the clock domain PCLK is configured, which is fixed to the APB bus clock PCLK, which is used for register configuration logic work; The transfer clock domain UCLK is used for data transceiver logic work, and its source can be selected from PCLK clock, external low-speed clock (LSE) and internal low-speed clock (LSI). The dual clock domain design facilitates baud rate setting and supports waking up the controller from DeepSleep mode.

UART transmitters and receivers have separate enable bits that support separate signal polarity control for transmitted and received data, and the TXD/RXD pin configurations are interchangeable for easy user configuration. The functional block diagram of the UART controller is shown in the following figure:

Figure 21-1 UART functional block diagram



The UART controller supports multiple operating modes. In different operating modes, each pin has different function and different pin configuration, as shown in the following table (where, in the UART chapter, x = 1、2、3.):

Table 21-1 UART ports configuration

| Operating mode           | UART pin                 | Function                        | GPIO configuration                                                   |
|--------------------------|--------------------------|---------------------------------|----------------------------------------------------------------------|
| Asynchronous full duplex | UARTx_TXD                | Data serial output              | Digital, push-pull output/open-drain output, multiplexed.            |
|                          | UARTx_RXD                | Data serial input               | Digital, pull-up input, multiplexed.                                 |
|                          | UARTx_RTS <sup>[1]</sup> | Request to transmit             | Digital, push-pull output/open-drain output, multiplexed.            |
|                          | UARTx_CTS                | Allow to transmit               | Digital, pull-up input, multiplexed.                                 |
| Synchronous half duplex  | UARTx_TXD                | Clock signal output             | Digital, push-pull output/open-drain output, multiplexed.            |
|                          | UARTx_RXD                | Data transmitting and receiving | Digital, push-pull output/open-drain output, multiplexed.            |
| Single-wire half duplex  | UARTx_TXD                | Data transmitting and receiving | Digital, open-drain output, multiplexed (external pull-up required). |
|                          | UARTx_RXD                | Do not use                      | Can be used as general IO.                                           |

[1] When the RS485 driver enable signal is set, the RTS pin is used as the DE function.





### 21.3.2 Synchronous mode

The UART supports the synchronous half-duplex working mode. In this mode, the UARTx\_TXD pin outputs a synchronous shift clock signal, and the UARTx\_RXD pin transmits and receives data. See [Table 21-1: UART ports configuration](#) for the specific configuration of the UARTx\_TXD and UARTx\_RXD pins.

The synchronous mode of the UART can communicate with the single-wire half-duplex mode of the SPI. At this time, the SPI must be configured in a fixed level mode: the clock polarity CPOL is 1, and the clock phase CPHA is 1.

Set the SYNC bit field of the control register UARTx\_CR1 to 1 to make the UART work in synchronous half-duplex mode. At this time, the UART can only be used as a master, the data word length can only be 8 bits, and UARTx\_CR1.SIGNAL and UARTx\_CR2.ADDREN must be cleared.

#### 21.3.2.1 Baud rate setting

In synchronous half-duplex mode, the baud rate calculation formula:

$$\text{BaudRate} = \text{UCLK}/12$$

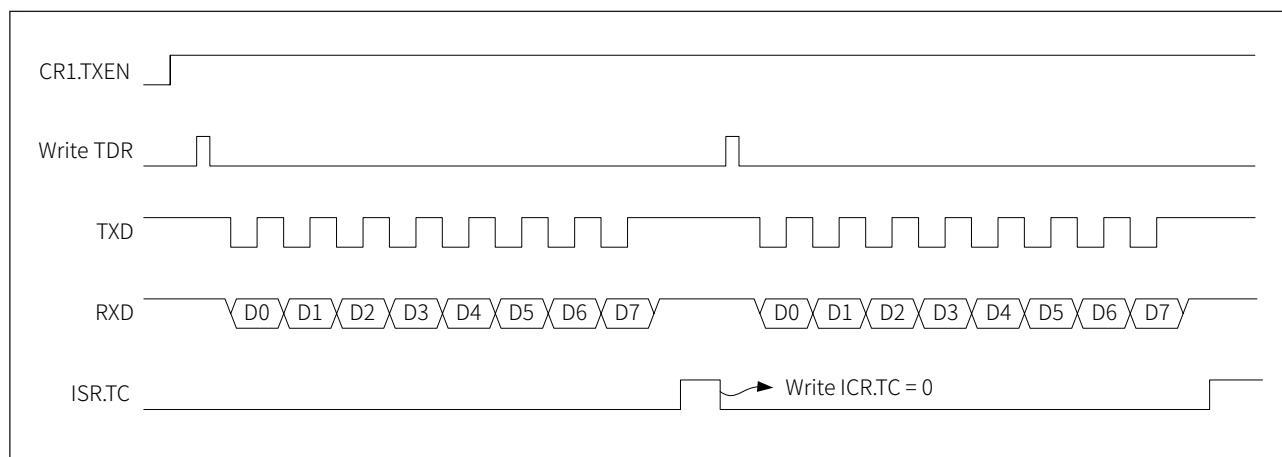
Where, UCLK is the transmission clock of the UART, and its source can be PCLK, LSE or LSI, which is selected by the SOURCE bit field of the control register UARTx\_CR1.



### 21.3.2.2 Data transmission and reception

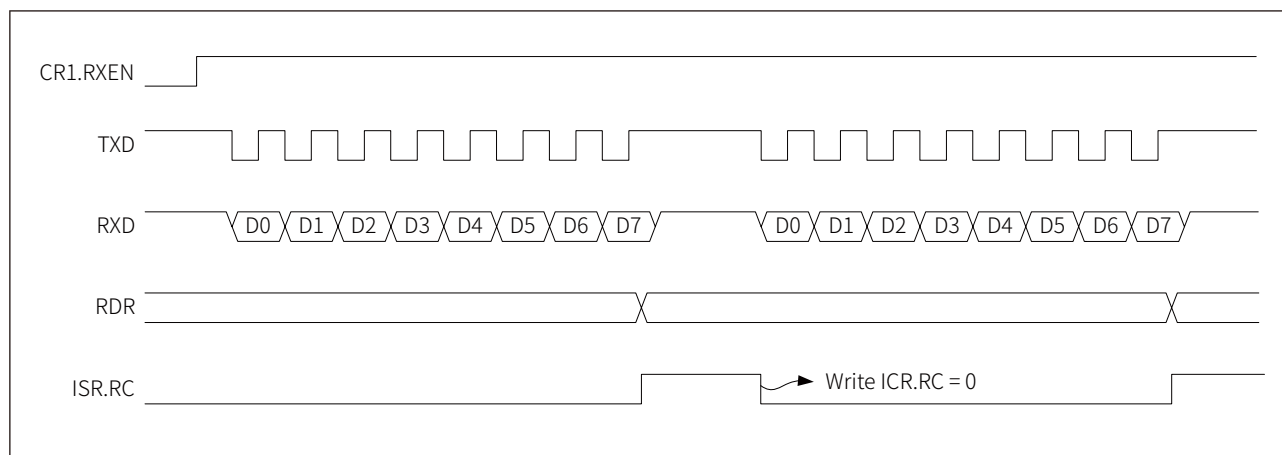
Set the UART transmit enable bit `UARTx_CR1.TXEN` to 1 to make the UART work in the transmitting state. After writing the data to the `UARTx_TDR` register, the data will be serially output from the `UARTx_RXD` pin, and the `UARTx_TXD` pin will output a synchronous shift clock signal. When the data in the `UARTx_TDR` register and shift register have been transmitted, the transmit completion interrupt bit `UARTx_ISR.TC` is set by hardware.

Figure 21-2 Transmit timing in synchronous half-duplex mode (transmit two bytes of data)



Set the UART receive enable bit `UARTx_CR1.RXEN` to 1 to make the UART work in the receiving state, the `UARTx_TXD` pin will output a synchronous shift clock signal, and the data will be serially input from the `UARTx_RXD` pin. After the data reception is completed, the reception completion interrupt flag `UARTx_ISR.RC` will be set by hardware, and the `UARTx_RDR` register can be read at this time. Before receiving the next byte, the `UARTx_ISR.RC` flag must be cleared to 0. After the `RC` flag is cleared, the next byte of data will be received immediately.

Figure 21-3 Receive timing in synchronous half-duplex mode (receive two bytes of data)



### 21.3.3 Asynchronous mode

#### 21.3.3.1 Data frame format

In UART asynchronous communication, data is transmitted and received in the form of data frames, and a frame of data includes a start bit, a data field, an optional parity bit and a stop bit with a programmable width.

##### Start bit

The length of the start bit is fixed at 1 bit, and the judgment method of the start bit can be selected by falling edge or low level, which is selected by the START bit field of the control register UARTx\_CR1. Generally, the falling edge is selected as the judgment method of the start bit. The low-level method to determine the start bit is mainly used in the low-power mode. Please refer to section [21.4 Low-power mode](#).

##### Data field

The data word length can be set to 8 or 9 bits and is configured by controlling the CHLEN bit field of register UARTx\_CR1. When parity is disabled (UARTx\_CR1.PARITYEN is set to 0), the data word length can be set to 8 or 9 bits; when parity is enabled (UARTx\_CR1.PARITYEN is set to 1), the data word length must be set to 9 bits, and if the character length is set to 8 bits, the PARITYEN bit field is automatically cleared. Please refer to [Table 21-2: Data frame structure](#).

The order in which the data bits are transmitted can be selected MSB or LSB first, selected by controlling the MSBF bit field of register UARTx\_CR1.

##### Parity bit

The parity mode supports even parity and odd parity, which is selected by the PARITY bit field of the control register UARTx\_CR1.

- Odd parity: The parity bit makes the total number of "1"s in the data bits and parity bits in a frame of data an odd number.
- Even parity: The parity bit makes the total number of "1" in the data bits and the parity bit in a frame of data an even number.

##### Stop bit

The length of the stop bit can be configured as 1, 1.5 or 2 bits, which is configured through the STOP bit field of the control register UARTx\_CR1.

Table 21-2 Data frame structure

| UARTx_CR1.CHLEN | UARTx_CR1.PARITYEN | Data frame                                      |
|-----------------|--------------------|-------------------------------------------------|
| 0               | X                  | Start bit + 8-bit data + stop bit.              |
| 1               | 0                  | Start bit + 9-bit data + stop bit.              |
| 1               | 1                  | Start bit + 8-bit data + parity bit + stop bit. |

### 21.3.3.2 Fractional baud rate generator

#### Generation of baud rate

The receive and transmit baud rates of the UART are the same, generated by the same baud rate generator.

The baud rate generator supports 4 sampling modes: 16 times sampling, 8 times sampling, 4 times sampling and dedicated sampling. The specific sampling mode is selected by the OVER bit field of the control register UARTx\_CR1.

1. OVER = 00, set 16 times sampling, baud rate calculation formula:

$$\text{BaudRate} = \text{UCLK} / (16 \times \text{BRRI} + \text{BRRF})$$

UCLK is the transmission clock of UART, and its source can be PCLK, LSE or LSI. The specific source is selected by UARTx\_CR1.SOURCE.

BRRI (UARTx\_BRRI[15:0]) is the integer part of the baud rate counter and can be set in the range of 1~65535.

BRRF (UARTx\_BRRF[3:0]) is the fractional part of the baud rate counter, and the settable range is 0~15.

**Example 1**, when the frequency of the transmission clock UCLK is 24MHz, set BRRI = 156 (ie UARTx\_BRRI = 0x9C), BRRF = 4 (ie UARTx\_BRRF = 0x04), then:

$$\text{BaudRate} = 24000000 / (16 \times 156 + 4) = 9600\text{bps}$$

**Example 2**, when the frequency of the transmission clock UCLK is 24MHz, it is required to configure BaudRate = 115200bps, and calculate:

$$16 \times \text{BRRI} + \text{BRRF} = 24000000 / 115200 = 208.33$$

then,  $\text{BRRI} = 208.33 / 16 = 13.02$ , the nearest integer is: 13 (0x0D), and  $\text{BRRF} = 0.02 \times 16 = 0.32$ , the nearest integer is: 0 (0x00).

That is, you need to set UARTx\_BRRI to 0x0D and UARTx\_BRRF to 0x00. At this time, the actual baud rate BaudRate = 115384.62bps, the error rate is 0.16%.

2. OVER = 01, set 8 times sampling, baud rate calculation formula:

$$\text{BaudRate} = \text{UCLK} / (8 \times \text{BRRI})$$

3. OVER = 10, set 4 times sampling, baud rate calculation formula:

$$\text{BaudRate} = \text{UCLK} / (4 \times \text{BRRI})$$

4. OVER = 11, set dedicated sampling, baud rate calculation formula:

$$\text{BaudRate} = (256 \times \text{UCLK}) / \text{BRRI}$$

*Note: Dedicated sampling is only suitable for UART communication at 2400bps, 4800bps or 9600bps baud rate when the transmission clock source is LSE or LSI.*

**Example 1**, the transmission clock UCLK selects LSE, the frequency is 32.768KHz, and the baud rate of 9600bps is required to be configured, then:

$$\text{BRRI} = 256 \times 32768 / 9600 = 873.81, \text{ the nearest integer is: } 874 \text{ (0x36A)}$$

That is, you need to set UARTx\_BRRI to 0x36A. At this time, the actual baud rate BaudRate = 9597.95bps, the error is 0.02%.



**Example of baud rate setting**

The following list the setting values of the baud rate count register and the baud rate error when the frequency of the common MCU is used.

Table 21-3 UCLK is 4MHz baud rate setting example (OVER = 00)

| Baud rate (bps) | BRR1 | BRRF | Actual baud rate | Error rate |
|-----------------|------|------|------------------|------------|
| 4800            | 52   | 1    | 4801.92          | 0.04%      |
| 9600            | 26   | 1    | 9592.33          | -0.08%     |
| 14400           | 17   | 6    | 14388.49         | -0.08%     |
| 19200           | 13   | 0    | 19230.77         | 0.16%      |
| 38400           | 6    | 8    | 38461.54         | 0.16%      |
| 56000           | 4    | 7    | 56338.03         | 0.60%      |
| 57600           | 4    | 5    | 57971.01         | 0.64%      |
| 115200          | 2    | 3    | 114285.71        | -0.79%     |
| 256000          | 1    | 0    | 250000           | -2.34%     |

Table 21-4 UCLK is 8MHz baud rate setting example (OVER = 00)

| Baud rate (bps) | BRR1 | BRRF | Actual baud rate | Error rate |
|-----------------|------|------|------------------|------------|
| 4800            | 104  | 3    | 4799.04          | -0.02%     |
| 9600            | 52   | 1    | 9603.84          | 0.04%      |
| 14400           | 34   | 12   | 14388.49         | -0.08%     |
| 19200           | 26   | 1    | 19184.65         | -0.08%     |
| 38400           | 13   | 0    | 38461.54         | 0.16%      |
| 56000           | 8    | 15   | 55944.06         | -0.10%     |
| 57600           | 8    | 11   | 57553.96         | -0.08%     |
| 115200          | 4    | 5    | 115942.03        | 0.64%      |
| 500000          | 1    | 0    | 500000           | 0.00%      |



Table 21-5 UCLK is 16MHz baud rate setting example (OVER = 00)

| Baud rate (bps) | BRR1 | BRRF | Actual baud rate | Error rate |
|-----------------|------|------|------------------|------------|
| 4800            | 208  | 5    | 4800.48          | 0.01%      |
| 9600            | 104  | 3    | 9598.08          | -0.02%     |
| 14400           | 69   | 7    | 14401.44         | 0.01%      |
| 19200           | 52   | 1    | 19207.68         | 0.04%      |
| 38400           | 26   | 1    | 38369.30         | -0.08%     |
| 56000           | 17   | 14   | 55944.06         | -0.10%     |
| 57600           | 17   | 6    | 57553.96         | -0.08%     |
| 115200          | 8    | 11   | 115107.91        | -0.08%     |
| 1000000         | 1    | 0    | 1000000          | 0.00%      |

Table 21-6 UCLK is 24MHz baud rate setting example (OVER = 00)

| Baud rate (bps) | BRR1 | BRRF | Actual baud rate | Error rate |
|-----------------|------|------|------------------|------------|
| 4800            | 312  | 8    | 4800.00          | 0.00%      |
| 9600            | 156  | 4    | 9600.00          | 0.00%      |
| 14400           | 104  | 3    | 14397.12         | -0.02%     |
| 19200           | 78   | 2    | 19200.00         | 0.00%      |
| 38400           | 39   | 1    | 38400.00         | 0.00%      |
| 56000           | 26   | 13   | 55944.06         | -0.10%     |
| 57600           | 26   | 1    | 57553.96         | -0.08%     |
| 115200          | 13   | 0    | 115384.62        | 0.16%      |
| 1500000         | 1    | 0    | 1500000          | 0.00%      |



Table 21-7 UCLK is 32MHz baud rate setting example (OVER = 00)

| Baud rate (bps) | BRR1 | BRRF | Actual baud rate | Error rate |
|-----------------|------|------|------------------|------------|
| 4800            | 416  | 11   | 4799.76          | 0.00%      |
| 9600            | 208  | 5    | 9600.96          | 0.01%      |
| 14400           | 138  | 14   | 14401.44         | 0.01%      |
| 19200           | 104  | 3    | 19196.16         | -0.02%     |
| 38400           | 52   | 1    | 38415.37         | 0.04%      |
| 56000           | 35   | 11   | 56042.03         | 0.08%      |
| 57600           | 34   | 12   | 57553.96         | -0.08%     |
| 115200          | 17   | 6    | 115107.91        | -0.08%     |
| 2000000         | 1    | 0    | 2000000          | 0.00%      |

Table 21-8 UCLK is 48MHz baud rate setting example (OVER = 00)

| Baud rate (bps) | BRR1 | BRRF | Actual baud rate | Error rate |
|-----------------|------|------|------------------|------------|
| 4800            | 625  | 0    | 4800.00          | 0.00%      |
| 9600            | 312  | 8    | 9600.00          | 0.00%      |
| 14400           | 208  | 5    | 14401.44         | 0.01%      |
| 19200           | 156  | 4    | 19200.00         | 0.00%      |
| 38400           | 78   | 2    | 38400.00         | 0.00%      |
| 56000           | 53   | 9    | 56009.33         | 0.02%      |
| 57600           | 52   | 1    | 57623.05         | 0.04%      |
| 115200          | 26   | 1    | 115107.91        | -0.08%     |
| 3000000         | 1    | 0    | 3000000          | 0.00%      |

Table 21-9 UCLK is 32.768KHz baud rate setting example (OVER = 11)

| Baud rate (bps) | BRR1 | Actual baud rate | Error rate |
|-----------------|------|------------------|------------|
| 2400            | 3495 | 2400.17          | 0.01%      |
| 4800            | 1748 | 4798.97          | -0.02%     |
| 9600            | 874  | 9597.95          | -0.02%     |



### 21.3.3.3 Transmit control

#### Data transmission

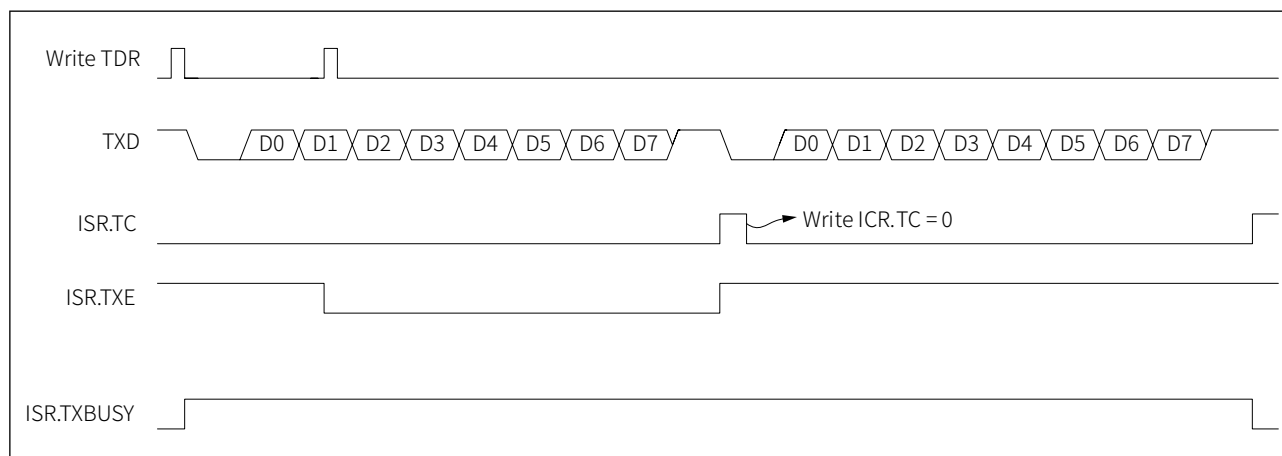
Set UARTx\_CR1.TXEN to 1 to enable the transmit circuit. After writing data to the UARTx\_TDR register, the data is transmitted by hardware to the transmit shift register, which shifts the data out serially (MSB or LSB first). For the register configuration of the specific transmission process, please refer to section [21.7 Programming examples](#).

When the data is transmitted to the transmission shift register by hardware, the transmit buffer empty interrupt flag UARTx\_ISR.TXE will be set by hardware, indicating that the UARTx\_TDR register is empty, and new data to be sent is allowed to be written to the UARTx\_TDR register. While writing data to the UARTx\_TDR register, the UARTx\_ISR.TXE flag will be automatically cleared. If there is still untransmitted data in the transmission shift register at this time, the newly written data will be temporarily stored in the UARTx\_TDR register. After the current data transmission is completed, the data in the UARTx\_TDR register will be transmitted by hardware to the transmission shift register. continue to transmit in the register.

When the data frame transmission in the transmission shift register is completed, if there is no data to be transmitted in the UARTx\_TDR register (i.e., UARTx\_ISR.TXE = 1), the transmission completion interrupt flag bit UARTx\_ISR.TC will be set by hardware, and the UART transmission busy flag bit UARTx\_ISR.TXBUSY will be cleared by hardware, indicating that all data transmission has been completed, and the UART transmitter is idle.

The timing diagram of the UART transmission control link is shown in the following figure:

Figure 21-4 Transmit control sequence in asynchronous working mode (transmitting two bytes of data)



#### Break frame

An break frame is a complete data frame consisting entirely of '0's (also '0' during the stop bit). At the end of the break frame, the transmitter inserts another 2 bits of logic '1' to ensure that the start bit of the next frame is recognised.

Setting UARTx\_TDR.BREAK to 1 will send the break frame, the length of which is determined by the UART data word length. If there is data being sent when BREAK is written, an break frame will be sent on TXD after waiting for the current data to be sent.

#### Idle frame

An idle frame is a complete data frame consisting entirely of '1's (also a '1' during the stop bit) followed by the start bit of the next frame containing the data.

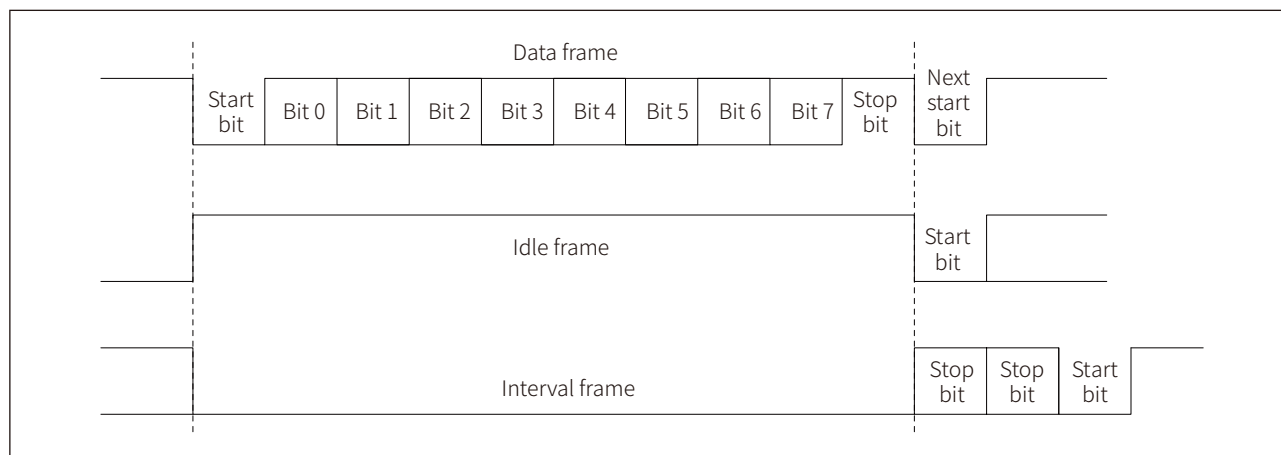




Setting UARTx\_TDR.IDLE to 1 will send an idle frame, the length of which is determined by the UART data word length. If there is data being sent when IDLE is written, an idle frame will be sent on TXD after waiting for the current data to be sent.

The following diagram shows the timing of break frame and idle frame, where the data word length is 8 bits:

Figure 21-5 Break frame and idle frame timing



### 21.3.3.4 Receive control

#### Data reception

Set UARTx\_CR1.RXEN to 1 to enable receive circuitry. When the receiver detects the start bit, it starts to receive data. During reception, the data is serially shifted into the receive shift register from the lowest bit or the highest bit, and transmitted to the UARTx\_RDR register in parallel. At this time, the data can be read out. For the register configuration of the specific receiving process, please refer to [21.7 Programming examples](#).

When the received data is transmitted from the shift register to the UARTx\_RDR register, the reception completion interrupt flag bit UARTx\_ISR.RC will be set by hardware, indicating that the reception of a frame of data has been completed. In the user program, once the UARTx\_ISR.RC flag is detected as 1, the UARTx\_RDR register should be read as soon as possible, and the UARTx\_ISR.RC flag should be cleared. If the UARTx\_RDR register is not read in time, the newly received data will overwrite the data in the UARTx\_RDR register, resulting in data loss.

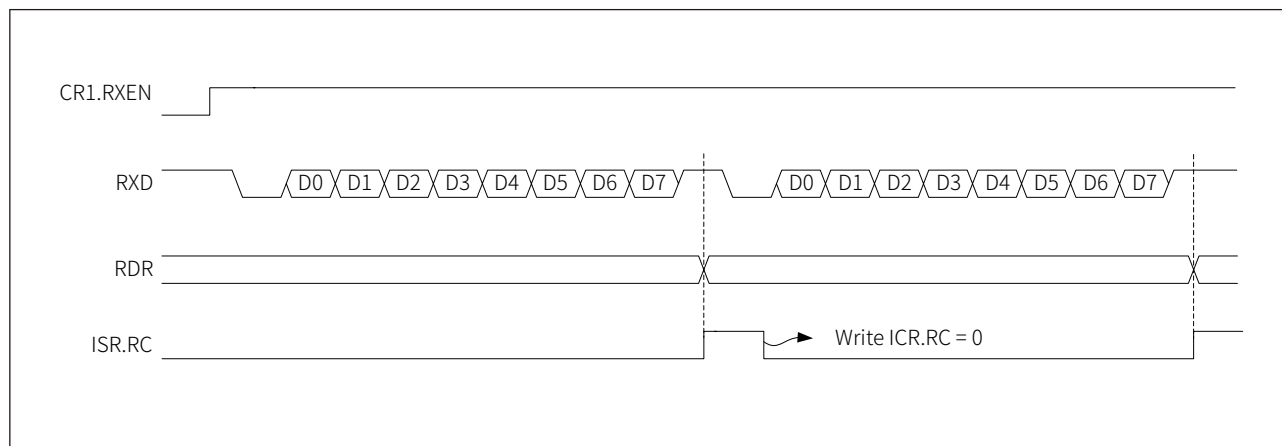
When a data frame is received, the parity check will be performed automatically. If a parity check error is detected, the UARTx\_ISR.PE flag will be set by hardware, indicating a parity check error.

When receiving a data frame, if the correct stop bit is not recognized within the expected time, it is determined that a frame structure error has occurred, and the UARTx\_ISR.FE flag will be set by hardware.

The timing diagram of the UART reception control circuit is shown in the figure below:



Figure 21-6 Receive control timing in asynchronous working mode (receive two bytes of data)



### Overflow error

If the UARTx\_ISR.RC flag bit is not cleared and another character is received, an overflow error occurs and the UARTx\_ISR.ORE flag bit is set by hardware. When an overflow error occurs, newly received data overwrites the data in the UARTx\_RDR register, causing data loss.

If UARTx\_IER.ORE is set to 1, an overflow interrupt is generated, and UARTx\_ICR.ORE is set to 0 to clear the overflow flag.

### Noise error

When noise is detected in a received frame, the UARTx\_ISR.NE flag bit is hardware set on the rising edge of the UARTx\_ISR.RC flag bit, and invalid data is transmitted from the shift register to the UARTx\_RDR register. If UARTx\_IER.NE is set to 1, a noise interrupt is generated, and UARTx\_ICR.NE is set to 0 to clear the noise flag. Note that noise errors are only valid when sampling by 16 or sampling by 8.

### Receive data match

Set UARTx\_CR2.RXMATCHEN to 1 to enable receive data match detection. When the received data matches the target received data configured in the UARTx\_RXMATCH register, the UARTx\_ISR.RXMATCH flag is set by hardware. If UARTx\_IER.RXMATCH is set to 1, a receive data match interrupt is generated. If UARTx\_ICR.RXMATCH is set to 0, the flag is cleared.

### Break frame

When the UART receives a break frame, the RXBRK, RC, and FE flags are set, and the UARTx\_RDR register receives a 0. If the corresponding interrupt is enabled, a corresponding interrupt request will be generated.

### Idle frame

Idle character detection is initiated only when the reception of a character is completed. When an idle character is detected, the RXIDLE flag is set and an interrupt request will be generated if the corresponding interrupt is enabled.

### 21.3.3.5 Single-wire half-duplex mode

Set UARTx\_CR1.SIGNAL to 1 to make the UART work in single-wire half-duplex mode. In this mode, the UARTx\_TXD pin is used for data transmission and reception without occupying the UARTx\_RXD pin (UARTx\_RXD can be used as general IO). See [Table 21-1: UART ports configuration](#) for configuration.

After writing data to the UARTx\_TDR register, the UARTx\_TXD pin immediately enters the transmit state and the output UARTx\_TDR data in the register. After the data transmission is complete, the UARTx\_TXD pin returns to its normal receive state.

When no data is transmitted, the UARTx\_TXD pin is in the receiving state. After the data is received, the reception completion flag UARTx\_ISR.RC will be set by hardware. At this time, the UARTx\_RDR register should be read as soon as possible, and the UARTx\_ISR.RC flag should be cleared.

*Attention: Users should take appropriate application layer protection mechanisms to ensure that multiple masters do not send data to the bus at the same time.*



### 21.3.3.6 Multi-machine communication

UART supports multi-machine communication mode. In this mode, there is one master and multiple slaves on the UART bus. Each slave has a unique slave address. During communication, the master will first send an address frame to address the slave. Only the slave with the matching address is activated and receives the data frame sent by the subsequent master.

#### Master transmits

In the multi-machine communication mode, the master needs to set the frame data length to 9 bits and disable parity check.

The highest bit of the UARTx\_TDR.TDR determines whether the master sends an address frame or a data frame. When UARTx\_TDR[8] is 1, it means that the master sends an address frame. When UARTx\_TDR[8] is 0, it means that the master sends a data frame.

#### Slave receives

In the multi-machine communication mode, the slave needs to set the frame data length to 9 bits, disable parity check, and set UARTx\_CR2.ADDREN to 1 to enable the slave address recognition, and the slave hardware automatically detects the address sent by the master and the local machine. whether the address matches.

If the address matches, the slave will save the received address frame into the UARTx\_RDR register, the UARTx\_ISR.RC flag is set by hardware, and the UARTx\_ISR.SLVMATCH flag is set by hardware, and the slave receives the subsequent data frame sent by the master. During the communication process, the slave should pay attention to:

1. The application program queries UARTx\_RDR[8] in the reception completion interrupt RC to determine whether it is an address frame or a data frame.
2. When the slave sends a data frame, it needs to set UARTx\_TDR[8] to 0 to avoid being regarded as an address frame by other slaves.

If the addresses do not match, the slave will not receive the data frame sent by the master, nor will it generate a reception completion interrupt, and the set UARTx\_ISR.SLVMATCH flag will be cleared.

#### Slave address and address mask

The slave address is configured by the UARTx\_ADDR register, and the slave address should be configured to be unique. The UARTx\_MASK register is an address mask. The slave address bit corresponding to the bit '1' in UARTx\_MASK[7:0] participates in the slave address matching operation, and the slave address bit corresponding to the bit '0' does not participate in the slave address matching operation.

When all bits in UARTx\_MASK[7:0] are set to '1', that is, write 0xFF to the UARTx\_MASK register, then all 8-bit address bits of the slave address participate in the slave address matching, and the master can uniquely identify the slave.

When some bits in UARTx\_MASK[7:0] are set to '0', the corresponding slave address bits do not participate in slave address matching, so as to realize that multiple slaves respond to the same address frame sent by the master, that is, the master addresses multiple slaves at the same time.



**Example 1:**

Set the slave A: UARTx\_ADDR = 0xA0, UARTx\_MASK = 0xFE.

Set the slave B: UARTx\_ADDR = 0xA1, UARTx\_MASK = 0xFE.

Set the slave C: UARTx\_ADDR = 0xA2, UARTx\_MASK = 0xFC.

Set the slave D: UARTx\_ADDR = 0xA3, UARTx\_MASK = 0xFC.

Set the slave E: UARTx\_ADDR = 0xA5, UARTx\_MASK = 0xFF.

When the master transmits a 0xA0 or 0xA1 address frame, it can address slave A, slave B, slave C, and slave D.

When the master transmits 0xA2 or 0xA3 address frame, it can address slave C, slave D.

When the master transmits a 0xA4 address frame, no slave is addressed.

When the master transmits a 0xA5 address frame, it can address the slave E.

**Broadcast address**

In multi-machine communication, the address 0xFF is defined as the broadcast address. When the master transmits the broadcast address frame, the slaves of all addresses are addressed.

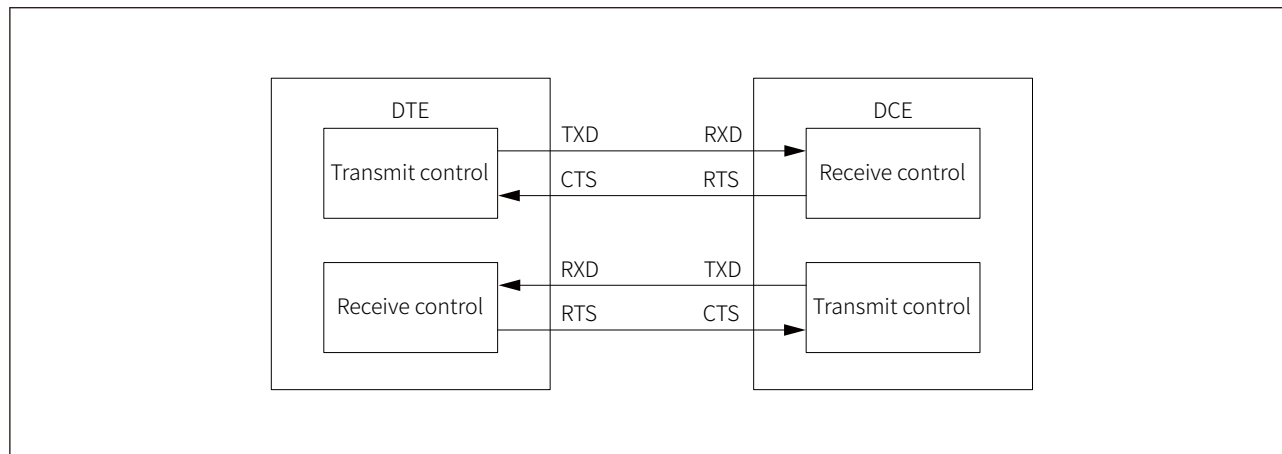


### 21.3.3.7 Hardware flow control

CW32L012 supports UART hardware flow control mode, that is, it supports request to transmit RTS and allow to transmit CTS, which is used to automatically control the transmission and reception of data.

In the data communication system, the schematic diagram of the communication hardware flow control between the data terminal equipment DTE and the data circuit terminating equipment DCE is shown in the following figure:

Figure 21-7 Schematic diagram of hardware flow control



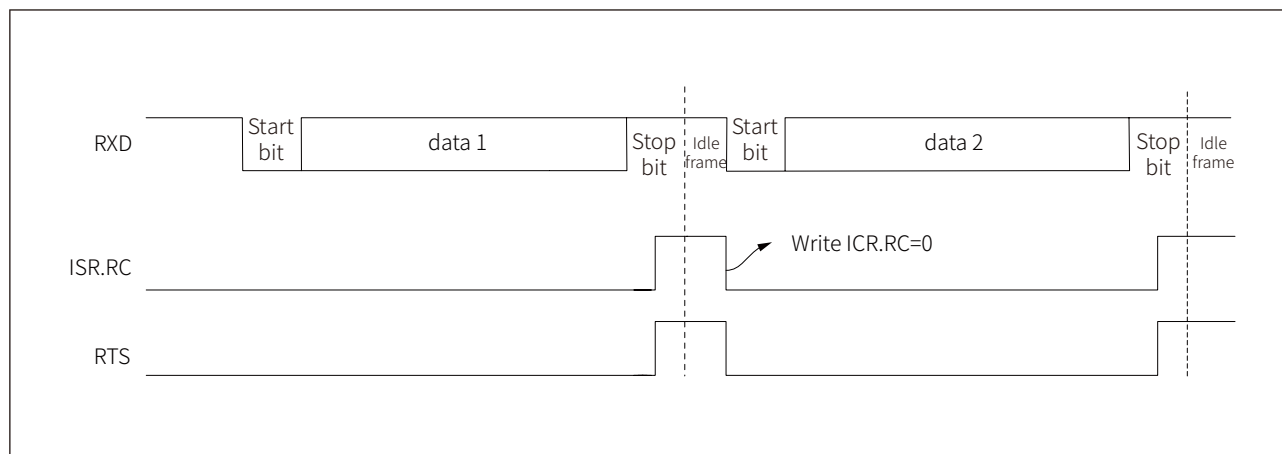
#### Request to transmit RTS

Set `UARTx_CR2.RTSN` to 1 to enable RTS flow control, RTS means request to transmit, and low level is active. When the receiver is ready to receive new data (ie `UARTx_ISR.RC = 0`), the RTS pin will output a low level, requesting the transmitting terminal to transmit a frame of data.

When the receiving terminal receives a frame of data (ie `UARTx_ISR.RC = 1`), the RTS pin will output a high level to notify the transmitting terminal to suspend transmitting.

In the user application, after reading the data in the `UARTx_RDR` register, if you want to continue to receive the next frame of data, the software must clear the `UARTx_ISR.RC` flag bit. At this time, the RTS pin will output a low level, requesting the transmitting terminal to transmit the next frame of data.

Figure 21-8 RTS flow control sequence



### Allow transmitting CTS

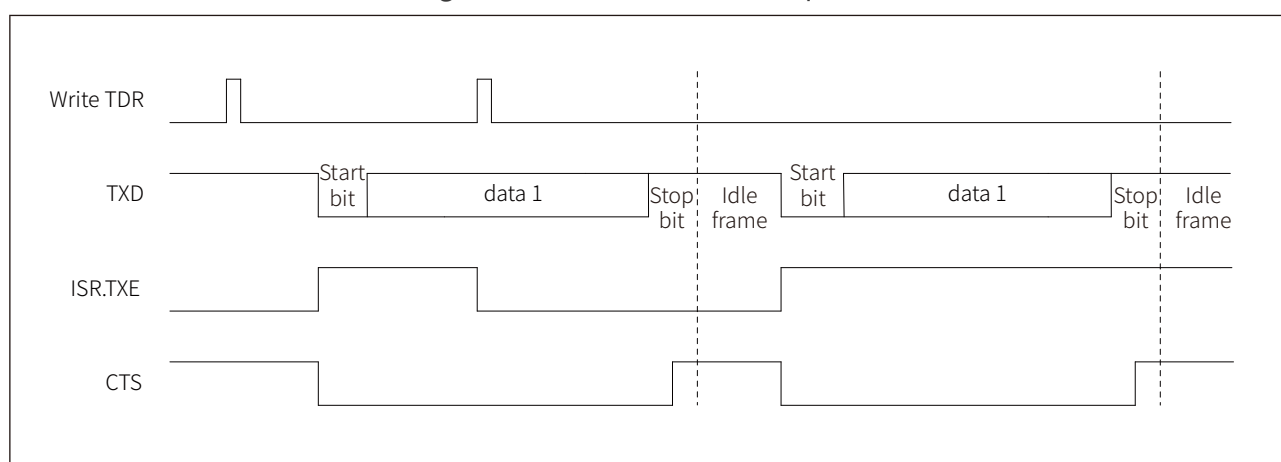
Set UARTx\_CR.CTSEN to 1 to enable CTS flow control, CTS means transmission is allowed, and low level is active.

When the transmitting terminal detects that the CTS signal is low, if there is data to be sent (i.e., UARTx\_ISR.TXE = 0), and there is no data currently being transmitted, this frame of data will be transmitted.

When the transmitting terminal detects that the CTS signal is at a high level, if there is ongoing data transmission, the current transmission will continue, and the transmission will be stopped after the current transmission is completed.

In the CTS flow control process, the CTS signal level flag bit UARTx\_ISR.CTSLV indicates the current level state of the CTS pin; the CTS signal change flag bit UARTx\_ISR.CTS indicates whether the CTS signal has changed, when the CTS signal changes, The UARTx\_ISR.CTS flag will be set by hardware. When the CTS interrupt is enabled (that is, set UARTx\_IER.CTS to 1), an interrupt will be generated. Write UARTx\_ICR.CTS to 0 to clear the interrupt flag.

Figure 21-9 CTS flow control sequence



### 21.3.3.8 RS485 driver enable

Set the DEM bit field of control register UARTx\_CR3 to 1 to enable the RS485 driver enable function, at this time, the RTS pin will be DE (Driver Enable) function, the user needs to configure this pin for the multiplexing function of the digital output.

After enabling this function, user can activate the transmit mode of the external transceiver by DE signal, the polarity of DE signal can be configured through the DEP bit field of UARTx\_CR3 register, and user can select the DE signal to be valid at high level or low level.

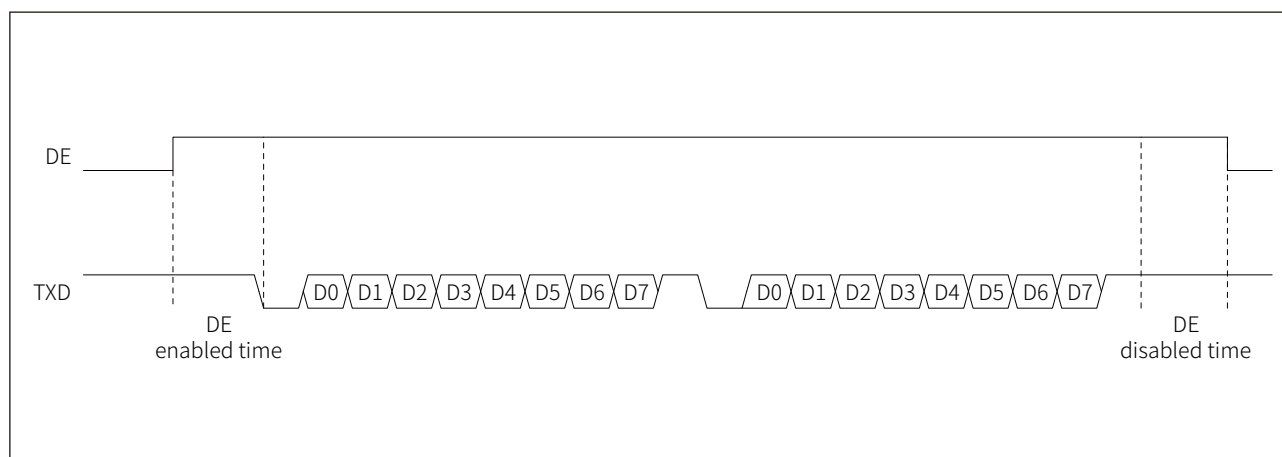
The driver enable time is the time between the valid level of output DE and the start of sending START bits, and the disable time is the time between the completion of sending the last STOP bit and the invalid level of output DE. Both the driver enable time and disable time are set through the DETIME bit field of the UARTx\_CR3 register, and the calculation formula is as follows:

$$\text{Driver enable and disable time} = \text{Sample time} \times \text{DETIME}$$

where the sample time is 1/4, 1/8, and 1/16 bit time, and the DE function is not supported with dedicated sampling (OVER = 3).

When several consecutive data are sent, the DE signal will become invalid level only after the last STOP bit is sent. The following figure shows the timing when DE is valid at high level:

Figure 21-10 Driver enable timing





### 21.3.3.9 LoopBack mode

Set UARTx\_CR2.LOOP to 1 to cause UART to operate in LoopBack mode. In this mode, the RXD signal is sourced from the internal TXD signal and does not occupy the external UARTx\_TXD and UARTx\_RXD pins (UARTx\_TXD and UARTx\_RXD can be used as general purpose GPIOs).

After writing data to the UARTx\_RDR register, the RXD receives data internally to the UARTx\_RDR register. After transmission and reception are complete, the TC and RC flag bits are set to 1. If interrupt is enabled, the corresponding interrupt request will be generated.



21.3.3.10 Input signal source/level shift

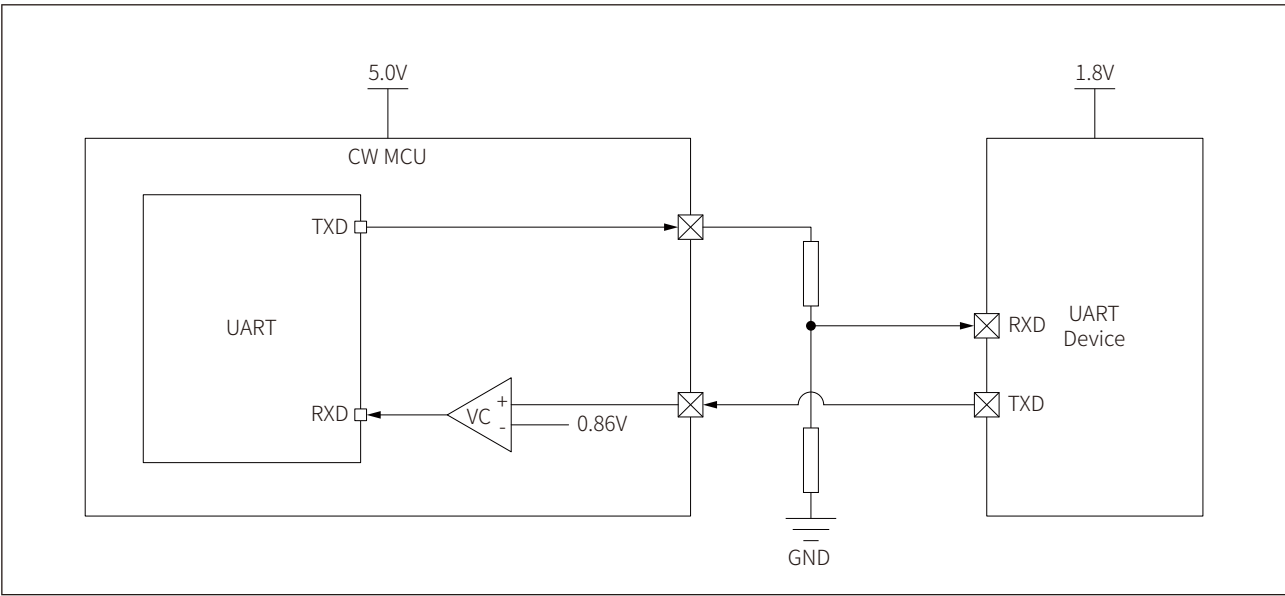
The RXD input signal of this module can be configured for multiple sources as shown in the following table:

Table 21-10 RXD input signal source

| UARTx_CR2.RXSRC | UARTx_CR1.SIGNAL | UARTx_CR2.SWAP | RXD input signal source |
|-----------------|------------------|----------------|-------------------------|
| 000             | 0                | 0              | RXD pin                 |
|                 |                  | 1              | TXD pin                 |
|                 | 1                | 0              | TXD pin                 |
|                 |                  | 1              | RXD pin                 |
| 001             | -                | -              | VC1_OUT                 |
| 010             | -                | -              | VC2_OUT                 |
| 011             | -                | -              | VC3_OUT                 |
| 100             | -                | -              | VC4_OUT                 |

When the input signal source of RXD is configured as VCx\_OUT, this module can communicate with UART devices that are lower than the operating voltage of this chip, and VCx acts as a level shifter in the communication link, and the typical working circuit is shown in the following figure:

Figure 21-11 Schematic diagram of communication between different voltage devices

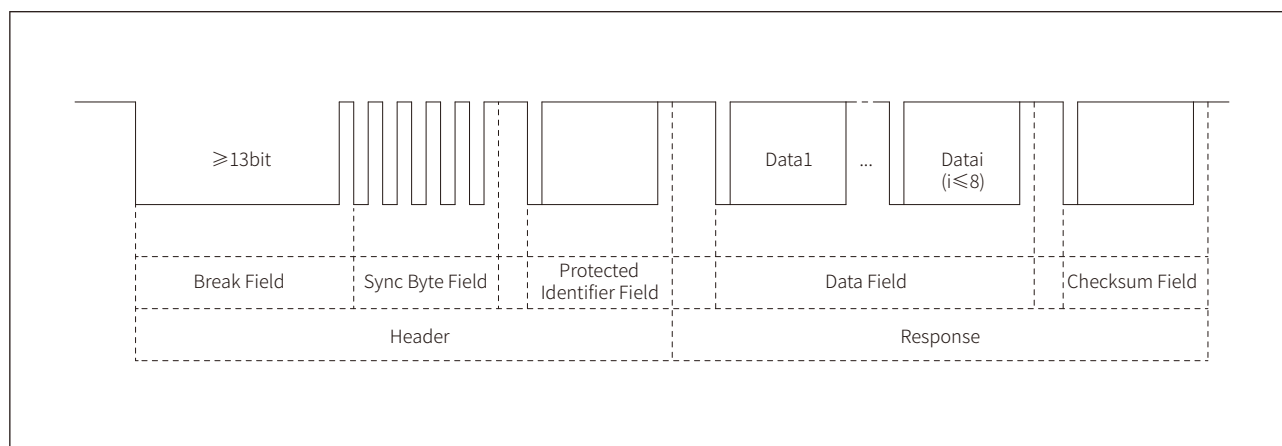


### 21.3.4 LIN Mode

LIN mode is selected by setting the LIN bit field in the UARTx\_CR3 register to 1. In LIN mode, the SIGNAL, SYNC, STOP, and PARITY bits in the UARTx\_CR1 register must be cleared.

A frame of LIN data contains two parts: Header and Response. The master task is responsible for transmitting the frame header, and the slave task receives the frame header and parses the information contained in the frame header, and reacts to the frame header (receive/send/ignore the response part). The frame structure of LIN is shown in the following figure:

Figure 21-12 Structure of the LIN frame



In a LIN frame, except for the Break Field, all other fields are transmitted in the format of a byte field. The byte field includes 1 start bit, 8 data bits, and 1 stop bit, and the data transmission is preceded by the least significant bit (LSB).

#### 21.3.4.1 LIN transmit

##### The transmission of Break Field

The Break Field indicates the beginning of a frame of data. The number of bits in the LIN Break Field is set by UARTx\_TDR.TDR. Note that the Break Field is a dominant level that lasts for at least 13 bits (based on the bit rate of the master node).

In LIN mode, set UARTx\_TDR.BREAK to 1 and write the number of bits of the Break Field to UARTx\_TDR.TDR to immediately transmit the Break Field of the specified length.

##### The transmission of the Sync Byte Field

Writing 0x55 to the UARTx\_TDR register will send the Sync Byte Field.

##### The transmission of the Protected Identifier Field

The first 6 bits of the Protected Identifier Field are the frame ID, plus two parity bits called the Protected ID.

According to the parity formula, the parity bits corresponding to the frame ID are calculated, and then the protected ID is written to the data register UARTx\_TDR, which will send the protected ID field.

### 21.3.4.2 LIN receive

#### Detection of the Break Field

Enabling LIN mode activates the Break Field detection circuit. The length threshold of the receive Break Field is set via the BRKL bit field of the control register UARTx\_CR3.

After enabling the receive circuit, the hardware starts monitoring the start signal on the RXD pin input. After the start bit is detected, the circuit will sample the next bits. When the low width of the RXD communication line is detected to be greater than or equal to the Break Field length threshold set in the BRKL bit field and followed by the break delimiter, the UARTx\_ISR.RXBRK flag bit will be set by the hardware, indicating that the Break Field reception is complete. If UARTx\_IER.RXBRK is 1, an interrupt request will be generated.

If a high level has been sampled before the set Break Field length threshold, the detection circuitry will cancel the current detection and searches for the start bit again.

In LIN mode, when a frame structure error occurs, the receiver stops immediately until the detection circuit receives a '1' (Break Field incomplete) or receives a spacer (Break Field detected).

During Break Field detection, the frame structure error flag bit UARTx\_ISR.FE is set in hardware. While clearing the UARTx\_ISR.RXBRK flag bit, the UARTx\_ISR.FE flag needs to be cleared and the UARTx\_RDR register needs to be read to confirm that the data received is 0x00.

#### Detection of the Sync Byte Field

The slave node calculates the master node bit rate by receiving the Sync Byte Field from the master node, and readjusts its own bit rate according to the calculation result. The slave node should set the TIMCR bit field of the control register UARTx\_CR2 to 0x6, configure it for automatic baud rate detection mode 2 and measure the frequency of the Sync Byte Field by counting the clock UCLK. The bit rate calculation formula is as follows:

$$1 \text{ bit time} = (\text{TIMCNT} + (\text{TIMARR} + 1) \times \text{ARR overflow times}) / (8 \times \text{UCLK})$$

If the UART is set to 16 times sampling (OVER = 00), the adjusted baud rate counter value will be as follows:

$$\text{BRR1} = (\text{uint16\_t}) ((\text{TIMCNT} + (\text{TIMARR} + 1) \times \text{ARR overflow times}) / 8) \gg 4$$

$$\text{BRRF} = (\text{uint16\_t}) ((\text{TIMCNT} + (\text{TIMARR} + 1) \times \text{ARR overflow times}) / 8) \& 0x0F$$



### 21.3.5 Timer operating mode

The UART has a built-in timer module containing a 24bit up counter TIMCNT and a 24bit auto reload register TIMARR. The timer supports five operating modes, as shown in the table below:

Table 21-11 Timer operating modes

| UARTx_CR2. TIMCR | Operating mode                      |
|------------------|-------------------------------------|
| 010              | Waiting for timeout detection mode. |
| 011              | Receive idle detection mode.        |
| 101              | Auto baud rate detection mode 1.    |
| 110              | Auto baud rate detection mode 2.    |
| 111              | General purpose timer mode.         |

#### 21.3.5.1 Waiting for timeout detection mode

Set the TIMCR bit field of control register UARTx\_CR2 to 0x2, and the timer operates in wait timeout detection mode with a timeout time:

$$(TIMARR+1)/f_{BaudRate}$$

Setting UARTx\_CR2.TIMCR to 0x2 will immediately start the counter TIMCNT counting up from 0. The counting clock is the baud rate clock, and re-counts from 0 at every stop bit received; When the count reaches the reload value TIMARR the timer overflows, and at the same time the timer overflow flag bit UARTx\_ISR.TIMOV is set by hardware, indicating that the receive wait timeout is reached and the timer stops automatically; If the timer overflow interrupt is enabled (set UARTx\_IER.TIMOV to 1), the corresponding interrupt service routine will be executed, and the corresponding flag bit should be cleared before exiting the interrupt service routine to avoid entering the interrupt service routine repeatedly.

#### 21.3.5.2 Receive idle detection mode

Set the TIMCR bit field of control register UARTx\_CR2 to 0x3, and the timer operates in receive idle detection mode, with a detection idle time :

$$(TIMARR+1)/f_{BaudRate}$$

Whenever a stop bit is received, the counter TIMCNT is driven by the baud rate clock to count up from 0; When the count reaches the reloaded value TIMARR the timer overflows, and at the same time the timer overflow flag bit UARTx\_ISR.TIMOV is set by the hardware, indicating that the receive idle timeout is reached, and the timer stops automatically; If the timer overflow interrupt is enabled (set UARTx\_IER.TIMOV to 1), the corresponding interrupt service routine will be executed, and the corresponding flag bit should be cleared before exiting the interrupt service routine to avoid entering the interrupt service routine repeatedly.



### 21.3.5.3 Auto baud rate detection mode

The UART supports two auto baud rate detection modes. Set the TIMCR bit field of control register UARTx\_CR2 to 0x5 to configure auto baud rate detection mode 1; set the TIMCR bit field of control register UARTx\_CR2 to 0x6 to configure auto baud rate detection mode 2.

#### Auto baud rate detection mode 1

The falling edge of UARTx\_RXD pin triggers the timer to start, the counter TIMCNT is driven by the counting clock UCLK to count up from 0, the timer overflows after counting to the reload value TIMARR, at the same time the timer overflow flag bit UARTx\_ISR.TIMOV is set by hardware, and the counter keeps the TIMARR value unchanged.

The rising edge of the UARTx\_RXD pin stops the timer counting while the UARTx\_ISR.BAUD flag bit will be set by hardware, indicating that the auto baud rate detection has been completed.

#### Auto baud rate detection mode 2

Auto baud rate detection mode 2 is mainly used for clock synchronization in LIN mode, please refer to [21.3.4 LIN Mode](#).

The falling edge of the UARTx\_RXD pin triggers the timer to start, the counter TIMCNT is driven by the counting clock UCLK to count up from 0, the timer overflows after counting to the reload value TIMARR, at the same time the timer overflow flag bit UARTx\_ISR.TIMOV is set by hardware, and the counter keeps the TIMARR value unchanged.

The 4th falling edge of the UARTx\_RXD pin stops the timer counting, while the UARTx\_ISR.BAUD flag bit will be set by hardware, indicating that the automatic baud rate detection has been completed.

### 21.3.5.4 General purpose timer mode

The timer module built into the UART can be used as a general purpose timer, contains a 24bit up counter and is driven by the count clock UCLK.

Setting the TIMCR bit field of control register UARTx\_CR2 to 0x7 will start the timer immediately and the counter TIMCNT will accumulate counts driven by the count clock UCLK. The timer will stop automatically after a timeout (TIMARR+1) of the UCLK clock.



## 21.4 Low-power mode

The UART controller works in a dual clock domain, supports normal data transmission and reception in DeepSleep mode, and wakes up the MCU back to Active mode through a reception completion interrupt.

If the source of the transmission clock UCLK is set as the low-speed clock, when the system enters the DeepSleep mode, the high-speed clock will stop, the low-speed clock will keep running, and the UART can still transmit and receive data normally (the baud rate only supports 2400bps, 4800bps and 9600bps).

To use the UART wake-up function in DeepSleep mode, you need to enable the UART reception completion interrupt (that is, set UARTx\_IER.RC to 1) before entering DeepSleep mode. When data reception is completed, the reception completion interrupt will wake up the MCU and return to Active mode.

If the source of the transmission clock UCLK is set as the high-speed clock, when the system enters the DeepSleep mode, the high-speed clock will stop running and the UART will not receive data. At this time, the MCU can still be woken up through the GPIO interrupt to receive data in DeepSleep mode. The reference configuration steps are as follows:

- Step 1:** Enable the GPIO falling edge interrupt of the corresponding pin of UARTx\_RXD.
- Step 2:** Set UARTx\_CR1.START to 1, and select the RXD signal start bit determination method to be low level.
- Step 3:** Enable UART reception (that is, set UARTx\_CR1.RXEN to 1).
- Step 4:** Enter DeepSleep mode.
- Step 5:** Wait for the master to transmit data, generate a GPIO falling edge interrupt, and wake up the MCU.
- Step 6:** Closed the GPIO interrupt function of the corresponding pin to RXD and wait for the completion of RXD reception.

