

GigaDevice Semiconductor Inc.

GD32M531

Arm® Cortex®-M33 32-bit MCU

Applicable to GD32M531

User Manual

Revision 1.0

(Mar. 2026)

Table of Contents

Table of Contents	2
List of Figures	20
List of Tables	27
1 System and memory architecture	31
1.1 Arm® Cortex®-M33 processor	31
1.2 System architecture	32
1.3 Memory map	34
1.3.1 On-chip SRAM memory	37
1.3.2 On-chip flash memory overview	39
1.4 Boot configuration.....	39
1.5 System configuration controller.....	40
1.6 System configuration registers	41
1.6.1 Configuration register 0 (SYSCFG_CFG0)	41
1.6.2 Configuration register 1 (SYSCFG_CFG1)	41
1.6.3 EXTI sources selection register 0 (SYSCFG_EXTISS0).....	42
1.6.4 EXTI sources selection register 1 (SYSCFG_EXTISS1).....	42
1.6.5 EXTI sources selection register 2 (SYSCFG_EXTISS2).....	43
1.6.6 EXTI sources selection register 3 (SYSCFG_EXTISS3).....	44
1.6.7 Status register (SYSCFG_STAT).....	45
1.6.8 Configuration register 2 (SYSCFG_CFG2)	47
1.6.9 Configuration register 3 (SYSCFG_CFG3)	48
1.6.10 Configuration register 4 (SYSCFG_CFG4)	49
1.6.11 Timer input selection register 0 (SYSCFG_TIMERCISEL0).....	50
1.6.12 Timer input selection register 1 (SYSCFG_TIMERCISEL1).....	51
1.6.13 Configuration register 5 (SYSCFG_CFG5)	52
1.6.14 SRAM configuration register (SYSCFG_SRAMCFG)	54
1.6.15 Protection configuration register (SYSCFG_PRCFG)	54
1.6.16 TIMERx ADC start request configuration register (TADSRCFG)	55
1.6.17 TIMERx configuration register 0 (SYSCFG_TIMERxCFG0, x=0, 7)	57
1.6.18 TIMERx configuration register 1 (SYSCFG_TIMERxCFG1, x=0, 7)	59
1.6.19 TIMERx configuration register 2 (SYSCFG_TIMERxCFG2, x=0, 7)	60
1.6.20 TIMERx configuration register 0 (SYSCFG_TIMERxCFG0, x=1, 2)	61
1.6.21 TIMERx configuration register 1 (SYSCFG_TIMERxCFG1, x=1, 2)	63
1.6.22 TIMERx configuration register 2 (SYSCFG_TIMERxCFG2, x=1, 2)	64
1.6.23 Configuration register 6 (SYSCFG_CFG6)	65
1.7 Device electronic signature	66

1.7.1	Memory density information.....	66
1.7.2	Unique device ID (96 bits)	66
2	Flash memory controller (FMC).....	68
2.1	Overview	68
2.2	Characteristics.....	68
2.3	Function overview	68
2.3.1	Flash memory architecture	68
2.3.2	Error checking and correcting (ECC).....	69
2.3.3	Read operations	70
2.3.4	Unlock the FMC_CTL register	72
2.3.5	Page erase.....	72
2.3.6	Mass erase	73
2.3.7	Flash programming	75
2.3.8	Flash fast programming	76
2.3.9	Data flash operation.....	79
2.3.10	Check blank command	79
2.3.11	OTP programming	80
2.3.12	Option bytes erase.....	80
2.3.13	Option bytes modify	80
2.3.14	Option bytes description	81
2.3.15	Page erase/program protection	84
2.3.16	Security protection	85
2.4	Register definition	87
2.4.1	Wait state register (FMC_WS).....	87
2.4.2	Unlock key register (FMC_KEY).....	88
2.4.3	Option byte unlock key register (FMC_OBKEY).....	88
2.4.4	Status register (FMC_STAT).....	89
2.4.5	Control register (FMC_CTL)	90
2.4.6	Address register (FMC_ADDR)	91
2.4.7	Program data register (FMC_DATA).....	92
2.4.8	ECC control and status register (FMC_ECCCS).....	92
2.4.9	Option byte status register (FMC_OBSTAT).....	94
2.4.10	Erase / Program protection register (FMC_WP).....	95
2.4.11	Product ID register 0 (FMC_PID0).....	95
2.4.12	Product ID register 1 (FMC_PID1).....	95
3	Power management unit (PMU).....	97
3.1	Overview	97
3.2	Characteristics.....	97
3.3	Function overview	97
3.3.1	V _{DD} / V _{DDA} power domain	98

3.3.2	V _{CORE} power domain	99
3.3.3	Power saving modes	100
3.3.4	LVD detectors	101
3.4	Register definition	105
3.4.1	Control register (PMU_CTL)	105
3.4.2	Control and status register (PMU_CS)	106
3.4.3	Low Voltage Detector 1 Control Register (PMU_LVD1CTL)	107
3.4.4	Low Voltage Detector 2 Control Register (PMU_LVD2CTL)	108
4	Reset and clock unit (RCU)	111
4.1	Reset control unit (RCTL)	111
4.1.1	Overview	111
4.1.2	Function overview	111
4.2	Clock control unit (CCTL)	112
4.2.1	Overview	112
4.2.2	Characteristics	113
4.2.3	Function overview	114
4.2.4	Operations about event interconnect	117
4.3	Register definition	118
4.3.1	Control register (RCU_CTL)	118
4.3.2	Configuration register 0 (RCU_CFG0)	120
4.3.3	Interrupt register (RCU_INT)	122
4.3.4	APB2 reset register (RCU_APB2RST)	125
4.3.5	APB1 reset register (RCU_APB1RST)	127
4.3.6	AHB enable register (RCU_AHBEN)	129
4.3.7	APB2 enable register (RCU_APB2EN)	131
4.3.8	APB1 enable register (RCU_APB1EN)	133
4.3.9	Reset source/clock register (RCU_RSTSCK)	135
4.3.10	AHB reset register (RCU_AHBRST)	138
4.3.11	Configuration register 1 (RCU_CFG1)	140
4.3.12	Configuration register 2 (RCU_CFG2)	141
5	Clock frequency measurement unit (CFMU)	143
5.1	Overview	143
5.2	Characteristics	143
5.3	Function overview	143
5.3.1	CFMU block diagram	143
5.3.2	Clock frequency measurement	144
5.3.3	Digital filter function on CFMUREF pin	144
5.3.4	CFMU interrupts	145
5.4	Register definition	146
5.4.1	Control register (CFMU_CTL)	146

5.4.2	Status register (CFMU_STAT)	148
5.4.3	Limit value configuration register (CFMU_LVCFG)	148
5.4.4	Counter value register (CFMU_CNT)	149
6	Interrupt/event controller (EXTI).....	150
6.1	Overview	150
6.2	Characteristics.....	150
6.3	Interrupts function overview.....	150
6.4	External interrupt and event (EXTI) block diagram	154
6.5	External Interrupt and event function overview	154
6.6	EXTI Register	157
6.6.1	Interrupt enable register (EXTI_INTEN)	157
6.6.2	Event enable register (EXTI_EVEN)	157
6.6.3	Rising edge trigger enable register (EXTI_RTEN)	158
6.6.4	Falling edge trigger enable register (EXTI_FTEN)	158
6.6.5	Software interrupt event register (EXTI_SWIEV)	158
6.6.6	Pending register (EXTI_PD)	159
6.6.7	Digital filter enable register (EXTI_DFEN).....	159
6.6.8	Sampling clock select register (EXTI_SCS)	160
7	General-purpose and alternate-function I/Os (GPIO and AFIO).....	161
7.1	Overview	161
7.2	Characteristics.....	161
7.3	Function overview	161
7.3.1	GPIO pin configuration	163
7.3.2	External interrupt / event lines	163
7.3.3	Alternate functions (AF)	163
7.3.4	Additional functions.....	163
7.3.5	Input configuration	164
7.3.6	Output configuration	164
7.3.7	Analog configuration	165
7.3.8	Alternate function (AF) configuration	165
7.3.9	GPIO locking function	166
7.3.10	GPIO single cycle toggle function.....	166
7.3.11	GPIO pin level hold function	166
7.4	Register definition	167
7.4.1	Port control register (GPIOx_CTL, x = A..G,N).....	167
7.4.2	Port output mode register (GPIOx_OMODE, x = A..G,N).....	169
7.4.3	Port output speed register (GPIOx_OSPD, x = A..G,N)	170
7.4.4	Port pull-up / pull-down register (GPIOx_PUD, x = A..G,N)	172
7.4.5	Port input status register (GPIOx_ISTAT, x = A..G,N).....	174

7.4.6	Port output control register (GPIOx_OCTL, x = A..G,N).....	174
7.4.7	Port bit operate register (GPIOx_BOP, x = A..G,N)	175
7.4.8	Port configuration lock register (GPIOx_LOCK, x = A..G,N)	175
7.4.9	Alternate function selected register 0 (GPIOx_AFSEL0, x = A..G,N)	176
7.4.10	Alternate function selected register 1 (GPIOx_AFSEL1, x = A..G,N)	177
7.4.11	Bit clear register (GPIOx_BC, x = A..G,N)	178
7.4.12	Port bit toggle register (GPIOx_TG, x = A..G,N).....	179
7.4.13	Pin level hold config register (GPIOx_HLDCFG, x = A..G,N)	179
7.4.14	Pin function select register (GPIOx_FUNCSEL, x = N)	181
7.4.15	Pin level hold control register (GPIOx_HLDCTL, x = N).....	181
8	Port Output Controller (POC).....	183
8.1	Overview	183
8.2	Characteristics.....	183
8.3	Function overview	183
8.3.1	POC clock	185
8.3.2	Output disable function	185
8.3.3	Target timer selection.....	189
8.3.4	Output switching	189
8.3.5	Detection masking function.....	190
8.3.6	POC interrupt.....	190
8.4	Register definition	191
8.4.1	Status register 0 (POC_STAT0).....	191
8.4.2	Input n detection configuration register (POC_INnDCFG) (n=0...5)	193
8.4.3	Control register 0 (POC_CTL0)	195
8.4.4	Software disabling request generation register (POC_SWDRG)	196
8.4.5	Complementary detection configuration register 0 (POC_CDCFG0)	197
8.4.6	Complementary detection configuration register 1 (POC_CDCFG1)	199
8.4.7	Output disable mode register 0 (POC_ODMODE0)	200
8.4.8	Output disable mode register 1 (POC_ODMODE1)	202
8.4.9	Request selection register 0 (POC_REQSEL0)	205
8.4.10	Request selection register 1 (POC_REQSEL1)	209
8.4.11	Status register 1 (POC_STAT1).....	216
8.4.12	Control register 1 (POC_CTL1)	217
8.4.13	Extended control register 0 (POC_EXTCTL0).....	218
8.4.14	Extended control register 1 (POC_EXTCTL1).....	220
8.4.15	Input n detection mask register (POC_INnDMK) (n=0...5)	224
8.4.16	Comparator n detection mask register (POC_CMPnDMK) (n=0...3)	225
9	GPTIMER Output Controller (GTOC).....	227
9.1	Overview	227
9.2	Characteristics.....	227

9.3	Function overview	227
9.3.1	GTOC clock	228
9.3.2	Output close function	228
9.3.3	GTOC interrupt	231
9.4	Register definition	232
9.4.1	GTOCx configuration register (GTOCx_CFG) (x=0...3).....	232
9.4.2	GTOCx output closing request control register (GTOCx_OCRCTL) (x=0...3)	234
9.4.3	GTOCx write protection register (GTOCx_WP) (x=0...3).....	235
9.4.4	GTOCx extended closing request control register (GTOCx_ECRCTL) (x=0...3)	236
10	Cyclic redundancy checks management unit (CRC)	238
10.1	Overview	238
10.2	Characteristics	238
10.3	Function overview.....	239
10.4	Register definition.....	241
10.4.1	Data register (CRC_DATA)	241
10.4.2	Free data register (CRC_FDATA)	241
10.4.3	Control register (CRC_CTL)	242
10.4.4	Initialization data register (CRC_IDATA).....	243
10.4.5	Polynomial register (CRC_POLY).....	243
11	Trigonometric Math Unit (TMU)	244
11.1	Overview	244
11.2	Characteristics.....	244
11.3	Block diagram.....	244
11.4	Function overview	245
11.4.1	Data format and configuration	245
11.4.2	Floating-point data format.....	246
11.4.3	Mode configuration	247
11.4.4	TMU operation pending	254
11.4.5	Zero-overhead mode	255
11.4.6	TMU instruction.....	255
11.4.7	Interrupt and DMA requests.....	256
11.5	Registers definition	256
11.5.1	Control and status register (TMU_CS)	256
11.5.2	Input data register (TMU_IDATA)	259
11.5.3	Output Data register (TMU_ODATA)	259
12	Direct memory access controller (DMA).....	261
12.1	Overview	261
12.2	Characteristics	261

12.3	Block diagram	262
12.4	Function overview.....	262
12.4.1	DMA operation	262
12.4.2	Peripheral handshake.....	264
12.4.3	Arbitration.....	264
12.4.4	Address generation.....	265
12.4.5	Circular mode.....	265
12.4.6	Memory to memory mode.....	265
12.4.7	Channel configuration.....	265
12.4.8	Interrupt.....	266
12.4.9	DMA request mapping	267
12.5	Register definition.....	268
12.5.1	Interrupt flag register (DMA_INTF)	268
12.5.2	Interrupt flag clear register (DMA_INTC).....	268
12.5.3	Channel x control register (DMA_CHxCTL)	269
12.5.4	Channel x counter register (DMA_CHxCNT).....	271
12.5.5	Channel x peripheral base address register (DMA_CHxPADDR).....	272
12.5.6	Channel x memory base address register (DMA_CHxMADDR).....	272
13	DMA request multiplexer (DMAMUX)	274
13.1	Overview	274
13.2	Characteristics	274
13.3	Block diagram	275
13.4	Function overview.....	275
13.4.1	DMAMUX signals.....	276
13.4.2	DMAMUX request multiplexer	276
13.4.3	DMAMUX request generator	279
13.4.4	Channel configurations	280
13.4.5	Interrupt.....	280
13.4.6	DMAMUX mapping.....	280
13.5	Register definition.....	286
13.5.1	Request multiplexer channel x configuration register (DMAMUX_RM_CHxCFG).....	286
13.5.2	Request multiplexer channel interrupt flag register (DMAMUX_RM_INTF).....	287
13.5.3	Request multiplexer channel interrupt flag clear register (DMAMUX_RM_INTC).....	287
13.5.4	Request generator channel x configuration register (DMAMUX_RG_CHxCFG)	288
13.5.5	Request generator interrupt flag register (DMAMUX_RG_INTF).....	289
13.5.6	Request generator interrupt flag clear register (DMAMUX_RG_INTC).....	289
14	Debug (DBG)	291
14.1	Overview	291
14.2	JTAG/SW function overview	291

14.2.1	Switch JTAG or SW interface	291
14.2.2	Pin assignment	291
14.2.3	JTAG daisy chained structure	292
14.2.4	Debug reset	292
14.2.5	JEDEC-106 ID code	292
14.3	Debug hold function overview	293
14.3.1	Debug support for power saving mode.....	293
14.3.2	Debug support for TIMER, I2C, WWDGT, FWDGT and CAN	293
14.4	Register definition.....	294
14.4.1	ID code register (DBG_ID).....	294
14.4.2	Control register (DBG_CTL)	294
15	Analog-to-digital converter (ADC).....	297
15.1	Overview	297
15.2	Characteristics	298
15.3	Pins and internal signals	299
15.4	Block diagram	300
15.5	Function overview.....	301
15.5.1	ADC clock	301
15.5.2	ADC enable.....	301
15.5.3	Scanning operation modes	302
15.5.4	Group_pri1 scan once mode	303
15.5.5	Group_pri1 scan continuous mode.....	311
15.5.6	Groups scan mode	319
15.5.7	Configurable channel priority function	344
15.5.8	Conversion result threshold monitor	346
15.5.9	Data storage mode	347
15.5.10	Sample time configuration	353
15.5.11	External trigger configuration.....	353
15.5.12	DMA function.....	355
15.5.13	DMA overflow detection	356
15.5.14	ADC internal channels	357
15.5.15	Programmable resolution (DRES)	358
15.5.16	Disconnection detection assist function (precharge and discharge)	359
15.5.17	Self-Diagnosis Function	359
15.5.18	Sample-and-Hold Function (for ADC0 only)	360
15.5.19	On-chip hardware oversample (data accumulation/average).....	362
15.5.20	ADC data registers automatic clearing function	362
15.5.21	ADC interrupts	363
15.6	Register definition.....	364
15.6.1	Software EOC control register (ADC_EOCCTL)	364

15.6.2	Control register 0 (ADC_CTL0)	367
15.6.3	Group config register 0 (ADC_GPCGF0)	370
15.6.4	Group config register 1 (ADC_GPCGF1)	371
15.6.5	Channel selection register 0 (ADC_CHSEL0)	374
15.6.6	Channel selection register 1 (ADC_CHSEL1)	376
15.6.7	Sample time register 0 (ADC_SAMPR0)	378
15.6.8	Sample time register 1 (ADC_SAMPR1)	379
15.6.9	Sample time register 2 (ADC_SAMPR2)	379
15.6.10	Channel priority register 0 (ADC_CHPRI0)	380
15.6.11	Channel priority register 1 (ADC_CHPRI1)	381
15.6.12	ADC channel addition times register 0 (ADC_ADDT0)	382
15.6.13	ADC channel addition times register 1 (ADC_ADDT1)	383
15.6.14	Self-diagnosis data register (ADC_SDDATA)	384
15.6.15	Group_pri x bifurcate data register (ADC_GPxBIDATA, x=1,2,3,4)	385
15.6.16	Group_pri x bifurcate data register 1(ADC_GPxBIDATA1, x=1,2,3,4)	385
15.6.17	Group_pri x bifurcate data register 2(ADC_GPxBIDATA2, x=1,2,3,4)	386
15.6.18	Group_pri x data DMA register (ADC_GPxDMAR, x=1,2,3,4)	386
15.6.19	Channel x data register (ADC_CHxDATA)	387
15.6.20	Temperature sensor channel data register (ADC_TEMPDATA)	387
15.6.21	Internal reference voltage channel data register (ADC_VINTDATA)	387
15.6.22	Watchdog control register (ADC_WDCTL)	388
15.6.23	Watchdog A threshold register (ADC_WDATHOLD)	390
15.6.24	Watchdog B threshold register (ADC_WDBTHOLD)	390
15.6.25	Watchdog A channel config register (ADC_WDACHCFG)	391
15.6.26	Watchdog A channel status register (ADC_WDACHSTAT)	393
15.6.27	Control register 1 (ADC_CTL1)	395
15.6.28	Sample-and-hold control register (ADC_SHCTL)	395
15.6.29	DMA control register (ADC_DMACTL)	397
15.6.30	Bifurcate trigger control register (ADC_BITRGCTL)	399
16	Digital-to-analog converter (DAC)	402
16.1	Overview	402
16.2	Characteristics	402
16.3	Function description	404
16.3.1	DAC enable	404
16.3.2	DAC data configuration	404
16.3.3	DAC trigger	404
16.3.4	DAC conversion	404
16.3.5	DAC noise wave	405
16.3.6	DAC output voltage	406
16.3.7	Prevent Interference between D/A and A/D	406
16.3.8	DAC concurrent conversion	407
16.3.9	DAC low-power modes	407

16.4	DAC register	408
16.4.1	DACx control register 0 (DAC_CTL0).....	408
16.4.2	DACx software trigger register (DAC_SWT)	411
16.4.3	DACx_OUT0 12-bit right-aligned data holding register (DAC_OUT0_R12DH)	411
16.4.4	DACx_OUT0 12-bit left-aligned data holding register (DAC_OUT0_L12DH)	412
16.4.5	DACx_OUT1 12-bit right-aligned data holding register (DAC_OUT1_R12DH)	412
16.4.6	DACx_OUT1 12-bit left-aligned data holding register (DAC_OUT1_L12DH)	413
16.4.7	DACx concurrent mode 12-bit right-aligned data holding register (DACC_R12DH).....	413
16.4.8	DACx concurrent mode 12-bit left-aligned data holding register (DACC_L12DH).....	414
16.4.9	DACx_OUT0 data output register (DAC_OUT0_DO).....	414
16.4.10	DACx_OUT1 data output register (DAC_OUT1_DO).....	415
17	Comparator (CMP)	416
17.1	Overview	416
17.2	Characteristic	416
17.3	Function overview.....	416
17.3.1	CMP clock	418
17.3.2	CMP I/O configuration.....	418
17.3.3	CMP hysteresis	419
17.3.4	CMP register write protection	419
17.3.5	CMP interrupt.....	420
17.3.6	CMP digital filter	420
17.3.7	CMP low-power mode.....	420
17.4	Register definition.....	422
17.4.1	CMP status register (CMP_STAT)	422
17.4.2	CMP interrupt flag clear register (CMP_IFC).....	423
17.4.3	CMP0 control/status register (CMP0_CS).....	424
17.4.4	CMP1 control/status register (CMP1_CS).....	426
17.4.5	CMP2 control/status register (CMP2_CS).....	428
17.4.6	CMP3 control/status register (CMP3_CS).....	430
18	Event interconnection unit (EVIC).....	433
18.1	Overview	433
18.2	Characteristics	433
18.3	Function overview.....	433
18.3.1	Event source selection	434
18.3.2	Event interconnect for I/O pin	437
18.4	Register definition.....	441
18.4.1	Software event register (EVIC_SWEV)	441
18.4.2	Event interconnect for single I/O register x (EVIC_SGIOx) (x=0...3)	441
18.4.3	Event interconnect for input group E register (EVIC INGRPE)	442
18.4.4	Event interconnect for input group F register (EVIC INGRPF).....	443

18.4.5	Event interconnect for output group E register (EVIC_OUTGRPE)	443
18.4.6	Event interconnect for output group F register (EVIC_OUTGRPF).....	444
18.4.7	Event interconnect for DAC0 conversion register (EVIC_DAC0COV)	445
18.4.8	Event interconnect for ADC0 conversion register (EVIC_ADC0COV)	445
18.4.9	Event interconnect for ADC2 conversion register (EVIC_ADC2COV)	446
18.4.10	Event interconnect for RCU register (EVIC_RCU)	447
18.4.11	Event interconnect for TIMER0 register (EVIC_TIMER0)	447
18.4.12	Event interconnect for TIMER7 register (EVIC_TIMER7)	448
18.4.13	Event interconnect for CPTIMER0 register (EVIC_CPTIMER0)	449
18.4.14	Event interconnect for CPTIMERW register (EVIC_CPTIMERW).....	449
18.4.15	Event interconnect for TIMER1 register (EVIC_TIMER1)	450
18.4.16	Event interconnect for TIMER2 register (EVIC_TIMER2)	451
18.4.17	Event interconnect for GPTIMER register 0 (EVIC_GPTIMER_0).....	451
18.4.18	Event interconnect for GPTIMER register 1 (EVIC_GPTIMER_1).....	452
18.4.19	Event interconnect for GPTIMER register 2 (EVIC_GPTIMER_2).....	453
18.4.20	Event interconnect for GPTIMER register 3 (EVIC_GPTIMER_3).....	453
18.4.21	Event interconnect for DAC0 enable register (EVIC_DAC0EN).....	454
18.4.22	Event interconnect for DMAMUX register 0 (EVIC_DMAMUX_0).....	455
18.4.23	Event interconnect for DMAMUX register 1 (EVIC_DMAMUX_1).....	455
18.4.24	Event interconnect for DMAMUX register 2 (EVIC_DMAMUX_2).....	456
18.4.25	Event interconnect for DMAMUX register 3 (EVIC_DMAMUX_3).....	457
18.4.26	Event interconnect for DMAMUX register 4 (EVIC_DMAMUX_4).....	458
18.4.27	Event interconnect for DMAMUX register 5 (EVIC_DMAMUX_5).....	458
18.4.28	Slave mode configuration register 0 (EVIC_SMCFG0)	459
18.4.29	Slave mode configuration register 1 (EVIC_SMCFG1)	459
18.4.30	Single I/O configuration register x (EVIC_SGIOCFGx) (x=0...3)	460
18.4.31	Group E configuration register (EVIC_GRPFCFG)	461
18.4.32	Group E data holding register (EVIC_GRPEDH)	463
18.4.33	Group F configuration register (EVIC_GRPFCFG)	463
18.4.34	Group F data holding register (EVIC_GRPFDH).....	465
19	Space vector pulse width modulation (SVPWM).....	466
19.1	Overview	466
19.2	Characteristics	466
19.2.1	Block diagram	466
19.3	Function overview.....	466
19.3.1	SVPWM clock	466
19.3.2	Data format configuration	467
19.3.3	SVPWM principle	467
19.3.4	SVPWM implementation	468
19.3.5	Five-segment SVPWM	469
19.3.6	Seven-segment SVPWM	470
19.3.7	OverModulation processing	472

19.4	Register definition.....	473
19.4.1	Control register 0 (SVPWM_CTL0)	473
19.4.2	Ualpha register (SVPWM_UALPHA)	473
19.4.3	Ubeta register (SVPWM_UBETA)	474
19.4.4	Counter auto reload register (SVPWM_CAR)	474
19.4.5	Ta register (SVPWM_TA).....	475
19.4.6	Tb register (SVPWM_TB)	475
19.4.7	Tc register (SVPWM_TC)	476
19.4.8	Sector register (SVPWM_SEC).....	476
19.4.9	Status register (SVPWM_STAT).....	476
20	Watchdog timer (WDGT)	479
20.1	Free watchdog timer (FWDGT).....	479
20.1.1	Overview	479
20.1.2	Characteristics	479
20.1.3	Function overview	479
20.1.4	Register definition	482
20.2	Window watchdog timer (WWDGT).....	486
20.2.1	Overview	486
20.2.2	Characteristics	486
20.2.3	Function overview	486
20.2.4	Register definition	490
21	TIMER (TIMERx).....	493
21.1	Advanced timer (TIMERx, x=0, 7)	494
21.1.1	Overview	494
21.1.2	Characteristics	494
21.1.3	Block diagram	495
21.1.4	Function overview	495
21.1.5	Registers definition (TIMERx, x=0, 7).....	547
21.2	General level0 timer (TIMERx, x=1,2)	607
21.2.1	Overview	607
21.2.2	Characteristics	607
21.2.3	Block diagram	607
21.2.4	Function overview	608
21.2.5	Registers definition (TIMERx, x=1,2).....	624
22	General purpose timer (GPTIMERx).....	646
22.1	Overview	646
22.2	Characteristics	646
22.3	Block diagram	647
22.4	Function overview.....	647

22.4.1	Clock selection	647
22.4.2	Clock prescaler	648
22.4.3	Enable/disable/reset the timer counter	649
22.4.4	Period counting mode	650
22.4.5	Event counting mode	654
22.4.6	count direction	655
22.4.7	Counter repetition	655
22.4.8	Shadow register loading function	657
22.4.9	Capture/compare channels	659
22.4.10	PWM mode	662
22.4.11	Complementary output and dead-time insertion	664
22.4.12	Duty cycle 0% or 100% output	664
22.4.13	Synchronization between GPTIMERx	665
22.4.14	Single pulse mode	668
22.4.15	Interrupt skipping function	669
22.4.16	ADC trigger request	671
22.4.17	Interaction with the EVIC	673
22.4.18	Protection function	673
22.4.19	Initialization of the output pin	674
22.4.20	Timer DMA mode	675
22.4.21	Timer debug mode	675
22.5	Registers definition (GPTIMERx, x=0, 1)	676
22.5.1	Write-protect register (GPTIMERx_WP)	676
22.5.2	Software enable register (GPTIMERx_SWEN)	677
22.5.3	Software disable register (GPTIMERx_SWDIS)	677
22.5.4	Software reset register (GPTIMERx_SWRST)	678
22.5.5	Counter enable source selection register (GPTIMERx_ESSEL)	678
22.5.6	Counter disable source selection register (GPTIMERx_DSSEL)	680
22.5.7	Counter reset source selection register (GPTIMERx_RSSEL)	681
22.5.8	Channel 0 capture source selection register (GPTIMERx_CH0CSSEL)	683
22.5.9	Channel 1 capture source selection register (GPTIMERx_CH1CSSEL)	684
22.5.10	Control register 0 (GPTIMERx_CTL0)	686
22.5.11	Count up event source selection register (GPTIMERx_CUPEVSSEL)	688
22.5.12	Count down event source selection register (GPTIMERx_CDNEVSSEL)	689
22.5.13	Channel 0 control register (GPTIMERx_CH0CTL)	691
22.5.14	Channel 1 control register (GPTIMERx_CH1CTL)	694
22.5.15	Channel control register (GPTIMERx_CHCTL)	696
22.5.16	DMA and interrupt enable register (GPTIMERx_DMAINTEN)	698
22.5.17	Interrupt flag register (GPTIMERx_INTF)	700
22.5.18	Update source selection register (GPTIMERx_UPSSEL)	703
22.5.19	Counter register (GPTIMERx_CNT)	704
22.5.20	Prescaler register (GPTIMERx_PSC)	704
22.5.21	Counter auto reload register (GPTIMERx_CAR)	705

22.5.22	Channel 0 capture/compare value register (GPTIMERx_CH0CV).....	705
22.5.23	Channel 1 capture/compare value register (GPTIMERx_CH1CV).....	706
22.5.24	Channel 0 additional compare value register (GPTIMERx_CH0COMV_ADD).....	706
22.5.25	Channel 1 additional compare value register (GPTIMERx_CH1COMV_ADD).....	707
22.5.26	Dead-time control register (GPTIMERx_DTCTL)	707
22.5.27	ADC trigger control register (GPTIMERx_ADCTL).....	709
22.5.28	ADC trigger compare value register 1 (GPTIMERx_ADCCR1).....	711
22.5.29	ADC trigger compare value register 2 (GPTIMERx_ADCCR2).....	711
22.5.30	ADC trigger skipping register (GPTIMERx_ADCTRGS)	711
22.5.31	Additional interrupt skipping control register 0 (GPTIMERx_ADDINTSCTL0)	713
22.5.32	Additional interrupt skipping control register 1 (GPTIMERx_ADDINTSCTL1)	714
22.5.33	Interrupt and ADC trigger signal skipping register (GPTIMERx_IADCTSS)	716
22.5.34	Counter repetition register (GPTIMERx_CREP)	717
22.5.35	Synchronous reset control register (GPTIMERx_SYNRSCTL).....	718
22.5.36	DMA configuration register (GPTIMERx_DMACFG)	719
22.5.37	DMA transfer buffer register (GPTIMERx_DMATB)	720
22.5.38	Configuration register (GPTIMERx_CFG)	720
23	Compare timer W(CPTIMERW)	722
23.1	Overview	722
23.2	Characteristics	722
23.3	Function overview.....	722
23.3.1	Clock source	723
23.3.2	Clock prescaler	723
23.3.3	Counter up counting	724
23.3.4	Counter width configuration	725
23.3.5	Input capture and output compare	725
23.3.6	EVIC interconnect.....	727
23.4	Register definition.....	729
23.4.1	Control register 0 (CPTIMERW_CTL0)	729
23.4.2	Control register 1 (CPTIMERW_CTL1)	730
23.4.3	Prescaler register (CPTIMERW_PSC)	731
23.4.4	Counter register (CPTIMERW_CNT).....	732
23.4.5	Interrupt enable register (CPTIMERW_INTEN).....	732
23.4.6	Interrupt flag register (CPTIMERW_INTF)	733
23.4.7	Software event generation register (CPTIMERW_SVEVG)	734
23.4.8	Input capture channel 0 capture value register (CPTIMERW_ICH0CV)	735
23.4.9	Input capture channel 1 capture value register (CPTIMERW_ICH1CV)	735
23.4.10	Output compare channel 0 compare value register (CPTIMERW_OCH0CV)	735
23.4.11	Output compare channel 1 compare value register (CPTIMERW_OCH1CV)	736
23.4.12	Output compare channel 2 compare value register (CPTIMERW_OCH2CV)	736
23.4.13	Output compare channel 3 compare value register (CPTIMERW_OCH3CV)	737
23.4.14	Counter auto reload register (CPTIMERW_CAR)	737

24	Compare timer (CPTIMER)	739
24.1	Overview	739
24.2	Characteristics	739
24.3	Function overview	739
24.3.1	Clock source	740
24.3.2	Clock prescaler	740
24.3.3	Counter up counting	740
24.3.4	EVIC interconnect	741
24.4	Register definition	743
24.4.1	Control register 0 (CPTIMER_CTL0)	743
24.4.2	Counter 0 register (CPTIMER_CNT0)	743
24.4.3	Counter 1 register (CPTIMER_CNT1)	744
24.4.4	Counter 0 prescaler register (CPTIMER_CNT0PSC)	744
24.4.5	Counter 1 prescaler register (CPTIMER_CNT1PSC)	745
24.4.6	Interrupt enable register (CPTIMER_INTEN)	745
24.4.7	Interrupt flag register (CPTIMER_INTF)	746
24.4.8	Software event generation register (CPTIMER_SWEVG)	746
24.4.9	Counter 0 auto reload register (CPTIMER_CNT0CAR)	747
24.4.10	Counter 1 auto reload register (CPTIMER_CNT1CAR)	748
25	Universal asynchronous receiver / transmitter (UART)	749
25.1	Overview	749
25.2	Characteristics	749
25.3	Function overview	750
25.3.1	UART frame format	750
25.3.2	Baud rate generation	751
25.3.3	UART transmitter	752
25.3.4	UART receiver	753
25.3.5	Use DMA for data buffer access	754
25.3.6	Multi-processor communication	756
25.3.7	LIN mode	757
25.3.8	IrDA SIR ENDEC mode	758
25.3.9	Half-duplex communication mode	759
25.3.10	UART interrupts	759
25.4	Register definition	761
25.4.1	Status register 0 (UART_STAT0)	761
25.4.2	Data register (UART_DATA)	763
25.4.3	Baud rate register (UART_BAUD)	763
25.4.4	Control register 0 (UART_CTL0)	764
25.4.5	Control register 1 (UART_CTL1)	766
25.4.6	Control register 2 (UART_CTL2)	767