For more information on the optimized design of low power mode entry and wake-up, please refer to chapter [3 Power control \(PWR\) and power consumption](#).



## 21.5 UART interrupt

The UART controller supports 13 interrupt sources. When the UART interrupt trigger event occurs, the interrupt flag bit will be set by hardware. If the corresponding interrupt enable control bit is set, an interrupt request will be generated.

A UART module of CW32L012 uses a same system UART interrupt. Whether the UART interrupt generates interrupt jump is controlled by the corresponding bit of the interrupt enable setting register NVIC\_ISER of the Nested Vectored Interrupt Controller (NVIC).

In the user's UART interrupt service routine, the relevant UART interrupt flag bit should be queried for corresponding processing. Before exiting the interrupt service routine, the interrupt flag bit should be cleared to avoid repeatedly entering the interrupt routine.

The flag bits, interrupt enable bits, interrupt flag clear bits and clear methods of each interrupt source of the UART are shown in the following table:

Table 21-12 UART interrupt control

| Interrupt event                         | Interrupt flag bit | Interrupt enable bit | Flag clear method        |
|-----------------------------------------|--------------------|----------------------|--------------------------|
| Transmit buffer empty                   | ISR.TXE            | IER.TXE              | Write UARTx_TDR register |
| Transmit completed                      | ISR.TC             | IER.TC               | Write 0 to ICR.TC        |
| Receive completed                       | ISR.RC             | IER.RC               | Write 0 to ICR.RC        |
| Idle character reception completed      | ISR.RXIDLE         | IER.RXIDLE           | Write 0 to ICR.RXIDLE    |
| Break Field reception completed         | ISR.RXBRK          | IER.RXBRK            | Write 0 to ICR. RXBRK    |
| Automatic baud rate detection completed | ISR.BAUD           | IER.BAUD             | Write 0 to ICR. BAUD     |
| Timer overflow                          | ISR.TIMOV          | IER.TIMOV            | Write 0 to ICR. TIMOV    |
| CTS signal change                       | ISR.CTS            | IER.CTS              | Write 0 to ICR.CTS       |
| Frame structure error                   | ISR.FE             | IER.FE               | Write 0 to ICR.FE        |
| Parity error                            | ISR.PE             | IER.PE               | Write 0 to ICR.PE        |
| Noise flag                              | ISR.NE             | IER.NE               | Write 0 to ICR.NE        |
| Overflow error                          | ISR.ORE            | IER.ORE              | Write 0 to ICR.ORE       |
| Received data match                     | ISR.RXMATCH        | IER.RXMATCH          | Write 0 to ICR.RXMATCH   |





## 21.6 Direct memory access (DMA)

The UART controller supports the DMA transfer function, allowing high-speed data transfer between the UART transceiver data registers and memory without CPU intervention. DMA requests can be triggered by software or hardware, and the DMA requests for transmitting TXD and receiving RXD are generated separately.

### DMA transmits

Set UARTx\_CR2.DMATX to 1 to enable DMA transmit. When the UARTx\_TDR register is empty, the DMA will be triggered to transfer a frame of data from the specified storage area to the UARTx\_TDR register, until all the data configured in the DMA controller is transferred.

For the specific register configuration process of DMA transmission, please refer to chapter 8 [Direct memory access \(DMA\)](#) and section 21.7 [Programming examples](#).

When the DMA transfer is completed, the corresponding DMA channel y transfer completion flag bit DMA\_ISR.TCy will be set by hardware. If the interrupt is enabled (that is, set DMA\_CSRy.TCIE to 1), an interrupt request will be generated.

In user application, before closing the UART, it is recommended to check whether the UARTx\_ISR.TXBUSY flag has been cleared to ensure that all data has been sent to avoid disrupting the last transmission process.

### DMA receives

Set UARTx\_CR2.DMARX to 1 to enable DMA reception. When a frame of data is received (that is, the UARTx\_RDR register is not empty), the DMA will be triggered to transfer the data from the UARTx\_RDR register to the specified storage area, until all the data configured in the DMA controller is transferred.

For the specific register configuration process of DMA reception, please refer to chapter 8 [Direct memory access \(DMA\)](#) and section 21.7 [Programming examples](#).

When the DMA transfer is completed, the corresponding DMA channel y transfer completion flag bit DMA\_ISR.TCy will be set by hardware. If the interrupt is enabled (that is, set DMA\_CSRy.TCIE to 1), an interrupt request will be generated.



## 21.7 Programming examples

### 21.7.1 Asynchronous full-duplex programming examples

#### 21.7.1.1 Transmit data in query mode

- Step 1:** Set `SYSCTRL_AHBEN.GPIOx` to 1, `SYSCTRL_APBENx.UARTx` to 1, enable the GPIO clock and UART configuration clock corresponding to the UART pin.
- Step 2:** Configure the `UARTx_TXD` pin to the push-pull multiplexed output mode. For the specific register configuration steps, please refer to chapter [9 General-purpose input/output \(GPIO\)](#).
- Step 3:** Set `UARTx_CR1.SYNC` to 0, and configure UARTx to be asynchronous full-duplex communication mode.
- Step 4:** Configure the data frame:
- Start bit determination method: Configure `UARTx_CR1.START`.
  - Data word length: Configure `UARTx_CR1.CHLEN`.
  - Check bit: Configure `UARTx_CR1.PARITYEN` and `UARTx_CR1.PARITY`.
  - Stop bit: Configure `UARTx_CR1.STOP`.
- Step 5:** Configure `UARTx_CR1.SOURCE` and select the transmission clock source.
- Step 6:** Configure `UARTx_CR1.OVER` and select sampling mode.
- Step 7:** Configure the `UARTx_BRRI` and `UARTx_BRRF` registers and configure the baud rate. For details, please refer to [21.3.3.2 Fractional baud rate generator](#).
- Step 8:** Set `UARTx_CR1.TXEN` to 1 to enable transmit.
- Step 9:** Set `UARTx_ICR.TC` to 0, clear the transmit completion flag.
- Step 10:** Write a frame of data to be transmitted into the `UARTx_TDR` register.
- Step 11:** Query and wait for the `UARTx_ISR.TC` flag to be set to 1 to confirm that one frame of data is transmitted.
- Step 12:** Repeat steps 9~12 to transmit the next frame of data.



### 21.7.1.2 Receive data in query mode

- Step 1:** Set SYSCTRL\_AHBEN.GPIOx to 1, SYSCTRL\_APBENx.UARTx to 1, enable the GPIO clock and UART configuration clock corresponding to the UART pin.
- Step 2:** Configure the RXD pin as the pull-up input multiplexed mode. For the specific register configuration steps, please refer to chapter [9 General-purpose input/output \(GPIO\)](#).
- Step 3:** Set UARTx\_CR1.SYNC to 0, and configure UARTx to be asynchronous full-duplex communication mode.
- Step 4:** Configure the data frame:
- Start bit determination method: Configure UARTx\_CR1.START.
  - Data word length: Configure UARTx\_CR1.CHLEN.
  - Check bit: Configure UARTx\_CR1.PARITYEN and UARTx\_CR1.PARITY.
  - Stop bit: Configure UARTx\_CR1.STOP
- Step 5:** Configure UARTx\_CR1.SOURCE and select the transmission clock source.
- Step 6:** Configure UARTx\_CR1.OVER and select sampling mode.
- Step 7:** Configure the UARTx\_BRRI and UARTx\_BRRF registers and configure the baud rate. For details, please refer to [21.3.3.2 Fractional baud rate generator](#).
- Step 8:** Write 0x00 to the UARTx\_ICR register to clear all flags.
- Step 9:** Set UARTx\_CR1.RXEN to 1 to enable receive.
- Step 10:** Query and wait for the UARTx\_ISR.RC flag to be set to 1 to confirm that one frame of data has been received.
- Step 11:** Query the error flags UARTx\_ISR.PE and UARTx\_ISR.FE to confirm whether the data is valid. If the data is invalid, perform error processing. If the data is valid, read the UARTx\_RDR register and save the data.
- Step 12:** Set UARTx\_ICR.RC to 0, clear the reception completion flag.
- Step 13:** Repeat steps 10~13 to receive the next frame of data.



### 21.7.1.3 Transmit data in interrupt mode

- Step 1:** Set `SYSCTRL_AHBEN.GPIOx` to 1, `SYSCTRL_APBENx.UARTx` to 1, enable GPIO clock and UART configuration clock corresponding to UART pins.
- Step 2:** Configure the TXD pin to the push-pull multiplexed output mode. For the specific register configuration steps, please refer to chapter [9 General-purpose input/output \(GPIO\)](#).
- Step 3:** Set `UARTx_CR1.SYNC` to 0, and configure UARTx to be asynchronous full-duplex communication mode.
- Step 4:** Configure the data frame:
- Start bit determination method: Configure `UARTx_CR1.START`.
  - Data word length: Configure `UARTx_CR1.CHLEN`.
  - Check bit: Configure `UARTx_CR1.PARITYEN` and `UARTx_CR1.PARITY`.
  - Stop bit: Configure `UARTx_CR1.STOP`.
- Step 5:** Configure `UARTx_CR1.SOURCE` and select the transmission clock source.
- Step 6:** Configure `UARTx_CR1.OVER` and select sampling mode.
- Step 7:** Configure the `UARTx_BRRI` and `UARTx_BRRF` registers and configure the baud rate. For details, For details, please refer to [21.3.3.2 Fractional baud rate generator](#).
- Step 8:** Configure the NVIC controller, see chapter [5 Interrupt](#).
- Step 9:** Set `UARTx_IER.TXE` to 1 to enable transmit buffer empty interrupt.
- Step 10:** Set `UARTx_CR1.TXEN` to 1 to enable transmit.
- Step 11:** Write a frame of data to be transmitted into the `UARTx_TDR` register.
- Step 12:** The data starts to transmit, the transmit buffer is empty, and the interrupt service function is entered: query and judge the `UARTx_ISR.TXE` flag bit, if the flag bit is 1, write a new frame of data to the `UARTx_TDR` register.
- Step 13:** Query and wait for the `UARTx_ISR.TXBUSY` flag to be cleared and turn off the UART.



### 21.7.1.4 Receive data in interrupt mode

- Step 1:** Set SYSCTRL\_AHBEN.GPIOx to 1, SYSCTRL\_APBENx.UARTx to 1, enable the GPIO clock and UART configuration clock corresponding to the UART pin.
- Step 2:** Configure the RXD pin as the pull-up input multiplexed mode. For the specific register configuration steps, please refer to chapter [9 General-purpose input/output \(GPIO\)](#).
- Step 3:** Set UARTx\_CR1.SYNC to 0, and configure UARTx to be asynchronous full-duplex communication mode.
- Step 4:** Configure the data frame:
- Start bit determination method: Configure UARTx\_CR1.START.
  - Data word length: Configure UARTx\_CR1.CHLEN.
  - Check bit: Configure UARTx\_CR1.PARITYEN and UARTx\_CR1.PARITY.
  - Stop bit: Configure UARTx\_CR1.STOP.
- Step 5:** Configure UARTx\_CR1.SOURCE and select the transmission clock source.
- Step 6:** Configure UARTx\_CR1.OVER and select sampling mode.
- Step 7:** Configure the UARTx\_BRRI and UARTx\_BRRF registers and configure the baud rate. For details, please refer to [21.3.3.2 Fractional baud rate generator](#).
- Step 8:** Configure the NVIC controller, see chapter [5 Interrupt](#).
- Step 9:** Set UARTx\_IER.RC to 1 to enable receive completion interrupt.
- Step 10:** Set UARTx\_CR1.RXEN to 1 to enable receive.
- Step 11:** Wait for the completion of receiving a frame of data, and enter the interrupt service function: query the UARTx\_ISR.PE and UARTx\_ISR.FE flag bits to confirm whether the data is valid, if invalid, perform error handling; if valid, query and judge the UARTx\_ISR.RC flag bit, If the flag bit is 1, read the UARTx\_RDR register and save the data, and clear the flag bit.



### 21.7.1.5 DMA transmits data

The following is an example of DMA channel 1 transmit data for UART1 (UART1\_TXD pin is PA09). In this application example, set the transfer clock UCLK to 24MHz, the baud rate to 115200bps, and the DMA data transmission amount to 100. The DMA transmission is started by hardware triggering, and the data in an 8-bit array defined in the SRAM area is transferred to the UART1\_TDR register.

- Step 1:** Set SYSCTRL\_AHBEN.GPIOA to 1, SYSCTRL\_APBEN1.UART1 to 1, SYSCTRL\_AHBEN.DMA to 1, enable the configuration clock and working clock of GPIOA, UART1 and DMA respectively.
- Step 2:** Configure the UART1\_TXD pin (ie PA09) to push-pull multiplexed output mode. For the specific register configuration steps, please refer to chapter 9 [General-purpose input/output \(GPIO\)](#).
- Step 3:** Set UART1\_CR1.SYNC to 0, and configure UART1 to be asynchronous full-duplex communication mode.
- Step 4:** Configure the data frame:
  - Start bit determination method: set UART1\_CR1.START to 0, the falling edge of RXD signal.
  - Parity bit: set UART1\_CR1.PARITY to 0, no parity.
  - Stop bit: configure UART1\_CR1.STOP as 0, 1-bit length of stop bit.
- Step 5:** Set UART1\_CR1.SOURCE to 0, select the transmission clock source as PCLK = 24MHz.
- Step 6:** Set UART1\_CR1.OVER to 0, select 16 times sampling mode.
- Step 7:** Set UART1\_BRR1.BRR1 to 0x0D, UART1\_BRRF.BRRF to 0x00, configure baud rate to 115200bps.
- Step 8:** Set UART1\_CR1.TXEN to 1 to enable transmission.
- Step 9:** Define a variable `uint8_t SendBuff[100]`, and assign it.
- Step 10:** Write 0x00 to UART1\_ICR to clear all flag bits of UART1.
- Step 11:** Set DMA\_ICR.TC1 and DMA\_ICR.TE1 to 0, clear all interrupt flags of DMA channel 1.
- Step 12:** Set DMA\_SRCADDR1= ( `uint32_t` ) SendBuff, the source address is SendBuff.
- Step 13:** Set DMA\_DSTADDR1= ( `uint32_t` ) &UART1\_TDR, the destination address is the UART1\_TDR register.
- Step 14:** Set DMA\_CSR1.SRCINC to 1, the source address is incremented automatically.
- Step 15:** Set DMA\_CSR1.DSTINC to 0, the destination address is fixed.
- Step 16:** Write 0x10064 to DMA\_CNT1, and the data transmission amount is 100.
- Step 17:** Set DMA\_CSR1.SIZE to 0, and the transmission data bit width to 8bit.
- Step 18:** Set DMA\_CSR1.TRANS to 1, block transmission (BLOCK).
- Step 19:** Set DMA\_TRIG1.Type to 1, hardware trigger mode.
- Step 20:** Set DMA\_TRIG1.HARDSRC to 0x01, trigger DMA transmissionr when UART1\_TDR register is empty.
- Step 21:** Set DMA\_CSR1.EN to 1 to enable DMA channel 1 transmit.
- Step 22:** Set UART1\_CR2.DMATX to 1, and enable the DMA hardware handshake logic of UART1 TXD.
- Step 23:** Query and wait for DMA\_ISR.TC1 to become 1 to confirm that the DMA transmission is completed.
- Step 24:** Set UART1\_CR2.DMATX to 0, and turn off the DMA hardware handshake logic of UART1 TXD.
- Step 25:** Query and wait for UART1\_ISR.TXBUSY to become 0, confirm that the data has been transmitted through UART1.
- Step 26:** Repeat steps 9 ~ 26 if new data is to be transferred.



### 21.7.1.6 DMA receives data

The following is an example of DMA channel 2 receive data for UART1 (UART1\_RXD pin is PA10). In this application example, the transmission clock UCLK is set to 24MHz, the baud rate is 115200bps, and the DMA data transmission amount is 100. The DMA transmission is started by hardware triggering, and the data in the UART1\_RDR register is transferred to an 8-bit array in the SRAM area. The specific configuration process is as follows:

- Step 1:** Set SYSCTRL\_AHBEN.GPIOA to 1, SYSCTRL\_APBEN1.UART1 to 1, SYSCTRL\_AHBEN.DMA to 1, enable the configuration clock and working clock of GPIOA, UART1 and DMA respectively.
- Step 2:** Configure the UART1\_RXD pin (that is, PA10) as the pull-up multiplexing input mode. For the specific register configuration steps, please refer to chapter 9 [General-purpose input/output \(GPIO\)](#).
- Step 3:** Set UART1\_CR1.SYNC to 0, and configure UART1 to be asynchronous full-duplex communication mode.
- Step 4:** Configure the data frame:
  - Start bit determination method: set UART1\_CR1.START to 0, the falling edge of RXD signal.
  - Parity bit: set UART1\_CR1.PARITY to 0, no parity.
  - Stop bit: configure UART1\_CR1.STOP as 0, 1-bit length of stop bit.
- Step 5:** Set UART1\_CR1.SOURCE to 0, select the transmission clock source as PCLK = 24MHz.
- Step 6:** Set UART1\_CR1.OVER to 0, select 16 times sampling mode.
- Step 7:** Set UART1\_BRRI.BRRI to 0x0D, UART1\_BRRF.BRRF to 0x00, configure baud rate to 115200bps.
- Step 8:** Set UART1\_CR1.RXEN to 1 to enable reception.
- Step 9:** Define a variable `uint8_t RecvBuff[100]`.
- Step 10:** Write 0x00 to UART1\_ICR to clear all flag bits of UART1.
- Step 11:** Set DMA\_ICR.TC2 and DMA\_ICR.TE2 to 0, clear all interrupt flags of DMA channel 2.
- Step 12:** Set DMA\_SRCADDR2 = (uint32\_t) &UART1\_RDR, the source address is the UART1\_RDR register.
- Step 13:** Set DMA\_DSTADDR2 = (uint32\_t) RecvBuff, the destination address is RecvBuff.
- Step 14:** Set DMA\_CSR2.SRCINC to 0, the source address is fixed.
- Step 15:** Set DMA\_CSR2.DSTINC to 1, and the destination address is incremented automatically.
- Step 16:** Write 0x10064 to DMA\_CNT2, and the data transmission amount is 100.
- Step 17:** Set DMA\_CSR2.SIZE to 0, and the transmission data bit width to 8bit.
- Step 18:** Set DMA\_CSR2.TRANS to 1, block transmission (BLOCK).
- Step 19:** Set DMA\_TRIG2.TYPE to 1, hardware trigger mode.
- Step 20:** Set DMA\_TRIG2.HARDSRC to 0x00, trigger DMA transmission when UART1\_RDR register is not empty.
- Step 21:** Set DMA\_CSR2.EN to 1 to enable DMA channel 2 transmission.
- Step 22:** Set UART1\_CR2.DMARX to 1, and enable the DMA hardware handshake logic of UART1 RXD.
- Step 23:** Query and wait for DMA\_ISR.TC2 to become 1 to confirm that the DMA transfer is completed.
- Step 24:** Set UART1\_CR2.DMARX to 0, and disable the DMA hardware handshake logic of UART1 RXD.
- Step 25:** Repeat steps 10 ~ 25 if new data is to be transferred.



## 21.7.2 Synchronous half-duplex programming examples

### 21.7.2.1 Transmit data in query mode

- Step 1: Set `SYSCTRL_AHBEN.GPIOx` to 1, `SYSCTRL_APBENx.UARTx` to 1, enable GPIO clock and UART configuration clock.
- Step 2: Configure the `UARTx_TXD` and `UARTx_RXD` pins as push-pull multiplexed output mode, and configure the slave chip select pin `NCS` as push-pull output mode. For the specific register configuration steps, please refer to chapter [9 General-purpose input/output \(GPIO\)](#).
- Step 3: Set `UARTx_CR1.SYNC` to 1, and configure UARTx to be synchronous half-duplex communication mode.
- Step 4: Configure `UARTx_CR1.SOURCE`, select the transmission clock source, and configure the baud rate. For details, please refer to [21.3.2 Synchronous mode](#).
- Step 5: Pull down the `NCS` signal and select a slave.
- Step 6: Set `UARTx_CR1.TXEN` to 1 to enable transmit.
- Step 7: Set `UARTx_ICR.TC` to 0, and clear the transmission completion flag.
- Step 8: Write the 8-bit data to be transmitted to the `UARTx_TDR` register.
- Step 9: Query and wait for the `UARTx_ISR.TC` flag bit to be set to 1 to confirm that the data transmission is completed.
- Step 10: Repeat steps 7~10 to transmit the next frame of data.
- Step 11: Pull up the `NCS` signal to end the communication.

### 21.7.2.2 Receive data in query mode

- Step 1: Set `SYSCTRL_AHBEN.GPIOx` to 1, `SYSCTRL_APBENx.UARTx` to 1, enable GPIO clock and UART configuration clock.
- Step 2: Configure the `UARTx_TXD` and `UARTx_RXD` pins as push-pull multiplexed output mode, and configure the slave chip select pin `NCS` as push-pull output mode. For the specific register configuration steps, please refer to chapter [9 General-purpose input/output \(GPIO\)](#).
- Step 3: Set `UARTx_CR1.SYNC` to 1, and configure UARTx to be synchronous half-duplex communication mode.
- Step 4: Configure `UARTx_CR1.SOURCE`, select the transmission clock source, and configure the baud rate. For details, please refer to [21.3.2 Synchronous mode](#).
- Step 5: Pull down the `NCS` signal and select a slave.
- Step 6: Write 0x00 to `UARTx_ICR` to clear all flags.
- Step 7: Set `UARTx_CR1.RXEN` to 1 to enable reception.
- Step 8: Query and wait for the `UARTx_ISR.RC` flag to be set to 1 to confirm that the data reception is completed.
- Step 9: Read the `UARTx_RDR` register and save the data.
- Step 10: Set `UARTx_ICR.RC` to 0, clear the reception completion flag.
- Step 11: Repeat steps 8~11 to receive the next 8-bit data.
- Step 12: Pull up the `NCS` signal to end the communication.





### 21.7.3 Single-wire half-duplex programming examples

#### 21.7.3.1 Transmit data in query mode

- Step 1:** Set `SYSCTRL_AHBEN.GPIOx` to 1, `SYSCTRL_APBENx.UARTx` to 1, enable the GPIO clock and UART configuration clock corresponding to the UART pin.
- Step 2:** Configure the `UARTx_TXD` pin to open-drain multiplexed output mode, and connect an external pull-up resistor. For the specific register configuration steps, please refer to chapter 9 [General-purpose input/output \(GPIO\)](#).
- Step 3:** Set `UARTx_CR1.SIGNAL` to 1, and configure UARTx to be single-wire half-duplex communication mode.
- Step 4:** Configure the data frame:
- Start bit determination method: Configure `UARTx_CR1.START`.
  - Data word length: Configure `UARTx_CR1.CHLEN`.
  - Check bit: Configure `UARTx_CR1.PARITYEN` and `UARTx_CR1.PARITY`.
  - Stop bit: Configure `UARTx_CR1.STOP`.
- Step 5:** Configure `UARTx_CR1.SOURCE` and select the transmission clock source.
- Step 6:** Configure `UARTx_CR1.OVER` and select sampling mode.
- Step 7:** Configure the `UARTx_BRRI` and `UARTx_BRRF` registers and configure the baud rate. For details, please refer to [21.3.3.2 Fractional baud rate generator](#).
- Step 8:** Set `UARTx_CR1.TXEN` to 1 and `UARTx_CR1.RXEN` to 1 to enable transmit and receive.
- Step 9:** Set `UARTx_ICR.TC` to 0, and clear the transmission completion flag.
- Step 10:** Write a frame of data to be transmitted into the `UARTx_TDR` register.
- Step 11:** Query and wait for the `UARTx_ISR.TC` flag bit to be set to 1 to confirm that the data transmission is completed.
- Step 12:** Repeat steps 9~12 to transmit the next frame of data.



### 21.7.3.2 Receive data in query mode

- Step 1:** Set SYSCTRL\_AHBEN.GPIOx to 1, SYSCTRL\_APBENx.UARTx to 1, enable the GPIO clock and UART configuration clock corresponding to the UART pin.
- Step 2:** Configure the UARTx\_TXD pin to open-drain multiplexed output mode, and connect an external pull-up resistor. For the specific register configuration steps, please refer to chapter 9 [General-purpose input/output \(GPIO\)](#).
- Step 3:** Set UARTx\_CR1.SIGNAL to 1, and configure UARTx to be single-wire half-duplex communication mode.
- Step 4:** Configure the data frame:
- Start bit determination method: Configure UARTx\_CR1.START.
  - Data word length: Configure UARTx\_CR1.CHLEN.
  - Check bit: Configure UARTx\_CR1.PARITYEN and UARTx\_CR1.PARITY.
  - Stop bit: Configure UARTx\_CR1.STOP.
- Step 5:** Configure UARTx\_CR1.SOURCE and select the transmission clock source.
- Step 6:** Configure UARTx\_CR1.OVER and select sampling mode.
- Step 7:** Configure the UARTx\_BRR1 and UARTx\_BRRF registers and configure the baud rate. For details, please refer to [21.3.3.2 Fractional baud rate generator](#).
- Step 8:** Write 0x00 to UARTx\_ICR to clear all flags.
- Step 9:** Set UARTx\_CR1.TXEN to 1 and UARTx\_CR1.RXEN to 1 to enable transmit and receive.
- Step 10:** Query and wait for the UARTx\_ISR.RC flag to be set to 1 to confirm that one frame of data has been received.
- Step 11:** Query the error flags UARTx\_ISR.PE and UARTx\_ISR.FE to confirm whether the data is valid. If the data is invalid, perform error processing. If the data is valid, read the UARTx\_RDR register and save the data.
- Step 12:** Set UARTx\_ICR.RC to 0, clear the reception completion flag.
- Step 13:** Repeat steps 10~13 to receive the next frame of data.



## 21.8 List of registers

UART1 base address: UART1\_BASE = 0x4000 0C00

UART2 base address: UART2\_BASE = 0x4000 1000

UART3 base address: UART3\_BASE = 0x4000 2000

Table 21-13 List of UART registers

| Register name | Register address  | Register description                       |
|---------------|-------------------|--------------------------------------------|
| UARTx_CR1     | UARTx_BASE + 0x00 | Control register 1                         |
| UARTx_CR2     | UARTx_BASE + 0x04 | Control register 2                         |
| UARTx_CR3     | UARTx_BASE + 0x38 | Control register 3                         |
| UARTx_BRR1    | UARTx_BASE + 0x0C | Baud rate counter integer part register    |
| UARTx_BRRF    | UARTx_BASE + 0x10 | Baud rate counter fractional part register |
| UARTx_TIMARR  | UARTx_BASE + 0x14 | Timer reload register                      |
| UARTx_TIMCNT  | UARTx_BASE + 0x18 | Timer count register                       |
| UARTx_IER     | UARTx_BASE + 0x08 | Interrupt enable register                  |
| UARTx_ISR     | UARTx_BASE + 0x1C | Interrupt flag register                    |
| UARTx_ICR     | UARTx_BASE + 0x20 | Interrupt flag clear register              |
| UARTx_RDR     | UARTx_BASE + 0x24 | Receive data register                      |
| UARTx_TDR     | UARTx_BASE + 0x28 | Transmit data register                     |
| UARTx_ADDR    | UARTx_BASE + 0x30 | Slave address register                     |
| UARTx_MASK    | UARTx_BASE + 0x34 | Slave address mask register                |
| UARTx_RXMATCH | UARTx_BASE + 0x3C | Receive data match register                |



## 21.9 Register description

See section 1 [Documentation conventions](#) for abbreviations used in register descriptions.

### 21.9.1 UARTx\_CR1 Control register 1

Address offset: 0x00      Reset value: 0x0000 0000

| Bit field | Name   | Permission | Function description                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|-----------|--------|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:15     | RFU    | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 14        | SYNC   | RW         | Transmission sync-async mode settings<br>0: Working in asynchronous mode.<br>1: Working in synchronous mode.<br><i>Attention: In LIN mode, this control bit must be kept at 0.</i>                                                                                                                                                                                                                                                                                                      |
| 13:12     | SOURCE | RW         | UART transmission clock source configuration<br>00: UCLK from PCLK.<br>01: UCLK from PCLK.<br>10: UCLK from LSE.<br>11: UCLK from LSI.                                                                                                                                                                                                                                                                                                                                                  |
| 11        | SIGNAL | RW         | Single bus mode enable control<br>0: Disable, transmit data via TXD port, receive data via RXD port.<br>1: Enable, transmit and receive data through TXD port.<br><i>Attention: In LIN mode, this control bit must be kept at 0.</i>                                                                                                                                                                                                                                                    |
| 10:9      | OVER   | RW         | Sampling mode settings<br>00: 16 times sampling, Baud $UCLK/(BRRI \times 16 + BRRF)$ .<br>01: 8 times sampling, Baud $UCLK/(BRRI \times 8)$ .<br>10: 4 times sampling, Baud $UCLK/(BRRI \times 4)$ .<br>11: dedicated sampling, Baud $UCLK \times 256/BRRI$ .<br><i>Note: Dedicated sampling is only suitable for 2400, 4800, 9600 communication when the transmission clock is LSI/LSE.</i><br><i>Note: When BRRF is a non-zero value, 16 times sampling is automatically adopted.</i> |
| 8         | START  | RW         | RXD signal START bit determination mode settings<br>0: RXD signal falling edge.<br>1: RXD signal low level.                                                                                                                                                                                                                                                                                                                                                                             |
| 7         | MSBF   | RW         | Data frame high and low bit order selection<br>0: The least significant bit (LSB) is sent and received first.<br>1: The most significant bit (MSB) is sent and received first.                                                                                                                                                                                                                                                                                                          |
| 6         | CHLEN  | RW         | Character length<br>0: 8bit.<br>1: 9bit.<br><i>Note: If parity is enabled, the last 1Bit of the character is the parity bit.</i>                                                                                                                                                                                                                                                                                                                                                        |



| Bit field | Name     | Permission | Function description                                                                                                                                                                                                |
|-----------|----------|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 5:4       | STOP     | RW         | Stop bit length settings<br>00: 1 bit.<br>01: 1.5 bit.<br>10: 2 bit.<br>11: Reserved.<br><i>Attention: STOP bit must be cleared in synchronous mode.</i><br><i>Attention: STOP bit must be cleared in LIN mode.</i> |
| 3         | PARITYEN | RW         | Parity enable<br>0: Disabled<br>1: Enabled<br><i>Note: Parity enable is automatically cleared when the character length is 8bit.</i>                                                                                |
| 2         | PARITY   | RW         | Parity selection<br>0: Even check.<br>1: Odd check.                                                                                                                                                                 |
| 1         | RXEN     | RW         | Receive circuit enable control<br>0: Disabled<br>1: Enabled                                                                                                                                                         |
| 0         | TXEN     | RW         | Transmit circuit enable control<br>0: Disabled<br>1: Enabled                                                                                                                                                        |



### 21.9.2 UARTx\_CR2 Control register 2

Address offset: 0x04

Reset value: 0x0000 0000

| Bit field | Name  | Permission | Function description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|-----------|-------|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:18     | RFU   | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| 17:15     | RXSRC | RW         | RXD input signal source configuration<br>000: UARTx_RXD pin<br>001: VC1_OUT<br>010: VC2_OUT<br>011: VC3_OUT<br>100: VC4_OUT<br>Other: Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 14        | LOOP  | RW         | LoopBack mode enable control<br>0: RXD select external RXD Pin.<br>1: RXD select internal TXD.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 13        | ADCTX | RW         | Transmit buffer empty triggers ADC enable control<br>0: Disabled<br>1: Enabled                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 12        | ADCRX | RW         | Receive data completion trigger ADC enable control<br>0: Disabled<br>1: Enabled                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 11        | SWAP  | RW         | TXD/RXD pin-swap enable control<br>0: No pin-swap.<br>1: Pin swap.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 10:8      | TIMCR | RW         | Timer operating mode configuration<br>000: Stop timer immediately.<br>010: Wait timeout detection mode.<br><ul style="list-style-type: none"> <li>Start timer immediately starts counting from 0 and starts counting from 0 every time a STOP bit is received; count clock is baudrate clock; automatically stops after timeout.</li> </ul> 011: Receive idle detection mode.<br><ul style="list-style-type: none"> <li>Counts from 0 for each STOP bit received; count clock is baudrate clock; automatically stops after timeout.</li> </ul> 101: Auto baud rate detection mode 1<br><ul style="list-style-type: none"> <li>RXD falling edge causes timer to start counting from 0, rising edge stops timer; counting clock is UCLK; detection is completed automatically stops.</li> </ul> 110: Auto baud rate detection mode 2<br><ul style="list-style-type: none"> <li>RXD falling edge causes timer to start counting from 0, 4th falling edge stops timer; counting clock is UCLK; detection is completed automatically stops.</li> </ul> 111: General timing function<br><ul style="list-style-type: none"> <li>Start timer immediately starts counting from 0; count clock is UCLK; automatically stops after timeout (TIMARR+1) UCLK.</li> </ul> |



| Bit field | Name      | Permission | Function description                                                                                                                                                                              |
|-----------|-----------|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 7         | DMATX     | RW         | DMA data transmit enable control<br>0: Disabled<br>1: Enabled                                                                                                                                     |
| 6         | DMARX     | RW         | DMA data receive enable control<br>0: Disabled<br>1: Enabled                                                                                                                                      |
| 5         | TXINV     | RW         | TXD pin outputs signal inversion control<br>0: The TXD pin outputs the non-inverting signal.<br>1: The TXD pin outputs the inverted signal.                                                       |
| 4         | RXINV     | RW         | RXD pin inputs signal inversion control<br>0: The RXD pin signal is directly transmitted to the receiving circuit.<br>1: The RXD pin signal is inverted and transmitted to the receiving circuit. |
| 3         | RTSEN     | RW         | RTS hardware signal enable control<br>0: Disabled<br>1: Enabled                                                                                                                                   |
| 2         | CTSEN     | RW         | CTS hardware signal enable control<br>0: Disabled<br>1: Enabled                                                                                                                                   |
| 1         | RXMATCHEN | RW         | Receive data matching detection enable control<br>0: Disabled<br>1: Enabled                                                                                                                       |
| 0         | ADDREN    | RW         | Slave address identification enable control<br>0: Disabled<br>1: Enabled                                                                                                                          |



### 21.9.3 UARTx\_CR3 Control register 3

Address offset: 0x38      Reset value: 0x0000 0000

| Bit field | Name   | Permission | Function description                                                                                                                                                                                                                                                                                                                                                                                                          |
|-----------|--------|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:9      | RFU    | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                                                                                                                 |
| 8         | BRKL   | RW         | LIN mode, receive break field length threshold<br>0: 10bit.<br>1: 11bit.                                                                                                                                                                                                                                                                                                                                                      |
| 7         | LIN    | RW         | LIN operating mode enabled<br>0: Working in UART mode.<br>1: Working in LIN mode.                                                                                                                                                                                                                                                                                                                                             |
| 6:2       | DETIME | RW         | Driver enabled and disabled time<br>After outputting the DE active level and waiting for this time, the START bit starts to be transmitted.<br>After the STOP bit is transmitted and waiting for this time, output DE invalid level.<br>This time = Sampling time × DETIME<br><i>Note: The sampling time is 1/4, 1/8, 1/16 bit time.</i><br><i>Note: The DE function is not supported when OVER = 3 (dedicated sampling).</i> |
| 1         | DEP    | RW         | Driver output signal polarity configuration<br>0: DE signal is valid at high level.<br>1: DE signal is valid at low level.                                                                                                                                                                                                                                                                                                    |
| 0         | DEM    | RW         | Driver enabled configuration<br>0: Disable driver function, RTS pin for RTS function.<br>1: Enable driver function, RTS pin for DE function.<br><i>Note: This control bit has higher priority than UARTx_CR2.RTSEN.</i>                                                                                                                                                                                                       |

### 21.9.4 UARTx\_BRRI Baud rate counter integer part register

Address offset: 0x0C      Reset value: 0x0000 0000

| Bit field | Name | Permission | Function description                          |
|-----------|------|------------|-----------------------------------------------|
| 31:16     | RFU  | -          | Reserved bits, please keep the default value. |
| 15:0      | BRRI | RW         | Baud rate counter integer part                |



### 21.9.5 UARTx\_BRRF Baud rate counter fractional register

Address offset: 0x10      Reset value: 0x0000 0000

| Bit field | Name | Permission | Function description                          |
|-----------|------|------------|-----------------------------------------------|
| 31:4      | RFU  | -          | Reserved bits, please keep the default value. |
| 3:0       | BRRF | RW         | Baud rate counter fractional part             |

### 21.9.6 UARTx\_TIMARR Timer reload register

Address offset: 0x14      Reset value: 0x0000 0000

| Bit field | Name   | Permission | Function description                          |
|-----------|--------|------------|-----------------------------------------------|
| 31:24     | RFU    | -          | Reserved bits, please keep the default value. |
| 23:0      | TIMARR | RW         | Timer reload values                           |

### 21.9.7 UARTx\_TIMCNT Timer count register

Address offset: 0x18      Reset value: 0x0000 0000

| Bit field | Name   | Permission | Function description                          |
|-----------|--------|------------|-----------------------------------------------|
| 31:24     | RFU    | -          | Reserved bits, please keep the default value. |
| 23:0      | TIMCNT | RO         | Timer count values                            |



### 21.9.8 UARTx\_TDR Transmit data register

Address offset: 0x28      Reset value: 0x0000 0000

| Bit field | Name  | Permission | Function description                                                                                                                                                                                                                                       |
|-----------|-------|------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:11     | RFU   | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                              |
| 10        | BREAK | WO         | Transmit break frame<br>0: Transmit normal data.<br>1: Transmit break frame.<br><i>Note: The Break Field length for LIN mode is determined by the value of UARTx_TDR[8:0]; the Break Field length for UART mode is determined by the UART word length.</i> |
| 9         | IDLE  | WO         | Transmit idle frame<br>0: Transmit normal data.<br>1: Transmit idle frame.<br><i>Note: Idle frame length is determined by the UART word length.</i><br><i>Note: Idle frame has higher priority than Break Frame.</i>                                       |
| 8:0       | TDR   | WO         | Length of data to be transmitted or LIN mode Break Field<br><i>Note: When sending Break Field in LIN mode, the data written to this register is used as the number of bits in the Break Field.</i>                                                         |

### 21.9.9 UARTx\_RDR Receive data register

Address offset: 0x24      Reset value: 0x0000 0000

| Bit field | Name | Permission | Function description                          |
|-----------|------|------------|-----------------------------------------------|
| 31:9      | RFU  | -          | Reserved bits, please keep the default value. |
| 8:0       | RDR  | RO         | Received data                                 |

### 21.9.10 UARTx\_IER Interrupt enable register

Address offset: 0x08      Reset value: 0x0000 0000

| Bit field | Name    | Permission | Function description                                                     |
|-----------|---------|------------|--------------------------------------------------------------------------|
| 31:13     | RFU     | -          | Reserved bits, please keep the default value.                            |
| 12        | RXMATCH | RW         | Receive data match interrupt enable control<br>0: Disabled<br>1: Enabled |
| 11        | ORE     | RW         | Overflow interrupt enable control<br>0: Disabled<br>1: Enabled           |
| 10        | NE      | RW         | Noise interrupt enable control<br>0: Disabled<br>1: Enabled              |



| Bit field | Name   | Permission | Function description                                                                           |
|-----------|--------|------------|------------------------------------------------------------------------------------------------|
| 9         | PE     | RW         | Parity error interrupt enable control<br>0: Disabled<br>1: Enabled                             |
| 8         | FE     | RW         | Frame structure error interrupt enable control<br>0: Disabled<br>1: Enabled                    |
| 7         | CTS    | RW         | CTS signal change interrupt enable control<br>0: Disabled<br>1: Enabled                        |
| 6         | TIMOV  | RW         | Timer overflow interrupt enable control<br>0: Disabled<br>1: Enabled                           |
| 5         | BAUD   | RW         | Automatic baud rate detection completion interrupt enable control<br>0: Disabled<br>1: Enabled |
| 4         | RXBRK  | RW         | Break Field reception completion interrupt enable control<br>0: Disabled<br>1: Enabled         |
| 3         | RXIDLE | RW         | Idle character reception completion interrupt enable control<br>0: Disabled<br>1: Enabled      |
| 2         | RC     | RW         | Receive completion interrupt enable control<br>0: Disabled<br>1: Enabled                       |
| 1         | TC     | RW         | Transmit completion interrupt enable control<br>0: Disabled<br>1: Enabled                      |
| 0         | TXE    | RW         | Transmit buffer empty interrupt enable control<br>0: Disabled<br>1: Enabled                    |



### 21.9.11 UARTx\_ISR Interrupt flag register

Address offset: 0x1C      Reset value: 0x0000 0001

| Bit field | Name     | Permission | Function description                                                                                                                                                                                                                                                                                                                                             |
|-----------|----------|------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:16     | RFU      | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                                                    |
| 15        | CTSLV    | RO         | CTS signal level status<br>0: CTS signal is low level.<br>1: CTS signal is high level.                                                                                                                                                                                                                                                                           |
| 14        | TXBUSY   | RO         | TXD transmit status<br>0: TDR register and transmit shift register are empty.<br>1: TDR register or transmit shift register is not empty.                                                                                                                                                                                                                        |
| 13        | SLVMATCH | RO         | Receive slave address match flag<br>0: No matching address detected.<br>1: Matching address detected.                                                                                                                                                                                                                                                            |
| 12        | RXMATCH  | RO         | Receive data match flag<br>0: Received data is different from UARTx_RXMATCH.<br>1: Received data is the same as UARTx_RXMATCH.                                                                                                                                                                                                                                   |
| 11        | ORE      | RO         | Overflow flag<br>When UARTx_ISR.RC is 1, hardware sets this bit when the data currently being received in the shift register is ready for transmitting to the UARTx_RDR register.<br>0: No overflow error.<br>1: Overflow error detected.<br><i>Note: After an overflow error occurs, the newly received data overwrites the data in the UARTx_RDR register.</i> |
| 10        | NE       | RO         | Noise flag<br>0: No noise detected.<br>1: Noise detected.<br><i>Note: When OVER is 2 or 3, this bit is invalid and remains 0.</i>                                                                                                                                                                                                                                |
| 9         | PE       | RO         | Parity error interrupt flag<br>0: No parity error detected.<br>1: Parity error detected.                                                                                                                                                                                                                                                                         |
| 8         | FE       | RO         | Frame structure error interrupt flag<br>0: No frame structure errors detected.<br>1: Frame structure error detected.                                                                                                                                                                                                                                             |
| 7         | CTS      | RO         | CTS signal change interrupt flag<br>0: No CTS level change detected.<br>1: CTS level change detected.                                                                                                                                                                                                                                                            |
| 6         | TIMOV    | RO         | Timer overflow flag<br>0: Timer does not overflow (TIMCNT < TIMARR).<br>1: Timer has overflowed (TIMCNT ≥ TIMARR).                                                                                                                                                                                                                                               |



| Bit field | Name   | Permission | Function description                                                                                                                                                                                                                                                                                                      |
|-----------|--------|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 5         | BAUD   | RO         | Auto-baud rate detection complete flag<br>0: Automatic baud rate detection is not completed.<br>1: Automatic baud rate detection has been completed.                                                                                                                                                                      |
| 4         | RXBRK  | RO         | Break Field reception completion flag<br>0: No Break Field received.<br>1: Break Field has been received.                                                                                                                                                                                                                 |
| 3         | RXIDLE | RO         | Idle character reception completion flag<br>0: No idle characters received.<br>1: Idle character received.<br><i>Note: Idle character detection is initiated each time a character is received.</i>                                                                                                                       |
| 2         | RC     | RO         | Receive completion interrupt flag<br>0: One frame of data has not been received.<br>1: One frame of data has been received.<br><i>Note: STOP is 1bit, RC receives 1bit STOP bit field set to 1; STOP is 1.5bit, RC receives 0.5bit STOP bit field set to 1; STOP is 2bit, RC receives 0.5bit STOP bit field set to 1.</i> |
| 1         | TC     | RO         | Transmit completion interrupt flag<br>0: Data in transmit buffer or shift register has not been transmitted.<br>1: The data in the transmit buffer and shift register have been transmitted.                                                                                                                              |
| 0         | TXE    | RO         | Transmit buffer empty flag<br>0: There is data in the transmit buffer.<br>1: No data in transmit buffer.<br><i>Note: Writing to the UARTx_TDR register clears this flag bit.</i>                                                                                                                                          |



### 21.9.12 UARTx\_ICR Interrupt flag clear register

Address offset: 0x20

Reset value: 0x0000 1FFF

| Bit field | Name    | Permission | Function description                                                                                                               |
|-----------|---------|------------|------------------------------------------------------------------------------------------------------------------------------------|
| 31:13     | RFU     | -          | Reserved bits, please keep the default value.                                                                                      |
| 12        | RXMATCH | R1W0       | Receive data match flag clear control<br>W0: Clear received data match flag.<br>W1: No function.                                   |
| 11        | ORE     | R1W0       | Overflow flag clear control<br>W0: Clear overflow flag.<br>W1: No function.                                                        |
| 10        | NE      | R1W0       | Noise flag clear control<br>W0: Clear noise flag.<br>W1: No function.                                                              |
| 9         | PE      | R1W0       | Parity error interrupt flag clear control<br>W0: Clear parity error interrupt flag.<br>W1: No function.                            |
| 8         | FE      | R1W0       | Frame structure error interrupt flag clear control<br>W0: Clear frame structure error interrupt flag.<br>W1: No function.          |
| 7         | CTS     | R1W0       | CTS signal change interrupt flag clear control<br>W0: Clear CTS signal change interrupt flag.<br>W1: No function.                  |
| 6         | TIMOV   | R1W0       | Timer overflow flag clear control<br>W0: Clear timer overflow flag.<br>W1: No function.                                            |
| 5         | BAUD    | R1W0       | Automatic baud rate detection complete flag clear control<br>W0: Clear Auto-Baud rate detection complete flag.<br>W1: No function. |
| 4         | RXBRK   | R1W0       | Break Field receive completion flag clear control<br>W0: Clear Break Field reception completion flag.<br>W1: No function.          |
| 3         | RXIDLE  | R1W0       | Idle character receive completion flag clear control<br>W0: Clear idle character reception completion flag.<br>W1: No function.    |
| 2         | RC      | R1W0       | Receive completion interrupt flag clear control<br>W0: Clear receive completion interrupt flag.<br>W1: No function.                |
| 1         | TC      | R1W0       | Transmit completion interrupt flag clear control<br>W0: Clear transmit completion interrupt flag.<br>W1: No function.              |
| 0         | RFU     | -          | Reserved bits, please keep the default value.                                                                                      |



**21.9.13 UARTx\_ADDR Slave address register**

Address offset: 0x30      Reset value: 0x0000 0000

| Bit field | Name | Permission | Function description                          |
|-----------|------|------------|-----------------------------------------------|
| 31:8      | RFU  | -          | Reserved bits, please keep the default value. |
| 7:0       | ADDR | RW         | Slave address register                        |

**21.9.14 UARTx\_MASK Slave address mask register**

Address offset: 0x34      Reset value: 0x0000 0000

| Bit field | Name | Permission | Function description                          |
|-----------|------|------------|-----------------------------------------------|
| 31:8      | RFU  | -          | Reserved bits, please keep the default value. |
| 7:0       | MASK | RW         | Slave address mask register                   |

**21.9.15 UARTx\_RXMATCH Receive data match register**

Address offset: 0x3C      Reset value: 0x0000 0000

| Bit field | Name    | Permission | Function description                          |
|-----------|---------|------------|-----------------------------------------------|
| 31:9      | RFU     | -          | Reserved bits, please keep the default value. |
| 8:0       | RXMATCH | RW         | Target to be matched receives data            |



## 22 Serial peripheral interface (SPI)

### 22.1 Overview

Serial peripheral interface (SPI) is a synchronous serial data communication interface, which is often used for synchronous serial communication between MCU and external devices. CW32L012 integrates 3 serial peripheral SPI interfaces, supports bidirectional full-duplex, single-wire half-duplex and simplex communication modes, can configure MCU as master or slave, supports multi-master communication mode.





## 22.2 Main features

- Support master mode, slave mode.
- Support full-duplex, single-wire half-duplex, simplex.
- Selectable 4 to 16 bit data frame width.
- Support sending and receiving data LSB or MSB first.
- Programmable clock polarity and clock phase.
- Communication rates up to  $PCLK/2$  in master mode.
- Communication rates up to  $PCLK/2$  in slave mode.
- Support multi-computer communication mode.
- 8 interrupt sources with flag bits.
- Support direct memory access (DMA).

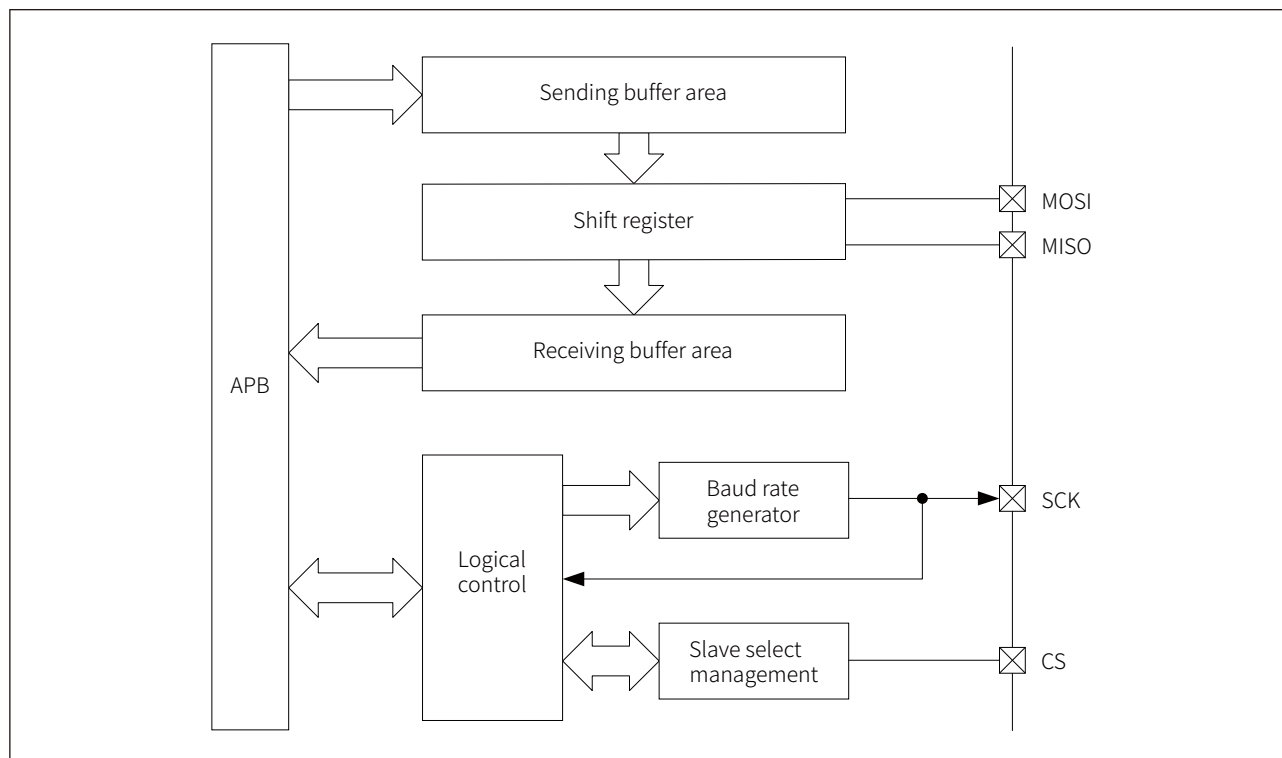


## 22.3 Functional description

### 22.3.1 Functional block diagram

The functional block diagram of the SPI controller is shown in the following figure:

Figure 22-1 SPI functional block diagram



SPI is generally connected to external devices through 4 pins:

1. MOSI  
Master output/slave input, used for data transmission in master mode and data reception in slave mode.
2. MISO  
Master input/slave output, used for data reception in master mode and data transmission in slave mode.
3. SCK  
Synchronous serial clock, master clock output and slave clock input, transmit or receive master synchronous clock.
4. CS  
Slave selection is also used for bus conflict detection in multi-computer communication. Refer to [22.3.3 Slave select pin CS](#).

The CW32L012 supports the user to flexibly select GPIO as the SPI communication pin, which is realized by multiplexing the AFR function, please refer to the pin definitions in the *Datasheet*.

SPI supports a variety of operating modes. In different operating modes, each pin has different functions. The pin configuration is shown in the following table:

Table 22-1 SPI ports configuration

| SPI pins | Mode configuration                   | Function                               | GPIO configuration                      |
|----------|--------------------------------------|----------------------------------------|-----------------------------------------|
| SPI_SCK  | Master                               | Serial clock output                    | Digital, push-pull output, multiplexed. |
|          | Slave                                | Serial clock input                     | Digital, floating input, multiplexed.   |
| SPI_MOSI | Full duplex/master                   | Master data output                     | Digital, push-pull output, multiplexed  |
|          | Full duplex/slave                    | Slave data input                       | Digital, floating input, multiplexed    |
|          | Single-wire half-duplex / master     | Master data transmitting and receiving | Digital, push-pull output, multiplexed  |
|          | Single-wire half-duplex / slave      | Do not use                             | Can be used as general purpose I/O      |
|          | Simplex single transmitting / master | Master data output                     | Digital, push-pull output, multiplexed  |
|          | Simplex single transmitting / slave  | Do not use                             | Can be used as general purpose I/O      |
|          | Simplex single receiving / master    | Do not use                             | Can be used as general purpose I/O      |
|          | Simplex single receiving / slave     | Slave data input                       | Digital, floating input, multiplexed    |
| SPI_MISO | Full duplex / master                 | Master data input                      | Digital, floating input, multiplexed    |
|          | Full duplex / slave                  | Slave data output                      | Digital, push-pull output, multiplexed  |
|          | Single-wire half-duplex / master     | Do not use                             | Can be used as general purpose I/O      |
|          | Single-wire half-duplex / slave      | Slave data transmission and reception  | Digital, push-pull output, multiplexed  |
|          | Simplex single transmitting / master | Do not use                             | Can be used as general purpose I/O      |
|          | Simplex single transmitting / slave  | Slave data output                      | Digital, push-pull output, multiplexed  |
|          | Simplex single receiving / master    | Master data output                     | Digital, push-pull output, multiplexed  |
|          | Simplex single receiving / slave     | Do not use                             | Can be used as general purpose I/O      |
| SPI_CS   | Master (SSM = 0)                     | Slave select input                     | Digital, floating input, multiplexed    |
|          | Master (SSM = 1)                     | Slave select output                    | Digital, push-pull output, multiplexed  |
|          | Slave (SSM = 0)                      | Slave select input                     | Digital, floating input, multiplexed    |
|          | Slave (SSM = 1)                      | Do not use                             | Can be used as general purpose I/O      |

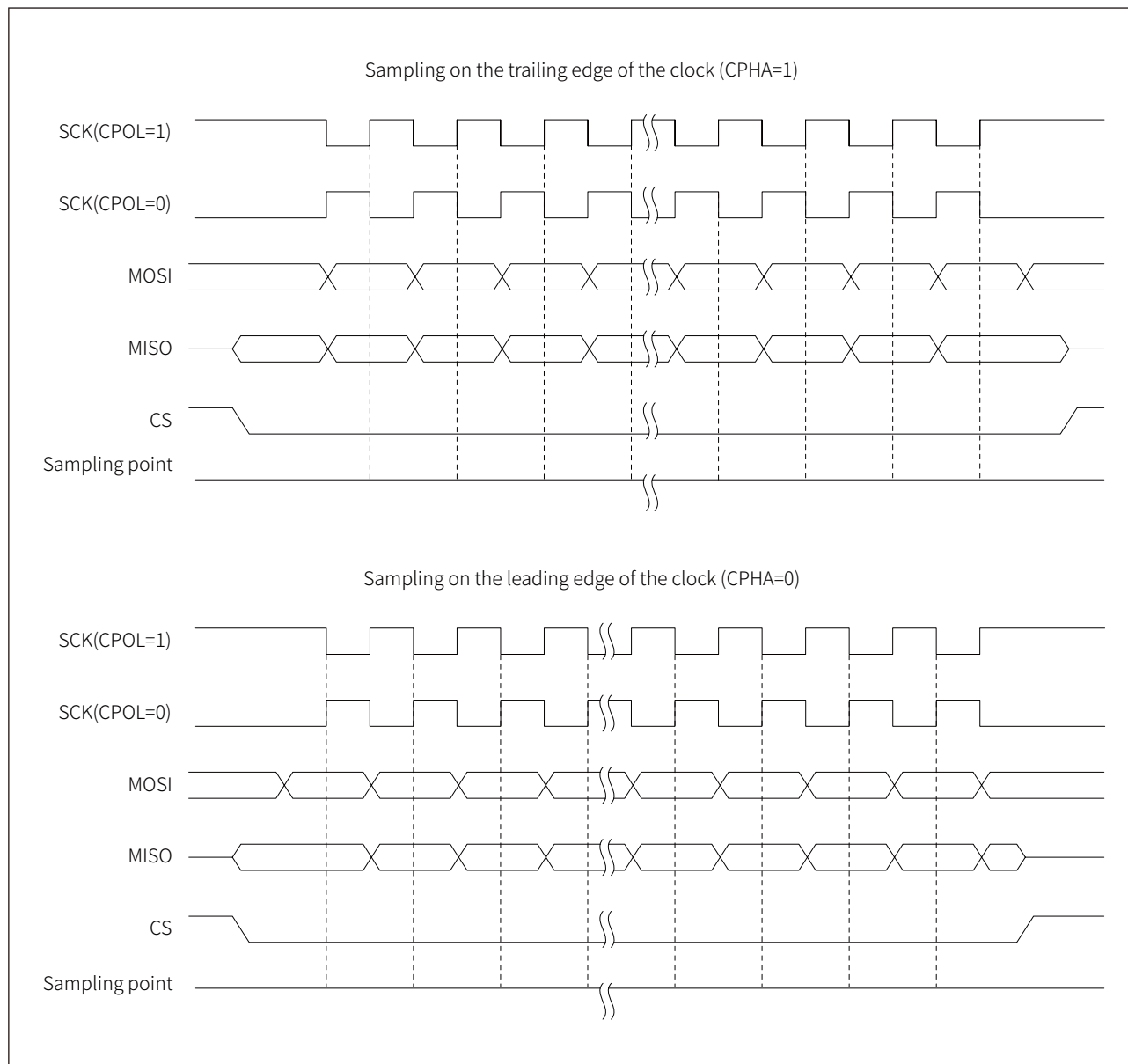


### 22.3.2 Communication timing and data format

#### Communication timing

The following figure shows the communication timing of SPI. The CS, SCK, and MOSI signals are all controlled by the master, and the MISO signal is the slave response signal. The slave selects the CS signal from high level to low level, which is the start signal of SPI communication. When the slave detects the start signal, it starts to communicate with the master. In each SCK clock cycle, the MOSI and MISO signal lines transmit one bit of data. CS changes from low level to high, which is the stop signal of communication, indicating the end of this communication.

Figure 22-2 SPI communication timing



### Data frame format

The data frame width is configured by the WIDTH bit field of the control register SPI\_CR1, which can be set to any data bit width from 4 to 16 bits.

The size of the data is configured by the LSBF bit field of the control register SPI\_CR1, and the most significant bit first (MSB) or the least significant bit first (LSB) can be selected.

The master mode frame interval is configured by the GAP bit field of control register SPI\_CR1, and can be selected from 0 to 15 SCK clock cycle intervals.

### Clock frequency

The synchronous serial clock SCK signal is controlled and generated by the SPI master, and its clock source is PCLK. The frequency division factor is set by configuring the BR bit field of the control register SPI\_CR1, and the frequency division can be selected from 2 to 256. For SPI slaves, configuring SPI\_CR1.BR has no effect.

When the SPI communication rate is significantly lower than the PCLK clock frequency ( $f_{SCK} \leq f_{PCLK}/16$ ), SPI\_CR1.FLTEN can be set to 1 to enable the SPI filter function.

### Clock polarity, clock phase

Clock polarity CPOL refers to the level state of the SCK serial clock line when the device is in an idle state without data transmission. Configure through the CPOL bit field of the control register SPI\_CR1: set SPI\_CR1.CPOL to 0, and the SCK clock line is low when it is idle; set SPI\_CR1.CPOL to 1, and the SCK clock line is high when it is idle.

The clock phase CPHA refers to the sampling and shifting moment of the data. Configure through the CPHA of the control register SPI\_CR1: set SPI\_CR1.CPHA to 0, sample on the front edge of SCK (the clock edge when SCK changes from idle state to non-idle state), and the trailing edge (SCK changes from non-idle state to idle state). Clock edge) shift; set SPI\_CR1.CPHA to 1, shift on the leading edge of SCK, and sample on the trailing edge.

According to the different configurations of clock polarity CPOL and clock phase CPHA, SPI can set 4 level modes. The master and slave need to be configured to the same level mode to ensure normal communication.

Table 22-2 SPI level mode

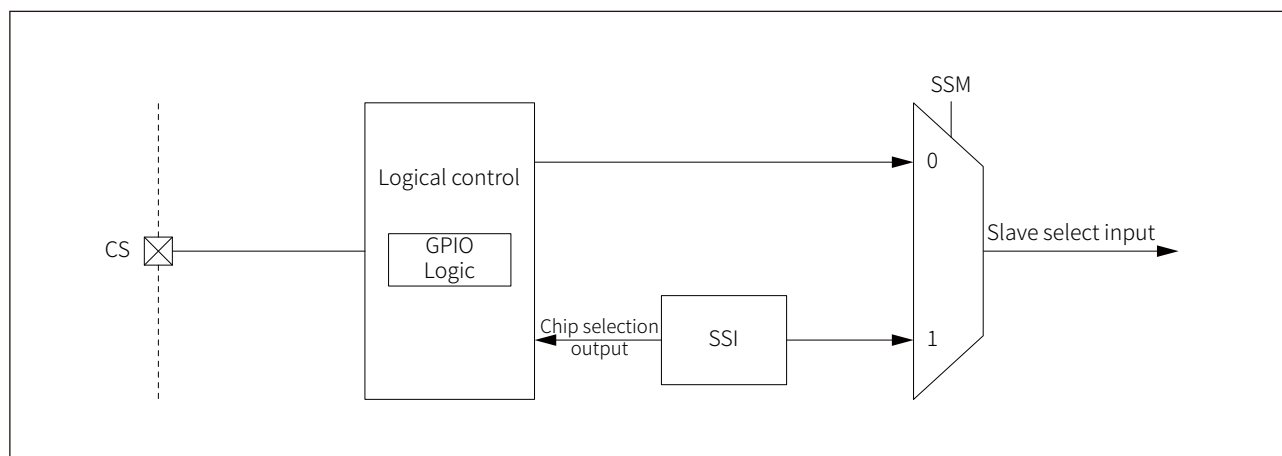
| CPOL | CPHA | SCK clock when idle | Sampling time |
|------|------|---------------------|---------------|
| 0    | 0    | Low level           | Front edge    |
| 0    | 1    | Low level           | Back edge     |
| 1    | 0    | High level          | Front edge    |
| 1    | 1    | High level          | Back edge     |



### 22.3.3 Slave select pin CS

The following figure shows a schematic diagram of the configuration management of slave select pin CS. In master mode, CS can be selected as input or output. When used as input, it is used to detect bus conflicts in multi-master mode, and when used as output, it is used to generate slave chip select signals. In slave mode, CS acts as a device chip select input.

Figure 22-3 Slave select pin management



In master and slave mode, the slave select CS is managed by the SSM bit field of the control register SPI\_CR1.

#### CS pin in master mode

Set SPI\_CR1.SSM to 0, and CS is configured as an input to detect whether there is a bus conflict in the multi-computer communication system (Caution that the master needs to select the slave to communicate through other GPIOs at this time), please refer to [22.3.7 Multi-machine communication](#).

Setting SPI\_CR1.SSM to 1, the slave select pin CS is configured as an output for generating the slave select signal. The CS pin output level is determined by the SSI bit field of the slave select register SPI\_SSI: set SPI\_SSI.SSI to 1, CS outputs a high level; set SPI\_SSI.SSI to 0, CS outputs a low level.

#### CS pin in slave mode

Set SPI\_CR1.SSM to 0 to use CS signal as device select. The CS level determines whether this unit is selected: when CS is low level, this unit is selected; when CS is high level, this unit is deselected.

Set SPI\_CR1.SSM to 1, do not use CS as the device selection, the slave select is determined by the SSI bit field value of the SPI\_SSI register. Set SPI\_SSI.SSI to 0, this machine is selected; set SPI\_SSI.SSI to 1, this machine is unchecked. In this configuration, the user program sets whether the machine is selected by software, and the CS pin can be used as a common IO.

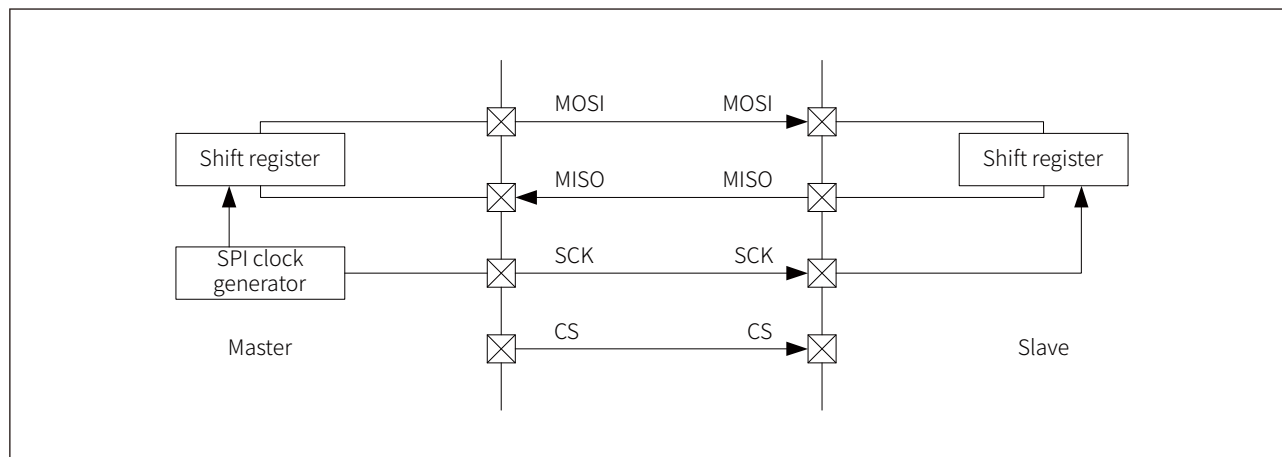
### 22.3.4 Full-duplex mode

SPI supports full-duplex communication mode. In this mode, the shift registers of the master and slave are connected through two unidirectional data lines, MOSI and MISO, and two lines of data are simultaneously transmitted under the control of the SCK clock signal provided by the master.

Set the MODE bit field of the control register SPI\_CR1 to 0x0 to make the SPI work in full-duplex communication mode.

A typical application block diagram of full-duplex mode is shown in the following figure:

Figure 22-4 Full-duplex communication mode



#### Master transmits and receives

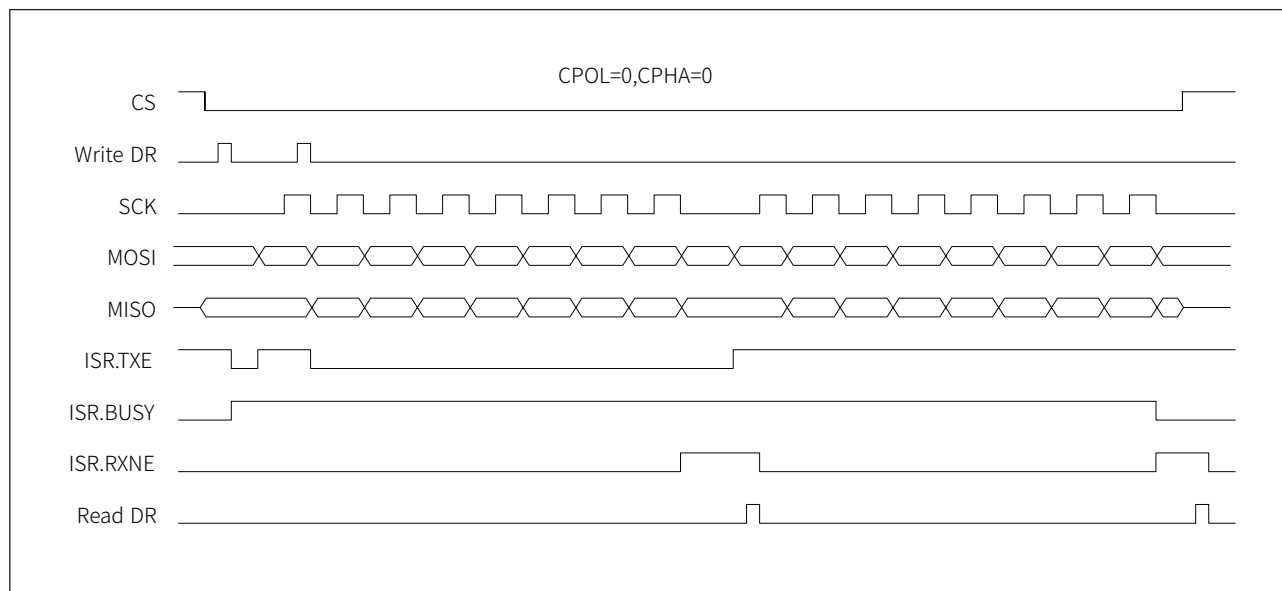
Set SPI\_CR1.MSTR to 1, SPI works in master mode.

The master sets SPI\_SSI.SSI to 0, and the slave selects the CS pin to output a low level as the communication start signal.

When the transmit buffer is empty, that is, the SPI\_ISR.TXE flag is 1, write a frame of data to be sent into the SPI\_DR register, the data is output from the MOSI pin under the control of the synchronous shift clock signal, and the MISO pin is simultaneously pin data is received into the shift register. At the end of transmitting a data frame, the non-empty flag bit SPI\_ISR.RXNE of the receive buffer is set to 1 by hardware, indicating that a frame of data has been received, and the SPI\_DR register can be read at this time.

Set SPI\_SSI.SSI to 1, and the slave select pin CS outputs a high level to end this communication.

Figure 22-5 Master transmits and receives timing in full-duplex mode (two bytes)



### Slave transmits and receives

Set SPI\_CR1.MSTR to 0, SPI works in slave mode.

Before the slave select CS signal is pulled low, the slave needs to set SPI\_ICR.FLUSH to 0 to clear the transmit buffer and shift register, and write the first frame data to be transmitted into the SPI\_DR register.

When the CS signal is pulled low, the written data will be output from the MISO pin under the control of the master's synchronous shift clock signal, and at the same time receive the data from the MOSI pin to the shift register.

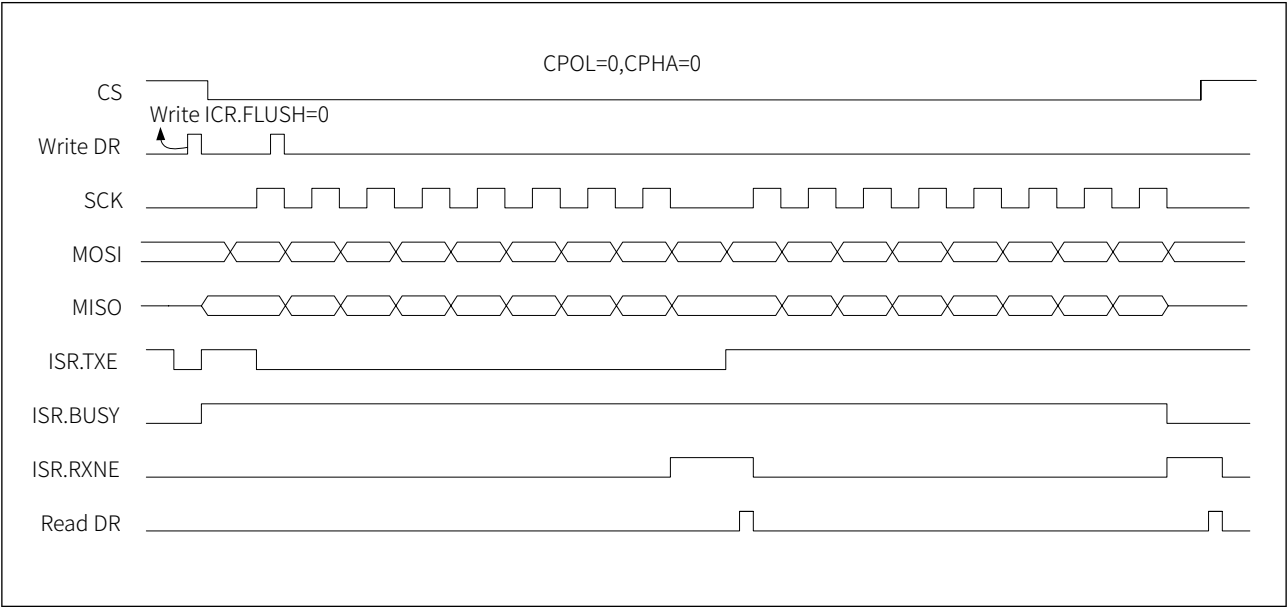
If it is continuous communication of multiple data frames, the user should continuously query the SPI\_ISR.TXE flag bit. Once the flag is 1, immediately write the data to be transmitted into the SPI\_DR register to avoid data leakage.

When the receive buffer is not empty, that is, the SPI\_ISR.RXNE flag is 1, indicating that a frame of data has been received, and the SPI\_DR register can be read at this time.

This communication ends when it is detected that the CS pin changes to a high level.



Figure 22-6 Slave transmit and receive timing in full-duplex mode (two bytes)



### 22.3.5 Single-wire half-duplex mode

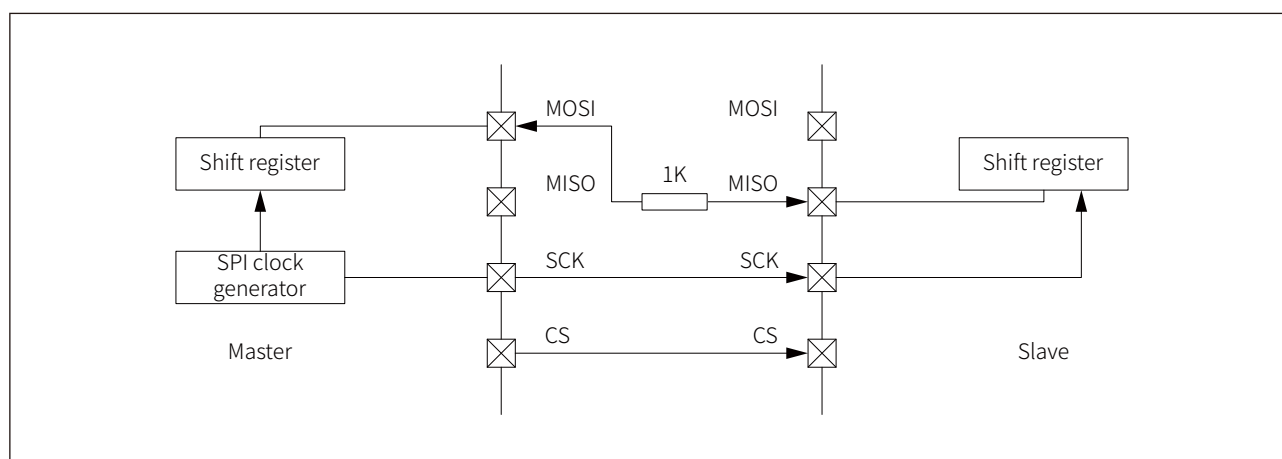
SPI supports single-wire half-duplex communication mode. In this mode, the master and slave communicate data through a bidirectional data line. The master uses MOSI, the slave uses MISO, and other unused SPI signal lines can be used by other functions.

Set the MODE bit field of the control register SPI\_CR1 to 0x3 to make the SPI work in single-wire half-duplex mode.

The HDOE bit field of the control register SPI\_CR3 selects the transmitting and receiving direction. When HDOE is 0, it is only received, and when HDOE is 1, it is only transmitted.

A typical application block diagram of single-wire half-duplex mode is shown in the following figure:

Figure 22-7 Single-wire half-duplex mode



During single-wire half-duplex communication, since the communication directions of the master and the slave cannot be guaranteed to switch synchronously, there may be a driving conflict and cause device damage. It is recommended to connect a resistor in series on the data line between the master and the slave to limit the current. .

When the SPI is used as a slave, the single-wire half-duplex mode can be used to communicate with a standard UART transceiver in synchronous mode. At this time, the SPI must be configured as a fixed level mode: the clock polarity CPOL is 1, and the clock phase is CPHA. is 1.

#### Master transmits

Set SPI\_CR1.MSTR to 1, SPI works in master mode; set SPI\_CR3.HDOE to 1, the master only transmits data.

The master sets SPI\_SSI.SSI to 0, and outputs a low level on the slave selection pin CS as a communication start signal.

When the transmit buffer is empty, that is, the SPI\_ISR.TXE flag bit is 1, a frame of data to be transmitted is written into the SPI\_DR register, and the data is output from the MOSI pin under the control of the synchronous shift clock signal.

After writing the last frame of data, you must wait for the transmit buffer to be empty, that is, the SPI\_ISR.TXE flag becomes 1, and the SPI\_ISR.BUSY flag becomes 0 to ensure that the data is transmitted. Then set SPI\_SSI.SSI to 1, select the CS pin to output a high level in the slave, and end this communication.

**Master receives**

Set SPI\_CR1.MSTR to 1, SPI works in master mode; set SPI\_CR3.HDOE to 0, the master only receives data.

The master sets SPI\_SSI.SSI to 0, and the slave selects the CS pin to output a low level as the communication start signal.

When the transmit buffer is empty, that is, the SPI\_ISR.TXE flag is 1, write a frame of dummy data to the SPI\_DR register to start the transmission.

When the receive buffer is not empty, that is, the SPI\_ISR.RXNE flag is 1, indicating that a frame of data has been received, and the SPI\_DR register can be read at this time.

If you want to receive multiple frames of data, repeat the above steps to write to SPI\_DR and read the received data from SPI\_DR.

When all data frames are received, set SPI\_SSI.SSI to 1, select the CS pin to output a high level in the slave, and end this communication.

**Slave transmits**

Set SPI\_CR1.MSTR to 0, SPI works in slave mode; set SPI\_CR3.HDOE to 1, the slave only transmits data.

Before the slave select CS signal is pulled low, the slave needs to set SPI\_ICR.FLUSH to 0 to clear the transmit buffer and shift register, and write the first frame data to be transmitted into the SPI\_DR register.

When the CS signal is pulled low, the written data will be output from the MISO pin under the control of the master's synchronous shift clock signal.

If it is continuous communication of multiple data frames, the user should continuously query the SPI\_ISR.TXE flag bit. Once the flag is 1, immediately write the data to be transmitted into the SPI\_DR register to avoid data leakage.

This communication ends when it is detected that the CS pin changes to a high level.

**Slave receives**

Set SPI\_CR1.MSTR to 0, SPI works in slave mode; set SPI\_CR3.HDOE to 0, the slave only receives data.

When the CS pin is detected to go low, the slave starts to communicate with the master.

When the receive buffer is not empty, that is, the SPI\_ISR.RXNE flag becomes 1, indicating that a frame of data has been received, and the SPI\_DR register can be read at this time.

This communication ends when it is detected that the CS pin changes to a high level.



### 22.3.6 Simplex mode

SPI supports simplex communication mode, and the master and slave communicate with single transmitting or single receiving through a unidirectional data line.

The master uses the MOSI signal line for single transmitting communication, and the MISO signal line for single receiving communication; the slave uses the MOSI signal line for single receiving communication, and the MISO signal line for single transmitting communication, and the unused signal line can be used for other functions.

Set the MODE bit field of the control register SPI\_CR1 to 0x1, and the SPI works in the simplex single transmitting communication mode; set the MODE bit field to 0x2, and the SPI works in the simplex single receiving communication mode.

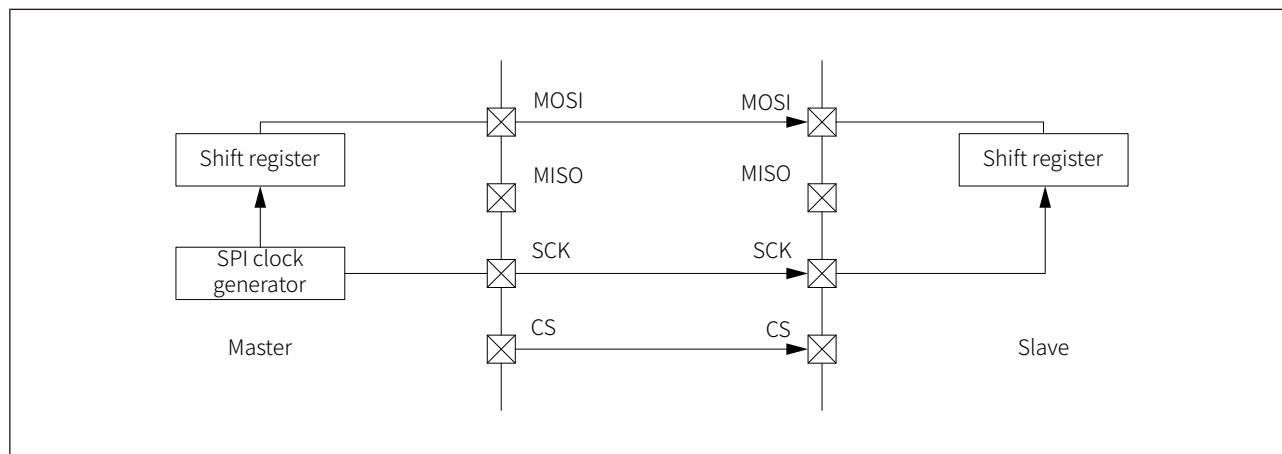
*Attention: In simplex single receiving mode, all events related to the receive stream on the transmitter side, such as the receive buffer overflow error flag bit SPI\_ISR.OV, must be ignored.*



### 22.3.6.1 Master single transmitting/slave single receiving

In the application scenario of single transmitting by the master and single receiving of the slave, the master and the slave use a MOSI data line to communicate. The application block diagram is shown in the following figure:

Figure 22-8 Master single transmitting, slave single receiving mode



#### Master single transmitting

Set SPI\_CR1.MODE to 0x1, SPI works in simplex single transmitting communication mode; set SPI\_CR1.MSTR to 1, SPI works in master mode.

Set SPI\_SSI.SSI to 0, and select the CS pin to output a low level in the slave as the communication start signal.

When the transmit buffer is empty, that is, the SPI\_ISR.TXE flag bit is 1, a frame of data to be sent is written into the SPI\_DR register, and the data is output from the MOSI pin under the control of the synchronous shift clock signal.

After writing the last frame of data, you must wait for transmit buffer empty flag SPI\_ISR.TXE to become 1, and the SPI bus busy flag SPI\_ISR.BUSY to become 0 to ensure that the data is transmitted. Then set SPI\_SSI.SSI to 1, so that the slave selects the CS pin to output a high level, and ends this communication.

#### Slave single receiving

Set SPI\_CR1.MODE to 0x2, SPI works in simplex single receiving communication mode; set SPI\_CR1.MSTR to 0, SPI works in slave mode.

When the CS pin is detected to go low, the slave starts to communicate with the master.

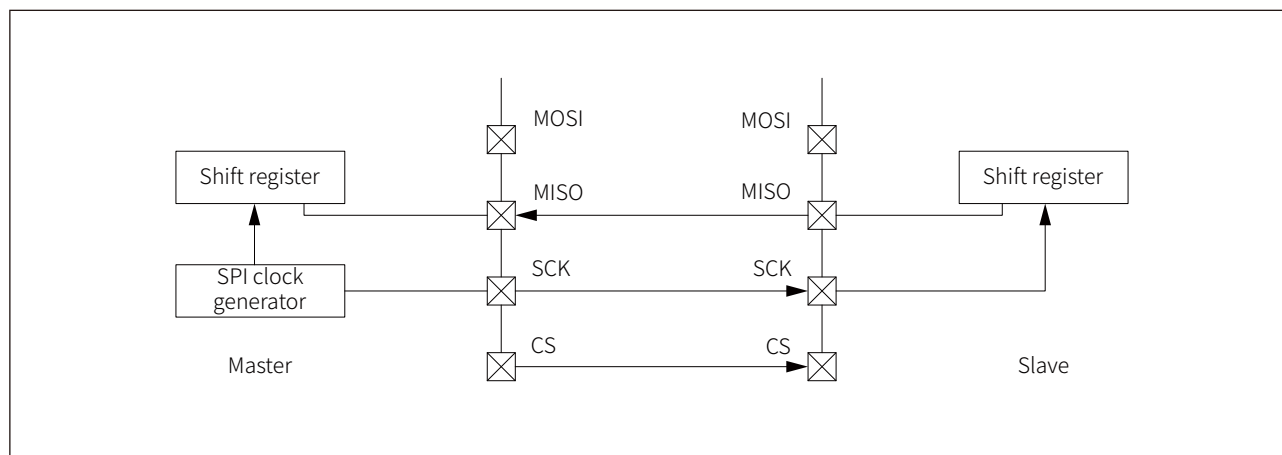
When the receive buffer is not empty, that is, the SPI\_ISR.RXNE flag is 1, indicating that a frame of data has been received, and the SPI\_DR register can be read at this time.

This communication ends when it is detected that the CS pin changes to a high level.

### 22.3.6.2 Master single receiving/slave single transmitting

In the application scenario of the master single receiving and the slave single transmitting, the master and the slave use a MISO data line to communicate. The application block diagram is shown in the following figure:

Figure 22-9 Master single receiving, slave single transmitting mode



#### Master single receiving

Set SPI\_CR1.MODE to 0x2, SPI works in simplex single receiving communication mode; set SPI\_CR1.MSTR to 1, SPI works in master mode.

Set SPI\_SSI.SSI to 0, and select the CS pin to output a low level in the slave as the communication start signal.

When the transmit buffer is empty, that is, the SPI\_ISR.TXE flag is 1, write a frame of dummy data to the SPI\_DR register to start the transmission.

When the receive buffer is not empty, that is, the SPI\_ISR.RXNE flag is 1, indicating that a frame of data has been received, and the SPI\_DR register can be read at this time.

If you want to receive multiple frames of data, repeat the above steps to write to SPI\_DR and read the received data from SPI\_DR.

When all data frames are received, set SPI\_SSI.SSI to 1, and the slave selects the CS pin to output a high level to end this communication.

#### Slave single transmitting

Set SPI\_CR1.MODE to 0x1, SPI works in simplex single transmitting communication mode; set SPI\_CR1.MSTR to 0, SPI works in slave mode.

Before the slave selection signal CS is pulled low, the slave needs to set SPI\_ICR.FLUSH to 0 to clear the transmit buffer and shift register, and write the first frame data to be transmitted into the SPI\_DR register.

When the CS signal is pulled low, the written data is output from the MISO pin under the control of the master's synchronous shift clock signal.

If it is continuous communication of multiple data frames, the user should continuously query the SPI\_ISR.TXE flag bit. Once the flag is 1, immediately write the data to be transmitted into the SPI\_DR register to avoid data leakage.

This communication ends when it is detected that the CS pin changes to a high level.

### 22.3.7 Multi-machine communication

SPI supports multi-machine communication mode. In this mode, the slave select CS pin of the master should be configured as an input and connected to the bus application signal of other masters to detect whether there is a conflict on the SPI bus.

If the slave select CS pin of a master is pulled to a low level, it means that another master is occupying the bus, the master will generate a mode error (please refer to [22.3.9 Error flag](#)), the master SPI will be disabled, and the SPI will be disabled. Avoid SPI bus conflicts.

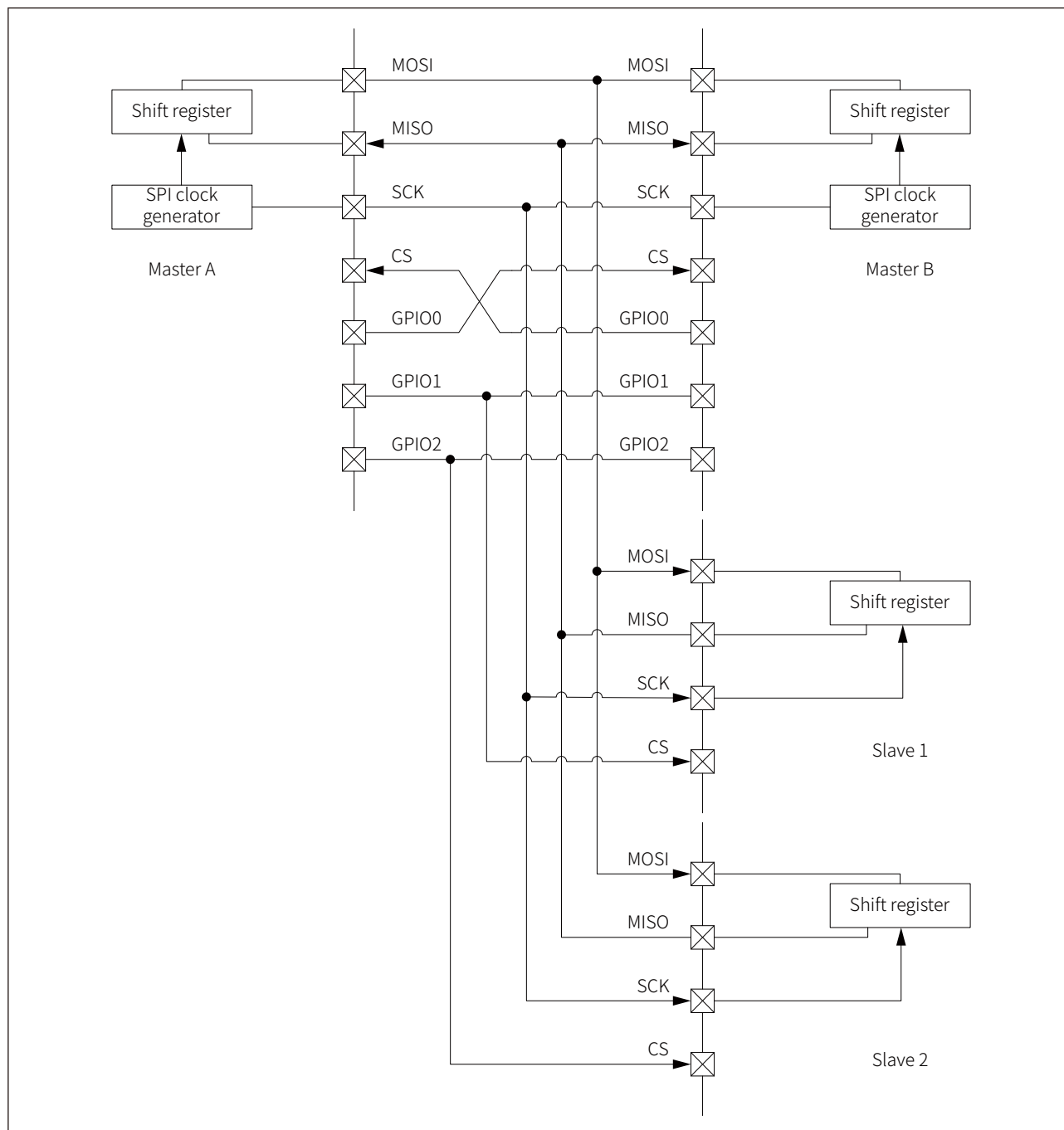
If the slave of a master selects the CS pin to detect a high level, it means that the SPI bus is idle. At this time, if the master wants to communicate with the slave, it must first apply for a signal on the bus to output a low level to other masters. CS signal, get SPI bus control, and then select a slave to start communication.

The following figure shows a typical multi-machine communication system. Taking the communication between master A and slave 1 as an example, the operation process is as follows.

- Step 1:** Master A detects the CS pin until a high level occurs.
- Step 2:** GPIO0 of master A outputs a low level to obtain bus control, and master B is disabled.
- Step 3:** GPIO1 of master A outputs low level, and slave 1 is selected.
- Step 4:** The master A communicates with the slave 1.
- Step 5:** GPIO1 of master A outputs a high level, releasing slave 1.
- Step 6:** GPIO0 of master A outputs a high level, releasing the SPI bus.



Figure 22-10 Multi-machine communication system



In order to ensure the communication reliability in the multi-machine communication mode, the MISOHD bit field of the slave control register SPI\_CR1 should be set to 1. When the slave is selected, the MISO pin is CMOS output; when it is not selected, the MISO pin is high resistance output.



### 22.3.8 Status flag

There are 6 status flags in the SPI controller to indicate the general working status of the SPI.

#### Transmit buffer empty flag (SPI\_ISR.TXE)

When the transmit data is transmitted from the SPI\_DR register to the shift register, the SPI\_ISR.TXE flag will be set by hardware, indicating that the transmit buffer is empty, and new data to be transmitted is allowed to be written to the SPI\_DR register. While writing data to the SPI\_DR register, the SPI\_ISR.TXE flag will be automatically cleared.

#### Receive buffer not empty flag (SPI\_ISR.RXNE)

When the received data is transmitted from the shift register to the SPI\_DR register, the SPI\_ISR.RXNE flag will be set by hardware, indicating that the reception of a frame of data has been completed. At this time, the SPI\_DR register can be read, and the SPI\_ISR will be cleared while reading the SPI\_DR register. RXNE flag bit.

Users can also manually clear the SPI\_ISR.RXNE flag by software.

#### Bus busy flag bit (SPI\_ISR.BUSY)

The SPI\_ISR.BUSY flag bit, set and cleared by hardware, is used to indicate the current SPI communication status.

In master mode, when preparing or performing data transmission (there is data in the transmit buffer or data in the shift register), the BUSY flag will be set by hardware, indicating that the SPI interface is in a busy state; When there is no data transmission (no data in the send buffer and no data in the shift register), the BUSY flag is automatically cleared by hardware, indicating that the SPI interface is in an idle state. When the user code inquires that the flag is 0, the CS signal can be pulled high to end the current round of communication.

*Note: This flag is not recommended in slave mode.*

#### Slave select signal status (SPI\_ISR.SSLVL)

The SPI\_ISR.SSLVL status bit, set and cleared by hardware, indicates the level status of the slave select input signal.

SPI\_ISR.SSLVL is 0, the slave select CS input is low, and the slave is selected; SPI\_ISR.SSLVL is 1, the slave select CS input is high, and the slave is not selected.

#### Slave select input rising edge flag (SPI\_ISR.SSR)

The SPI\_ISR.SSR flag bit is used to indicate whether the slave select CS input signal has a rising edge. It only makes sense when CS uses hardware pins in slave mode (MSTR = 0 and SSM = 0).

SPI\_ISR.SSR is 0, indicating that the slave select CS input has no rising edge; SPI\_ISR.SSR is 1, indicating that the slave select input has a rising edge.

#### Slave select input falling edge flag (SPI\_ISR.SSF)

The SPI\_ISR.SSF flag bit is used to indicate whether the slave select CS input signal has a falling edge. It only makes sense when CS uses hardware pins in slave mode (MSTR = 0 and SSM = 0).

SPI\_ISR.SSF is 0, indicating that the slave select CS input has no falling edge; SPI\_ISR.SSF is 1, indicating that the slave select CS input has a falling edge.



### 22.3.9 Error flag

There are 4 error flags in the SPI controller to indicate whether an error has occurred in the SPI.

#### Mode error flag (SPI\_ISR.MODF)

SPI\_ISR.MODF flag bit, used to detect bus conflict in SPI multi-machine communication mode.

When the SPI works in master mode and the slave select CS pin is configured as an input, if the CS pin input is low, a mode error will occur, and the SPI\_ISR.MODF flag is set by hardware, indicating that other SPI masters are occupying the bus at this time.

After a mode error occurs, SPI\_CR2.EN is cleared and the SPI is forced to shut down.

Setting SPI\_ICR.MODF to 0 can clear the SPI\_ISR.MODF flag. After clearing, the user needs to reconfigure the SPI controller.

After a mode error occurs, if the SPI controller is re-enabled, the SPI can still perform normal data transmission, but the SPI\_ISR.MODF flag will remain set until cleared by user software.

#### Slave select error flag in slave mode (SPI\_ISR.SSERR)

When the SPI is working in slave mode and a valid data transmit is in progress, if the slave select CS input is pulled high, a slave select error will occur, and the SPI\_ISR.SSERR flag will be set by hardware.

The occurrence of a slave select error will cause an error in the current data transmission, and the slave will enter the unselected state.

After a slave select error occurs, if the slave select CS input is pulled low again, the SPI can still perform normal data transmission, but the SPI\_ISR.SSERR flag will remain set until cleared by user software.

#### Receive buffer overflow error flag (SPI\_ISR.OV)

When the master or slave does not clear the SPI\_ISR.RXNE flag after receiving a frame of data, that is, neither read the SPI\_DR register nor clear the SPI\_ICR.RXNE to 0 by software, and then when a new frame of data is received, a receive buffer overflow error will occur, and the SPI\_ISR.OV flag is set by hardware.

After a receive buffer overflow error occurs, the newly received data will overwrite the original data in the buffer.

After a receive buffer overflow error occurs, the SPI can still perform normal data transmission, but the SPI\_ISR.OV flag will remain set until cleared by user software.

#### Transmit buffer underflow error flag in slave mode (SPI\_ISR.UD)

When the SPI works in slave mode, if the transmit buffer is empty and the transmission of a new frame has already started, a transmit buffer underflow error will occur, and the SPI\_ISR.UD flag will be set by hardware, indicating that the slave did not send the data in time. The SPI\_DR register writes data and misses transmitting data to the master.

After the transmit buffer underflow error occurs, the SPI can still perform normal data transmission, but the SPI\_ISR.UD flag will remain set until cleared by user software.

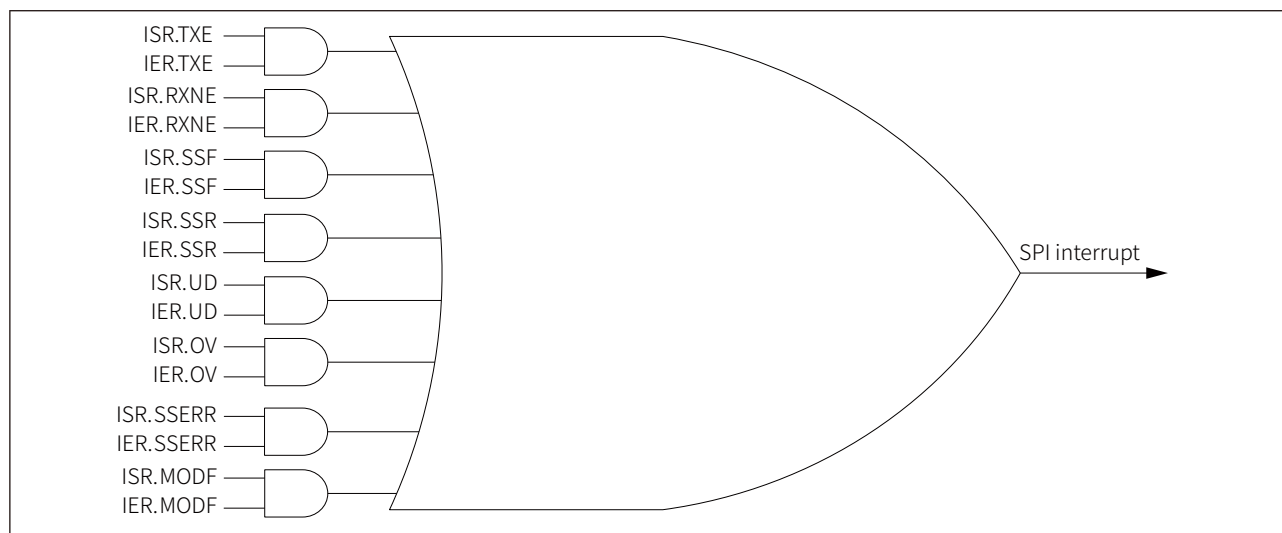


## 22.4 SPI interrupts

The SPI controller supports 8 interrupt sources. When the SPI interrupt trigger event occurs, the interrupt flag bit will be set by hardware. If the corresponding interrupt enable control bit is set, an interrupt request will be generated.

An SPI module of CW32L012 uses a same system SPI interrupt, whether the SPI interrupt generates an interrupt jump is controlled by the corresponding bit of the interrupt enable setting register NVIC\_ISER. The schematic diagram of the system SPI interrupt is shown in the following figure:

Figure 22-11 SPI interrupts



In the user's SPI interrupt service routine, the relevant SPI interrupt flag bit should be queried for corresponding processing. Before exiting the interrupt service routine, the interrupt flag bit should be cleared to avoid repeatedly entering the interrupt routine.

The flag bit, interrupt enable bit, interrupt flag clear bit and clear method of each interrupt source of SPI are shown in the following table:

Table 22-3 SPI interrupt control

| Interrupt event                                | Interrupt flag bit | Interrupt enable bit | Flag clear method                             |
|------------------------------------------------|--------------------|----------------------|-----------------------------------------------|
| Transmit buffer empty                          | ISR.TXE            | IER.TXE              | Write SPI_DR register.                        |
| Receive buffer non-empty                       | ISR.RXNE           | IER.RXNE             | Read SPI_DR register, or write 0 to ICR.RXNE. |
| Falling edge appears on the slave select input | ISR.SSF            | IER.SSF              | Write 0 to ICR.SSF.                           |
| Rising edge appears on the slave select input  | ISR.SSR            | IER.SSR              | Write 0 to ICR.SSR.                           |
| Transmit buffer underflow in slave mode        | ISR.UD             | IER.UD               | Write 0 to ICR.UD.                            |
| Receive buffer overflow                        | ISR.OV             | IER.OV               | Write 0 to ICR.OV.                            |
| Slave select error in slave mode               | ISR.SSERR          | IER.SSERR            | Write 0 to ICR.SSERR.                         |
| Mode error                                     | ISR.MODF           | IER.MODF             | Write 0 to ICR.MODF.                          |

## 22.5 Programming examples

### 22.5.1 Full duplex/master mode

#### 22.5.1.1 Transmit and receive in query mode

- Step 1:** Set `SYSCTRL_AHBEN.GPIOx` to 1, `SYSCTRL_APBEN1.SPI` to 1, and enable the GPIO clock and SPI working clock corresponding to the SPI pin.
- Step 2:** Configure the `SPI_SCK`, `SPI_MOSI` and `SPI_CS` pins as push-pull multiplexed output mode, and configure the `SPI_MISO` pin as floating input and multiplexed mode. For specific register configuration steps, please refer to chapter [9 General-purpose input/output \(GPIO\)](#).
- Step 3:** Set `SPI_CR1.MODE` to 0x0, and configure SPI to be bidirectional full-duplex communication mode.
- Step 4:** Set `SPI_CR1.MSTR` to 1, and configure SPI to master mode.
- Step 5:** Configure `SPI_CR1.WIDTH` and set the width of each frame of data.
- Step 6:** Configure `SPI_CR1.LSBF` and set the data bit sending and receiving sequence.
- Step 7:** Configure `SPI_CR1.CPOL` and set the clock polarity.
- Step 8:** Configure `SPI_CR1.CPHA` and set the clock phase.
- Step 9:** Set `SPI_CR1.SSM` to 1, and determine the slave select output value through the SSI register.
- Step 10:** Configure `SPI_CR1.BR` and set the SCK baud rate.
- Step 11:** Set `SPI_CR2.EN` to 1 to enable the SPI controller.
- Step 12:** Set `SPI_SSI.SSI` to 0, communication starts.
- Step 13:** Query and wait for the `SPI_ISR.TXE` flag to be set to 1, and confirm that the transmit buffer is empty.
- Step 14:** Write a frame of data to be transmitted into the `SPI_DR` register.
- Step 15:** Query and wait for the `SPI_ISR.RXNE` flag to be set to 1 to confirm that one frame of data has been received.
- Step 16:** Read the `SPI_DR` register and save the data.
- Step 17:** Repeat steps 13~17 to send and receive the next frame of data until all data is transmitted and received.
- Step 18:** Set `SPI_SSI.SSI` to 1, communication ends.



### 22.5.1.2 Transmit and receive in interrupt mode

- Step 1:** Set `SYSCTRL_AHBEN.GPIOx` to 1, `SYSCTRL_APBEN1.SPI` to 1, and enable the GPIO clock and SPI working clock corresponding to the SPI pin.
- Step 2:** Configure the `SPI_SCK`, `SPI_MOSI` and `SPI_CS` pins as push-pull multiplexed output mode, and configure the `SPI_MISO` pin as floating input and multiplexed mode. For specific register configuration steps, please refer to chapter [9 General-purpose input/output \(GPIO\)](#).
- Step 3:** Set `SPI_CR1.MODE` to 0x0, and configure SPI to be bidirectional full-duplex communication mode.
- Step 4:** Set `SPI_CR1.MSTR` to 1, and configure SPI to master mode.
- Step 5:** Configure `SPI_CR1.WIDTH` and set the width of each frame of data.
- Step 6:** Configure `SPI_CR1.LSBF` and set the data bit sending and receiving sequence.
- Step 7:** Configure `SPI_CR1.CPOL` and set the clock polarity.
- Step 8:** Configure `SPI_CR1.CPHA` and set the clock phase.
- Step 9:** Set `SPI_CR1.SSM` to 1, and determine the slave selection output value through the SSI register.
- Step 10:** Configure `SPI_CR1.BR` and set the SCK baud rate.
- Step 11:** Set `SPI_CR2.EN` to 1 to enable the SPI controller.
- Step 12:** Write 0x00 to `SPI_ICR` to clear all flags.
- Step 13:** Configure the NVIC controller, please refer to chapter [5 Interrupt](#).
- Step 14:** Set `SPI_SSI.SSI` to 0, communication starts.
- Step 15:** Set `SPI_IER.RXNE` to 1 to enable receive buffer non-empty interrupt.
- Step 16:** Set `SPI_IER.TXE` to 1 to enable transmit buffer empty interrupt.
- Step 17:** Enter the interrupt service function: query and judge the `SPI_ISR.TXE` flag bit, if the flag bit is 1, write a frame of data to be transmitted into the `SPI_DR` register; query and judge the `SPI_ISR.RXNE` flag bit, if the flag bit is 1, then read the `SPI_DR` register and save the data.
- Step 18:** Set `SPI_SSI.SSI` to 1, communication ends.



## 22.5.2 Single-wire half-duplex/master mode

### 22.5.2.1 Transmit in query mode

- Step 1: Set `SYSCTRL_AHBEN.GPIOx` to 1, `SYSCTRL_APBEN1.SPI` to 1, and enable the GPIO clock and SPI working clock corresponding to the SPI pin.
- Step 2: Configure the `SPI_SCK`, `SPI_MOSI` and `SPI_CS` pins as push-pull multiplexed output mode. For specific register configuration steps, please refer to chapter 9 [General-purpose input/output \(GPIO\)](#).
- Step 3: Set `SPI_CR1.MODE` to 0x3, and configure SPI to be single-wire half-duplex communication mode.
- Step 4: Set `SPI_CR1.MSTR` to 1, and configure SPI to master mode.
- Step 5: Configure `SPI_CR1.WIDTH` and set the width of each frame of data.
- Step 6: Configure `SPI_CR1.LSBF` and set the data bit transmitting and receiving sequence.
- Step 7: Configure `SPI_CR1.CPOL` and set the clock polarity.
- Step 8: Configure `SPI_CR1.CPHA` and set the clock phase.
- Step 9: Set `SPI_CR1.SSM` to 1, and determine the slave select output value through the SSI register.
- Step 10: Configure `SPI_CR1.BR` and set the SCK baud rate.
- Step 11: Set `SPI_CR2.EN` to 1 to enable the SPI controller.
- Step 12: Set `SPI_SSI.SSI` to 0, communication starts.
- Step 13: Set `SPI_CR3.HDOE` to 1, only transmit.
- Step 14: Query and wait for the `SPI_ISR.TXE` flag to be set to 1, and confirm that the transmit buffer is empty.
- Step 15: Write a frame of data to be transmitted into the `SPI_DR` register.
- Step 16: If there is still data to be transmitted, repeat steps 14-16 to transmit the next frame of data.
- Step 17: Query and wait for the `SPI_ISR.TXE` flag bit to be 1, and the last frame of data starts to be transmitted.
- Step 18: Query and wait for the `SPI_ISR.BUSY` flag to become 0, and the SPI bus is idle.
- Step 19: Set `SPI_SSI.SSI` to 1, communication ends.



### 22.5.2.2 Receive in query mode

- Step 1: Set SYSCTRL\_AHBEN.GPIOx to 1, SYSCTRL\_APBEN1.SPI to 1, and enable the GPIO clock and SPI working clock corresponding to the SPI pin.
- Step 2: Configure the SPI\_SCK, SPI\_MOSI and SPI\_CS pins as push-pull multiplexed output mode. For specific register configuration steps, please refer to chapter 9 [General-purpose input/output \(GPIO\)](#).
- Step 3: Set SPI\_CR1.MODE to 11, and configure SPI to be single-wire half-duplex communication mode.
- Step 4: Set SPI\_CR1.MSTR to 1, and configure SPI to master mode.
- Step 5: Configure SPI\_CR1.WIDTH and set the width of each frame of data.
- Step 6: Configure SPI\_CR1.LSBF and set the data bit transmitting and receiving sequence.
- Step 7: Configure SPI\_CR1.CPOL and set the clock polarity.
- Step 8: Configure SPI\_CR1.CPHA and set the clock phase.
- Step 9: Set SPI\_CR1.SSM to 1, and determine the slave select output value through the SSI register.
- Step 10: Configure SPI\_CR1.BR and set the SCK baud rate.
- Step 11: Set SPI\_CR2.EN to 1 to enable the SPI controller.
- Step 12: Set SPI\_SSI.SSI to 0, communication starts.
- Step 13: Set SPI\_CR3.HDOE to 0, only receive.
- Step 14: Query and wait for the SPI\_ISR.TXE flag to be set to 1, and confirm that the transmit buffer is empty.
- Step 15: Write a frame of dummy data into the SPI\_DR register.
- Step 16: Query and wait for the SPI\_ISR.RXNE flag to be set to 1 to confirm that one frame of data has been received.
- Step 17: Read the SPI\_DR register and save the data.
- Step 18: Repeat steps 14~18 to receive the next frame of data until all data is received.
- Step 19: Set SPI\_SSI.SSI to 1, communication ends.



## 22.5.3 Simplex mode/master mode

### 22.5.3.1 Transmit in query mode

- Step 1: Set `SYSCTRL_AHBEN.GPIOx` to 1, `SYSCTRL_APBEN1.SPI` to 1, and enable the GPIO clock and SPI working clock corresponding to the SPI pin.
- Step 2: Configure the `SPI_SCK`, `SPI_MOSI` and `SPI_CS` pins as push-pull multiplexed output mode. For specific register configuration steps, please refer to chapter 9 [General-purpose input/output \(GPIO\)](#).
- Step 3: Set `SPI_CR1.MODE` to 01, and configure SPI as simplex single-transmit communication mode.
- Step 4: Set `SPI_CR1.MSTR` to 1, and configure SPI to master mode.
- Step 5: Configure `SPI_CR1.WIDTH` and set the width of each frame of data.
- Step 6: Configure `SPI_CR1.LSBF` and set the data bit transmitting and receiving sequence.
- Step 7: Configure `SPI_CR1.CPOL` and set the clock polarity.
- Step 8: Configure `SPI_CR1.CPHA` and set the clock phase.
- Step 9: Set `SPI_CR1.SSM` to 1, and determine the slave select output value through the SSI register.
- Step 10: Configure `SPI_CR1.BR` and set the SCK baud rate.
- Step 11: Set `SPI_CR2.EN` to 1 to enable the SPI controller.
- Step 12: Set `SPI_SSI.SSI` to 0, communication starts.
- Step 13: Query and wait for the `SPI_ISR.TXE` flag to be set to 1, and confirm that the transmit buffer is empty.
- Step 14: Write a frame of data to be transmitted into the `SPI_DR` register.
- Step 15: If there is still data to be transmitted, repeat steps 13~15 to transmit the next frame of data.
- Step 16: Query and wait for the `SPI_ISR.TXE` flag bit to be 1, and the last frame of data starts to be transmitted.
- Step 17: Query and wait for the `SPI_ISR.BUSY` flag to become 0, and the SPI bus is idle.
- Step 18: Set `SPI_SSI.SSI` to 1, communication ends.





### 22.5.3.2 Receive in query mode

- Step 1: Set SYSCTRL\_AHBEN.GPIOx to 1, SYSCTRL\_APBEN1.SPI to 1, and enable the GPIO clock and SPI working clock corresponding to the SPI pin.
- Step 2: Configure the SPI\_SCK and SPI\_CS pins as push-pull multiplexing output mode, and configure the SPI\_MISO pin as floating input and multiplexing mode. For specific register configuration steps, please refer to chapter 9 [General-purpose input/output \(GPIO\)](#).
- Step 3: Set SPI\_CR1.MODE to 10, and configure SPI as simplex single receiving communication mode.
- Step 4: Set SPI\_CR1.MSTR to 1, and configure SPI to master mode.
- Step 5: Configure SPI\_CR1.WIDTH and set the width of each frame of data.
- Step 6: Configure SPI\_CR1.LSBF and set the data bit transmitting and receiving sequence.
- Step 7: Configure SPI\_CR1.CPOL and set the clock polarity.
- Step 8: Configure SPI\_CR1.CPHA and set the clock phase.
- Step 9: Set SPI\_CR1.SSM to 1, and determine the slave select output value through the SSI register.
- Step 10: Configure SPI\_CR1.BR and set the SCK baud rate.
- Step 11: Set SPI\_CR2.EN to 1 to enable the SPI controller.
- Step 12: Set SPI\_SSI.SSI to 0, communication starts.
- Step 13: Query and wait for the SPI\_ISR.TXE flag to be set to 1, and confirm that the transmit buffer is empty.
- Step 14: Write a frame of dummy data into the SPI\_DR register.
- Step 15: Query and wait for the SPI\_ISR.RXNE flag to be set to 1 to confirm that one frame of data has been received.
- Step 16: Read the SPI\_DR register and save the data.
- Step 17: Repeat steps 13~17 to receive the next frame of data until all data is received.
- Step 18: Set SPI\_SSI.SSI to 1, communication ends.



## 22.6 List of registers

SPI base address: SPI\_BASE = 0x4000 0800

Table 22-4 List of SPI registers

| Register name | Register address | Register description          |
|---------------|------------------|-------------------------------|
| SPI_CR1       | SPI_BASE + 0x00  | Control register 1            |
| SPI_CR2       | SPI_BASE + 0x04  | Control register 2            |
| SPI_CR3       | SPI_BASE + 0x08  | Control register 3            |
| SPI_SSI       | SPI_BASE + 0x0C  | Slave select register         |
| SPI_IER       | SPI_BASE + 0x10  | Interrupt enable register     |
| SPI_ISR       | SPI_BASE + 0x14  | Interrupt flag register       |
| SPI_ICR       | SPI_BASE + 0x18  | Interrupt flag clear register |
| SPI_DR        | SPI_BASE + 0x1C  | Data register                 |



## 22.7 Register description

See section 1 [Documentation conventions](#) for abbreviations used in register descriptions.