25.4.7	Guard time and prescaler register (UART_GP).....	769
25.4.8	Coherence control register (UART_CHC)	770
26	Inter-integrated circuit interface (I2C).....	771
26.1	Overview	771
26.2	Characteristics	771
26.3	Function overview.....	771
26.3.1	Clock requirements.....	772
26.3.2	I2C communication flow.....	773
26.3.3	Digital Noise filter	776
26.3.4	I2C timings configuration	776
26.3.5	I2C reset	778
26.3.6	Data transfer	778
26.3.7	I2C slave mode	780
26.3.8	I2C master mode	785
26.3.9	SMBus support	790
26.3.10	SMBus mode	793
26.3.11	Wakeup from power saving modes	794
26.3.12	Use DMA for data transfer	795
26.3.13	I2C error and interrupts.....	795
26.3.14	I2C debug mode	796
26.3.15	Event Link Function (Output).....	796
26.4	Register definition.....	797
26.4.1	Control register 0 (I2C_CTL0)	797
26.4.2	Control register 1 (I2C_CTL1)	799
26.4.3	Slave address register 0 (I2C_SADDR0)	801
26.4.4	Slave address register 1 (I2C_SADDR1)	802
26.4.5	Timing register (I2C_TIMING)	803
26.4.6	Timeout register (I2C_TIMEOUT).....	804
26.4.7	Status register (I2C_STAT)	805
26.4.8	Status clear register (I2C_STATC)	808
26.4.9	PEC register (I2C_PEC)	809
26.4.10	Receive data register (I2C_RDATA)	809
26.4.11	Transmit data register (I2C_TDATA).....	810
26.4.12	Control register 2 (I2C_CTL2)	810
27	Serial peripheral interface (SPI).....	812
27.1	Overview	812
27.2	Characteristics	812
27.2.1	SPI characteristics	812
27.3	SPI function overview	813
27.3.1	SPI block diagram.....	813

27.3.2	SPI signal description	813
27.3.3	SPI clock timing and data format	814
27.3.4	NSS function	815
27.3.5	SPI operating modes	816
27.3.6	DMA function	825
27.3.7	CRC function	825
27.3.8	SPI interrupts	826
27.4	Register definition	828
27.4.1	Control register 0 (SPI_CTL0)	828
27.4.2	Control register 1 (SPI_CTL1)	830
27.4.3	Status register (SPI_STAT)	831
27.4.4	Data register (SPI_DATA)	832
27.4.5	CRC polynomial register (SPI_CRCPOLY)	832
27.4.6	RX CRC register (SPI_RCRC)	833
27.4.7	TX CRC register (SPI_TCRC)	834
27.4.8	Quad-SPI mode control register (SPI_QCTL)	834
28	Controller area network (CAN)	836
28.1	Overview	836
28.2	Characteristics	836
28.3	Function overview	837
28.3.1	Working mode	837
28.3.2	Communication modes	838
28.3.3	Data transmission	839
28.3.4	Data reception	841
28.3.5	Filtering function	842
28.3.6	Time-triggered communication	845
28.3.7	Communication parameters	846
28.3.8	Error flags	848
28.3.9	CAN interrupts	848
28.4	CAN registers	851
28.4.1	Control register (CAN_CTL)	851
28.4.2	Status register (CAN_STAT)	852
28.4.3	Transmit status register (CAN_TSTAT)	854
28.4.4	Receive message FIFO0 register (CAN_RFIFO0)	857
28.4.5	Receive message FIFO1 register (CAN_RFIFO1)	858
28.4.6	Interrupt enable register (CAN_INTEN)	858
28.4.7	Error register (CAN_ERR)	860
28.4.8	Bit timing register (CAN_BT)	861
28.4.9	Transmit mailbox identifier register (CAN_TMIx) (x = 0...2)	862
28.4.10	Transmit mailbox property register (CAN_TMPx) (x=0..2)	863
28.4.11	Transmit mailbox data0 register (CAN_TMDATA0x) (x = 0...2)	863

28.4.12	Transmit mailbox data1 register (CAN_TMDATA1x) (x=0..2).....	864
28.4.13	Receive FIFO mailbox identifier register (CAN_RFIFOMIx) (x = 0, 1)	864
28.4.14	Receive FIFO mailbox property register (CAN_RFIFOMPx) (x = 0, 1)	865
28.4.15	Receive FIFO mailbox data0 register (CAN_RFIFOMDATA0x) (x = 0, 1).....	865
28.4.16	Receive FIFO mailbox data1 register (CAN_RFIFOMDATA1x) (x = 0, 1).....	866
28.4.17	Filter control register (CAN_FCTL).....	866
28.4.18	Filter mode configuration register (CAN_FMCFG)	867
28.4.19	Filter scale configuration register (CAN_FSCFG)	867
28.4.20	Filter associated FIFO register (CAN_FAFIFO).....	868
28.4.21	Filter working register (CAN_FW).....	868
28.4.22	Filter x data y register (CAN_FxDATAy) (x = 0...27, y = 0, 1)	869
29	Appendix	870
29.1	List of abbreviations used in register	870
29.2	List of terms.....	870
29.3	Available peripherals	871
30	Revision history.....	872

List of Figures

Figure 1-1. The structure of the Cortex®-M33 processor	32
Figure 1-2. GD32M531 system architecture	34
Figure 1-3. ECC decoder	38
Figure 2-1. Process of page erase operation	73
Figure 2-2. Process of mass erase operation	74
Figure 2-3. Process of program operation	76
Figure 2-4. Process of fast programming operation	78
Figure 3-1. Power supply overview	98
Figure 3-2. Waveform of the POR / PDR	99
Figure 3-3. Low Voltage Detectors 0 block diagram	102
Figure 3-4. Low Voltage Detectors 1 block diagram	102
Figure 3-5. Low Voltage Detectors 2 block diagram	103
Figure 4-1. The system reset circuit	112
Figure 4-2. Clock tree	113
Figure 4-3. HXTAL clock source	114
Figure 4-4. HXTAL clock source in bypass mode	114
Figure 5-1. Block diagram of CFMU	144
Figure 5-2. Clock frequency accuracy measurement	144
Figure 6-1. Block diagram of EXTI	154
Figure 7-1. Basic structure of a standard I / O	162
Figure 7-2. Basic structure of Input configuration	164
Figure 7-3. Basic structure of Output configuration	165
Figure 7-4. Basic structure of Analog configuration	165
Figure 7-5. Basic structure of Alternate function configuration	166
Figure 8-1. POC block diagram	184
Figure 8-2. Output disable control for TIMER and GPTIMER channel pin	184
Figure 8-3. POC pins falling edge detection	186
Figure 8-4. POC pins low-level detection (INDSNUM[3:0]=4'b0000)	186
Figure 8-5. Complementary PWM channel output detection	187
Figure 8-6. POC pin falling input detection masking	190
Figure 9-1. GTOCx (x=0...3) block diagram	228
Figure 9-2. GTOCx_IN pin input detection with digital filter (DFSNUM[1:0] = 2'b00, INPL=1, INDM=0)	229
Figure 10-1. Block diagram of CRC calculation unit	239
Figure 11-1. TMU block diagram	244
Figure 12-1. Block diagram of DMA	262
Figure 12-2. Handshake mechanism	264
Figure 12-3. DMA interrupt logic	266
Figure 13-1. Block diagram of DMAMUX	275
Figure 13-2. Synchronization mode	277
Figure 13-3. Event generation	278

Figure 15-1. ADC0 module block diagram	300
Figure 15-2. ADC2 module block diagram	301
Figure 15-3. Basic scan once operation of ADC0 (without channel-dedicated sample-and-hold circuits)	303
Figure 15-4. Basic scan once operation of ADC0 (with channel-dedicated sample-and-hold circuits, constant sampling disabled)	304
Figure 15-5. Basic scan once operation of ADC0 (with channel-dedicated sample-and-hold circuits, constant sampling enabled)	305
Figure 15-6. Channel selection and self-diagnosis in scan once mode of ADC2 (without channel-dedicated sample-and-hold circuits)	306
Figure 15-7. Channel selection and self-diagnosis in scan once mode of ADC0 (with channel-dedicated sample-and-hold circuits, constant sampling disabled)	308
Figure 15-8. Channel selection and self-diagnosis in scan once mode of ADC0 (with channel-dedicated sample-and-hold circuits, constant sampling enabled)	309
Figure 15-9. A/D conversion in Group_pri1 bifurcate trigger mode of ADC0	311
Figure 15-10. Basic continuous scan operation of ADC0 (without channel-dedicated sample-and-hold circuits)	312
Figure 15-11. Basic continuous scan operation of ADC0 (with channel-dedicated sample-and-hold circuits, constant sampling disabled)	313
Figure 15-12. Basic continuous scan operation of ADC0 (with channel-dedicated sample-and-hold circuits, constant sampling enabled)	314
Figure 15-13. Channel selection and self-diagnosis of continuous scan operation of ADC2 (without channel-dedicated sample-and-hold circuits)	315
Figure 15-14. Channel selection and self-diagnosis of continuous scan operation of ADC0 (with channel-dedicated sample-and-hold circuits, constant sampling disabled)	317
Figure 15-15. Channel selection and self-diagnosis of continuous scan operation of ADC2 (with channel-dedicated sample-and-hold circuits, constant sampling enabled)	318
Figure 15-16. Groups scan mode basic operation	320
Figure 15-17. Basic A/D conversion in bifurcate trigger mode (without channel-dedicated sample-and-hold circuits)	321
Figure 15-18. Basic A/D conversion in bifurcate trigger mode (with channel-dedicated sample-and-hold circuits)	322
Figure 15-19. Basic A/D conversion in bifurcate trigger mode rescan operation (without channel-dedicated sample-and-hold circuits)	322
Figure 15-20. Basic A/D conversion in bifurcate trigger mode rescan operation (with channel-dedicated sample-and-hold circuits)	323
Figure 15-21. Group rescan operation and group priority control in bifurcate trigger mode (without channel-dedicated sample-and-hold circuits)	323
Figure 15-22. A/D conversion in bifurcate trigger mode rescan operation under group priority control with group rescan enabled (with channel-dedicated sample-and-hold circuits)	324
Figure 15-23. Group_pri1 trigger input during Group_pri2 scan, with rescan setting (GP2RTCH = 1)	332
Figure 15-24. Group_pri1 trigger input during Group_pri2 scan, with rescan setting (GP2RTCH = 0)	332

Figure 15-25. Group_pri1 trigger input during Group_pri2 rescan, with rescan setting (GP2RTCH = 1)	333
Figure 15-26. Group_pri1 trigger input during Group_pri2 rescan, with rescan setting (GP2RTCH = 0)	333
Figure 15-27. Group_pri2 trigger input during Group_pri1 scan, with rescan setting (GP2RTEN = 1)	335
Figure 15-28. Group_pri1 trigger input during Group_pri2 scan, without rescan setting (GP2RTEN = 0)	336
Figure 15-29. Continuous single scan operation on Group_pri2(LPGCNT = 1, GP2RTCH = 1)	337
Figure 15-30. Alternative self-diagnosis (SDSFIX = 0) with rescan setting (GP2RTCH = 0)	337
Figure 15-31. Fixed self-diagnosis with rescan setting (GP2RTCH = 0)	337
Figure 15-32. High priority group trigger input during low-priority group scan, with rescan setting (GP2RTCH = 1, GP3RTCH = 1)	339
Figure 15-33. High priority group trigger input during low-priority group scan, with rescan setting (GP2RTCH = 0, GP3RTCH = 0)	339
Figure 15-34. High priority group trigger input during low-priority group scan, with rescan setting (GP2RTCH = 1, GP3RTCH = 1)	340
Figure 15-35. Low-priority group trigger input during priority group scan, with rescan setting (GPxRTEN = 1)	342
Figure 15-36. High priority group trigger input during low-priority group scan, without rescan setting (GPxRTEN = 0)	342
Figure 15-37. Continuous single scan operation on Group_pri3 (GP2RTCH = 1, GP3RTCH = 1)	344
Figure 15-38. Actual A/D conversion sequence of Group_pri1 in ADC2	345
Figure 15-39. Actual A/D conversion sequence of Group_pri2 in ADC2	345
Figure 15-40. Data storage mode of 12-bit resolution	347
Figure 15-41. Data storage mode of 10-bit resolution	347
Figure 15-42. Data storage mode of 8-bit resolution	347
Figure 15-43. Data storage mode of 6-bit resolution	348
Figure 15-44. Self-Diagnosis Data storage mode of 12-bit resolution	348
Figure 15-45. Self-Diagnosis Data storage mode of 10-bit resolution	348
Figure 15-46. Self-Diagnosis Data storage mode of 8-bit resolution	348
Figure 15-47. Self-Diagnosis Data storage mode of 6-bit resolution	349
Figure 15-48. Example of a DMA transfer of Group_pri1 in ADC2	355
Figure 15-49. Example of a DMA transfer of Group_pri1 (bifurcate trigger function enabled)	356
Figure 15-50. Example of an overflow event of Group_pri2 in ADC2	357
Figure 15-51. Operation of A/D conversion when the disconnection detection assist function is used	359
Figure 15-52. High priority group trigger input during low-priority group rescan	361
Figure 16-1. DAC block diagram	403
Figure 16-2. DAC LFSR algorithm	406
Figure 16-3. DAC triangle noise wave	406
Figure 16-4. Prevent Interference between D/A and A/D	407
Figure 17-1. CMP block diagram	417

Figure 17-2. CMP hysteresis	419
Figure 17-3. The CMP output signal filtering	420
Figure 18-1. EVIC block diagram	433
Figure 18-2. Example of single I/O pin output.....	438
Figure 18-3. Example of group E I/O pin input loading	439
Figure 18-4. Example of group I/O pin output (GRPEOSEL[2:0] = 3b'011)	439
Figure 18-5. group output port rotate mode	440
Figure 19-1. SVPWM module block diagram	466
Figure 19-2. SVPWM output the basic voltage vector	467
Figure 19-3. Mean space vector modulation	468
Figure 19-4. Five-segment SVPWM waveform	469
Figure 19-5. Five-segment SVPWM algorithm	470
Figure 19-6. Seven-segment SVPWM waveform.....	471
Figure 19-7. Seven-segment SVPWM algorithm	472
Figure 20-1. Free watchdog block diagram	480
Figure 20-2. Window watchdog timer block diagram	487
Figure 20-3. Window watchdog timing diagram	489
Figure 21-1. Advanced timer block diagram	495
Figure 21-2. Normal mode, internal clock divided by 1	496
Figure 21-3. Counter timing diagram with prescaler division change from 1 to 2	497
Figure 21-4. Timing diagram of up counting mode, PSC=0/2.....	498
Figure 21-5. Timing diagram of up counting mode, change TIMEx_CAR ongoing (ARSE=0, CAR value updates immediately. ARSE=1, CARUPSEL =2'b00, CAR value updates when an update event occurs.).....	498
Figure 21-6. Timing diagram of down counting mode, PSC=0/2.....	500
Figure 21-7. Timing diagram of down counting mode, change TIMEx_CAR ongoing (ARSE=0, CAR value updates immediately. ARSE=1, CARUPSEL =2'b00, CAR value updates when an update event occurs.).....	500
Figure 21-8. Timing diagram of center-aligned counting mode	503
Figure 21-9. Update repetition counter timing diagram of center-aligned counting mode	504
Figure 21-10. Update repetition counter timing diagram of up counting mode	505
Figure 21-11. Update repetition counter timing diagram of down counting mode	505
Figure 21-12. Interrupt repetition counter timing diagram of center-aligned counting mode	506
Figure 21-13. Interrupt repetition counter timing diagram of up counting mode	506
Figure 21-14. Interrupt repetition counter timing diagram of down counting mode	507
Figure 21-15. Timing diagram for updates in global update mode on update events	508
Figure 21-16. Input capture logic for channel 0	509
Figure 21-17. Input capture logic for multi mode channel 0	509
Figure 21-18. Output compare logic (when MCHxMSEL = 2'b00, x=0, 1, 2, 3)	511
Figure 21-19. Output compare logic (when MCHxMSEL = 2'b11, x=0,1,2,3)	511
Figure 21-20. Output-compare in three modes	513
Figure 21-21. Timing diagram of EAPWM	514
Figure 21-22. Timing diagram of CAPWM.....	514
Figure 21-23. Channel x output PWM with (CHxVAL < CHxCOMVAL_ADD)	516

Figure 21-24. Channel x output PWM with (CHxVAL = CHxCOMVAL_ADD)	517
Figure 21-25. Channel x output PWM with (CHxVAL > CHxCOMVAL_ADD)	517
Figure 21-26. Channel x output PWM with CHxVAL or CHxCOMVAL_ADD exceeds CARL	517
Figure 21-27. Channel x output PWM duty cycle changing with CHxCOMVAL_ADD	518
Figure 21-28. Four Channels outputs in Composite PWM mode	519
Figure 21-29. Channel x output PWM with (CHxVAL < CHxCOMVAL_ADD / CHxCOMVAL_ADD < CHxVAL)	520
Figure 21-30. Channel x output PWM with (CHxVAL = CHxCOMVAL_ADD)	522
Figure 21-31. Channel x output PWM with CHxVAL or CHxCOMVAL_ADD exceeds CARL	522
Figure 21-32. CHx_O output with a pulse in edge-aligned mode (CHxOMPSEL#2'b00)	524
Figure 21-33. CHx_O output with a pulse in center-aligned mode (CHxOMPSEL#2'b00)	524
Figure 21-34. Complementary output with dead time insertion	528
Figure 21-35. BREAK function logic diagram	529
Figure 21-36. Output behavior of the channel in response to BREAK (the break input high active and IOS=1)	529
Figure 21-37. Example of counter operation in decoder interface mode	531
Figure 21-38. Example of decoder interface mode with CI0FE0 polarity inverted	531
Figure 21-39. Example of counter operation in non-quadrature decoder mode 0 / 1 with CH1P=0	532
Figure 21-40. Example of counter operation in non-quadrature decoder mode 2 / 3 (CH0P / CH1P=0)	533
Figure 21-41. Hall sensor is used for BLDC motor	535
Figure 21-42. Hall sensor timing between two timers	535
Figure 21-43. Restart mode	536
Figure 21-44. Pause mode	537
Figure 21-45. Event mode	537
Figure 21-46. Single pulse mode TIMERx_CHxCV=0x04, TIMERx_CAR=0x60	538
Figure 21-47. Trigger mode of TIMER0 controlled by enable signal of TIMER2	539
Figure 21-48. Trigger mode of TIMER0 controlled by update signal of TIMER2	539
Figure 21-49. Pause mode of TIMER0 controlled by enable signal of TIMER2	540
Figure 21-50. Pause mode of TIMER0 controlled by O0CPRE signal of TIMER2	541
Figure 21-51. Trigger TIMER0 and TIMER2 by the CI0 signal of TIMER2	542
Figure 21-52. ADC trigger singal operation in ADC compare match trigger mode(ADT1UPGTEN=1、ADT2UPGTEN=1、ADT1DWGTEN=1、ADT2DWGTEN=0)	543
Figure 21-53. ADC repetition counting function in ADC trigger mode(ADT1UPGTEN =1、ADT2UPGTEN =1、ADT1DWGTEN =1、ADT2DWGTEN =1)	544
Figure 21-54. Timing diagram of ADC trigger synchronization monitor signal (ADCSM1SEL =3'b010)	545
Figure 21-55. General Level 0 timer block diagram	608
Figure 21-56. Normal mode, internal clock divided by 1	609
Figure 21-57. Counter timing diagram with prescaler division change from 1 to 2	610
Figure 21-58. Timing chart of up counting mode, PSC=0/2	611
Figure 21-59. Timing chart of up counting, change TIMERx_CAR ongoing	611
Figure 21-60. Timing chart of down counting mode, PSC=0/2	612

Figure 21-61. Timing chart of down counting mode, change <code>TIMERx_CAR</code> ongoing	613
Figure 21-62. Timing chart of center-aligned counting mode	614
Figure 21-63. Input capture logic.....	615
Figure 21-64. Output compare logic ($x=0,1,2,3$)	616
Figure 21-65. Output-compare under three modes	617
Figure 21-66. Timing chart of EAPWM	618
Figure 21-67. Timing chart of CAPWM	618
Figure 21-68. Restart mode.....	620
Figure 21-69. Pause mode.....	621
Figure 21-70. Event mode	621
Figure 21-71. Single pulse mode <code>TIMERx_CHxCV = 0x04</code> , <code>TIMERx_CAR=0x60</code>	622
Figure 22-1. General purpose timer block diagram.....	647
Figure 22-2. Normal mode, internal clock divided by 1	648
Figure 22-3. Counter timing diagram with prescaler division change from 1 to 2	649
Figure 22-4. Timing diagram of up counting mode, <code>PSC=0/2</code>	651
Figure 22-5. Timing diagram of up counting mode, change <code>GPTIMERx_CAR</code> ongoing	651
Figure 22-6. Timing diagram of down counting mode, <code>PSC=0/2</code>	652
Figure 22-7. Timing diagram of down counting mode, change <code>GPTIMERx_CAR</code> ongoing	653
Figure 22-8. Timing diagram of center-aligned counting mode	654
Figure 22-9. Repetition counter timing diagram of center-aligned counting mode	656
Figure 22-10. Repetition counter timing diagram of up counting mode	656
Figure 22-11. Repetition counter timing diagram of down counting mode	657
Figure 22-12. Timing diagram for updates in non-global one-time reload mode on update events	657
Figure 22-13. Timing diagram for updates in global one-time reload mode on update events (<code>GLBUPSEL[1:0] ≠ 2'b11</code>)	658
Figure 22-14. Timing diagram for updates in global one-time reload mode on update events (<code>GLBUPSEL[1:0] = 2'b11</code>)	659
Figure 22-15. Input capture logic for channel 0	660
Figure 22-16. Output compare logic	661
Figure 22-17. Output-compare in three modes	662
Figure 22-18. Timing diagram of composite PWM mode	663
Figure 22-19. Timing diagram of asymmetric PWM mode	663
Figure 22-20. Schematic of symmetric and asymmetric dead time insertion	664
Figure 22-21. Simultaneous operation by software with no initial value	666
Figure 22-22. Simultaneous operation by software with initial value.....	666
Figure 22-23. Simultaneous operation by hardware source	667
Figure 22-24. Simultaneous reset source by channel	668
Figure 22-25. Single pulse mode <code>GPTIMERx_CHxCV=0x04</code> , <code>GPTIMERx_CAR=0x60</code> , <code>CH0COMUP[1:0] = 2'b10</code>	668
Figure 22-26. Example of interrupt skipping in center alignment mode (skip value is 2)	670
Figure 22-27. Example of interrupt skipping in edge alignment mode (changing direction and skip value is 2).....	670
Figure 22-28. ADC trigger signal operation in ADC compare match trigger	

mode(ADT1UPGTEN=1、ADT2UPGTEN=1、ADT1DWGTEN=1、ADT2DWGTEN=0).....	671
Figure 22-29. Timming diagram of ADC trigger synchronization monitor signal (ADCSM3SEL = 0)	672
Figure 22-30. Channel output after reset	674
Figure 23-1. Compare timer W block diagram	723
Figure 23-2. Timing chart of PSC value change from 0 to 2, CAR = 99	724
Figure 23-3. Timing chart of up counting mode, PSC=0/2	725
Figure 23-4. Channel input capture principle.....	726
Figure 23-5. Output-compare under two modes	727
Figure 24-1. Compare timer block diagram	739
Figure 24-2. Timing chart of CNTxPSC value change from 0 to 2, CNTxCAR = 99	740
Figure 24-3. Counter x timing chart of up counting mode, CNTxPSC=0/2	741
Figure 25-1. UART module block diagram	750
Figure 25-2. UART character frame (8 bits data and 1 stop bit)	751
Figure 25-3. UART transmit procedure	753
Figure 25-4. Receiving a frame bit by oversampling method (OSB=0)	754
Figure 25-5. Configuration step when use DMA for UART transmission	755
Figure 25-6. Configuration step when use DMA for UART reception	756
Figure 25-7. Break frame occurs during idle state	757
Figure 25-8. Break frame occurs during a frame	757
Figure 25-9. IrDA SIR ENDEC module	758
Figure 25-10. IrDA data modulation	759
Figure 25-11. UART interrupt mapping diagram	760
Figure 26-1. I2C module block diagram	772
Figure 26-2. Data validation	773
Figure 26-3. START and STOP signal	774
Figure 26-4. I2C communication flow with 10-bit address (Master Transmit)	774
Figure 26-5. I2C communication flow with 7-bit address (Master Transmit)	775
Figure 26-6. I2C communication flow with 7-bit address (Master Receive)	775
Figure 26-7. I2C communication flow with 10-bit address (Master Receive when HEAD10R=0)	775
Figure 26-8. I2C communication flow with 10-bit address (Master Receive when HEAD10R=1)	775
Figure 26-9. Data hold time	776
Figure 26-10. Data setup time	776
Figure 26-11. Data transmission	778
Figure 26-12. Data reception	779
Figure 26-13. I2C initialization in slave mode	782
Figure 26-14. Programming model for slave transmitting when SS=0	783
Figure 26-15. Programming model for slave transmitting when SS=1	784
Figure 26-16. Programming model for slave receiving	785
Figure 26-17. I2C initialization in master mode	786
Figure 26-18. Programming model for master transmitting (N<=255)	787
Figure 26-19. Programming model for master transmitting (N>255)	788

Figure 26-20. Programming model for master receiving (N<=255)	789
Figure 26-21. Programming model for master receiving (N>255)	790
Figure 26-22. SMBus master transmitter and slave receiver communication flow	794
Figure 26-23. SMBus master receiver and slave transmitter communication flow	794
Figure 27-1. Block diagram of SPI	813
Figure 27-2. SPI timing diagram in normal mode	814
Figure 27-3. SPI timing diagram in Quad-SPI mode (CKPL=1, CKPH=1, LF=0)	815
Figure 27-4. A typical full-duplex connection	817
Figure 27-5. A typical simplex connection (Master: Receive, Slave: Transmit)	818
Figure 27-6. A typical simplex connection (Master: Transmit only, Slave: Receive)	818
Figure 27-7. A typical bidirectional connection	818
Figure 27-8. Timing diagram of TI master mode with discontinuous transfer	820
Figure 27-9. Timing diagram of TI master mode with continuous transfer	820
Figure 27-10. Timing diagram of TI slave mode	821
Figure 27-11. Timing diagram of NSS pulse with continuous transmit	822
Figure 27-12. Timing diagram of quad write operation in Quad-SPI mode	823
Figure 27-13. Timing diagram of quad read operation in Quad-SPI mode	824
Figure 28-1. CAN module block diagram	837
Figure 28-2. Transmission register	839
Figure 28-3. State of transmit mailbox	840
Figure 28-4. Reception register	841
Figure 28-5. 32-bit filter	842
Figure 28-6. 16-bit filter	843
Figure 28-7. 32-bit mask mode filter	843
Figure 28-8. 16-bit mask mode filter	843
Figure 28-9. 32-bit list mode filter	843
Figure 28-10. 16-bit list mode filter	843
Figure 28-11. The bit time	847

List of Tables

Table 1-1. Bus Interconnection Matrix	32
Table 1-2. Memory map of GD32M531 devices	35
Table 1-3. ECC interrupt enable and flag	38
Table 1-4. Boot modes	39
Table 2-1. GD32M531 base address and size for flash memory	69
Table 2-2. The relation between WSCNT and AHB clock frequency	70
Table 2-3. Option bytes	81
Table 2-4. OB_WP bit for pages protected	85
Table 3-1. Power saving mode summary	101
Table 4-1. Clock source select	117
Table 5-1. CFMU interrupt requests	145

Table 6-1. NVIC exception types in Cortex®-M33	151
Table 6-2. Interrupt vector table	151
Table 6-3. EXTI source	155
Table 7-1. GPIO configuration table	162
Table 8-1. The target timers and channels for output disable	185
Table 8-2. Complementary PWM channel for output detection	187
Table 8-3. Disabling request selection for target timer	189
Table 9-1. GPTIMER channels for output closing control	228
Table 11-1. Input data configuration	245
Table 11-2. Output data configuration	246
Table 11-3 IEEE754 32-bit single precision floating-point format	246
Table 11-4. TMU mode configuration	247
Table 11-5. Mode 0 description, when IFLTEN = 1 and OFLTEN = 1	247
Table 11-6. Mode 0 description, when IFLTEN = 0 or OFLTEN = 0	248
Table 11-7. Mode 1 description, when IFLTEN = 1 and OFLTEN = 1	249
Table 11-8. Mode 1 description, when IFLTEN = 0 or OFLTEN = 0	250
Table 11-9. Mode 2 description, when IFLTEN = 1 and OFLTEN = 1	251
Table 11-10. Mode 2 description, when IFLTEN = 0 or OFLTEN = 0	251
Table 11-11. Mode 3 description, when IFLTEN = 1 and OFLTEN = 1	252
Table 11-12. Mode 3 description, when IFLTEN = 0 or OFLTEN = 0	253
Table 11-13. Mode 4 description, when IFLTEN = 1 and OFLTEN = 1	254
Table 11-14. Mode 4 description, when IFLTEN = 0 or OFLTEN = 0	254
Table 11-15. Recommended scaling factors in mode 4	254
Table 11-16. TMU instruction	255
Table 12-1. DMA transfer operation	263
Table 12-2. interrupt events	266
Table 13-1. DMAMUX signals	276
Table 13-2. Interrupt events	280
Table 13-3. Request multiplexer input mapping	281
Table 13-4. Trigger input mapping	283
Table 13-5. Synchronization input mapping	284
Table 14-1. Pin assignment	292
Table 15-1. ADC internal signals	299
Table 15-2. ADC pins definition	299
Table 15-3. Control of scanning operations for groups according to the settings of the GPxRTEN bit (GPRIEN = 1, GP3EN = 1, GP4EN = 1)	326
Table 15-4. Group priority operation setting and operating mode for Groups_pri1/pri2 (GPRIEN = 1, GP3EN = 0, GP4EN = 0)	328
Table 15-5. Group priority operation setting and operating mode for three Groups_pri1/pri2/pri3 (GPRIEN = 1, GP3EN = 1, GP4EN = 0)	328
Table 15-6. Group priority operation setting and operating mode for four Groups(pri1/pri2/pri3/pri4) (GPRIEN = 1, GP3EN = 1, GP4EN = 1)	329
Table 15-7. Example of channel priority setting in ADC2	345
Table 15-8. ADC synchronous/asynchronous trigger enable	354

Table 15-9. External trigger edge	354
Table 15-10. ADC synchronous trigger source	354
Table 15-11. ADC asynchronous trigger source	355
Table 15-12. t_{CONV} timings depending on resolution	359
Table 16-1. DAC I/O description	403
Table 16-2. DAC triggers and outputs summary	403
Table 16-3. Triggers of DAC	404
Table 16-4. Conversion mode of DAC	404
Table 17-1. CMP inputs and outputs summary	418
Table 18-1. Event source bit fields selection	434
Table 19-1. The correspondence between the sector and the time	469
Table 20-1. Min/max FWDGT timeout period at 32 kHz (IRC32K)	481
Table 20-2. Min/max timeout value at 90 MHz (f_{PCLK1})	489
Table 21-1. Timers (TIMERx) are divided into two sorts	493
Table 21-2. Advanced timer channel description	495
Table 21-3. The Composite PWM pulse width	515
Table 21-4. The Asymmetric PWM pulse width	520
Table 21-5. Complementary outputs controlled by parameters (MCHxMSEL = 2'b11)	527
Table 21-6. Counting direction in different quadrature decoder signals	531
Table 21-7. the counter operation in non-quadrature decoder mode 1	532
Table 21-8. the counter operation in non-quadrature decoder mode 2 / 3	533
Table 21-9. Examples of slave mode	536
Table 21-10. TIMER0/TIMER7 ADC trigger signals and their operation	543
Table 21-11. Examples of slave mode	619
Table 22-1. Schematic diagram of the waveform output	665
Table 22-2. GPTIMER0/1 ADC trigger signals and their operation	672
Table 25-1. Description of UART important pins	750
Table 25-2. Configuration of stop bits	751
Table 25-3. UART interrupt requests	759
Table 26-1. Definition of I2C-bus terminology (refer to the I2C specification of Philips semiconductors)	772
Table 26-2. Data setup time and data hold time	777
Table 26-3. Communication modes to be shut down	779
Table 26-4. I2C error flags	795
Table 26-5. I2C interrupt events	796
Table 27-1. SPI signal description	813
Table 27-2. Quad-SPI signal description	814
Table 27-3. NSS function in slave mode	815
Table 27-4. NSS function in master mode	816
Table 27-5. SPI operating modes	816
Table 27-6. SPI interrupt requests	827
Table 28-1. 32-bit filter number	844
Table 28-2. Filtering index	845
Table 28-3. CAN Event / Interrupt flags	849

Table 29-1. List of abbreviations used in register	870
Table 29-2. List of terms	870
Table 30-1. Revision history.....	872

1 System and memory architecture

The devices of GD32M531 series are 32-bit general-purpose microcontrollers based on the Arm® Cortex® processor. The Arm® Cortex®-M33 processor includes two AHB buses known as Code and System buses. All memory accesses of the Arm® Cortex®-M33 processor are executed on these two buses according to the different purposes and the target memory spaces. The memory organization uses a Harvard architecture, pre-defined memory map and up to 4 GB of memory space, making the system flexible and extendable.

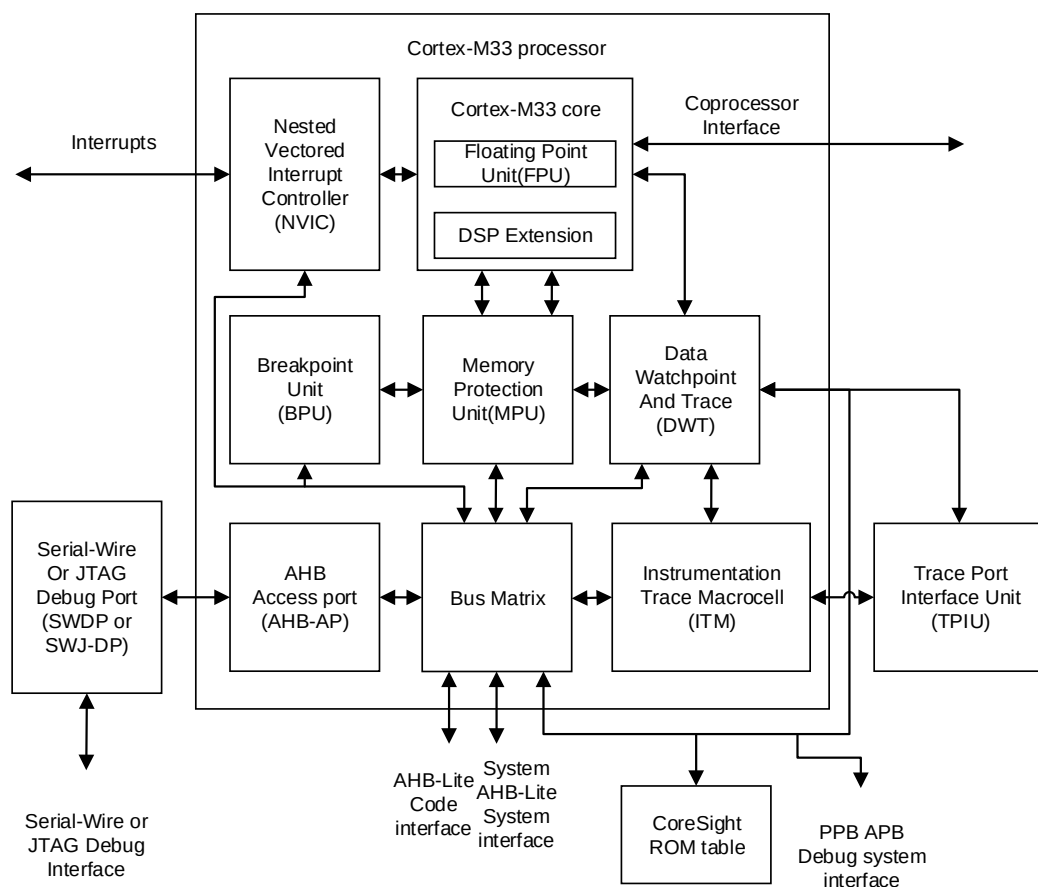
1.1 Arm® Cortex®-M33 processor

The Cortex®-M33 processor is a 32-bit processor that possesses low interrupt latency and low-cost debug. The characteristics of integrated and advanced make the Cortex®-M33 processor suitable for market products that require microcontrollers with high performance and low power consumption. The Cortex®-M33 processor is based on the Armv8 architecture and supports a powerful and scalable instruction set including general data processing I/O control tasks, advanced data processing bit field manipulations and DSP. Some system peripherals listed below are also provided by Cortex®-M33:

- Internal Bus Matrix connected with Code bus, System bus, and Private Peripheral Bus (PPB) and debug accesses.
- Nested Vectored Interrupt Controller (NVIC).
- Breakpoint Unit (BPU).
- Data Watchpoint and Trace (DWT).
- Instrumentation Trace Macrocell (ITM).
- Serial Wire JTAG Debug Port (SWJ-DP).
- Trace Port Interface Unit (TPIU).
- Memory Protection Unit (MPU)
- Floating Point Unit (FPU).
- DSP Extension (DSP).
- Support External coprocessor for TMU.