### 22.7.1 SPI\_CR1 Control register 1

Address offset: 0x00      Reset value: 0x0000 0700

| Bit field | Name   | Permission | Function description                                                                                                                                                                                                      |
|-----------|--------|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31        | RFU    | -          | Reserved bits, please keep the default value.                                                                                                                                                                             |
| 30:24     | BR     | RW         | Master mode SCK baud rate is configured as N, where N ranges from 0 to 127, $SCK = PCLK/(2 \times (n+1))$ , for example:<br>0: $SCK = PCLK/2$<br>1: $SCK = PCLK/4$<br>2: $SCK = PCLK/6$<br>.....<br>127: $SCK = PCLK/256$ |
| 23:21     | RFU    | -          | Reserved bits, please keep the default value.                                                                                                                                                                             |
| 20        | SMP    | RW         | Master mode delayed sampling<br>0: Normal sampling mode, the sampling time is determined by CPOL and CPHA.<br>1: Delayed sampling mode, sampling time is delayed about 20ns compared to normal sampling mode.             |
| 19        | MISOHD | RW         | Slave MISO output configuration<br>0: MISO is always COMS output.<br>1: MISO is COMS output when the slave is selected, high resistance output when it is unselected.                                                     |
| 18        | FLTEN  | RW         | Filter function enabled<br>0: Disabled<br>1: Enabled<br><i>Note: It is only used when the SPI communication rate is significantly lower than the PCLK clock frequency (<math>f_{SCK} \leq f_{PCLK}/16</math>).</i>        |
| 17:16     | MODE   | RW         | Communication mode configuration<br>00: Bidirectional full duplex.<br>01: Simplex single transmitting.<br>10: Simplex single receiving.<br>11: Single-wire half-duplex.                                                   |
| 15:12     | GAP    | RW         | Master mode frame gap<br>0000: No gap<br>0001: $1T_{SCK}$<br>0010: $2T_{SCK}$<br>.....<br>1111: $15T_{SCK}$                                                                                                               |



| Bit field | Name  | Permission | Function description                                                                                                                                                                                                                                                                                                                                                                                        |
|-----------|-------|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 11:8      | WIDTH | RW         | The width of each frame of data is WIDTH+1, for example:<br>0011: 4bit<br>0100: 5bit<br>...<br>1110: 15bit<br>1111: 16bit<br><i>Note: Write 0~2 automatically becomes 8 bits.</i>                                                                                                                                                                                                                           |
| 7         | LSBF  | RW         | Data frame high and low sequence selection<br>0: Most significant bit (MSB) is sent and received first.<br>1: Least significant bit (LSB) is sent and received first.                                                                                                                                                                                                                                       |
| 6:4       | RFU   | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                                                                                               |
| 3         | CPOL  | RW         | Serial clock polarity configuration<br>0: Low level in standby.<br>1: High level in standby.                                                                                                                                                                                                                                                                                                                |
| 2         | CPHA  | RW         | Serial clock phase configuration<br>0: Front edge sampling/back edge shifting.<br>1: Front edge shifting/back edge sampling.                                                                                                                                                                                                                                                                                |
| 1         | SSM   | RW         | <b>Slave select configuration (master mode)</b><br>0: SPI_CS pin is in input mode, pin being at low level to select the local machine.<br>1: SPI_CS pin is in output mode, output level from SSI.<br><b>Slave select configuration (slave mode)</b><br>0: SPI_CS pin level determines whether this machine is selected or not.<br>1: SSI register value determines whether this machine is selected or not. |
| 0         | MSTR  | RW         | Operating mode configuration<br>0: Slave mode.<br>1: Master mode.                                                                                                                                                                                                                                                                                                                                           |

*Attention: This register can only be modified if SPI\_CR2.EN is 0.*



### 22.7.2 SPI\_CR2 Control register 2

Address offset: 0x04      Reset value: 0x0000 0000

| Bit field | Name  | Permission | Function description                                              |
|-----------|-------|------------|-------------------------------------------------------------------|
| 31:3      | RFU   | -          | Reserved bits, please keep the default value.                     |
| 2         | DMATX | RW         | DMA data transmission enable control<br>0: Disabled<br>1: Enabled |
| 1         | DMARX | RW         | DMA data reception enable control<br>0: Disabled<br>1: Enabled    |
| 0         | EN    | RW         | Enable control<br>0: Disable SPI module.<br>1: Enable SPI module. |

### 22.7.3 SPI\_CR3 Control register 3

Address offset: 0x08      Reset value: 0x0000 0000

| Bit field | Name | Permission | Function description                                                                                                                                                                             |
|-----------|------|------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:1      | RFU  | -          | Reserved bits, please keep the default value.                                                                                                                                                    |
| 0         | HDOE | RW         | When single-wire half-duplex, data transmission/reception status control<br>0: Only transmitted.<br>1: Only received.<br><i>Attention: Only valid for single-wire half-duplex communication.</i> |



### 22.7.4 SPI\_SSI Slave select register

Address offset: 0x0C      Reset value: 0x0000 0001

| Bit field | Name | Permission | Function description                          |           |     |                                       |
|-----------|------|------------|-----------------------------------------------|-----------|-----|---------------------------------------|
| 31:1      | RFU  | -          | Reserved bits, please keep the default value. |           |     |                                       |
| 0         | SSI  | RW         | Slave select, valid when SPI_CR1.SSM is 1     |           |     |                                       |
|           |      |            | SSM                                           | MSTR      | SSI | Description                           |
|           |      |            | 0                                             | -         | -   | Invalid.                              |
|           |      |            | 1                                             | 0(Slave)  | 0   | Set the machine to selected state.    |
|           |      |            |                                               |           | 1   | Set this machine to unselected state. |
|           |      |            | 1                                             | 1(Master) | 0   | SPI_CS pins output low.               |
|           |      |            |                                               |           | 1   | SPI_CS pins output high.              |



### 22.7.5 SPI\_IER Interrupt enable register

Address offset: 0x10

Reset value: 0x0000 0000

| Bit field | Name  | Permission | Function description                                                                                |
|-----------|-------|------------|-----------------------------------------------------------------------------------------------------|
| 31:8      | RFU   | -          | Reserved bits, please keep the default value.                                                       |
| 7         | MODF  | RW         | Mode error interrupt enable control<br>0: Disabled<br>1: Enabled                                    |
| 6         | SSERR | RW         | Slave select error interrupt enable control in slave mode<br>0: Disabled<br>1: Enabled              |
| 5         | OV    | RW         | Receive buffer overflow error interrupt enable control<br>0: Disabled<br>1: Enabled                 |
| 4         | UD    | RW         | Transmit buffer underflow error interrupt enable control in slave mode<br>0: Disabled<br>1: Enabled |
| 3         | SSR   | RW         | Slave select input rising edge interrupt enable control<br>0: Disabled<br>1: Enabled                |
| 2         | SSF   | RW         | Slave select input falling edge interrupt enable control<br>0: Disabled<br>1: Enabled               |
| 1         | RXNE  | RW         | Receive buffer non-empty interrupt enable control<br>0: Disabled<br>1: Enabled                      |
| 0         | TXE   | RW         | Transmit buffer empty interrupt enable control<br>0: Disabled<br>1: Enabled                         |



### 22.7.6 SPI\_ISR Interrupt flag register

Address offset: 0x14      Reset value: 0x0000 0001

| Bit field | Name  | Permission | Function description                                                                                                                                                                                        |
|-----------|-------|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:10     | RFU   | -          | Reserved bits, please keep the default value.                                                                                                                                                               |
| 9         | SSLVL | RO         | Slave select signal state<br>0: Slave select signal is low level.<br>1: Slave select signal is high level.                                                                                                  |
| 8         | BUSY  | RO         | Bus busy flag<br>0: Bus idle.<br>1: Bus busy.                                                                                                                                                               |
| 7         | MODF  | RO         | Mode error flag<br>0: Normal.<br>1: Error.<br><i>Note: In master mode, the slave select input is low to trigger MODF.</i>                                                                                   |
| 6         | SSERR | RO         | Slave select error flag in slave mode<br>0: Normal.<br>1: Error.                                                                                                                                            |
| 5         | OV    | RO         | Receive buffer overflow error flag bit<br>0: Normal.<br>1: Error.                                                                                                                                           |
| 4         | UD    | RO         | Transmit buffer underflow error flag in slave mode<br>0: Normal.<br>1: Error.                                                                                                                               |
| 3         | SSR   | RO         | Slave select input rising edge flag bit<br>0: No rising edge occurred.<br>1: Rising edge has occurred.<br><i>Note: It only makes sense when CS uses hardware pins in slave mode (MSTR = 0 and SSM = 0).</i> |
| 2         | SSF   | RO         | Slave select input falling edge flag bit<br>0: No falling edge occurred.<br>1: Falling edge occurred.<br><i>Note: It only makes sense when CS uses hardware pins in slave mode (MSTR = 0 and SSM = 0).</i>  |
| 1         | RXNE  | RO         | Receive buffer not empty flag bit<br>0: Receive buffer is empty.<br>1: Receive buffer is non-empty.<br><i>Note: This flag can be cleared by reading the DR register or writing 0 to ICR.RXNE.</i>           |
| 0         | TXE   | RO         | Send buffer empty flag bit<br>0: Send buffer is non-empty.<br>1: Send buffer is empty.                                                                                                                      |





### 22.7.7 SPI\_ICR Interrupt flag clear register

Address offset: 0x18      Reset value: 0x0000 00FF

| Bit field | Name  | Permission | Function description                                                                                                                                     |
|-----------|-------|------------|----------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:8      | RFU   | -          | Reserved bits, please keep the default value.                                                                                                            |
| 7         | MODF  | R1W0       | Mode error flag cleared<br>W0: Mode error flag cleared.<br>W1: No function.                                                                              |
| 6         | SSERR | R1W0       | Slave select error flag cleared in slave mode<br>W0: Slave select error flag in slave mode cleared.<br>W1: No function.                                  |
| 5         | OV    | R1W0       | Receive buffer overflow error flag cleared<br>W0: Receive buffer overflow error flag cleared.<br>W1: No function.                                        |
| 4         | UD    | R1W0       | Transmit buffer underflow error flag cleared in slave mode<br>W0: Transmit buffer underflow error flag in slave mode cleared.<br>W1: No function.        |
| 3         | SSR   | R1W0       | Slave select input rising edge flag cleared<br>W0: Slave select input rising edge flag cleared.<br>W1: No function.                                      |
| 2         | SSF   | R1W0       | Slave select input falling edge flag cleared<br>W0: Slave select input falling edge flag cleared.<br>W1: No function.                                    |
| 1         | RXNE  | R1W0       | Receive buffer not empty flag cleared<br>W0: Receive buffer non-empty flag cleared.<br>W1: No function.                                                  |
| 0         | FLUSH | R1W0       | Clear transmit buffer and shift register<br>W0: The transmit buffer and shift register (ISR.TXE is set to 1 after clearing) cleared.<br>W1: No function. |

### 22.7.8 SPI\_DR Data register

Address offset: 0x1C      Reset value: 0x0000 0000

| Bit field | Name | Permission | Function description                                    |
|-----------|------|------------|---------------------------------------------------------|
| 31:16     | RFU  | -          | Reserved bits, please keep the default value.           |
| 15:0      | DR   | RW         | Write data to be transmitted.<br>Read as received data. |



## 23 I2C interface

### 23.1 Overview

The CW32L012 integrates 2 I2C controllers, which can transmit data serially to the I2C bus according to the I2C specification at the set transmission rate (standard, fast, fast +) , detect the status during communication, and support bus conflict and arbitration processing in multi-master communication, support direct memory access (DMA) and operation in DeepSleep mode.

The I2C module supports the system management bus (SMBus) protocol version 3, a two-wire serial bus typically used for low-bandwidth communications.



## 23.2 Main features

- Support master transmit/receive and slave transmit/receive four working modes.
- Support clock stretching (clock synchronization) and multi-master communication conflict arbitration.
- Support standard (100Kbps) mode/fast (400Kbps) mode/fast+ (1Mbps) mode.
- Support fast+ (1Mbps) mode in slave mode.
- Support 7bit and 10bit addressing function.
- Support 2 slave addresses.
- Support broadcast call function.
- Support input signal noise filtering function.
- Support interrupt status query function.

### 23.2.1 Master

- 1-byte command/transmit FIFO (8-bit transmit data + 3-bit command).
- 1-byte receive FIFO (8-bit receive data).
- Command FIFO waits for the I2C bus to become idle and then initiates transmission.
- Command FIFO can initiate (Repeated) START signal and STOP signal and one or more master receive transmits.
- The STOP signal can be generated by the command FIFO or automatically when the transmit FIFO is empty.
- Supports receive data matching.
- Supports bus idle detection and pin low timeout detection.

### 23.2.2 Slave

- Separate I2C slave registers to minimise software overhead caused by master-slave switching.
- Supports 7-bit or 10-bit addressing, address ranges, SMBus alarms, and broadcast call addresses.
- Software-controllable ACK or NACK with optional clock stretch on ACK/NACK bits.
- Configurable clock stretch to avoid transmit FIFO underrun and receive FIFO overflow errors.
- Flags and optional interrupts for end-of-packet, STOP stop signals or bit-conflict errors.



## 23.3 Protocol description

The I2C bus uses two signal lines (the data line SDA and the clock line SCL) to transmit data between devices. SCL is a unidirectional clock line, which is fixedly driven by the master. SDA is a bidirectional data line, which is driven by the transmitting and receiving terminals in time sharing during data transmission.

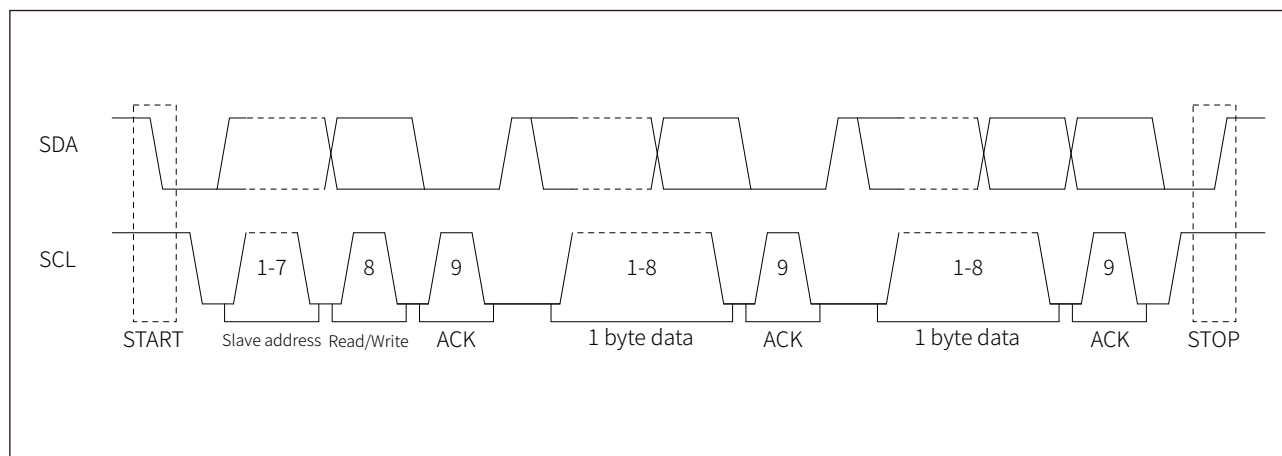
Multiple devices can be connected to the I2C bus. All devices are in an idle state when they are not transmitting data (the slave receiving mode is not addressed). Any device can send a START signal as a master to start data transmission. The bus is occupied until the STOP signal appears on the bus.

I2C communication adopts the master-slave structure, and the communication is initiated and terminated by the master. The master initiates communication by transmitting the START signal, then transmits a total of 8 bits of SLA+W/R data (where SLA is the 7-bit slave address, and W/R is the read-write bit), and releases the SDA bus at the ninth SCL clock, the corresponding slave occupies the SDA bus at the ninth SCL clock and outputs an ACK response signal to complete the slave addressing. After that, the transmitter and receiver of data communication are determined according to the W/R bit of the first byte transmitted by the master. For each byte of data transmitted by the transmitter, the receiver must respond with an ACK response signal. After the data transmission is completed, the master transmits a STOP signal to end the communication.

### 23.3.1 Protocol frame format

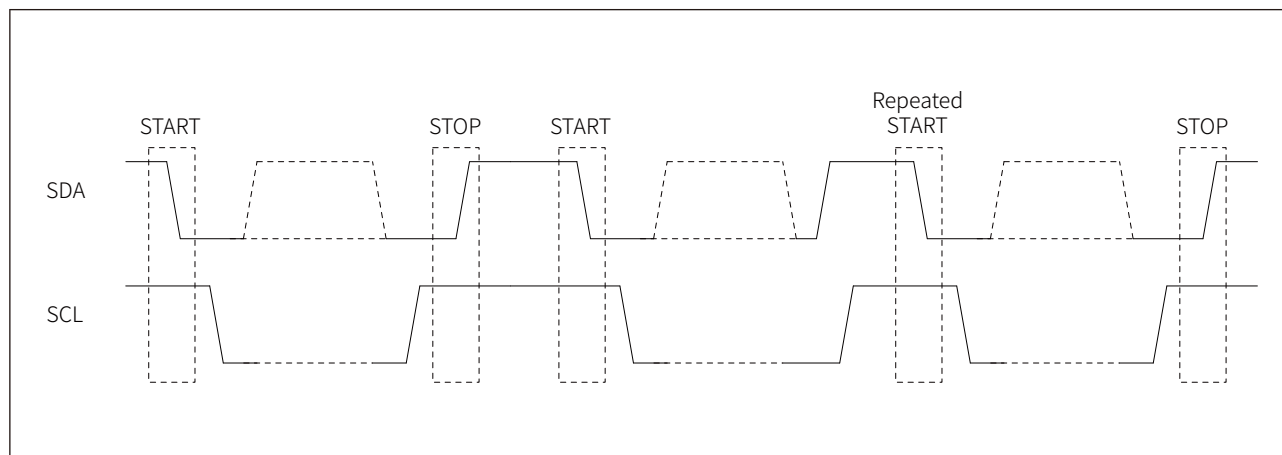
The standard I2C transmission protocol frame consists of four parts: start signal (START) or repeated start signal (Repeated START), slave address and read and write bits, data transmission, and stop signal (STOP). As shown in the following figure:

Figure 23-1 I2C protocol frame



- **Start signal (START)**  
When the bus is in an idle state (SCL and SDA lines are high at the same time), a falling edge signal from high to low appears on the SDA line, which is defined as a start signal. After the master transmits a start signal to the bus, data transmission starts and the bus is occupied.
- **Repeated start signal (Repeated START)**  
When a start signal does not appear before a stop signal, a new start signal appears, and the new start signal is defined as a repeated start signal. Before the master transmits the stop signal, the SDA bus is always in the occupied state, and other masters cannot occupy the bus.
- **Stop signal (STOP)**  
When the SCL line is high, a rising edge signal from low to high appears on the SDA line, which is defined as a stop signal. The master transmits a stop signal to the bus to end the data transmit and releases the bus.

Figure 23-2 Start and stop signals



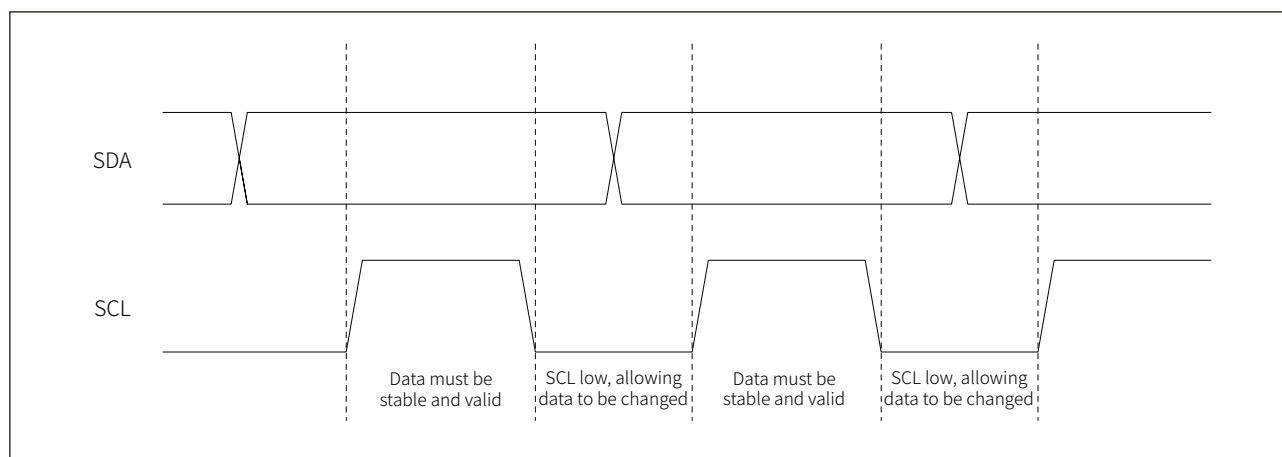
- Slave address and read/write bit

When the start signal is generated, the master starts to transmit the first byte of data: 7 bit slave address + read and write bits. The read and write bits (1: read; 0: write) control the direction of data transmit on the bus. The addressed slave occupies the SDA bus during the 9th SCL clock cycle and acknowledges the ACK by asserting SDA low.

- Data transmission

The master outputs the serial clock signal on the SCL line, and the master and slave devices transmit data through the SDA line. During data transmission, 1 SCL clock pulse transmits 1 data bit (the most significant bit MSB is first), and the data on the SDA line changes only when SCL is low, and each transmitted 1 byte is followed by an acknowledge bit. As shown in the following figure:

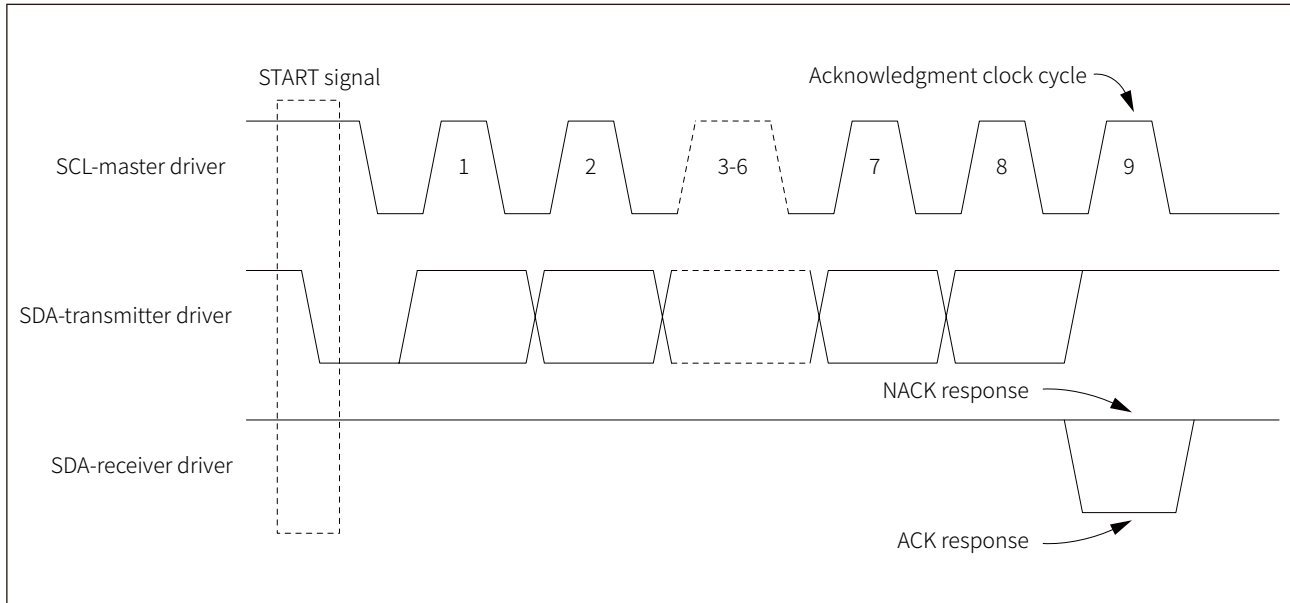
Figure 23-3 Data transmission



### 23.3.2 Transmission response

When transmitting data on the bus, the transmitter gives up the control of SDA in the 9th SCL clock cycle after transmitting 1 byte of data, and the receiver must reply a response bit in the 9th SCL clock cycle: the reception is successful, Send ACK response, receive exception send NACK response.

Figure 23-4 Transmission response



### 23.3.3 Conflict detection and arbitration

In a multi-master communication system, each node on the bus has a slave address. Each node can be accessed by other nodes as a slave node, and can also be used as a master node to send control bytes and transmit data to other nodes. A bus collision occurs when two or more nodes send a start signal to the bus at the same time and start transmitting data. The I2C controller has a built-in arbiter that can detect and arbitrate I2C bus conflicts to ensure the reliability and integrity of data communication.

#### Conflict detection principle

In the physical realization, the SDA and SCL pins have the same circuit structure, and the output drive of the pin is connected with the input buffer. The output structure is an open-drain field effect transistor, and the input structure is an inverter with high input impedance. Based on this structure:

1. Since SDA and SCL are open-drain structures, the "wire-and" logic of the signal is realized by means of an external pull-up resistor
2. The device reads data while writing data to the bus, which can be used to detect bus conflicts and realize "clock synchronization" and "bus arbitration".

According to the "wire-and" logic, if two masters send logic 1 or logic 0 at the same time, neither of the two masters can detect the conflict, and it needs to wait until the next bit of data is transmitted before continuing to detect the conflict; if one of the two hosts sends a logic 1, When a logic 0 is transmitted, the bus is logic 0 at this time, the master transmitting logic 1 detects a conflict, and the master transmitting logic 0 does not detect a conflict.

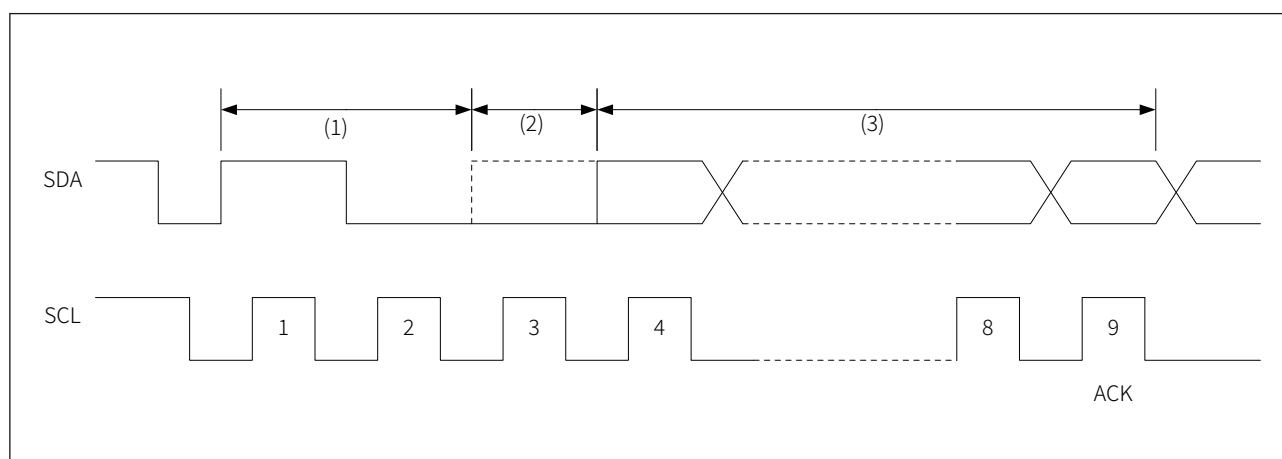
#### Conflict arbitration principle

When the master detects a bus conflict, the master loses arbitration, exits the master transmit mode, enters the unaddressed slave mode, releases the SDA data line, and returns to the address detection state, and then enters the state according to the received SLA+W/R corresponding slave mode (SLA address match enters addressed slave mode, SLA address mismatch enters unaddressed slave mode). The master that loses arbitration will still send the SCL serial clock until the end of the current byte transmit.

When the master does not detect a bus conflict, the master wins the arbitration and continues to dominate the data transmit until the communication is complete.

The following figure is a schematic diagram of transmitting conflict and arbitration between master A and master B on an I2C bus (assuming that the transmitting clocks of the two masters A and B are synchronized):

Figure 23-5 Conflict detection and arbitration



1. In stage (1), both master A and master B transmit logic 1 and logic 0 successively. Since the data is the same and synchronous, neither host will detect a bus conflict.
2. In stage (2), master A transmits a logic 1, master B transmits a logic 0, and the data on the SDA bus is a logic 0. The A host controller detects a data error and exits the transmission competition, that is, the arbitration is lost, and the A master enters the unaddressed slave receiver mode. Host B does not detect a bus conflict, wins the arbitration, and continues this data transmit.
3. In stage (3), master A is in unaddressed slave receive mode, but still generates SCL clock pulses until the end of the current byte transmit. After that, the A host I2C controller no longer transmits the clock signal. Since the B master wins the arbitration, both SCL and SDA are dominated by the B master to control the transmit.



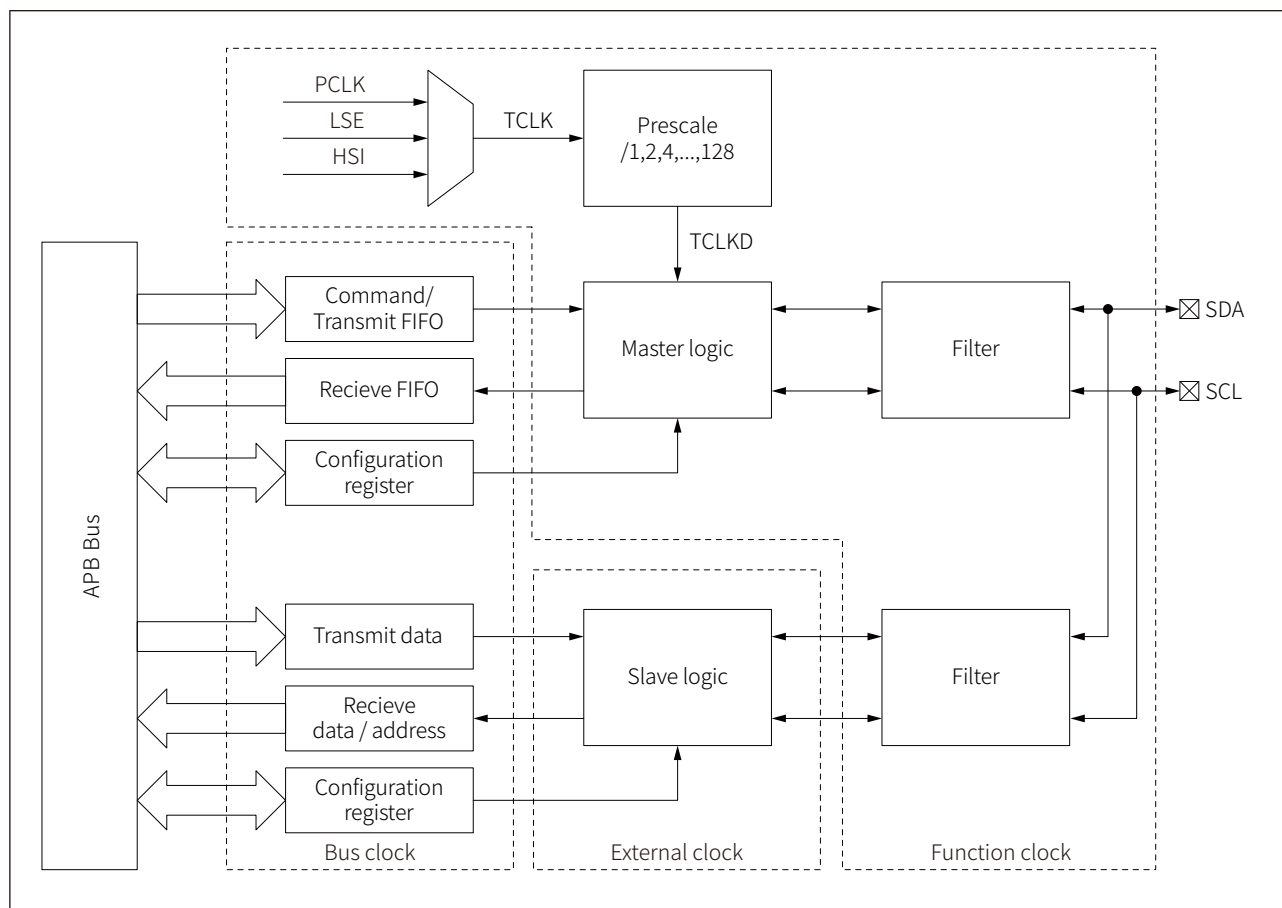


## 23.4 Functional description

### 23.4.1 Functional block diagram

The I2C module mainly includes clock generator, input filter, address comparator, protocol control logic, arbitration and synchronization logic, and related registers. Its functional block diagram is shown in the following figure:

Figure 23-6 I2C functional block diagram



### 23.4.1.1 Clock and reset

#### Function clock

The I2C function clock, TCLK, is asynchronous to the bus clock and can be held enabled in low-power mode to support I2C bus transmits from the I2C master controller. The I2C slave also uses the function clock to support digital filters and data hold time configurations. The I2C master divides the function clock via a prescaler, and the resulting frequency must be at least 8 times faster than the I2C bus bandwidth.

#### External clock

The I2C slave logic provides clocks directly from the external pins SCL and SDA so that the slave remains operational even if the I2C function clock is disabled. If the I2C function clock is disabled, the I2C slave digital filter must be disabled, which may affect compliance with certain timing parameters of the I2C specification, such as data hold time.

#### Bus clock

The bus clock is used only for bus access to control and configuration registers. The bus clock frequency must be sufficient to support the data bandwidth requirements of the I2C master and slave registers.

#### Software reset

I2Cx\_MCR0.RESET resets all master logic and registers to their default state, except the I2Cx\_MCR0 register itself. I2Cx\_SCR0.RESET resets all slave logic and registers to their default state, except the I2Cx\_SCR0 register itself.

#### FIFO reset

The I2C master resets the FIFO via the write-only control bit. Write 1 to I2Cx\_MCR0.TXFIFORST to reset the transmit FIFO, and write 1 to I2Cx\_MCR.RXFIFORST to reset the receive FIFO. After the FIFO is reset, the FIFO is empty.

The I2C slave resets the data register by write-only control bit. Write 1 to I2Cx\_SCR0.TXFIFORST to reset the transmit data register and write 1 to I2Cx\_SCR0.RXFIFORST to reset the receive data register. After the data register is reset, this data register is empty.

### 23.4.1.2 Input filter

The input filter can digitally filter the SDA and SCL input signals, which can effectively eliminate the interference signals on the bus. The filter is configurable via the I2Cx\_yCR3.FLTSDA and I2Cx\_yCR3.FLTSCS control bits (In the I2C chapter, y=M, S, where M represents the Master and S represents the Slave. The same below.).

The input filter cycle count is not affected by the I2Cx\_MCR2.PRESCALE configuration and the input filter is automatically disabled in high speed mode.



### 23.4.1.3 Address comparator

Each device on the I2C bus has a slave address, and each slave address is different. The master addresses the slave according to the slave address, and the slave automatically detects whether the addressing address transmitted by the master matches the local address through the address comparator to determine whether to communicate with the master.

The I2C controller supports 2 programmable slave addresses, configurable through the ADDR0 and ADDR1 bit fields of the I2Cx\_SAMR register.

The I2C slave can be configured to:

- Match one of two addresses, using either 7-bit or 10-bit addressing modes for each address.
- Match a range of addresses in 7-bit or 10-bit addressing modes.
- Match broadcast call addresses and generate appropriate flags.
- Match the SMBus alarm address and generate the appropriate flags.

After matching a valid address, the I2C slave automatically performs slave transmit or slave receive until:

- NACK is detected (unless I2Cx\_MCR2.IGNACK is set).
- A bit conflict error is detected (the I2C slave is driving SDA but sampling a different value).
- A (Repeated) START or STOP signal is detected.

### 23.4.1.4 Command/transmit FIFO

The transmit FIFO stores command data to initiate various I2C operations:

- Transmit the (Repeated) START signal with an address byte, expecting to receive an ACK or NACK.
- Transmit data (this is the default value for writing 0 extended bytes to the transmit FIFO).
- Receive 1 to 256 bytes of data (can be configured to discard received data without storing it in the receive FIFO).
- Transmit STOP signal (can also be configured to send a stop signal when the transmit FIFO is empty).

Multiple transmit and receive commands may be inserted between start conditions and stop conditions; transmit and receive commands may not be interleaved (to comply with the I2C specification). Receive data commands and receive data and discard commands may be interleaved to ensure that only the required receive data is stored in the receive FIFO (or compared with data matching logic).

The I2C master automatically transmits a NACK on the last byte of the receive data command unless the next command in the FIFO is also a receive data command. When the receive data command is completed, if the transmit FIFO is empty, the NACK is also transmitted automatically.

The I2C master supports 10-bit addressing with the (Repeated) START start signal, followed by the transmit data byte containing the second address byte, and finally any number of data bytes transmitted by the master.

If a NACK is expected after a START start signal or a Repeated START repeat start signal (e.g., high-speed mode master code), a STOP stop signal or a Repeated START repeat start signal must follow.



### 23.4.1.5 Receive FIFO and data match

The receive FIFO is used to store receive data during master receive transmissions. Receive data can also be configured to discard receive data without storing it in the receive FIFO, which is configured by a command word in the transmit FIFO.

I2C supports the data receive match function, the specific data match method is configured via the I2Cx\_MCR2.MATCFG control bit. Received data discarded due to a command word does not result in a data match. It can be configured via the I2Cx\_MCR1.RDMO control bit to discard all receive data until a data match is detected; when the I2Cx\_MCR1.RDMO control bit is cleared following a data match, I2Cx\_MCR1.RDMO should be cleared before I2Cx\_MISR.MATCH is cleared to allow reception of all subsequent data.

### 23.4.1.6 Slave transmits/receives data

The Slave transmit and receive data registers are double-buffered and are updated only during slave transmit and slave receive transmissions, respectively.

The slave transmit data register can be configured to request data only after a slave transmit transmission is detected, or to request new data when the transmit data register is empty, as configured via the TXCFG bit field of the I2Cx\_SCR1 register. The slave transmit data register should only be written when the transmit data flag I2Cx\_SISR.TXE is set.

The received slave address can be configured to be read from the receive data register I2Cx\_SRDR (e.g. when transferring data using DMA) or from the address status register I2Cx\_SASR, as configured via the RXCFG bit field of the I2Cx\_SCR1 register. The receive data register can only be read if the receive data flag I2Cx\_SISR.RXNE is set (or the address valid flag I2Cx\_SISR.ADDR is set and I2Cx\_SCR1.RXCFG is 1). The address status register can be read only when the address valid flag is set.



## 23.4.2 Arbitration and synchronization logic

### 23.4.2.1 SCL synchronization

I2C supports clock synchronization (clock stretching) function, the time of SCL clock low level is determined by the device with the longest SCL clock low level width, and the SCL clock high level time is determined by the device with the shortest clock high level width.

I2C slave supports enabling clock stretching under a variety of conditions:

- During the 9th clock pulse of the address byte, and the address valid flag is set.
- During the 9th clock pulse of a slave-transmit transfer and the transmit data flag is set.
- During the 9th clock pulse of a slave-receive transfer and the Receive Data flag is set.
- During the 8th clock pulse of an address byte or a slave-receive transfer and the transmit response flag is set.
- Clock stretching can also be extended for CLKHOLD cycles, to allow additional setup time to sample the SDA pin externally.

Unless extended via the CLKHOLD configuration, when clock stretching is enabled, the clock stretch extends the peripheral bus clock by one cycle after an SDA update.

### 23.4.2.2 SDA arbitration

I2C supports SDA conflict detection and arbitration, which can ensure that the data on the I2C bus is not destroyed when multiple masters attempt to control the I2C bus.

When each master transmits data, it will simultaneously compare whether the data on the bus is consistent with the data transmitted by itself. If they are inconsistent, it is considered that a bus conflict is detected, and the transmitting competition will be withdrawn, that is, the arbitration will be lost. A master that loses arbitration will immediately switch to an unaddressed slave state to ensure that it can be addressed by a master that succeeds in arbitration. A master that loses arbitration continues to output the SCL serial clock until the current byte transfer is complete.

SDA arbitration generally occurs during the master's transmit address data phase, and if two masters are transmitting data to a slave at the same time, i.e., both masters are transmitting the same slave address, the arbitration lasts for the second byte.



### 23.4.3 Timing configuration

#### 23.4.3.1 Master timing

The I2C master timing parameters are configured as follows, and they must be configured to meet the I2C timing specification for the desired mode.

Table 23-1 Timing parameter

| I2C specification timing parameter                  | I2C specification timing symbol | I2C timing parameter (unit: function clock cycle) |
|-----------------------------------------------------|---------------------------------|---------------------------------------------------|
| SCL clock cycle                                     | $t_{SCL}$                       | $(CLKHI+CLKLO+2+SCL\_LATENCY) \times 2^{PRESALE}$ |
| (Repeated) Hold time of initial signal              | $t_{HD:STA}$                    | $(SETHOLD+1) \times 2^{PRESALE}$                  |
| Low level cycle of SCL clock                        | $t_{LOW}$                       | $(CLKLO+1) \times 2^{PRESALE}$                    |
| High level cycle of SCL clock                       | $t_{HIGH}$                      | $(CLKHI+1+SCL\_LATENCY) \times 2^{PRESALE}$       |
| Repeated start signal and stop signal build-up time | $t_{SU:STA}, t_{SU:STO}$        | $(SETHOLD+1+SCL\_LATENCY) \times 2^{PRESALE}$     |
| Data hold time                                      | $t_{HD:DAT}$                    | $(DATAVD+1) \times 2^{PRESALE}$                   |
| Data setup time                                     | $t_{SU:DAT}$                    | $(SDA\_LATENCY+1) \times 2^{PRESALE}$             |
| Bus idle time between stop signal and start signal  | $t_{BUF}$                       | $(CLKLO+1+SDA\_LATENCY) \times 2^{PRESALE}$       |
| Data valid time, data valid answer time             | $t_{VD:DAT}, t_{VD:ACK}$        | $(DATAVD+1) \times 2^{PRESALE}$                   |

The following table defines the delay parameters that assume a rise time of less than one I2C function clock cycle. The rise time depends on a variety of factors, including I/O transmit delay, I2C bus load, and external pull-up resistor size, to name a few. The longer the rise time, the more cycles it takes for the signal to propagate through the synchroniser (and burr filter), which also increases the delay.

Table 23-2 Synchronisation delay

| Timing parameter | Time definition                                        |
|------------------|--------------------------------------------------------|
| SCL_LATENCY      | $ROUNDDOWN((2+FLT_{SCL}+SCL\_RISETIME)/(2^{PRESALE}))$ |
| SDA_LATENCY      | $ROUNDDOWN((2+FLT_{SDA}+SDA\_RISETIME)/(2^{PRESALE}))$ |

The following timing constraints must be implemented to ensure that the SDA level cannot change while SCL is high outside of the transmitted (Repeated) START signal or STOP signal.



Table 23-3 I2C timing parameter limits

| Timing parameter | Min.                                           | Max.                                          | Note                                                                                            |
|------------------|------------------------------------------------|-----------------------------------------------|-------------------------------------------------------------------------------------------------|
| CLKLO            | 0x03                                           | -                                             | It also guarantees:<br>$\text{CLKLO} \times 2^{\text{PRESCALE}} > \text{SCL\_LATENCY}$ .        |
| CLKHI            | 0x01                                           | -                                             | Configure CLKHI to meet duty cycle requirements in the I2C specification.                       |
| SETHOLD          | 0x02                                           | -                                             | It also guarantees:<br>$\text{SETHOLD} \times 2^{\text{PRESCALE}} > \text{SDA\_LATENCY}$ .      |
| DATAVD           | 0x01                                           | $\text{CLKLO} - \text{SDA\_LATENCY} - 1$      | Configure DATAVD to meet the data hold requirements of the I2C specification.                   |
| FLTSCL           | 0x00                                           | $\text{CLKLO} \times 2^{\text{PRESCALE} - 3}$ | FLTSCL and FLTSDA are the only parameters that are not multiplied by $2^{\text{PRESCALE}}$ .    |
| FLTSDA           | FLTSCL                                         | $\text{CLKLO} \times 2^{\text{PRESCALE} - 3}$ | Configuring FLTSDA greater than FLTSCL delays the SDA input to compensate for board-level bias. |
| BUSIDLE          | $(\text{CLKLO} + \text{SETHOLD} + 2) \times 2$ | -                                             | Must be greater than CLKHI+1.                                                                   |

The following table provides some timing configuration examples. When switching between two modes (e.g., fast and fast+ modes) using different timing configuration registers, the division factor must be constant between the two modes.

Table 23-4 I2C timing configuration example

| I2C mode | Clock frequency | Baud-rate | Division factor | FLTSCL/FLTSDA | SETHOLD | CLKLO | CLKHI | DATAVD |
|----------|-----------------|-----------|-----------------|---------------|---------|-------|-------|--------|
| Fast     | 8MHz            | 400kbps   | 0x0             | 0x0/0x0       | 0x04    | 0x0B  | 0x05  | 0x02   |
| Fast+    | 8MHz            | 1Mbps     | 0x0             | 0x0/0x0       | 0x02    | 0x03  | 0x01  | 0x01   |
| Fast     | 48MHz           | 400Kbps   | 0x0             | 0x1/0x1       | 0x1D    | 0x3E  | 0x35  | 0x0F   |
| Fast     | 48MHz           | 400Kbps   | 0x2             | 0x1/0x1       | 0x07    | 0x11  | 0x0B  | 0x03   |
| Fast+    | 48MHz           | 1Mbps     | 0x2             | 0x1/0x1       | 0x03    | 0x06  | 0x04  | 0x04   |



### 23.4.3.2 Slave timing

The I2C slave can be configured with the following timing parameters, which are configured through the I2Cx\_SCR3 register. These parameters are disabled when I2Cx\_SCR0.FLTEN is 0. When disabled, the I2C slave is clocked directly from the I2C bus and may not meet all the timing requirements of the I2C specification (e.g., SDA minimum hold time in standard/fast mode).

- SDA data valid time from SCL falling edge to SDA update.
- SCL hold time when clock stretching is enabled to increase the build-up time when sampling external SDA.
- SCL glitch filter time.
- SDA glitch filter time.





### 23.4.4 Operating modes

The I2C controller supports 4 operating modes: master transmit mode, master receive mode, slave transmit mode, and slave receive mode. In addition, the broadcast receive mode is also supported, and its working method is similar to that of the slave receive mode. The I2C master logic operates independently of the slave logic and performs all master mode transmissions on the I2C bus. And in order to perform all slave transmissions on the I2C bus, the I2C slave logic also operates independently of the master logic.

Whenever the I2C master is enabled, it monitors the I2C bus to detect if the I2C bus is idle (I2Cx\_MISR.BUSBUSY flag). If SCL or SDA is at low level, the I2C bus is not idle; if a STOP stop signal is detected or a bus idle timeout is detected (configured via I2Cx\_MCR3.BUSIDLE), the I2C bus is idle. After the I2C bus is idle, the I2C master initiates a transmission on the I2C bus if the transmit FIFO is not empty. This operation consists of the following steps:

- Wait for bus idle time equal to  $(CLKLO+1) \times 2^{PRESCALE} \times t_{TCLKO}$
- The START start signal and address byte are transmitted using the timing configuration in the I2Cx\_MCCR register.
- According to the configuration of the sending FIFO, performs master transmitting or master receiving.
- Transmit a NACK on the last byte of a master receive transmission unless the next command in the transmit FIFO is also a receive data command and the transmit FIFO is non-empty.
- Transmit a repeated start signal or stop signal by transmitting the configuration of the FIFO or I2Cx\_MCR2.AUTOSTOP. The timing configuration registers can be changed when transmitting a repeated start signal.

When the I2C master is disabled (due to I2Cx\_MCR0.MEN being cleared or automatically disabled due to a mode change), I2C continues to be cleared of transmitting the FIFO until a STOP stop signal has been transmitted. However, I2C will no longer suspend the I2C bus waiting for a transmit or receive FIFO, and I2C automatically generates a STOP stop signal after the transmit FIFO is empty.

The I2C master can suspend the I2C bus under certain conditions, which will cause SCL to continue to pull low on the first bit of the byte until the condition is removed:

- I2C master is enabled and busy, transmit FIFO is empty, I2Cx\_MCR2.AUTOSTOP is in the clear state.
- I2C master is enabled and receiving data, the received data has not been discarded (due to a command or receive data match), and the receive FIFO is full.



### 23.4.4.1 Master transmit mode

In master transmit mode, the master actively transmits multiple bytes to the slave.

The SCL clock is generated by the master, so it is necessary to set the I2Cx\_MCR2.PRESCALE divider coefficient according to the transmission baud rate and configure the corresponding bit field of the master timing configuration register, for details please refer to [23.4.3 Timing configuration](#).

The master writes 100 to the CMD bit field of the I2Cx\_MTDR register, and also writes the slave address and the write flag bit (SLA+W) to the DATA bit field to notify the controller to transmit a START start signal; the controller receives the notification and detects whether the bus is idle or not, and when it is idle, the master controller transmits a START start signal to the bus, and sends the slave address to the bus; queries and waits for the I2Cx\_MISR.TXE flag bit to change to 1.

Thereafter the master transmits multiple bytes of user-defined data and detects the ACK answer signal according to the needs of the application. Each byte is transmitted by writing a byte of data directly to the DATA bit field of the I2Cx\_MTDR register.

If the master receives a NACK answer signal during data transmission, indicating that the slave is no longer receiving the data transmitted by the master, the master should transmit a STOP signal to end this data transmission or a Repeated START repeat start signal to start a new round of transmit.

When the master completes transmitting all the data and writes 010 to the CMD bit field of the I2Cx\_MTDR register, the master controller will transmit the STOP stop signal to the I2C bus, completing this data transmit and releasing the bus.

When the master has completed transmitting all the data, instead of transmitting the STOP stop signal, it can also directly transmit the Repeated START repeat start signal to continue occupying the bus for a new round of data transmission.

When the master loses arbitration due to a bus conflict, it enters the unaddressed slave receive mode.



### 23.4.4.2 Master receive mode

The master receive mode is used to receive the data transmitted from the slave, and the master should respond to the ACK answer signal after every 1 byte of data received, so the master needs to set I2Cx\_STAR.TXNACK to 0.

The SCL clock is generated by the master, so it is necessary to set the I2Cx\_MCR2.PRESCALE divider coefficient according to the transmission baud rate and configure the corresponding bit field of the master timing configuration register, for details see [23.4.3 Timing configuration](#).

The master writes 100 to the CMD bit field of the I2Cx\_MTDR register, and at the same time writes the slave address and the read flag bit (SLA+R) to the DATA bit field to notify the controller to transmit a START start signal; the controller receives the notification and detects whether the bus is idle or not, and when it is idle, the master controller transmits a START start signal to the bus, and transmits the slave address to the bus; queries and waits for the I2Cx\_MISR.TXE flag bit to change to 1.

The master writes 001 to the CMD bit field of the I2Cx\_MTDR register, and at the same time writes the number of bytes to be received to the DATA bit field, and starts to receive the data transmitted from the slave; after every 1 byte of data is received, an ACK answer signal is replied, and it is necessary to set I2Cx\_STAR.TXNACK before the last byte is received, i.e., the master does not generate an ACK answer signal when the last byte is received, so as to notify the slave to stop data transmission. This means that the master does not generate an ACK response signal when it receives the last byte, so as to notify the slave to stop transmitting data. During data receiving, it should query and wait for the I2Cx\_MISR.RXNE flag bit to change to 1 to read the received data in time.

Master in the process of receiving data, if the slave for some reason no longer transmits the data required by the master, the master will subsequently receive the all 1 signal, at this time, the master needs to judge the data in the application layer, determining that the slave is unresponsive, should be transmitted STOP STOP signal to end the transmission, or transmit Repeated START Repeated start signal to open a new round of transmit.

When the master completes receiving all the data, it writes 010 to the CMD bit field of the I2Cx\_MTDR register, and the master controller will transmit the STOP stop signal to the I2C bus, completing this data transmit and releasing the bus.

When the master completes receiving all the data, instead of transmitting the STOP stop signal, it can also directly transmit the Repeated START repeat start signal to continue occupying the bus for a new round of data transmit.

When the master loses arbitration due to a bus conflict, it enters the unaddressed slave receive mode.



### 23.4.4.3 Slave receive mode

In slave receive mode, the slave receives the data transmitted by the master.

Before transfer, the slave needs to configure the slave address via the ADDR0 and ADDR1 bit fields of the I2Cx\_SAMR register and set the address comparison method via I2Cx\_SCR1.ADDRCFG; and set I2Cx\_STAR.TXNACK to 0 in response to master addressing.

After completing the above initialisation, the slave enters the idle state (unaddressed slave receive mode) and waits to be addressed by the write signal (SLA+W) transmitted by the master. When the slave receives SLA+W, if the address matches I2Cx\_SAMR.ADDR0/1, the slave responds with an ACK answer and enters the addressed slave receive mode, and the address valid flag I2Cx\_SISR.ADDR is set to 1 at the same time. At this point, the I2Cx\_SASR register must be read to clear the address valid flag in order to receive data transmitted from the master on the bus.

The slave responds with 1 ACK answer for each byte of data received. The operation of the application reading the I2Cx\_SRDR register clears the receive data flag, I2Cx\_SISR.RXNE, to be ready to receive the next byte of data.

If I2Cx\_STAR.TXNACK is set while the slave is receiving data, the slave will return the NACK signal when it receives the next byte, and the slave's own state is switched to the unaddressed slave receive mode, ending the communication with the master and no longer receiving data, and the I2Cx\_SRDR register maintains the previously received data. From this feature, the slave application can actively switch the slave from the addressed slave receive mode to the unaddressed slave receive mode by setting I2Cx\_STAR.TXNACK to 1.

When the master loses arbitration due to bus collision in SLA+ read/write phase, it will enter unaddressed slave receive mode, and then if it receives SLA+W corresponding to local address and replies ACK, it will enter addressed slave receive mode.



#### 23.4.4.4 Slave transmit mode

In slave transmit mode, data is transmitted from the slave to the master.

Before transfer, the slave needs to configure the slave address via the ADDR0 and ADDR1 bit fields of the I2Cx\_SAMR register and set the address comparison method via I2Cx\_SCR1.ADDRCFG; and set I2Cx\_STAR.TXNACK to 0 in response to master addressing.

After completing the above initialisation, the slave enters the idle state (unaddressed slave receive mode) and waits to be addressed by the read signal (SLA+R) transmitted from the master. When the slave receives SLA+R, if the address matches I2Cx\_SAMR.ADDR0/1, the slave responds with an ACK answer and enters the addressed slave transmit mode, and the address valid flag I2Cx\_SISR.ADDR is set to 1. At the same time, the I2Cx\_SASR register must be read to clear the address valid flag. At this time, the data to be sent should be written to the I2Cx\_STDR register in time, wait for 1 byte of data to be transmitted, and confirm the ACK answer after each byte of data has been sent until all the data has been transmitted.

If the slave receives a NACK answer during transmit, the slave stops transmitting data and enters the unaddressed slave receive mode.

When the master loses arbitration due to a bus conflict during the SLA+ read/write phase it will enter the unaddressed slave receive mode, and then if it receives an SLA+R that matches the local address and responds with an ACK, it will enter the addressed slave transmit mode.



### 23.4.4.5 Broadcast receive mode

The broadcast receive mode is a special slave receive mode, which is entered when the slave is in the idle state (unaddressed slave receive mode) and receives an SLA+W from the master and the SLA is broadcast address 0x00.

To receive broadcast messages from the master, the slave needs to set I2Cx\_STAR.TXNACK to 0 and I2Cx\_SCR1.GCEN to 1 in response to broadcast addressing and broadcast data from the master.

After completing the above initialisation, the slave enters the idle state (unaddressed slave receive mode) and waits to be addressed by the write signal (SLA+W) transmitted from the master. When the slave receives SLA+W, if the address matches to the broadcast address 0x00, the slave responds with an ACK answer and enters the broadcast receive mode, and the broadcast address match flag, I2Cx\_SISR.GC, is set to 1 at the same time. At this time, the I2Cx\_SASR register must be read in time to clear the broadcast call flag and to receive the data transmitted from the master on the bus.

The slave responds with 1 ACK answer for each byte of data received. The operation of the application reading the I2Cx\_SRDR register clears the receive data flag, I2Cx\_SISR.RXNE, to be ready to receive the next byte of data.

If I2Cx\_STAR.TXNACK is set while the slave is receiving data, the slave will return the NACK signal when it receives the next byte, and the slave's own state is switched to the unaddressed slave receive mode, ending the communication with the master and no longer receiving data, and the I2Cx\_SRDR register maintains the previously received data. From this feature, it can be seen that the slave application can actively switch the slave from the addressed broadcast receive mode to the unaddressed slave receive mode by setting I2Cx\_STAR.TXNACK to 1.

When the master loses arbitration due to a bus conflict during the SLA+ read/write phase it will enter the unaddressed slave receive mode, and then if it receives an SLA+W that matches the broadcast address and responds with an ACK, it will enter the addressed broadcast receive mode.



## 23.4.5 Error flag

### Master arbitration failure flag (I2Cx\_MISR.ARBI)

The I2Cx\_MISR.ARBI flag is set by hardware when the master transmits a logic 1 but detects a logic 0 on the I2C bus, or when the master detects a start or stop signal while transmitting data.

When the ARBI flag is set, the I2C master releases the I2C bus (becomes idle) and the I2C master does not initiate a new START start signal until the ARBI flag is cleared.

### Master NACK flag (I2Cx\_MISR.NACK)

I2Cx\_MISR.NACK flag bit to indicate whether the master has received an expected ACK or NACK. The following conditions set the NACK flag:

- Master detects unexpected NACK while transmitting data.
- Master detects NACK when transmitting address byte expecting an ACK.
- Master detects ACK when transmitting an address byte expecting a NACK.

When the NACK flag is set, the master does not transmit a new START start signal until the NACK flag is cleared. When the NACK flag is set, the I2C master automatically transmits the STOP stop signal if I2Cx\_MCR2.AUTOSTOP is 1 or the transmit FIFO is non-empty.

### Master FIFO error flag (I2Cx\_MISR.FIFO)

The master detects an attempt to transmit or receive data without a Repeated START signal, which causes FIFO errors and the I2Cx\_MISR.FIFO flag bit is set by hardware.

When the FIFO flag is set, the I2C master transmits the STOP stop signal (if busy) and does not issue a new START start signal until the FIFO flag is cleared.

### Master pin low time-out flag (I2Cx\_MISR.PINLOW)

When the duration of SDA (if I2Cx\_MCR2.TIMECFG is set) and SCL being low exceeds (I2Cx\_MCR4.PINLOW  $\times$  256) prescaler cycles, the I2Cx\_MISR.PINLOW flag is set in hardware, even if the I2C master is in an idle state.

The PINLOW flag cannot be cleared as long as the pin low timeout persists. The PINLOW flag must be cleared before the I2C generates the START start signal.

### Slave bit conflict error flag (I2Cx\_SISR.BIT)

When the slave transmits a logic 1 on the I2C bus but detects a logic 0, the I2Cx\_SISR.BIT flag is set. With the BIT flag set, the slave ignores the remaining transmissions until the next (Repeated) START signal.

### Slave FIFO error flag (I2Cx\_SISR.FIFO)

The FIFO error flag will be set due to insufficient transmitted data or received data overflow, or address overrun when I2Cx\_SCR1.RXCFG is 1.

The FIFO flag can only be set when clock stretching is disabled; clock stretching needs to be enabled to eliminate the possibility of underruns and overruns.



## 23.5 I2C interrupt

The I2C master supports 9 interrupt sources and the slave supports 12 interrupt sources. When an I2C interrupt trigger event occurs, the interrupt flag in the I2Cx\_yISR (In the I2C chapter, x = 1, 2; y = M, S, M stands for Master, and S stands for Slave.) register will be set, and an interrupt request will be generated if the interrupt in the corresponding bit of the I2Cx\_yIER register is in the enable state. The user should query the value of the I2Cx\_yISR register in the interrupt service routine to determine the cause of the interrupt generation, and before exiting the interrupt service routine, the corresponding flag should be cleared through the I2Cx\_yICR register.

Table 23-5 Master interrupt control

| Interrupt event                  | Interrupt flag bit | Interrupt enable bit | Flag clearing method        |
|----------------------------------|--------------------|----------------------|-----------------------------|
| Transmit FIFO empty              | MISR.TXE           | MIER.TXE             | Write data to transmit FIFO |
| Receive FIFO non-empty           | MISR.RXNE          | MIER.RXNE            | Read receive FIFO           |
| End of packet                    | MISR.PACKET        | MIER.PACKET          | Write 0 to MICR.PACKET      |
| STOP signal generated            | MISR.STOP          | MIER.STOP            | Write 0 to MICR.STOP        |
| No expected ACK or NACK received | MISR.NACK          | MIER.NACK            | Write 0 to MICR.NACK        |
| Master lost arbitration          | MISR.ARBI          | MIER.ARBI            | Write 0 to MICR.ARBI        |
| FIFO error                       | MISR.FIFO          | MIER.FIFO            | Write 0 to MICR.FIFO        |
| Pin low timeout                  | MISR.PINLOW        | MIER.PINLOW          | Write 0 to MICR.PINLOW      |
| Receive data match               | MISR.MATCH         | MIER.MATCH           | Write 0 to MICR.MATCH       |

Table 23-6 I2C slave interrupt control

| Interrupt event                  | Interrupt flag bit | Interrupt enable bit | Flag clearing method    |
|----------------------------------|--------------------|----------------------|-------------------------|
| Request to transmit data         | SISR.TXE           | SIER.TXE             | Write STDR register     |
| Data received                    | SISR.RXNE          | SIER.RXNE            | Read SRDR register      |
| Matching address received        | SISR.ADDR          | SIER.ADDR            | Read SASR register      |
| Response needs to be transmitted | SISR.TACK          | SIER.TACK            | Write STAR register     |
| Repeat start signal detected     | SISR.RESTART       | SIER.RESTART         | Write 0 to SICR.RESTART |
| Stop signal detected             | SISR.STOP          | SIER.STOP            | Write 0 to SICR.STOP    |
| Conflict in bit error detected   | SISR.BIT           | SIER.BIT             | Write 0 to SICR.BIT     |
| FIFO error                       | SISR.FIFO          | SIER.FIFO            | Write 0 to SICR.FIFO    |
| Address 0 match                  | SISR.AM0           | SIER.AM0             | Read SASR register      |
| Address 1 match                  | SISR.AM1           | SIER.AM1             | Read SASR register      |
| Broadcast address match          | SISR.GC            | SIER.GC              | Read SASR register      |
| SMBus Alert address match        | SISR.ALERT         | SIER.ALERT           | Read SASR register      |





## 23.6 Programming examples

### 23.6.1 Master transmits example

- Step 1:** According to the relevant description in GPIO section [9.3.5 Multiplexing function](#), map SCL and SDA to the required pins and configure the SCL and SDA pins for open-drain output mode.
- Step 2:** Set `SYSCTRL_APBEN2.I2C` to 1 to enable the I2C module clock.
- Step 3:** Configure `I2Cx_MCR2.PRESCALE` so that the SCL clock rate meets the application requirements.
- Step 4:** Configure `I2Cx_MCCR.CLKHI` so that the SCL clock high level cycle meets the application requirements.
- Step 5:** Configure `I2Cx_MCCR.CLKLO` so that the SCL clock low level cycle meets the application requirements.
- Step 6:** Set `I2Cx_MCR0.MEN` to 1 to enable the I2C master.
- Step 7:** Set `I2Cx_MCR2.PINCFG` to 0 to configure the SCL/SDA pins as open-drain inputs and outputs.
- Step 8:** Write 100 to `I2Cx_MTDR.CMD` to attempt to transmit the START signal, and write `SLA+W` to `I2Cx_MTDR.DATA`.
- Step 9:** Query and wait for the `I2Cx_MISR.TXE` flag bit to change to 1.
- Step 10:** Write the data to be transmitted to `I2Cx_MTDR.DATA`.
- Step 11:** Query and wait for the `I2Cx_MISR.TXE` flag bit to change to 1.
- Step 12:** If the data to be transmitted is not completed, jump to step 10 to continue execution.
- Step 13:** Write 010 to `I2Cx_MTDR.CMD` to attempt to transmit the STOP signal to end this data transfer.