[Figure 1-1. The structure of the Cortex®-M33 processor](#) shows the Cortex®-M33 processor block diagram. For more information, please refer to the Arm® Cortex®-M33 Technical Reference Manual.

Figure 1-1. The structure of the Cortex®-M33 processor



1.2 System architecture

A 32-bit multilayer bus is implemented in the GD32M531 devices, which enables parallel access paths between multiple masters and slaves in the system. The multilayer bus consists of an AHB interconnect matrix, two AHB bus and two APB buses. The interconnection relationship of the AHB interconnect matrix is shown below. In the following table, “1” indicates the corresponding master is able to access the corresponding slave through the AHB interconnect matrix, while the blank means the corresponding master cannot access the corresponding slave through the AHB interconnect matrix.

This architecture is shown in [Table 1-1. Bus Interconnection Matrix](#).

Table 1-1. Bus Interconnection Matrix

	CBUS	SBUS	DMA0	DMA1
FMC	1	0	1	1
SRAM	1	1	1	1
AHB1	0	1	1	1
AHB2	0	1	1	1

	CBUS	SBUS	DMA0	DMA1
APB1	0	1	1	1
APB2	0	1	1	1

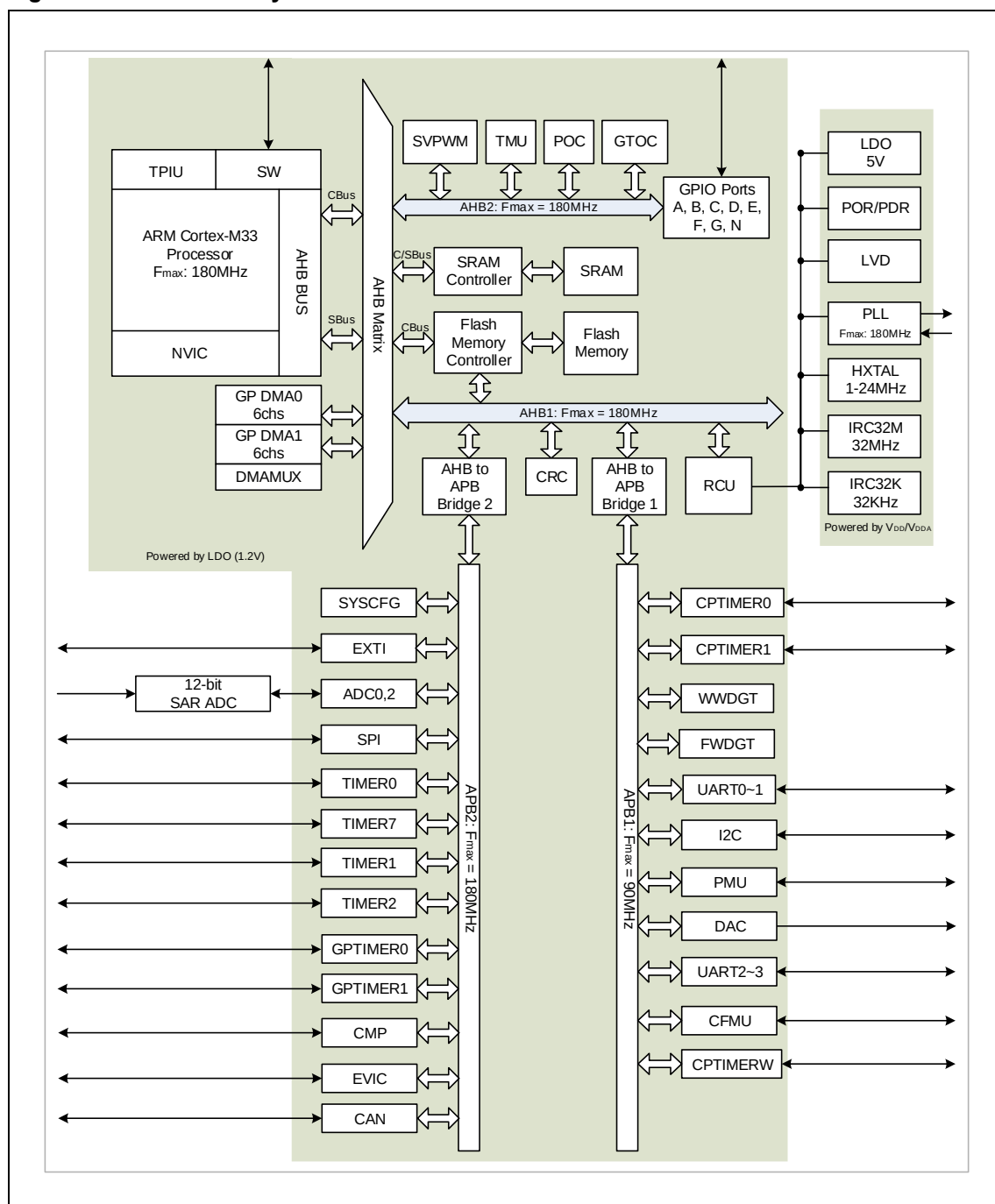
As is shown above, there are several masters connected with the AHB interconnect matrix, including CBUS, SBUS, DMA0 and DMA1. CBUS is the code bus of the Cortex®-M33 core, which is used for any instruction fetch and data access to the Code region. Similarly, SBUS is the system bus of the Cortex®-M33 core, which is used for instruction/vector fetches, data loading / storing and debugging access of the system regions. The system regions include the internal SRAM region and the Peripheral region. DMA0 and DMA1 are the buses of DMA0 and DMA1 respectively.

There are also several slaves connected with the AHB interconnect matrix, including FMC, SRAM, AHB1, AHB2, APB1, APB2. FMC is the bus interface of the flash memory controller. SRAM is on-chip static random access memories. AHB1 is the AHB bus connected with all of the AHB slaves except GPIO. AHB2 is the AHB bus connected with GPIO. While APB1 and APB2 are the two APB buses connected with all of the APB slaves.

The two APB buses connect with all the APB peripherals. APB1 is up to 90MHz, APB2 operates at full speed (up to 180MHz depending on the device).

These are interconnected using a multilayer AHB bus architecture as shown in [Figure 1-2. GD32M531 system architecture](#) below:

Figure 1-2. GD32M531 system architecture



1.3 Memory map

The Arm® Cortex®-M33 processor is structured in Harvard architecture which can use separate buses to fetch instructions and load/store data. Program memory, data memory, registers and I/O ports are organized within the same linear 4-Gbyte address space which is the maximum address range of the Cortex®-M33 since the bus address width is 32-bit. Additionally, a pre-defined memory map is provided by the Cortex®-M33 processor to reduce the software complexity of repeated implementation of different device vendors. In the map,

some regions are used by the Arm® Cortex®-M33 system peripherals which can not be modified. However, the other regions are available to the vendors. [Table 1-2. Memory map of GD32M531 devices](#) shows the memory map of the GD32M531 devices, including Code, SRAM, peripheral, and other pre-defined regions. Almost each peripheral is allocated 1KB of space. This allows simplifying the address decoding for each peripheral.

Table 1-2. Memory map of GD32M531 devices

Pre-defined Regions	Bus	Address	Peripherals
External RAM		0x6000 0000 - 0xDFFF FFFF	Reserved
Peripheral	AHB2	0x5000 0000 - 0x5FFF FFFF	Reserved
		0x4802 4800 - 0x4FFF FFFF	Reserved
		0x4802 4400 - 0x4802 47FF	TMU
		0x4802 4000 - 0x4802 43FF	SVPWM
		0x4800 4C00 - 0x4802 3FFF	Reserved
		0x4800 4800 - 0x4800 4BFF	GTOC
		0x4800 4400 - 0x4800 47FF	POC
		0x4800 4000 - 0x4801 43FF	GPION
		0x4800 1C00 - 0x4800 3FFF	Reserved
		0x4800 1800 - 0x4800 1BFF	GPIOG
		0x4800 1400 - 0x4800 17FF	GPIOF
		0x4800 1000 - 0x4800 13FF	GPIOE
		0x4800 0C00 - 0x4800 0FFF	PIOD
		0x4800 0800 - 0x4800 0BFF	PIOC
		0x4800 0400 - 0x4800 07FF	PIOB
		0x4800 0000 - 0x4800 03FF	PIOA
	AHB1	0x4002 3400 - 0x47FF FFFF	Reserved
		0x4002 3000 - 0x4002 33FF	CRC
		0x4002 2400 - 0x4002 2FFF	Reserved
		0x4002 2000 - 0x4002 23FF	FMC
		0x4002 1400 - 0x4002 1FFF	Reserved
		0x4002 1000 - 0x4002 13FF	RCU
		0x4002 0C00 - 0x4002 0FFF	Reserved
		0x4002 0800 - 0x4002 0BFF	DMAMUX
		0x4002 0400 - 0x4002 07FF	DMA1
		0x4002 0000 - 0x4002 03FF	DMA0
	APB2	0x4001 B000 - 0x4001 FFFF	Reserved
		0x4001 A000 - 0x4001 AFFF	CAN
		0x4001 8800 - 0x4001 9FFF	Reserved
		0x4001 8400 - 0x4001 87FF	EVIC
		0x4001 8000 - 0x4001 83FF	Reserved
		0x4001 7C00 - 0x4001 7FFF	CMP
		0x4001 6800 - 0x4001 7BFF	Reserved

Pre-defined Regions	Bus	Address	Peripherals
		0x4001 6100 - 0x4001 67FF	GPTIMER1
		0x4001 6000 - 0x4001 60FF	GPTIMER0
		0x4001 4800 - 0x4001 5FFF	Reserved
		0x4001 4400 - 0x4001 47FF	TIMER2
		0x4001 4000 - 0x4001 43FF	TIMER1
		0x4001 3800 - 0x4001 7BFF	Reserved
		0x4001 3400 - 0x4001 37FF	TIMER7
		0x4001 3000 - 0x4001 33FF	SPI
		0x4001 2C00 - 0x4001 2FFF	TIMER0
		0x4001 2800 - 0x4001 2BFF	Reserved
		0x4001 2400 - 0x4001 27FF	ADC2
		0x4001 2000 - 0x4001 23FF	ADC0
		0x4001 0800 - 0x4001 1FFF	Reserved
		0x4001 0400 - 0x4001 07FF	EXTI
		0x4001 0000 - 0x4001 03FF	SYSCFG
	APB1	0x4000 E400 - 0x4000 FFFF	Reserved
		0x4000 E000 - 0x4000 E3FF	CPTIMERW
		0x4000 CC00 - 0x4000 DFFF	Reserved
		0x4000 C800 - 0x4000 CBFF	CFMU
		0x4000 8000 - 0x4000 C7FF	Reserved
		0x4000 7C00 - 0x4000 7FFF	UART3
		0x4000 7800 - 0x4000 7BFF	UART2
		0x4000 7400 - 0x4000 77FF	DAC
		0x4000 7000 - 0x4000 73FF	PMU
		0x4000 5800 - 0x4000 6FFF	Reserved
		0x4000 5400 - 0x4000 57FF	I2C
		0x4000 5000 - 0x4000 53FF	UART1
		0x4000 4C00 - 0x4000 4FFF	UART0
		0x4000 3400 - 0x4000 4BFF	Reserved
		0x4000 3000 - 0x4000 33FF	FWDGT
		0x4000 2C00 - 0x4000 2FFF	WWDGT
		0x4000 0800 - 0x4000 2BFF	Reserved
		0x4000 0400 - 0x4000 07FF	CPTIMER1
		0x4000 0000 - 0x4000 03FF	CPTIMER0
SRAM		0x2000 8000 - 0x3FFF FFFF	Reserved
		0x2000 0000 - 0x2000 7FFF	SRAM (32KB)
Code		0x1FFF F820 - 0x1FFF FFFF	Reserved
		0x1FFF F800 - 0x1FFF F81F	Option Bytes (32B)
		0x1FFF E000 - 0x1FFF F7FF	System Memory (6KB)
		0x1FFF 7200 - 0x1FFF DFFF	Reserved

Pre-defined Regions	Bus	Address	Peripherals
		0x1FFF 7000~0x1FFF 71FF	OTP (512B)
		0x1000 8000 - 0x1FFF 6FFF	Reserved
		0x1000 0000 - 0x1000 7FFF	SRAM Aliased
		0x0881 0000 - 0x0FFF FFFF	Reserved
		0x0880 0000 - 0x0880 FFFF	Data Flash (64KB)
		0x0804 0000 - 0x0807 FFFF	Reserved
		0x0800 0000 - 0x0803 FFFF	Main Flash (256KB)
		0x0002 6000 - 0x07FF FFFF	Reserved
		0x0000 0000 - 0x0002 5FFF	Aliased to Flash or system memory

1.3.1 On-chip SRAM memory

The GD32M531 series of devices contain up to 32KB of on-chip SRAM which starts at the address 0x2000 0000. It supports byte (8 bits), half-word (16 bits), and word (32 bits) accesses.

SRAM_REMAP_SEL[1:0] in SYSCFG_SRAMCFG register can be used to control the size of SRAM aliased in 0x1000 0000 (remap of SRAM in 0x2000 0000). The CPU can access and SRAM aliased in 0x1000 0000 via CBUS instead of SBUS.

ECC

When reading and writing SRAM, it supports 7-bit ECC function. It can correct 1 bit error and detect multiple bits (multi-bit) error.

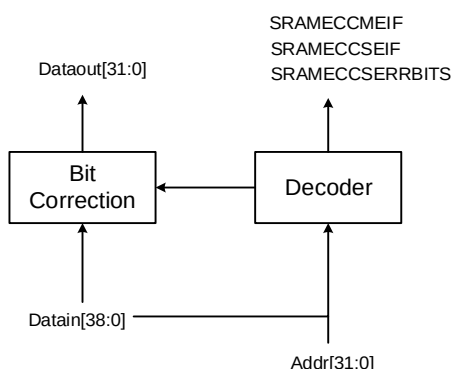
It must be written before reading SRAM, otherwise it may cause ECC error. Unaligned read operations will be performed in accordance with 32-bit read operations. Non-aligned write operations will produce a read-modify-write process. For example, when 16-bit data is written into SRAM, firstly the another 16-bit data is read out from the SRAM, and the 16-bit that need to be written are combined to a 32-bit data, and finally the 32-bit data is written into the SRAM together. Therefore, when initializing SRAM, it can only be written in a 32-bit width.

The ECC module is composed of an encoder and a decoder.

Encoder: When performing a SRAM write operation, a 7-bit ECC code will be generated and written into the SRAM together with the data.

Decoder: When performing a SRAM read operation, it uses the same algorithm as the encoder to decode and generate a 7-bit ECC code. The ECC code includes ECC error status and information which specific bit of the 32-bit data has single-bit error.

The decoder is shown in the figure [Figure 1-3. ECC decoder](#) below:

Figure 1-3. ECC decoder


EEIC

The EEIC (ECC Error Interrupt Control) module provides the function of ECC error status management and ECC interrupt configuration.

Table 1-3. ECC interrupt enable and flag

ECC	Interrupt enable	Event flag	Lock connect enable
SRAM	ECCSEIE0	ECCSEIF0	-
	ECCMEIE0	ECCMEIF0	SRAM_LOCK
CAN receive FIFO	ECCSEIE1	ECCSEIF1	-
	ECCMEIE1	ECCMEIF1	CANRX_LOCK
CAN fillter	ECCSEIE2	ECCSEIF2	-
	ECCMEIE2	ECCMEIF2	CANFILTER_LOCK
Flash SRAM	ECCSEIE3	ECCSEIF3	-
	ECCMEIE3	ECCMEIF3	FLASHSRAM_LOCK
Flash	ECCMEIE4	ECCMEIF4	FLASH_LOCK

Single-bit correction error event

When a single-bit correction error event is detected in SRAM / CAN receive FIFO / CAN fillter / Flash SRAM, EEIC:

- (1) The ECCSEIFx(x = 0..3) bit in SYSCFG_STAT register will be set. Software can clear it by writing 1.
- (2) The ECCEADDR[12:0] and ECCSEBIT[5:0] bit in SYSCFG_CFG2 records the address and bit where the single-bit error event occurred.

Multi-bit non-correction error event

When a multi-bit error event is detected in SRAM / CAN receive FIFO / CAN fillter / Flash SRAM / Flash, EEIC:

- (1) The ECCMEIFx(x = 0..4) bit in SYSCFG_STAT register will be set. Software can clear it

by writing 1.

- (2) The ECCEADDR[12:0] bit in SYSCFG_CFG2 records the SRAM address where the multi-bit error event occurred.

Single-bit correction error interrupt

Set the ECCSEIEx(x = 0..3) bit. When a single-bit error event is detected, a corresponding interrupt will be generated.

Multi-bit non-correction error interrupt

Set the ECCMEIEx(x = 0..4) bit. When a multi-bit error event is detected, a NMI interrupt will be generated.

1.3.2 On-chip flash memory overview

The devices provide high density on-chip flash memory, which is organized as follows:

- Up to 256KB of main flash memory.
- Up to 64KB of data flash memory.
- Up to 512B OTP(One Time Programmable) memory for user data storage.
- Up to 6KB of information blocks for the bootloader.
- 32B Option bytes to configure the device.

Refer to [Flash memory controller \(FMC\)](#) Chapter for more details.

1.4 Boot configuration

The GD32M531 devices provide two kinds of boot sources which can be selected by the BOOT-PN6. The details are shown in the following table. The value on the pin is latched on the 4th rising edge of CK_SYS after a reset. It is up to the user to set the BOOT-PN6 pin after a power-on reset or a system reset to select the required boot source. Once the pin has been sampled, it is free and can be used for other purposes.

Table 1-4. Boot modes

Selected boot source	Boot mode selection pin
	BOOT-PN6
System Memory	0
Main Flash Memory	1

After power-on sequence or a system reset, the Arm® Cortex®- M33 processor fetches the top-of-stack value from address 0x0000 0000 and the base address of boot code from 0x0000 0004 in sequence. Then, it starts executing code from the base address of boot code.

According to the selected boot source, either the main flash memory (original memory space beginning at 0x0800 0000) or the system memory (original memory space beginning at 0x1FFF E000) is aliased in the boot memory space which begins at the address 0x0000 0000.

The embedded bootloader is located in the system memory, which is used to reprogram the Flash memory.

1.5 System configuration controller

The main purposes of the system configuration controller (SYSCFG) are the following:

- I2C Fm+ configuration.
- Managing the external interrupt line connection to the GPIOs.
- Remapping memory areas.
- Analog switch configuration management.
- Managing timer break or POC input lock.

1.6 System configuration registers

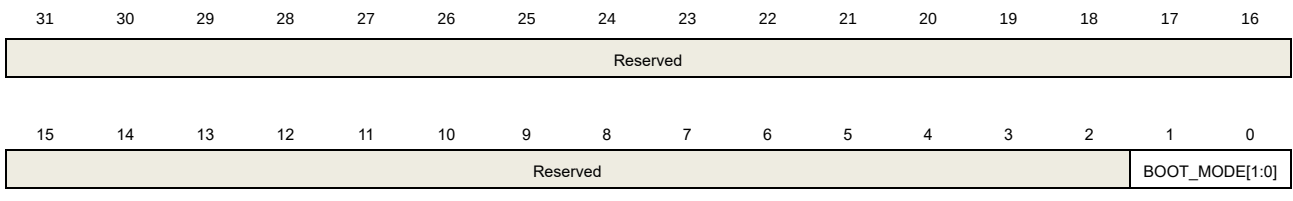
SYSCFG base address: 0x4001 0000

1.6.1 Configuration register 0 (SYSCFG_CFG0)

Address offset: 0x00

Reset value: 0x0000 000X (X indicates BOOT_MODE[1:0] may be any value according to the BOOT-PN6 pin)

This register has to be accessed by word (32-bit)



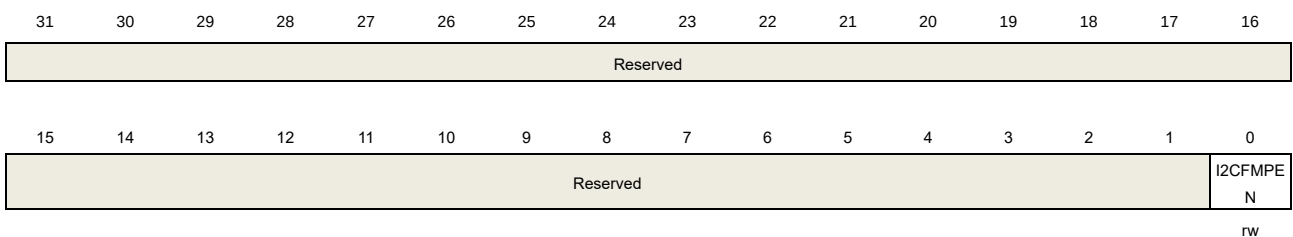
Bits	Fields	Descriptions
31:2	Reserved	Must be kept at reset value.
1:0	BOOT_MODE[1:0]	Boot mode (Refer to Table 1-4. Boot modes for details) Bit0 is mapping to the BOOT-PN6 pin; the value of Bit1 is 0. x0: Boot from the system memory 01: Boot from the main FLASH 11: Reserved

1.6.2 Configuration register 1 (SYSCFG_CFG1)

Address offset: 0x04

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit)



Bits	Fields	Descriptions
31:1	Reserved	Must be kept at reset value.
0	I2CFMPEN	Enable Fast mode+ on I2C 0: Disable Fm+ mode

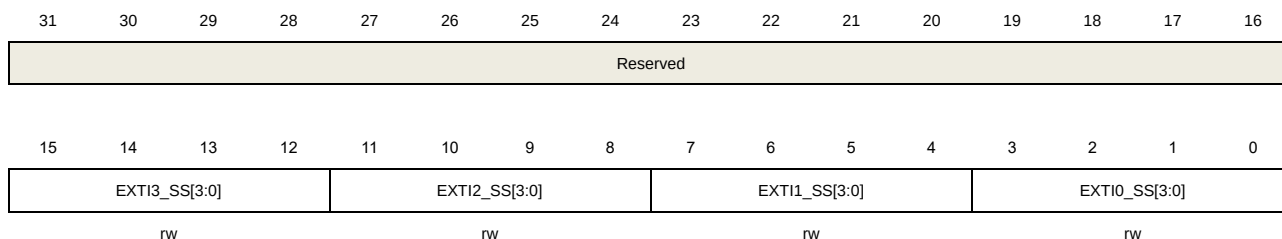
1: Enable Fm+ mode

1.6.3 EXTI sources selection register 0 (SYSCFG_EXTISS0)

Address offset: 0x08

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit)



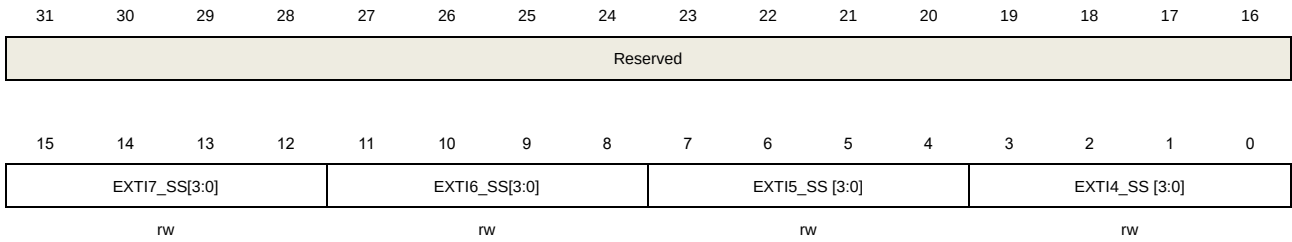
Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:12	EXTI3_SS[3:0]	EXTI3 sources selection 0000: PF12 pin Other configurations are reserved.
11:8	EXTI2_SS[3:0]	EXTI2 sources selection 0000: PA0 pin 0001: PC12 pin 0010: PF14 pin 0011: PG12 pin Other configurations are reserved.
7:4	EXTI1_SS[3:0]	EXTI1 sources selection 0000: PA9 pin 0001: PC11 pin 0010: PE13 pin 0011: PC9 pin Other configurations are reserved.
3:0	EXTI0_SS[3:0]	EXTI0 sources selection 0000: PA8 pin 0001: PC10 pin 0010: PN2 pin 0011: PC8 pin Other configurations are reserved.

1.6.4 EXTI sources selection register 1 (SYSCFG_EXTISS1)

Address offset: 0x0C

Reset value: 0x0000 0000

This register has to be accessed by word(32-bit)



Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:12	EXTI7_SS[3:0]	EXTI7 sources selection 0000: PB0 pin 0001: PC7 pin 0010: PN5 pin Other configurations are reserved.
11:8	EXTI6_SS[3:0]	EXTI6 sources selection 0000: PB1 pin 0001: PC6 pin 0010: PD2 pin 0011: PG13 pin Other configurations are reserved.
7:4	EXTI5_SS[3:0]	EXTI5 sources selection 0000: PD8 pin 0001: PG14 pin 0010: PN7 pin Other configurations are reserved.
3:0	EXTI4_SS[3:0]	EXTI4 sources selection 0000: PA1 pin 0001: PE14 pin 0010: PF9 pin 0011: PF13 pin Other configurations are reserved.

1.6.5 EXTI sources selection register 2 (SYSCFG_EXTISS2)

Address offset: 0x10

Reset value: 0x0000 0000

This register has to be accessed by word(32-bit)



Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTI11_SS[3:0]				EXTI10_SS[3:0]				EXTI9_SS[3:0]				EXTI8_SS[3:0]			
rw				rw				rw				rw			

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:12	EXTI11_SS[3:0]	EXTI11 sources selection 0000: PC1 pin 0001: PD11 pin 0010: PG11 pin Other configurations are reserved.
11:8	EXTI10_SS[3:0]	EXTI10 sources selection 0000: PB2 pin 0001: PC0 pin 0010: PD10 pin 0011: PE10 pin 0100: PF10 pin Other configurations are reserved.
7:4	EXTI9_SS[3:0]	EXTI9 sources selection 0000: PD5 pin 0001: PD9 pin 0010: PE9 pin 0011: PF11 pin Other configurations are reserved.
3:0	EXTI8_SS[3:0]	EXTI8 sources selection 0000: PD4 pin 0001: PE8 pin 0010: PF8 pin 0011: PG15 pin Other configurations are reserved.

1.6.6 EXTI sources selection register 3 (SYSCFG_EXTISS3)

Address offset: 0x14

Reset value: 0x0000 0000

This register has to be accessed by word(32-bit)

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTI15_SS[3:0]				EXTI14_SS[3:0]				EXTI13_SS[3:0]				EXTI12_SS[3:0]			
rw				rw				rw				rw			

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:12	EXTI15_SS[3:0]	EXTI15 sources selection 0000: PB15 pin 0001: PC5 pin Other configurations are reserved.
11:8	EXTI14_SS[3:0]	EXTI14 sources selection 0000: PB14 pin 0001: PC4 pin 0010: PD14 pin 0011: PE11 pin Other configurations are reserved.
7:4	EXTI13_SS[3:0]	EXTI13 sources selection 0000: PC3 pin 0001: PD13 pin Other configurations are reserved.
3:0	EXTI12_SS[3:0]	EXTI12 sources selection 0000: PC2 pin 0001: PD12 pin 0010: PE12 pin Other configurations are reserved.

1.6.7 Status register (SYSCFG_STAT)

Address offset: 0x1C

Reset value: 0x0000 0000

This register can be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		FWDGT_NMIIF	WWDGT_NMIIF	PIN_NMIIF	HXTAL_NMIIF	Reserved									
rc_w1		r	r	r	r										
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						ECCMEIF4	ECCSEIF3	ECCMEIF3	ECCSEIF2	ECCMEIF2	ECCSEIF1	ECCMEIF1	ECCSEIF0	ECCMEIF0	
						rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1

Bits	Fields	Descriptions

31:30	Reserved	Must be kept at reset value
29	FWDGT_NMIIF	FWDGT NMI interrupt flag. The software can clear it by writing 1. 0: FWDGT NMI interrupt is not detected. 1: FWDGT NMI interrupt is detected.
28	WWDGT_NMIIF	WWDGT NMI interrupt flag 0: WWDGT NMI interrupt is not detected. 1: WWDGT NMI interrupt is detected.
27	PIN_NMIIF	Interrupt flag from NMI pin 0: no input 1: interrupt input from NMI pin
26	HXTAL_NMIIF	HXTAL clock monitor NMI interrupt flag 0: no HXTAL clock monitor error. 1: HXTAL clock monitor is detected.
25:9	Reserved	Must be kept at reset value
8	ECCMEIF4	Flash multi-bit non-correction error flag. The software can clear it by writing 1. 0: no Flash multi-bit error is detected. 1: Flash multi-bit error is detected.
7	ECCSEIF3	Flash SRAM single-bit correction error flag. Flash SRAM includes I-cache, D-cache, and fast programming RAM. The software can clear it by writing 1. 0: no Flash SRAM single-bit error is detected. 1: Flash SRAM single-bit error is detected.
6	ECCMEIF3	Flash SRAM multi-bit non-correction error flag. Flash SRAM includes I-cache, D-cache, and fast programming RAM. The software can clear it by writing 1. 0: no Flash SRAM multi-bit error is detected. 1: Flash SRAM multi-bit error is detected.
5	ECCSEIF2	CAN filter single-bit correction error flag. The software can clear it by writing 1. 0: no CAN filter single-bit error is detected. 1: CAN filter single-bit error is detected.
4	ECCMEIF2	CAN filter multi-bit non-correction error flag The software can clear it by writing 1. 0: no CAN filter multi-bit error is detected. 1: CAN filter multi-bit error is detected.
3	ECCSEIF1	CAN receive FIFO single-bit correction error flag

		The software can clear it by writing 1. 0: no CAN receive FIFO single-bit error is detected. 1: CAN receive FIFO single-bit error is detected.
2	ECCMEIF1	CAN receive FIFO multi-bit non-correction error flag The software can clear it by writing 1. 0: no CAN receive FIFO SRAM multi-bit error is detected. 1: CAN receive FIFO SRAM multi-bit error is detected.
1	ECCSEIF0	SRAM single-bit correction error flag The software can clear it by writing 1. 0: no SRAM single-bit error is detected. 1: SRAM single-bit error is detected.
0	ECCMEIF0	SRAM multi-bit non-correction error flag The software can clear it by writing 1. 0: no SRAM multi-bit error is detected. 1: SRAM multi-bit error is detected.

1.6.8 Configuration register 2 (SYSCFG_CFG2)

Address offset: 0x20

Reset value: 0x0000 0008

This register can be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ECCEADDR[12:0]													Reserved		
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ECCSEBIT[5:0]						LVD1_NMI IE	LVD2_NMI IE	FWDGT_ NMISEL	WWDGT_ NMISEL	Reserved	PIN_NMI E	HXTAL_N MIIE	Reserved	ECCSEIE0	ECCMEIE 0
r						rw		rw	rw		rw	rw		rw	rw

Bits	Fields	Descriptions
31:19	ECCEADDR[12:0]	Indicates the last address of ECC error on SRAM occurred. Note: Physical SRAM word address = SRAM base address/4 + ECCEADDR
18:16	Reserved	Must be kept at reset value
15:10	ECCSEBIT[5:0]	Indicates the error bit which one bit has an ECC single-bit correction error 0: no error 1: bit 0 ... 32: bit 31
9	LVD1_NMIIE	Low Voltage Detectors1 NMI interrupt enable 0: disable

		1: enable
8	LVD2_NMIIE	Low Voltage Detectors2 NMI interrupt enable 0: disable 1: enable
7	FWDGT_NMISEL	FWDGT NMI interrupt type select 0: Maskable interrupt 1: Non-maskable interrupt
6	WWDGT_NMISEL	WWDGT NMI interrupt type select 0: Maskable interrupt 1: Non-maskable interrupt
5	Reserved	Must be kept at reset value
4	PIN_NMIIE	NMI pin interrupt enable. 0: disable 1: enable
3	HXTAL_NMIIE	HXTAL clock monitor NMI interrupt enable. 0: disable 1: enable
2	Reserved	Must be kept at reset value
1	ECCSEIE0	SRAM single-bit correction interrupt enable 0: SRAM single-bit correction interrupt is disabled. 1: SRAM single-bit correction interrupt is enabled.
0	ECCMEIE0	SRAM multi-bit non-correction interrupt enable 0: SRAM multi-bit interrupt is disabled. 1: SRAM multi-bit interrupt is enabled.