### 23.6.2 Master receives example

- Step 1:** According to the relevant description in GPIO section [9.3.5 Multiplexing function](#), map SCL and SDA to the required pins and configure the SCL and SDA pins for open-drain output mode.
- Step 2:** Set `SYSCTRL_APBEN2.I2C` to 1 to enable the I2C module clock.
- Step 3:** Configure `I2Cx_MCR2.PRESCALE` to make the clock rate of SCL meet the application requirements.
- Step 4:** Configure `I2Cx_MCCR.CLKHI` so that the SCL clock high level cycle meets the application requirements.
- Step 5:** Configure `I2Cx_MCCR.CLKLO` so that the SCL clock low level cycle meets the application requirements.
- Step 6:** Set `I2Cx_MCR0.MEN` to 1 to enable the I2C master.
- Step 7:** Set `I2Cx_MCR2.PINCFG` to 0 to configure the SCL/SDA pins as open-drain inputs and outputs.
- Step 8:** Write 100 to `I2Cx_MTDR.CMD` to attempt to transmit the START signal and SLA+R to `I2Cx_MTDR.DATA`.
- Step 9:** Query and wait for the `I2Cx_MISR.TXE` flag bit to change to 1.
- Step 10:** Write 001 to `I2Cx_MTDR.CMD` and also write the number of bytes of data to be received to `I2Cx_MTDR.DATA`.
- Step 11:** Query and wait for the `I2Cx_MISR.RXNE` flag bit to change to 1 to read the received data from the `I2Cx_MRDR` register.
- Step 12:** If the data to be received is not completed, jump to step 11 to continue execution.
- Step 13:** Write 010 to `I2Cx_MTDR.CMD` to attempt to transmit the STOP signal to end this data transmit.



### 23.6.3 Slave receives example

- Step 1: According to the relevant description in GPIO section [9.3.5 Multiplexing function](#), map SCL and SDA to the required pins and configure the SCL and SDA pins for open-drain output mode.
- Step 2: Set `SYSCTRL_APBEN2.I2C` to 1 to enable the I2C module clock.
- Step 3: Configure `I2Cx_SCR1.ADDRCFG` to set the address comparison method, e.g. set `ADDRCFG` to 0 to match address `ADDR0` (7bit).
- Step 4: Configure `I2Cx_SAMR.ADDR0` to set the slave address.
- Step 5: Set `I2Cx_SCR0.SEN` to 1 to enable the I2C slave.
- Step 6: Set `I2Cx_STAR.TXNACK` to 0 to enable the answer flag.
- Step 7: Query and wait for `I2Cx_SISR.ADDR` to change to 1.
- Step 8: Read the `I2Cx_SASR` register and clear the `I2Cx_SISR.ADDR` flag.
- Step 9: Query and wait for `I2Cx_SISR.RXNE` to change to 1 to read the received data from the `I2Cx_SRDR` register.
- Step 10: If the data to be received is not completed, jump to step 9 to continue execution.
- Step 11: Set `I2Cx_STAR.TXNACK` to 1 to return NACK (this must be done when `I2Cx_SCR1.ACKSTALL` is 1).

### 23.6.4 Slave transmits example

- Step 1: According to the relevant description in GPIO section [9.3.5 Multiplexing function](#), map SCL and SDA to the required pins and configure the SCL and SDA pins for open-drain output mode.
- Step 2: Set `SYSCTRL_APBEN2.I2C` to 1 to enable the I2C module clock.
- Step 3: Configure `I2Cx_SCR1.ADDRCFG` to set the address comparison method, e.g. set `ADDRCFG` to 0 to match address `ADDR0` (7bit).
- Step 4: Configure `I2Cx_SAMR.ADDR0` to set the slave address.
- Step 5: Set `I2Cx_SCR0.SEN` to 1 to enable the I2C slave.
- Step 6: Set `I2Cx_STAR.TXNACK` to 0 to enable the answer flag.
- Step 7: Query and wait for `I2Cx_SISR.ADDR` to change to 1.
- Step 8: Read the `I2Cx_SASR` register and clear the `I2Cx_SISR.ADDR` flag.
- Step 9: Query and wait for `I2Cx_SISR.TXE` to change to 1 to write the data to be transmitted to the `I2Cx_STDR` register.
- Step 10: If the data to be transmitted is not completed, jump to step 9 to continue execution.
- Step 11: Set `I2Cx_STAR.TXNACK` to 1 to return NACK (this must be done when `I2Cx_SCR1.ACKSTALL` is 1).



## 23.7 List of registers

I2C1 base address: I2C1\_BASE = 0x4000 5800

I2C2 base address: I2C2\_BASE = 0x4000 5C00

Table 23-7 List of I2C registers

| Register name | Register address  | Register description                 |
|---------------|-------------------|--------------------------------------|
| I2Cx_INSEL    | I2Cx_BASE + 0x08  | Bus input signal selection register  |
| I2Cx_MCR0     | I2Cx_BASE + 0x10  | Master control register 0            |
| I2Cx_MCR1     | I2Cx_BASE + 0x20  | Master control register 1            |
| I2Cx_MCR2     | I2Cx_BASE + 0x24  | Master control register 2            |
| I2Cx_MCR3     | I2Cx_BASE + 0x28  | Master control register 3            |
| I2Cx_MCR4     | I2Cx_BASE + 0x2C  | Master control register 4            |
| I2Cx_MCCR     | I2Cx_BASE + 0x48  | Master timing configuration register |
| I2Cx_MMATCH   | I2Cx_BASE + 0x40  | Master data match register           |
| I2Cx_MFIFO CR | I2Cx_BASE + 0x58  | Master FIFO control register         |
| I2Cx_MFIFO SR | I2Cx_BASE + 0x5C  | Master FIFO status register          |
| I2Cx_MTDR     | I2Cx_BASE + 0x60  | Master transmit data register        |
| I2Cx_MRDR     | I2Cx_BASE + 0x70  | Master receive data register         |
| I2Cx_MDER     | I2Cx_BASE + 0x1C  | Master DMA enable register           |
| I2Cx_MIER     | I2Cx_BASE + 0x18  | Master interrupt enable register     |
| I2Cx_MISR     | I2Cx_BASE + 0x14  | Master interrupt flag register       |
| I2Cx_MICR     | I2Cx_BASE + 0x0C  | Master interrupt flag clear register |
| I2Cx_SCR0     | I2Cx_BASE + 0x110 | Slave control register 0             |
| I2Cx_SCR1     | I2Cx_BASE + 0x124 | Slave control register 1             |
| I2Cx_SCR2     | I2Cx_BASE + 0x24  | Slave control register 2             |
| I2Cx_SCR3     | I2Cx_BASE + 0x128 | Slave control register 3             |
| I2Cx_SAMR     | I2Cx_BASE + 0x140 | Slave address match register         |
| I2Cx_STAR     | I2Cx_BASE + 0x154 | Slave transmit response register     |
| I2Cx_STDR     | I2Cx_BASE + 0x160 | Slave transmit data register         |
| I2Cx_SRDR     | I2Cx_BASE + 0x170 | Slave receive data register          |
| I2Cx_SASR     | I2Cx_BASE + 0x150 | Slave address status register        |
| I2Cx_SDER     | I2Cx_BASE + 0x11C | Slave DMA enable register            |
| I2Cx_SIER     | I2Cx_BASE + 0x118 | Slave interrupt enable register      |
| I2Cx_SISR     | I2Cx_BASE + 0x114 | Slave interrupt flag register        |
| I2Cx_SICR     | I2Cx_BASE + 0x10C | Slave interrupt flag clear register  |



## 23.8 Register description

See section [1 Documentation conventions](#) for abbreviations used in register descriptions.

### 23.8.1 I2Cx\_INSEL Bus input signal selection register

Address offset: 0x08      Reset value: 0x0000 0000

| Bit field | Name     | Permission | Function description                                                                                                                                             |
|-----------|----------|------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:6      | RFU      | -          | Reserved bits, please keep the default value.                                                                                                                    |
| 5:3       | SDAINSEL | RW         | SDA signal input selection<br>000: I2C_SDA pin from GPIO.<br>001: From VC1_OUT<br>010: From VC2_OUT<br>011: From VC3_OUT<br>100: From VC4_OUT<br>Other: Reserved |
| 2:0       | SCLINSEL | RW         | SCL signal input selection<br>000: I2C_SCL pin from GPIO<br>001: From VC1_OUT<br>010: From VC2_OUT<br>011: From VC3_OUT<br>100: From VC4_OUT<br>Other: Reserved  |



### 23.8.2 I2Cx\_MCR0 Master control register 0

Address offset: 0x10      Reset value: 0x0000 0000

| Bit field | Name      | Permission | Function description                                                                           |
|-----------|-----------|------------|------------------------------------------------------------------------------------------------|
| 31:10     | RFU       | -          | Reserved bits, please keep the default value.                                                  |
| 9         | RXFIFORST | R0W1       | Receive FIFO reset control<br>W0: No function.<br>W1: Reset receive FIFO.                      |
| 8         | TXFIFORST | R0W1       | Transmit FIFO reset control<br>W0: No function.<br>W1: Reset transmit FIFO.                    |
| 7:6       | CLKSRC    | RW         | I2C function clock source configuration<br>00: PCLK<br>10: LSE<br>11: HSI                      |
| 5:4       | RFU       | -          | Reserved bits, please keep the default value.                                                  |
| 3         | DBGEN     | RW         | I2C master enable control in debug mode<br>0: Enabled<br>1: Disabled                           |
| 2         | SLEEPEN   | RW         | DeepSleep mode I2C master enable control<br>0: Enabled<br>1: Disabled                          |
| 1         | RESET     | RW         | I2C master soft reset control<br>0: I2C Master in normal mode.<br>1: I2C Master in reset mode. |
| 0         | MEN       | RW         | I2C master enable control<br>0: Disabled<br>1: Enabled                                         |



### 23.8.3 I2Cx\_MCR1 Master control register 1

Address offset: 0x20

Reset value: 0x0000 0000

| Bit field | Name    | Permission | Function description                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|-----------|---------|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:10     | RFU     | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 9         | RDMO    | R0W1       | <p>Receive data match enable control</p> <p>0: Disabled, all received data is stored in the receive FIFO.</p> <p>1: Enabled, when MISR.MATCH is 1, the received data will be stored in the receive FIFO.</p> <p><i>Attention: When changing from enabled to disabled, RDMO should be disabled before clearing the MISR.MATCH flag.</i></p> <p><i>Attention: When changing from disabled to enabled, the MISR.MATCH flag should be cleared before enabling RDMO.</i></p> |
| 8         | CIRFIFO | RW         | <p>Send FIFO auto reload enable control</p> <p>0: Disabled, user needs to write data again after the transmitting a empty FIFO.</p> <p>1: Enabled, automatically load the last written data after transmitting an empty FIFO.</p>                                                                                                                                                                                                                                       |
| 7:0       | RFU     | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                                                                                                                                                           |



### 23.8.4 I2Cx\_MCR2 Master control register 2

Address offset: 0x24      Reset value: 0x0000 0000

| Bit field | Name     | Permission | Function description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|-----------|----------|------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:27     | RFU      | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 26:24     | PINCFG   | RW         | SCL/SDA pin configuration<br>000: Open-drain input output.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 23:19     | RFU      | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 18:16     | MATCFG   | RW         | Data match mode configuration<br>000: Disable data matching.<br>001: Reserved.<br>010: First byte matches MATCH0 or MATCH1.<br>011: Any byte matches MATCH0 or MATCH1.<br>100: First byte matches MATCH0 and second byte matches MATCH1.<br>101: Any byte matches MATCH0 and the next byte matches MATCH1.<br>110: The value obtained by performing a bitwise AND operation on the first byte and MATCH1 is matched with the value obtained by performing a bitwise AND operation on MATCH0 and MATCH1.<br>111: The value obtained by performing a bitwise AND operation on any byte and MATCH1 is matched with the value obtained by performing a bitwise AND operation on MATCH0 and MATCH1. |
| 15        | TRGSEL   | RW         | Master trigger signal output selection<br>0: smbus mext trig<br>1: smbus sext trig                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 14:11     | RFU      | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 10        | TIMECFG  | RW         | Pin low level timeout detection configuration<br>0: Time to detect SCL pin low level.<br>1: Time to detect SCL and SDA pin low level.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 9         | IGNACK   | RW         | NACK ignore enable control<br>0: Normal receive ACK and NACK<br>1: Receive both ACK and NACK as ACK.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 8         | AUTOSTOP | RW         | Automatic generation of STOP signal enable control<br>0: Disabled.<br>1: Enabled, automatically generates STOP signal on bus when transmit FIFO is empty.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 7:3       | RFU      | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 2:0       | PRESCALE | RW         | TCLKD and TCLK division ratio configuration<br>$TCLKD = TCLK/2^{PRESCALE}$<br><i>Note: PRESCALE is 0 to 7, the division ratio is 1 to 128.</i>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |





### 23.8.5 I2Cx\_MCR3 Master control register 3

Address offset: 0x28      Reset value: 0x0000 0000

| Bit field | Name    | Permission | Function description                                                                                                                                                                                                                                                                    |
|-----------|---------|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:28     | RFU     | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                           |
| 27:24     | FLTSDA  | RW         | SDA signal digital filter configuration<br>0000: Disable digital filtering.<br>Other values: Filter the pulse signal whose width is not greater than $t_{CLK} \times FLTSDA$ .                                                                                                          |
| 23:20     | RFU     | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                           |
| 19:16     | FLTSCL  | RW         | SCL signal digital filter configuration<br>0000: Disabled digital filtering.<br>Other values: Filter the pulse signal whose width is not greater than $t_{CLK} \times FLTSCL$ .                                                                                                         |
| 15:12     | RFU     | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                           |
| 11:0      | BUSIDLE | RW         | I2C bus idle detection<br>0000: Disable I2C bus idle detection.<br>Other values: When both SCL and SDA are detected to be high level for longer than the set value, the I2C bus is considered to be in idle state.<br>Setting value = $t_{CLK} \times (BUSIDLE + 1) \times 2^{PRESALE}$ |

### 23.8.6 I2Cx\_MCR4 Master control register 4

Address offset: 0x2C      Reset value: 0x0000 0000

| Bit field | Name   | Permission | Function description                                                                                                                                                                                                                                                                                                                                                                            |
|-----------|--------|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:20     | RFU    | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                                                                                   |
| 19:8      | PINLOW | RW         | SCL/SDA signal low level timeout detection configuration<br>0000: Disable low level timeout detection.<br>Other values: When SCL and SDA are detected to be low for longer than the set value, the MISR.PINLOW flag is set.<br>Setting value = $t_{CLK} \times PINLOW \times 256 \times 2^{PRESALE}$<br><i>Note: The MCR2.TIMCFG control bit determines whether SCL or SCL/SDA is detected.</i> |
| 7:0       | RFU    | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                                                                                   |



### 23.8.7 I2Cx\_MCCR Master timing configuration register

Address offset: 0x48      Reset value: 0x0000 0000

| Bit field | Name    | Permission | Function description                                                                                                                                                                          |
|-----------|---------|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:20     | RFU     | -          | Reserved bits, please keep the default value.                                                                                                                                                 |
| 29:24     | DATAVD  | RW         | SDA data hold time configuration.                                                                                                                                                             |
| 23:22     | RFU     | -          | Reserved bits, please keep the default value.                                                                                                                                                 |
| 21:16     | SETHOLD | RW         | I2C signal setup hold time: <ul style="list-style-type: none"> <li>Initial signal hold time.</li> <li>Repeat start signal build and hold time.</li> <li>Stop signal build-up time.</li> </ul> |
| 15:14     | RFU     | -          | Reserved bits, please keep the default value.                                                                                                                                                 |
| 13:8      | CLKHI   | RW         | SCL high level duration.                                                                                                                                                                      |
| 7:6       | RFU     | -          | Reserved bits, please keep the default value.                                                                                                                                                 |
| 5:0       | CLKLO   | RW         | SCL low level duration.                                                                                                                                                                       |

*Note: Applicable to standard mode, fast mode, fast+ mode.*

*Attention: When MCR0.MEN is set to 1, modification of this register is prohibited.*

### 23.8.8 I2Cx\_MMATCH Master data match register

Address offset: 0x40      Reset value: 0x0000 0000

| Bit field | Name   | Permission | Function description                          |
|-----------|--------|------------|-----------------------------------------------|
| 31:24     | RFU    | -          | Reserved bits, please keep the default value. |
| 23:16     | MATCH1 | RW         | Data to be matched 1.                         |
| 15:8      | RFU    | -          | Reserved bits, please keep the default value. |
| 7:0       | MATCH0 | RW         | Data to be matched 0.                         |

*Note: See the description related to the MCR2.MATCFG bit field for the data matching method.*



### 23.8.9 I2Cx\_MFIFOCR Master FIFO control register

Address offset: 0x58      Reset value: 0x0000 0000

| Bit field | Name    | Permission | Function description                                                                                                                      |
|-----------|---------|------------|-------------------------------------------------------------------------------------------------------------------------------------------|
| 31:28     | RFU     | -          | Reserved bits, please keep the default value.                                                                                             |
| 27:16     | RXWATER | RW         | Receive FIFO waterline configuration.<br>When the number of data in the receive FIFO is greater than RXWATER, MISR.RXNE is set to 1.      |
| 15:2      | RFU     | -          | Reserved bits, please keep the default value.                                                                                             |
| 1:0       | TXWATER | RW         | Transmit FIFO waterline configuration.<br>When the number of data in the transmit FIFO is not greater than TXWATER, MISR.TXE is set to 1. |

### 23.8.10 I2Cx\_MFIFOSR Master FIFO status register

Address offset: 0x5C      Reset value: 0x0000 0000

| Bit field | Name  | Permission | Function description                                       |
|-----------|-------|------------|------------------------------------------------------------|
| 31:28     | RFU   | -          | Reserved bits, please keep the default value.              |
| 27:16     | RXCNT | RO         | The number of data received in the receive FIFO.           |
| 15:2      | RFU   | -          | Reserved bits, please keep the default value.              |
| 1:0       | TXCNT | RO         | The number of data to be transmitted in the transmit FIFO. |

### 23.8.11 I2Cx\_MTDR Master transmit data register

Address offset: 0x60      Reset value: 0x0000 0000

| Bit field | Name | Permission | Function description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|-----------|------|------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:11     | RFU  | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 10:8      | CMD  | WO         | Instructions to be executed<br>000: Transmit one byte of data to the bus MTDR[7:0].<br>001: Receive (MTDR[7:0]+1) bytes of data from bus and store in MRDR register.<br>010: Generate STOP signal on bus.<br>011: Receive (MTDR[7:0]+1) bytes of data from bus, not stored in MRDR register.<br>100: Repeat START signal on bus and send one byte of data MTDR[7:0] to bus expecting ACK from bus.<br>101: Repeat START signal on bus and send one byte data MTDR[7:0] to bus expecting NACK from bus. |
| 7:0       | DATA | WO         | Instruction parameters.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |



### 23.8.12 I2Cx\_MRDR Master receive data register

Address offset: 0x70      Reset value: 0x0000 4000

| Bit field | Name  | Permission | Function description                                                            |
|-----------|-------|------------|---------------------------------------------------------------------------------|
| 31:15     | RFU   | -          | Reserved bits, please keep the default value.                                   |
| 14        | EMPTY | RO         | Receive FIFO empty flag<br>0: Receive FIFO non-empty.<br>1: Receive FIFO empty. |
| 13:8      | RFU   | -          | Reserved bits, please keep the default value.                                   |
| 7:0       | DATA  | RO         | Data received from I2C bus.                                                     |

*Note: Whether data received from the I2C bus is stored in the MRDR register is determined by MTDR.CMD and MCR1.RDMO.*

### 23.8.13 I2Cx\_MDER Master DMA enable register

Address offset: 0x1C      Reset value: 0x0000 0000

| Bit field | Name | Permission | Function description                                                           |
|-----------|------|------------|--------------------------------------------------------------------------------|
| 31:2      | RFU  | -          | Reserved bits, please keep the default value.                                  |
| 1         | RXNE | RW         | Receive FIFO non-empty DMA trigger enable control<br>0: Disabled<br>1: Enabled |
| 0         | TXE  | RW         | Transmit FIFO empty DMA trigger enable control<br>0: Disabled<br>1: Enabled    |



### 23.8.14 I2Cx\_MIER Master interrupt enable register

Address offset: 0x18

Reset value: 0x0000 0000

| Bit field | Name   | Permission | Function description                                                          |
|-----------|--------|------------|-------------------------------------------------------------------------------|
| 31:15     | RFU    | -          | Reserved bits, please keep the default value.                                 |
| 14        | MATCH  | RW         | Receive data match interrupt enable control<br>0: Disabled<br>1: Enabled      |
| 13        | PINLOW | RW         | Pin low level timeout interrupt enable control<br>0: Disabled<br>1: Enabled   |
| 12        | FIFO   | RW         | FIFO error interrupt enable control<br>0: Disabled<br>1: Enabled              |
| 11        | ARBI   | RW         | Arbitration failure interrupt enable control<br>0: Disabled<br>1: Enabled     |
| 10        | NACK   | RW         | NACK interrupt enable control<br>0: Disabled<br>1: Enabled                    |
| 9         | STOP   | RW         | STOP signal detected to interrupt enable control<br>0: Disabled<br>1: Enabled |
| 8         | PACKET | RW         | End of packet detected interrupt enable control<br>0: Disabled<br>1: Enabled  |
| 7:2       | RFU    | -          | Reserved bits, please keep the default value.                                 |
| 1         | RXNE   | RW         | Receive FIFO non-empty interrupt enable control<br>0: Disabled<br>1: Enabled  |
| 0         | TXE    | RW         | Transmit FIFO empty interrupt enable control<br>0: Disabled<br>1: Enabled     |



### 23.8.15 I2Cx\_MISR Master interrupt flag register

Address offset: 0x14      Reset value: 0x0000 0001

| Bit field | Name    | Permission | Function description                                                                                                                                                                      |
|-----------|---------|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:26     | RFU     | -          | Reserved bits, please keep the default value.                                                                                                                                             |
| 25        | BUSBUSY | RO         | I2C bus busy flag<br>0: Idle.<br>1: Busy.                                                                                                                                                 |
| 24        | MSTBUSY | RO         | I2C Master busy flag<br>0: Idle.<br>1: Busy.                                                                                                                                              |
| 23:15     | RFU     | -          | Reserved bits, please keep the default value.                                                                                                                                             |
| 14        | MATCH   | RO         | Received data match flag<br>0: No matched data received.<br>1: Matched data received.                                                                                                     |
| 13        | PINLOW  | RO         | Pin low level duration timeout flag<br>0: Pin low level duration has not exceeded the definition of MCR4.PINLOW.<br>1: Pin low level duration has exceeded the definition of MCR4.PINLOW. |
| 12        | FIFO    | RO         | FIFO error flag<br>0: FIFO error status not detected.<br>1: FIFO error status detected.                                                                                                   |
| 11        | ARBI    | RO         | Arbitration failure flag<br>0: Master succeeds in contention for the bus.<br>1: Master lost arbitration in contention for the bus.                                                        |
| 10        | NACK    | RO         | NACK flag<br>0: Master receives expected ACK or NACK.<br>1: Master does not receive expected ACK or NACK.<br><i>Note: This bit field is not affected by the IGNACK bit field.</i>         |
| 9         | STOP    | RO         | STOP signal flag detected<br>0: No STOP signal detected on bus.<br>1: STOP signal detected on bus.                                                                                        |
| 8         | PACKET  | RO         | End of packet flag detected<br>0: No Repeated START or STOP signal detected on bus.<br>1: Repeated START signal or STOP signal detected on bus.                                           |
| 7:2       | RFU     | -          | Reserved bits, please keep the default value.                                                                                                                                             |
| 1         | RXNE    | RO         | Receive FIFO non-empty flag<br>0: Number of data in receive FIFO $\leq$ MFIFOCR.RXWATER.<br>1: Number of data in receive FIFO $>$ MFIFOCR.RXWATER.                                        |



| Bit field | Name | Permission | Function description                                                                                                                       |
|-----------|------|------------|--------------------------------------------------------------------------------------------------------------------------------------------|
| 0         | TXE  | RO         | Transmit FIFO empty flag<br>0: Number of data in transmit FIFO > MFIFOCR.TXWATER.<br>1: Number of data in transmit FIFO ≤ MFIFOCR.TXWATER. |

### 23.8.16 I2Cx\_MICR Master interrupt flag clear register

Address offset: 0x0C

Reset value: 0x0000 7F04

| Bit field | Name   | Permission | Function description                                                                                                                                                                   |
|-----------|--------|------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:15     | RFU    | -          | Reserved bits, please keep the default value.                                                                                                                                          |
| 14        | MATCH  | R1W0       | Received data match flag cleared<br>W0: Clear received data match flag.<br>W1: No function.                                                                                            |
| 13        | PINLOW | R1W0       | Pin low level duration timeout flag cleared<br>W0: Clear pin low level duration timeout flag.<br>W1: No function.                                                                      |
| 12        | FIFO   | R1W0       | FIFO error flag cleared<br>W0: Clear FIFO error flag.<br>W1: No function.<br><i>Note: The MISR.FIFO flag cannot be cleared with MICR.FIFO until TXFIFO is reset by MCR0.TXFIFORST.</i> |
| 11        | ARBI   | R1W0       | Arbitration failure flag cleared<br>W0: Clear arbitration failure flag.<br>W1: No function.                                                                                            |
| 10        | NACK   | R1W0       | NACK flag cleared<br>W0: Clear NACK flag.<br>W1: No function.                                                                                                                          |
| 9         | STOP   | R1W0       | STOP signal detected flag cleared<br>W0: Clear STOP signal flag.<br>W1: No function.                                                                                                   |
| 8         | PACKET | R1W0       | End of packet detected flag cleared<br>W0: Clear the end-of-packet flag.<br>W1: No function.                                                                                           |
| 7:0       | RFU    | -          | Reserved bits, please keep the default value.                                                                                                                                          |



### 23.8.17 I2Cx\_SCR0 Slave control register 0

Address offset: 0x110

Reset value: 0x0000 0000

| Bit field | Name      | Permission | Function description                                                                                                                                                                                                                                                                              |
|-----------|-----------|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:10     | RFU       | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                     |
| 9         | RXFIFORST | R0W1       | Receive FIFO reset control<br>W0: No function.<br>W1: Reset receive FIFO.                                                                                                                                                                                                                         |
| 8         | TXFIFORST | R0W1       | Transmit FIFO reset control.<br>W0: No function.<br>W1: Reset transmit FIFO.                                                                                                                                                                                                                      |
| 7:6       | CLKSRC    | RW         | I2C Slave clock extension and filter clock source selection<br>00: PCLK<br>10: LSE<br>11: LSI                                                                                                                                                                                                     |
| 5         | RFU       | RW         | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                     |
| 4         | FLTEN     | RW         | Digital filter enable control<br>0: Disabled<br>1: Enabled<br><i>Note: This bit field can only be modified if I2C Slave is not enabled.</i><br><i>Note: When the filtered clock source of I2C Slave is PCLK, the digital filtering function needs to be turned off before entering DeepSleep.</i> |
| 3:2       | RFU       | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                     |
| 1         | RESET     | RW         | I2C Slave soft reset control<br>0: I2C Slave in normal mode.<br>1: I2C Slave in reset mode.                                                                                                                                                                                                       |
| 0         | SEN       | RW         | I2C Slave enable control<br>0: Disabled<br>1: Enabled                                                                                                                                                                                                                                             |





### 23.8.18 I2Cx\_SCR1 Slave control register 1

Address offset: 0x124      Reset value: 0x0000 0000

| Bit field | Name     | Permission | Function description                                                                                                                                                                                                                                                                                                                                                                                                        |
|-----------|----------|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:19     | RFU      | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                                                                                                               |
| 18:16     | ADDRCFG  | RW         | Address comparison method configuration<br>000: Compare ADDR0 (7bit)<br>001: Compare ADDR0 (10bit)<br>010: Compare ADDR0 (7bit)/Compare ADDR1 (7bit)<br>011: Compare ADDR0 (10bit) /Compare ADDR1 (10bit)<br>100: Compare ADDR0 (7bit) /Compare ADDR1 (10bit)<br>101: Compare ADDR0 (10bit)/Compare ADDR1 (7bit)<br>110: Compare address space ADDR0~ADDR1, 7bit mode<br>111: Compare address space ADDR0~ADDR1, 10bit mode |
| 15        | TRGSEL   | RW         | Slave Trigger signal output selection<br>0: smbus mext trig<br>1: smbus sext trig                                                                                                                                                                                                                                                                                                                                           |
| 14:13     | RFU      | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                                                                                                               |
| 12        | IGNACK   | RW         | NACK ignore enable control<br>0: NACK signal received, end of data transmission.<br>1: NACK signal received, continue data transmission.<br><i>Note: Fast+ mode requires IGNACK to be set to 1.</i>                                                                                                                                                                                                                         |
| 11        | RXCFG    | RW         | Read SRDR register return data configuration<br>0: Return received data.<br>1: Returns received data (when SISR.ADDR = 0); Returns SASR register value (when SISR.ADDR = 1).                                                                                                                                                                                                                                                |
| 10        | TXCFG    | RW         | TXE flag set to 1 mode configuration<br>0: TXE flag set when data needs to be transmitted to I2C bus.<br>1: TXE flag set when transmit FIFO is empty.                                                                                                                                                                                                                                                                       |
| 9         | ALERTEN  | RW         | SMBus Alert detection enable control<br>0: Disabled<br>1: Enabled                                                                                                                                                                                                                                                                                                                                                           |
| 8         | GCEN     | RW         | Broadcast address detection enable control<br>0: Disabled<br>1: Enabled                                                                                                                                                                                                                                                                                                                                                     |
| 7:4       | RFU      | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                                                                                                               |
| 3         | ACKSTALL | RW         | ACK clock stretching enable control<br>0: Disabled.<br>1: Enabled, write ACK/NACK to STAR register before releasing bus.                                                                                                                                                                                                                                                                                                    |
| 2         | TXSTALL  | RW         | Transmit data clock stretching enable control<br>0: Disabled.<br>1: Enabled, clear SISR.TXE flag before releasing bus.                                                                                                                                                                                                                                                                                                      |



| Bit field | Name     | Permission | Function description                                                                                                           |
|-----------|----------|------------|--------------------------------------------------------------------------------------------------------------------------------|
| 1         | RXSTALL  | RW         | Receive data clock stretching enable control<br>0: Disabled.<br>1: Enabled, clear the SISR.RXNE flag before releasing the bus. |
| 0         | ADRSTALL | RW         | Address valid clock stretching enable control<br>0: Disabled.<br>1: Enabled, clear SISR.ADDR flag before releasing bus.        |

### 23.8.19 I2Cx\_SCR2 Slave control register 2

Address offset: 0x24      Reset value: 0x0000 0000

| Bit field | Name   | Permission | Function description                                                                                                                    |
|-----------|--------|------------|-----------------------------------------------------------------------------------------------------------------------------------------|
| 31:27     | RFU    | -          | Reserved bits, please keep the default value.                                                                                           |
| 26:24     | PINCFG | RW         | SCL/SDA pin configuration<br>000: Open-drain input output.<br><i>Note: Only open-drain inputs and outputs support clock stretching.</i> |
| 23:0      | RFU    | -          | Reserved bits, please keep the default value.                                                                                           |

*Note: The I2C\_SCR2 register and the I2C\_MCR2 register use the same address.*



### 23.8.20 I2Cx\_SCR3 Slave control register 3

Address offset: 0x128      Reset value: 0x0000 0000

| Bit field | Name    | Permission | Function description                                                                                                                                                                                                                        |
|-----------|---------|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:28     | RFU     | -          | Reserved bits, please keep the default value.                                                                                                                                                                                               |
| 27:24     | FLTSDA  | RW         | SDA signal digital filter configuration<br>0000: Disable digital filtering.<br>Other values: Filter pulse signals whose width is not greater than $t_{CLK} \times FLTSDA$ .<br><b>Note: FLTSDA must be greater than or equal to FLTSCl.</b> |
| 23:20     | RFU     | -          | Reserved bits, please keep the default value.                                                                                                                                                                                               |
| 19:16     | FLTSCl  | RW         | SCL signal digital filter configuration<br>0000: Disable digital filtering.<br>Other values: Filter the pulse signal whose width is not greater than $t_{CLK} \times FLTSCl$ .                                                              |
| 15:14     | RFU     | -          | Reserved bits, please keep the default value.                                                                                                                                                                                               |
| 13:8      | DATAVD  | RW         | SDA data effective delay time configuration<br>The data effective delay time is: $t_{CLK} \times (DATAVD+3)$ .<br><b>Note: The data effective delay must be configured to be less than the minimum SCL low period.</b>                      |
| 7:4       | RFU     | -          | Reserved bits, please keep the default value.                                                                                                                                                                                               |
| 3:0       | CLKHOLD | RW         | SCL clock hold time configuration<br>SCL clock hold time is: $t_{CLK} \times (CLKHOLD+3)$ .                                                                                                                                                 |

### 23.8.21 I2Cx\_SAMR Slave address match register

Address offset: 0x140      Reset value: 0x0000 0000

| Bit field | Name  | Permission | Function description                          |
|-----------|-------|------------|-----------------------------------------------|
| 31:27     | RFU   | -          | Reserved bits, please keep the default value. |
| 26:17     | ADDR1 | RW         | Address 1 configuration value to be matched . |
| 16:11     | RFU   | -          | Reserved bits, please keep the default value. |
| 10:1      | ADDR0 | RW         | Address 0 configuration value to be matched.  |
| 0         | RFU   | -          | Reserved bits, please keep the default value. |



### 23.8.22 I2Cx\_STAR Slave transmit response register

Address offset: 0x154      Reset value: 0x0000 0000

| Bit field | Name   | Permission | Function description                                                                                                                                                     |
|-----------|--------|------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:1      | RFU    | -          | Reserved bits, please keep the default value.                                                                                                                            |
| 0         | TXNACK | RW         | Slave transmit response configuration<br>0: ACK is transmitted during the transmission response phase.<br>1: NACK is transmitted during the response transmission phase. |

### 23.8.23 I2Cx\_STDR Slave transmit data register

Address offset: 0x160      Reset value: 0x0000 0000

| Bit field | Name | Permission | Function description                          |
|-----------|------|------------|-----------------------------------------------|
| 31:8      | RFU  | -          | Reserved bits, please keep the default value. |
| 7:0       | DATA | WO         | Data to be transmitted to the bus.            |

### 23.8.24 I2Cx\_SRDR Slave receive data register

Address offset: 0x170      Reset value: 0x0000 4000

| Bit field | Name    | Permission | Function description                                                                                                                                             |
|-----------|---------|------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:8      | RFU     | -          | Reserved bits, please keep the default value.                                                                                                                    |
| 15        | SOF     | RO         | Frame start flag<br>0: The received data is not the first byte after the (Repeated) START.<br>1: The received data is the first byte after the (Repeated) START. |
| 14        | RXEMPTY | RO         | Receive FIFO empty flag<br>0: Receive FIFO non-empty.<br>1: Receive FIFO empty.                                                                                  |
| 13:8      | RFU     | -          | Reserved bits, please keep the default value.                                                                                                                    |
| 7:0       | DATA    | RO         | Data received from I2C bus.                                                                                                                                      |



### 23.8.25 I2Cx\_SASR Slave address status register

Address offset: 0x150      Reset value: 0x0000 4000

| Bit field | Name  | Permission | Function description                                                                                                                                                                                                        |
|-----------|-------|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:15     | RFU   | -          | Reserved bits, please keep the default value.                                                                                                                                                                               |
| 14        | ANV   | RO         | Address match status<br>0: Valid address received.<br>1: Invalid address received.                                                                                                                                          |
| 13:11     | RFU   | -          | Reserved bits, please keep the default value.                                                                                                                                                                               |
| 10:0      | RADDR | RO         | Received address value and RW bit: <ul style="list-style-type: none"> <li>7bit mode: RADDR[7:1] stores address, RADDR[0] stores RW bit.</li> <li>10bit mode: RADDR[10:1] stores address, RADDR[0] stores RW bit.</li> </ul> |

### 23.8.26 I2Cx\_SDER Slave DMA enable register

Address offset: 0x11C      Reset value: 0x0000 0000

| Bit field | Name | Permission | Function description                                                  |
|-----------|------|------------|-----------------------------------------------------------------------|
| 31:3      | RFU  | -          | Reserved bits, please keep the default value.                         |
| 2         | ADDR | RW         | Address valid DMA trigger enable control<br>0: Disabled<br>1: Enabled |
| 1         | RXNE | RW         | Receive data DMA trigger enable control<br>0: Disabled<br>1: Enabled  |
| 0         | TXE  | RW         | Transmit data DMA trigger enable control<br>0: Disabled<br>1: Enabled |

### 23.8.27 I2Cx\_SIER Slave interrupt enable register

Address offset: 0x118      Reset value: 0x0000 0000

| Bit field | Name  | Permission | Function description                                                            |
|-----------|-------|------------|---------------------------------------------------------------------------------|
| 31:16     | RFU   | -          | Reserved bits, please keep the default value.                                   |
| 15        | ALERT | RW         | SMBus Alert address match interrupt enable control<br>0: Disabled<br>1: Enabled |
| 14        | GC    | RW         | Broadcast address match interrupt enable control<br>0: Disabled<br>1: Enabled   |



| Bit field | Name    | Permission | Function description                                                                    |
|-----------|---------|------------|-----------------------------------------------------------------------------------------|
| 13        | AM1     | RW         | Address 1 match interrupt enable control<br>0: Disabled<br>1: Enabled                   |
| 12        | AM0     | RW         | Address 0 match interrupt enable control<br>0: Disabled<br>1: Enabled                   |
| 11        | FIFO    | RW         | FIFO error interrupt enable control<br>0: Disabled<br>1: Enabled                        |
| 10        | BIT     | RW         | Bit conflict error interrupt enable control<br>0: Disabled<br>1: Enabled                |
| 9         | STOP    | RW         | STOP signal detected to interrupt enable control<br>0: Disabled<br>1: Enabled           |
| 8         | RESTART | RW         | Repeated START signal detected to interrupt enable control<br>0: Disabled<br>1: Enabled |
| 7:4       | RFU     | -          | Reserved bits, please keep the default value.                                           |
| 3         | TACK    | RW         | Response interrupt enable control needs to be transmitted<br>0: Disabled<br>1: Enabled  |
| 2         | ADDR    | RW         | Address valid interrupt enable control<br>0: Disabled<br>1: Enabled                     |
| 1         | RXNE    | RW         | Receive data interrupt enable control<br>0: Disabled<br>1: Enabled                      |
| 0         | TXE     | RW         | Transmit data interrupt enable control<br>0: Disabled<br>1: Enabled                     |



### 23.8.28 I2Cx\_SISR Slave interrupt flag register

Address offset: 0x114      Reset value: 0x0000 0000

| Bit field | Name    | Permission | Function description                                                                                                                                   |
|-----------|---------|------------|--------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:26     | RFU     | -          | Reserved bits, please keep the default value.                                                                                                          |
| 25        | BUSBUSY | RO         | I2C bus busy flag<br>0: Idle.<br>1: Busy.                                                                                                              |
| 24        | SLVBUSY | RO         | I2C Slave busy flag<br>0: Idle.<br>1: Busy.                                                                                                            |
| 23:16     | RFU     | -          | Reserved bits, please keep the default value.                                                                                                          |
| 15        | ALERT   | RO         | SMBus Alert address match flag<br>0: No match.<br>1: Match.                                                                                            |
| 14        | GC      | RO         | Broadcast address match flag<br>0: No match.<br>1: Match.                                                                                              |
| 13        | AM1     | RO         | Address 1 match flag<br>0: No match.<br>1: Match.                                                                                                      |
| 12        | AM0     | RO         | Address 0 match flag<br>0: No match.<br>1: Match.                                                                                                      |
| 11        | FIFO    | RO         | FIFO error flag<br>0: FIFO has no errors.<br>1: FIFO error.                                                                                            |
| 10        | BIT     | RO         | Bit conflict error flag<br>0: Not detected.<br>1: Detected.                                                                                            |
| 9         | STOP    | RO         | STOP signal flag detected<br>0: Not detected.<br>1: Detected.                                                                                          |
| 8         | RESTART | RO         | Repeated START flag detected<br>0: Not detected.<br>1: Detected.                                                                                       |
| 7:4       | RFU     | -          | Reserved bits, please keep the default value.                                                                                                          |
| 3         | TACK    | RO         | Need to transmit response flag<br>0: No need to transmit response through STAR registers.<br>1: Need to transmit a response through the STAR register. |



| Bit field | Name | Permission | Function description                                                                                                                          |
|-----------|------|------------|-----------------------------------------------------------------------------------------------------------------------------------------------|
| 2         | ADDR | RO         | Address valid flag<br>0: No matching address received.<br>1: Matching address has been received, see SASR register for address value details. |
| 1         | RXNE | RO         | Receive data flag<br>0: Cannot receive data.<br>1: Can receive data.                                                                          |
| 0         | TXE  | RO         | Transmit data flag<br>0: Transmit data has not been requested.<br>1: Transmit data has been requested.                                        |

### 23.8.29 I2Cx\_SICR Slave interrupt flag clear register

Address offset: 0x10C

Reset value: 0x0000 0F00

| Bit field | Name    | Permission | Function description                                                                      |
|-----------|---------|------------|-------------------------------------------------------------------------------------------|
| 31:12     | RFU     | -          | Reserved bits, please keep the default value.                                             |
| 11        | FIFO    | R1W0       | FIFO error flag cleared<br>W0: Clear FIFO error flag.<br>W1: No function.                 |
| 10        | BIT     | R1W0       | Bit conflict error flag cleared<br>W0: Clear bit conflict error flag.<br>W1: No function. |
| 9         | STOP    | R1W0       | STOP signal detected flag cleared<br>W0: Clear STOP signal flag.<br>W1: No function.      |
| 8         | RESTART | R1W0       | Repeated START signal flag cleared<br>W0: Clear RESTART signal flag.<br>W1: No function.  |
| 7:0       | RFU     | -          | Reserved bits, please keep the default value.                                             |





## 24 Infrared modulation transmitter (IR)

### 24.1 Overview

The CW32L012 integrates an internal infrared modulated transmitter (IR), which can be used in conjunction with the timer, UART and IRSW soft control bits to easily realize various standard PWM or PPM encoding methods, as well as infrared modulated transmission of UART data.



## 24.2 Main features

- Supports SIR of IrDA standard 1.0.
- Maximum data rate 115.2kbps.
- Can adapt to high and low level infrared emission tube.



## 24.3 Functional description

When implementing an infrared modulation transmitter, a timer channel is used to generate a square wave signal with a fixed frequency, and another timer or UART is used to generate modulated data, which is output from the IR\_OUT pin after 'AND' or 'OR' operation with IRSW soft control bits. For specific IR\_OUT pins, please refer to the pin definition in the data manual.

The MOD bit field of the IR modulation control register IRMOD\_CR is used to select the source of the carrier signal and the data signal, and the 'AND' 'OR' operation with the IRSW soft control bit. The selection of the 'AND' 'OR' is determined by the drive level of the user's hardware IR emitter. The IR\_OUT output can be controlled inversely through the INV bit field.

The carrier signal frequency can be set by the user, the most common carrier frequency is 38KHz.

In order to reduce the transmission power consumption, the duty cycle of the carrier is generally set to about 20%, and the timer that generates the carrier can use the PWM mode, please refer to chapter [16 General-purpose timer \(GTIM\)](#). In order to increase the transmission distance, the duty cycle can be appropriately increased. In the IrDA standard 1.0, in order to achieve a fast IR communication rate, the width of the pulse is specified as 3/16 of a bit period or a fixed 1.63μs, and the minimum cannot be lower than 1.41μs.

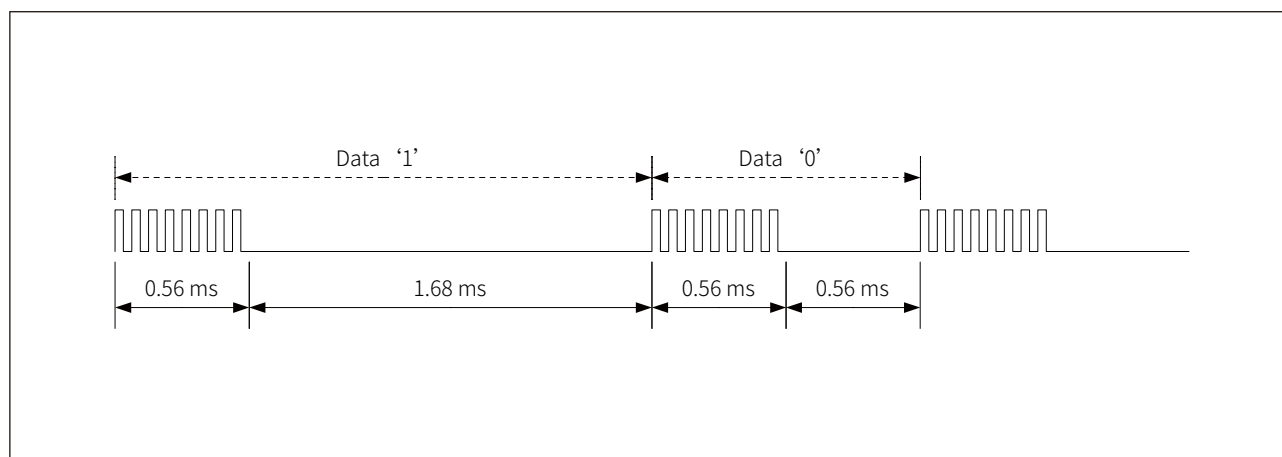


### 24.3.1 Infrared modulation mode

There are two common IR infrared encoding methods, pulse width modulation PWM and pulse position modulation PPM.

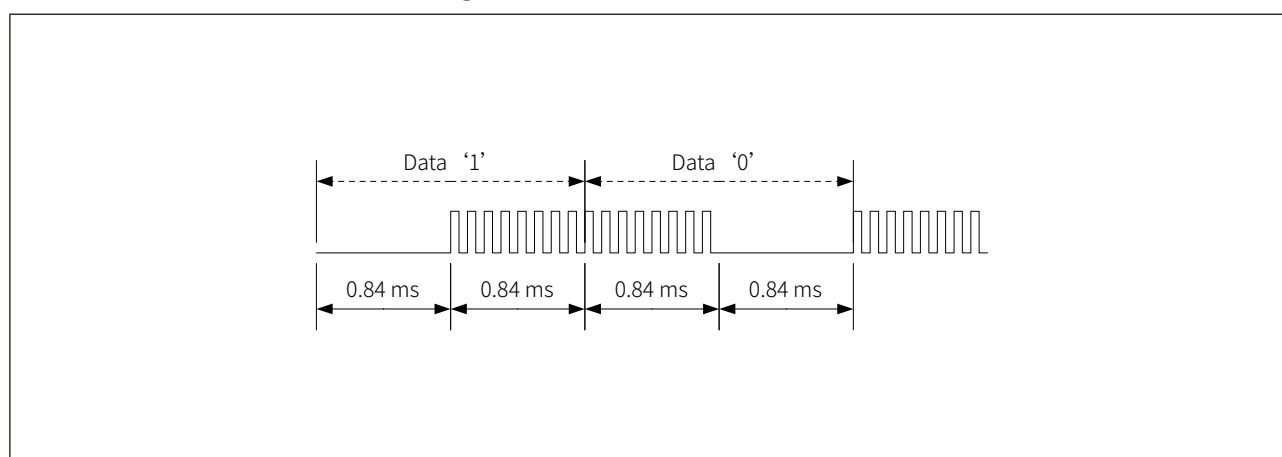
PWM pulse width modulation to represent '0' and '1' with the duty cycle of the transmitted infrared carrier. For example, UPD6121, the carrier transmits 0.56ms, and does not transmit 0.56ms, which means '0'; the carrier transmits 0.56ms, and does not transmit 1.68ms, which means '1'. The reference diagram is shown in the following figure:

Figure 24-1 PWM modulation mode



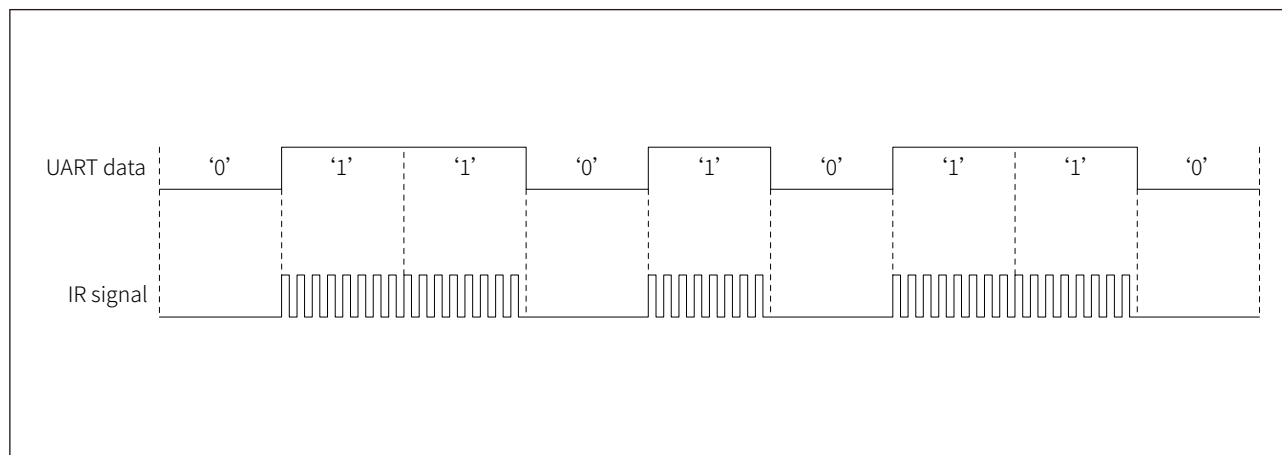
PPM pulse position modulation to represent '0' and '1' with the position of the transmitted carrier. It is '0' from transmitting carrier to not transmitting carrier, and '1' from never transmitting carrier to transmitting carrier. For example, SAA3010, the carrier transmits 0.84ms, and does not transmit 0.84ms, which means '0'; does not transmit 0.84ms, and the carrier transmits 0.84ms, means '1'. The reference diagram is shown in the following figure:

Figure 24-2 PPM modulation mode



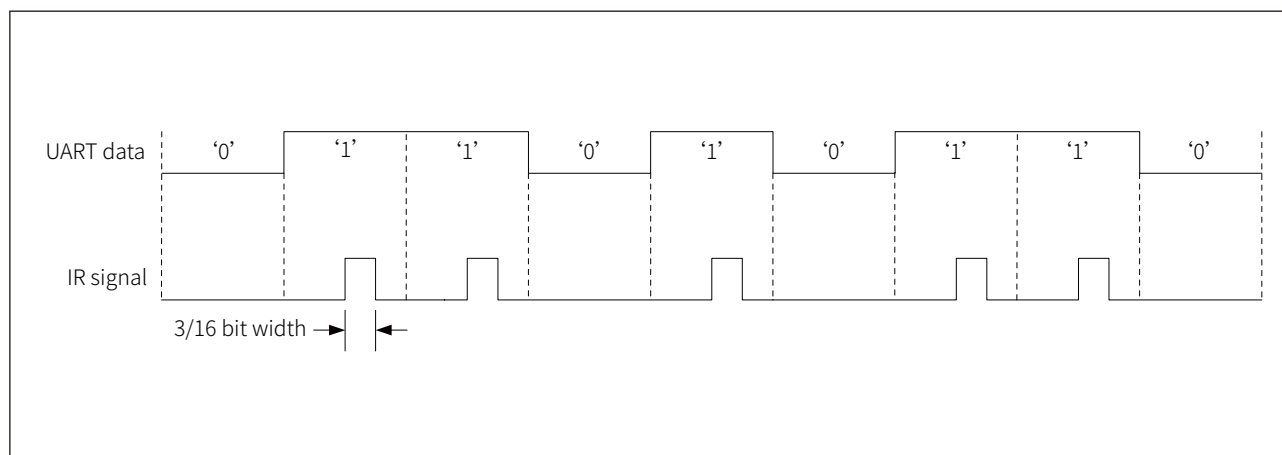
UART modulation is a relatively simple infrared modulation method, which directly performs 'AND' or 'OR' operation on serial data and carrier signal. When the UART serial baud rate is lower than the carrier frequency, the reference schematic diagram is shown in the following figure:

Figure 24-3 UART modulation mode



When the UART serial baud rate is higher than the carrier frequency, the carrier pulse width should be set to 3/16 of the bit width, and the carrier frequency should be kept consistent with the serial frequency. For example, when the UART baud rate is set to 115.2kbps, the carrier frequency is also set to 115.2kHz, and the width of the carrier pulse is  $1.63\mu\text{s}$  ( $3/16 \times 1/115200\text{Hz} = 1.63\mu\text{s}$ ). The reference schematic diagram is shown in the following figure:

Figure 24-4 High-speed UART modulation mode



For the convenience of debugging, usually a standard data frame also includes boot code, end bit, retransmission code, etc. Please refer to the relevant data manual by yourself.

### 24.3.2 Infrared modulation initialization configuration

The initialization process of a complete IR infrared modulation transmitter should include the configuration of timer, UART, and GPIO. The reference steps are as follows:

- Step 1:** Set the SYSCTRL\_APBEN1 and SYSCTRL\_APBEN2 registers to enable the timer and UART peripheral clock to be selected.
- Step 2:** Set the allowed bits of the GPIO clock corresponding to the IR\_OUT pin in the SYSCTRL\_AHBEN register.
- Step 3:** Configure the timer selected as the carrier, set the carrier frequency, duty cycle (common frequency 38KHz, 1/3 duty cycle), and set the timer working mode (PWM output is recommended).
- Step 4:** Configure the timer or UART selected as data, set the bit width, and UART working mode (no need to define output pins).
- Step 5:** Set GPIOx\_ANALOG as IR\_OUT output port as digital mode, set GPIOx\_DIR as output, and set appropriate GPIOx\_OPENDRAIN, GPIOx\_PUR.
- Step 6:** Start the timer and UART as carrier and data.
- Step 7:** Set the MOD bit field of the IR infrared modulation control register IRMOD\_CR, and set the infrared modulation mode configuration.
- Step 8:** Set the GPIOx\_AFRH/GPIOx\_AFRL of the IR\_OUT output port to IR\_OUT, and enable the IR multiplexing function.

### 24.3.3 Infrared reception

CW32L012 does not have an IR receiving demodulation module inside. In IR receiving applications, it is necessary to use an integrated infrared receiving head with demodulation function. With the capture function of GTIM (UART mode can directly use the RXD pin input), it can be easily realized IR receiving function.



## 24.4 IRMOD\_CR Infrared modulation control register

Address: 0x4000 4000 + 0x80      Reset value: 0x0000 0000

| Bit field | Name | Permission | Function description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|-----------|------|------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:6      | RFU  | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 5         | INV  | RW         | IR output reverse control<br>0: Positive output.<br>1: Reverse output.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 4         | IRSW | RW         | Infrared modulation soft control bit<br>Functions detailed in MOD bit field.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 3:0       | MOD  | RW         | Infrared modulation mode configuration<br>0000: GTIM1_CH1 & GTIM2_CH1 & IRSW<br>0001: GTIM1_CH1 & GTIM2_CH2 & IRSW<br>0010: GTIM1_CH2 & GTIM2_CH1 & IRSW<br>0011: GTIM1_CH2 & GTIM2_CH2 & IRSW<br>0100: GTIM1_CH1   GTIM2_CH1   IRSW<br>0101: GTIM1_CH1   GTIM2_CH2   IRSW<br>0110: GTIM1_CH2   GTIM2_CH1   IRSW<br>0111: GTIM1_CH2   GTIM2_CH2   IRSW<br>1000: UART1_TXD & GTIM1_CH1 & IRSW<br>1001: UART1_TXD   GTIM1_CH1   IRSW<br>1010: UART1_TXD & GTIM1_CH2 & IRSW<br>1011: UART1_TXD   GTIM1_CH2   IRSW<br>1100: UART2_TXD & GTIM2_CH1 & IRSW<br>1101: UART2_TXD   GTIM2_CH1   IRSW<br>1110: UART2_TXD & GTIM2_CH2 & IRSW<br>1111: UART2_TXD   GTIM2_CH2   IRSW |



## 25 Analog-to-digital converter (ADC)

### 25.1 Overview

CW32L012 integrates 2 successive approximation analog-to-digital converter (SAR ADC) with 12-bit precision and maximum conversion speed of 1M SPS, which can convert up to 16 analog signals into digital signals. The vast majority of signals in the real world are analog, such as light, electricity, sound, image signals, etc., which must be converted into digital signals by ADC before they can be digitally processed by MCU.





## 25.2 Main features

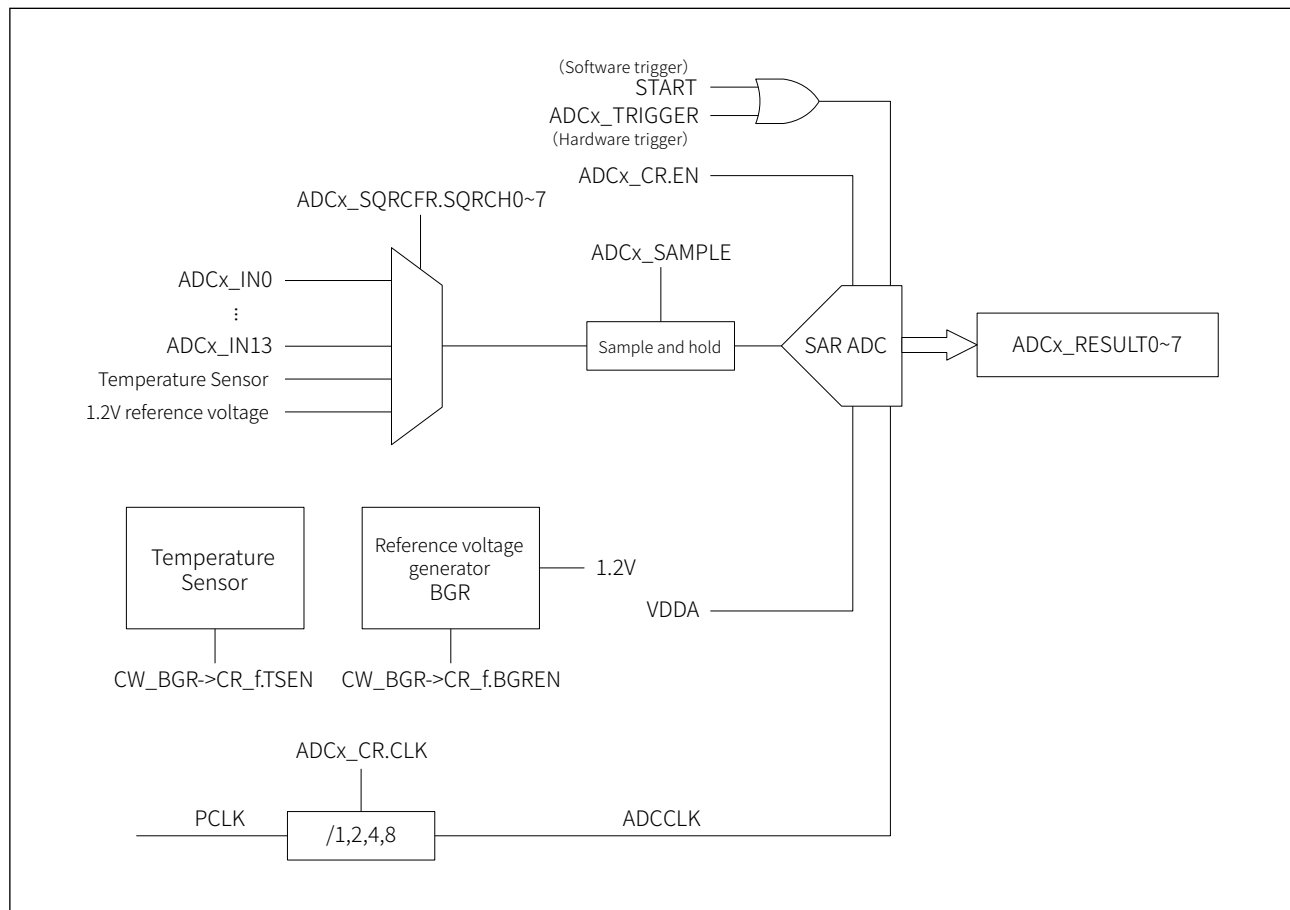
- 12-bit precision.
- Individually configurable conversion speeds for each sequence channel, up to 1M SPS.
- 16 channels of input conversion:
  - 12 external pins input.
  - DAC\_OUT1 internal signal.
  - DAC\_OUT2 internal signal.
  - Built-in temperature sensor.
  - Built-in BGR 1.2V reference.
- VDDA supply voltage as reference voltage source ( $V_{ref}$ )
- Sampling voltage input range:  $0 \sim V_{ref}$ .
- Supports sequence channel conversion modes:
  - Supports single and continuous conversions.
  - Supports up to 8 conversion channels, each channel can select one of the 16 conversion sources.
- Supports input channel voltage threshold monitoring.
- Supports on-chip peripherals to automatically trigger ADC conversion.