1.6.9 Configuration register 3 (SYSCFG_CFG3)

Address offset: 0x24

Reset value: 0x0000 0000

This register can be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												ECCSEIE	ECCMEIE	ECCSEIE1	ECCMEIE
												2	2	1	1
												rw	rw	rw	rw

Bits	Fields	Descriptions

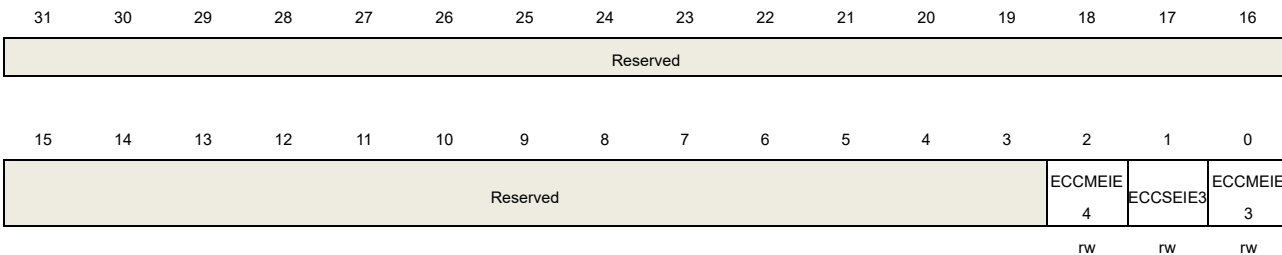
31:4	Reserved	Must be kept at reset value
3	ECCSEIE2	CAN filter single-bit correction interrupt enable 0: CAN filter single-bit correction interrupt is disabled. 1: CAN filter single-bit correction interrupt is enabled.
2	ECCMEIE2	CAN filter multi-bit non-correction interrupt enable 0: CAN filter multi-bit interrupt is disabled. 1: CAN filter multi-bit interrupt is enabled.
1	ECCSEIE1	CAN receive FIFO single-bit correction interrupt enable 0: CAN receive FIFO single-bit correction interrupt is disabled. 1: CAN receive FIFO single-bit correction interrupt is enabled.
0	ECCMEIE1	CAN receive FIFO multi-bit non-correction interrupt enable 0: CAN receive FIFO multi-bit interrupt is disabled. 1: CAN receive FIFO multi-bit interrupt is enabled.

1.6.10 Configuration register 4 (SYSCFG_CFG4)

Address offset: 0x28

Reset value: 0x0000 0000

This register can be accessed by word(32-bit).



Bits	Fields	Descriptions
31:3	Reserved	Must be kept at reset value
2	ECCMEIE4	FLASH multi-bit non-correction interrupt enable 0: Disable. 1: Enable.
1	ECCSEIE3	FLASH SRAM single-bit correction interrupt enable 0: Disable. 1: Enable.
0	ECCMEIE3	FLASH SRAM multi-bit non-correction interrupt enable 0: Disable. 1: Enable.

1.6.11 Timer input selection register 0 (SYSCFG_TIMERCISEL0)

Address offset: 0x34

Reset value: 0x0000 0000

This register can be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TIMER0_CI3_SEL[3:0]				TIMER0_CI2_SEL[3:0]				TIMER0_CI1_SEL[3:0]				TIMER0_CI0_SEL[3:0]			
rw				rw				rw				rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIMER7_CI3_SEL[3:0]				TIMER7_CI2_SEL[3:0]				TIMER7_CI1_SEL[3:0]				TIMER7_CI0_SEL[3:0]			
rw				rw				rw				rw			

Bits	Fields	Descriptions
31:28	TIMER0_CI3_SEL[3:0]	Selects TIMER0_CI3 input selection These bits select the TIMER input source. 0000: TIMER0_CH3 input Others: Reserved
27:24	TIMER0_CI2_SEL[3:0]	Selects TIMER0_CI2 input selection These bits select the TIMER input source. 0000: TIMER0_CH2 input Others: Reserved
23:20	TIMER0_CI1_SEL[3:0]	Selects TIMER0_CI1 input selection These bits select the TIMER input source. 0000: TIMER0_CH1 input Others: Reserved
19:16	TIMER0_CI0_SEL[3:0]	Selects TIMER0_CI0 input selection These bits select the TIMER input source. 0000: TIMER0_CH0 input 0001: CMP0 output Others: Reserved
15:12	TIMER7_CI3_SEL[3:0]	Selects TIMER7_CI3 input selection These bits select the TIMER input source. 0000: TIMER7_CH3 input Others: Reserved
11:8	TIMER7_CI2_SEL[3:0]	Selects TIMER7_CI2 input selection These bits select the TIMER input source. 0000: TIMER7_CH2 input Others: Reserved
7:4	TIMER7_CI1_SEL[3:0]	Selects TIMER7_CI1 input selection These bits select the TIMER input source.

0000: TIMER7_CH1 input

Others: Reserved

3:0 TIMER7_CIO_SEL[3:0] Selects TIMER7_CIO input selection
0] These bits select the TIMER input source.
0000: TIMER7_CH0 input
0001: CMP1 output
Others: Reserved

1.6.12 Timer input selection register 1 (SYSCFG_TIMERCISEL1)

Address offset: 0x38

Reset value: 0x0000 0000

This register can be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TIMER1_CI3_SEL[3:0]				TIMER1_CI2_SEL[3:0]				TIMER1_CI1_SEL[3:0]				TIMER1_CIO_SEL[3:0]			
rw				rw				rw				rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIMER2_CI3_SEL[3:0]				TIMER2_CI2_SEL[3:0]				TIMER2_CI1_SEL[3:0]				TIMER2_CIO_SEL[3:0]			
rw				rw				rw				rw			

Bits	Fields	Descriptions
31:28	TIMER1_CI3_SEL[3:0] 0]	TIMER1_CI3 input selection These bits select the TIMER input source. 0000: TIMER1_CH3 input 0001: CMP0 output 0010: CMP1 output 0011: CMP0 output or CMP1 output Others: Reserved
27:24	TIMER1_CI2_SEL[3:0] 0]	TIMER1_CI2 input selection These bits select the TIMER input source. 0000: TIMER1_CH2 input Others: Reserved
23:20	TIMER1_CI1_SEL[3:0] 0]	TIMER1_CI1 input selection These bits select the TIMER input source. 0000: TIMER1_CH1 input 1000: TIMER external trigger 0 1001: TIMER external trigger 1 1010: TIMER external trigger 2 1011: TIMER external trigger 3 Others: Reserved
19:16	TIMER1_CIO_SEL[3:0]	TIMER1_CIO input selection

	0]	These bits select the TIMER input source. 0000: TIMER1_CH0 input 1000: TIMER external trigger 0 1001: TIMER external trigger 1 1010: TIMER external trigger 2 1011: TIMER external trigger 3 Others: Reserved
15:12	TIMER2_CI3_SEL[3:	TIMER2_CI3 input selection
	0]	These bits select the TIMER input source. 0000: TIMER2_CH3 input 1110: IRC32K 1111: CK_OUT Others: Reserved
11:8	TIMER2_CI2_SEL[3:	TIMER2_CI2 input selection
	0]	These bits select the TIMER input source. 0000: TIMER2_CH2 input Others: Reserved
7:4	TIMER2_CI1_SEL[3:	TIMER2_CI1 input selection
	0]	These bits select the TIMER input source. 0000: TIMER2_CH1 input 1000: TIMER external trigger 0 1001: TIMER external trigger 1 1010: TIMER external trigger 2 1011: TIMER external trigger 3 Others: Reserved
3:0	TIMER2_CI0_SEL[3:	TIMER2_CI0 input selection
	0]	These bits select the TIMER input source. 0000: TIMER2_CH0 input 0001: CMP0 output 0010: CMP1 output 0011: CMP0 output or CMP1 output 1000: TIMER external trigger 0 1001: TIMER external trigger 1 1010: TIMER external trigger 2 1011: TIMER external trigger 3 Others: Reserved

1.6.13 Configuration register 5 (SYSCFG_CFG5)

Address offset: 0x54

Reset value: 0x0000 0000 (SYSTEM RESET)

This register can be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									FLASH_LO CK	FLASHSR AM_LOCK	CANFILTE R_LOCK	CANRX_L OCK	SRAM_L OCK	LVD_LOC K	LOCKUP _LOCK
									rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:7	Reserved	Must be kept at reset value.
6	FLASH_LOCK	Flash ECC multi-bit error lock. This bit is set by software and cleared only by a system reset. It can be used to enable and lock the Flash ECC check error connection to TIMER0/7 Break inputs. 0: Disconnect 1: Connect
5	FLASHSRAM_LOCK	Flash SRAM ECC multi-bit error lock. This bit is set by software and cleared only by a system reset. It can be used to enable and lock the Flash SRAM ECC check error connection to TIMER0/7 Break inputs. 0: Disconnect 1: Connect
4	CANFILTER_LOCK	CAN filter ECC multi-bit error lock This bit is set by software and cleared only by a system reset. It can be used to enable and lock the CAN filter ECC check error connection to TIMER0/7 Break inputs. 0: Disconnect 1: Connect
3	CANRX_LOCK	CAN receive FIFO ECC multi-bit error lock This bit is set by software and cleared only by a system reset. It can be used to enable and lock the CAN receive FIFO ECC check error connection to TIMER0/7 Break inputs. 0: Disconnect 1: Connect
2	SRAM_LOCK	SRAM ECC multi-bit error lock. This bit is set by software and cleared only by a system reset. It can be used to enable and lock the SRAM ECC check error connection to TIMER0/7 Break inputs. 0: Disconnect

1: Connect

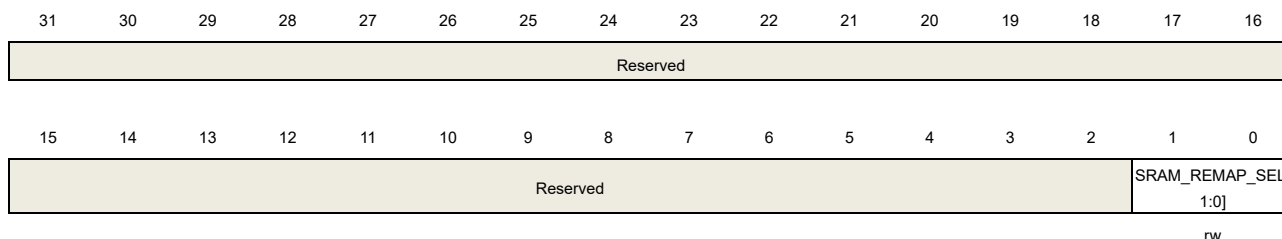
1	LVD_LOCK	Low voltage detector lock. This bit is set by software and cleared by a system reset. It can be used to enable and lock the LVD ECC error connection to TIMER0/7 Break inputs. 0: Disconnect 1: Connect
0	LOCKUP_LOCK	Cortex-M33 lockup (Hardfault) output lock. This bit is set by software and cleared by a system reset. It can be used to enable and lock the lockup output connection to TIMER0/7 Break and POC inputs. 0: Disconnect 1: Connect

1.6.14 SRAM configuration register (SYSCFG_SRAMCFG)

Address offset: 0x68

Reset value: 0x0000 0003

This register can be accessed by word(32-bit).



Bits	Fields	Descriptions
31:2	Reserved	Must be kept at reset value
1:0	SRAM_REMAP_SEL [1:0]	This bit is used to control the size of SRAM aliased in 0x1000 0000 (remap of SRAM in 0x2000 0000). The CPU can access and SRAM aliased in 0x1000 0000 via CBUS instead of SBUS. 00: 0KB 01: 8KB 10: 16KB 11: 32KB

1.6.15 Protection configuration register (SYSCFG_PRCFG)

Address offset: 0x6C

Reset value: 0x0000 0000 (POWER RESET)

This register has to be accessed by word(32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRKEY[7:0]								Reserved			PRCFG2	Reserved	PRCFG1	PRCFG0	
rw											rw		rw	rw	

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:8	PRKEY[7:0]	Protect area write key. Only software writes the right Key(0xA5) to these bits, protect configuration select PRCFG2 / PRCFG1 / PRCFG0 can be valid.
7:4	Reserved	Must be kept at reset value.
3	PRCFG2	Protect configuration select 2. Enable writing to the registers of PMU: PMU_LVD1CTL, PMU_LVD2CTL. 0: Write disabled 1: Write enabled
2	Reserved	Must be kept at reset value.
1	PRCFG1	Protect configuration select 1. Enable writing to the registers of RCU: RCU_APB1RST, RCU_APB2RST, RCU_AHBEN, RCU_APB1EN, RCU_APB2EN, RCU_RSTSCK, RCU_AHBRST. 0: Write disabled 1: Write enabled
0	PRCFG0	Protect configuration select 0. Enable writing to the registers of RCU: RCU_CTL, RCU_CFG0, RCU_INT, RCU_CFG1, RCU_CFG2. 0: Write disabled 1: Write enabled

1.6.16 TIMERx ADC start request configuration register (TADSRCFG)

Address offset: 0x70

Reset value: 0x0000 0000

This register has to be accessed by word(32-bit)

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

ADCSM2 EN	Reserved	ADCSM2SEL[3:0]	ADCSM1 EN	Reserved	ADCSM1SEL[3:0]
rw		rw	rw		rw

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15	ADCSM2EN	ADCSM2 pin output enable 0: ADCSM2 pin output disabled 1: ADCSM2 pin output enabled
14:12	Reserved	Must be kept at reset value.
11:8	ADCSM2SEL[3:0]	Select the ADC trigger source connected to ADCSM2 pin. These bits select the ADC trigger source. 0000: Source of monitor is TIMER0_TRGOF 0001: Source of monitor is TIMER0_TRGUF 0010: Source of monitor is TIMER0_TRGA 0011: Source of monitor is TIMER0_TRGB 0100: Source of monitor is TIMER0_TRGAB 0101: Source of monitor is TIMER7_TRGOF 0110: Source of monitor is TIMER7_TRGUF 0111: Source of monitor is TIMER7_TRGA 1000: Source of monitor is TIMER7_TRGB 1001: Source of monitor is TIMER7_TRGAB 1010~1111: Reserved
7	ADCSM1EN	ADCSM1 pin output enable 0: ADCSM1 pin output disabled 1: ADCSM1 pin output enabled
6:4	Reserved	must be kept at reset value
3:0	ADCSM1SEL[3:0]	Select the ADC trigger source connected to ADCSM1 pin. These bits select the ADC trigger source. 0000: Source of monitor is TIMER0_TRGOF 0001: Source of monitor is TIMER0_TRGUF 0010: Source of monitor is TIMER0_TRGA 0011: Source of monitor is TIMER0_TRGB 0100: Source of monitor is TIMER0_TRGAB 0101: Source of monitor is TIMER7_TRGOF 0110: Source of monitor is TIMER7_TRGUF 0111: Source of monitor is TIMER7_TRGA 1000: Source of monitor is TIMER7_TRGB 1001: Source of monitor is TIMER7_TRGAB 1010~1111: Reserved

1.6.17 TIMERx configuration register 0 (SYSCFG_TIMERxCFG0, x=0, 7)

Address offset: 0x100 for TIMER0

Address offset: 0x124 for TIMER7

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved	TSCFG5[4:0]					TSCFG4[4:0]					TSCFG3[4:0]				
	rw					rw					rw				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	TSCFG2[4:0]					TSCFG1[4:0]					TSCFG0[4:0]				
	rw					rw					rw				

Bits	Fields	Descriptions
31	Reserved	Must be kept at reset value.
30:26	TSCFG5[4:0]	Event mode configuration A rising edge of the trigger input enables the counter. 00000: Event mode disable 00001: Internal trigger input 0 (ITI0) 00010: Internal trigger input 1 (ITI1) 00011: Internal trigger input 2 (ITI2) 00100: Internal trigger input 3 (ITI3) 00101: CI0 edge flag (CI0F_ED) 00110: The filtered output of channel 0 input (CI0FE0) 00111: The filtered output of channel 1 input (CI1FE1) 01000: The filtered output of external trigger input (ETIFP) 01001: The filtered output of channel 2 input (CI2FE2) 01010: The filtered output of channel 3 input (CI3FE3) 01011: The filtered output of multi mode channel 0 input (MCI0FEM0) 01100: The filtered output of multi mode channel 1 input (MCI1FEM1) 01101: The filtered output of multi mode channel 2 input (MCI2FEM2) 01110: The filtered output of multi mode channel 3 input (MCI3FEM3) Others: Reserved
25:21	TSCFG4[4:0]	Pause mode configuration The trigger input enables the counter clock when it is high and disables the counter when it is low when these bits are not 0. 00000: Pause mode disable 00001: Internal trigger input 0 (ITI0) 00010: Internal trigger input 1 (ITI1) 00011: Internal trigger input 2 (ITI2) 00100: Internal trigger input 3 (ITI3) 00101: CI0 edge flag (CI0F_ED)

		00110: The filtered output of channel 0 input (CI0FE0) 00111: The filtered output of channel 1 input (CI1FE1) 01000: The filtered output of external trigger input (ETIFP) 01001: The filtered output of channel 2 input (CI2FE2) 01010: The filtered output of channel 3 input (CI3FE3) 01011: The filtered output of multi mode channel 0 input (MCI0FEM0) 01100: The filtered output of multi mode channel 1 input (MCI1FEM1) 01101: The filtered output of multi mode channel 2 input (MCI2FEM2) 01110: The filtered output of multi mode channel 3 input (MCI3FEM3) Others: Reserved
20:16	TSCFG3[4:0]	Restart mode configuration The counter is reinitialized and the shadow registers are updated on the rising edge of the selected trigger input when these bits are not 0. 00000: Restart mode disable 00001: Internal trigger input 0 (ITI0) 00010: Internal trigger input 1 (ITI1) 00011: Internal trigger input 2 (ITI2) 00100: Internal trigger input 3 (ITI3) 00101: CI0 edge flag (CI0F_ED) 00110: The filtered output of channel 0 input (CI0FE0) 00111: The filtered output of channel 1 input (CI1FE1) 01000: The filtered output of external trigger input (ETIFP) 01001: The filtered output of channel 2 input (CI2FE2) 01010: The filtered output of channel 3 input (CI3FE3) 01011: The filtered output of multi mode channel 0 input (MCI0FEM0) 01100: The filtered output of multi mode channel 1 input (MCI1FEM1) 01101: The filtered output of multi mode channel 2 input (MCI2FEM2) 01110: The filtered output of multi mode channel 3 input (MCI3FEM3) Others: Reserved
15	Reserved	Must be kept at reset value.
14:10	TSCFG2[4:0]	Quadrature decoder mode 2 configuration 00000: Quadrature decoder mode 2 disable Others: The counter counts on both CI0FE0 and CI1FE1 edges, while the direction depends on each other.
9:5	TSCFG1[4:0]	Quadrature decoder mode 1 configuration 00000: Quadrature decoder mode 1 disable Others: The counter counts on CI1FE1 edge, while the direction depends on CI0FE0 level.
4:0	TSCFG0[4:0]	Quadrature decoder mode 0 configuration 00000: Quadrature decoder mode 0 disable Others: The counter counts on CI0FE0 edge, while the direction depends on

CI1FE1 level.

1.6.18 TIMERx configuration register 1 (SYSCFG_TIMERxCFG1, x=0, 7)

Address offset: 0x104 for TIMER0

Address offset: 0x128 for TIMER7

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved	TSCFG11[4:0]					TSCFG10[4:0]					TSCFG9[4:0]				
rw					rw					rw					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						TSCFG7[4:0]					TSCFG6[4:0]				
						rw					rw				

Bits	Fields	Descriptions
31	Reserved	Must be kept at reset value.
30:26	TSCFG11[4:0]	Non-quadrature decoder mode 2 configuration 00000: disable Others: The counter will count on both rising and falling edges of CI0FE0 and CI1FE1 signals. And the counting direction determined by the CH0P and CH1P bits.
25:21	TSCFG10[4:0]	Non-quadrature decoder mode 1 configuration 00000: disable Others: The CI0FE0 is used as the count direction signal and the CI1FE1 signal is used as the count pulse. And the counter counts on the edge of CI1FE1, which is set by the CH1P.
20:16	TSCFG9[4:0]	Non-quadrature decoder mode 0 configuration 00000: Disable Others: The CI0FE0 is used as the count direction signal and the CI1FE1 signal is used as the count pulse. The counter will count on both rising and falling edges of the CI1FE1 signal.
15:10	Reserved	Must be kept at reset value.
9:5	TSCFG7[4:0]	Restart + event mode configuration The counter is reinitialized and started, the shadow registers are updated on the rising edge of the selected trigger input when these bits are not 0. 00000: Restart + event mode disable 00001: Internal trigger input 0 (ITI0) 00010: Internal trigger input 1 (ITI1) 00011: Internal trigger input 2 (ITI2) 00100: Internal trigger input 3 (ITI3)

		00101: CI0 edge flag (CI0F_ED)
		00110: The filtered output of channel 0 input (CI0FE0)
		00111: The filtered output of channel 1 input (CI1FE1)
		01000: The filtered output of external trigger input (ETIFP)
		01001: The filtered output of channel 2 input (CI2FE2)
		01010: The filtered output of channel 3 input (CI3FE3)
		01011: The filtered output of multi mode channel 0 input (MCI0FEM0)
		01100: The filtered output of multi mode channel 1 input (MCI1FEM1)
		01101: The filtered output of multi mode channel 2 input (MCI2FEM2)
		01110: The filtered output of multi mode channel 3 input (MCI3FEM3)
		Others: Reserved
4:0	TSCFG6[4:0]	External clock mode 0 configuration
		The counter counts on the rising edges of the selected trigger when these bits are not 0.
		00000: External clock mode 0 disable
		00001: Internal trigger input 0 (ITI0)
		00010: Internal trigger input 1 (ITI1)
		00011: Internal trigger input 2 (ITI2)
		00100: Internal trigger input 3 (ITI3)
		00101: CI0 edge flag (CI0F_ED)
		00110: The filtered output of channel 0 input (CI0FE0)
		00111: The filtered output of channel 1 input (CI1FE1)
		01000: The filtered output of external trigger input (ETIFP)
		01001: The filtered output of channel 2 input (CI2FE2)
		01010: The filtered output of channel 3 input (CI3FE3)
		01011: The filtered output of multi mode channel 0 input (MCI0FEM0)
		01100: The filtered output of multi mode channel 1 input (MCI1FEM1)
		01101: The filtered output of multi mode channel 2 input (MCI2FEM2)
		01110: The filtered output of multi mode channel 3 input (MCI3FEM3)
		Others: Reserved

1.6.19 TIMERx configuration register 2 (SYSCFG_TIMERxCFG2, x=0, 7)

Address offset: 0x108 for TIMER0

Address offset: 0x12C for TIMER7

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved											TSCFG15[4:0]				
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved											TSCFG12[4:0]				

Bits	Fields	Descriptions
31:21	Reserved	Must be kept at reset value.
20:16	TSCFG15[4:0]	Internal trigger input source configuration 00000: Internal trigger input 0 (ITI0) 00001: Internal trigger input 0 (ITI0) 00010: Internal trigger input 1 (ITI1) 00011: Internal trigger input 2 (ITI2) 00100: Internal trigger input 3 (ITI3) 00101: CI0 edge flag (CI0F_ED) 00110: The filtered output of channel 0 input (CI0FE0) 00111: The filtered output of channel 1 input (CI1FE1) 01000: The filtered output of external trigger input (ETIFP) 01001: The filtered output of channel 2 input (CI2FE2) 01010: The filtered output of channel 3 input (CI3FE3) 01011: The filtered output of multi mode channel 0 input (MCI0FEM0) 01100: The filtered output of multi mode channel 1 input (MCI1FEM1) 01101: The filtered output of multi mode channel 2 input (MCI2FEM2) 01110: The filtered output of multi mode channel 3 input (MCI3FEM3) Others: Reserved Note: When TSCFG15[4:0] is used, TSCFGy[4:0](y=0..9) should be zero, otherwise, the ITS trigger cooperate with TSCFGy[4:0], and input source depend on TSCFGy[4:0].
15:5	Reserved	Must be kept at reset value.
4:0	TSCFG12[4:0]	Non-quadrature decoder mode 3 configuration 00000: disable Others: the counter will count on rising or falling edge of CI0FE0 and CI1FE1 signals. When CHxP=0, the counter will counter on the high level or the falling edge or the ClxFEx signal; When CHxP=1, the counter will counter on the low level or the rising edge or the ClxFEx signal.

1.6.20 TIMERx configuration register 0 (SYSCFG_TIMERxCFG0, x=1, 2)

Address offset: 0x10C for TIMER1

Address offset: 0x118 for TIMER2

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved	TSCFG5[4:0]					TSCFG4[4:0]					TSCFG3[4:0]				
rw					rw					rw					

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	TSCFG2[4:0]					TSCFG1[4:0]					TSCFG0[4:0]				
	rw					rw					rw				

Bits	Fields	Descriptions
31	Reserved	Must be kept at reset value.
30:26	TSCFG5[4:0]	Event mode configuration A rising edge of the trigger input enables the counter. 00000: Event mode disable 00001: Internal trigger input 0 (ITI0) 00010: Internal trigger input 1 (ITI1) 00011: Internal trigger input 2 (ITI2) 00100: Internal trigger input 3 (ITI3) 00101: CI0 edge flag (CI0F_ED) 00110: The filtered output of channel 0 input (CI0FE0) 00111: The filtered output of channel 1 input (CI1FE1) 01000: The filtered output of external trigger input (ETIFP) Others: Reserved
25:21	TSCFG4[4:0]	Pause mode configuration The trigger input enables the counter clock when it is high and disables the counter when it is low when these bits are not 0. 00000: Pause mode disable 00001: Internal trigger input 0 (ITI0) 00010: Internal trigger input 1 (ITI1) 00011: Internal trigger input 2 (ITI2) 00100: Internal trigger input 3 (ITI3) 00101: CI0 edge flag (CI0F_ED) 00110: The filtered output of channel 0 input (CI0FE0) 00111: The filtered output of channel 1 input (CI1FE1) 01000: The filtered output of external trigger input (ETIFP) Others: Reserved
20:16	TSCFG3[4:0]	Restart mode configuration The counter is reinitialized and the shadow registers are updated on the rising edge of the selected trigger input when these bits are not 0. 00000: Restart mode disable 00001: Internal trigger input 0 (ITI0) 00010: Internal trigger input 1 (ITI1) 00011: Internal trigger input 2 (ITI2) 00100: Internal trigger input 3 (ITI3) 00101: CI0 edge flag (CI0F_ED) 00110: The filtered output of channel 0 input (CI0FE0) 00111: The filtered output of channel 1 input (CI1FE1)

		01000: The filtered output of external trigger input (ETIFP) Others: Reserved
15	Reserved	Must be kept at reset value.
14:10	TSCFG2[4:0]	Quadrature decoder mode 2 configuration 00000: Quadrature decoder mode 2 disable Others: The counter counts on both CI0FE0 and CI1FE1 edges, while the direction depends on each other
9:5	TSCFG1[4:0]	Quadrature decoder mode 1 configuration 00000: Quadrature decoder mode 1 disable Others: The counter counts on CI1FE1 edge, while the direction depends on CI0FE0 level
4:0	TSCFG0[4:0]	Quadrature decoder mode 0 configuration 00000: Quadrature decoder mode 0 disable Others: The counter counts on CI0FE0 edge, while the direction depends on CI1FE1 level.

1.6.21 TIMERx configuration register 1 (SYSCFG_TIMERxCFG1, x=1, 2)

Address offset: 0x110 for TIMER1

Address offset: 0x11C for TIMER2

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved	TSCFG11[4:0]					TSCFG10[4:0]					TSCFG9[4:0]				
rw					rw					rw					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						TSCFG7[4:0]					TSCFG6[4:0]				
						rw					rw				

Bits	Fields	Descriptions
31	Reserved	Must be kept at reset value.
30:26	TSCFG11[4:0]	Non-quadrature decoder mode 2 configuration 00000: disable Others: The counter will count on both rising and falling edges of CI0FE0 and CI1FE1 signals. And the counting direction determined by the CH0P and CH1P bits.
25:21	TSCFG10[4:0]	Non-quadrature decoder mode 1 configuration 00000: disable Others: The CI0FE0 is used as the count direction signal and the CI1FE1 signal is used as the count pulse. And the counter counts on the edge of CI1FE1, which is

		set by the CH1P.
20:16	TSCFG9[4:0]	Non-quadrature decoder mode 0 configuration 00000: Disable Others: The CI0FE0 is used as the count direction signal and the CI1FE1 signal is used as the count pulse. The counter will count on both rising and falling edges of the CI1FE1 signal.
15:10	Reserved	Must be kept at reset value.
9:5	TSCFG7[4:0]	Restart + event mode configuration The counter is reinitialized and started, the shadow registers are updated on the rising edge of the selected trigger input when these bits are not 0. 00000: Restart + event mode disable 00001: Internal trigger input 0 (ITI0) 00010: Internal trigger input 1 (ITI1) 00011: Internal trigger input 2 (ITI2) 00100: Internal trigger input 3 (ITI3) 00101: CI0 edge flag (CI0F_ED) 00110: The filtered output of channel 0 input (CI0FE0) 00111: The filtered output of channel 1 input (CI1FE1) 01000: The filtered output of external trigger input (ETIFP) Others: Reserved
4:0	TSCFG6[4:0]	External clock mode 0 configuration The counter counts on the rising edges of the selected trigger when these bits are not 0. 00000: External clock mode 0 disable 00001: Internal trigger input 0 (ITI0) 00010: Internal trigger input 1 (ITI1) 00011: Internal trigger input 2 (ITI2) 00100: Internal trigger input 3 (ITI3) 00101: CI0 edge flag (CI0F_ED) 00110: The filtered output of channel 0 input (CI0FE0) 00111: The filtered output of channel 1 input (CI1FE1) 01000: The filtered output of external trigger input (ETIFP) Others: Reserved

1.6.22 TIMERx configuration register 2 (SYSCFG_TIMERxCFG2, x=1, 2)

Address offset: 0x114 for TIMER1

Address offset: 0x120 for TIMER2

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

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

Bits	Fields	Descriptions
31:21	Reserved	Must be kept at reset value.
20:16	TSCFG15[4:0]	Internal trigger input source configuration 00000: Reserved 00001: Internal trigger input 0 (ITI0) 00010: Internal trigger input 1 (ITI1) 00011: Internal trigger input 2 (ITI2) 00100: Internal trigger input 3 (ITI3) 00101: CI0 edge flag (CI0F_ED) 00110: The filtered output of channel 0 input (CI0FE0) 00111: The filtered output of channel 1 input (CI1FE1) 01000: The filtered output of external trigger input (ETIFP) Others: Reserved
15:5	Reserved	Must be kept at reset value.
4:0	TSCFG12[4:0]	Non-quadrature decoder mode 3 configuration 00000: disable Others: the counter will count on rising or falling edge of CI0FE0 and CI1FE1 signals. When CHxP=0, the counter will counter on the high level or the falling edge or the ClxFEx signal; When CHxP=1, the counter will counter on the low level or the rising edge or the ClxFEx signal.

Bits	Fields	Descriptions

31:1 Reserved Must be kept at reset value.

0 CK_CMP_SEL CK_CMP select
0: CK_CMP=fPCLK/2
1: CK_CMP=fPCLK/4

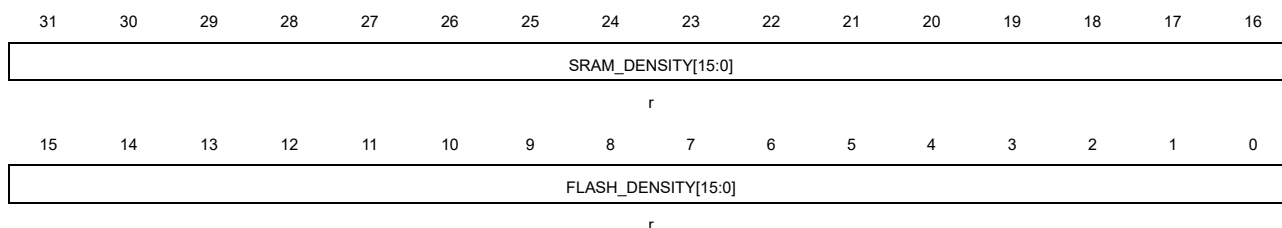
1.7 Device electronic signature

The device electronic signature contains memory size information and the 96-bit unique device ID. It is stored in the information block of the Flash memory. The 96-bit unique device ID is unique for any device. It can be used as serial numbers, or part of security keys, etc.

1.7.1 Memory density information

Base address: 0x1FFF F7E0

The value is factory programmed and can never be altered by user.

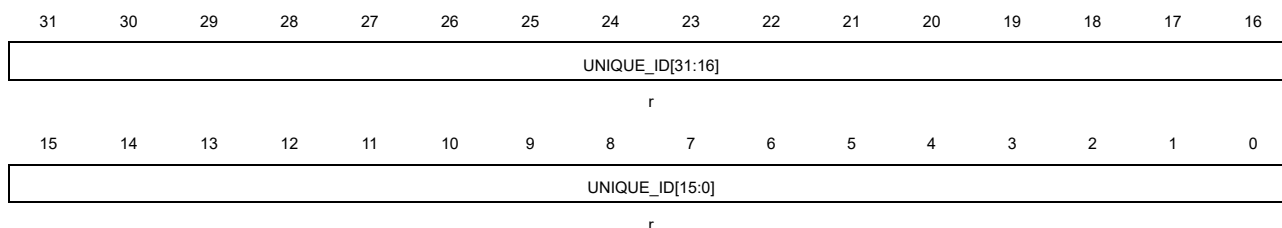


Bits	Fields	Descriptions
31:16	SRAM_DENSITY [15:0]	SRAM density The value indicates the on-chip SRAM density of the device in Kbytes. Example: 0x0020 indicates 32 Kbytes.
15:0	FLASH_DENSITY [15:0]	Flash memory density The value indicates the Flash memory density of the device in Kbytes. Example: 0x0100 indicates 256 Kbytes.

1.7.2 Unique device ID (96 bits)

Base address: 0x1FFF F7E8

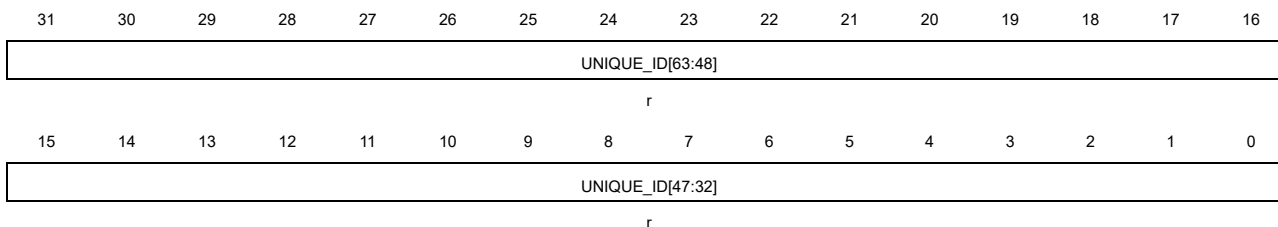
The value is factory programmed and can never be altered by user.



Bits	Fields	Descriptions
31:0	UNIQUE_ID[31:0]	Unique device ID

Base address: 0x1FFF F7EC

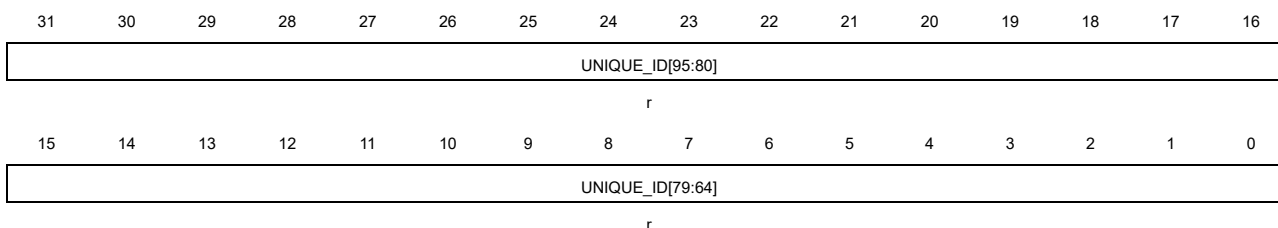
The value is factory programmed and can never be altered by user.



Bits	Fields	Descriptions
31:0	UNIQUE_ID[63:32]	Unique device ID

Base address: 0x1FFF F7F0

The value is factory programmed and can never be altered by user.



Bits	Fields	Descriptions
31:0	UNIQUE_ID[95:64]	Unique device ID

2 Flash memory controller (FMC)

2.1 Overview

The flash memory controller, FMC, provides all the necessary functions for the on-chip flash memory. A little waiting time is needed while CPU executes instructions stored from the main flash and data flash. It also provides page erase, mass erase, and program operations for flash memory.

2.2 Characteristics

- Up to 256KB of on-chip main flash memory for instruction and data. Up to 512B OTP. Up to 64KB data flash. 6KB information block.
 - Main flash: up to 256KB
 - Data flash: 64KB
 - OTP: 512B, which can be used for user data storage
 - Information block: 6KB
 - Option bytes: 32B, for user application requirements.
- Four words programming, page erase and mass erase operation.
- The flash page size is 1KB.
- ECC with single bit error corrected and double bit errors detected.
- Support fast program.
- Support blank check.
- 0~6 waiting time within 256KB bytes main flash and 64KB data flash when CPU executes instructions.
- Flash memory supports 128 wide data reading.
- Pre-fetch buffer to speed read operations.
- Support 8K bytes I-cache and 256 bytes D-cache.
- Option bytes are uploaded to the option byte control registers when the system is reset.
- Flash security protection to prevent illegal code / data access.
- Page erase / program protection to prevent unexpected operation for main flash.

2.3 Function overview

2.3.1 Flash memory architecture

The page size is 1KB. Each page can be erased individually.

The [Table 2-1. GD32M531 base address and size for flash memory](#) shows the details of flash organization.

Table 2-1. GD32M531 base address and size for flash memory

Block	Name	Address range	Size(bytes)
Main flash block	Page 0	0x0800 0000 - 0x0800 03FF	1KB
	Page 1	0x0800 0400 - 0x0800 07FF	1KB
	Page 2	0x0800 0800 - 0x0800 0BFF	1KB
	Page 255	0x0803 FC00 - 0x0803 FFFF	1KB
Data flash block	Page 0	0x0880 0000 - 0x0880 03FF	1KB
	Page 1	0x0880 0400 - 0x0880 07FF	1KB
	Page 2	0x0880 0800 - 0x0880 0BFF	1KB
	Page 63	0x0880 FC00 - 0x0880 FFFF	1KB
Information block	Boot loader area	0x1FFF E000- 0x1FFF F7FF	6KB
Option bytes block	Option bytes	0x1FFF F800 - 0x1FFF F81F	32B
One-time program block	OTP bytes	0x1FFF 7000~0x1FFF 71FF	512B

Note: The information block stores the boot loader. This block cannot be programmed or erased by user.

2.3.2 Error checking and correcting (ECC)

The ECC mechanism supports:

- One bit error detection and correction
- Two bits errors detection

When one error is detected and corrected:

- When ECC error occurs, the ECCCOR bit in FMC_ECCCS register will be set. If the ECCCORIE bit in FMC_ECCCS register is set, an interrupt is generated. The OB_ECC / OTP_ECC / MF_ECC / DF_ECC / SYS_ECC / SYSECCBK notice the space where error occurred. The ECCADDR bits notice the offset address in each space. The ECCCOR bit in FMC_ECCCS register can be cleared by writing 1 to ECCCOR bit.

When two errors are detected:

- When ECC error occurs, the ECCDET bit in FMC_ECCCS register will be set. If the ECCDETIE bit in FMC_ECCCS register is set, an interrupt is generated. The OB_ECC / OTP_ECC / MF_ECC / DF_ECC / SYS_ECC / SYSECCBK notice the space where error occurred. The ECCADDR bits notice the offset address in each space. And the data return all F. The ECCDET bit in FMC_ECCCS register can be cleared by writing 1 to ECCDET bit.

Note:

1. Data in main Flash memory are 137-bits words: 9 bits are added per four words (128 bits), but the added 9 bits are calculated by hardware and cannot be accessed by user. Data in data Flash memory are 72-bits words: 8 bits are added per double words (64 bits), but the added 8 bits are calculated by hardware and cannot be accessed by user.
2. For a virgin main flash data 0x1FF FFFF FFFF FFFF FFFF FFFF FFFF FFFF, ECC is not supported. For a virgin data flash data 0xFF FFFF FFFF FFFF FFFF, ECC is not supported.
3. The OB_ECC / OTP_ECC / SYS_ECC / MF_ECC / ECCADDR / SYSECCBK notice the first error occurred in single bit error and double bit errors. If ECCOR or ECCDET is set, the value will not be changed even if a new error occurred.
4. When an ECC error is reported, a new read at the error address may not generate an ECC error if the data is still present in the current buffer / prefetch buffer, even if ECCOR and ECCDET bits are cleared.
5. When the original data programmed is all F, the ECC calculation will bypass this operation. Erasing is not required, and programming other values again can be written normally.

2.3.3 Read operations

The flash can be addressed directly as a common memory space. Any instruction fetch and the data access from the flash are through the AHB BUS from the CPU.

Wait state added:

The WSCNT bits in the FMC_WS register needs to be configured correctly depend on the AHB clock frequency when reading the flash memory. The relation between WSCNT and AHB clock frequency is show as the [Table 2-2. The relation between WSCNT and AHB clock frequency](#).

Table 2-2. The relation between WSCNT and AHB clock frequency

AHB clock frequency	WSCNT configured
<= 25MHz	0 (0 wait state added)
<= 50MHz	1 (1 wait state added)
<= 75MHz	2 (2 wait state added)
<= 100MHz	3 (3 wait state added)
<= 125MHz	4 (4 wait state added)
<= 150MHz	5 (5 wait state added)
<= 180MHz	6 (6 wait state added)

If system reset occurs, the AHB clock frequency is 32MHz and the WSCNT is 1.

Note:

- If want to increase the AHB clock frequency. First, refer to [Table 2-2. The relation between WSCNT and AHB clock frequency](#) between WSCNT bit and AHB clock

frequency, configure the WSCNT bits according to the target AHB clock frequency. Then, increase the AHB clock frequency to the target frequency. It is forbidden to increase the AHB clock frequency before configure the WSCNT.

- If want to decrease the AHB clock frequency. First, decrease the AHB clock frequency to target frequency. Then refer to [Table 2-2. The relation between WSCNT and AHB clock frequency](#) between WSCNT bit and AHB clock frequency, configure the WSCNT bits according the target AHB clock frequency. It is forbidden to configure the WSCNT bits before decrease the AHB clock frequency.

Because the wait state is added, the read efficiency is very low (such as, add 3 wait state when 120MHz). In order to speed up the read access, there are some functions performed.

Current buffer:

The current buffer is always enabled. Each time read from flash memory, data get and store in current buffer.(64-bit current buffer for data and 128-bit current buffer for instruction.) The CPU only need 32-bit or 16-bit in each read operation. So in the case of sequential code, the next data can get from current buffer without repeat fetch from flash memory.

Pre-fetch buffer:

The pre-fetch buffer is enabled by set the PFEN bit in the FMC_WS register. The pre-fetch buffer is only performed only when fetching instructions. In the case of sequential code, when CPU execute the current buffer instruction(128-bit), it takes at least 4 clocks for 32-bit operation and at least 8 clocks for 16-bit operation. In this case, pre-fetch the data of next double-word address from flash memory and store to pre-fetch buffer. So when the CPU finish the current buffer and need execute the next data, the pre-fetch buffer hit.

I-cache:

I-cache is enabled by set the ICEN bit in the FMC_WS register. The I-cache is only used when fetching instructions. The I-cache have 8K bytes which organized as 256 cache lines, each cache lines is 2 * 128bits.

If the instructions data is in I-cache (I-cache hit), the CPU read data from I-cache without any wait state. If the instructions data is not in I-cache (I-cache miss) and not in current buffer / pre-fetch buffer, the cache line fetch from flash memory and copied to I-cache. If all cache line filled, LRU (least recently used) policy used to replace the cache line.

D-cache:

D-cache is enabled by set the DCEN bit in the FMC_WS register. The D-cache is only used when fetching data by CPU (not by DMA). And the option byte is not cacheable. The D- cache have 256 bytes which organized as 8 cache lines, each cache lines is 4 X 64bits.

If the data is in D-cache (D-cache hit), the CPU read data from D-cache without any wait state. If the data is not in D-cache (D-cache miss) and not in current buffer, the cache line fetch from

flash memory and copied to D-cache. If all cache line filled, LRU (least recently used) policy used to replace the cache line.

2.3.4 Unlock the FMC_CTL register

After reset, the FMC_CTL register is not accessible in write mode, and the LK bit in the FMC_CTL register is reset to 1. An unlocking sequence consists of two write operations to the FMC_KEY register to open the access to the FMC_CTL register. The two write operations are writing 0x45670123 and 0xCDEF89AB to the FMC_KEY register. After the two write operations, the LK bit in the FMC_CTL register is reset to 0 by hardware. The software can lock the FMC_CTL again by setting the LK bit in the FMC_CTL register to 1. Any wrong operations to the FMC_KEY, will set the LK bit to 1, and lock the FMC_CTL register, and lead to a bus error.

The OBPB bit and OBER bit in the FMC_CTL are still protected even the FMC_CTL is unlocked. The unlocking sequence consists of two write operations, which are writing 0x45670123 and 0xCDEF89AB to the FMC_OBKEY register. Then the hardware sets the OBWEN bit in the FMC_CTL register to 1. The software can reset OBWEN bit to 0 to protect the OBPB bit and OBER bit in the FMC_CTL register again.

2.3.5 Page erase

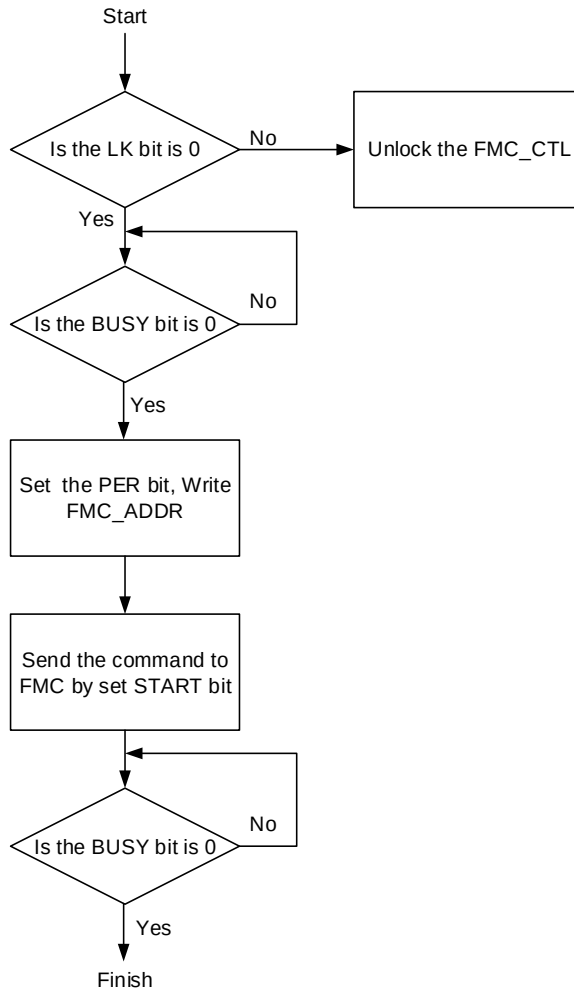
The FMC provides a page erase function which is used to initialize the contents of a main flash memory (BKSEL = 0) and data flash (BKSEL = 1) page to a high state. Each page can be erased independently without affecting the contents of other pages. The following steps show the access sequence of the registers for a page erase operation.

- Unlock the FMC_CTL register if necessary.
- Check the BUSY bit in the FMC_STAT register to confirm that no flash memory operation is in progress (BUSY equals to 0). Otherwise, wait until the operation has finished.
- Set the PER bit in the FMC_CTL register.
- Write the page absolute address (0x08XX XXXX) into the FMC_ADDR registers.
- Send the page erase command to the FMC by setting the START bit in the FMC_CTL register.
- Wait until all the operations have finished by checking the value of the BUSY bit in the FMC_STAT register.
- Read and verify the page if required.

When the operation is executed successfully, the ENDF bit in the FMC_STAT register is set, and an interrupt will be triggered by FMC if the ENDIE bit in the FMC_CTL register is set. Note that a correct target page address must be confirmed. Otherwise, the software may run out of control if the target erase page is being used to fetch codes or access data. The FMC will not provide any notification when that happens. Additionally, the page erase operation will be ignored on erase / program protected pages. In this condition, a flash operation error interrupt will be triggered by the FMC if the ERRIE bit in the FMC_CTL register is set. The

software can check the WPERR bit in the FMC_STAT register to detect this condition in the interrupt handler. The [Figure 2-1. Process of page erase operation](#) shows the page erase operation flow.

Figure 2-1. Process of page erase operation



2.3.6 Mass erase

The FMC provides a complete erase function which is used to initialize the main flash (BKSEL = 0) and data flash (BKSEL = 1) block contents. This erase can affect entire flash block by setting the MER bit to 1 in the FMC_CTL register. The following steps show the mass erase register access sequence.

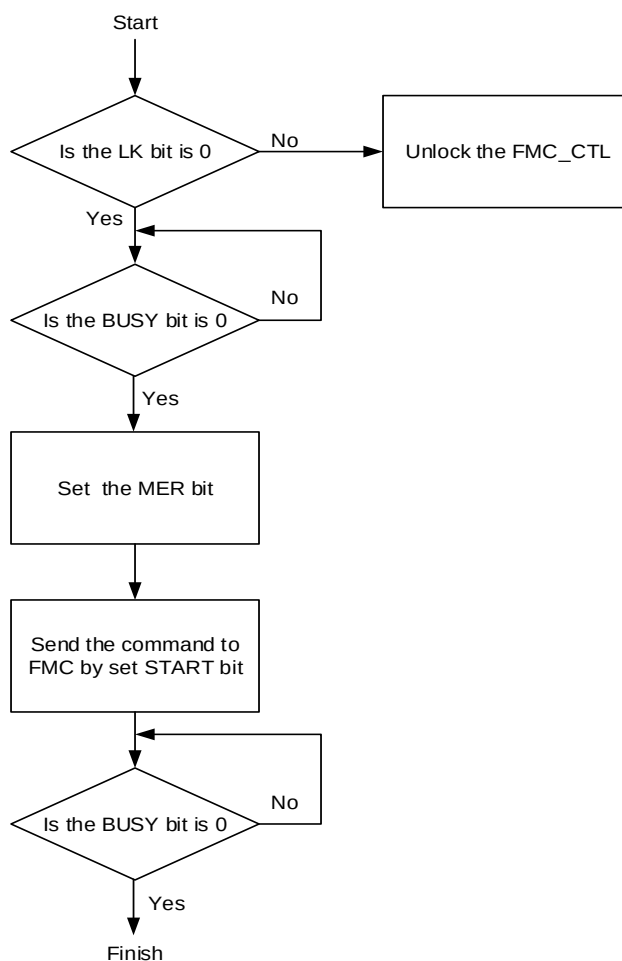
- Unlock the FMC_CTL register if necessary.
- Check the BUSY bit in the FMC_STAT register to confirm that no flash memory operation is in progress (BUSY equals to 0). Otherwise, wait until the operation has finished.
- Set the MER bit in the FMC_CTL register if erase entire flash.
- Send the mass erase command to the FMC by setting the START bit in the FMC_CTL register.

- Wait until all the operations have been finished by checking the value of the BUSY bit in the FMC_STAT register.
- Read and verify the flash memory if required.

When the operation is executed successfully, the ENDF bit in the FMC_STAT register is set, and an interrupt will be triggered by FMC if the ENDIE bit in the FMC_CTL register is set. Since all flash data will be modified to a value of 0xFFFF_FFFF, the mass erase operation can be implemented using a program that runs in SRAM or using the debugging tool that accesses the FMC registers directly. Additionally, the mass erase operation will be ignored if any page is erase / program protected. In this condition, a flash operation error interrupt will be triggered by the FMC if the ERRIE bit in the FMC_CTL register is set. The software can check the WPERR bit in the FMC_STAT register to detect this condition in the interrupt handler.

The [Figure 2-2. Process of mass erase operation](#) indicates the mass erase operation flow.

Figure 2-2. Process of mass erase operation



2.3.7 Flash programming

The FMC provides a 128-bit four words programming function which is used to modify the the main flash (BKSEL = 0) and data flash (BKSEL = 1) memory contents.

The following steps show the register access sequence of the programming operation.

- Unlock the FMC_CTL register if necessary.
- Check the BUSY bit in the FMC_STAT register to confirm that no flash memory operation is in progress (BUSY equals to 0). Otherwise, wait until the operation has finished.
- Set the PG bit in the FMC_CTL register.
- Write the 128-bit data to be programed with desired absolute address (0x08XX XXXX).
- Wait until all the operations have been finished by checking the value of the BUSY bit in the FMC_STAT register.
- Read and verify the flash memory if required.

When the operation is executed successfully, the ENDF bit in the FMC_STAT register is set, and an interrupt will be triggered by FMC if the ENDIE bit in the FMC_CTL register is set. Note that there are some program error need caution:

The programming operation checks the address if it has been erased or not. If the address has not been erased, the PGERR bit in the FMC_STAT register will be set. Each four-word can be programmed only one time after erase and before next erase. Note that the PG bit must be set before the programming operation.

Additionally, the program operation will be ignored on erase / program protected pages and the WPERR bit in the FMC_STAT will be set.

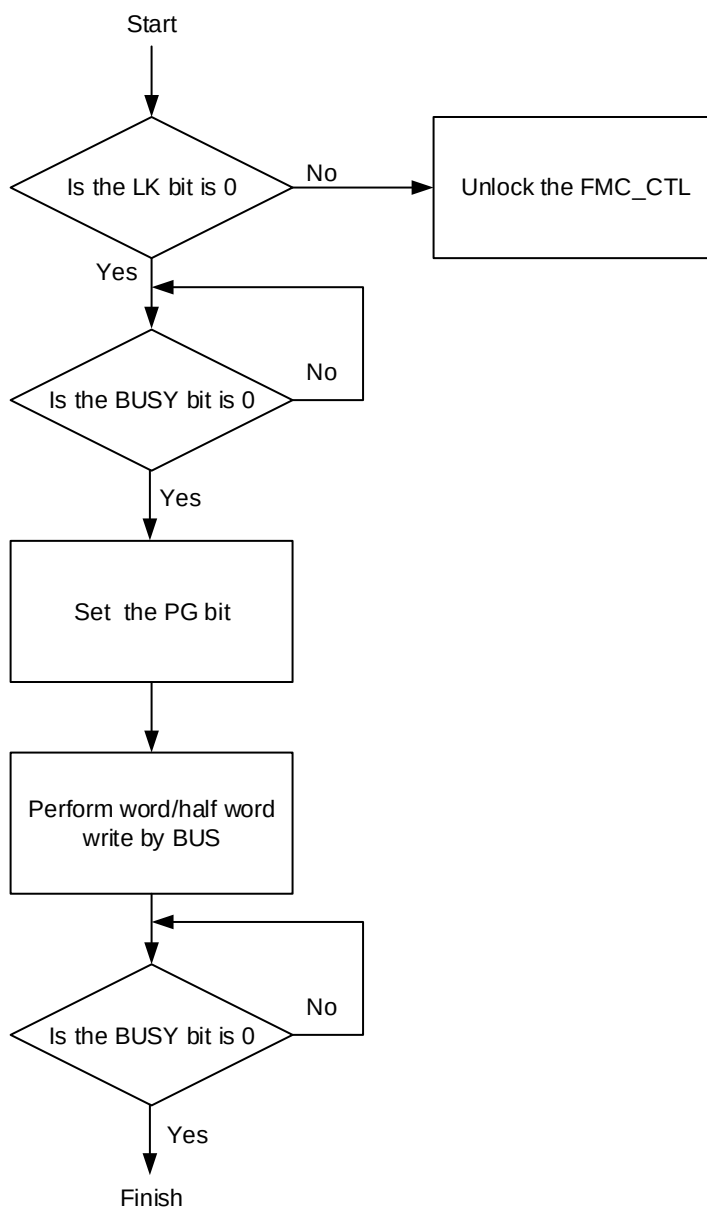
In the following cases, the PGAERR bit in the FMC_STAT register will be set.

- The BUS program use byte write (not 32-bit or 16-bit write)
- The BUS program size is not equal previous size. It not allows mix 32-bit with 16-bit write.
- The BUS write address is not alignment.

Note: If the program is not write total 128bits, the data is not program to the flash memory without any notice.

In these conditions, a flash operation error interrupt will be triggered by the FMC if the ERRIE bit in the FMC_CTL register is set. The software can check the PGERR bit, PGAERR bit or WPERR bit in the FMC_STAT register to detect which condition occurred in the interrupt handler. The [Figure 2-3. Process of program operation](#) shows the word programming operation flow.

Figure 2-3. Process of program operation



Note: Reading the flash should be avoided when a program / erase operation is ongoing in the same bank. And flash memory accesses will fail if the CPU enters the power saving modes.

2.3.8 Flash fast programming

The FMC provides a fast programming function which is used to modify the main flash (BKSEL = 0) and data flash (BKSEL = 1) memory contents. A row (16 four-word for main flash or 4 four-word for data flash) could be programmed to main flash and data flash memory in this mode to reduce the page programming time by eliminating the need for verifying the flash locations before they are programmed and to avoid rising and falling time of high voltage for each word.

The following steps show the register access sequence of the programming operation.

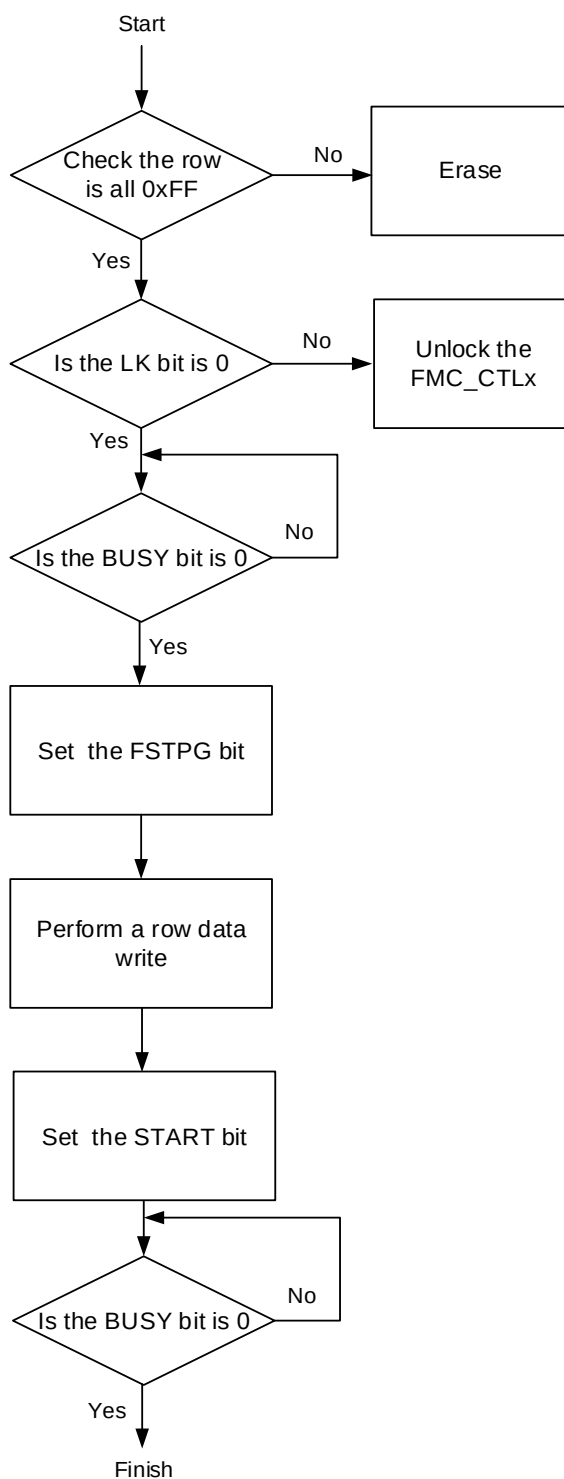
- Check the row in flash to confirm all data in flash is all 0xFF.
- Unlock the FMC_CTL register if necessary.
- Check the BUSY bit in FMC_STAT register to confirm that no flash memory operation is in progress (BUSY equals to 0). Otherwise, wait until the operation has finished.
- Set the FSTPG bit in FMC_CTL register.
- Write the one row data to be programmed by BUS with desired absolute address (0x08XX XXXX) (or write to FMC_DATA). The row data should be written sequentially and continuously. Or else the PGERR bit in FMC_CTL register will be set after setting the START bit in the FMC_CTL register.
- Set START bit in the FMC_CTL register to launch fast program operation to flash.
- Wait until all the operations have been finished by checking the value of the BUSY bit in FMC_STAT register.
- Read and verify the Flash memory if required using a BUS access. It is recommended to flush the cache if cache is enabled.

When the operation is executed successfully, the ENDF in FMC_STAT register is set, and an interrupt will be generated if the ENDIE bit in the FMC_CTL register is set. In sequential fast programming, if the row is confirmed erased, repeat 6~8 to fast program next row.

The program operation will be ignored on erase / program protected pages and WPERR bit in FMC_STAT will be set.

In these conditions, a flash operation error interrupt will be generated if the ERRIE bit in the FMC_CTL register is set. The software can check the PGSERR / PGAERR / WPERR / PGERR bit in the FMC_STAT register to detect which condition occurred in the interrupt handler. [Figure 2-4. Process of fast programming operation](#) shows the fast programming operation flow.

Figure 2-4. Process of fast programming operation



Note:

- The one row data must be written successively.
- The one row data must be aligned.
- Because the fast program does not check 0xFF in flash macro by hardware, the software must check 0xFF first and don't program one row twice or more between 2 erase

operations. If program one row twice or more between 2 erase operations, unpredictable result may occur.

- Between setting FSTPG and START, read operation is allowed, but the read address should not be the address of the row where to be programmed, otherwise the data read out may be old data.
- The cache must be flushed before fast programming.

2.3.9 Data flash operation

Read

The read of data flash is same as main flash. Refer to [Read operations](#).

Program

The program is same as main flash program. Refer to [Flash programming](#) and [Flash fast programming](#).

Page erase

The page erase is same as main flash page erase. Refer to [Page erase](#).

Mass erase

The mass erase is same as main flash mass erase. Refer to [Mass erase](#).

Note:

1. By configuring the BKSEL bit of the FMC_CTL register, select whether main flash or data flash to perform page erase operation, mass erase operation and fast programming operation.
2. Write data flash shall not block running program in main flash. Simultaneous program main flash and data flash are not supported.

2.3.10 Check blank command

The check blank command is used to check if the flash area which is specified by FMC_ADDR and CBCMDLEN bits in FMC_CTL register are all 0xFF or not. Configure the check blank command by setting the CBCMD bit in FMC_CTL register and send the check blank command to the FMC by setting the START bit in FMC_CTL register, the BUSY bit will be set, and the hardware will check if the flash area are all 0xFF or not. If the flash area are all 0xFF, the ENDF will be set, and the BUSY bit will be cleared, or else the CBCMDERR will be set. The check blank command only support for main flash / data flash.

Note:

1. The flash area to be checked must be in one page and should not exceed 1KB boundary.

2. The start address must be four-word aligned.

2.3.11 OTP programming

The OTP programming method is same as the main flash programming. The BKSEL bit need to be set to 1. The OTP block can only be programed once and cannot be erased.

Note: It must ensure the OTP programming sequence completely without any unexpected interrupt, such as system reset or power down. If unexpected interrupt occurs, there is very little probability of corrupt the data stored in flash memory.

2.3.12 Option bytes erase

The FMC provides an erase function which is used to initialize the option bytes block in flash. The following steps show the erase sequence.

- Unlock the FMC_CTL register if necessary.
- Check the BUSY bit in the FMC_STAT register to confirm that no flash memory operation is in progress (BUSY equal to 0). Otherwise, wait until the operation has finished.
- Unlock the option bytes operation bits in the FMC_CTL register if necessary.
- Wait until the OBWEN bit is set in the FMC_CTL register.
- Set BKSEL bit in the FMC_CTL register.
- Set the OBER bit in the FMC_CTL register.
- Send the option bytes erase command to the FMC by setting the START bit in the FMC_CTL register.
- Wait until all the operations have been finished by checking the value of the BUSY bit in the FMC_STAT register.
- Read and verify the flash memory if required.

When the operation is executed successful, the ENDF bit in the FMC_STAT register is set, and an interrupt will be triggered by FMC if the ENDIE bit in the FMC_CTL register is set.

2.3.13 Option bytes modify

The FMC provides an erase / program function which is used to modify the option bytes block in flash. The MSB is the complement of the LSB in each pair. When the option bytes are modified, the MSB is generated by FMC automatically, not the value of input data. The following steps show the erase sequence.

- Unlock the FMC_CTL register if necessary.
- Check the BUSY bit in the FMC_STAT register to confirm that no flash memory operation is in progress (BUSY equals to 0). Otherwise, wait until the operation has finished.
- Unlock the option bytes operation bits in the FMC_CTL register if necessary.
- Wait until the OBWEN bit is set in the FMC_CTL register.
- Set the OBPG bit in the FMC_CTL register.
- A 128-bit data write at desired address.

- Wait until all the operations have been finished by checking the value of the BUSY bit in the FMC_STAT register.
- Read and verify the flash memory if required access.

When the operation is executed successfully, the ENDF bit in the FMC_STAT register is set, and an interrupt will be triggered by FMC if the ENDIE bit in the FMC_CTL register is set. Note that these programming errors may occur. The PGERR bit and PGAERR bit can be set which is similar to main flash programming.

The modified option bytes only take effect after a system reset or set the OBRDL bit.

2.3.14 Option bytes description

The option bytes block is reloaded to the FMC_OBSTAT and FMC_WP registers after each system reset, then the option bytes take effect. The complement option bytes are the opposite of the option bytes. When reload the option bytes, if the complement option byte and option byte do not match, the OBERR bit in the FMC_OBSTAT register will be set, and the option byte will be set to 0xFF. The OBERR bit will not be set if both the option bytes and its complement bytes are 0xFF. The following table shows the detail of option bytes.

Table 2-3. Option bytes

Address	Name	Factory value	Description
0x1fff f800	SPC	0xA5	Option bytes security protection value 0xA5 : No security protection Any value except 0xA5 or 0xCC : Protection level low 0xCC : Protection level high
0x1fff f801	SPC_N	0x5A	SPC complement value
0x1fff f802	USER	0xFF	[7:6]: LVD0T[1:0] (LVD0 voltage level select) 00: Reserved 01: Reserved 10: 2.90V/2.82V (rising/falling) 11: 4.31V/4.23V (rising/falling) Settings other than above are prohibited when the voltage detection 0 circuit is used. [5]: FWDGSPD_STDBY 0: Suspend the free watchdog in standby mode 1: Run the free watchdog in standby mode [4]: FWDGSPD_DPSLP 0: Suspend the free watchdog in deepsleep mode 1: Run the free watchdog in deepsleep mode [3]: LVD0EN 0: Voltage monitoring 0 reset is enabled after a reset 1: Voltage monitoring 0 reset is disabled after a

Address	Name	Factory value	Description
			reset [2]: nRST_STDBY 0: Generate a reset instead of entering standby mode 1: No reset when entering standby mode [1]: nRST_DPSLP 0: Generate a reset instead of entering deep-sleep mode 1: No reset when entering deep-sleep mode [0]: nFWDG_HW 0: Hardware free watchdog 1: Software free watchdog
0x1fff f803	USER_N	0xFF	USER complement value
0x1fff f804	DATA[7:0]	0xFF	User defined data bit 7 to 0
0x1fff f805	DATA_N[7:0]	0xFF	DATA complement value bit 7 to 0
0x1fff f806	DATA[15:8]	0xFF	User defined data bit 15 to 8
0x1fff f807	DATA_N[15:8]	0xFF	DATA complement value bit 15 to 8
0x1fff f808	WP[7:0]	0xFF	Page erase / program protection bit 7 to 0 0: Protection active 1: Unprotected
0x1fff f809	WP_N[7:0]	0xFF	WP complement value bit 7 to 0
0x1fff f80a	WP[15:8]	0xFF	Page erase / program protection bit 15 to 8
0x1fff f80b	WP_N[15:8]	0xFF	WP complement value bit 15 to 8
0x1fff f80c	WP[23:16]	0xFF	Page erase / program protection bit 23 to 16 WP[22:0]: Each bit is related to 8KB flash protection. These bits totally controls the first 184KB flash protection. WP[23]: Bit 23 controls the protection of the rest main flash memory.
0x1fff f80d	WP_N[23:16]	0xFF	WP complement value bit 23 to 16
0x1fff f80e	WP[31:24]	0xFF	Page erase / program protection bit 31 to 24 WP[31:24]: Each bit is related to 8KB flash protection. These bits totally controls the 64KB data flash protection.
0x1fff f80f	WP_N[31:24]	0xFF	WP complement value bit 31 to 24
0x1fff f810	USER_1	0xFF	[7:6]: nRST_MODE 00: GPIO: standard GPIO pad functionality, only internal RESET possible 01: Bidirectional reset: NRST pin configured in reset input/output mode (legacy internal RESET not propagated to the NSRT pin)

Address	Name	Factory value	Description
			10: Bidirectional reset: NRST pin configured in reset input/output mode (legacy internal RESET not propagated to the NRST pin) 11: Reset input only: a low level on the NRST pin generates system reset mode [5:3]: Reserved [2]: SWDMOD 0: SWD off 1: SWD normal work mode [1]: SRAMECCEN 0: Disable SRAM ECC 1: Enable SRAM ECC [0]: nWWDG_HW 0: Hardware window watchdog 1: Software window watchdog
0x1fff f811	USER_1_N	0xFF	USER_1 complement value
0x1fff f812	WWDGT_0	0xFF	[7]: WWDGT_WDGTEN: 0: Disable WWDGT to generate a reset 1: Enable WWDGT to generate a reset [6]: WWDGT_EWIE 0: Disable early wakeup interrupt 1: Enable early wakeup interrupt [5:4]: WWDGT_WEPS: WWDGT window end position in hardware start mode. 00: 75% 01: 50% 10: 25% 11: 0% (No window end position setting) [3:2]: WWDGT_WSPS: WWDGT window start position in hardware start mode. 00: 25% 01: 50% 10: 75% 11: 100% (No window start position setting) [1:0]: WWDGT_PSC: WWDGT prescaler in hardware start mode.