## 25.3 Functional block diagram

The ADC functional block diagram is shown in the following figure:

Figure 25-1 ADC functional block diagram

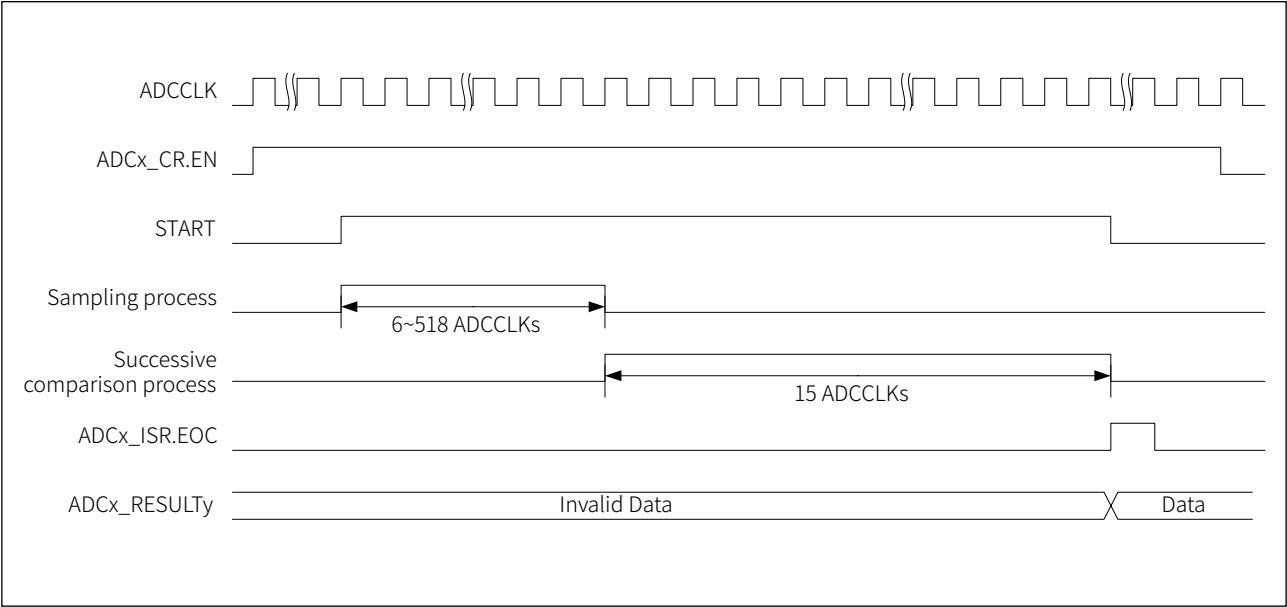


## 25.4 Conversion timing, conversion speed, conversion accuracy, and conversion results

### 25.4.1 Conversion timing

The conversion timing of the ADC is shown in the following figure:

Figure 25-2 ADC conversion timing diagram



Write 1 to the EN bit field of the ADC control register ADCx\_CR (x=1, 2) to enable the ADC module.

About 1μs after ADCx\_CR.EN changes from 0 to 1, the initialization of the analog circuit is completed, and the ADC conversion can be started.

Write 1 to the START bit field of the ADC start register ADCx\_START to start the ADC conversion, and the hardware will automatically clear it after the conversion is completed.

The ADC working clock ADCCLK is obtained by dividing the system clock PCLK by the prescaler. The CLK bit field of the control register ADCx\_CR can be selected to divide the frequency from 1/2/4/8, as shown in the following table:

Table 25-1 ADC clock configuration table

| ADCx_CR.CLK | ADCCLK |
|-------------|--------|
| 000         | PCLK   |
| 001         | PCLK/2 |
| 010         | PCLK/4 |
| 011         | PCLK/8 |

A complete ADC conversion requires 21~533 ADCCLK clock cycles, including sampling phase and successive comparison phase:

- Sampling phase: 6~518 ADCCLK clock cycles are required. The sample period for each sequence conversion channel can be configured individually and is set through the SQRCHy bit field of the sample period configuration register ADCx\_SAMPLE as shown in the following table:

Table 25-2 ADC sampling period of each sequence channel

| ADCx_SAMPLE.SQRCHy | Fixed sampling period (number of ADCCLKs) |
|--------------------|-------------------------------------------|
| 0000               | 6                                         |
| 0001               | 7                                         |
| 0010               | 9                                         |
| 0011               | 12                                        |
| 0100               | 18                                        |
| 0101               | 24                                        |
| 0110               | 30                                        |
| 0111               | 42                                        |
| 1000               | 54                                        |
| 1001               | 70                                        |
| 1010               | 102                                       |
| 1011               | 134                                       |
| 1100               | 166                                       |
| 1101               | 198                                       |
| 1110               | 262                                       |
| 1111               | 518                                       |

The sampling moment and sampling period of ADC can be output through the external ADCx\_SAM (x = 1, 2) pin, please refer to the *Datasheet* about pin definition for the specific pin.

ADC sampling period length is determined by the user's speed requirement for sampling and the electrical characteristics of the sampled signal, the user should choose the appropriate sampling period to achieve the best conversion effect.

- Successive comparison phase: 15 ADCCLK clock cycles are required.

After each ADC conversion is completed, the conversion completion flag bit ADCx\_ISR.EOC will be set to 1 by hardware, and the ADC conversion result is stored in the corresponding sequence conversion result register ADCx\_RESULTy (y = 0, 1, 2, 3, 4, 5, 6, 7), the user can clear this flag bit by setting ADCx\_ICR.EOC to 0.



### 25.4.2 Conversion speed

ADC conversion speed is closely related to VDDA supply voltage, and the maximum conversion speed under various conditions is shown in the following table:

Table 25-3 ADC conversion speed and voltage comparison table

| VDDA voltage | Maximum conversion speed | Maximum ADCCLK frequency |
|--------------|--------------------------|--------------------------|
| 1.7V ~ 1.8V  | 200K SPS                 | 6MHz                     |
| 1.8V ~ 2.8V  | 500K SPS                 | 12MHz                    |
| 2.8V ~ 3.3V  | 1M SPS                   | 24MHz                    |
| 3.3V ~ 5.5V  | 1M SPS                   | 48MHz                    |

The corresponding relationship between the conversion speed of the ADC and the working clock ADCCLK is as follows:

$$\text{ADC conversion speed} = f_{\text{ADCCLK}}/N_T$$

Where,  $f_{\text{ADCCLK}}$  is the clock frequency of ADCCLK, and  $N_T$  is ADCCLK number required for one ADC conversion.

### 25.4.3 Conversion results

After the ADC completes each conversion, the conversion result of the currently converted sequence channel SQRCH<sub>y</sub> ( $y = 0, 1, 2, 3, 4, 5, 6, 7$ ) is stored in the corresponding ADC<sub>x</sub>\_RESULT<sub>y</sub> ( $y = 0, 1, 2, 3, 4, 5, 6, 7$ ) register.

The sequence conversion result register is 16 bits wide, the conversion result is right-aligned, and the valid value is stored in the low 12 bits (Bits 11:0) of the ADC<sub>x</sub>\_RESULT<sub>y</sub> register, and the high bits (Bits 15:12) are automatically filled with 0.



## 25.5 Operating modes

The ADC supports sequence channel conversion mode, which can convert up to 8 sequence channels (SQRCH0~SQRCH7) in turn, and the number of sequence channels to be converted is configured by the ENS bit field of the ADCx\_CR register. Each sequence channel SQRCHy (y = 0, 1, 2, 3, 4, 5, 6, 7) can select one of the 16 conversion sources, which is configured by the SQRCHy bit field of the sequence conversion channel configuration register ADCx\_SQRCFR, as shown in the following table:

Table 25-4 Sequence channel to be converted source

| ADCx_SQRCFR.SQRCHy | Source to be converted         | GPIO              |      |
|--------------------|--------------------------------|-------------------|------|
|                    |                                | ADC1              | ADC2 |
| 0000               | ADCx_IN0                       | PA00              | PA05 |
| 0001               | ADCx_IN1                       | PA01              | PA06 |
| 0010               | ADCx_IN2                       | PA02              | PA07 |
| 0011               | ADCx_IN3                       | PA03              | PB00 |
| 0100               | ADCx_IN4                       | PA04              | PB01 |
| 0101               | ADCx_IN5                       | PA05              | PA08 |
| 0110               | ADCx_IN6                       | PA06              | PA09 |
| 0111               | ADCx_IN7                       | PA07              | PA10 |
| 1000               | ADCx_IN8                       | PB00              | PA11 |
| 1001               | ADCx_IN9                       | PB01              | PA12 |
| 1010               | ADCx_IN10                      | PB10              |      |
| 1011               | ADCx_IN11                      | PB02              |      |
| 1100               | ADCx_IN12                      | Internal DAC_OUT2 |      |
| 1101               | ADCx_IN13                      | Internal DAC_OUT1 |      |
| 1110               | TS built-in temperature sensor | -                 |      |
| 1111               | 1.2V core voltage reference    | -                 |      |

ADC conversion is initiated by writing a 1 to the START bit field of the ADC start register ADCx\_START or by other peripherals, see [25.6 External trigger source](#).

The ADC supports single and continuous modes. In single mode (set ADCx\_CR.CONT to 0), the conversion configured in the ENS bit field is performed only once; in continuous mode (set ADCx\_CR.CONT to 1), the conversion configured in the ENS bit field is performed continuously.



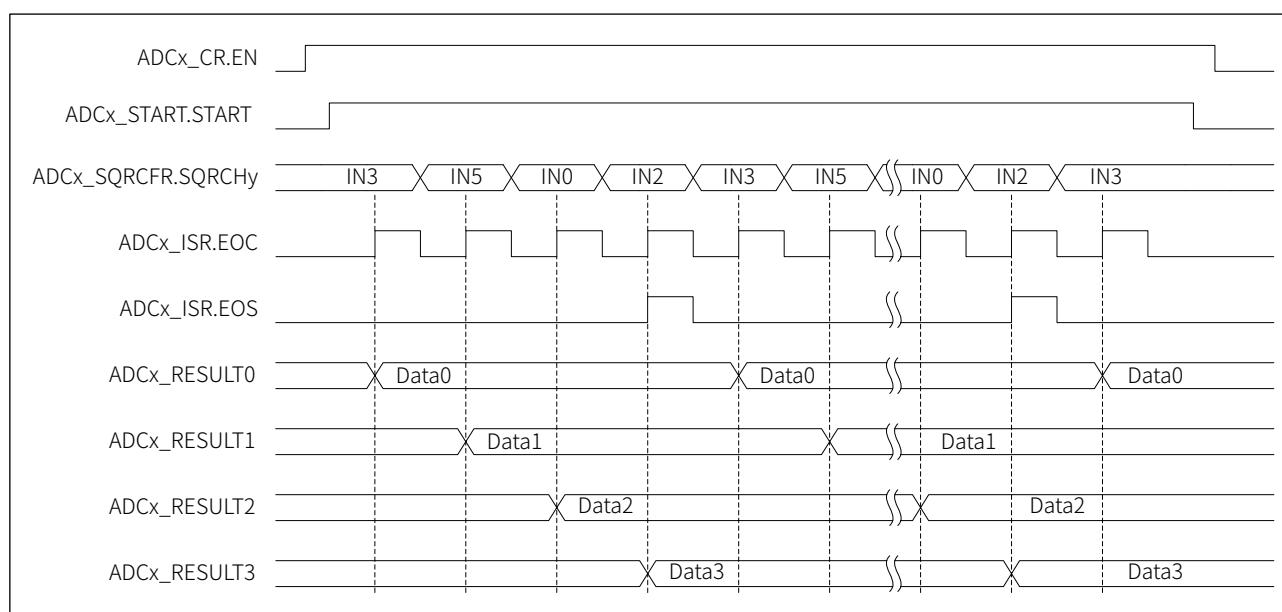
### 25.5.1 Sequence continuous conversion mode (CONT = 1)

Set ADCx\_CR.CONT to 1 to enable continuous conversion mode. In this mode, whether the ADC is enabled by the software START bit or by an external trigger, once the ADC is enabled, the sequence conversion channel configured in the ENS bit field will continue to convert until the ADCx\_START.START bit is cleared.

After each ADC conversion is completed, the ADCx\_ISR.EOC flag bit is automatically set to 1, and the conversion results are stored in the sequence conversion result registers ADCx\_RESULT0~ADCx\_RESULT7 of the same number as the sequence conversion channels SQRCH0 to SQRCH7. When all conversion of the selected sequence channel is completed, the sequence conversion completion flag bit ADCx\_ISR.EOS is set to 1. Users should read the conversion results in time to avoid overflow of conversion results. Clearing the ADCx\_START.START bit stops the conversion and resets the conversion channel of the sequence (i.e., starting the ADC again will restart the conversion from channel 0 of the sequence).

Taking the sequence conversion channels SQRCH0~SQRCH3 as an example, the timing of the sequence continuous conversion mode is shown in the following figure:

Figure 25-3 Sequence continuous conversion timing diagram



The ADC sequence continuous conversion is started by the START bit, and the reference operation flow is as follows:

- Step 1:** Set SYSCTRL\_AHBEN.GPIOx to 1, SYSCTRL\_APBEN1.ADC to 1, enable the GPIO clock corresponding to the ADC channel, and the ADC working clock.
- Step 2:** Set the GPIO pin corresponding to the ADC channel to the analog function. For the specific register configuration, please refer to chapter 9 General-purpose input/output (GPIO).
- Step 3:** Set ADCx\_CR.EN to 1 to enable ADC module.
- Step 4:** Delay about 1μs to wait for the completion of initialization of analog circuit.
- Step 5:** Set ADCx\_CR.CONT to 1, select continuous conversion mode
- Step 6:** Configure the corresponding bit field of the ADCx\_SAMPLE register and ADCx\_CR.CLK, set ADC sampling speed and clock selection.
- Step 7:** Configure ADCx\_CR.ENS, select the number of channels to-be-converted in sequence, as shown in the figure above, set ADCx\_CR.ENS to 3, and the sequence converted channels are SQRCH0~SQRCH3.

- Step 8:** Configure `ADCx_SQRCFR.SQRCH0`, select the sequence channel 0 to-be-converted source, as shown in the figure above, set `ADCx_SQRCFR.SQR CH0` to 3, and the channel 0 to-be-converted source is `ADCx_IN3`.
- Step 9:** Configure `ADCx_SQRCFR.SQRCH1`, select the sequence channel 1 to-be-converted source, as shown in the figure above, set `ADCx_SQRCFR.SQR CH1` to 5, and the channel 1 to-be-converted source is `ADCx_IN5`.
- Step 10:** Configure `ADCx_SQRCFR.SQRCH2`, select the sequence channel 2 to-be-converted source, as shown in the figure above, set `ADCx_SQRCFR.SQR CH2` to 0, and the channel 2 to-be-converted source is `ADCx_IN0`.
- Step 11:** Configure `ADCx_SQRCFR.SQRCH3`, select the sequence 1 channel 3 to-be-converted source, as shown in the figure above, set `ADCx_SQRCFR.SQR CH3` to 2, and the channel 3 to-be-converted source is `ADCx_IN2`.
- Step 12:** Set `ADCx_ICR` to 0, clear ADC interrupt flag.
- Step 13:** Set `ADCx_START.START` to 1 to start ADC conversion.
- Step 14:** Wait for `ADCx_ISR.EOS` to become 1, and read `ADCx_RESULT0~ADCx_RESULT3` registers in turn to obtain ADC conversion results for the corresponding channels. When `ADCx_ISR.EOS` becomes 1, it indicates that the sequence conversion of 4 channels at a time is completed.
- Step 15:** Set `ADCx_START.START` to 0 to stop ADC conversion.
- Step 16:** If you need to convert other channels, repeat steps 6 to 15.
- Step 17:** Set `ADCx_CR.EN` to 0 to turn off the ADC module.





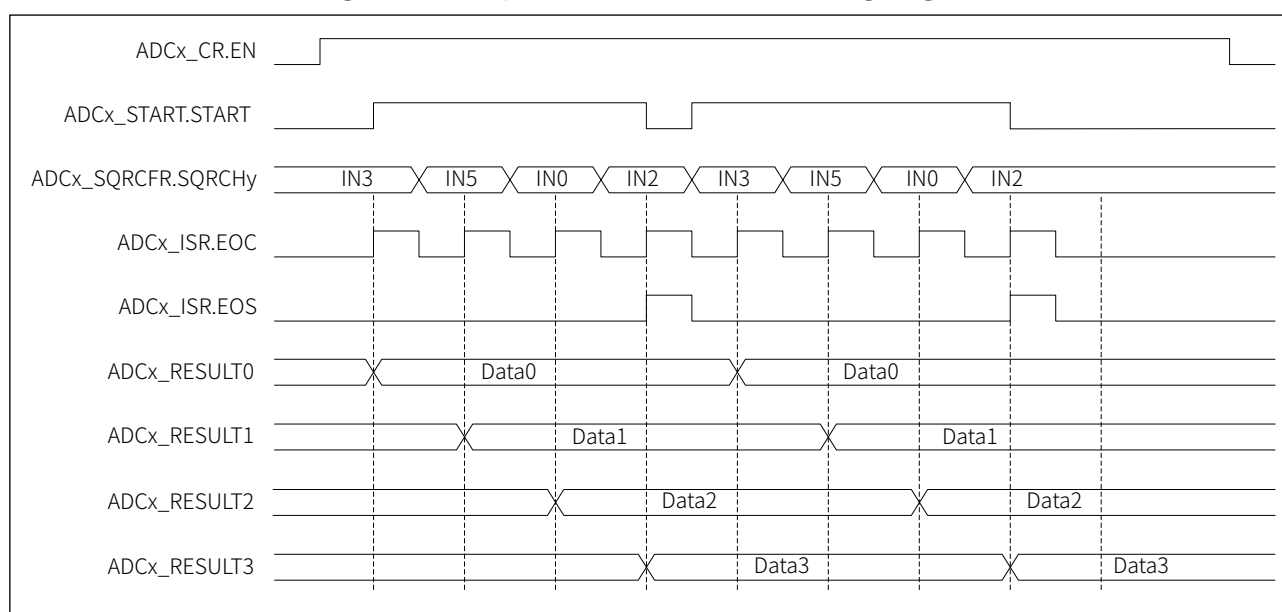
### 25.5.2 Sequence scan conversion mode (CONT = 0)

Set ADCx\_CR.CONT to 0 to enable single conversion mode. In this mode, whether the ADC is enabled by the software START bit or by an external trigger, enabling the ADC once will only convert all of the sequence conversion channels configured in the ENS bit field once.

After each ADC conversion is completed, the ADCx\_ISR.EOC flag bit is automatically set to 1, and the conversion results are stored in the sequence conversion result registers ADCx\_RESULT0~ADCx\_RESULT7 of the same number as the sequence conversion channels SQRCH0 to SQRCH7. When all the selected sequence channels are converted, the sequence conversion completion flag bit ADCx\_ISR.EOS is set to 1, the ADCx\_START.START bit is automatically cleared to 0, and the ADC stops conversion.

Taking the sequence conversion channels SQRCH0 ~ SQRCH3 as an example, the timing of the sequence scan conversion mode is shown in the following figure:

Figure 25-4 Sequence scan conversion timing diagram



The ADC sequence scan conversion is started by the START bit, and the reference operation flow is as follows:

- Step 1:** Set SYSCTRL\_AHBEN.GPIOx to 1, SYSCTRL\_APBEN1.ADC to 1, enable the GPIO clock corresponding to the ADC channel, and the ADC working clock.
- Step 2:** Set the GPIO pin corresponding to the ADC channel to the analog function. For the specific register configuration, please refer to chapter 9 [General-purpose input/output \(GPIO\)](#).
- Step 3:** Set ADCx\_CR.EN to 1 to enable ADC module.
- Step 4:** Delay about 1μs to wait for the completion of initialization of analog circuit.
- Step 5:** Set ADCx\_CR.CONT to 0, select single conversion mode.
- Step 6:** Configure the corresponding bit field of the ADCx\_SAMPLE register and ADCx\_CR.CLK, set ADC sampling speed and clock selection.
- Step 7:** Configure ADCx\_CR.ENS, select the number of channels to be converted in sequence, as shown in the figure above, set ADCx\_CR.ENS to 3, and the sequence converted channels are SQRCH0 ~ SQRCH3.
- Step 8:** Configure ADCx\_SQRCFR.SQRCH0, select the sequence channel 0 to-be-converted source, as shown in the figure above, set ADCx\_SQRCFR.SQR CH0 to 3, and the channel 0 to-be-converted source is ADCx\_IN3.

- Step 9:** Configure `ADCx_SQRCFR.SQRCH1`, select the sequence channel 1 to-be-converted source, as shown in the figure above, set `ADCx_SQRCFR.SQR CH1` to 5, and the channel 1 to-be-converted source is `ADCx_IN5`.
- Step 10:** Configure `ADCx_SQRCFR.SQRCH2`, select the sequence channel 2 to-be-converted source, as shown in the figure above, set `ADCx_SQRCFR.SQR CH2` to 0, and the channel 2 to-be-converted source is `ADCx_IN0`.
- Step 11:** Configure `ADCx_SQRCFR.SQRCH3`, select the sequence channel 3 to-be-converted source, as shown in the figure above, set `ADCx_SQRCFR.SQR CH3` to 2, and the channel 3 to-be-converted source is `ADCx_IN2`.
- Step 12:** Set `ADCx_ICR` to 0, clear ADC interrupt flag.
- Step 13:** Set `ADCx_START.START` to 1 to start ADC conversion.
- Step 14:** Wait for `ADCx_ISR.EOS` to become 1, and read `ADCx_RESULT0~ADCx_RESULT3` registers in turn to obtain ADC conversion results for the corresponding channels. When `ADCx_ISR.EOS` becomes 1, it indicates that the sequence conversion of 4 channels at a time is completed, the `ADCx_START.START` bit is automatically cleared to 0, and the ADC conversion stops.
- Step 15:** If you need to convert other channels, repeat steps 6 to 14.
- Step 16:** Set `ADCx_CR.EN` to 0 to turn off the ADC module.



## 25.6 External trigger source

ADC conversion can be started either by software (i.e., `ADCx_START.START` is set to 1) or by external trigger. The trigger source is selected by the external trigger start register `ADCx_TRIGGER`. There are 19 ways to trigger ADC. See the relevant register description for details.



## 25.7 Analog watchdog

The analog watchdog function is often used for automatic monitoring of analog quantities, i.e., comparing ADC conversion results with user-set thresholds and generating interrupts.

The analog watchdog protects multiple selected channels by programming the INy (y = 0~15) bit field of the ADCx\_AWDCR register, i.e., when the INy bit field is set to 1, the watchdog for the corresponding channel is enabled.

The analog watchdog supports upper and lower threshold comparisons, which are set via the VTH and VT bit fields of the ADCx\_AWDTR register, respectively. When the analog voltage converted by ADC is lower than the lower threshold or higher than the upper threshold, the corresponding flag bit of analog watchdog will be set to 1. If the corresponding bit field (AWDH, AWDL) of interrupt enable register ADCx\_IER is set, an interrupt request will be generated. The upper and lower thresholds are compared as follows:

- Upper threshold comparison: When the conversion result is within the interval (AWDTR.VTH, 4095], the ADCx\_ISR.AWDH flag bit is set to 1.
- Lower threshold comparison: When the conversion result is within the interval [0, AWDTR.VTL), the ADCx\_ISR.AWDL flag bit is set to 1.

The internal hardware signal ADCx\_AWD at the analog watchdog output can also be connected directly to the ETR inputs of GTIM and ATIM or as a clear source for the timer reference signal OCREF, see the corresponding timer section.



## 25.8 BGR module

The nominal output voltage of the built-in BGR module of this chip is 1.2V. The voltage output by BGR module of each chip is slightly different. The voltage output by the BGR module has been accurately measured at the factory and stored in FLASH. The voltage value can be read out by using the expression \*((uint16\_t\*)0x001007D2), and its unit is mV.

The internal resistance of the output voltage is relatively large, and the duration of the sampling process should be configured not less than 40μs when converting the voltage. By converting the voltage at the BGR output, the exact value of the current V<sub>DD</sub> supply voltage can be calculated using the formula shown below:

$$V_{DD} = V_{BGR} \times 4095 / \text{RESULT}$$

The specific operation steps are as follows:

- Step 1:** Set SYSCTRL\_APBEN1.ADC to 1 to enable ADC configuration clock and operating clock.
- Step 2:** Set ADC\_CR.EN to 1 to enable ADC module.
- Step 3:** Set CW\_BGR->CR.f.BGREN to 1 to enable BGR module.
- Step 4:** Configure ADC\_SQRCFR.SQRCH0 as 15, select the channel to be converted as BGR 1.2V.
- Step 5:** Configure ADC\_CR.CLK to 3, select ADCCLK source to PCLK/8.
- Step 6:** Configure ADC\_SAMPLE.SQRCH0 to 14 and configure sampling time to 262 ADCCLK clock cycles.  
*Note: Here, PCLK frequency is assumed to be 48MHz, ensuring that the duration of the sampling process is not less than 40μs.*
- Step 7:** Configure ADC\_CR.ENS to 0 to convert only one channel.
- Step 8:** Configure ADC\_CR.CONT to 0, single conversion mode.
- Step 9:** Set ADC\_START.START to 1 to start ADC conversion.
- Step 10:** Query and wait for ADC\_START.START to become 0, ADC conversion completed.
- Step 11:** Read out the conversion result from ADC\_RESULT0.
- Step 12:** Set ADC\_CR.EN to 0, turn off ADC module.
- Step 13:** Calculate the exact value of V<sub>DD</sub> in mV as shown below.

$$V_{DD} = (*(uint16_t*)0x001007D2) * 4095 / \text{RESULT0}$$



## 25.9 Temperature sensor

CW32L012 has a built-in temperature sensor module. The output voltage of the sensor changes with temperature. Set the sampling channel of the ADC module as an internal temperature sensor, and the current ambient temperature can be calculated through the ADC measurement results.

The temperature sensor is off by default. By setting the TSEN bit field of the control register BGR\_CR to 1, the temperature sensor is enabled, and enable for approximately 30μs before switching TS channels.

The formula for calculating the ambient temperature is as follows:

$$\text{Ambient temperature} = T_0 \times 0.5 + 0.095765 \times (V_{\text{ref}} \times \text{AdcValue} - 4.095 \times \text{Trim})$$

Where:

$V_{\text{ref}}$  is the reference voltage of the current ADC module, which takes the value of  $V_{\text{DDA}}$  in V.

$T_0$  is the 8-bit initial calibration temperature value, which is recorded in the FLASH memory of the chip. Its address is 0x0001 07CD, and the unit is 0.5 degrees Celsius. The read value needs to be divided by 2 to be the actual temperature.

AdcValue is the ADC conversion result of the ADC module measuring the output voltage of the temperature sensor, and the value range is 0 ~ 4095.

Trim is a 16-bit calibration value, which needs to be read from the FLASH memory of the chip during calculation, and its storage address is shown in the following table:

Table 25-5 ADC calibration value address

| ADC reference voltage | Calibration value storage address | Calibration value accuracy |
|-----------------------|-----------------------------------|----------------------------|
| $V_{\text{DDA}}$      | 0x0010 07CE - 0x0010 07CF         | $\pm 3^{\circ}\text{C}$    |

A calculation example is as follows:

Condition:  $V_{\text{ref}} = 3.3$ ,  $\text{AdcValue} = 0x3CD$ ,  $\text{Trim} = 0x311$ ,  $T_0 = 0x32$

Temperature:  $0x32 \times 0.5 + 0.095765 \times (3.3 \times 0x3CD - 4.095 \times 0x311) = 24.65^{\circ}\text{C}$

The reference operation flow for measuring the ambient temperature through the ADC is as follows:

- Step 1:** Set SYSCTRL\_APBEN1.ADC to 1, enable ADC configuration clock and working clock.
- Step 2:** Set ADCx\_CR.EN to 1 to enable the ADC module.
- Step 3:** Delay about 1μs to wait for the completion of initialization of analog circuit.
- Step 4:** Set ADCx\_CR.ENS to 0, convert SQRCH to 0.
- Step 5:** Set ADCx\_SQRCFR.SQRCH0 to 0xE, select sequence channel 0 to-be-converted source as built-in temperature sensor.
- Step 6:** Configure ADCx\_SAMPLE.SQRCH0 and ADCx\_CR.CLK to set ADC conversion speed.
- Step 7:** Set CW\_BGR->CR.f.TSEN to 1 to enable the temperature sensor and wait approximately 30μs for it to stabilize.
- Step 8:** Set ADCx\_ICR.EOC to 0, clear ADCx\_ISR.EOC flag.
- Step 9:** Set ADCx\_START.START to 1 to start ADC conversion.
- Step 10:** Wait for ADCx\_ISR.EOC to become 1, and read the ADCx\_RESULT0 register to obtain the ADC conversion result.
- Step 11:** Set ADCx\_CR.EN to 0, turn off ADC module.
- Step 12:** Read  $T_0$  and Trim, and calculate the current ambient temperature according to the formula.



## 25.10 ADC interrupts

ADC interrupt request, as shown in the following table:

Table 25-6 ADC interrupt sources and interrupt flags

| Interrupt source                   | Interrupt flag bit | Interrupt enable              | Flag clear                        |
|------------------------------------|--------------------|-------------------------------|-----------------------------------|
| Conversion result > ADCx_AWDTR.VTH | ADCx_ISR.AWDH      | ADCx_IER.AWDH<br>is set to 1. | ADCx_ICR.AWDH<br>is cleared to 0. |
| Conversion result < ADCx_AWDTR.VTL | ADCx_ISR.AWDL      | ADCx_IER.AWDL<br>is set to 1. | ADCx_ICR.AWDL<br>is cleared to 0. |
| ADC sequence conversion completed  | ADCx_ISR.EOS       | ADCx_IER.EOS<br>is set to 1.  | ADCx_ICR.EOS<br>is cleared to 0.  |
| ADC conversion completed           | ADCx_ISR.EOC       | ADCx_IER.EOC<br>is set to 1.  | ADCx_ICR.EOC<br>is cleared to 0.  |



## 25.11 List of registers

ADC1 base address: ADC1\_Base = 0x4000 0000

ADC2 base address: ADC2\_Base = 0x4000 0100

BGR base address: BGR\_BASE = 0x4000 0000

Table 25-7 List of ADC registers

| Register name | Register address | Register description                                    |
|---------------|------------------|---------------------------------------------------------|
| ADCx_CR       | ADCx_BASE + 0x00 | ADCx Control register                                   |
| ADCx_START    | ADCx_BASE + 0x04 | ADCx Startup register                                   |
| ADCx_SAMPLE   | ADCx_BASE + 0x18 | ADCx Sample cycle configuration register                |
| ADCx_SQRCFR   | ADCx_BASE + 0x20 | ADCx Sequence conversion channel configuration register |
| ADCx_AWDTR    | ADCx_BASE + 0x08 | ADCx Analogue watchdog threshold register               |
| ADCx_AWDCR    | ADCx_BASE + 0x0C | ADCx Analogue watchdog configuration register           |
| ADCx_TRIGGER  | ADCx_BASE + 0x10 | ADCx External trigger register                          |
| ADCx_IER      | ADCx_BASE + 0x70 | ADCx Interrupt enable register                          |
| ADCx_ISR      | ADCx_BASE + 0x78 | ADCx Interrupt flag register                            |
| ADCx_ICR      | ADCx_BASE + 0x74 | ADCx Interrupt flag clear register                      |
| ADCx_RESULT0  | ADCx_BASE + 0x30 | ADCx Sequence conversion result register 0              |
| ADCx_RESULT1  | ADCx_BASE + 0x34 | ADCx Sequence conversion result register 1              |
| ADCx_RESULT2  | ADCx_BASE + 0x38 | ADCx Sequence conversion result register 2              |
| ADCx_RESULT3  | ADCx_BASE + 0x3C | ADCx Sequence conversion result register 3              |
| ADCx_RESULT4  | ADCx_BASE + 0x40 | ADCx Sequence conversion result register 4              |
| ADCx_RESULT5  | ADCx_BASE + 0x44 | ADCx Sequence conversion result register 5              |
| ADCx_RESULT6  | ADCx_BASE + 0x48 | ADCx Sequence conversion result register 6              |
| ADCx_RESULT7  | ADCx_BASE + 0x4C | ADCx Sequence conversion result register 7              |
| BGR_CR        | BGR_BASE + 0xFC  | BGR Control register                                    |





## 25.12 Register description

See section 1 [Documentation conventions](#) for abbreviations used in register descriptions.

### 25.12.1 ADCx\_CR Control register

Address offset: 0x00      Reset value: 0x0000 0100

| Bit field | Name  | Permission | Function description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|-----------|-------|------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:8      | RFU   | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 7         | SLAVE | RW         | ADC synchronous start conversion enable control<br>0: Self start conversion.<br>1: Slave mode, can only be started synchronously by the master ADC start.<br><i>Note: When a certain ADC is set to 1, it will start synchronously with another ADC.</i>                                                                                                                                                                                                                                                                                                                                |
| 6:4       | ENS   | RW         | Number of channels to be converted in sequence 1 configuration<br>000: Convert SQRCH0, continuously convert 1 channel.<br>001: Convert SQRCH0–SQRCH1, continuously convert 2 channels.<br>010: Convert SQRCH0–SQRCH2, continuously convert 3 channels.<br>011: Convert SQRCH0–SQRCH3, continuously convert 4 channels.<br>100: Convert SQRCH0–SQRCH4, continuously convert 5 channels.<br>101: Convert SQRCH0–SQRCH5, continuously convert 6 channels.<br>110: Convert SQRCH0–SQRCH6, continuously convert 7 channels.<br>111: Convert SQRCH0–SQRCH7, continuously convert 8 channels. |
| 3:2       | CLK   | RW         | ADC working clock ADCCLK source configuration<br>00: ADCCLK = PCLK/1                      10: ADCCLK = PCLK/4<br>01: ADCCLK = PCLK/2                      11: ADCCLK = PCLK/8                                                                                                                                                                                                                                                                                                                                                                                                          |
| 1         | CONT  | RW         | Continuous conversion enable configuration<br>0: Single mode (performs the conversion configured by the ENS once).<br>1: Continuous mode (performs ENS-configured conversions continuously).                                                                                                                                                                                                                                                                                                                                                                                           |
| 0         | EN    | RW         | ADC enable control<br>0: Disable ADC.<br>1: Enable ADC.<br><i>Attention: External channels cannot be converted until approximately 1μs after enabled.</i>                                                                                                                                                                                                                                                                                                                                                                                                                              |



### 25.12.2 ADCx\_START Startup register

Address offset: 0x04      Reset value: 0x0000 0000

| Bit field | Name  | Permission | Function description                                                                                                                                                                       |
|-----------|-------|------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:1      | RFU   | -          | Reserved bits, please keep the default value.                                                                                                                                              |
| 0         | START | RW         | ADC start conversion control<br>0: Stop ADC conversion.<br>1: Start ADC conversion.<br><i>Attention: Software write 0 to stop conversion and reset the conversion channel of sequence.</i> |

### 25.12.3 ADCx\_SAMPLE Sample cycle configuration register

Address offset: 0x18      Reset value: 0x0000 0000

| Bit field | Name   | Permission | Function description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|-----------|--------|------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:28     | SQRCH7 | RW         | ADCx sequence channel 7 sample time configuration, see SQRCH0 bit field.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 27:24     | SQRCH6 | RW         | ADCx sequence channel 6 sample time configuration, see SQRCH0 bit field.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 23:20     | SQRCH5 | RW         | ADCx sequence channel 5 sample time configuration, see SQRCH0 bit field.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 19:16     | SQRCH4 | RW         | ADCx sequence channel 4 sample time configuration, see SQRCH0 bit field.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 15:12     | SQRCH3 | RW         | ADCx sequence channel 3 sample time configuration, see SQRCH0 bit field.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 11:8      | SQRCH2 | RW         | ADCx sequence channel 2 sample time configuration, see SQRCH0 bit field.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 7:4       | SQRCH1 | RW         | ADCx sequence channel 1 sample time configuration, see SQRCH0 bit field.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 3:0       | SQRCH0 | RW         | ADCx sequence channel 0 sample time configuration:<br>0000: 6 ADCCLK clock cycles      1000: 54 ADCCLK clock cycles<br>0001: 7 ADCCLK clock cycles      1001: 70 ADCCLK clock cycles<br>0010: 9 ADCCLK clock cycles      1010: 102 ADCCLK clock cycles<br>0011: 12 ADCCLK clock cycles      1011: 134 ADCCLK clock cycles<br>0100: 18 ADCCLK clock cycles      1100: 166 ADCCLK clock cycles<br>0101: 24 ADCCLK clock cycles      1101: 198 ADCCLK clock cycles<br>0110: 30 ADCCLK clock cycles      1110: 262 ADCCLK clock cycles<br>0111: 42 ADCCLK clock cycles      1111: 518 ADCCLK<br><i>Note: When the converted channel is BGR or TS, the sampling duration needs at least 40μs.</i> |

### 25.12.4 ADCx\_SQRCFR Sequence conversion channel configuration register

Address offset: 0x20      Reset value: 0x0000 0000

| Bit field | Name   | Permission | Function description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|-----------|--------|------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:28     | SQRCH7 | RW         | Sequence channel 7 to-be-converted source configuration, see SQRCH0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 27:24     | SQRCH6 | RW         | Sequence channel 6 to-be-converted source configuration, see SQRCH0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 23:20     | SQRCH5 | RW         | Sequence channel 5 to-be-converted source configuration, see SQRCH0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 19:16     | SQRCH4 | RW         | Sequence channel 4 to-be-converted source configuration, see SQRCH0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 15:12     | SQRCH3 | RW         | Sequence channel 3 to-be-converted source configuration, see SQRCH0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 11:8      | SQRCH2 | RW         | Sequence channel 2 to-be-converted source configuration, see SQRCH0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 7:4       | SQRCH1 | RW         | Sequence channel 1 to-be-converted source configuration, see SQRCH0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 3:0       | SQRCH0 | RW         | Sequence channel 0 to-be-converted source configuration:<br>0000: ADC_IN0                      1000: ADC_IN8<br>0001: ADC_IN1                      1001: ADC_IN9<br>0010: ADC_IN2                      1010: ADC_IN10<br>0011: ADC_IN3                      1011: ADC_IN11<br>0100: ADC_IN4                      1100: DAC_OUT2 internal signal<br>0101: ADC_IN5                      1101: DAC_OUT1 internal signal<br>0110: ADC_IN6                      1110: Built-in temperature sensor<br>0111: ADC_IN7                      1111: 1.2V core voltage reference |

### 25.12.5 ADCx\_AWDTR Analogue watchdog threshold register

Address offset: 0x08      Reset value: 0x0FFF 0000

| Bit field | Name | Permission | Function description                          |
|-----------|------|------------|-----------------------------------------------|
| 31:28     | RFU  | -          | Reserved bits, please keep the default value. |
| 27:16     | VTH  | RW         | Analogue watchdog threshold high limit value. |
| 15:12     | RFU  | -          | Reserved bits, please keep the default value. |
| 11:0      | VTL  | RW         | Analogue watchdog threshold low limit value.  |

### 25.12.6 ADCx\_AWDCR Analogue watchdog configuration register

Address offset: 0x0C      Reset value: 0x0000 0000

| Bit field | Name | Permission | Function description                                                                                                               |
|-----------|------|------------|------------------------------------------------------------------------------------------------------------------------------------|
| 31:16     | RFU  | -          | Reserved bits, please keep the default value.                                                                                      |
| 15        | IN15 | RW         | Analog watchdog monitoring ADCx_IN15 enable control (i.e., 1.2V core voltage reference)<br>0: Disabled<br>1: Enabled               |
| 14        | IN14 | RW         | Analog watchdog monitoring ADCx_IN14 enable control (i.e., built-in temperature sensor output signal)<br>0: Disabled<br>1: Enabled |
| 13        | IN13 | RW         | Analog watchdog monitoring ADCx_IN13 enable control (i.e., internal signal DAC_OUT1)<br>0: Disabled<br>1: Enabled                  |
| 12        | IN12 | RW         | Analog watchdog monitoring ADCx_IN12 enable control (i.e., internal signal DAC_OUT2)<br>0: Disabled<br>1: Enabled                  |
| 11        | IN11 | RW         | Analog watchdog monitoring ADCx_IN11 enable control<br>0: Disabled<br>1: Enabled                                                   |
| 10        | IN10 | RW         | Analog watchdog monitoring ADCx_IN10 enable control<br>0: Disabled<br>1: Enabled                                                   |
| 9         | IN9  | RW         | Analog watchdog monitoring ADCx_IN9 enable control<br>0: Disabled<br>1: Enabled                                                    |
| 8         | IN8  | RW         | Analog watchdog monitoring ADCx_IN8 enable control<br>0: Disabled<br>1: Enabled                                                    |
| 7         | IN7  | RW         | Analog watchdog monitoring ADCx_IN7 enable control<br>0: Disabled<br>1: Enabled                                                    |
| 6         | IN6  | RW         | Analog watchdog monitoring ADCx_IN6 enable control<br>0: Disabled<br>1: Enabled                                                    |
| 5         | IN5  | RW         | Analog watchdog monitoring ADCx_IN5 enable control<br>0: Disabled<br>1: Enabled                                                    |



| Bit field | Name | Permission | Function description                                                            |
|-----------|------|------------|---------------------------------------------------------------------------------|
| 4         | IN4  | RW         | Analog watchdog monitoring ADCx_IN4 enable control<br>0: Disabled<br>1: Enabled |
| 3         | IN3  | RW         | Analog watchdog monitoring ADCx_IN3 enable control<br>0: Disabled<br>1: Enabled |
| 2         | IN2  | RW         | Analog watchdog monitoring ADCx_IN2 enable control<br>0: Disabled<br>1: Enabled |
| 1         | IN1  | RW         | Analog watchdog monitoring ADCx_IN1 enable control<br>0: Disabled<br>1: Enabled |
| 0         | IN0  | RW         | Analog watchdog monitoring ADCx_IN0 enable control<br>0: Disabled<br>1: Enabled |



### 25.12.7 ADCx\_TRIGGER External trigger register

Address offset: 0x10      Reset value: 0x0000 0000

| Bit field | Name         | Permission | Function description                                                               |
|-----------|--------------|------------|------------------------------------------------------------------------------------|
| 31        | HALLTIMTRGO  | RW         | HALLTIMTRGO rising edge signal triggers ADC startup<br>0: Disabled<br>1: Enabled   |
| 30        | BTIM3TRGO    | RW         | BTIM3 TRGO triggers ADC startup<br>0: Disabled<br>1: Enabled                       |
| 29        | BTIM2TRGO    | RW         | BTIM2 TRGO triggers ADC startup<br>0: Disabled<br>1: Enabled                       |
| 28        | BTIM1TRGO    | RW         | BTIM1 TRGO triggers ADC startup<br>0: Disabled<br>1: Enabled                       |
| 27        | GTIM4OC4REFC | RW         | GTIM4 OC4REFC rising edge signal triggers ADC startup<br>0: Disabled<br>1: Enabled |
| 26        | GTIM4OC3REFC | RW         | GTIM4 OC3REFC rising edge signal triggers ADC startup<br>0: Disabled<br>1: Enabled |
| 25        | GTIM4OC2REFC | RW         | GTIM4 OC2REFC rising edge signal triggers ADC startup<br>0: Disabled<br>1: Enabled |
| 24        | GTIM4OC1REFC | RW         | GTIM4 OC1REFC rising edge signal triggers ADC startup<br>0: Disabled<br>1: Enabled |
| 23        | GTIM4TRGO    | RW         | GTIM4 TRGO triggers ADC startup<br>0: Disabled<br>1: Enabled                       |
| 22        | GTIM3OC4REFC | RW         | GTIM3 OC4REFC rising edge signal triggers ADC startup<br>0: Disabled<br>1: Enabled |
| 21        | GTIM3OC3REFC | RW         | GTIM3 OC3REFC rising edge signal triggers ADC startup<br>0: Disabled<br>1: Enabled |
| 20        | GTIM3OC2REFC | RW         | GTIM3 OC2REFC rising edge signal triggers ADC startup<br>0: Disabled<br>1: Enabled |
| 19        | GTIM3OC1REFC | RW         | GTIM3 OC1REFC rising edge signal triggers ADC startup<br>0: Disabled<br>1: Enabled |



| Bit field | Name         | Permission | Function description                                                               |
|-----------|--------------|------------|------------------------------------------------------------------------------------|
| 18        | GTIM3TRGO    | RW         | GTIM3 TRGO triggers ADC startup<br>0: Disabled<br>1: Enabled                       |
| 17        | GTIM2OC4REFC | RW         | GTIM2 OC4REFC rising edge signal triggers ADC startup<br>0: Disabled<br>1: Enabled |
| 16        | GTIM2OC3REFC | RW         | GTIM2 OC3REFC rising edge signal triggers ADC startup<br>0: Disabled<br>1: Enabled |
| 15        | GTIM2OC2REFC | RW         | GTIM2 OC2REFC rising edge signal triggers ADC startup<br>0: Disabled<br>1: Enabled |
| 14        | GTIM2OC1REFC | RW         | GTIM2 OC1REFC rising edge signal triggers ADC startup<br>0: Disabled<br>1: Enabled |
| 13        | GTIM2TRGO    | RW         | GTIM2 TRGO triggers ADC startup<br>0: Disabled<br>1: Enabled                       |
| 12        | GTIM1OC4REFC | RW         | GTIM1 OC4REFC rising edge signal triggers ADC startup<br>0: Disabled<br>1: Enabled |
| 11        | GTIM1OC3REFC | RW         | GTIM1 OC3REFC rising edge signal triggers ADC startup<br>0: Disabled<br>1: Enabled |
| 10        | GTIM1OC2REFC | RW         | GTIM1 OC2REFC rising edge signal triggers ADC startup<br>0: Disabled<br>1: Enabled |
| 9         | GTIM1OC1REFC | RW         | GTIM1 OC1REFC rising edge signal triggers ADC startup<br>0: Disabled<br>1: Enabled |
| 8         | GTIM1TRGO    | RW         | GTIM1 TRGO triggers ADC startup<br>0: Disabled<br>1: Enabled                       |
| 7         | ATIMOC6REFC  | RW         | ATIM OC6REFC rising edge signal triggers ADC startup<br>0: Disabled<br>1: Enabled  |
| 6         | ATIMOC5REFC  | RW         | ATIM OC5REFC rising edge signal triggers ADC startup<br>0: Disabled<br>1: Enabled  |



| Bit field | Name        | Permission | Function description                                                              |
|-----------|-------------|------------|-----------------------------------------------------------------------------------|
| 5         | ATIMOC4REFC | RW         | ATIM OC4REFC rising edge signal triggers ADC startup<br>0: Disabled<br>1: Enabled |
| 4         | ATIMOC3REFC | RW         | ATIM OC3REFC rising edge signal triggers ADC startup<br>0: Disabled<br>1: Enabled |
| 3         | ATIMOC2REFC | RW         | ATIM OC2REFC rising edge signal triggers ADC startup<br>0: Disabled<br>1: Enabled |
| 2         | ATIMOC1REFC | RW         | ATIM OC1REFC rising edge signal triggers ADC startup<br>0: Disabled<br>1: Enabled |
| 1         | ATIMTRGO2   | RW         | ATIM TRGO2 triggers ADC startup<br>0: Disabled<br>1: Enabled                      |
| 0         | ATIMTRGO    | RW         | ATIM TRGO triggers ADC startup<br>0: Disabled<br>1: Enabled                       |





### 25.12.8 ADCx\_IER Interrupt enable register

Address offset: 0x70

Reset value: 0x0000 0000

| Bit field | Name   | Permission | Function description                                                                            |
|-----------|--------|------------|-------------------------------------------------------------------------------------------------|
| 31:7      | RFU    | -          | Reserved bits, please keep the default value.                                                   |
| 6         | DMAEOS | RW         | Sequence 1 conversion completion DMA trigger signal enable control<br>0: Disabled<br>1: Enabled |
| 5         | DMAEOC | RW         | Each conversion completion DMA trigger signal enable control<br>0: Disabled<br>1: Enabled       |
| 4         | AWDH   | RW         | Analogue watchdog threshold high limit interrupt enable control<br>0: Disabled<br>1: Enabled    |
| 3         | AWDL   | RW         | Analogue watchdog threshold low limit interrupt enable control<br>0: Disabled<br>1: Enabled     |
| 2         | RFU    | -          | Reserved bits, please keep the default value.                                                   |
| 1         | EOS    | RW         | Sequence conversion completion interrupt enable control<br>0: Disabled<br>1: Enabled            |
| 0         | EOC    | RW         | Single conversion completion interrupt enable control<br>0: Disabled<br>1: Enabled              |



### 25.12.9 ADCx\_ISR Interrupt flag register

Address offset: 0x78      Reset value: 0x0000 0000

| Bit field | Name | Permission | Function description                                                                                                                                                       |
|-----------|------|------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:5      | RFU  | -          | Reserved bits, please keep the default value.                                                                                                                              |
| 4         | AWDH | RO         | Analog watchdog threshold high limit flag<br>0: The conversion result is in the interval [0, AWDTR.VTH].<br>1: The conversion result is in the interval (AWDTR.VTH, 4095]. |
| 3         | AWDL | RO         | Analog watchdog threshold low limit flag<br>0: The conversion result is in the interval [AWDTR.VTL, 4095].<br>1: The conversion result is in the interval [0, AWDTR.VTL).  |
| 2         | RFU  | -          | Reserved bits, please keep the default value.                                                                                                                              |
| 1         | EOS  | RO         | Sequence conversion completion flag<br>0: Sequence conversion not completed.<br>1: Sequence conversion completed.                                                          |
| 0         | EOC  | RO         | Conversion completion flag<br>0: One ADC conversion not completed.<br>1: One ADC conversion completed.                                                                     |

### 25.12.10 ADCx\_ICR Interrupt flag clear register

Address offset: 0x74      Reset value: 0x0000 000F

| Bit field | Name | Permission | Function description                                                                                                                     |
|-----------|------|------------|------------------------------------------------------------------------------------------------------------------------------------------|
| 31:5      | RFU  | -          | Reserved bits, please keep the default value.                                                                                            |
| 4         | AWDH | R1W0       | Analogue watchdog threshold high limit flag clear 0 control<br>W0: Clear the corresponding flag in the ISR register.<br>W1: No function. |
| 3         | AWDL | R1W0       | Analogue watchdog threshold low limit flag clear 0 control<br>W0: Clear the corresponding flags in the ISR register.<br>W1: No function. |
| 2         | RFU  | -          | Reserved bits, please keep the default value.                                                                                            |
| 1         | EOS  | R1W0       | Sequence conversion completion flag clear 0 control<br>W0: Clear the corresponding flags in the ISR register.<br>W1: No function.        |
| 0         | EOC  | R1W0       | Conversion completion flag clear 0 control<br>W0: Clear the corresponding flags in the ISR register.<br>W1: No function.                 |



**25.12.11 ADCx\_RESULT0 Sequence conversion result register 0**

Address offset: 0x30      Reset value: 0x0000 0000

| Bit field | Name   | Permission | Function description                          |
|-----------|--------|------------|-----------------------------------------------|
| 31:12     | RFU    | -          | Reserved bits, please keep the default value. |
| 11:0      | RESULT | RO         | ADC sequence conversion result 0 .            |

**25.12.12 ADCx\_RESULT1 Sequence conversion result register 1**

Address offset: 0x34      Reset value: 0x0000 0000

| Bit field | Name   | Permission | Function description                          |
|-----------|--------|------------|-----------------------------------------------|
| 31:12     | RFU    | -          | Reserved bits, please keep the default value. |
| 11:0      | RESULT | RO         | ADC sequence conversion result 1.             |

**25.12.13 ADCx\_RESULT2 Sequence conversion result register 2**

Address offset: 0x38      Reset value: 0x0000 0000

| Bit field | Name   | Permission | Function description                          |
|-----------|--------|------------|-----------------------------------------------|
| 31:12     | RFU    | -          | Reserved bits, please keep the default value. |
| 11:0      | RESULT | RO         | ADC sequence conversion result 2.             |

**25.12.14 ADCx\_RESULT3 Sequence conversion result register 3**

Address offset: 0x3C      Reset value: 0x0000 0000

| Bit field | Name   | Permission | Function description                          |
|-----------|--------|------------|-----------------------------------------------|
| 31:12     | RFU    | -          | Reserved bits, please keep the default value. |
| 11:0      | RESULT | RO         | ADC sequence conversion result 3.             |

**25.12.15 ADCx\_RESULT4 Sequence conversion result register 4**

Address offset: 0x40      Reset value: 0x0000 0000

| Bit field | Name   | Permission | Function description                          |
|-----------|--------|------------|-----------------------------------------------|
| 31:12     | RFU    | -          | Reserved bits, please keep the default value. |
| 11:0      | RESULT | RO         | ADC sequence conversion result 4.             |



### 25.12.16 ADCx\_RESULT5 Sequence conversion result register 5

Address offset: 0x44      Reset value: 0x0000 0000

| Bit field | Name   | Permission | Function description                          |
|-----------|--------|------------|-----------------------------------------------|
| 31:12     | RFU    | -          | Reserved bits, please keep the default value. |
| 11:0      | RESULT | RO         | ADC sequence conversion result 5.             |

### 25.12.17 ADCx\_RESULT6 Sequence conversion result register 6

Address offset: 0x48      Reset value: 0x0000 0000

| Bit field | Name   | Permission | Function description                          |
|-----------|--------|------------|-----------------------------------------------|
| 31:12     | RFU    | -          | Reserved bits, please keep the default value. |
| 11:0      | RESULT | RO         | ADC sequence conversion result 6.             |

### 25.12.18 ADCx\_RESULT7 Sequence conversion result register 7

Address offset: 0x4C      Reset value: 0x0000 0000

| Bit field | Name   | Permission | Function description                          |
|-----------|--------|------------|-----------------------------------------------|
| 31:12     | RFU    | -          | Reserved bits, please keep the default value. |
| 11:0      | RESULT | RO         | ADC sequence conversion result 7.             |

### 25.12.19 BGR\_CR Control register

Address offset: 0xFC      Reset value: 0x0000 0000

| Bit field | Name  | Permission | Function description                                                                                                                                                                                                                                                                                                                                                                                    |
|-----------|-------|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:2      | RFU   | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                                                                                           |
| 1         | TSEN  | RW         | Built-in temperature sensor enable control<br>0: Disable built-in temperature sensor.<br>1: Enable built-in temperature sensor.<br><i>Attention: Enable approximately 30μs before switching TS channels.</i>                                                                                                                                                                                            |
| 0         | BGREN | RW         | Built-in BGR reference enable control<br>0: Disable built-in BGR reference.<br>1: Enable built-in BGR reference.<br><i>Attention: Enable approximately 30μs before using BGR (including hardware startup and software startup).</i><br><i>Note: Enabling BGREN, ADCEN, TSEN, VC1/2/3/4, and OPA1/2 will automatically start BGR.</i><br><i>Note: BGR can only be disabled by software or POR reset.</i> |



## 26 Digital to analog converter (DAC)

### 26.1 Overview

CW32L012 integrates a 2-channel digital-to-analog converter (DAC) internally, offering 8-bit or 12-bit resolution, and is equipped with a built-in noise generator and a triangular wave generator. The digital signals of the Data Holding Register (DHR), the built-in noise generator and the triangular wave generator can be converted into analog signals. The conversion results can be output through 2 independent output channels and support both single-channel and dual-channel modes. The conversion results can also provide data for ADC, VC and OPA. Meanwhile, the DAC supports DMA transmission function, allowing for high-speed data transmit between DHR registers and memory without CPU intervention.



## 26.2 Main features

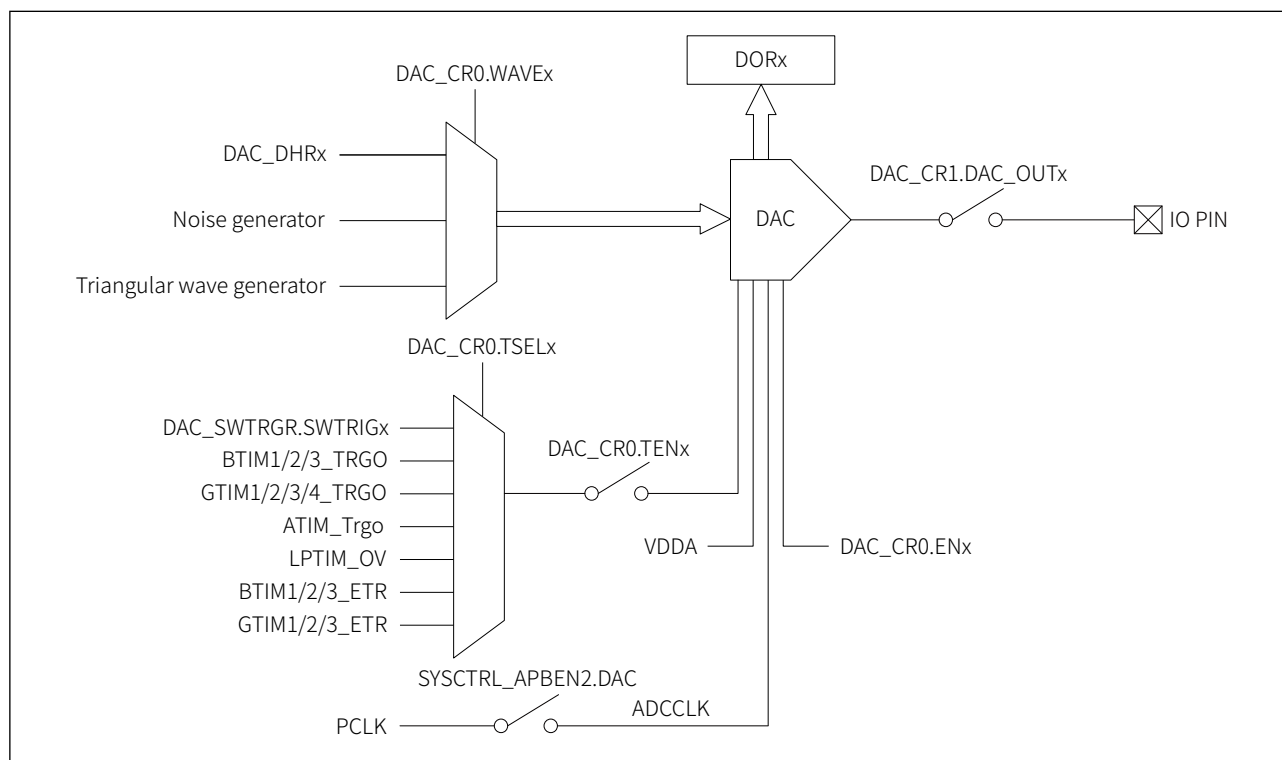
- 8-bit or 12-bit resolution
- Noise generator
- Triangular wave generator
- 3 input conversion channels
  - Data Holding Register (DHR)
  - Built-in noise generator
  - Triangular wave generator
- 2 independent output channels
- Supports single-channel left-aligned/right-aligned data formats and dual-channel mode in 12-bit mode
- Supports single-channel left-aligned/right-aligned data formats and dual-channel mode in 8-bit mode
- The VDDA power supply voltage serves as both the power supply and reference voltage source (Vref)
- Supports on-chip peripheral automatic triggering of DAC conversion
- Supports direct memory access (DMA)



## 26.3 Functional block diagram

The functional block diagram of DAC is shown in the following figure:

Figure 26-1 DAC functional block diagram



## 26.4 Digital signal conversion source

The data source of the DAC can be the Data Holding Register (DHR), the internal noise generator, or the triangular wave generator. Users can flexibly select the data source through the WAVEx bit field of the CR0 register.

DHR is used to store digital data that is to be converted into analog signals. During the DAC's operation, it serves as a temporary data storage area, waiting for the appropriate moment to output the data to the DAC's data output register (DOR), thereby achieving the conversion from digital to analog signals. The DAC offers multiple DHR formats for user to choose from to accommodate different resolutions and data alignment methods, such as single-channel 12-bit right-aligned, single-channel 12-bit left-aligned, single-channel 8-bit right-aligned, as well as dual-channel 12-bit right-aligned, dual-channel 12-bit left-aligned, and dual-channel 8-bit right-aligned data formats.

When using DHR as the data source, the WAVEx bit field of the CR0 register should be configured to 00 to disable the internal waveform generator. The value of DHR is updated to DOR, and the DAC outputs the corresponding level.

When using the internal noise generator as the data source, the WAVEx bit field of the CR0 register should be configured to 01. At this time, the DOR data for the DAC is generated by the linear feedback shift register (LSFR) of the internal noise generator, producing a pseudo-random sequence to simulate noise generation. The noise amplitude is determined by the MAMP bit field of the CR0 register.

When using the internal triangular wave generator as the data source, the WAVEx bit field of the CR0 register should be configured to 10 or 11. Similarly, the DOR data for the DAC is generated by the internal triangular wave generator at this time. The amplitude of the triangular wave is determined by the MAMP bit field of the CR0 register.

**Attention:** When switching the DAC's data source, the DOR register does not automatically clear its current value. The user must manually write a 0 to the DHR to update the DOR register's value.





## 26.5 Data format

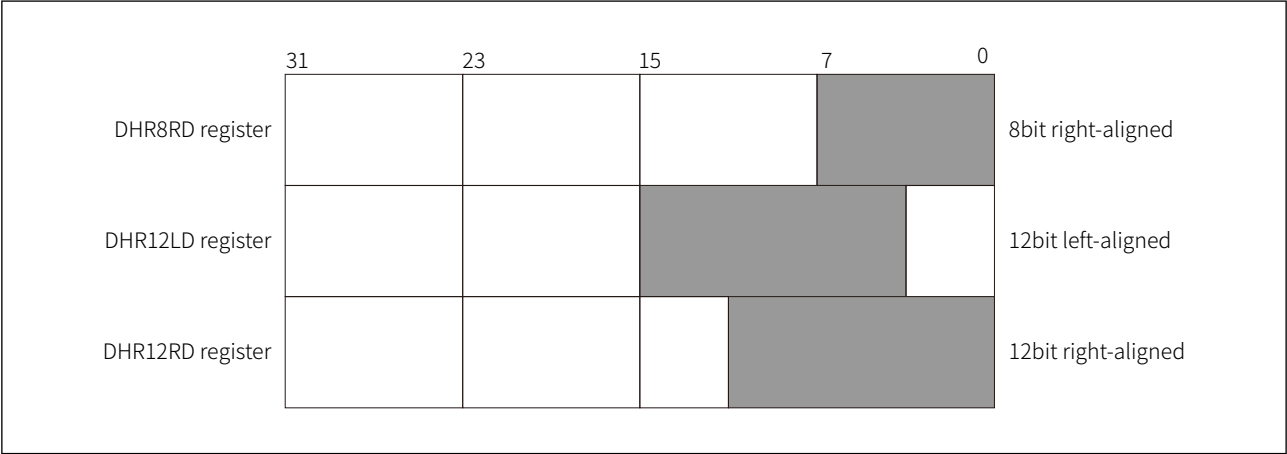
To accommodate different resolutions and data alignment methods, DAC offers multiple DHR formats for users to choose from.

### 26.5.1 Single-channel mode

In the DAC single-channel mode, users can choose the 12-bit right-aligned, 12-bit left-aligned or 8-bit right-aligned format for a single channel. Fill the data into the single-channel 12-bit right-aligned data hold register DHR12Rx[11:0], the single-channel 12-bit left-aligned data hold register DHR12L[15:4], or the channel 8-bit right-aligned data hold register DHR8Rx[7:0] respectively. Then, the DHRx register will output the data to the data output register (DOR) of the DAC at an appropriate time.

According to the selected configuration mode, the data must be written to the specified register as follows:

Figure 26-2 Single-channel register configuration format

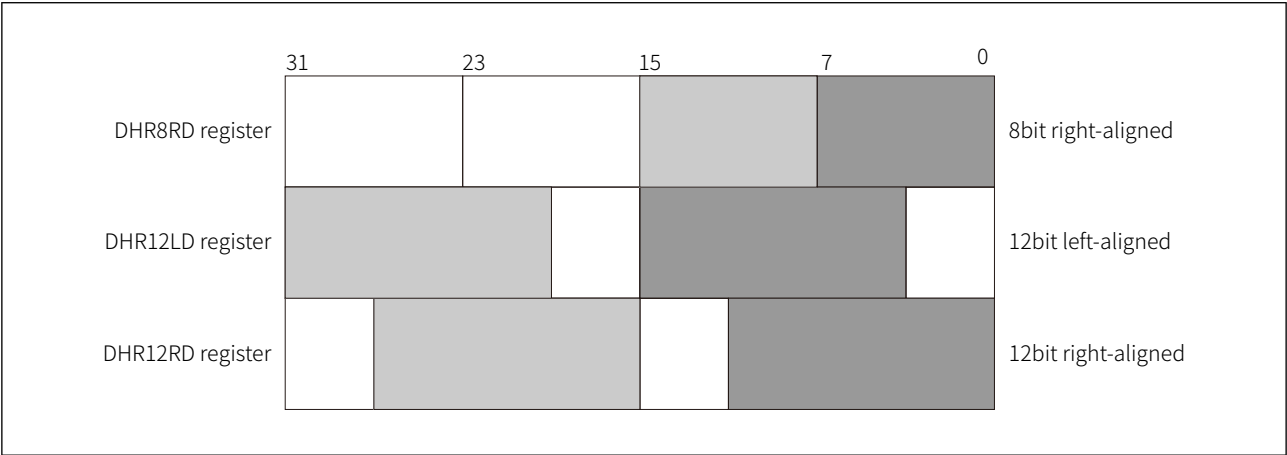


26.5.2 Dual-channel mode

To optimize data transfer efficiency, users can assign values to both channels simultaneously. In dual-channel mode, users may select dual-channel 12-bit right-aligned, dual-channel 12-bit left-aligned, or dual-channel 8-bit right-aligned modes. Fill the data into the dual-channel 12-bit right-aligned data hold register DHR12RD, dual-channel 12-bit left-aligned data hold register DHR12LD, or dual-channel 8-bit right-aligned data hold register DHR8RD respectively.

According to the selected configuration mode, the data must be written to the specified register as follows:

Figure 26-3 Dual-channel register configuration format



## 26.6 Trigger source

There are 2 ways to load the data in the DOR register of the DAC: direct loading and trigger loading, which can be achieved by setting the TENx bit field of the configuration register CR0. When the TENx bit of the CR0 register is set to 0, the data written to the DHR register will be directly loaded into the DOR register, and the output level of the DAC will change immediately. When the TENx bit of the CR0 register is set to 1, the data written to the DHR register will be saved in the DHR, and the output level of the DAC is determined by the value of the previous DOR register until the DAC receives the trigger signal generated by the trigger source selected by the TSELx bit of the CR0 register, the value of the DOR register has just been updated.

The DAC supports 16 trigger sources, which are set by configuring the TSELx bit field of the CR0 configuration register. The trigger signal sources are shown in the following table:

Table 26-1 DAC trigger source

| TSELx bit field | Trigger source   |
|-----------------|------------------|
| 0000            | Software trigger |
| 0001            | BTIM1_Trgo       |
| 0010            | BTIM2_Trgo       |
| 0011            | BTIM3_Trgo       |
| 0100            | GTIM1_Trgo       |
| 0101            | GTIM2_Trgo       |
| 0110            | GTIM3_Trgo       |
| 0111            | GTIM4_Trgo       |
| 1000            | ATIM_Trgo        |
| 1001            | LPTIM_OV         |
| 1010            | BTIM1_ETR        |
| 1011            | BTIM2_ETR        |
| 1100            | BTIM3_ETR        |
| 1101            | GTIM1_ETR        |
| 1110            | GTIM2_ETR        |
| 1111            | GTIM3_ETR        |



## 26.7 Direct memory access (DMA)

The DAC allows the value of the DHR register to be updated through the DMA controller. Set the DMAENx of the CR0 register to 1, enable the DMA mode of the DAC, set the DMA\_TRIGy register to DMA for hardware trigger, and the hardware trigger source of DMA is the DMA overflow event of the corresponding DAC channel. After the DAC has completed the data loading of the DOR register, it will automatically move the next data to be loaded to the DHR through DMA. If the rate of trigger loading exceeds the data transmission speed of DMA, that is, during the period when DMA loads data into DHR, a new trigger loading signal is generated by the DAC, then the DAC will generate an overflow event under DMA and produce the corresponding overflow flag in the DORx register. If the DMAUDRIEx bit field of the CR0 register is set to 1, an interrupt request will be generated.



## 26.8 DAC interrupt

The DAC supports 2 interrupt sources. When a DAC interrupt event occurs, the interrupt flag bit will be set by the hardware. If the corresponding interrupt enable control bit is set, an interrupt request will be generated.

Within the user DAC interrupt service routine, the relevant DAC interrupt flag bits should be checked to perform the corresponding handling. Before exiting the interrupt service routine, the interrupt flag bit must be cleared to prevent repeated entry into the interrupt service routine.

The flag bits, interrupt enable bits, interrupt flag clear bits, or clearing methods for each DAC interrupt source are shown in the following table:

Table 26-2 DAC interrupt control

| Interrupt event                               | Interrupt flag bit | Interrupt the enable bit | Interrupt flag clear    |
|-----------------------------------------------|--------------------|--------------------------|-------------------------|
| DMA underflow error occurred in DAC channel 1 | DOR1.DMAUDR1       | CR0.DMAUDRIE1            | Write 0 to DOR1.DMAUDR1 |
| DMA underflow error occurred in DAC channel 2 | DOR2.DMAUDR2       | CR0.DMAUDRIE2            | Write 0 to DOR2.DMAUDR2 |



## 26.9 List of registers

DACx base address: DACx\_Base = 0x4000 00C0

Table 26-3 List of DAC registers

| Register name | Register address | Register description                                 |
|---------------|------------------|------------------------------------------------------|
| DACx_CR0      | DACx_Base + 0x00 | Control register 0                                   |
| DACx_SWTRGR   | DACx_Base + 0x04 | Software trigger register                            |
| DACx_DHR12R1  | DACx_Base + 0x08 | Channel 1 12-bit right-aligned data hold register    |
| DACx_DHR12L1  | DACx_Base + 0x0C | Channel 1 12-bit left-aligned data hold register     |
| DACx_DHR8R1   | DACx_Base + 0x10 | Channel 1 8-bit right-aligned data hold register     |
| DACx_DHR12R2  | DACx_Base + 0x14 | Channel 2 12-bit right-aligned data hold register    |
| DACx_DHR12L2  | DACx_Base + 0x18 | Channel 2 12-bit left-aligned data hold register     |
| DACx_DHR8R2   | DACx_Base + 0x1C | Channel 2 8-bit right-aligned data hold register     |
| DACx_DHR12RD  | DACx_Base + 0x20 | Dual-channel 12-bit right-aligned data hold register |
| DACx_DHR12LD  | DACx_Base + 0x24 | Dual-channel 12-bit left-aligned data hold register  |
| DACx_DHR8RD   | DACx_Base + 0x28 | Dual-channel 8-bit right-aligned data hold register  |
| DACx_DOR1     | DACx_Base + 0x2C | Channel 1 data output register                       |
| DACx_DOR2     | DACx_Base + 0x30 | Channel 2 data output register                       |
| DACx_CR1      | DACx_Base + 0x34 | Control register 1                                   |



## 26.10 Register description

See section 1 [Documentation conventions](#) for abbreviations used in register descriptions.

### 26.10.1 DAC\_CR0 Control register 0

Address offset: 0x00      Reset value: 0x0000 0000

| Bit field | Name      | Permission | Function description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|-----------|-----------|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:30     | RFU       | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 29        | DMAUDRIE2 | RW         | DAC channel 2 DMA underflow interrupt enable<br>This bit is set and cleared by software.<br>0: Disable DAC channel 2 DMA underflow interrupt.<br>1: Enable DAC channel 2 DMA underflow interrupt.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 28        | DMAEN2    | RW         | DAC channel 2 DMA enable<br>This bit is set and cleared by software.<br>0: Disable DAC channel 2 DMA mode.<br>1: Enable DAC channel 2 DMA mode.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 27:24     | MAMP2     | RW         | DAC channel 2 mask/amplitude selector<br>These bits are written by the software to select the mask in the noise wave generation mode or the amplitude in the triangular wave generation mode.<br>0000: Unmasked LFSR bit 0/triangular wave amplitude is equal to 1.<br>0001: Unmasked LFSR bit [1:0]/triangular wave amplitude is equal to 3.<br>0010: Unmasked LFSR bit [2:0]/triangular wave amplitude is equal to 7.<br>0011: Unmasked LFSR bit [3:0]/triangular wave amplitude is equal to 15.<br>0100: Unmasked LFSR bit [4:0]/triangular wave amplitude is equal to 31.<br>0101: Unmasked LFSR bit [5:0]/triangular wave amplitude is equal to 63.<br>0110: Unmasked LFSR bit [6:0]/triangular wave amplitude is equal to 127.<br>0111: Unmasked LFSR bit [7:0]/triangular wave amplitude is equal to 255.<br>1000: Unmasked LFSR bit [8:0]/triangular wave amplitude is equal to 511.<br>1001: Unmasked LFSR bit [9:0]/triangular wave amplitude is equal to 1023.<br>1010: Unmasked LFSR bit [10:0]/triangular wave amplitude is equal to 2047.<br>≥ 1011: Unmasked LFSR bit [11:0]/triangular wave amplitude is equal to 4095. |



| Bit field | Name      | Permission | Function description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|-----------|-----------|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 23:22     | WAVE2     | RW         | DAC channel 2 noise/triangular wave generation enable<br>These bits are set and cleared by software.<br>00: Disable wave generation.<br>01: Enable noise wave generation.<br>1x: Enable triangular wave generation.<br><i>Note: Only active when bit TEN2 = 1 (enable DAC channel 2 trigger).</i>                                                                                                                                                                                                                                                                                                                             |
| 21:18     | TSEL2     | RW         | DAC channel 2 trigger selection<br>These bits select the external trigger event for DAC channel 2. For detailed information on trigger configuration and mapping, please refer to <a href="#">Table 26-1: DAC trigger source</a> .<br><i>Note: Used only when bit TEN2 = 1 (enables DAC channel 2 trigger).</i><br><i>Note: Only the output comparison flag of GTIMx_CC is valid; input capture is invalid.</i>                                                                                                                                                                                                               |
| 17        | TEN2      | RW         | DAC channel 2 trigger enable control<br>This bit is set to 1 and cleared by the software to enable/disable the triggering of DAC channel 2.<br>0: Disable DAC channel 2 triggering. Data written to the DAC_DHR2 register is transferred to the DAC_DOR2 register after one dac_pclk clock cycle.<br>1: Enable DAC channel 2 triggering. Data in the DAC_DHR2 register is transferred to the DAC_DOR2 register after three dac_pclk clock cycles.<br><i>Note: If software triggering is selected, the contents of the DAC_DHR2 register can be transferred to the DAC_DOR2 register within a single dac_pclk clock cycle.</i> |
| 16        | EN2       | RW         | DAC channel 2 enable control<br>This bit is set to 1 and cleared by software to enable/disable DAC channel 2.<br>0: Disable DAC channel 2.<br>1: Enable DAC channel 2.<br><i>Note: After the DAC clock is enabled, the value of DHRx is updated to DOR.</i><br><i>Note: It is necessary to configure CR1.DAC_OUTx = 1 for there to be output on the OUT pin.</i>                                                                                                                                                                                                                                                              |
| 15:14     | RFU       | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 13        | DMAUDRIE1 | RW         | DAC channel 1 DMA underflow interrupt enable<br>This bit is set and cleared by software.<br>0: Disable DAC channel 1 DMA underflow interrupt.<br>1: Enable DAC channel 1 DMA underflow interrupt.                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 12        | DMAEN1    | RW         | DAC channel 1 DMA enable<br>This bit is set to 1 and cleared by software.<br>0: Disable DAC channel 1 DMA mode.<br>1: Enable DAC channel 1 DMA mode.                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |





| Bit field | Name  | Permission | Function description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|-----------|-------|------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 11:8      | MAMP1 | RW         | <p>DAC channel 1 mask/amplitude selector</p> <p>These bits are written by the software to select the mask in the noise wave generation mode or the amplitude in the triangular wave generation mode.</p> <p>0000: Unmasked LFSR bit 0/triangular wave amplitude is equal to 1.</p> <p>0001: Unmasked LFSR bit [1:0]/triangular wave amplitude is equal to 3.</p> <p>0010: Unmasked LFSR bit [2:0]/triangular wave amplitude is equal to 7.</p> <p>0011: Unmasked LFSR bit [3:0]/triangular wave amplitude is equal to 15.</p> <p>0100: Unmasked LFSR bit [4:0]/triangular wave amplitude is equal to 31.</p> <p>0101: Unmasked LFSR bit [5:0]/triangular wave amplitude is equal to 63.</p> <p>0110: Unmasked LFSR bit [6:0]/triangular wave amplitude is equal to 127.</p> <p>0111: Unmasked LFSR bit [7:0]/triangular wave amplitude is equal to 255.</p> <p>1000: Unmasked LFSR bit [8:0]/triangular wave amplitude is equal to 511.</p> <p>1001: Unmasked LFSR bit [9:0]/triangular wave amplitude is equal to 1023.</p> <p>1010: Unmasked LFSR bit [10:0]/triangular wave amplitude is equal to 2047.</p> <p><math>\geq 1011</math>: Unmasked LFSR bit [11:0]/triangular wave amplitude is equal to 4095.</p> |
| 7:6       | WAVE1 | RW         | <p>DAC channel 1 noise/triangular wave generation enable</p> <p>These bits are set and cleared by software.</p> <p>00: Disable wave generation.</p> <p>01: Enable noise wave generation.</p> <p>1x: Enable triangular wave generation</p> <p><b>Note:</b> <i>Used only when bit TEN1 = 1 (enable DAC channel 1 trigger).</i></p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| 5:2       | TSEL1 | RW         | <p>DAC channel 1 trigger selection</p> <p>These bits select the external trigger event for DAC channel 1. For detailed information on trigger configuration and mapping, please refer to <a href="#">Table 26-1: DAC trigger source</a>.</p> <p><b>Note:</b> <i>Used only when bit TEN1 = 1 (enable DAC channel 1 trigger).</i></p> <p><b>Note:</b> <i>Only the output comparison flag of GTIMx_CC is valid; input capture is invalid.</i></p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |



| Bit field | Name | Permission | Function description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|-----------|------|------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1         | TEN1 | RW         | <p>DAC channel 1 trigger enable control</p> <p>This bit is set to 1 and cleared by the software to enable/disable the triggering of DAC channel 1.</p> <p>0: Disable DAC channel 1 triggering. Data written to the DAC_DHR1 register is transferred to the DAC_DOR1 register after one dac_pclk clock cycle.</p> <p>1: Enable DAC channel 1 triggering. Data in the DAC_DHR1 register is transferred to the DAC_DOR1 register after three dac_pclk clock cycles.</p> <p><i>Note: If software triggering is selected, the contents of the DAC_DHR1 register can be transferred to the DAC_DOR1 register within a single dac_pclk clock cycle.</i></p> |
| 0         | EN1  | RW         | <p>DAC channel 1 enable control</p> <p>This bit is set to 1 and cleared by software to enable/disable DAC channel 1.</p> <p>0: Disable DAC channel 1.</p> <p>1: Enable DAC channel 1.</p> <p><i>Note: After the DAC clock is enabled, the value of DHRx is updated to DOR.</i></p> <p><i>Note: It is necessary to configure CR1.DAC_OUTx = 1 for there to be output on the OUT pin.</i></p>                                                                                                                                                                                                                                                          |



### 26.10.2 DAC\_SWTRGR Software trigger register

Address offset: 0x04      Reset value: 0x0000 0000

| Bit field | Name    | Permission | Function description                                                                                                                                                                                                                                                                                        |
|-----------|---------|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:2      | RFU     | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                               |
| 1         | SWTRIG2 | RW         | DAC channel 2 software trigger<br>This bit is set to 1 by software to trigger the DAC in software trigger mode.<br>0: Not triggered<br>1: Triggered<br><i>Note: Once the DAC_DHR2 register value is loaded into the DAC_DOR2 register, the bit is cleared by hardware (after one dac_pclk clock cycle).</i> |
| 0         | SWTRIG1 | RW         | DAC channel 1 software trigger<br>This bit is set to 1 by software to trigger the DAC in software trigger mode.<br>0: Not triggered<br>1: Triggered<br><i>Note: Once the DAC_DHR1 register value is loaded into the DAC_DOR1 register, the bit is cleared by hardware (after one dac_pclk clock cycle).</i> |

### 26.10.3 DAC\_DHR12R1 Channel 1 12-bit right-aligned data hold register

Address offset: 0x08      Reset value: 0x0000 0000

| Bit field | Name     | Permission | Function description                                                                                                                    |
|-----------|----------|------------|-----------------------------------------------------------------------------------------------------------------------------------------|
| 31:12     | RFU      | -          | Reserved bits, please keep the default value.                                                                                           |
| 11:0      | DACC1DHR | RW         | DAC channel 1 12-bit right-aligned data<br><i>Note: These bits are written by software to specify the 12-bit data of DAC channel 1.</i> |

### 26.10.4 DAC\_DHR12L1 Channel 1 12-bit left-aligned data hold register

Address offset: 0x0C      Reset value: 0x0000 0000

| Bit field | Name     | Permission | Function description                                                                                                                   |
|-----------|----------|------------|----------------------------------------------------------------------------------------------------------------------------------------|
| 31:16     | RFU      | -          | Reserved bits, please keep the default value.                                                                                          |
| 15:4      | DACC1DHR | RW         | DAC channel 1 12-bit left-aligned data<br><i>Note: These bits are written by software to specify the 12-bit data of DAC channel 1.</i> |
| 3:0       | RFU      | -          | Reserved bits, please keep the default value.                                                                                          |



### 26.10.5 DAC\_DHR8R1 Channel 1 8-bit right-aligned data hold register

Address offset: 0x10      Reset value: 0x0000 0000

| Bit field | Name     | Permission | Function description                                                                                                                  |
|-----------|----------|------------|---------------------------------------------------------------------------------------------------------------------------------------|
| 31:8      | RFU      | -          | Reserved bits, please keep the default value.                                                                                         |
| 7:0       | DACC1DHR | RW         | DAC channel 1 8-bit right-aligned data<br><i>Note: These bits are written by software to specify the 8-bit data of DAC channel 1.</i> |

### 26.10.6 DAC\_DHR12R2 Channel 2 12-bit right-aligned data hold register

Address offset: 0x14      Reset value: 0x0000 0000

| Bit field | Name     | Permission | Function description                                                                                                                    |
|-----------|----------|------------|-----------------------------------------------------------------------------------------------------------------------------------------|
| 31:12     | RFU      | -          | Reserved bits, please keep the default value.                                                                                           |
| 11:0      | DACC2DHR | RW         | DAC channel 2 12-bit right-aligned data<br><i>Note: These bits are written by software to specify the 12-bit data of DAC channel 2.</i> |

### 26.10.7 DAC\_DHR12L2 Channel 2 12-bit left-aligned data hold register

Address offset: 0x18      Reset value: 0x0000 0000

| Bit field | Name     | Permission | Function description                                                                                                                   |
|-----------|----------|------------|----------------------------------------------------------------------------------------------------------------------------------------|
| 31:16     | RFU      | -          | Reserved bits, please keep the default value.                                                                                          |
| 15:4      | DACC2DHR | RW         | DAC channel 2 12-bit left-aligned data<br><i>Note: These bits are written by software to specify the 12-bit data of DAC channel 2.</i> |
| 3:0       | RFU      | -          | Reserved bits, please keep the default value.                                                                                          |

### 26.10.8 DAC\_DHR8R2 Channel 2 8-bit right-aligned data hold register

Address offset: 0x1C      Reset value: 0x0000 0000

| Bit field | Name     | Permission | Function description                                                                                                                  |
|-----------|----------|------------|---------------------------------------------------------------------------------------------------------------------------------------|
| 31:8      | RFU      | -          | Reserved bits, please keep the default value.                                                                                         |
| 7:0       | DACC2DHR | RW         | DAC channel 2 8-bit right-aligned data<br><i>Note: These bits are written by software to specify the 8-bit data of DAC channel 2.</i> |



### 26.10.9 DAC\_DHR12RD Dual-channel 12-bit right-aligned data hold register

Address offset: 0x20      Reset value: 0x0000 0000

| Bit field | Name     | Permission | Function description                                                                                                                    |
|-----------|----------|------------|-----------------------------------------------------------------------------------------------------------------------------------------|
| 31:28     | RFU      | -          | Reserved bits, please keep the default value.                                                                                           |
| 27:16     | DACC2DHR | RW         | DAC channel 2 12-bit right-aligned data<br><i>Note: These bits are written by software to specify the 12-bit data of DAC channel 2.</i> |
| 15:12     | RFU      | -          | Reserved bits, please keep the default value.                                                                                           |
| 11:0      | DACC1DHR | RW         | DAC channel 1 12-bit right-aligned data<br><i>Note: These bits are written by software to specify the 12-bit data of DAC channel 1.</i> |

### 26.10.10 DAC\_DHR12LD Dual-channel 12-bit left-aligned data hold register

Address offset: 0x24      Reset value: 0x0000 0000

| Bit field | Name     | Permission | Function description                                                                                                                   |
|-----------|----------|------------|----------------------------------------------------------------------------------------------------------------------------------------|
| 31:20     | DACC2DHR | RW         | DAC channel 2 12-bit left-aligned data<br><i>Note: These bits are written by software to specify the 12-bit data of DAC channel 2.</i> |
| 19:16     | RFU      | -          | Reserved bits, please keep the default value.                                                                                          |
| 15:4      | DACC1DHR | RW         | DAC channel 1 12-bit left-aligned data<br><i>Note: These bits are written by software to specify the 12-bit data of DAC channel 1.</i> |
| 3:0       | RFU      | -          | Reserved bits, please keep the default value.                                                                                          |

### 26.10.11 DAC\_DHR8RD Dual-channel 8-bit right-aligned data hold register

Address offset: 0x28      Reset value: 0x0000 0000

| Bit field | Name     | Permission | Function description                                                                                                                  |
|-----------|----------|------------|---------------------------------------------------------------------------------------------------------------------------------------|
| 31:16     | RFU      | -          | Reserved bits, please keep the default value.                                                                                         |
| 15:8      | DACC2DHR | RW         | DAC channel 2 8-bit right-aligned data<br><i>Note: These bits are written by software to specify the 8-bit data of DAC channel 2.</i> |
| 7:0       | DACC1DHR | RW         | DAC channel 1 8-bit right-aligned data<br><i>Note: These bits are written by software to specify the 8-bit data of DAC channel 1.</i> |



### 26.10.12 DAC\_DOR1 Channel 1 data output register

Address offset: 0x2C      Reset value: 0x0000 0000

| Bit field | Name     | Permission | Function description                                                                                                                                                                                                                                                                                       |
|-----------|----------|------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:16     | RFU      | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                              |
| 15        | DMAUDR1  | RW0        | DAC channel 1 DMA underflow flag<br>0: No DMA underflow error situation occurred in DAC channel 1.<br>1: A DMA underflow error situation occurred in DAC channel 1 (the currently selected trigger source is driving the conversion of DAC channel 1 at a frequency higher than the DMA service capacity). |
| 14:12     | RFU      | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                              |
| 11:0      | DACC1DOR | RO         | DAC channel 1 data output<br><i>Note: These bits are read-only and are used to specify the data output of DAC channel 1.</i>                                                                                                                                                                               |

### 26.10.13 DAC\_DOR2 Channel 2 data output register

Address offset: 0x30      Reset value: 0x0000 0000

| Bit field | Name     | Permission | Function description                                                                                                                                                                                                                                                                                       |
|-----------|----------|------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:16     | RFU      | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                              |
| 15        | DMAUDR2  | RW0        | DAC channel 2 DMA underflow flag<br>0: No DMA underflow error situation occurred in DAC channel 2.<br>1: A DMA underflow error situation occurred in DAC channel 2 (the currently selected trigger source is driving the conversion of DAC channel 2 at a frequency higher than the DMA service capacity). |
| 14:12     | RFU      | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                              |
| 11:0      | DACC2DOR | RO         | DAC channel 2 data output<br><i>Note: These bits are read-only and are used to specify the data output of DAC channel 2.</i>                                                                                                                                                                               |



### 26.10.14 DAC\_CR1 Control register 1

Address offset: 0x34

Reset value: 0x0000 0000

| Bit field | Name     | Permission | Function description                                                                                                                                                                                                                                                                                                                                                |
|-----------|----------|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:2      | RFU      | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                                                       |
| 1         | DAC_OUT2 | RW         | DAC channel 2 output to pin enable control (internal interconnect is not affected)<br>0: Disable DAC channel 2 output.<br>1: Enable DAC channel 2 output.<br><i>Note: It is necessary to configure CR0.ENx = 1 for there to be an output on the OUT pin. The common output pin of OPA/DAC can only have one output at a time; otherwise, a conflict will occur.</i> |
| 0         | DAC_OUT1 | RW         | DAC channel 1 output to pin enable control (internal interconnect is not affected)<br>0: Disable DAC channel 1 output.<br>1: Enable DAC channel 1 output.<br><i>Note: It is necessary to configure CR0.ENx = 1 for there to be an output on the OUT pin. The common output pin of OPA/DAC can only have one output at a time; otherwise, a conflict will occur.</i> |



## 27 Analog voltage comparator (VC)

### 27.1 Overview

CW32L012 integrates 4 analog voltage comparators (VC), which are used to compare two analog input voltages, and output the comparison results from the pins. The positive terminal input of the voltage comparator supports 4 external analog inputs, and the negative terminal supports not only 2 external analog inputs, but also internal DAC conversion results and internal resistance voltage divider output voltage. The comparison result output has filtering function, hysteresis window function, and polarity selection. Support compare interrupt, which can be used to wake up MCU in low power mode.





## 27.2 Main features

- 2 channel analog voltage comparators VC1, VC2, VC3, VC4
- Internal 8-step resistor divider.
- 4 external analog signal inputs.
- 2 on-chip analog input signals:
  - Built-in resistor divider output voltage.
  - Internal DAC conversion result.
- Selectable output polarity.
- Supports hysteresis window compare function.
- Programmable filters and filter time.
- 3 interrupt trigger modes, which can be used in combination:
  - High level trigger.
  - Rising edge trigger.
  - Falling edge trigger.
- Supports running in low power mode, interrupt wake-up MCU.

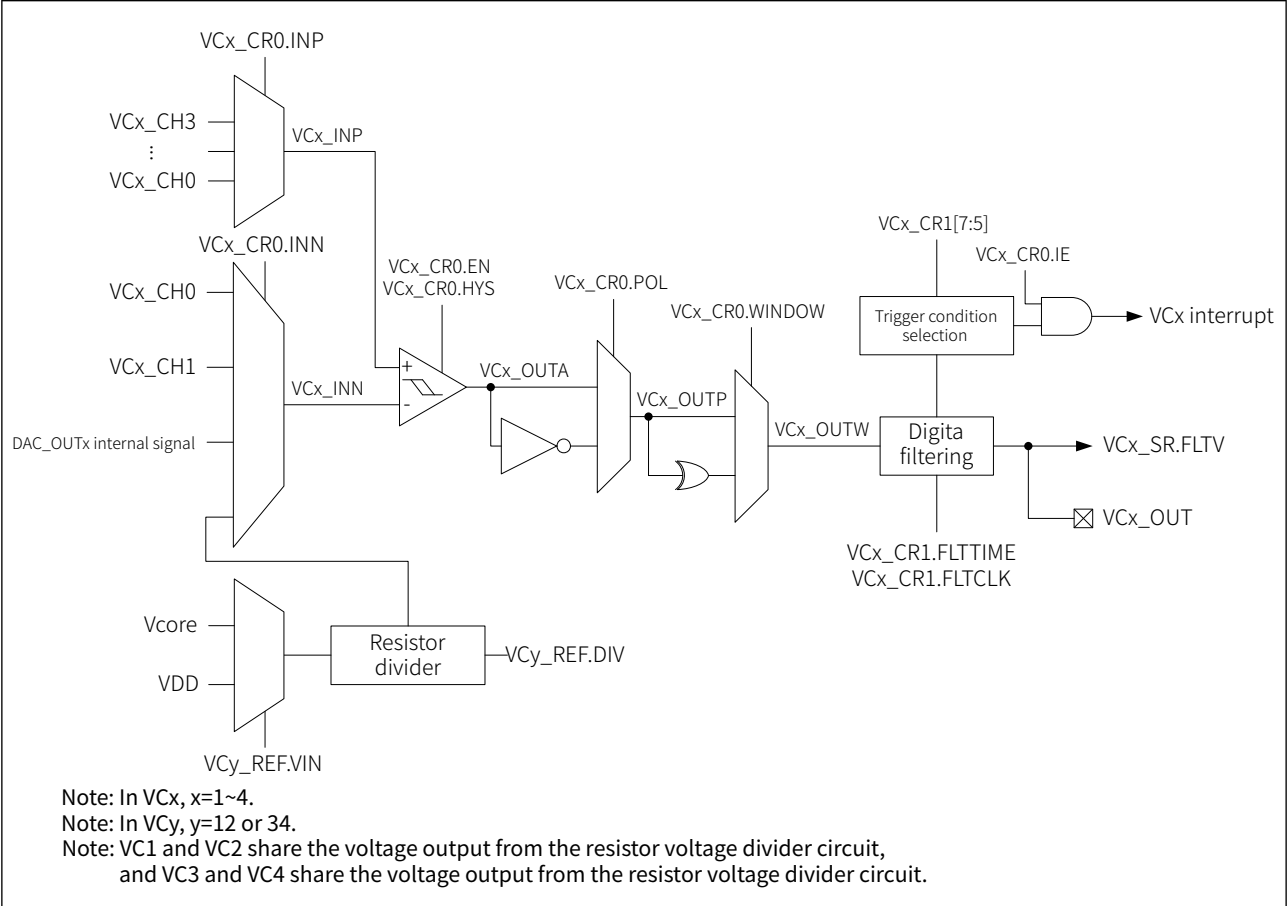


27.3 Functional description

27.3.1 Functional block diagram

The functional block diagram of the analog voltage comparator (VC) is shown in the following figure:

Figure 27-1 VC functional block diagram



The input selection of the positive and negative terminals of VCx (In VC chapter, x = 1, 2, 3 or 4, i.e., x = 1~4.) is selected by the INP and INN bit fields of the control register VCx\_CR0, as shown in the following table:

Table 27-1 VC positive and negative input signal configuration

| VCx_CR0.INP | VCx positive terminal input signal | VCx_CR0.INN | VCx negative terminal input signal       |
|-------------|------------------------------------|-------------|------------------------------------------|
| 00          | VCx_CH0                            | 00          | VCx_CH0                                  |
| 01          | VCx_CH1                            | 01          | VCx_CH1                                  |
| 10          | VCx_CH2                            | 10          | Internal signal DAC_OUTx                 |
| 11          | VCx_CH3                            | 11          | Built-in resistor divider output voltage |

The built-in resistor divider configuration is controlled by three bit fields in the VCy\_REF (In VC chapter, y = 12 or 34, the same applies below.) register:

- EN bit field, enable resistor divider.
- VIN bit field, select the input voltage of the voltage divider:
  - VIN is 0, select V<sub>DDA</sub>.
  - VIN is 1, select V<sub>core</sub>, about 1.6V.
- DIV bit field, voltage divider proportional coefficient, valid range is 0~7, select the output voltage of the voltage divider, the calculation formula is as follows:

$$\text{Voltage divider output voltage} = \text{input voltage} \times (1 + \text{DIV}) / 8$$

*Note: VC1 and VC2 share the voltage output from the resistor voltage divider circuit, and VC3 and VC4 share the voltage output from the resistor voltage divider circuit.*

When selecting the built-in 1.2V reference voltage as the negative input of the comparator, it is necessary to enable the BGR module inside the chip. The startup time of the internal BGR is about 30μs, the VC voltage comparator needs to wait for the internal BGR to stabilize before it can work normally.



### 27.3.2 I/O pins

The analog voltage comparator supports 8 external analog signal inputs, and the user must configure the corresponding GPIO port as an analog function (GPIOx\_ANALOG.PINy = 1).

The analog voltage comparator supports outputting the comparison result from the pin. The user must configure the corresponding GPIO port as a digital output and select the function multiplexing.

The I/O pins supported by VCx are shown in the following table:

Table 27-2 VC I/O pins configuration

| VC1 I/O | GPIO | Configuration               | VC2 I/O | GPIO | Configuration               | VC3 I/O | GPIO | Configuration               | VC4 I/O | GPIO | Configuration               |
|---------|------|-----------------------------|---------|------|-----------------------------|---------|------|-----------------------------|---------|------|-----------------------------|
| VC1_CH0 | PA00 | Analog                      | VC2_CH0 | PA01 | Analog                      | VC3_CH0 | PA06 | Analog                      | VC4_CH0 | PA09 | Analog                      |
| VC1_CH1 | PA02 | Analog                      | VC2_CH1 | PA03 | Analog                      | VC3_CH1 | PA07 | Analog                      | VC4_CH1 | PA10 | Analog                      |
| VC1_CH2 | PA04 | Analog                      | VC2_CH2 | PA05 | Analog                      | VC3_CH2 | PB02 | Analog                      | VC4_CH2 | PB10 | Analog                      |
| VC1_CH3 | PB00 | Analog                      | VC2_CH3 | PB01 | Analog                      | VC3_CH3 | PB00 | Analog                      | VC4_CH3 | PB01 | Analog                      |
| VC1_OUT | PA00 | Digital output<br>(AFR = 4) | VC2_OUT | PA02 | Digital output<br>(AFR = 4) | VC3_OUT | PA00 | Digital output<br>(AFR = 9) | VC4_OUT | PA08 | Digital output<br>(AFR = 2) |
|         | PA11 | Digital output<br>(AFR = 2) |         | PA12 | Digital output<br>(AFR = 2) |         | PB14 | Digital output<br>(AFR = 5) |         | PB10 | Digital output<br>(AFR = 5) |
|         | PB12 | Digital output<br>(AFR = 5) |         | PB13 | Digital output<br>(AFR = 5) |         | -    | -                           |         | -    | -                           |

### 27.3.3 Latency/Response time

Set the EN bit field of the control register VCx\_CR0 to 1 to enable the VC module.

The time from when VC is enabled or the input voltage at the positive and negative ends of VC changes to the time when the voltage comparator outputs the correct comparison result is defined as the delay/response time of the comparator. The delay/response time is configured by the RESP bit field of the control register VCx\_CR0. The shorter the response time, the greater the power consumption of the VC module.

### 27.3.4 Polarity selection

The polarity of the output signals of the voltage comparators VC1, VC2, VC3, VC4 is set by the POL bit field of the control register VCx\_CR0:

- POL is 1, the polarity of VCx\_OUTP signal is opposite to that of VCx\_OUTA signal, that is, when the positive terminal is greater than the negative terminal, VCx outputs a low level.
- POL is 0, and the VCx\_OUTP signal has the same polarity as the VCx\_OUTA signal, that is, when the positive terminal is greater than the negative terminal, VCx outputs a high level.



### 27.3.5 Digital filter

The built-in digital filter of the voltage comparator is used to digitally filter the output signal of the voltage comparator. Users can use the filtering function to filter system noise, such as high current noise when the motor is stopped, to avoid system malfunction caused by the noise output of the comparator.

The clock for the digital filter is selected by the FLTCLK bit field in the control register VCx\_CR1:

- FLTCLK is 1, use PCLK as filter clock.
- FLTCLK is 0, the built-in low speed RC oscillator clock LSI is used as the filter clock.

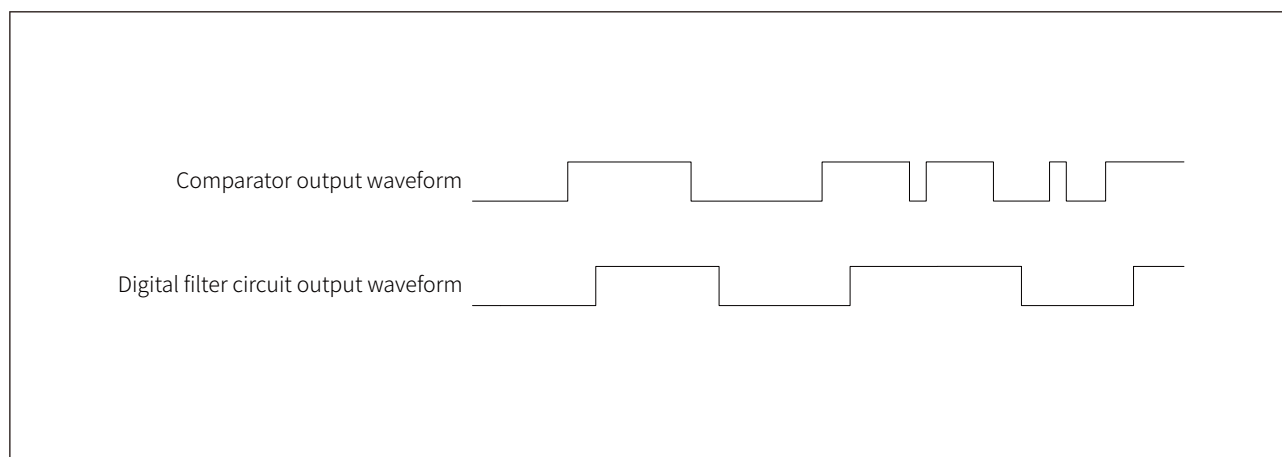
The filter samples the output signal with a certain sampling frequency. When the continuous sampling reaches N same levels, the signal is valid, otherwise the signal is invalid, so as to filter out high-frequency clutter signals. The sampling clock of the filter unit is FLTCLK or the frequency division of FLTCLK, and the frequency division ratio of FLTCLK and the number of sampling points N can be selected through the FLTTIME bit field of the VCx\_CR1 register.

The digitally filtered output level of the voltage comparator can be read out through the FLTV bit field of the status register VCx\_SR.

When the user sets the VCx\_OUT pin as a digital output and selects the multiplexing function of the VC comparison output, the VCx\_OUT pin will output the digitally filtered output level of the voltage comparator.

The filter response waveform of VC is shown in the following figure:

Figure 27-2 VC filter response time



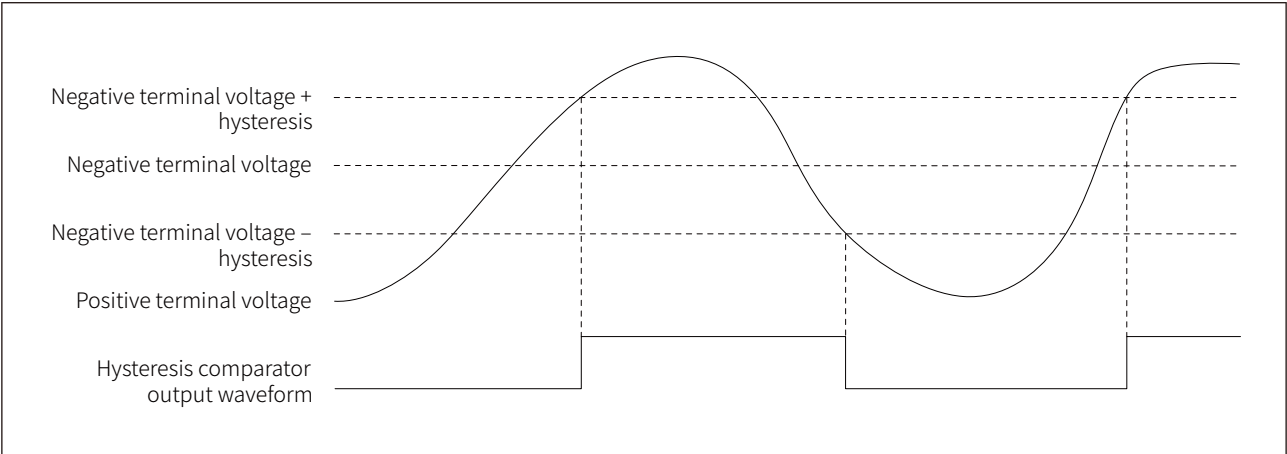
The output of the voltage comparator can be used as the brake input or OCREF\_CLR input of the Advanced Timer (ATIM), for details about the brake and OCREF\_CLR, please refer to [17 Advanced-control timer \(ATIM\)](#). This function is usually used in motor protection applications, please refer to the relevant application notes for details.

27.3.6 Hysteresis function

The analog voltage comparator supports the hysteresis function. After the hysteresis function is used, the output result of the comparator will not be reversed immediately with the change of the input signal, but will reverse only after the offset value of the two input signals is higher or lower than the hysteresis threshold voltage.

The waveform diagram of VC hysteresis function is shown in the following figure:

Figure 27-3 VC hysteresis function



The hysteresis function can effectively enhance the anti-interference ability of the chip, and avoid unnecessary frequent flipping of the output end of the comparator due to the small amplitude jitter of the input signal.

The hysteresis threshold voltage is determined by the HYS bit field of the control register VCx\_CR0, as shown in the following table:

Table 27-3 VC hysteresis threshold voltage

| VCx_CR0.HYS | Hysteresis window configuration |
|-------------|---------------------------------|
| 0           | No hysteresis.                  |
| 1           | Hysteresis window about 20mV.   |

27.3.7 Window compare function

The analog voltage comparator supports the window compare function, which can output the comparison results of VC1 and VC2 after XOR operation, which is enabled by the WINDOW bit field of the control register VCx\_CR0:

- When WINDOW is 1, the VCx\_OUTW signal is the XOR value of the VC1\_OUTP signal and the VC2\_OUTP signal.
- When WINDOW is 0, the VCx\_OUTW signal is at the same level as the VCx\_OUTP signal.

### 27.3.8 BLANK window function

While keeping the VCx module working, if you want to temporarily stop the voltage comparison function, or in some application systems (such as motor control), the short-term reasonable fluctuation of the monitored signal may cause unnecessary reverse of output levels of voltage comparator, the voltage comparator of this chip has added the BLANK window function. That is, when the BLANK window is activated by a specified external trigger condition, no voltage comparison is performed during the set BLANK window period, and the voltage comparator outputs a set high or low level. level status. After the BLANK window period, the voltage comparator resumes normal operation.

The BLANK window duration is configured by the BLANKTIME bit field of the control register VCx\_CR1. The window duration is:

$$(2^{(\text{BLANKTIME} + 2)} \sim 2^{(\text{BLANKTIME} + 2)} + 2) \text{ PCLK cycles}$$

As shown in the following table:

Table 27-4 BLANK window duration configuration

| VCx_CR1.BLANKTIME | The number of PCLK clocks in the BLANK window |
|-------------------|-----------------------------------------------|
| 000               | 4 ~ 6                                         |
| 001               | 8 ~ 10                                        |
| 010               | 16 ~ 18                                       |
| 011               | 32 ~ 34                                       |
| 100               | 64 ~ 66                                       |
| 101               | 128 ~ 130                                     |
| 110               | 256 ~ 258                                     |
| 111               | 512 ~ 514                                     |

The output level of the voltage comparator during the BLANK window duration is set by the BLANKLVL bit field of control register VCx\_CR1. BLANKLVL outputs a low level when BLANKLVL is 0, and outputs a high level when BLANKLVL is 1.

The trigger start condition of the BLANK window is configured by the BLANKCH1, BLANKCH2, BLANKCH3, BLANKCH4, BLANKCH5, and BLANKCH6 bit fields of the control register VCx\_CR1, and the BLANK window is triggered by the rising edges of the ATIM's OC1REFC, OC2REFC, OC3REFC, OC4REFC, OC5REFC, and OC6REFC respectively.

The trigger start conditions for the BLANK window are configured by control register VCx\_CR2. For details, refer to the register description.



## 27.4 VC interrupts

The voltage comparator of CW32L012 supports working in low power consumption mode, and the comparison interrupt can wake up the chip from low power consumption mode.

Set the IE bit field of the control register VCx\_CR0 to 1 to enable the VCx interrupt. When an interrupt occurs, the interrupt flag bit INTF of the status register VCx\_SR will be set to 1 by hardware. The user can write 0 to the INTF bit to clear the interrupt flag.

Set the HIGHIE, RISEIE, FALLIE bit fields of the control register VCx\_CR1 to select different interrupt triggering methods:

- HIGHIE is 1, VCx\_OUT output signal high level triggers interrupt.
- RISEIE is 1, the rising edge of VCx\_OUT output signal triggers interrupt.
- FALLIE is 1, the falling edge of VCx\_OUT output signal triggers interrupt.





## 27.5 Programming examples

In this example, the digital filtering function is used, and the compare interrupt is configured as follows:

- Step 1:** Configure corresponding VCy\_REF.EN to 1, enable the voltage divider.
- Step 2:** Configure corresponding VCy\_REF.DIV and set the voltage division coefficient.
- Step 3:** Configure VCx\_CR0.INP and select the voltage source to be monitored at the positive terminal.
- Step 4:** Configure VCx\_CR0.INN and select the voltage source to be monitored at the negative terminal.
- Step 5:** Configure VCx\_CR1.FLTTIME and select the filter time.
- Step 6:** Configure VCx\_CR1.FLTCLK, select filter clock.
- Step 7:** Set the HIGHIE, RISEIE, and FALLIE of the VCx\_CR1 register to 1, and select the interrupt trigger mode.
- Step 8:** Set VCx\_CR0.IE to 1 to enable VCx interrupt.
- Step 9:** Set VCx\_CR0.EN to 1 to enable VCx.
- Step 10:** In the initialization routine of the VC module and the interrupt service routine of the VC module, write 0 to the INTF bit of the VCx\_SR register, clear the interrupt flag, and enable the generation of the VCx interrupt.



## 27.6 List of registers

VC12REF base address: VC12REF\_BASE = 0x4000 0080

VC34REF base address: VC34REF\_BASE = 0x4000 0180

VC1 base address: VC1\_Base = 0x4000 0080

VC2 base address: VC2\_Base = 0x4000 0090

VC3 base address: VC3\_Base = 0x4000 0180

VC4 base address: VC4\_Base = 0x4000 0190

Table 27-5 List of VC registers

| Register name | Register address    | Register description                      |
|---------------|---------------------|-------------------------------------------|
| VC12_REF      | VC12REF_BASE + 0x00 | VC1/VC2 resistor divider control register |
| VC34_REF      | VC34REF_BASE + 0x00 | VC3/VC4 resistor divider control register |
| VCx_CR0       | VCx_BASE + 0x04     | Control register 0                        |
| VCx_CR1       | VCx_BASE + 0x08     | Control register 1                        |
| VCx_CR2       | VCx_BASE + 0x0C     | Control register 2                        |
| VCx_SR        | VCx_BASE + 0x10     | Status register                           |



## 27.7 Register description

See section 1 [Documentation conventions](#) for abbreviations used in register descriptions.

### 27.7.1 VC12\_REF VC1/VC2 resistor divider control register

Address offset: 0x00      Reset value: 0x0000 0000

| Bit field | Name | Permission | Function description                                                                                       |
|-----------|------|------------|------------------------------------------------------------------------------------------------------------|
| 31:6      | RFU  | -          | Reserved bits, please keep the default value.                                                              |
| 5         | VIN  | RW         | Resistor divider circuit input voltage setting<br>0: VDDA<br>1: Vcore (about 1.6V)                         |
| 4         | EN   | RW         | Resistor divider circuit enable control<br>0: Disabled<br>1: Enabled                                       |
| 3:0       | DIV  | RW         | Resistor divider circuit output voltage configuration<br>Output voltage = Input voltage $\times$ (1+DIV)/8 |

*Note: VC1 and VC2 share the voltage output from the resistor voltage divider circuit.*

### 27.7.2 VC34\_REF VC3/VC4 resistor divider control register

Address offset: 0x00      Reset value: 0x0000 0000

| Bit field | Name | Permission | Function description                                                                                       |
|-----------|------|------------|------------------------------------------------------------------------------------------------------------|
| 31:6      | RFU  | -          | Reserved bits, please keep the default value.                                                              |
| 5         | VIN  | RW         | Resistor divider circuit input voltage setting<br>0: VDDA<br>1: Vcore (about 1.6V)                         |
| 4         | EN   | RW         | Resistor divider circuit enable control<br>0: Disabled<br>1: Enabled                                       |
| 3:0       | DIV  | RW         | Resistor divider circuit output voltage configuration<br>Output voltage = Input voltage $\times$ (1+DIV)/8 |

*Note: VC3 and VC4 share the voltage output from the resistor voltage divider circuit.*



Address offset: 0x04      Reset value: 0x0000 0000


**Reference Manual**  
**武汉芯源半导体有限公司**  
 WUHAN XINYUAN SEMICONDUCTOR CO., LTD.

### 27.7.4 VCx\_CR1 Control register 1

Address offset: 0x08      Reset value: 0x0000 0010

| Bit field | Name      | Permission | Function description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|-----------|-----------|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:12     | RFU       | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 11        | BLANKLVL  | RW         | BLANK window duration output level configuration<br>0: Output low level.<br>1: Output high level.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 10:8      | BLANKTIME | RW         | BLANK window duration configuration:<br>$(2^{(\text{BLANKTIME}+2)} \sim 2^{(\text{BLANKTIME}+2)+2})$ PCLK cycles.<br><i>Note: Any rising edge of OC1REFC, OC2REFC, OC3REFC, OC4REFC, OC5REFC, OC6REFC of ATIM will trigger the start of the BLANK window. During the duration of the BLANK window, no voltage comparison will be performed.</i>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| 7         | HIGHIE    | RW         | VCx output signal high level trigger interrupt enabled<br>0: Disabled<br>1: Enabled                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| 6         | RISEIE    | RW         | VCx output signal rising edge trigger interrupt enabled<br>0: Disabled<br>1: Enabled                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 5         | FALLIE    | RW         | VCx output signal falling edge trigger interrupt enabled<br>0: Disabled<br>1: Enabled                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 4         | FLTCLK    | RW         | Digital filter module filter clock settings<br>0: LSI<br>1: PCLK<br><i>Note: Before entering DeepSleep mode, when the digital filtering function is not required, please set the filtering clock source to PCLK, otherwise the LSI will generate power consumption when running in DeepSleep mode (The default value has been configured as PCLK).</i>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 3:0       | FLTTIME   | RW         | Digital filter module filter time configuration<br><div> <div> 0000: No filter<br/> 0001: <math>f_{\text{SAMPLING}} = f_{\text{FLTCLK}}/1, N = 2</math><br/> 0010: <math>f_{\text{SAMPLING}} = f_{\text{FLTCLK}}/1, N = 4</math><br/> 0011: <math>f_{\text{SAMPLING}} = f_{\text{FLTCLK}}/1, N = 8</math><br/> 0100: <math>f_{\text{SAMPLING}} = f_{\text{FLTCLK}}/2, N = 6</math><br/> 0101: <math>f_{\text{SAMPLING}} = f_{\text{FLTCLK}}/2, N = 8</math><br/> 0110: <math>f_{\text{SAMPLING}} = f_{\text{FLTCLK}}/4, N = 6</math><br/> 0111: <math>f_{\text{SAMPLING}} = f_{\text{FLTCLK}}/4, N = 8</math> </div> <div> 1000: <math>f_{\text{SAMPLING}} = f_{\text{FLTCLK}}/8, N = 6</math><br/> 1001: <math>f_{\text{SAMPLING}} = f_{\text{FLTCLK}}/8, N = 8</math><br/> 1010: <math>f_{\text{SAMPLING}} = f_{\text{FLTCLK}}/16, N = 5</math><br/> 1011: <math>f_{\text{SAMPLING}} = f_{\text{FLTCLK}}/16, N = 6</math><br/> 1100: <math>f_{\text{SAMPLING}} = f_{\text{FLTCLK}}/16, N = 8</math><br/> 1101: <math>f_{\text{SAMPLING}} = f_{\text{FLTCLK}}/32, N = 5</math><br/> 1110: <math>f_{\text{SAMPLING}} = f_{\text{FLTCLK}}/32, N = 6</math><br/> 1111: <math>f_{\text{SAMPLING}} = f_{\text{FLTCLK}}/32, N = 8</math> </div> </div> |



### 27.7.5 VCx\_CR2 Control register 2

Address offset: 0x0C      Reset value: 0x0000 0000

| Bit field | Name         | Permission | Function description                                                                                  |
|-----------|--------------|------------|-------------------------------------------------------------------------------------------------------|
| 31        | HALLTIMTRGO  | RW         | HALLTIMTRGO signal triggers the BLANK window enable control<br>0: Disabled<br>1: Enabled              |
| 30        | BTIM3TRGO    | RW         | BTIM3TRGO signal triggers the BLANK window enable control<br>0: Disabled<br>1: Enabled                |
| 29        | BTIM2TRGO    | RW         | BTIM2TRGO signal triggers the BLANK window enable control<br>0: Disabled<br>1: Enabled                |
| 28        | BTIM1TRGO    | RW         | BTIM1TRGO signal triggers the BLANK window enable control<br>0: Disabled<br>1: Enabled                |
| 27        | GTIM4OC4REFC | RW         | GTIM4OC4REFC rising edge signal triggers the BLANK window enable control<br>0: Disabled<br>1: Enabled |
| 26        | GTIM4OC3REFC | RW         | GTIM4OC3REFC rising edge signal triggers the BLANK window enable control<br>0: Disabled<br>1: Enabled |
| 25        | GTIM4OC2REFC | RW         | GTIM4OC2REFC rising edge signal triggers the BLANK window enable control<br>0: Disabled<br>1: Enabled |
| 24        | GTIM4OC1REFC | RW         | GTIM4OC1REFC rising edge signal triggers the BLANK window enable control<br>0: Disabled<br>1: Enabled |
| 23        | GTIM4TRGO    | RW         | GTIM4TRGO signal triggers the BLANK window enable control<br>0: Disabled<br>1: Enabled                |
| 22        | GTIM3OC4REFC | RW         | GTIM3OC4REFC rising edge signal triggers the BLANK window enable control<br>0: Disabled<br>1: Enabled |



| Bit field | Name         | Permission | Function description                                                                                  |
|-----------|--------------|------------|-------------------------------------------------------------------------------------------------------|
| 21        | GTIM3OC3REFC | RW         | GTIM3OC3REFC rising edge signal triggers the BLANK window enable control<br>0: Disabled<br>1: Enabled |
| 20        | GTIM3OC2REFC | RW         | GTIM3OC2REFC rising edge signal triggers the BLANK window enable control<br>0: Disabled<br>1: Enabled |
| 19        | GTIM3OC1REFC | RW         | GTIM3OC1REFC rising edge signal triggers the BLANK window enable control<br>0: Disabled<br>1: Enabled |
| 18        | GTIM3TRGO    | RW         | GTIM3TRGO signal triggers the BLANK window enable control<br>0: Disabled<br>1: Enabled                |
| 17        | GTIM2OC4REFC | RW         | GTIM2OC4REFC rising edge signal triggers the BLANK window enable control<br>0: Disabled<br>1: Enabled |
| 16        | GTIM2OC3REFC | RW         | GTIM2OC3REFC rising edge signal triggers the BLANK window enable control<br>0: Disabled<br>1: Enabled |
| 15        | GTIM2OC2REFC | RW         | GTIM2OC2REFC rising edge signal triggers the BLANK window enable control<br>0: Disabled<br>1: Enabled |
| 14        | GTIM2OC1REFC | RW         | GTIM2OC1REFC rising edge signal triggers the BLANK window enable control<br>0: Disabled<br>1: Enabled |
| 13        | GTIM2TRGO    | RW         | GTIM2TRGO signal triggers the BLANK window enable control<br>0: Disabled<br>1: Enabled                |
| 12        | GTIM1OC4REFC | RW         | GTIM1OC4REFC rising edge signal triggers the BLANK window enable control<br>0: Disabled<br>1: Enabled |
| 11        | GTIM1OC3REFC | RW         | GTIM1OC3REFC rising edge signal triggers the BLANK window enable control<br>0: Disabled<br>1: Enabled |



| Bit field | Name         | Permission | Function description                                                                                  |
|-----------|--------------|------------|-------------------------------------------------------------------------------------------------------|
| 10        | GTIM1OC2REFC | RW         | GTIM1OC2REFC rising edge signal triggers the BLANK window enable control<br>0: Disabled<br>1: Enabled |
| 9         | GTIM1OC1REFC | RW         | GTIM1OC1REFC rising edge signal triggers the BLANK window enable control<br>0: Disabled<br>1: Enabled |
| 8         | GTIM1TRGO    | RW         | GTIM1TRGO signal triggers the BLANK window enable control<br>0: Disabled<br>1: Enabled                |
| 7         | ATIMOC6REFC  | RW         | ATIM OC6REFC rising edge signal triggers the BLANK window enable control<br>0: Disabled<br>1: Enabled |
| 6         | ATIMOC5REFC  | RW         | ATIM OC5REFC rising edge signal triggers the BLANK window enable control<br>0: Disabled<br>1: Enabled |
| 5         | ATIMOC4REFC  | RW         | ATIM OC4REFC rising edge signal triggers the BLANK window enable control<br>0: Disabled<br>1: Enabled |
| 4         | ATIMOC3REFC  | RW         | ATIM OC3REFC rising edge signal triggers the BLANK window enable control<br>0: Disabled<br>1: Enabled |
| 3         | ATIMOC2REFC  | RW         | ATIM OC2REFC rising edge signal triggers the BLANK window enable control<br>0: Disabled<br>1: Enabled |
| 2         | ATIMOC1REFC  | RW         | ATIM OC1REFC rising edge signal triggers the BLANK window enable control<br>0: Disabled<br>1: Enabled |
| 1         | ATIMTRGO2    | RW         | ATIM TRGO2 signal triggers the BLANK window enable control<br>0: Disabled<br>1: Enabled               |
| 0         | ATIMTRGO     | RW         | ATIM TRGO signal triggers the BLANK window enable control<br>0: Disabled<br>1: Enabled                |





### 27.7.6 VCx\_SR Status register

Address offset: 0x10      Reset value: 0x0000 0000

| Bit field | Name | Permission | Function description                                                                                                                    |
|-----------|------|------------|-----------------------------------------------------------------------------------------------------------------------------------------|
| 31:2      | RFU  | -          | Reserved bits, please keep the default value.                                                                                           |
| 1         | FLTV | RO         | Level value of digital filter output<br>0: Digital filter output low level.<br>1: Digital filter output high level.                     |
| 0         | INTF | RW0        | Interrupt flag<br>R0: No VC interrupt occurred.<br>R1: VC interrupt has occurred.<br>W0: VC interrupt flag cleared.<br>W1: No function. |



## 28 Low voltage detector (LVD)

### 28.1 Overview

Low voltage detector (LVD) is used to monitor VDDA power supply voltage or external pin input voltage. When the comparison result between the monitored voltage and the LVD threshold meets the trigger condition, an LVD interrupt or reset signal will be generated, which is usually used to handle some urgent tasks .