Address	Name	Factory value	Description
0x1fff f813	WWDGT_0_N	0xFF	WWDGT_0 complement value
0x1fff f814	WWDGT_1	0xFF	[7:0]: WWDGT_CNT [13:6] The value of the window watchdog timer counter bit 13 to 6 in hardware start mode.
0x1fff f815	WWDGT_1_N	0xFF	WWDGT_1 complement value
0x1fff f816	WWDGT_2	0xFF	[7:6]: Reserved [5:0]: WWDGT_CNT [5:0] The value of the watchdog timer counter bit 5 to 0 in hardware start mode.
0x1fff f817	WWDGT_2_N	0xFF	WWDGT_2 complement value
0x1fff f818	FWDGT_0	0xFF	[7:0]: FWDGT_RLD [7:0] The value of the free watchdog timer counter reload value bit 7 to 0 in hardware start mode.
0x1fff f819	FWDGT_0_N	0xFF	FWDGT_0 complement value
0x1fff f81a	FWDGT_1	0xFF	[7]: FWDGT_WDGTEN 0: Disable FWDGT to generate a reset 1: Enable FWDGT to generate a reset [6:4]: FWDGT_PSC [2:0] Free watchdog timer prescaler selection. [3:0]: FWDGT_WND [3:0] The value of the free watchdog counter window value bit 3 to 0 in hardware start mode.
0x1fff f81b	FWDGT_1_N	0xFF	FWDGT_1 complement value
0x1fff f81c	FWDGT_2	0xFF	[7:0]: FWDGT_WND [11:4] The value of the free watchdog counter window value bit 11 to 4 in hardware start mode.
0x1fff f81d	FWDGT_2_N	0xFF	FWDGT_2 complement value
0x1fff f81e	FWDGT_3	0xFF	[3:0]: FWDGT_RLD [11:8] The value of the free watchdog timer counter reload value bit 11 to 8 in hardware start mode.
0x1fff f81f	FWDGT_3_N	0xFF	FWDGT_3 complement value

Note: The detailed description of FWDGT and WWDGT bits, please refer to [Watchdog timer \(WDGT\)](#).

2.3.15 Page erase/program protection

The FMC provides page erase / program protection functions to prevent inadvertent operations on the flash memory. The page erase or program will not be accepted by the FMC on protected pages. If the page erase or program command is sent to the FMC on a protected page, the WPERR bit in the FMC_STAT register will be set by the FMC. If the WPERR bit is set and the ERRIE bit is also set to 1 to enable the corresponding interrupt, then the flash

operation error interrupt will be triggered by the FMC to draw the attention of the CPU. The page protection function can be individually enabled by configuring the WP [31:0] bit field to 0 in the option bytes. If a page erase operation is executed on the option bytes block, all the flash Memory page protection functions will be disabled. When WP in the option bytes is modified, then a system reset is necessary.

Table 2-4. OB_WP bit for pages protected

OB_WP bit	pages protected
OB_WP[0]	main flash page 0 ~ 7
OB_WP[1]	main flash page 8 ~ 15
OB_WP[2]	main flash page 16 ~ 23
.	.
.	.
.	.
OB_WP[22]	main flash page 176 ~ 183
OB_WP[23]	main flash page 184 ~ 256
OB_WP[24]	data flash page 0 ~ 7
OB_WP[25]	data flash page 8 ~ 15
OB_WP[26]	data flash page 16 ~ 23
.	.
.	.
.	.
OB_WP[30]	data flash page 48 ~ 55
OB_WP[31]	data flash page 56 ~ 63

2.3.16 Security protection

The FMC provides a security protection function to prevent illegal code / data access to the flash memory. This function is useful for protecting the software / firmware from illegal users.

No protection: when setting SPC byte and its complement value to 0x5AA5, no protection performed. The main flash and option bytes block are accessible by all operations.

Protection level low: when setting SPC byte value to any value except 0xA5 or 0xCC, the low security protection is performed. Note that a power reset should be followed instead of a system reset if the SPC modification has been performed while the debug module is still connected to JTAG / SWD device. Under the low security protection, the main flash can only be accessed by user code and the first 8KB flash is under erase / program protection. In debug mode, boot from SRAM or boot loader mode, all operations to main flash is forbidden. If a read operation to main flash in debug mode, boot from SRAM or boot loader mode, a bus error will be generated. If a program / erase operation to main flash in debug mode, boot from SRAM or boot from boot loader mode, the WPERR bit in the FMC_STAT register will be set. Option bytes block are accessible by all operations, which can be used to disable the security protection. Back to no protection level by setting SPC byte and its complement value to 0x5AA5, then a mass erase for main flash will be performed.

Protection level high: when setting SPC byte to 0xCC, protection level high performed. When this level is programmed, debug mode, boot from SRAM or boot from boot loader mode are disabled. The main flash block is accessible by all operations from user code. The SPC byte cannot be reprogrammed. So, if protection level high is programmed, it cannot move back to protection level low or no protection level.

2.4 Register definition

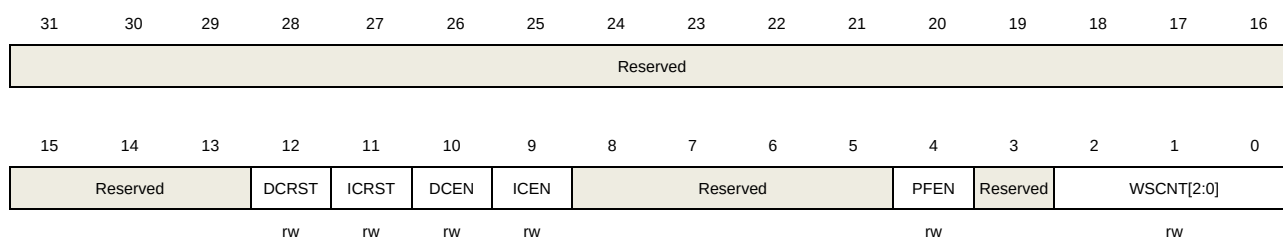
FMC base address: 0x4002 2000

2.4.1 Wait state register (FMC_WS)

Address offset: 0x00

Reset value: 0x0000 0611

This register has to be accessed by word (32-bit).



Bits	Fields	Descriptions
31:13	Reserved	Must be kept at reset value.
12	DCRST	D-cache reset. This bit can be write only when DCEN is set to 0. 0: No effect 1: D-cache reset
11	ICRST	I-cache reset. This bit can be write only when ICEN is set to 0. 0: No effect 1: I-cache reset
10	DCEN	D-cache enable 0: D-cache disable 1: D-cache enable
9	ICEN	I-cache enable 0: I-cache disable 1: I-cache enable
8:5	Reserved	Must be kept at reset value.
4	PFEN	Pre-fetch enable 0: Pre-fetch disable 1: Pre-fetch enable
3	Reserved	Must be kept at reset value.
2:0	WSCNT[2:0]	Wait state counter register These bits are set and reset by software. 000: 0 wait state added

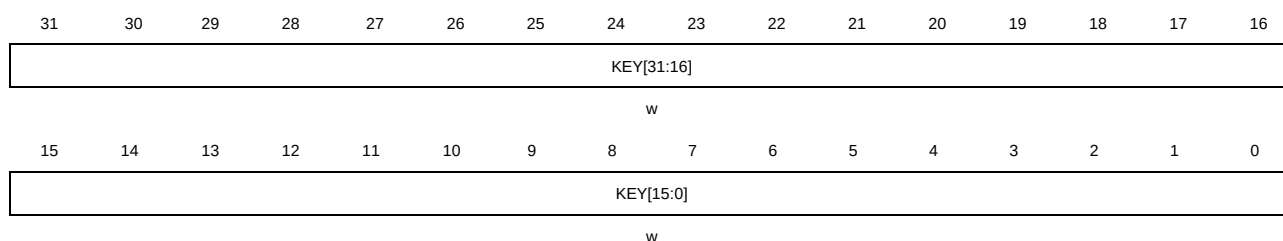
001: 1 wait state added
010: 2 wait state added
011: 3 wait state added
100: 4 wait state added
101: 5 wait state added
110: 6 wait state added
111: reserved

2.4.2 Unlock key register (FMC_KEY)

Address offset: 0x04

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit)



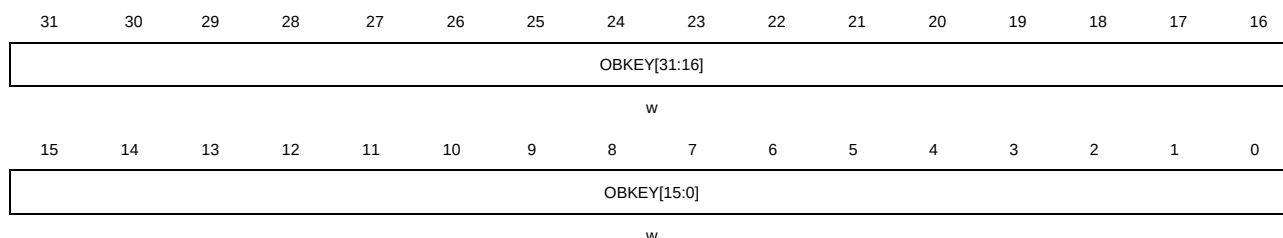
Bits	Fields	Descriptions
31:0	KEY[31:0]	FMC_CTL unlock register These bits are only written by software. Write KEY[31:0] with keys to unlock FMC_CTL register.

2.4.3 Option byte unlock key register (FMC_OBKEY)

Address offset: 0x08

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit)



Bits	Fields	Descriptions
31:0	OBKEY[31:0]	FMC_CTL option bytes operation unlock register These bits are only written by software. Write OBKEY[31:0] with keys to unlock option bytes command in the FMC_CTL

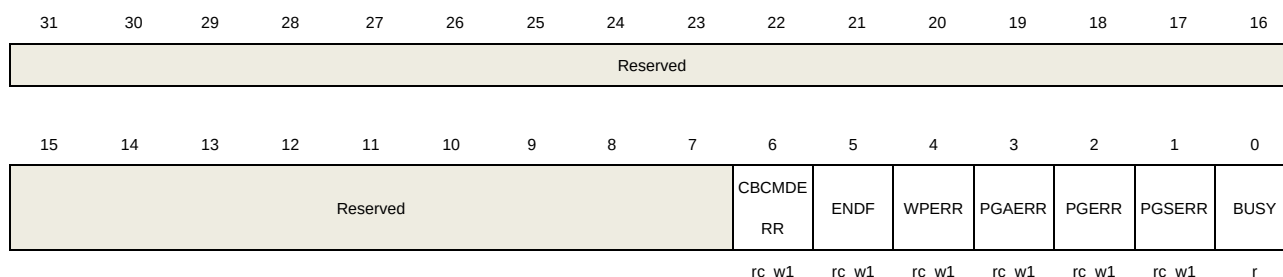
register.

2.4.4 Status register (FMC_STAT)

Address offset: 0x0C

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit)



Bits	Fields	Descriptions
31:7	Reserved	Must be kept at reset value.
6	CBCMDERR	The checked area by the check blank command is all 0xFF or not. 0: The checked area is all 0xFF. 1: The checked area is not all 0xFF.
5	ENDF	End of operation flag bit When the operation executed successfully, this bit is set by hardware. The software can clear it by writing 1.
4	WPERR	Erase / Program protection error flag bit When erase / program on protected pages, this bit is set by hardware. The software can clear it by writing 1.
3	PGAERR	Program alignment error flag bit This bit is set by hardware when write data is not alignment. The software can clear it by writing 1.
2	PGERR	Program error flag bit When program to the flash while it is not 0xFFFF, this bit is set by hardware. The software can clear it by writing 1.
1	PGSERR	Programming sequence error flag bit When the programming sequence is wrong, this bit is set by hardware. The software can clear it by writing 1.
0	BUSY	The flash is busy bit When the operation is in progress, this bit is set to 1. When the operation is end or an error is generated, this bit is cleared to 0.

2.4.5 Control register (FMC_CTL)

Address offset: 0x10

Reset value: 0x0000 0080

This register has to be accessed by word (32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CBCMDLEN[2:0]			Reserved											OBRLD	CBCMD
rw														rc_w1	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BKSEL	Reserved		ENDIE	Reserved	ERRIE	OBWEN	FSTPG	LK	START	OBER	OBPG	Reserved	MER	PER	PG
rw			rw		rw	rw	rw	rs	rs	rw	rw		rw	rw	rw

Bits	Fields	Descriptions
31:29	CBCMDLEN[2:0]	CBCMD read length 2 ^{CBCMDLEN} . The read length by check blank command. The read length is 2 ^{CBCMDLEN} four words.
28:18	Reserved	Must be kept at reset value.
17	OBRLD	Option byte reload bit This bit is set by software. 0: No effect 1: Force option byte reload and generate a system reset Note: It cannot be written if OBLK is set.
16	CBCMD	The command to check the selected area is blank or not.
15	BKSEL	Select the flash bank for the operation. This bit cannot be modified when flash is busy. 0: MER/PER/FSTPG/PG for main flash. 1: MER/PER/FSTPG/PG for data flash.
14:13	Reserved	Must be kept at reset value.
12	ENDIE	End of operation interrupt enable bit This bit is set or cleared by software 0: No interrupt generated by hardware. 1: End of operation interrupt enable
11	Reserved	Must be kept at reset value.
10	ERRIE	Error interrupt enable bit This bit is set or cleared by software 0: No interrupt generated by hardware. 1: Error interrupt enable
9	OBWEN	Option byte erase / program enable bit

		This bit is set by hardware when right sequence written to the FMC_OBKEY register. This bit can be cleared by software.
8	FSTPG	Flash fast program command bit This bit is set or clear by software 0: No effect 1: Flash fast program command
7	LK	FMC_CTL lock bit This bit is cleared by hardware when right sequence written to the FMC_KEY register. This bit can be set by software.
6	START	Send erase command to FMC bit This bit is set by software to send erase command to FMC. This bit is cleared by hardware when the BUSY bit is cleared.
5	OBER	Option bytes erase command bit This bit is set or clear by software 0: No effect 1: Option byte erase command
4	OBPG	Option bytes program command bit This bit is set or clear by software 0: No effect 1: Option bytes program command
3	Reserved	Must be kept at reset value.
2	MER	Flash mass erase command bit This bit is set or cleared by software 0: No effect 1: Flash mass erase command
1	PER	Flash page erase command bit This bit is set or clear by software 0: No effect 1: Flash page erase command
0	PG	Flash program command bit This bit is set or clear by software 0: No effect 1: Flash program command

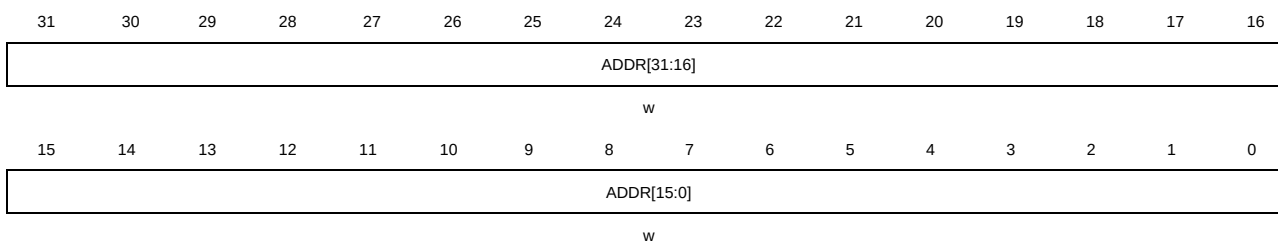
Note: This register should be reset after the corresponding flash operation completed.

2.4.6 Address register (FMC_ADDR)

Address offset: 0x14

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit)



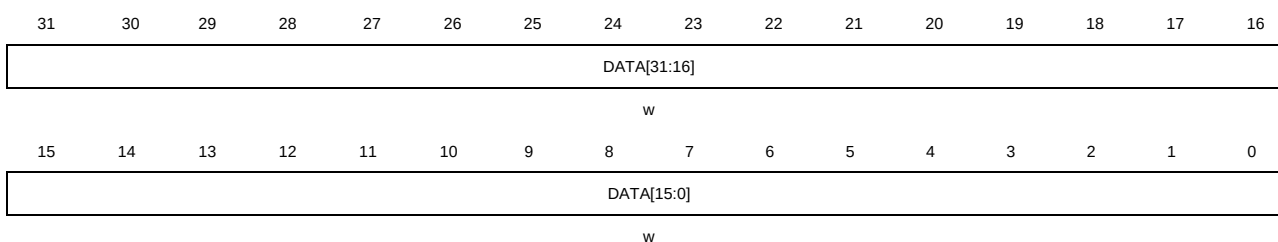
Bits	Fields	Descriptions
31:0	ADDR[31:0]	Flash erase / program command address bits These bits are configured by software. ADDR bits are the address of flash to be erased / programmed.

2.4.7 Program data register (FMC_DATA)

Address offset: 0x18

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit)



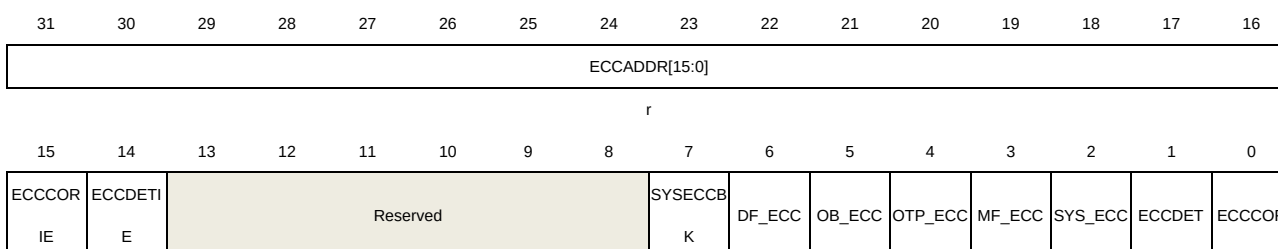
Bits	Fields	Descriptions
31:0	DATA[31:0]	Flash erase / program command DATA bits These bits are configured by software. This is used in main flash fast program.

2.4.8 ECC control and status register (FMC_ECCCS)

Address offset: 0x18

Reset value: 0x0000 0000

This register has to be accessed by word(32-bit).



Bits	Field	Descriptions
31:16	ECCADDR[15:0]	The offset address of four word where an ECC error is detected. Error address = base address + ECCADDR[15:0] * 8, the base address can be the start address of main Flash, system area, option bytes and OTP. Main flash: 0 ~ 0x7FFF (256K/8 – 1) Data flash: 0 ~ 0x1FFF (64K/8 – 1) Option bytes: 0 ~ 0x01 (16/8 -1) OTP: 0 ~ 0x3F (512/8 -1) System memory: 0 ~ 0x6FF (14K/8 – 1)
15	ECCCORIE	One bit error correct interrupt enable. 0: One bit error correct interrupt disable. 1: One bit error correct interrupt enable.
14	ECCDETIE	Two bits errors detect interrupt enable. 0: Two bits errors detect interrupt disable. 1: Two bits errors detect interrupt enable.
13:8	Reserved	Must be kept at reset value.
7	SYSECCBK	This bit in read only to indicate whether the system memory ECC error occurs in the first 3k or later 3k region of the information block. 0: System memory ECC error occurs in the first 3K region of the information block. 1: System memory ECC error occurs in the later 3K region of the information block.
6	DF_ECC	If an ECC bit error is detected in data flash, this bit will be set. And the ECCADDR records the offset address of main Flash. ECCDET bit and ECCCOR bit indicate there is a one bit error or a two bits error. This bit is cleared by writing 1. 0: No ECC error is detected in data flash. 1: An ECC bit error is detected in data flash.
5	OB_ECC	If an ECC bit error is detected in reading option bytes, this bit will be set. And the ECCADDR records the offset address of option bytes. ECCDET bit and ECCCOR bit indicate there is a one bit error or a two bits error. This bit is cleared by writing 1. This bit is cleared by writing 1. 0: No ECC error is detected in reading option bytes. 1: An ECC bit error is detected in reading option bytes.
4	OTP_ECC	If an ECC bit error is detected in OTP, this bit will be set. And the ECCADDR records the offset address of OTP. ECCDET bit and ECCCOR bit indicate there is a one bit error or a two bits error. This bit is cleared by writing 1. 0: No ECC error is detected in OTP. 1: An ECC bit error is detected in OTP.
3	MF_ECC	If an ECC bit error is detected in main Flash, this bit will be set. And the ECCADDR records the offset address of main Flash. ECCDET bit and ECCCOR bit indicate

		there is a one bit error or a two bits error. This bit is cleared by writing 1. 0: No ECC error is detected in main Flash. 1: An ECC bit error is detected in main Flash.
2	SYS_ECC	If an ECC bit error is detected in system memory, this bit will be set. And the ECCADDR records the offset address of system memory. ECCDET bit and ECCCOR bit indicate there is a one bit error or a two bits error. This bit is cleared by writing 1. 0: No ECC error is detected in system memory. 1: An ECC bit error is detected in system memory.
1	ECCDET	Two bit errors detect flag. This bit set when two bit errors is detected. This bit is cleared by writing 1. 0: Two ECC errors are not detected. 1: Two ECC errors are detected.
0	ECCCOR	One bit error detected and correct flag. This bit is cleared by writing 1. 0: No ECC error is detected and corrected. 1: An ECC error is detected and corrected.

2.4.9 Option byte status register (FMC_OBSTAT)

Address offset: 0x38

Reset value: 0x0XXX XXXX.

This register has to be accessed by word(32-bit)



Bits	Fields	Descriptions
31:26	Reserved	Must be kept at reset value.
25:10	DATA[15:0]	Store DATA[15:0] of option bytes block after system reset.
9:2	USER[7:0]	Store USER of option bytes block after system reset.
1	SPC	Option bytes security protection code 0: No protection 1: Protection
0	OBERR	Option bytes read error bit. This bit is set by hardware when the option bytes and its complement byte do not

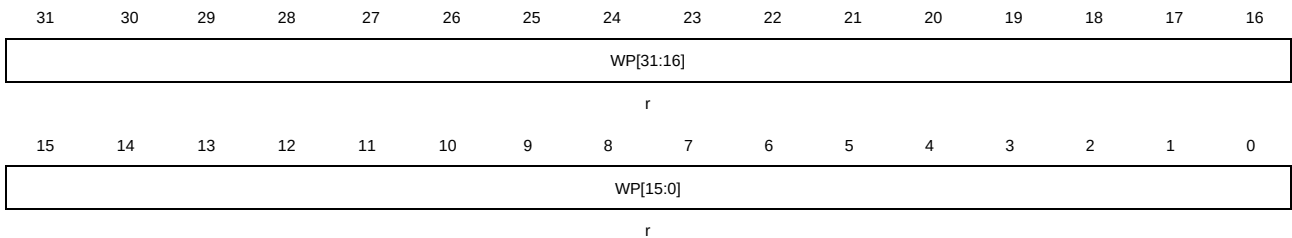
match, then the option bytes is set to 0xFF.

2.4.10 Erase / Program protection register (FMC_WP)

Address offset: 0x3C

Reset value: 0xFFFF XXXX

This register has to be accessed by word(32-bit)



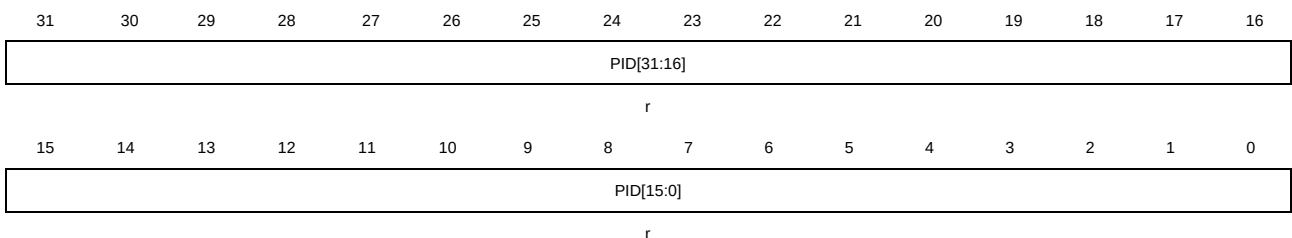
Bits	Fields	Descriptions
31:0	WP[31:0]	Store WP[31:0] of option bytes block after system reset

2.4.11 Product ID register 0 (FMC_PID0)

Address offset: 0x100

Reset value: 0xFFFF XXXX

This register has to be accessed by word(32-bit)



Bits	Field	Descriptions
31:0	PID[31:0]	Product reserved ID code register These bits are read only by software. These bits are unchanged constant after power on. These bits are one time program when the chip produced.

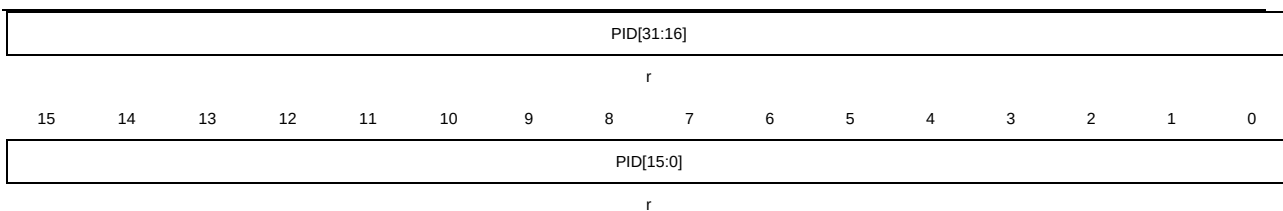
2.4.12 Product ID register 1 (FMC_PID1)

Address offset: 0x104

Reset value: 0xFFFF XXXX

This register has to be accessed by word(32-bit)





Bits	Field	Descriptions
31:0	PID[31:0]	Product reserved ID code register These bits are read only by software. These bits are unchanged constant after power on. These bits are one time program when the chip produced.

3 Power management unit (PMU)

3.1 Overview

The power consumption is regarded as one of the most important issues for the devices of GD32M531 series. Power management unit (PMU) provides three types of power saving modes, including Sleep, Deep-sleep and Standby mode. These modes reduce the power consumption and allow the application to achieve a best tradeoff among the conflicting demands of CPU operating time, speed and power consumption. For GD32M531 devices, there are two power domains, including V_{DD} / V_{DDA} domain and V_{CORE} domain, as is shown in the [Figure 3-1. Power supply overview](#). The power of the V_{DD} domain is supplied directly by V_{DD} . An embedded LDO in the V_{DD}/V_{DDA} domain is used to supply the V_{CORE} domain power.

3.2 Characteristics

- Two power domains: V_{DD} / V_{DDA} domain and V_{CORE} domain.
- Three power saving modes: Sleep, Deep-sleep and Standby modes.
- Internal Voltage regulator (LDO) supplies around 1.2V voltage source for V_{CORE} domain.
- Three Low Voltage Detectors (LVD) to detect low voltage events. LVD1 / LVD2 can generate interrupts. LVD0 / LVD1 / LVD2 can generate a system reset.

3.3 Function overview

[Figure 3-1. Power supply overview](#) provides details on the internal configuration of the PMU and the relevant power domains.

The diagram illustrates the power domain architecture. It shows three main voltage domains: V_{DD} , V_{DDA0} , and V_{DDA2} . Each domain has a corresponding input pin (VDD, VDDA0, VDDA2) and a 5V reference point. The V_{DD} domain contains the PMU CTL, FWDGT, HXTAL, POR/PDR, LDO, IRC32M, IRC32K, LVD, and PLL. The V_{DDA0} domain contains ADC0. The V_{DDA2} domain contains ADC2, DAC, and CMP. The V_{CORE} domain contains BSRAM, Cortex-M33, AHB IPs, and APB IPs. Signals include WKUP, WKUPN, WKUPF, SLEEPING, SLEEPDEEP, and V_{CORE} .

Legend:

- LVD: Low Voltage Detector
- LDO: Voltage Regulator
- POR: Power On Reset
- PDR: Power Down Reset

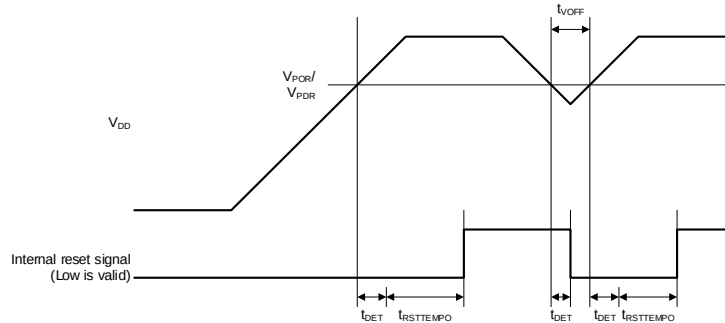
V_{DD} / V_{DDA} domain includes two parts: V_{DD} domain and V_{DDA} domain. V_{DD} domain includes HXTAL (High Speed Crystal oscillator), LDO (Voltage Regulator), POR / PDR (Power On / Down Reset), FWDGT (Free Watchdog Timer), and all PADS except PD4 / PD5, PC0-PC12, as well as the IRC32M (internal 32M RC oscillator), IRC32K (internal 32KHz RC oscillator), PLLs (phase-locked loops), and LVD (low voltage detector), etc. The V_{DDA} domain includes two parts: V_{DDA0} and V_{DDA2} , where V_{DDA0} includes ADC0, and V_{DDA2} includes ADC2, DAC (analog-to-digital / digital-to-analog converter), CMP (comparator), and so on.

The LDO, which is implemented to supply power for the V_{CORE} domain, is always enabled after the reset. It can be configured to operate in three different status, including the Sleep mode (full power on), in the Deep-sleep mode (on or low power), and in the Standby mode (power off).

The POR / PDR circuit is implemented to detect V_{DD} / V_{DD} and generate the power reset signal which resets the whole chip when the supply voltage is lower than the specified threshold. **Figure 3-2. Waveform of the POR / PDR** shows the relationship between the supply voltage and the power reset signal. V_{POR} represents the threshold of power on reset, V_{PDR} represents

the threshold of power down reset, $V_{PDRhyst}$ represents the hysteresis value. Please refer to the datasheet for the values of these parameters.

Figure 3-2. Waveform of the POR / PDR



- (1) The state of the internal reset signal can be approximated by monitoring the GPIO flag value. By measuring the time difference between the V_{POR} and the low-to-high transition of the GPIO flag, the sum of t_{DET} and $t_{RSTTEMPO}$ can be roughly validated.
- (2) The voltage on the NRST pin does not strictly equate to the internal reset signal state.

VDDA domain

Generally, digital circuits are powered by V_{DD} , while most of analog circuits are powered by V_{DDA} . To improve the ADC and DAC conversion accuracy, the independent power supply V_{DDA} is implemented to achieve better performance of analog circuits. V_{DDA} can be externally connected to V_{DD} through the external filtering circuit that avoids noise on V_{DDA} , and V_{SSA} should be connected to V_{SS} through the specific circuit independently. Otherwise, when the V_{DD} and V_{DDA} are provided by different power supplies, the difference between V_{DD} and V_{DDA} during power-up and running time should not exceed 0.3V.

To ensure a high accuracy on ADC and DAC, the ADC / DAC independent external reference voltage should be connected to VREFP / VREFN pins. According to the different packages, VREFP pin can be connected to V_{DDA} pin, or external reference voltage which refers to [Table 15-2. ADC pins definition](#) and [Table 16-1. DAC I/O description](#), VREFN pin must be connected to V_{SSA} pin.

3.3.2 V_{CORE} power domain

The V_{CORE} power domain supplies power for Cortex®-M33 logic, AHB / APB peripherals, the APB interfaces for the V_{DD} / V_{DDA} domain, etc. Once the V_{CORE} domain is powered up, the POR will generate a reset sequence on the V_{CORE} power domain. If it is needed to enter the specified power saving mode, the associated control bits must be configured. Then, once a WFI (Wait for Interrupt) or WFE (Wait for Event) instruction is executed, the device will enter the specified power saving mode which will be discussed in the following section. Refer to the datasheet for the V_{CORE} voltage value.

3.3.3 Power saving modes

After a system reset or a power reset, the GD32M531 MCU operates at full function state and all power domains are active. Users can achieve lower power consumption through slowing down the system clocks (HCLK, PCLK1 and PCLK2), closing the clocks of the unused peripherals. Besides, three power saving modes are provided to achieve even lower power consumption. They are Sleep mode, Deep-sleep mode and Standby mode.

Sleep mode

The Sleep mode is corresponding to the SLEEPING mode of the Cortex®-M33. In Sleep mode, only clock of Cortex®-M33 is off. To enter the Sleep mode, it is only necessary to clear the SLEEPDEEP bit in the Cortex®-M33 System Control Register, and execute a WFI or WFE instruction. If the Sleep mode is entered by executing a WFI instruction, any interrupt can wake up the system. If it is entered by executing a WFE instruction, any wakeup event can wake up the system (If SEVONPEND is 1, any interrupt can wake up the system, refer to Cortex®-M33 Technical Reference Manual). The mode costs the lowest wakeup time as no time is wasted in interrupt entry or exit.

According to the SLEEPONEXIT bit in the Cortex®-M33 System Control Register, there are two options to select the Sleep mode entry mechanism.

- Sleep-now: if the SLEEPONEXIT bit is cleared, the MCU enters Sleep mode as soon as a WFI or WFE instruction is executed.
- Sleep-on-exit: if the SLEEPONEXIT bit is set, the MCU enters Sleep mode as soon as it exits from the ISR with the lowest priority.

Deep-sleep mode

The Deep-sleep mode is based on the SLEEPDEEP mode of the Cortex®-M33. In Deep-sleep mode, all clocks in the V_{CORE} domain are off, and all of IRC32M, HXTAL and PLLs are disabled. The contents of SRAM and registers are preserved. Before entering the Deep-sleep mode, it is necessary to set the SLEEPDEEP bit in the Cortex®-M33 System Control Register, and clear the STBMOD bit in the PMU_CTL register. Then, the device enters the Deep-sleep mode after a WFI or WFE instruction is executed. If the Deep-sleep mode is entered by executing a WFI instruction, any interrupt from EXTI lines can wake up the system. If it is entered by executing a WFE instruction, any wakeup event from EXTI lines can wake up the system (If SEVONPEND is 1, any interrupt from EXTI lines can wake up the system, refer to Cortex®-M33 Technical Reference Manual). When exiting the Deep-sleep mode, the IRC32M is selected as the system clock. Notice that an additional wakeup delay will be incurred if the LDO operates in low power mode.