The interrupt and reset flags generated by the LVD can only be cleared by software; only after the interrupt or reset flag is cleared and the trigger condition is reached again, the LVD can generate an interrupt or reset signal again.



## 28.2 Main features

- 2-channel monitoring voltage source: VDDA power supply voltage, PA00 pin input.
- 8-step threshold voltage, range 1.8V~4.6V.
- 3 trigger conditions, which can be used in combination:
  1. Level trigger: voltage below threshold.
  2. Falling edge trigger: The falling edge when the voltage drops below the threshold.
  3. Rising edge trigger: The rising edge when the voltage rises above the threshold.
- Can trigger to generate interrupt or reset signal, both cannot be generated at the same time.
- Programmable filter and filter time.
- Supports hysteresis function.
- Supports running in low power mode, interrupt wake-up MCU.

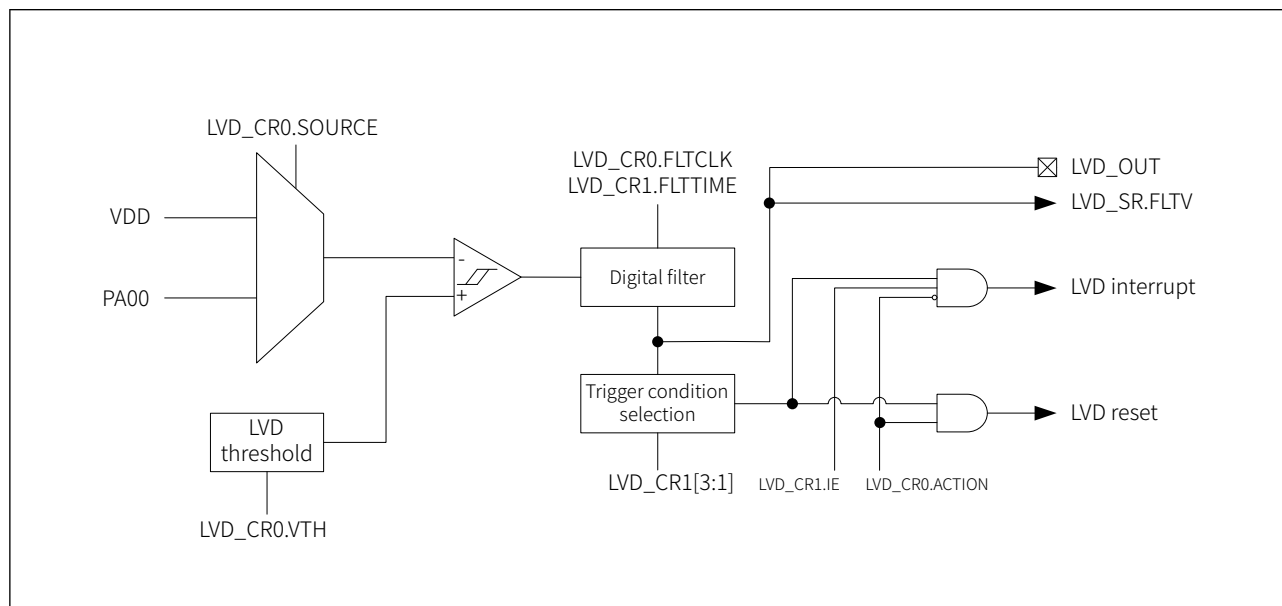


## 28.3 Functional description

### 28.3.1 Functional block diagram

The functional block diagram of the Low voltage detector (LVD) is shown in the following figure:

Figure 28-1 LVD functional block diagram



The LVD can monitor the VDDA power supply voltage, as well as the external pin input voltage (PA00), which is selected by the SOURCE bit field of the control register LVD\_CR0. When using external analog signal input, the user must configure the corresponding GPIO port as analog function (GPIOx\_ANALOG.PINy = 1).

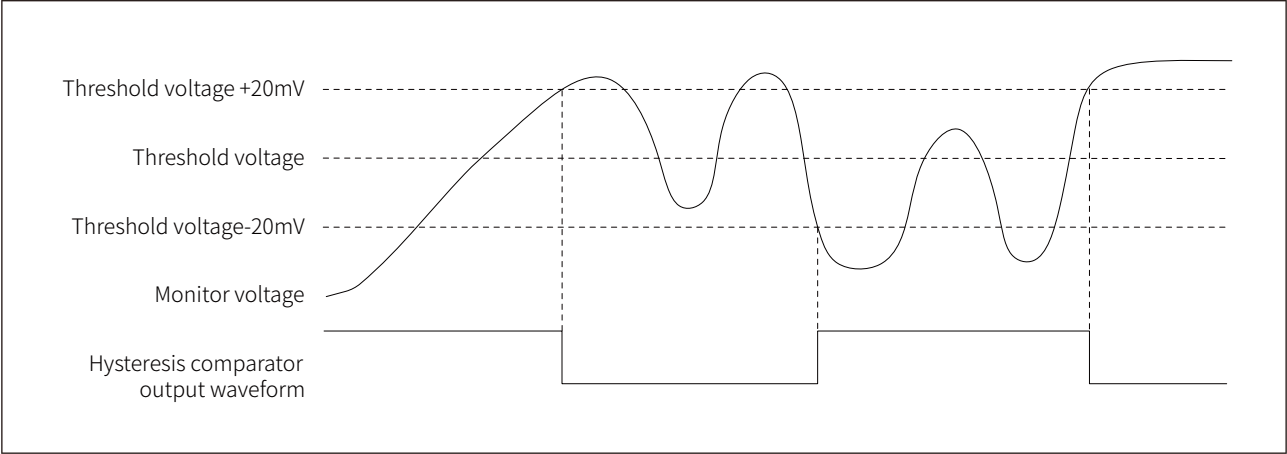
The LVD output result can be output from the PA08/PF03 pin, the user must configure the corresponding GPIO port as a digital output, and select function multiplexing at the same time.

28.3.2 Hysteresis function

The built-in voltage comparator of the LVD has a hysteresis function, which can avoid frequent flipping of the output result of the voltage comparator when the monitored voltage of the LVD is near the threshold voltage, and enhance the anti-interference ability of the system.

The comparator output signal toggles only when the monitored voltage is 20mV above or below the threshold voltage. The specific waveform is shown in the following figure:

Figure 28-2 LVD hysteretic response



The threshold voltage of LVD is determined by the VTH bit value of the control register LVD\_CR0, and the effective value is 0 ~ 7, as shown in the following table:

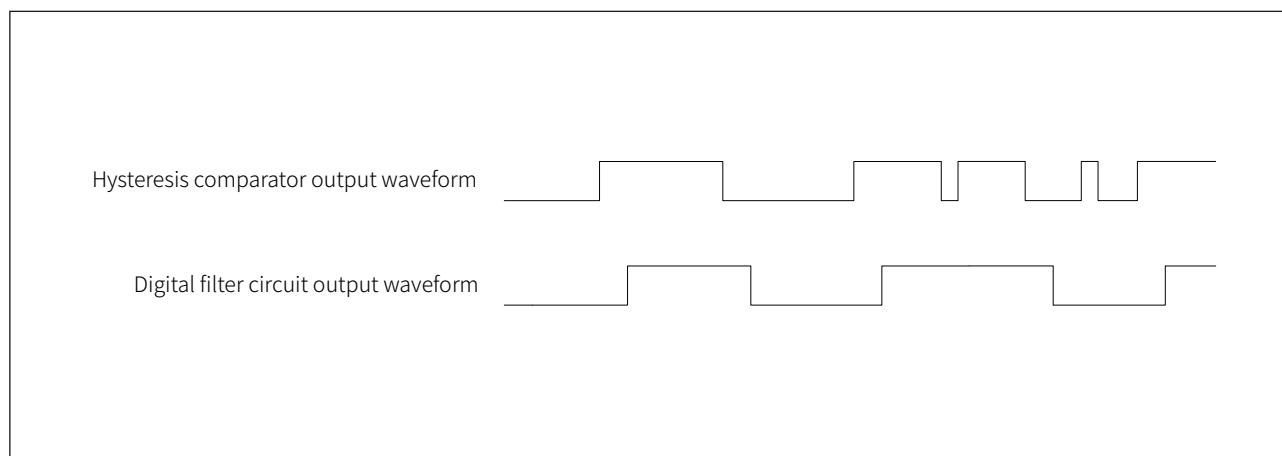
Table 28-1 LVD threshold voltage

| LVD_CR0.VTH | Threshold voltage<br>(unit: V) | LVD_CR0.VTH | Threshold voltage<br>(unit: V) |
|-------------|--------------------------------|-------------|--------------------------------|
| 000         | 1.8                            | 100         | 3.4                            |
| 001         | 2.2                            | 101         | 3.8                            |
| 010         | 2.6                            | 110         | 4.2                            |
| 011         | 3.0                            | 111         | 4.6                            |

### 28.3.3 Digital filter

In order to enhance the robustness of the system, the LVD supports the digital filtering function, which can digitally filter the output result signal of the LVD voltage comparison, and the signal smaller than the filtering width is filtered out without triggering an interrupt or reset. The specific waveform is shown in the following figure:

Figure 28-3 LVD filter output



The FLTCLK bit field of the control register LVD\_CR0 is used to select the clock for digital filter:

- FLTCLK bit is 1, SYSCLK is selected as filter clock.
- FLTCLK bit is 0, and the built-in low speed RC oscillator clock LSI is selected as the filter clock.

The filter samples the output signal with a certain sampling frequency. When the continuous sampling reaches N same levels, the signal is valid, otherwise the signal is invalid, so as to filter out high-frequency clutter signals. The sampling clock of the filter unit is FLTCLK or the frequency division of FLTCLK, and the frequency division ratio of FLTCLK and the number of sampling points N can be selected through the FLTIME bit field of LVD\_CR1 register.

From the FLTV bit field of the LVD status register LVD\_SR, the signal level after LVD digital filter can be read out; When the function of GPIO is multiplexed as LVD\_OUT, the digitally filtered signal can be output from GPIO to facilitate observation and measurement.

## 28.4 LVD interrupts

LVD supports working in low power mode, and interrupt output can wake up the chip from low power mode.

When the comparison result of the monitored voltage and the LVD threshold meets the trigger condition, an interrupt or reset signal can be generated. Whether to generate an interrupt or a reset signal is controlled by the ACTION bit field of the control register LVD\_CR0:

- When ACTION is 1, LVD triggers a reset.
- When ACTION is 0, LVD triggers an interrupt.

Set the IE bit field of the control register LVD\_CR1 to 1 to enable the LVD interrupt. When the trigger condition is met, an LVD interrupt will be generated. The interrupt flag bit LVD\_SR.INTF will be set to 1 by hardware. The user can write 0 to the INTF bit to clear the interrupt flag.

Set the LEVEL, FALL, and RISE bit fields of the control register LVD\_CR1 to select different interrupt or reset trigger methods. The three can be used in combination:

- LEVEL is 1, when the monitored voltage is lower than the threshold, an interrupt is triggered or a reset is generated.
- FALL is 1, the falling edge of the monitored voltage falls below the threshold to triggers an interrupt or generates a reset.
- RISE is 1, the rising edge of the monitored voltage rises above the threshold to trigger an interrupt or generate a reset.



## 28.5 Programming examples

### 28.5.1 Brown-out reset programming example

In this example, the MCU is reset when the monitored voltage falls below the threshold voltage, using digital filtering.

The configuration method is as follows:

- Step 1:** Configure LVD\_CR0.SOURCE and select the voltage source to be monitored.
- Step 2:** Configure LVD\_CR0.VTH and set the threshold voltage.
- Step 3:** Configure LVD\_CR1.FLTTIME and select LVD filter time.
- Step 4:** Configure LVD\_CR0.FLTCLK and select the filter clock.
- Step 5:** Set LVD\_CR1.LEVEL to 1, and select to trigger LVD action when the monitored voltage is lower than the threshold.
- Step 6:** Set LVD\_CR0.ACTION to 1, and select LVD trigger action as system reset.
- Step 7:** Set LVD\_CR0.EN to 1 to enable LVD.

### 28.5.2 Interrupt programming example

In this example, an interrupt is generated when the monitored voltage is above or below a threshold voltage, using digital filtering.

The configuration method is as follows:

- Step 1:** Configure LVD\_CR0.SOURCE and select the voltage source to be monitored.
- Step 2:** Configure LVD\_CR0.VTH and select the threshold voltage.
- Step 3:** Configure LVD\_CR1.FLTTIME and select LVD filter time.
- Step 4:** Configure LVD\_CR0.FLTCLK and select the filter clock.
- Step 5:** Set both LVD\_CR1.RISE and FALL to 1, and select rising and falling edge triggers.
- Step 6:** Set LVD\_CR0.ACTION to 0, and select LVD trigger action as interrupt.
- Step 7:** Set LVD\_CR1.IE to 1 to enable LVD interrupt.
- Step 8:** Enable the LVD interrupt in the NVIC interrupt vector table.
- Step 9:** Set LVD\_CR0.EN to 1 to enable LVD.
- Step 10:** In the LVD initialization routine and the LVD interrupt service routine, write 0 to the LVD\_ISR.INTF bit to clear the interrupt flag and allow a new LVD interrupt to be generated.





## 28.6 List of registers

LVD base address: LVD\_BASE = 0x4000 00A4

Table 28-2 List of LVD registers

| Register name | Register address | Register description |
|---------------|------------------|----------------------|
| LVD_CR0       | LVD_BASE + 0x00  | Control register 0   |
| LVD_CR1       | LVD_BASE + 0x04  | Control register 1   |
| LVD_SR        | LVD_BASE + 0x08  | Status register      |



## 28.7 Register description

See section 1 [Documentation conventions](#) for abbreviations used in register descriptions.

### 28.7.1 LVD\_CR0 Control register 0

Address offset: 0x00      Reset value: 0x0000 0100

| Bit field | Name   | Permission | Function description                                                                                                                                                                                                                                                                                                                                 |
|-----------|--------|------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:9      | RFU    | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                                        |
| 8         | FLTCLK | RW         | Digital filter module filter clock setting<br>0: LSI<br>1: SYSCLK<br><i>Note: Before entering DeepSleep mode, when the digital filter function is not required, please set the filter clock source to SYSCLK, otherwise the LSI will generate power consumption when running under DeepSleep mode (Default value has been configured as SYSCLK).</i> |
| 7         | RFU    | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                                        |
| 6:4       | VTH    | RW         | Threshold voltage selection (Based on $V_{core} = 1.6V$ )<br>000: 1.8V<br>001: 2.2V<br>010: 2.6V<br>011: 3.0V<br>100: 3.4V<br>101: 3.8V<br>110: 4.2V<br>111: 4.6V                                                                                                                                                                                    |
| 3         | RFU    | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                                        |
| 2         | SOURCE | RW         | Monitor source configuration<br>0: VDDA supply voltage<br>1: PA00 port input voltage                                                                                                                                                                                                                                                                 |
| 1         | ACTION | RW         | Trigger action configuration<br>0: NVIC interrupts.<br>1: System resets.                                                                                                                                                                                                                                                                             |
| 0         | EN     | RW         | Enable control<br>0: LVD disabled.<br>1: LVD enabled.                                                                                                                                                                                                                                                                                                |



### 28.7.2 LVD\_CR1 Control register 1

Address offset: 0x04      Reset value: 0x0000 0000

| Bit field | Name    | Permission | Function description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|-----------|---------|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:8      | RFU     | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 7:4       | FLTTIME | RW         | Digital filter module filter time configuration<br>0000: No filter      1000: $f_{\text{SAMPLING}} = f_{\text{FLTCLK}}/8, N = 6$<br>0001: $f_{\text{SAMPLING}} = f_{\text{FLTCLK}}/1, N = 2$ 1001: $f_{\text{SAMPLING}} = f_{\text{FLTCLK}}/8, N = 8$<br>0010: $f_{\text{SAMPLING}} = f_{\text{FLTCLK}}/1, N = 4$ 1010: $f_{\text{SAMPLING}} = f_{\text{FLTCLK}}/16, N = 5$<br>0011: $f_{\text{SAMPLING}} = f_{\text{FLTCLK}}/1, N = 8$ 1011: $f_{\text{SAMPLING}} = f_{\text{FLTCLK}}/16, N = 6$<br>0100: $f_{\text{SAMPLING}} = f_{\text{FLTCLK}}/2, N = 6$ 1100: $f_{\text{SAMPLING}} = f_{\text{FLTCLK}}/16, N = 8$<br>0101: $f_{\text{SAMPLING}} = f_{\text{FLTCLK}}/2, N = 8$ 1101: $f_{\text{SAMPLING}} = f_{\text{FLTCLK}}/32, N = 5$<br>0110: $f_{\text{SAMPLING}} = f_{\text{FLTCLK}}/4, N = 6$ 1110: $f_{\text{SAMPLING}} = f_{\text{FLTCLK}}/32, N = 6$<br>0111: $f_{\text{SAMPLING}} = f_{\text{FLTCLK}}/4, N = 8$ 1111: $f_{\text{SAMPLING}} = f_{\text{FLTCLK}}/32, N = 8$ |
| 3         | FALL    | RW         | Triggered by the falling edge of the monitored voltage falling below the threshold (i.e., triggered by the rising edge of the LVD output signal)<br>0: Disabled<br>1: Enabled                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 2         | RISE    | RW         | Triggered by the rising edge of the monitored voltage rising above the threshold (i.e., triggered by the falling edge of the LVD output signal)<br>0: Disabled<br>1: Enabled                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 1         | LEVEL   | RW         | Triggered when the monitored voltage is lower than the threshold voltage (i.e., triggered by a high-level LVD output signal)<br>0: Disabled<br>1: Enabled                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 0         | IE      | RW         | LVD interrupt enable control<br>0: Disabled<br>1: Enabled                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |



### 28.7.3 LVD\_SR Status register

Address offset: 0x08      Reset value: 0x0000 0000

| Bit field | Name | Permission | Function description                                                                                                                          |
|-----------|------|------------|-----------------------------------------------------------------------------------------------------------------------------------------------|
| 31:2      | RFU  | -          | Reserved bits, please keep the default value.                                                                                                 |
| 1         | FLTV | RO         | Level value of digital filter output<br>0: Digital filter output low level.<br>1: Digital filter output high level.                           |
| 0         | INTF | RW0        | Interrupt flags<br>R0: LVD interrupt did not occur.<br>R1: LVD interrupt has occurred.<br>W0: LVD interrupt flag cleared.<br>W1: No function. |



## 29 Operational amplifier (OPA)

### 29.1 Overview

CW32L012 integrates 2 operational amplifiers (OPAs) internally, which, when combined with external circuitry, can perform various functions of a general-purpose op-amp. The positive input of the OPA supports up to three external analog inputs and one internal DAC, while the negative input supports 2 external analog inputs. OPA has a calibration function and can be triggered to start OPA calibration in multiple ways.



## 29.2 Main features

- 2 independent operational amplifiers.
- 4 positive-terminal analog inputs: OPAX\_INP1~4, where INP4 connects to the internal DAC.
- 2 negative-terminal analog inputs: OPAX\_INN1~2.
- 1 analog output: OPAX\_OUT.
- Supports multiple methods to trigger the calibration function.
- Supports operation in DeepSleep mode.

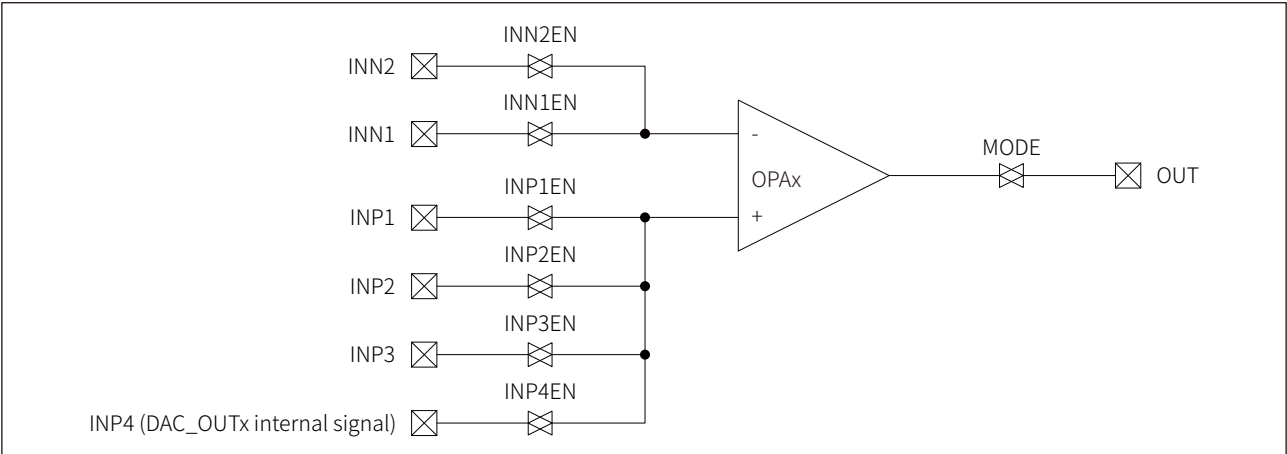


## 29.3 Functional description

### 29.3.1 Functional block diagram

The operational amplifier (OPA) supports flexible input and output port configurations. Whether to connect to the OPA can be selected through program-controlled switches. For specific register configurations, please refer to the control register OPAx\_CR. The OPA is enabled by setting the EN bit field of the OPAx\_CR register to 1, allowing it to perform various functions of a general-purpose operational amplifier in conjunction with external circuitry. The functional block diagram of the OPA is shown in the following figure:

Figure 29-1 OPA functional block diagram



### 29.3.2 Input/output pins

Both the input and output ports of the OPA must be configured for analog functionality. The flexible port configuration is shown in the following table:

Table 29-1 OPA port configuration

| OPA1 input/output | GPIO | OPA2 input/output | GPIO |
|-------------------|------|-------------------|------|
| INP1              | PA03 | INP1              | PA04 |
| INP2              | PA06 | INP2              | PA06 |
| INP3              | PB01 | INP3              | PB00 |
| INN1              | PA04 | INN1              | PA05 |
| INN2              | PA07 | INN2              | PA07 |
| OUT               | PB00 | OUT               | PB01 |

### 29.3.3 Calibration function

OPA supports calibration functionality. The calibration function is enabled by setting the CALEN bit field of the calibration register OPAX\_CAL to 1. The calibration methods support level calibration and trigger calibration.

In the level calibration mode, setting the START bit to 1 starts the calibration, and clearing the START bit to zero stops the calibration.

In the trigger calibration mode, users can flexibly configure multiple hardware trigger sources to start calibration, or start calibration through software by setting the SOFTTRIG bit to 1. The duration of the calibration process can be configured through the CALPERIOD bit field.





## 29.4 Programming examples

The following is an example of the basic configuration of OPA:

- Step 1:** Set SYSCTRL\_AHBEN.GPIOx to 1 and SYSCTRL\_APBEN2.OPA to 1 to 1 to enable the GPIO clock and OPA clock.
- Step 2:** Configure the GPIO pins corresponding to the OPA channels for analog functionality. For details about register configuration please see the [9 General-purpose input/output \(GPIO\)](#).
- Step 3:** Set OPAX\_CR.INPyEN (y = 1~4) to 1 to enable the positive terminal analog input signal INPy.
- Step 4:** Set OPAX\_CR.INNyEN (y = 1~2) to 1 to enable the negative terminal analog input signal INNy.
- Step 5:** Set OPAX\_CR.MODE to 0x to select external amplification mode and enable the analog output OUT.
- Step 6:** Set CW\_BGR->CR\_f.BGREN to 1 to enable the internal BGR module and wait for stabilization.
- Step 7:** Set OPAX\_CR.EN to 1 to enable the OPA.
- Step 8:** Set OPAX\_CAL.CALMODE to 1 to select level calibration.
- Step 9:** Set OPAX\_CAL.CALEN to 1 to enable the calibration function.
- Step 10:** Set OPAX\_CAL.START to 1 to initiate level calibration.
- Step 11:** Wait for the specified calibration duration.
- Step 12:** Set OPAX\_CAL.START to 0 to stop level calibration.
- Step 13:** The output signal of OPA can be sampled through ADC.



## 29.5 List of registers

OPA1 base address: OPA1\_BASE = 0x4000 00B0

OPA2 base address: OPA2\_BASE = 0x4000 00B8

Table 29-2 List of OPA registers

| Register name | Register address | Register description |
|---------------|------------------|----------------------|
| OPAx_CR       | OPAx_BASE + 0x00 | Control register     |
| OPAx_CAL      | OPAx_BASE + 0x04 | Calibration register |

See section [1 Documentation conventions](#) for abbreviations used in register descriptions.

Address offset: 0x00      Reset value: 0x0000 E000

| Bit field | Name   | Permission | Function description                                                                                                                                                                                            |
|-----------|--------|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:16     | RFU    | -          | Reserved bits, please keep the default value.                                                                                                                                                                   |
| 15:13     | BIAS   | RW         | OPAx bias current selection<br>000: 1μA                      100: 5μA<br>001: 2μA                      101: 6μA<br>010: 3μA                      110: 7μA<br>011: 4μA                      111: 8μA             |
| 12:10     | AMP    | RW         | PGA positive gain configuration<br>000: 64/32<br>001: 64/16<br>010: 64/8<br>011: 64/4<br>100: 64/2                                                                                                              |
| 9         | INN2EN | RW         | OPAx negative terminal input signal 2 conduction enable control<br>0: Disabled<br>1: Enabled                                                                                                                    |
| 8         | INN1EN | RW         | OPAx negative terminal input signal 1 conduction enable control<br>0: Disabled<br>1: Enabled                                                                                                                    |
| 7         | INP4EN | RW         | OPA1 positive terminal input signal 4 conduction enable control (internal DAC1_OUT_INT)<br>OPA2 positive terminal input signal 4 conduction enable control (internal DAC2_OUT_INT)<br>0: Disabled<br>1: Enabled |
| 6         | INP3EN | RW         | OPAx positive terminal input signal 3 conduction enable control<br>0: Disabled<br>1: Enabled                                                                                                                    |
| 5         | INP2EN | RW         | OPAx positive terminal input signal 2 conduction enable control<br>0: Disabled<br>1: Enabled                                                                                                                    |
| 4         | INP1EN | RW         | OPAx positive terminal input signal 1 conduction enable control<br>0: Disabled<br>1: Enabled                                                                                                                    |
| 3         | RFU    | -          | Reserved bits, please keep the default value.                                                                                                                                                                   |

| Bit field | Name | Permission | Function description                                                                                                                                                                        |
|-----------|------|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 2:1       | MODE | RW         | OPAx operation mode configuration<br>0x: External amplification mode, controlled by feedback network.<br>10: Internal PGA mode, controlled by AMP bit field.<br>11: Internal follower mode. |
| 0         | EN   | RW         | OPAx enable control<br>0: Disabled<br>1: Enabled                                                                                                                                            |

### 29.6.2 OPAx\_CAL Calibration register

Address offset: 0x04      Reset value: 0x0000 0000

| Bit field | Name      | Permission | Function description                                                                              |
|-----------|-----------|------------|---------------------------------------------------------------------------------------------------|
| 31        | GTIM4OC2  | RW         | GTIM4 OC2REFC rising edge signal triggers calibration enable control<br>0: Disabled<br>1: Enabled |
| 30        | GTIM4OC1  | RW         | GTIM4 OC2REFC rising edge signal triggers calibration enable control<br>0: Disabled<br>1: Enabled |
| 29        | GTIM4TRGO | RW         | GTIM4 TRGO rising edge signal triggers calibration enable control<br>0: Disabled<br>1: Enabled    |
| 28        | GTIM3OC2  | RW         | GTIM3 OC2REFC rising edge signal triggers calibration enable control<br>0: Disabled<br>1: Enabled |
| 27        | GTIM3OC1  | RW         | GTIM3 OC2REFC rising edge signal triggers calibration enable control<br>0: Disabled<br>1: Enabled |
| 26        | GTIM3TRGO | RW         | GTIM3 TRGO rising edge signal triggers calibration enable control<br>0: Disabled<br>1: Enabled    |
| 25        | GTIM2OC2  | RW         | GTIM2 OC2REFC rising edge signal triggers calibration enable control<br>0: Disabled<br>1: Enabled |



| Bit field | Name       | Permission | Function description                                                                              |
|-----------|------------|------------|---------------------------------------------------------------------------------------------------|
| 24        | GTIM2OC1   | RW         | GTIM2 OC2REFC rising edge signal triggers calibration enable control<br>0: Disabled<br>1: Enabled |
| 23        | GTIM2TRGO  | RW         | GTIM2 TRGO rising edge signal triggers calibration enable control<br>0: Disabled<br>1: Enabled    |
| 22        | GTIM1OC2   | RW         | GTIM1 OC2REFC rising edge signal triggers calibration enable control<br>0: Disabled<br>1: Enabled |
| 21        | GTIM1OC1   | RW         | GTIM1 OC2REFC rising edge signal triggers calibration enable control<br>0: Disabled<br>1: Enabled |
| 20        | GTIM1TRGO  | RW         | GTIM1 TRGO rising edge signal triggers calibration enable control<br>0: Disabled<br>1: Enabled    |
| 19        | ATIM1OC6   | RW         | ATIM1 OC6REFC rising edge signal triggers calibration enable control<br>0: Disabled<br>1: Enabled |
| 18        | ATIM1OC5   | RW         | ATIM1 OC5REFC rising edge signal triggers calibration enable control<br>0: Disabled<br>1: Enabled |
| 17        | ATIM1OC4   | RW         | ATIM1 OC4REFC rising edge signal triggers calibration enable control<br>0: Disabled<br>1: Enabled |
| 16        | ATIM1TRGO  | RW         | ATIM1TRGO rising edge signal triggers calibration enable control<br>0: Disabled<br>1: Enabled     |
| 15        | ADC2START1 | RW         | ADC2 START1 signal triggers calibration enable control<br>0: Disabled<br>1: Enabled               |
| 14        | ADC1START1 | RW         | ADC1 START1 signal triggers calibration enable control<br>0: Disabled<br>1: Enabled               |



| Bit field | Name      | Permission | Function description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|-----------|-----------|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 13:11     | CLKDIV    | RW         | OPACLK source configuration<br>000: PCLK/1                      100: PCLK/16<br>001: PCLK/2                    101: PCLK/32<br>010: PCLK/4                    110: PCLK/64<br>011: PCLK/8                    111: PCLK/128                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 10        | AZRUN     | RO         | Calibration running flag<br>1: Calibration is running<br>0: Calibration has completed<br><i>Attention: This flag is only applicable for indicating the completion of automatic calibration triggered by software or hardware; it does not function for level calibration. The flag remains set to 1 for twice the duration of PERIOD.</i>                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| 9:8       | RFU       | -          | Reserved bits, please keep the default value.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| 7:4       | CALPERIOD | RW         | Trigger calibration process duration setting<br>0000: 8 OPACLK cycles              1000: 2048 OPACLK cycles<br>0001: 16 OPACLK cycles            1001: 4096 OPACLK cycles<br>0010: 32 OPACLK cycles            1010: 8192 OPACLK cycles<br>0011: 64 OPACLK cycles            1011: 16384 OPACLK cycles<br>0100: 128 OPACLK cycles           1100: 32768 OPACLK cycles<br>0101: 256 OPACLK cycles           1101: 65536 OPACLK cycles<br>0110: 512 OPACLK cycles           1110: 131072 OPACLK cycles<br>0111: 1024 OPACLK cycles          1111: 262144 OPACLK cycles<br><i>Note: This duration is only applicable to the delay for automatic calibration triggered by software or hardware. If level calibration is selected, the calibration duration needs to be manually configured.</i> |
| 3         | SOFTTRIG  | RW         | Software-triggered calibration enable<br>W0: No function.<br>W1: Software-triggered calibration.<br><i>Note: The effect of software triggering is the same as that of hardware triggering. The software writes 1 to start triggering, and the duration of trigger calibration is determined by CALPERIOD.</i>                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| 2         | START     | RW         | Level calibration enable control<br>0: Stop level calibration.<br>1: Start level calibration.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| 1         | CALMODE   | RW         | Calibration mode configuration<br>0: Trigger calibration: Calibration is initiated by a hardware or software trigger signal.<br>1: Level calibration: Set START to 1 to start calibration, and clear START to 0 to stop calibration.<br><i>Note: The duration of triggering the calibration process is controlled by the CALPERIOD bit field.</i>                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 0         | CALEN     | RW         | Calibration function enable control<br>0: Disabled<br>1: Enabled                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |



## 30 Debug interface (DBG)

### 30.1 Overview

The core of CW32L012 is ARM® Cortex®-M0+, the core has built-in DAP hardware debug module, and supports SWD mode debugging. The hardware debug module can be suspended when fetching instructions (instruction breakpoints) or accessing data (data breakpoints), and the program stops running. query; and the core and peripherals can be restored, and program execution continues.

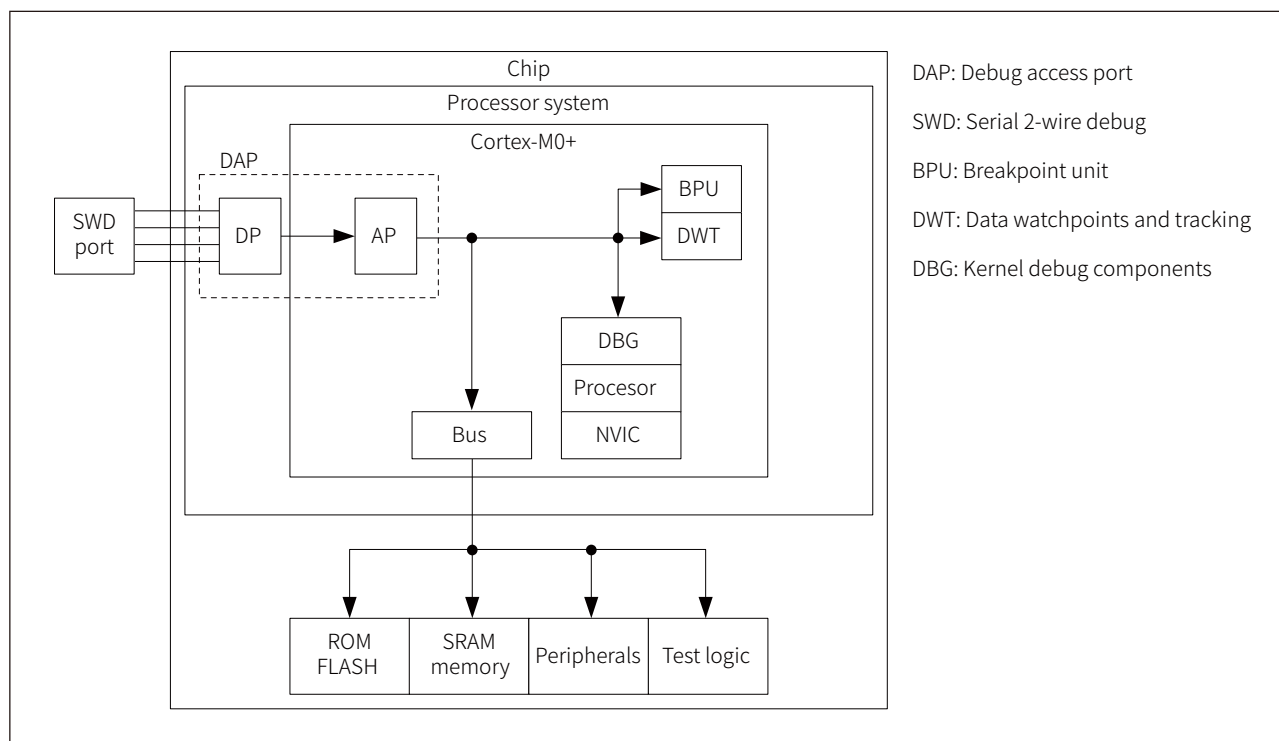
When using the debugging simulation tool to connect to the CW32L012 through the SWD interface, enter the debugging mode, and perform debugging operations through the DAP hardware debugging module in the chip core.



## 30.2 Serial wire debug port SWD

Use the SWD interface of the CW special debugger or the general debugging simulation tool to connect with the DAP debugging module inside the target chip, and carry out data exchange through the packet transmission protocol to realize the debugging operation. The SWD mode is a 2-wire serial communication, including a clock line SWCLK and a bidirectional data line SWDIO. As shown in the following figure:

Figure 30-1 SWD functional block diagram



The pin assignment of the SWD interface of the CW32L012 series is shown in the following table. The PA13/PA14 pin is the SWD function by default when the chip leaves the factory.

Table 30-1 SWD pin assignment

| SWD port name | Pin function                 | Pin assignment |
|---------------|------------------------------|----------------|
| SWCLK         | Serial clock input           | PA14           |
| SWDIO         | Serial data input and output | PA13           |

The PA13/PA14 pins can be configured as SWD function or GPIO/ISP function, and the function configuration is performed by the SWDIO bit field of the system control register SYSCTRL\_CR2. When SWDIO is 0, the PA13/PA14 pin is configured as SWD function; when SWDIO is 1, it is configured as GPIO/ISP function.

It is generally recommended to use a 10KΩ pull-up resistor for the SWD pin. When the PA13/PA14 of the CW32L012 is used as the SWD function, there is a built-in pull-up resistor with a resistance value between 55kΩ ~ 70kΩ. Users can add a pull-up resistor externally to improve the anti-interference performance.

The PA13/PA14 pin of the CW32L012 series is the SWD function by default. If the user has set the encryption level, it is necessary to judge whether the SWD function is supported according to the set level. The configuration and function of the SWD pin are shown in the following table:



Table 30-2 SWD pin function

| Encryption level    | SYSCTRL_CR2.SWDIO | PA13/PA14 function |
|---------------------|-------------------|--------------------|
| Level0/1 encryption | 0 (default)       | SWD function       |
|                     | 1                 | GPIO/ISP function  |
| Level2 encryption   | 0 (default)       | NA, no function    |
|                     | 1                 | GPIO/ISP function  |
| Level3 encryption   | 0 (default)       | NA, no function    |
|                     | 1                 | GPIO function      |

*Attention: The encryption level can be set through the ISP communication method. For details, please refer to the ISP communication protocol document.*

*Attention: After the chip is powered on, both SWDIO and SWCLK are internal pull-ups by default, and users do not need to connect external pull-up resistors.*

*Attention: When the chip encryption level is set to 2, the SWD function is disabled and can only be programmed by ISP.*

*Attention: When the chip encryption level is set to 3, the SWD function and the ISP function are disabled, and the chip cannot be programmed with new programs again.*



### 30.3 Debug communication protocol

The debugger and the DAP debug module of the target chip communicate through the SWD packet transmission protocol. The packet transmission protocol is a 2-wire synchronous serial protocol, using the SWCLK clock signal and the SWDIO data signal:

- SWCLK is a one-way clock signal, which is output by the debugger to the target chip.
- SWDIO is a bidirectional data signal, which is driven by the debugger and the target chip bidirectionally time-sharing.

The protocol defines the conversion time of the transceiver and the transceiver with a length of one SWCLK cycle. During the conversion time of the transceiver and the target chip, neither the debugger nor the target chip drives SWDIO, and SWDIO is pulled up to a high level by a pull-up resistor.

When transmitting data on the SWDIO signal line, follow the principle that the lowest LSB is transmitted first, and the highest MSB is transmitted last.

Through the packet transfer protocol, the debugger can read and write access to the DP register and AP register in the DAP debug module of the target chip, hereinafter referred to as SW-DP and SW-AP.

#### 30.3.1 Transmission protocol format

The DAP debug module includes the DP debug port register and the AP access port register. The communication between the debugger and the DAP debug module of the target chip is actually the read and write operations on the DP register and the AP register.

The transmission communication frame usually contains three fields:

- Package request: The length is 8-bit, the debugger to the target chip.
- Response: The length is 3-bit; the target chip to the debugger
- Data transmission: The length is 33-bit, the target chip to the debugger or the debugger to the target chip, optional field.

The function definition of each request package is as follows:

Table 30-3 Definition of packet request bits

| Bit | Name         | Description                                                      |
|-----|--------------|------------------------------------------------------------------|
| 0   | Start up     | Must be 1.                                                       |
| 1   | APnDP        | 0: DP access; 1: AP access.                                      |
| 2   | RnW          | 0: Write request; 1: Read request.                               |
| 3:4 | A[3:2]       | Address field of DP or AP register, bit3 is A2, bit4 is A3.      |
| 5   | Parity check | Bit parity bit of the previous 5-bit.                            |
| 6   | Stop         | 0                                                                |
| 7   | Reside       | Not driven by the master. Target chip reads as 1 due to pull-up. |

The request packet is always followed by the conversion time of the transceiver (default 1bit). At this time, neither the master nor the target drives SWDIO.

The response bits transmitted by the target are defined as follows:



Table 30-4 Definition of target transmit response bits

| Bit | Name | Description                                                                                                   |
|-----|------|---------------------------------------------------------------------------------------------------------------|
| 0:2 | ACK  | 001: FAULT (bit0:0, bit1:0, bit2:1)<br>010: WAIT (bit0:0, bit1:1, bit2:0)<br>100: OK (bit0:1, bit1:0, bit2:0) |

After the ACK response is the conversion time of the transceiver (default 1bit), at this time, neither the master nor the target will drive SWDIO.

Data transmitted by the debugger or target chip, the bits are defined as follows:

Table 30-5 Definition of data transmission bits

| Bit  | Name           | Description                    |
|------|----------------|--------------------------------|
| 0:31 | WDATA or RDATA | Write or read data.            |
| 32   | Parity check   | Bit parity bit of 32-bit data. |

The data transmission phase is not necessary, and data transmission occurs only in the following two cases:

- The response received after the data read or write request is the OK response.
- The ORUNDETECT flag bit of the DP-CTRL/STAT register is set to 1. At this time, no matter what kind of response (including WAIT and FAULT) is received, data transmission is required.

### 30.3.2 SW-DP state machine

There is an ID CODE register inside SW-DP with a fixed value of 0x0BC11477 (Cortex®-M0+ identification code of ARM company). Before using the SWD packet protocol to read and write the target register, the ID number must be read to activate the SW-DP logic of the target chip, otherwise the SW-DP state machine of the target chip will not work.

The operation sequence is as follows:

- Step 1:** After power-on reset or after the SWDIO line is at a high level for more than 50 cycles, the SW-DP state machine enters the reset state.
- Step 2:** After entering the reset state, keep the SWDIO line at a low level for at least 2 cycles, and the SW-DP state machine enters the idle state.
- Step 3:** After entering the idle state, perform read access to the ID CODE register of DP-SW.  
*Attention: If this operation is not performed, the target board will send a FAULT response in the subsequent packet communication response phase.*
- Step 4:** Read and write access to the registers to be accessed according to the packet protocol.



### 30.3.3 Read and write access for SW-DP and SW-AP

- Write access for SW-DP/SW-DP: After the debugger receives the ACK, the data must be transmitted immediately after the conversion time of 1 clock cycle.
- Read access for SW-DP:
  - When the target chip is ready to transmit data, it transmits an OK response, and then transmits data immediately.
  - When the target chip DAP is not ready for data transmission, it transmits a WAIT response to end the communication.
  - When the DAP status of the target chip is wrong, a FAULT response is transmitted to end the communication.
- Read access for SW-AP: The read access to the AP is registered, that is, the read AP operation cannot return the required result, but the result of the read operation cannot be returned until the next AP operation. If the next access to be performed is not an AP read access operation, the DP-RDBUFF register must be read to obtain the result of this read operation.
- The READOK flag in the DP-CTRL/STAT register of the M0 core's debug control/status register is updated every time a SW-AP read access or DP-RDBUFF read request is made to indicate whether the AP read access is successful.
- SW-DP has write buffers (for SW-DP or SW-AP writes) that can accept the next write operation while the read and write operations are not complete. If the SW-DP's write buffer is full, the chip SW-DP logic will reply with a WAIT response to notify the master to suspend the operation. The exception is special operations such as IDCODE read, DP-CTRL/STAT read or ABORT write operations, which are also accepted when the write buffer is full.
- Since SWCLK and HCLK are not synchronous clocks and belong to the asynchronous clock domain, two additional SWCLK cycles are required after the write operation (after the parity bit of the data transmission) to ensure that the write operation takes effect. The SWDIO line should be driven low (idle state) while the master is driving these 2 SWCLK cycles. This is especially important when writing power-on request operations to the DP-CTRL/STAT registers, otherwise the next operation (one that is valid after the core is powered on) will be executed immediately and will fail.



### 30.3.4 SW-DP register

When APnDP is 0, access the DP register, which is addressed by A[3:2]. When the address is the same, the meaning of the register may be different due to different read and write operations, as shown in the following table:

Table 30-6 List and definition of DP registers

| A[3:2] | Read/write | Register name                        | Register content description                                                                                                                                                                                                                                                                           |
|--------|------------|--------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 00     | Read       | IDCODE                               | Fixed at 0x0BC1 1477 (to identify SW-DP), ARM Cortex® -M0+ identification code                                                                                                                                                                                                                         |
|        | Write      | ABORT                                | See <a href="#">Table 30-7: Definition of DP ABORT register bits</a> .                                                                                                                                                                                                                                 |
| 01     | Read/Write | DP-CTRL/STAT<br>(SELECT.CTRLSEL = 0) | Mainly used for: <ol style="list-style-type: none"> <li>1. Request system or debug power-on.</li> <li>2. Configure transmit operation for AP access.</li> <li>3. Control comparison and verification operations.</li> <li>4. Read some status flags (overflow and power-up acknowledgment).</li> </ol> |
|        |            | WIRE CONTROL<br>(SELECT.CTRLSEL = 1) | Used to configure the physical serial port protocol (such as the duration of transition times, etc.).                                                                                                                                                                                                  |
| 10     | Read       | READ RESEND                          | Allow recovery of read data from corrupted debug transmits without repeating the original AP transmit.                                                                                                                                                                                                 |
|        | Write      | SELECT                               | Used to select the current access AP port and the 4-word register BANK in the AP port.<br>See <a href="#">Table 30-8: DP SELECT register</a> .                                                                                                                                                         |
| 11     | Read/Write | READ BUFFER                          | Since an AP access has been issued, this read buffer is useful (providing the result of reading this AP request when executing the next AP transaction). This read buffer captures the data in the AP, displayed as the result of the previous read, without starting a new operation.                 |



Table 30-7 Definition of DP ABORT register bits

| Bit  | Name       | Description                                                                                                                                                                                      |
|------|------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:5 | RESERVE    |                                                                                                                                                                                                  |
| 4    | ORUNERRCLR | Write 1 to clear the STICKYORUN error flag.                                                                                                                                                      |
| 3    | WDATAERR   | Write 1 to clear the WDATAERR error flag.                                                                                                                                                        |
| 2    | STICKYERR  | Write 1 to clear the STICKYERR error flag.                                                                                                                                                       |
| 1    | STICKYCMP  | Write 1 to clear the STICKYCMP flag.                                                                                                                                                             |
| 0    | DAPABORT   | Write 1 to generate the DAP ABORT signal and abandon the current access operation; when the target chip responds to the WAIT response, this operation must be performed to abort this operation. |

Table 30-8 DP SELECT register

| Bit   | Name      | Description                                                                                                                                                                                                          |
|-------|-----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:24 | APSEL     | Select the current AP port, fixed as b00000000.                                                                                                                                                                      |
| 23:8  | RESERVE   |                                                                                                                                                                                                                      |
| 7:4   | APBANKSEL | Select the active 4-word register BANK on the current AP.                                                                                                                                                            |
| 3:1   | RESERVE   |                                                                                                                                                                                                                      |
| 0     | CTRLSEL   | DP port register selection: <ul style="list-style-type: none"> <li>• The default is 0 after reset, select the CTRL/STAT register.</li> <li>• Set to 1 to select the WCR register (Wire Control Register).</li> </ul> |



### 30.3.5 SW-AP register

When APnDP is 1, access the AP register. Since there are many SW-AP registers, it is necessary to combine A[3:2] and the APBANKSEL bit field value of the SW-DP selection register SELECT to uniquely address the SW-AP register.

This article will not go into details about SW-AP one by one, but only introduces a common IDR public register (address 0xFC), as shown in the following table:

Table 30-9 AP IDR public register

| Bit   | Definition                                                                                                                                            |
|-------|-------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:28 | AP designed version Rervision.                                                                                                                        |
| 27:24 | AP designer identification code, this chip is fixed to 0x04.                                                                                          |
| 23:17 | AP designer identification code, this chip is fixed to 0x3B.                                                                                          |
| 16:13 | AP type. For example, the memory access port type identification code is b1000, and the undefined type access port type identification code is b0000. |
| 12:8  | Reserved                                                                                                                                              |
| 7:0   | AP ID. Such as MEM-AP or JTAG-AP and so on.                                                                                                           |



## 30.4 Kernel debugging

The core can be debugged by manipulating the core debug registers, and the debugger accesses these registers through DAP debugging.

The kernel debug registers mainly include 4 registers, as shown in the following table:

Table 30-10 Kernel debug registers

| Register | Description                                                                                                                                                                                                                                  |
|----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| DHCSR    | Debug suspend control and status register, 32-bit;<br>Control processor suspend, single step and restart actions.                                                                                                                            |
| DCRSR    | Debug kernel register selector register, 17-bit;<br>Controls reading and writing of kernel registers during suspend.                                                                                                                         |
| DCRDR    | Debug kernel register data register, 32-bit;<br>Data transfer register for reading and writing kernel registers during pause.                                                                                                                |
| DEMCR    | Debug exception monitoring control register, 32-bit;<br>Used to enable data watchpoint unit and vector capture feature, using vector capture, the debugger can suspend the processor when the processor is reset or a hardware error occurs. |

These registers can only be reset by a power-on reset. See *Cortex®-M0+ Technical Reference Manual* for more details.





## 30.5 Breakpoint unit BPU

The Cortex®-M0+ BPU breakpoint unit provides four breakpoint registers to implement the PC pointer-based breakpoint function.

For more information on the identification registers and access types of the BPU CoreSight component, see *ARMv6-M Architecture Reference Manual* and *ARM® CoreSight Component Technical Reference Manual*.



## 30.6 Data observation points and tracking DWT

Cortex®-M0+ DWT provides two observation point register sets. Realize the following functions:

- Set data watchpoints: Data or peripheral addresses can be marked as watch variables, and access to this address will generate a debug event and suspend program execution.
- Optional DWT program counter sampling register (DWT\_PCSR) feature in ARMv6-M. Allows the debugger to periodically sample the PC pointer, providing a rough analysis without stopping the processor.

See *ARMv6-M Architecture Reference Manual* for more information.



## 30.7 Debug components DBG

The debugger implements clock control support for peripherals such as timers, watchdogs, and RTCs during breakpoints through the debug components DBG. Through the setting of the debug state timer control register SYSCTRL\_DEBUG, you can control the timer, watchdog timer, RTC, etc. to run normally or pause during the breakpoint in the debug state, as shown in the following table:

Table 30-11 Debug status timer control register SYSCTRL\_DEBUG

| 位  | Name    | Vaule | Counter function configuration during breakpoints in debug state |
|----|---------|-------|------------------------------------------------------------------|
| 11 | HALLTIM | 1     | In debug state, HALLTIM counter pauses counting.                 |
|    |         | 0     | In debug state, HALLTIM counter counts normally.                 |
| 10 | WWDT    | 1     | In debug state, WWDT counter pauses counting.                    |
|    |         | 0     | In debug state, WWDT counter counts normally.                    |
| 9  | IWDT    | 1     | In debug state, IWDT counter pauses counting.                    |
|    |         | 0     | In debug state, IWDT counter counts normally.                    |
| 8  | RTC     | 1     | In debug state, RTC counter pauses counting.                     |
|    |         | 0     | In debug state, RTC counter counts normally.                     |
| 6  | LPTIM   | 1     | In debug state, LPTIM counter pauses counting.                   |
|    |         | 0     | In debug state, LPTIM counter counts normally.                   |
| 5  | BTIM123 | 1     | In debug state, BTIM1、BTIM2、BTIM3 counter pauses counting.       |
|    |         | 0     | In debug state, BTIM1、BTIM2、BTIM3 counter counts normally.       |
| 4  | GTIM4   | 1     | In debug state, GTIM4 counter pauses counting.                   |
|    |         | 0     | In debug state, GTIM4 counter counts normally.                   |
| 3  | GTIM3   | 1     | In debug state, GTIM3 counter pauses counting.                   |
|    |         | 0     | In debug state, GTIM3 counter counts normally.                   |
| 2  | GTIM2   | 1     | In debug state, GTIM2 counter pauses counting.                   |
|    |         | 0     | In debug state, GTIM2 counter counts normally.                   |
| 1  | GTIM1   | 1     | In debug state, GTIM1 counter pauses counting.                   |
|    |         | 0     | In debug state, GTIM1 counter counts normally.                   |
| 0  | ATIM    | 1     | In debug state, ATIM counter pauses counting.                    |
|    |         | 0     | In debug state, ATIM counter counts normally.                    |

For example, when the timer outputs PWM for motor control, the timer cannot be stopped, otherwise the motor will be damaged. For another example, the watchdog timer should stop counting during the breakpoint to prevent an undesired reset of the system.

## 30.8 Cautions

CW32L012 needs to use HCLK for debugging connection during debugging. It is not allowed to turn off HCLK during the debugging session. Therefore, executing WFI in the debugging environment will turn off the kernel HCLK, causing the debugging function to fail.

If the target chip has entered the DeepSleep mode, the DAP hardware debugging module does not work at this time, the debugger cannot connect with the DAP hardware debugging module of the target chip, and the debugging function cannot be realized.

The user must ensure that the chip is in Active mode or Sleep mode to use the debugger for debugging.

Therefore, it is recommended that users complete all functional debugging of the device in non-DeepSleep mode before merging the relevant code of DeepSleep, and then debug the code related to DeepSleep.



## 31 Digital signature

### 31.1 Overview

The digital signature is mainly used to store information such as the unique identification (UID) of the chip, product model, FLASH capacity, SRAM capacity, and the number of pins in the chip package, which can be read by SWD or CPU. The digital signature-related information is programmed at the factory, and user firmware or external devices can verify the legitimacy of the chip by reading the digital signature.



## 31.2 Product unique identification (UID) register (80bit)

The UID register stores the unique identifier of the chip, and its address is 0x0010 07B0 - 0x0010 07B9, a total of 80 bits. The UID is written during chip production and cannot be modified by the user. UID registers support reading in single-byte/half-word/full-word, etc., and then concatenated using custom algorithms.

Typical application scenarios of unique identifiers:

- Used as device serial number.
- Device legitimacy verification to prevent piracy:  
When the device is produced, the user uses the private key to encrypt the UID, and stores the calculation result in the main FLASH memory or OTP memory. If the result is compared with the previously stored calculation result, it is considered that the device is legal, otherwise the program will not start, which can effectively prevent the user's device from being illegally copied (piracy).
- Used as a security key:  
Combined with UID and private algorithm, users can perform security verification before users program FLASH to improve the security of the code in FLASH.
- Activate the secure boot process, etc.



### 31.3 Product model register

The product model register stores the ASCII code of the product model, and its address is 0x0010 07E0 - 0x0010 07EF, with a total of 16 bytes. When the product model is less than 16 bytes, it is filled with 0x00.

For example, the model of the chip is CW32L012C8T6, the corresponding data stored (starting from the low address) is: 0x43 0x57 0x33 0x32 0x4C 0x30 0x31 0x31 0x4B 0x38 0x54 0x36 0x00 0x00 0x00 0x00.

*Note: If there are letters in the product model, the letters need to be capitalized.*



## 31.4 FLASH capacity register

The FLASH capacity register stores the capacity of the built-in FLASH memory of the chip, and its address is 0x0010 07D8 - 0x0010 07DB, a total of 4 bytes.

The FLASH capacity size read from the FLASH capacity register is in bytes, such as 0x0001 0000 for 64KB, 0x0000 8000 for 32KB.





## 31.5 SRAM capacity register

The SRAM capacity register stores the capacity of the built-in SRAM memory in the chip, and its address is 0x0010 07D6 - 0x0010 07D7, a total of 2 bytes.

The SRAM capacity size read from the SRAM capacity register is in bytes, such as 0x1000 for 4KB.



## 31.6 Pin number register

The pin number register stores the chip pin number, its address is 0x0010 07D5, a total of 1 bytes. For example, 0x30 represents 48Pin.



# 32 Revision history

Table 32-1 Document revision history

| Date             | Revision | Changes          |
|------------------|----------|------------------|
| January 16, 2026 | Rev 1.0  | Initial release. |
|                  |          |                  |
|                  |          |                  |
|                  |          |                  |
|                  |          |                  |
|                  |          |                  |
|                  |          |                  |