Note: In order to enter Deep-sleep mode smoothly, all EXTI line pending status (in the EXTI_PD register) and related peripheral flags must be reset, refer to [Table 6-3. EXTI source](#). If not, the program will skip the entry process of Deep-sleep mode to continue to execute the following procedure.

Standby mode

The Standby mode is also based on the SLEEPDEEP mode of the Cortex®-M33. In Standby mode, the whole V_{CORE} domain is powered off, the LDO is shut down, and all of IRC32M, HXTAL and PLL are disabled. Before entering the Standby mode, it is necessary to set the SLEEPDEEP bit in the Cortex®-M33 System Control Register, set the STBMOD bit in the PMU_CTL register, and clear WUF bit in the PMU_CS register. Then, the device enters the Standby mode after a WFI or WFE instruction is executed, and the STBF status flag in the PMU_CS register indicates whether the MCU has been in Standby mode. There are three wakeup sources for the Standby mode, including the external reset from NRST pin, the FWDGT reset and the rising edge on WKUP pin. The Standby mode achieves the lowest power consumption, but spends longest time to wake up. Besides, the contents of SRAM and registers in V_{CORE} domain are lost in Standby mode. When exiting from the Standby mode, a power-on reset occurs and the Cortex®-M33 will execute instruction code from the 0x0000 0000 address.

Table 3-1. Power saving mode summary

Mode	Sleep	Deep-sleep	Standby
Description	Only CPU clock is off	1. All clocks in the V_{CORE} domain are off 2. Disable IRC32M, HXTAL and PLL	1. The V_{CORE} domain is powered off 2. Disable IRC32M, HXTAL and PLL
LDO Status	On (normal power mode)	On (normal or low power mode)	Off
Configuration	SLEEPDEEP = 0	SLEEPDEEP = 1 STBMOD = 0	SLEEPDEEP = 1 STBMOD = 1, WURST=1
Entry	WFI or WFE	WFI or WFE	WFI or WFE
Wakeup	Any interrupt for WFI. Any event (or interrupt when SEVONPEND is 1) for WFE	Any interrupt from EXTI lines for WFI Any event(or interrupt when SEVONPEND is 1) from EXTI for WFE	1. NRST pin 2. WKUP pin 3. FWDGT reset
Wakeup Latency	None	IRC32M wakeup time, LDO wakeup time added if LDO is in low power mode	Power on sequence

Note: In Standby mode, all I / Os are in high-impedance state except NRST pin, and WKUP pin if enabled.

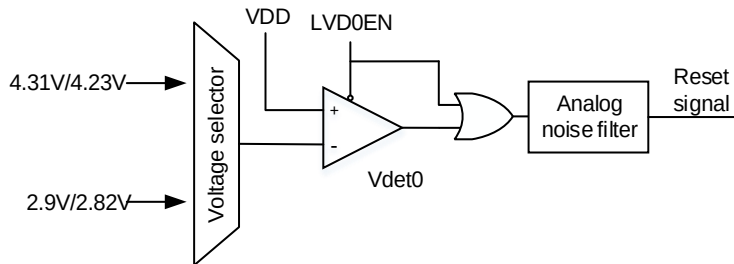
3.3.4 LVD detectors

Three Low Voltage Detectors (LVD) to detect low voltage events.

LVD0

The LVD0 is used to detect whether the V_{DD} supply voltage is lower than a programmed threshold selected by the LVD0T[7:6] bits in the option bytes. If LVD0EN in the option byte is set to 0, a system reset will be generated when LVD0 detects that the voltage is lower than the set threshold voltage.

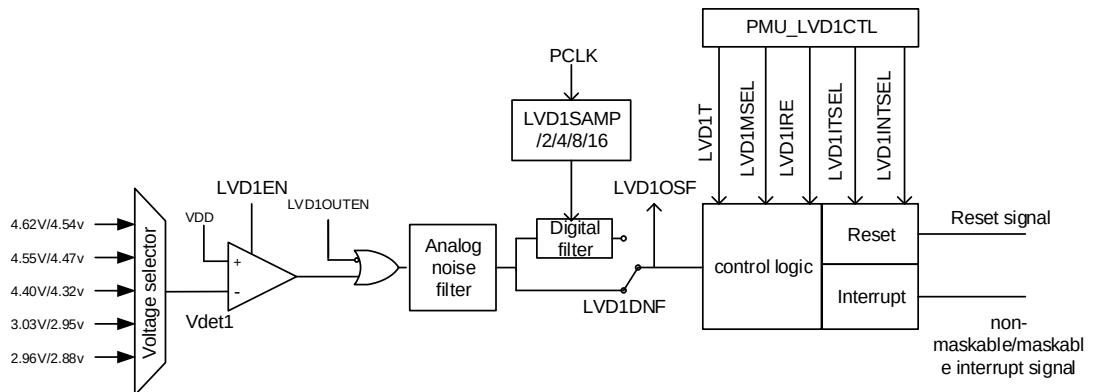
Figure 3-3. Low Voltage Detectors 0 block diagram



LVD1

The Low Voltage Detectors 1 (LVD1) is used to detect whether the V_{DD} / V_{DDA} supply voltage is lower or higher than a programmed threshold selected by the LVD1T[3:0] bits in the Low Voltage Detector 1 control register (PMU_LVD1CTL). When LVD1 is enabled, if $V_{DD} / V_{DDA} > V_{det1}$ or $V_{DD} / V_{DDA} \leq V_{det1}$ is detected, LVD1 can generate a system reset or interrupt.

Figure 3-4. Low Voltage Detectors 1 block diagram



Through the following register setting sequence, LVD1 can generate an interrupt.

1. Unlock the PMU_LVD1CTL by setting the PRCFG2 bits in the SYSCFG_PRCFG.
2. Select the detection voltage threshold for LVD1 by setting the LVD1T[3:0] bits.
3. Select the interrupt mode of LVD1 by setting the LVD1MSEL bit.
4. Set the condition for generating LVD1 interrupts by configuring the LVD1INTSEL[1:0] bits.
5. Set the type of interrupt by configuring the LVD1ITSEL bit.
6. Enable LVD1 by setting the LVD1EN bit.
7. Enable LVD1 comparison result output by setting the LVD1OUTEN bit.
8. Set the LVD1DETF bit to 0.

9. Enable LVD1 interrupt by setting the LVD1IRE bit.

Note: Step 1.2.5 is not required in the following two situations:

- a. Monitoring is restarted without changing the monitoring settings after stopping monitoring.
- b. After monitoring is stopped, changing LVD1ITSEL and LVD1INTSEL monitoring will restart.

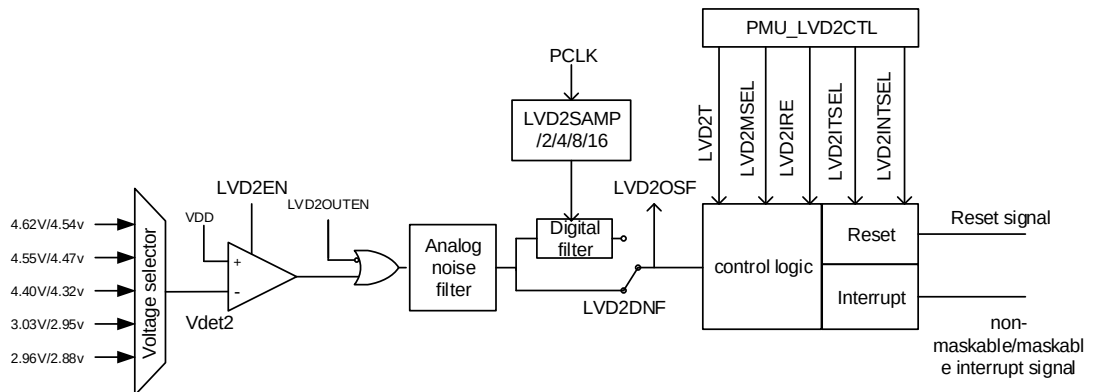
Through the following step register setting sequence, LVD1 can generate a system reset.

1. Select the detection voltage threshold for LVD1 by setting the LVD1T[3:0] bits.
2. Select the Reset mode of LVD1 by setting the LVD1MSEL bit.
3. Set the type of reset negation by setting the LVD1RSTSEL bit.
4. Enable LVD1 Reset mode by setting the LVD1IRE bit.
5. Enable LVD1 by setting the LVD1EN bit.
6. Enable LVD1 comparison result output by setting the LVD1OUTEN bit.

LVD2

The Low Voltage Detectors 2 (LVD2) is used to detect whether the V_{DD} / V_{DDA} supply voltage is lower or higher than a programmed threshold selected by the LVD2T[3:0] bits in the Low Voltage Detector 2 control register(PMU_LVD2CTL). When LVD2 is enabled, if V_{DD} / V_{DDA} > Vdet2 or V_{DD} / V_{DDA} <= Vdet2 is detected, LVD2 can generate a system reset or interrupt.

Figure 3-5. Low Voltage Detectors 2 block diagram



Through the following register setting sequence, LVD2 can generate an interrupt.

1. Unlock the PMU_LVD2CTL by setting the PRCFG2 bits in the SYSCFG_PRCFG.
2. Select the detection voltage threshold for LVD2 by setting the LVD2T[1:0] bits.
3. Select the interrupt mode of LVD2 by setting the LVD2MSEL bit.
4. Set the condition for generating LVD2 interrupts by configuring the LVD2INTSEL[1:0] bits.
5. Set the type of interrupt by configuring the LVD2ITSEL bit.
6. Enable LVD2 by setting the LVD2EN bit.
7. Enable LVD2 comparison result output by setting the LVD2OUTEN bit.
8. Set the LVD2DETF bit to 0.

9. Enable LVD2 interrupt by setting the LVD2IRE bit.

NOTE: Step 1.2.5 is not required in the following two situations:

- a. Monitoring is restarted without changing the monitoring settings after stopping monitoring.
- b. After monitoring is stopped, changing LVD2ITSEL and LVD2INTSEL monitoring will restart.

Through the following step register setting sequence, LVD2 can generate a system reset.

1. Select the detection voltage threshold for LVD2 by setting the LVD2T[1:0] bits.
2. Select the Reset mode of LVD2 by setting the LVD2MSEL bit.
3. Set the type of reset negation by setting the LVD2RSTSEL bit.
4. Enable LVD2 Reset mode by setting the LVD2IRE bit.
5. Enable LVD2 by setting the LVD2EN bit.
6. Enable LVD2 comparison result output by setting the LVD2OUTENbit.

3.4 Register definition

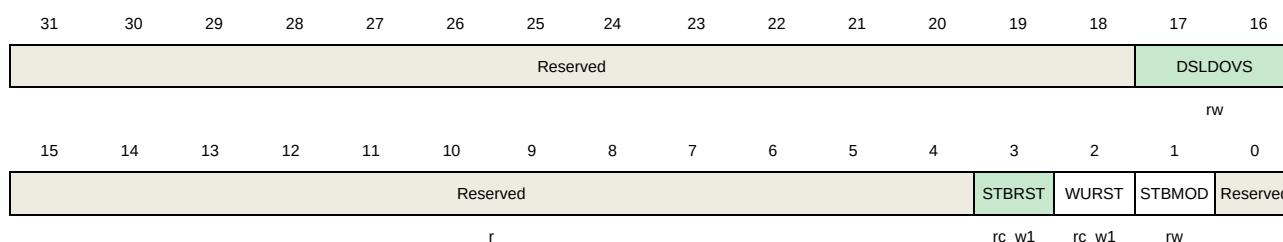
PMU base address: 0x4000 7000

3.4.1 Control register (PMU_CTL)

Address offset: 0x00

Reset value: 0x0001 0000 (reset after wakeup from Standby mode)

This register can be accessed by word(32-bit)



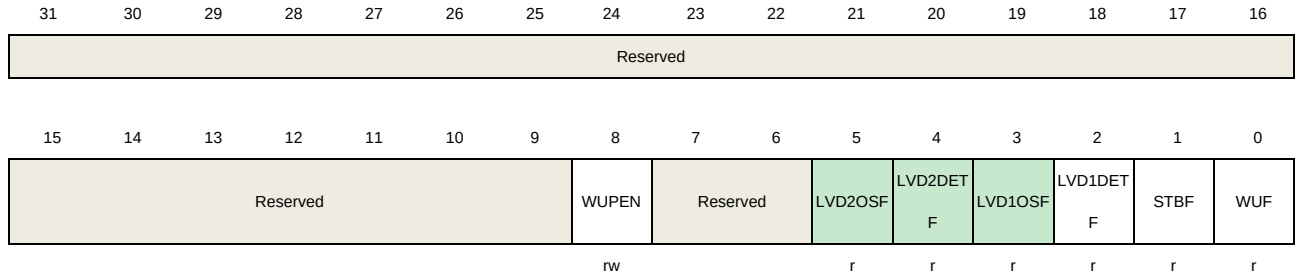
Bits	Fields	Descriptions
31:18	Reserved	Must be kept at reset value.
17:16	DSLDOVS[1:0]	Deep-sleep mode voltage scaling selection 00: LDO output voltage is 0.9V 01: LDO output voltage is 1.0V 10: LDO output voltage is 1.1V 11: LDO output voltage is 1.2V
15:4	Reserved	Must be kept at reset value.
3	STBRST	Standby flag reset 0: No effect 1: Reset the standby flag This bit is always read as 0.
2	WURST	Wakeup flag reset 0: No effect 1: Reset the wakeup flag This bit is always read as 0.
1	STBMOD	Standby mode 0: Enter the Deep-sleep mode when the Cortex®-M33 enters SLEEPDEEP mode 1: Enter the Standby mode when the Cortex®-M33 enters SLEEPDEEP mode
0	Reserved	Must be kept at reset value.

3.4.2 Control and status register (PMU_CS)

Address offset: 0x04

Reset value: 0x0000 0028 (will not reset after wakeup from Standby mode)

This register can be accessed by word(32-bit).



Bits	Fields	Descriptions
31:9	Reserved	Must be kept at reset value.
8	WUPEN	WKUP pin enable 0: Disable WKUP pin function 1: Enable WKUP pin function If WUPEN is set before entering the Standby mode, a rising edge on the WKUP pin will wake up the system from the Standby mode. As the WKUP pin is active high, the WKUP pin is internally configured to input pull down mode. And setting this bit will trigger a wakeup event when the input is already high.
7:6	Reserved	Must be kept at reset value.
5	LVD2OSF	Low voltage detector 2 output signal flag 0: $VCC < V_{det2}$ 1: $VCC \geq V_{det2}$ or $LVD2EN = 0$
4	LVD2DETF	Low voltage detector 2 voltage change detection flag 0: Not detected 1: V_{det2} passage detection
3	LVD1OSF	Low voltage detector 1 output signal flag 0: $VCC < V_{det1}$ 1: $VCC \geq V_{det1}$ or $LVD1EN = 0$
2	LVD1DETF	Low voltage detector 1 voltage change detection flag 0: Not detected 1: V_{det1} passage detection
1	STBF	Standby flag 0: The device has not entered the Standby mode 1: The device has been in the Standby mode This bit is cleared only by a POR/PDR or by setting the STBRST bit in the PMU_CTL

register.

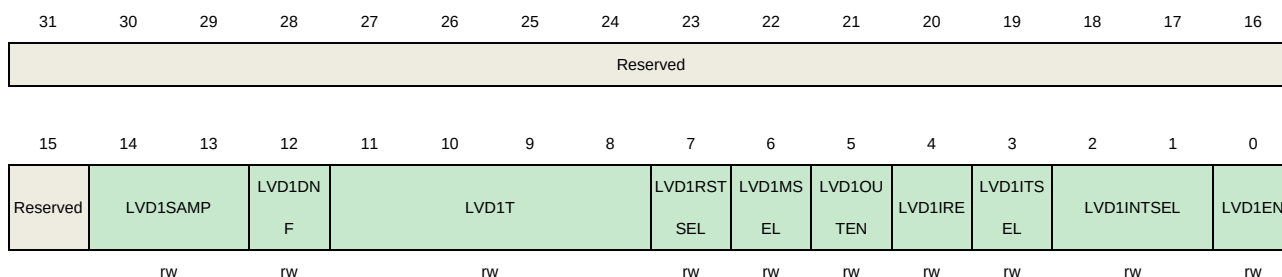
0	WUF	Wakeup flag 0: No wakeup event has been received 1: Wakeup event occurred from the WKUP pin. This bit is reset by the system or cleared by setting the WURST bit in the PMU_CTL register.

3.4.3 Low Voltage Detector 1 Control Register (PMU_LVD1CTL)

Address offset: 0x08

Reset value: 0x0000 1682 (not reset after wakeup from Standby mode)

This register can be accessed by word(32-bit).



Bits	Fields	Descriptions
31:15	Reserved	Must be kept at reset value.
14:13	LVD1SAMP	Sampling clock select 00: 1 / 2 PCLK1 frequency 01: 1 / 4 PCLK1 frequency 10: 1 / 8 PCLK1 frequency 11: 1 / 16 PCLK1 frequency
12	LVD1DNF	Low voltage detector 1 digital noise filter enable 0: Digital noise filter enabled 1: Digital noise filter disabled
11:8	LVD1T[3:0]	Low voltage detector 1 threshold 0100: 4.62V / 4.54V(rising / falling) 0101: 4.55V / 4.47V(rising / falling) 0110: 4.40V / 4.32V(rising / falling) 1010: 3.03V / 2.95V(rising / falling) 1011: 2.96V / 2.88V(rising / falling) 1100-1111: Reserved
7	LVD1RSTSEL	Low voltage detector 1 reset signal negation selection 0: The reset signal will be negated after a stabilization time (tLVD1) following the detection of VCC > Vdet1 (the MCU remains in reset state as long as VCC <=

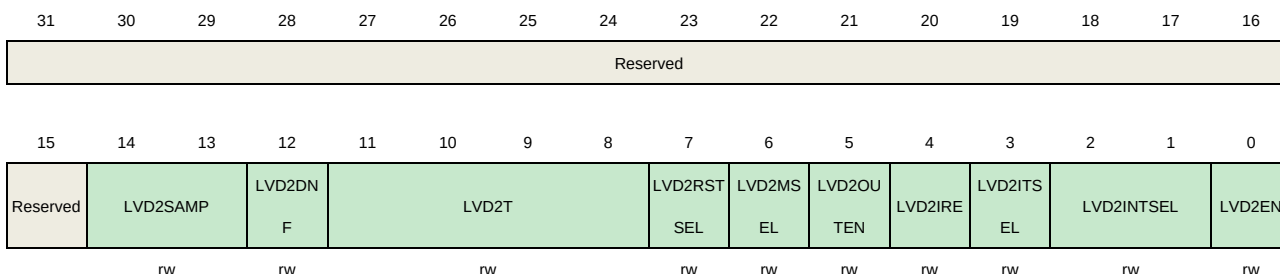
		Vdet1).
		1: After the LVD1 reset signal is asserted, the reset signal will be negated after a stabilization time (tLVD1) (the MCU undergoes a reset once).
6	LVD1MSEL	Low voltage detector 1 Mode Selection 0: Interrupt mode 1: Reset mode
5	LVD1OUTEN	Low voltage detector 1 comparison result output enable 0: Low voltage detector 1 results output disabled 1: Low voltage detector 1 results output enabled
4	LVD1IRE	Low voltage detector 1 Interrupt / Reset Enable 0: LVD1 interrupt / reset disabled 1: LVD1 interrupt / reset enabled
3	LVD1ITSEL	Low voltage detector 1 Interrupt Type Select 0: Non-maskable interrupt 1: Maskable interrupt
2:1	LVD1INTSEL[1:0]	Low voltage detector 1 interrupt trigger mode 00: VCC >= Vdet1 (rising) 01: VCC < Vdet1 (falling) 10: VCC >= Vdet1 or VCC < Vdet1 (rising or falling) 11: Reserved
0	LVD1EN	Low voltage detector 1 Enable 0: Low voltage detector 1 disabled 1: Low voltage detector 1 enabled

3.4.4 Low Voltage Detector 2 Control Register (PMU_LVD2CTL)

Address offset: 0x0C

Reset value: 0x0000 1682 (not reset after wakeup from Standby mode)

This register can be accessed by word(32-bit).



Bits	Fields	Descriptions
31:15	Reserved	Must be kept at reset value.

14:13	LVD2SAMP	Sampling clock select 00: 1 / 2 PCLK1 frequency 01: 1 / 4 PCLK1 frequency 10: 1 / 8 PCLK1 frequency 11: 1 / 16 PCLK1 frequency
12	LVD2DNF	Low voltage detector 2 digital noise filter enable 0: Enable digital noise filter 1: Disable digital noise filter
11:8	LVD2T[3:0]	Low voltage detector 2 threshold 0100: 4.62V / 4.54V(rising / falling) 0101: 4.55V / 4.47V(rising / falling) 0110: 4.40V / 4.32V(rising / falling) 1010: 3.03V / 2.95V(rising / falling) 1011: 2.96V / 2.88V(rising / falling) 1100-1111: Reserved
7	LVD2RSTSEL	Low voltage detector 2 reset signal negation selection 0: The reset signal will be negated after a stabilization time (tLVD2) following the detection of $VCC > V_{det2}$ (the MCU remains in reset state as long as $VCC \leq V_{det2}$). 1: After the LVD2 reset signal is asserted, the reset signal will be negated after a stabilization time (tLVD2) (the MCU undergoes a reset once).
6	LVD2MSEL	Low voltage detector 2 Mode Selection 0: Interrupt mode 1: Reset mode
5	LVD2OUTEN	Low voltage detector 2 comparison result output enabled 0: Low voltage detector 2 results output disabled 1: Low voltage detector 2 results output enabled
4	LVD2IRE	Low voltage detector 2 Interrupt / Reset enable 0: Disable LVD2 interrupt / reset 1: Enable LVD2 interrupt / reset
3	LVD2ITSEL	Low voltage detector 2 Interrupt Type Selection 0: Non-maskable interrupt 1: Maskable interrupt
2:1	LVD2INTSEL[1:0]	Low voltage detector 2 interrupt trigger mode 00: $VCC \geq V_{det2}$ (rising) 01: $VCC < V_{det2}$ (falling) 10: $VCC \geq V_{det2}$ or $VCC < V_{det2}$ (rising or falling) 11: Reserved
0	LVD2EN	Low voltage detector 2 enable

0: Disable Low voltage detector 2

1: Enable Low voltage detector 2

4 Reset and clock unit (RCU)

4.1 Reset control unit (RCTL)

4.1.1 Overview

GD32M531 reset control includes the control of 2 kinds of reset: power reset, system reset. The power on reset, known as a cold reset, resets the full system during a power up. A system reset resets the processor core and peripheral IP components with the exception of the SW-DP controller. The resets can be triggered by an external signal, internal events and the reset generators. More information about these resets will be described in the following sections.

4.1.2 Function overview

Power Reset

The power reset is generated by either an external reset as power on and power down reset (POR/PDR reset), or by the internal reset generator when exiting standby mode. The power reset sets all registers to their reset values. The power reset which active signal is low will be de-asserted when the internal LDO voltage regulator is ready to provide 1.2V power for GD32M531 series. The reset service routine vector is fixed at address 0x0000_0004 in the memory map.

System Reset

A system reset is generated by the following events:

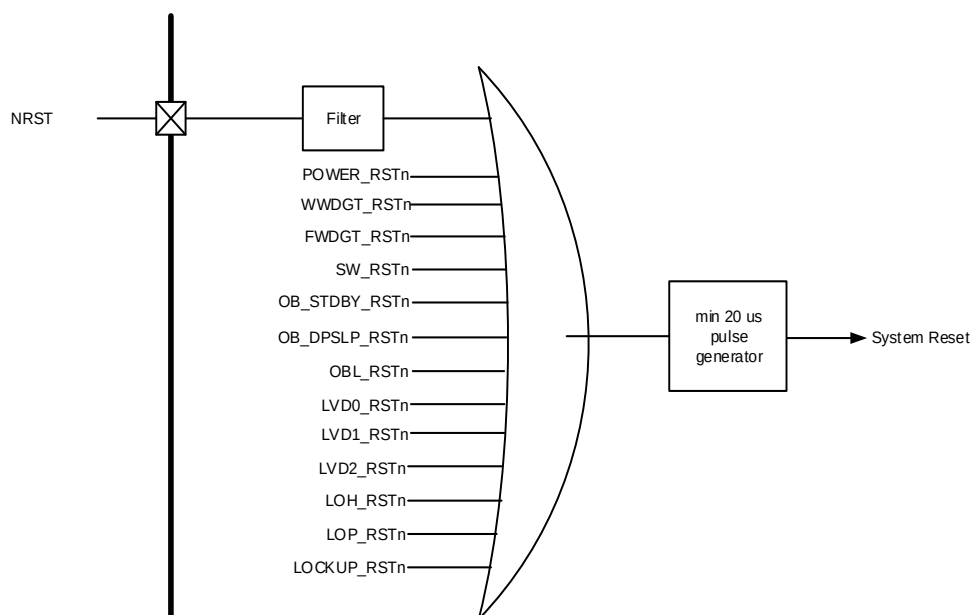
- A power reset (POWER_RSTn)
- An external pin reset (NRST)
- A window watchdog timer reset (WWDGT_RSTn)
- A free watchdog timer reset (FWDGT_RSTn)
- The SYSRESETREQ bit in Cortex®-M33 application interrupt and reset control register is set (SW_RSTn)
- Option byte loader reset (OBL_RSTn)
- Reset generated when entering Standby mode when resetting nRST_STDBY bit in user option bytes (OB_STDBY_RSTn)
- Reset generated when entering Deep-sleep mode when resetting nRST_DPSLP bit in user option bytes (OB_DPSLP_RSTn)
- Low voltage 0 detect reset (LVD0_RSTn)
- Low voltage 1 detect reset (LVD1_RSTn)
- Low voltage 2 detect reset (LVD2_RSTn)
- Loss-of-HXTAL reset (LOH_RSTn)
- Loss-of-PLL reset (LOP_RSTn)
- CPU lockup reset (LOCKUP_RSTn)

LVD_RSTn / LOH_RSTn / LOP_RSTn / LOCKUP_RSTn should be enable by software.

A system reset resets the processor core and peripheral IP components except for the SW-DP controller.

A system reset pulse generator guarantees low level pulse duration of 20 μ s for each reset source (external or internal reset).

Figure 4-1. The system reset circuit



Note: It is conditional for LVD_RSTn / LOH_RSTn / LOP_RSTn / LOCKUP_RSTn to generate a reset. LVD0RSTEN / LVD1RSTEN / LVD2RSTEN in PMU section and LOHRSTEN / LOPRSTEN / LOCKUPRSTEN bit in Reset source/clock register (RCU_RSTSCK) must be set.

4.2 Clock control unit (CCTL)

4.2.1 Overview

The clock control unit provides a range of frequencies and clock functions. These include an Internal 32 MHz RC oscillator (IRC32M), a High speed crystal oscillator (HXTAL), an Internal 32KHz RC oscillator (IRC32K), a Phase Lock Loop (PLL), a HXTAL clock monitor, PLL Clock Monitor (PLLM), clock prescalers, clock multiplexers and clock gating circuitry.

The clocks of the AHB, APB and Cortex®-M33 are derived from the system clock (CK_SYS) which can source from the IRC32M, HXTAL or PLL. The maximum operating frequency of the system clock (CK_SYS) can be up to 180 MHz.

The ADCs are clocked by the clock of AHB clock, the clock formula: $CK\_ADC = CK\_AHB / (ADCPSC + 2)$.

The I2C is clocked by the clock of CK_APB1, CK_PLL, CK_IRC32M or CK_HXTAL which defined by I2CSEL bits in RCU_CFG2. The I2C supports clock switch dynamically.

If the APB prescaler is 1, the timer clock frequencies are set to AHB frequency divide by 1. Otherwise, they are set to the AHB frequency divide by half of APB prescaler.

- 1 to 24 MHz High speed crystal oscillator (HXTAL)
- Internal 32 MHz RC oscillator (IRC32M)

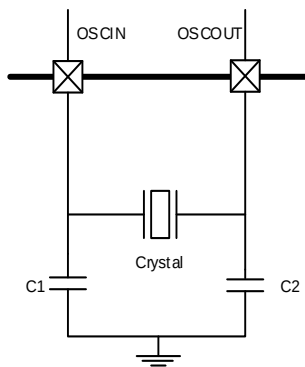
- Internal 32 KHz RC oscillator (IRC32K)
- PLL clock source can be HXTAL or IRC32M
- HXTAL clock monitor

4.2.3 Function overview

High Speed Crystal Oscillator (HXTAL)

The high speed crystal oscillator (HXTAL), which has a frequency from 1 to 24 MHz, produces a highly accurate clock source for use as the system clock. A crystal with a specific frequency must be connected and located close to the two HXTAL pins. The external resistor and capacitor components connected to the crystal are necessary for proper oscillation.

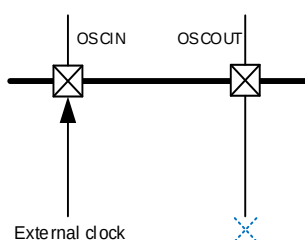
Figure 4-3. HXTAL clock source



The HXTAL crystal oscillator can be switched on or off using the HXTALEN bit in the control register, RCU_CTL. The HXTALSTB flag in control register, RCU_CTL indicates if the high-speed external crystal oscillator is stable. When the HXTAL is powered up, it will not be released for use until this HXTALSTB bit is set by the hardware. This specific delay period is known as the oscillator “Start-up time”. As the HXTAL becomes stable, an interrupt will be generated if the related interrupt enable bit HXTALSTBIE in the interrupt register RCU_INT is set. At this point the HXTAL clock can be used directly as the system clock source or the PLL input clock.

Select external clock bypass mode by setting the HXTALBPS and HXTALEN bits in the Control Register RCU_CTL. During bypass mode, the signal is connected to OSCIN, and OSCOUT remains in the suspended state, as shown in [Figure 4-4. HXTAL clock source in bypass mode](#). The CK_HXTAL is equal to the external clock which drives the OSCIN pin.

Figure 4-4. HXTAL clock source in bypass mode



Select the HXTAL frequency scale by using the HXTALSCAL bit in the control register, RCU_CTL. If HXTAL frequency is higher than 10MHz, HXTALSCAL bit must be set.

Internal 32 MHz RC Oscillator (IRC32M)

The Internal 32 MHz RC oscillator, IRC32M, has a fixed frequency of 32 MHz and is the default clock source selection for the CPU when the device is powered up. The IRC32M oscillator provides a lower cost type clock source as no external components are required. The IRC32M RC oscillator can be switched on or off using the IRC32MEN bit in the control register, RCU_CTL. The IRC32MSTB flag in the control register, RCU_CTL is used to indicate if the internal RC oscillator is stable. The start-up time of the IRC32M oscillator is shorter than the HXTAL crystal oscillator. An interrupt can be generated if the related interrupt enable bit, IRC32MSTBIE, in the interrupt register, RCU_INT, is set when the IRC32M becomes stable. The IRC32M clock can also be used as the PLL input clock.

The frequency accuracy of the IRC32M can be calibrated by the manufacturer, but its operating frequency is still less accurate than HXTAL. The application requirements, environment and cost will determine which oscillator type is selected.

If the HXTAL or PLL is the system clock source, to minimize the time required for the system to recover from the Deep-sleep Mode, the hardware forces the IRC32M clock to be the system clock when the system initially wakes-up.

Phase Locked Loop (PLL)

The internal Phase Locked Loop, PLL, can provide 40~180 MHz clock output which is 4 ~64 multiples of a fundamental reference frequency of 4 ~ 16 MHz.

The PLL can be switched on or off by using the PLEN bit in the control register, RCU_CTL. The PLLSTB flag in the control register, RCU_CTL will indicate if the PLL clock is stable. An interrupt can be generated if the related interrupt enable bit, PLLSTBIE, in the interrupt register, RCU_INT, is set as the PLL becomes stable.

The PLL can be switched on or off by using the PLEN bit in the Control register, RCU_CTL. The PLLSTB flag in the Control register, RCU_CTL will indicate if the PLL clock is stable. An interrupt can be generated if the related interrupt enable bit, PLLSTBIE, in the Interrupt register, RCU_INT, is set as the PLL becomes stable.

Internal 32KHz RC Oscillator (IRC32K)

The Internal 32KHz RC Oscillator has a frequency of about 32kHz and is a low power clock source for the real time clock circuit or the free watchdog timer. The IRC32K offers a low cost clock source as no external components are required. The IRC32K RC oscillator can be switched on or off by using the IRC32KEN bit in the reset source/clock register, RCU_RSTSCK. The IRC32KSTB flag in the reset source/clock register RCU_RSTSCK will indicate if the IRC32K clock is stable. An interrupt can be generated if the related interrupt enable bit IRC32KSTBIE in the Interrupt register RCU_INT is set when the IRC32K becomes stable.

System Clock (CK_SYS) Selection

After the system reset, the default CK_SYS source will be IRC32M and can be switched to HXTAL or PLL by changing the system clock switch bits, SCS, in the configuration register 0, RCU_CFG0. When the SCS value is changed, the CK_SYS will continue to operate using the original clock source until the target clock source is stable. When a clock source is used directly by the CK_SYS or the PLL, it is not possible to stop it.

PLL Clock Monitor (PLLM)

The PLL clock monitor function is enabled by the PLL clock monitor enable bit, PLLMEN, in the control register, RCU_CTL. This function should be enabled after the PLL start-up delay and disabled when the PLL is stopped or slow down. Once the PLL failure is detected, Loss-of-PLL reset or an interrupt will generate. Reset function is decided by LOPRSTEN in the Reset source /clock register, RCU_RSTSCK. Interrupt function is decided by PLLMIE in the interrupt register RCU_INT.

The PLL clock is monitored by IRC32M.

If LOPRSTEN is 0, the PLL lost flag, PLLMIF, in the interrupt register, RCU_INT, will be set and the PLL failure interrupt will be generated. The PLL will be automatically disabled. If the PLL is selected as the clock source of CK_SYS, the PLL failure will force the CK_SYS source to IRC32M and the PLL will be disabled automatically.

If LOPRSTEN is 1, Loss-of-PLL reset will generate.

HXTAL Clock Monitor (CKM)

The HXTAL clock monitor function is enabled by the HXTAL clock monitor enable bit, CKMEN, in the control register, RCU_CTL. This function should be enabled after the HXTAL start-up delay and disabled when the HXTAL is stopped. Once the HXTAL failure is detected, Loss-of-HXTAL reset will generate decided by LOHRSTEN in the reset source /clock register, RCU_RSTSCK. Or an NMI Interrupt may generate decided by the HXTAL_NMIIE bit in SYSCFG_CFG2 register.

If LOHRSTEN is 0, the HXTAL clock stuck flag, CKMIF, in the interrupt register, RCU_INT, will be set and the HXTAL failure event will be generated. The HXTAL will be automatically disabled. This failure interrupt is connected to the Non-Maskable interrupt, NMI, of the Cortex®-M33. If the HXTAL is selected as the clock source of CK_SYS or PLL, the HXTAL failure will force the CK_SYS source to IRC32M and the PLL will be disabled automatically.

If LOHRSTEN is 1, Loss-of-HXTAL reset will generate.

Clock Output Capability

The clock output capability is ranging from 32 kHz to 180 MHz. There are several clock signals can be selected via the CK_OUT clock source selection bits, CKOUTSEL, in the configuration register 0 (RCU_CFG0). The corresponding GPIO pin should be configured in the properly alternate function I/O (AFIO) mode to output the selected clock signal.

Table 4-1. Clock source select

Clock Source Selection bits	Clock Source
000	No Clock
001	Reserved
010	CK_IRC32K
011	Reserved
100	CK_SYS
101	CK_IRC32M
110	CK_HXTAL
111	CK_PLL or CK_PLL/8

The CK_OUT frequency can be reduced by a configurable binary divider, controlled by the CKOUTDIV[2:0] bits, in the configuration register 0(RCU_CFG0).

4.2.4 Operations about event interconnect

Event signal output to EVIC

The clock generation circuit is capable of operation interconnect with another module set in advance when its interrupt request signal is used as an even signal by the event interconnect unit (EVIC) on detection of stopping of the HXTAL clock oscillation.

The clock generation circuit outputs the event signal regardless of the setting of the corresponding HXTAL stop detection interrupt enable bit (RCU_INT.CKMIF). For details, Event interconnect unit (EVIC).

Event signal input from EVIC

The clock generation circuit is capable of switching the clock source to the IRC32M on-chip oscillator in response to the event set in advance when the event specified in the EVIC_RCU register of the EVIC occurs. While this function is in use, clock source switching on return from stop mode cannot be used.

4.3 Register definition

RCU base address: 0x4002 1000

4.3.1 Control register (RCU_CTL)

Address offset: 0x00

Reset value: 0x0000 XX83 where X is undefined.

Note: Set the SYSCFG.PRCFG0 bit to 1 (write enabled) before rewriting this register.

This register can be accessed by byte(8-bit), half-word(16-bit) and word(32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved						PLLSTB	PLLEN	Reserved	HXTALS CAL	PLLMEN	CKMEN	HXTALBP S	HXTALST B	HXTALR ST	HXTALE N
						r	rw		rw	rw	rw	rw	r	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IRC32MCALIB[7:0]								IRC32MADJ[4:0]				Reserved	IRC32MS TB	IRC32ME N	
r								rw					r	rw	

Bits	Fields	Descriptions
31:26	Reserved	Must be kept at reset value.
25	PLLSTB	PLL clock stabilization flag Set by hardware to indicate if the PLL output clock is stable and ready for use. 0: PLL is not stable 1: PLL is stable
24	PLLEN	PLL enable Set and reset by software. This bit cannot be reset if the PLL clock is used as the system clock. Reset by hardware when entering Deep-sleep or Standby mode. 0: PLL is switched off 1: PLL is switched on
23	Reserved	Must be kept at reset value.
22	HXTALSCAL	HXTAL frequency scale select The HXTALSCAL bit can be written only if the HXTALEN is 0. When the HXTAL frequency is equal to 10MHz, this bit must be set to 1. This bit needs to configured only when HXTALBPS bit is 0. 0: HXTAL scale is 1 ~ 10MHz 1: HXTAL scale is 10 ~ 24MHz
21	PLLMEN	PLL clock monitor enable 0: Disable the PLL clock monitor

		1: Enable the PLL clock monitor PLLMEN enable the hardware detects that the PLL clock is stuck at a low/high state. This bit cannot be set to 1, if the PLL is disabled. It reset by hardware when entering Deep-sleep or Standby mode or PLL unlock are detected.
20	CKMEN	HXTAL clock monitor enable 0: Disable the external 1 ~ 24 MHz crystal oscillator (HXTAL) clock monitor 1: Enable the external 1 ~ 24 MHz crystal oscillator (HXTAL) clock monitor When the hardware detects that the HXTAL clock is stuck at a low or high state, the internal hardware will switch the system clock to be the internal high speed IRC32M RC clock. The way to recover the original system clock is by either an external reset, power on reset or clearing CKMIF by software. Note: When the HXTAL clock monitor is enabled, the hardware will automatically enable the IRC32M internal RC oscillator regardless of the control bit, IRC32MEN, state.
19	HXTALBPS	External crystal oscillator (HXTAL) clock bypass mode enable The HXTALBPS bit can be written only if the HXTALEN is 0. 0: Disable the HXTAL bypass mode 1: Enable the HXTAL bypass mode in which the HXTAL output clock is equal to the input clock.
18	HXTALSTB	External crystal oscillator (HXTAL) clock stabilization flag Set by hardware to indicate if the HXTAL oscillator is stable and ready for use. 0: HXTAL oscillator is not stable 1: HXTAL oscillator is stable
17	HXTALRST	External high speed oscillator ready reset Set and reset by software. This bit cannot be reset if the HXTAL clock is used as the system clock or the PLL input clock. Reset by hardware when entering Deep-sleep or Standby mode. 0: No reset 1: Reset HXTAL Note: This bit can be set when HXTAL is not used as system clock.
16	HXTALEN	External high speed oscillator enable Set and reset by software. This bit cannot be reset if the HXTAL clock is used as the system clock or the PLL input clock. Reset by hardware when entering Deep-sleep or Standby mode. 0: External 1 ~ 24 MHz crystal oscillator disabled 1: External 1 ~ 24 MHz crystal oscillator enabled
15:8	IRC32MCALIB[7:0]	Internal 32M RC oscillator calibration value register These bits are load automatically at power on.
7:3	IRC32MADJ[4:0]	Internal 32M RC oscillator clock trim adjust value

These bits are set by software. The trimming value is there bits (IRC32MADJ) added to the IRC32MCALIB[7:0] bits. The trimming value should trim the IRC32M to 32 MHz $\pm$ 1%.

2	Reserved	Must be kept at reset value.
1	IRC32MSTB	IRC32M high speed internal oscillator stabilization flag Set by hardware to indicate if the IRC32M oscillator is stable and ready for use. 0: IRC32M oscillator is not stable 1: IRC32M oscillator is stable
0	IRC32MEN	Internal high speed oscillator enable Set and reset by software. This bit cannot be reset if the IRC32M clock is used as the system clock. Set by hardware when leaving Deep-sleep or Standby mode or the HXTAL clock is stuck at a low or high state when HXTALCKM is set. 0: Internal 32MHz RC oscillator disabled 1: Internal 32MHz RC oscillator enabled

4.3.2 Configuration register 0 (RCU_CFG0)

Address offset: 0x04

Reset value: 0x001E 0400

Note: Set the SYSCFG.PRCFG0 bit to 1 (write enabled) before rewriting this register.

This register can be accessed by byte(8-bit), half-word(16-bit) and word(32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PLLDV	CKOUTDIV[2:0]			Reserved	CKOUTSEL[2:0]			PLLMF[6:0]						PLLSEL	
rw	rw				rw			rw						rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		APB2PSC[2:0]			APB1PSC[2:0]			AHBPSC[3:0]			SCSS[1:0]		SCS[1:0]		
		rw			rw			rw			r		rw		

Bits	Fields	Descriptions
31	PLLDV	The CK_PLL divide by 1 or 8 for CK_OUT 0: CK_PLL divide by 8 for CK_OUT 1: CK_PLL divide by 1 for CK_OUT
30:28	CKOUTDIV[2:0]	The CK_OUT divider which the CK_OUT frequency can be reduced, see bits 26:24 of RCU_CFG0 for CK_OUT. 000: The CK_OUT is divided by 1 001: The CK_OUT is divided by 2 010: The CK_OUT is divided by 4 011: The CK_OUT is divided by 8 100: The CK_OUT is divided by 16 101: The CK_OUT is divided by 32

		110: The CK_OUT is divided by 64 111: The CK_OUT is divided by 128
27	Reserved	Must be kept at reset value.
26:24	CKOUTSEL[2:0]	CK_OUT clock source selection Set and reset by software. 000: No clock selected 001: Reserved 010: Internal 32K RC oscillator clock selected 011: Reserved 100: System clock selected 101: Internal 32MHz RC oscillator clock selected 110: External high speed oscillator clock selected 111: (CK_PLL / 8) or CK_PLL selected depend on PLLDV
23:17	PLLMF[6:0]	PLL multiply factor 0000111: (PLL source clock x 4) 0001000: (PLL source clock x 4.5) 0001001: (PLL source clock x 5) 0001010: (PLL source clock x 5.5) 1111110: (PLL source clock x 63.5) 1111111: (PLL source clock x 64) Note: The PLL output frequency must not exceed 180 MHz.
16	PLLSEL	PLL clock source selection Set and reset by software to control the PLL clock source. 0: HXTAL selected as source clock of PLL 1: IRC32M clock selected as source clock of PLL
15:14	Reserved	Must be kept at reset value
13:11	APB2PSC[2:0]	APB2 prescaler selection Set and reset by software to control the APB2 clock division ratio. 0xx: CK_AHB selected 100: (CK_AHB / 2) selected 101: (CK_AHB / 4) selected 110: (CK_AHB / 8) selected 111: (CK_AHB / 16) selected
10:8	APB1PSC[2:0]	APB1 prescaler selection Set and reset by software to control the APB1 clock division ratio. 0xx: CK_AHB selected 100: (CK_AHB / 2) selected 101: (CK_AHB / 4) selected 110: (CK_AHB / 8) selected

		111: (CK_AHB / 16) selected
7:4	AHBPSC[3:0]	AHB prescaler selection Set and reset by software to control the AHB clock division ratio 0xxx: CK_SYS selected 1000: (CK_SYS / 2) selected 1001: (CK_SYS / 4) selected 1010: (CK_SYS / 8) selected 1011: (CK_SYS / 16) selected 1100: (CK_SYS / 64) selected 1101: (CK_SYS / 128) selected 1110: (CK_SYS / 256) selected 1111: (CK_SYS / 512) selected
3:2	SCSS[1:0]	System clock switch status Set and reset by hardware to indicate the clock source of system clock. 00: Select CK_IRC32M as the CK_SYS source 01: Select CK_HXTAL as the CK_SYS source 10: Select CK_PLL as the CK_SYS source 11: Reserved
1:0	SCS[1:0]	System clock switch Set by software to select the CK_SYS source. Because the change of CK_SYS has inherent latency, software should read SCSS to confirm whether the switching is complete or not. The switch will be forced to IRC32M when leaving Deep-sleep and Standby mode or by HXTAL clock monitor when the HXTAL failure is detected and the HXTAL is selected as the clock source of CK_SYS or PLL. 00: Select CK_IRC32M as the CK_SYS source 01: Select CK_HXTAL as the CK_SYS source 10: Select CK_PLL as the CK_SYS source 11: Reserved

4.3.3 Interrupt register (RCU_INT)

Address offset: 0x08

Reset value: 0x0000 0000

Note: Set the SYSCFG.PRCFG0 bit to 1 (write enabled) before rewriting this register.

This register can be accessed by byte(8-bit), half-word(16-bit) and word(32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								CKMIC	PLLMIC	Reserved	PLL STBIC	HXTAL STBIC	IRC32M STBIC	Reserved	IRC32K STBIC
								w	w		w	w	w		w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	PLLMIE	Reserved	PLL	HXTAL	IRC32M	Reserved	IRC32K	CKMIF	PLLMIF	Reserved	PLL	HXTAL	IRC32M	Reserved	IRC32K

			STBIE	STBIE	STBIE		STBIE				STBIF	STBIF	STBIF		STBIF
	<i>rw</i>		<i>rw</i>	<i>rw</i>	<i>rw</i>		<i>rw</i>	<i>r</i>	<i>r</i>		<i>r</i>	<i>r</i>	<i>r</i>		<i>r</i>

Bits	Fields	Descriptions
31:24	Reserved	Must be kept at reset value
23	CKMIC	HXTAL clock stuck interrupt clear Write 1 by software to reset the CKMIF flag. 0: Not reset CKMIF flag 1: Reset CKMIF flag
22	PLLMIC	PLL clock monitor interrupt clear Write 1 by software to reset the PLLMIF flag. 0: Not reset PLLMIF flag 1: Reset PLLMIF flag
21	Reserved	Must be kept at reset value
20	PLLSTBIC	PLL stabilization interrupt clear Write 1 by software to reset the PLLSTBIF flag. 0: Not reset PLLSTBIF flag 1: Reset PLLSTBIF flag
19	HXTALSTBIC	HXTAL stabilization interrupt clear Write 1 by software to reset the HXTALSTBIF flag. 0: Not reset HXTALSTBIF flag 1: Reset HXTALSTBIF flag
18	IRC32MSTBIC	IRC32M stabilization interrupt clear Write 1 by software to reset the IRC32MSTBIF flag. 0: Not reset IRC32MSTBIF flag 1: Reset IRC32MSTBIF flag
17	Reserved	Must be kept at reset value
16	IRC32KSTBIC	IRC32K stabilization interrupt clear Write 1 by software to reset the IRC32KSTBIF flag. 0: Not reset IRC32KSTBIF flag 1: Reset IRC32KSTBIF flag
15	Reserved	Must be kept at reset value
14	PLLMIE	PLL clock monitor interrupt enable Set and reset by software to enable/disable the PLL clock monitor interrupt. 0: Disable the PLL clock monitor interrupt 1: Enable the PLL clock monitor interrupt
13	Reserved	Must be kept at reset value

12	PLLSTBIE	PLL stabilization interrupt enable Set and reset by software to enable/disable the PLL stabilization interrupt. 0: Disable the PLL stabilization interrupt 1: Enable the PLL stabilization interrupt
11	HXTALSTBIE	HXTAL stabilization interrupt enable Set and reset by software to enable/disable the HXTAL stabilization interrupt 0: Disable the HXTAL stabilization interrupt 1: Enable the HXTAL stabilization interrupt
10	IRC32MSTBIE	IRC32M stabilization interrupt enable Set and reset by software to enable/disable the IRC32M stabilization interrupt 0: Disable the IRC32M stabilization interrupt 1: Enable the IRC32M stabilization interrupt
9	Reserved	Must be kept at reset value
8	IRC32KSTBIE	IRC32K stabilization interrupt enable IRC32K stabilization interrupt enable/disable control 0: Disable the IRC32K stabilization interrupt 1: Enable the IRC32K stabilization interrupt
7	CKMIF	HXTAL clock stuck interrupt flag Set by hardware when the HXTAL clock is stuck. Reset by software when setting the CKMIC bit. Reset by software when setting the CKMNMIIE. 0: Clock operating normally 1: HXTAL clock stuck
6	PLLMIF	PLL clock monitor interrupt flag Set by hardware when PLL clock is stuck. Reset by software when setting the PLLMIC bit. 0: PLL clock operating normally 1: PLL clock stuck
5	Reserved	Must be kept at reset value
4	PLLSTBIF	PLL stabilization interrupt flag Set by hardware when the PLL is stable and the PLLSTBIE bit is set. Reset by software when setting the PLLSTBIC bit. 0: No PLL stabilization interrupt generated 1: PLL stabilization interrupt generated
3	HXTALSTBIF	HXTAL stabilization interrupt flag Set by hardware when the external 1 ~ 24 MHz crystal oscillator clock is stable and the HXTALSTBIE bit is set. Reset by software when setting the HXTALSTBIC bit. 0: No HXTAL stabilization interrupt generated

		1: HXTAL stabilization interrupt generated
2	IRC32MSTBIF	IRC32M stabilization interrupt flag Set by hardware when the internal 32MHz RC oscillator clock is stable and the IRC32MSTBIE bit is set. Reset by software when setting the IRC32MSTBIC bit. 0: No IRC32M stabilization interrupt generated 1: IRC32M stabilization interrupt generated
1	Reserved	Must be kept at reset value
0	IRC32KSTBIF	IRC32K stabilization interrupt flag Set by hardware when the internal 32KHz RC oscillator clock is stable and the IRC32KSTBIE bit is set. Reset by software when setting the IRC32KSTBIC bit. 0: No IRC32K stabilization clock ready interrupt generated 1: IRC32K stabilization interrupt generated

4.3.4 APB2 reset register (RCU_APB2RST)

Address offset: 0x0C

Reset value: 0x0000 0000

Note: Set the SYSCFG.PRCFG1 bit to 1 (write enabled) before rewriting this register.

This register can be accessed by byte(8-bit), half-word(16-bit) and word(32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved	CANRST	Reserved						GPTIMER1RST	GPTIMER0RST	Reserved			TIMER2RST	TIMER1RST	
	rw							rw	rw				rw	rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	TIMER7RST	SPIRST	TIMER0RST	Reserved	ADC2RST	ADC0RST	Reserved						CMRST	CFGRST	
	rw	rw	rw		rw	rw							rw	rw	

Bits	Fields	Descriptions
31	Reserved	Must be kept at reset value
30	CANRST	CAN reset This bit is set and reset by software. 0: No reset 1: Reset the CAN
29:23	Reserved	Must be kept at reset value
22	GPTIMER1RST	GPTIMER1 reset This bit is set and reset by software.

		0: No reset 1: Reset the GPTIMER1
21	GPTIMER0RST	GPTIMER0 reset This bit is set and reset by software. 0: No reset 1: Reset the GPTIMER0
20:18	Reserved	Must be kept at reset value
17	TIMER2RST	TIMER2 reset This bit is set and reset by software. 0: No reset 1: Reset the TIMER2
16	TIMER1RST	TIMER1 reset This bit is set and reset by software. 0: No reset 1: Reset the TIMER1
15:14	Reserved	Must be kept at reset value
13	TIMER7RST	TIMER7 reset This bit is set and reset by software. 0: No reset 1: Reset the TIMER7
12	SPIRST	SPI reset This bit is set and reset by software. 0: No reset 1: Reset the SPI
11	TIMER0RST	TIMER0 reset This bit is set and reset by software. 0: No reset 1: Reset the TIMER0
10	Reserved	Must be kept at reset value
9	ADC2RST	ADC2 reset This bit is set and reset by software. 0: No reset 1: Reset the ADC2
8	ADC0RST	ADC0 reset This bit is set and reset by software. 0: No reset 1: Reset the ADC0

7:2	Reserved	Must be kept at reset value
1	CMPRST	Comparator reset This bit is set and reset by software. 0: No reset 1: Reset comparator
0	CFGRST	System configuration reset This bit is set and reset by software. 0: No reset 1: Reset system configuration

4.3.5 APB1 reset register (RCU_APB1RST)

Address offset: 0x10

Reset value: 0x0000 0000

Note: Set the SYSCFG.PRCFG1 bit to 1 (write enabled) before rewriting this register.

This register can be accessed by byte(8-bit), half-word(16-bit) and word(32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved.	DACRST	PMURST	Reserved	Reserved	Reserved	Reserved	CFMURS	Reserved	I2CRST	Reserved	UART3R	UART2R	UART1R	UART0R	
	rw	rw					T		rw		rw	rw	rw	rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	Reserved	WWDGTR	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	CPTIMER	Reserved	CPTIMER	CPTIMER		
		ST								WRST		1RST	0RST		
										rw		rw	rw	rw	

Bits	Fields	Descriptions
31:30	Reserved	Must be kept at reset value
29	DACRST	DAC reset This bit is set and reset by software. 0: No reset 1: Reset DAC
28	PMURST	Power control reset This bit is set and reset by software. 0: No reset 1: Reset power control unit
27:25	Reserved	Must be kept at reset value
24	CFMURST	CFMU reset This bit is set and reset by software. 0: No reset

		1: Reset CFMU
23:22	Reserved	Must be kept at reset value
21	I2CRST	I2C reset This bit is set and reset by software. 0: No reset 1: Reset I2C
20	Reserved	Must be kept at reset value
19	UART3RST	UART3 reset This bit is set and reset by software. 0: No reset 1: Reset UART3
18	UART2RST	UART2 reset This bit is set and reset by software. 0: No reset 1: Reset UART2
17	UART1RST	UART1 reset This bit is set and reset by software. 0: No reset 1: Reset UART1
16	UART0RST	UART0 reset This bit is set and reset by software. 0: No reset 1: Reset UART0
15:13	Reserved	Must be kept at reset value
12	WWDGTRST	Window watchdog timer reset This bit is set and reset by software. 0: No reset 1: Reset window watchdog timer
11:5	Reserved	Must be kept at reset value
4	CPTIMERWRST	CPTIMERW timer reset This bit is set and reset by software. 0: No reset 1: Reset CPTIMERW timer
3:2	Reserved	Must be kept at reset value
1	CPTIMER1RST	CPTIMER1 timer reset This bit is set and reset by software. 0: No reset

1: Reset CPTIMER1 timer

0 CPTIMER0RST CPTIMER0 timer reset

This bit is set and reset by software.

0: No reset

1: Reset CPTIMER0 timer

4.3.6 AHB enable register (RCU_AHBEN)

Address offset: 0x14

Reset value: 0x0000 0014

Note: Set the SYSCFG.PRCFG1 bit to 1 (write enabled) before rewriting this register. Bit 4 cannot be cleared to 0.

This register can be accessed by byte(8-bit), half-word(16-bit) and word(32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved	TMUEN	SVPWME N	GTOCEN	POCEN	Reserved	PNEN	PGEN	PFEN	PEEN	PDEN	PCEN	PBEN	PAEN		
	rw	rw	rw	rw		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								CRCEN	Reserved	DMAMUX EN	SRAMSP EN	DMA1EN	DMA0EN		
								rw		rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:30	Reserved	Must be kept at reset value
29	TMUEN	TMU clock enable This bit is set and reset by software. 0: Disabled TMU clock 1: Enabled TMU clock
28	SVPWMEN	SVPWM clock enable This bit is set and reset by software. 0: Disabled SVPWM clock 1: Enabled SVPWM clock
27	GTOCEN	GTOC clock enable This bit is set and reset by software. 0: Disabled GTOC clock 1: Enabled GTOC clock
26	POCEN	POC clock enable This bit is set and reset by software. 0: Disabled POC clock

		1: Enabled POC clock
25:24	Reserved	Must be kept at reset value
23	PNEN	GPIO port N clock enable This bit is set and reset by software. 0: Disabled GPIO port N clock 1: Enabled GPIO port N clock
22	PGEN	GPIO port G clock enable This bit is set and reset by software. 0: Disabled GPIO port G clock 1: Enabled GPIO port G clock
21	PFEN	GPIO port F clock enable This bit is set and reset by software. 0: Disabled GPIO port F clock 1: Enabled GPIO port F clock
20	PEEN	GPIO port E clock enable This bit is set and reset by software. 0: Disabled GPIO port E clock 1: Enabled GPIO port E clock
19	PDEN	GPIO port D clock enable This bit is set and reset by software. 0: Disabled GPIO port D clock 1: Enabled GPIO port D clock
18	PCEN	GPIO port C clock enable This bit is set and reset by software. 0: Disabled GPIO port C clock 1: Enabled GPIO port C clock
17	PBEN	GPIO port B clock enable This bit is set and reset by software. 0: Disabled GPIO port B clock 1: Enabled GPIO port B clock
16	PAEN	GPIO port A clock enable This bit is set and reset by software. 0: Disabled GPIO port A clock 1: Enabled GPIO port A clock
15:7	Reserved	Must be kept at reset value
6	CRCEN	CRC clock enable This bit is set and reset by software. 0: Disabled CRC clock

		1: Enabled CRC clock
5:4	Reserved	Must be kept at reset value
3	DMAMUXEN	DMAMUX clock enable This bit is set and reset by software. 0: Disabled DMAMUX clock 1: Enabled DMAMUX clock
2	SRAMSPEN	SRAM interface clock enable This bit is set and reset by software to enable/disable SRAM interface clock during Sleep mode. 0: Disabled SRAM interface clock during sleep mode. 1: Enabled SRAM interface clock during sleep mode
1	DMA1EN	DMA1 clock enable This bit is set and reset by software. 0: Disabled DMA1 clock 1: Enabled DMA1 clock
0	DMA0EN	DMA0 clock enable This bit is set and reset by software. 0: Disabled DMA0 clock 1: Enabled DMA0 clock

4.3.7 APB2 enable register (RCU_APB2EN)

Address offset: 0x18

Reset value: 0x0000 0001

Note: Set the SYSCFG.PRCFG1 bit to 1 (write enabled) before rewriting this register.

This register can be accessed by byte(8-bit), half-word(16-bit) and word(32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
Reserved	CANEN	Reserved						EVICEN	Reserved	GPTIMER1EN	GPTIME R0EN	Reserved			TIMER2E N	TIMER1E N
rw							rw		rw		rw		rw			rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Reserved		TIMER7E N	SPIEN	TIMER0E N	Reserved	ADC2EN	ADC0EN	Reserved						CMPEN		CFGEN
rw			rw	rw	rw		rw								rw	rw

Bits	Fields	Descriptions
31	Reserved	Must be kept at reset value
30	CANEN	CAN clock enable This bit is set and reset by software.

		0: Disabled CAN clock 1: Enabled CAN clock
29:25	Reserved	Must be kept at reset value
24	EVICEN	EVIC clock enable This bit is set and reset by software. 0: Disabled EVIC clock 1: Enabled EVIC clock
23	Reserved	Must be kept at reset value
22	GPTIMER1EN	GPTIMER1 timer clock enable This bit is set and reset by software. 0: Disabled GPTIMER1 timer clock 1: Enabled GPTIMER1 timer clock
21	GPTIMER0EN	GPTIMER0 timer clock enable This bit is set and reset by software. 0: Disabled GPTIMER0 timer clock 1: Enabled GPTIMER0 timer clock
20:18	Reserved	Must be kept at reset value
17	TIMER2EN	TIMER2 timer clock enable This bit is set and reset by software. 0: Disabled TIMER2 timer clock 1: Enabled TIMER2 timer clock
16	TIMER1EN	TIMER1 timer clock enable This bit is set and reset by software. 0: Disabled TIMER1 timer clock 1: Enabled TIMER1 timer clock
15:14	Reserved	Must be kept at reset value
13	TIMER7EN	TIMER7 timer clock enable This bit is set and reset by software. 0: Disabled TIMER7 timer clock 1: Enabled TIMER7 timer clock
12	SPIEN	SPI clock enable This bit is set and reset by software. 0: Disabled SPI clock 1: Enabled SPI clock
11	TIMER0EN	TIMER0 timer clock enable This bit is set and reset by software. 0: Disabled TIMER0 timer clock

		1: Enabled TIMER0 timer clock
10	Reserved	Must be kept at reset value
9	ADC2EN	ADC2 interface clock enable This bit is set and reset by software. 0: Disabled ADC2 interface clock 1: Enabled ADC2 interface clock
8	ADC0EN	ADC interface clock enable This bit is set and reset by software. 0: Disabled ADC0 interface clock 1: Enabled ADC0 interface clock
7:2	Reserved	Must be kept at reset value
1	CMPEN	Comparator clock enable This bit is set and reset by software. 0: Disabled comparator clock 1: Enabled comparator clock
0	CFGEN	System configuration clock enable This bit is set and reset by software. 0: Disabled system configuration clock 1: Enabled system configuration clock

4.3.8 APB1 enable register (RCU_APB1EN)

Address offset: 0x1C

Reset value: 0x0000 0000

Note: Set the SYSCFG.PRCFG1 bit to 1 (write enabled) before rewriting this register.

This register can be accessed by byte(8-bit), half-word(16-bit) and word(32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved	DACEN	PMUEN	Reserved	CFMUEN	Reserved	I2CEN	Reserved	UART3E	UART2E	UART1E	UART0E				
	rw	rw		rw		rw		rw	rw	rw	rw				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	WWDGT	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	CPTIMER	Reserved	CPTIMER	CPTIMER				
	EN							WEN		1EN	0EN				
	rw							rw		rw	rw				

Bits	Fields	Descriptions
31:30	Reserved	Must be kept at reset value
29	DACEN	DAC clock enable This bit is set and reset by software.

		0: Disabled DAC clock 1: Enabled DAC clock
28	PMUEN	Power interface clock enable This bit is set and reset by software. 0: Disabled Power interface clock 1: Enabled Power interface clock
27:25	Reserved	Must be kept at reset value
24	CFMUEN	CFMU clock enable This bit is set and reset by software. 0: Disabled CFMU clock 1: Enabled CFMU clock
23:22	Reserved	Must be kept at reset value
21	I2CEN	I2C clock enable This bit is set and reset by software. 0: Disabled I2C clock 1: Enabled I2C clock
20	Reserved	Must be kept at reset value
19	UART3EN	UART3 clock enable This bit is set and reset by software. 0: Disabled UART3 clock 1: Enabled UART3 clock
18	UART2EN	UART2 clock enable This bit is set and reset by software. 0: Disabled UART2 clock 1: Enabled UART2 clock
17	UART1EN	UART1 clock enable This bit is set and reset by software. 0: Disabled UART1 clock 1: Enabled UART1 clock
16	UART0EN	UART0 clock enable This bit is set and reset by software. 0: Disabled UART0 clock 1: Enabled UART0 clock
15:13	Reserved	Must be kept at reset value
12	WWDGTEN	Window watchdog timer clock enable This bit is set and reset by software. 0: Disabled window watchdog timer clock

		1: Enabled window watchdog timer clock
11:5	Reserved	Must be kept at reset value
4	CPTIMERWEN	CPTIMERW timer clock enable This bit is set and reset by software. 0: Disabled CPTIMERW timer clock 1: Enabled CPTIMERW timer clock
3:2	Reserved	Must be kept at reset value
1	CPTIMER1EN	CPTIMER1 timer clock enable This bit is set and reset by software. 0: Disabled CPTIMER1 timer clock 1: Enabled CPTIMER1 timer clock
0	CPTIMER0EN	CPTIMER0 timer clock enable This bit is set and reset by software. 0: Disabled CPTIMER0 timer clock 1: Enabled CPTIMER0 timer clock

4.3.9 Reset source/clock register (RCU_RSTSCK)

Address offset: 0x24

Reset value: 0x0C80 0000, reset flags reset by power reset only, other reset by system reset.

Note: Set the SYSCFG.PRCFG1 bit to 1 (write enabled) before rewriting this register.

This register can be accessed by byte(8-bit), half-word(16-bit) and word(32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LP RSTF	WWDGT RSTF	FWDGT RSTF	SW RSTF	POR RSTF	EP RSTF	OBL RSTF	RSTFC	V11 RSTF	LOP RSTF	LOH RSTF	LVD2RST F	LVD1RST F	LOCKUP RSTF	Reserved	LVD0RST F
r	r	r	r	r	r	r	rw	r	r	r	r	r	r		r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	LOPRST EN	LOHRST EN	Reserved	LOCKUP RSTEN	Reserved								IRC32K STB	IRC32K EN	
	rw	rw		rw									r	rw	

Bits	Fields	Descriptions
31	LPRSTF	Low-power reset flag Set by hardware when Deep-sleep /standby reset generated. Reset by writing 1 to the RSTFC bit. 0: No Low-power management reset generated 1: Low-power management reset generated
30	WWDGTRSTF	Window watchdog timer reset flag Set by hardware when a window watchdog timer reset generated.

		Reset by writing 1 to the RSTFC bit. 0: No window watchdog reset generated 1: Window watchdog reset generated
29	FWDGTRSTF	Free Watchdog timer reset flag Set by hardware when a Free Watchdog timer generated. Reset by writing 1 to the RSTFC bit. 0: No Free Watchdog timer reset generated 1: Free Watchdog timer reset generated
28	SWRSTF	Software reset flag Set by hardware when a software reset generated. Reset by writing 1 to the RSTFC bit. 0: No software reset generated 1: Software reset generated
27	PORRSTF	Power reset flag Set by hardware when a power reset generated. Reset by writing 1 to the RSTFC bit. 0: No power reset generated 1: Power reset generated
26	EPRSTF	External PIN reset flag Set by hardware when an External PIN generated. Reset by writing 1 to the RSTFC bit. 0: No external PIN reset generated 1: External PIN reset generated
25	OBLRSTF	Option byte loader reset flag Set by hardware when an option byte loader generated. Reset by writing 1 to the RSTFC bit. 0: No option byte loader reset generated 1: Option byte loader reset generated
24	RSTFC	Reset flag clear This bit is set by software to clear all reset flags. 0: Not clear reset flags 1: Clear reset flags
23	V11RSTF	1.1V domain power reset flag Set by hardware when a 1.1V domain Power reset generated. Reset by writing 1 to the RSTFC bit. 0: No 1.1V domain power reset generated 1: 1.1V domain power reset generated
22	LOPRSTF	Lost of PLL error reset flag Set by hardware when a Lost of PLL Error reset generated.

		Reset by writing 1 to the RSTFC bit. 0: No lost of PLL error reset generated 1: Lost of PLL error reset generated
21	LOHRSTF	Lost of HXTAL error reset flag Set by hardware when a Lost of HXTAL Error reset generated. Reset by writing 1 to the RSTFC bit. 0: No lost of HXTAL error reset generated 1: Lost of HXTAL error reset generated
20	LVD2RSTF	Low voltage 2 detect error reset flag Set by hardware when a low voltage 2 detect error reset generated. Reset by writing 1 to the RSTFC bit. 0: No low voltage detect error reset generated 1: Low voltage detect error reset generated
19	LVD1RSTF	Low voltage 1 detect error reset flag Set by hardware when a low voltage 1 detect error reset generated. Reset by writing 1 to the RSTFC bit. 0: No low voltage detect error reset generated 1: Low voltage detect error reset generated
18	LOCKUPRSTF	CPU Lock-Up error reset flag Set by hardware when a CPU Lock-Up error reset generated. Reset by writing 1 to the RSTFC bit. 0: No CPU Lock-Up error reset generated 1: CPU Lock-Up error reset generated
17	Reserved	Must be kept at reset value
16	LVD0RSTF	Low voltage 0 detect error reset flag Set by hardware when a low voltage 0 detect error reset generated. Reset by writing 1 to the RSTFC bit. 0: No low voltage detect error reset generated 1: Low voltage detect error reset generated
15	Reserved	Must be kept at reset value
14	LOPRSTEN	Lost of PLL reset enable 0: PLL monitor generates interrupt when error detected, if PLLMIE is 1 1: PLL monitor generates reset when error detected
13	LOHRSTEN	Lost of HXTAL reset enable 0: No reset generated 1: HXTAL Monitor generates reset when error detected
12:11	Reserved	Must be kept at reset value
10	LOCKUPRSTEN	CPU Lock-Up reset enable

4.3.10 AHB reset register (RCU_AHBRST)

Reset value: 0x0000 0000

Note: Set the SYSCFG.PRCFG1 bit to 1 (write enabled) before rewriting this register.

This register can be accessed by byte(8-bit), half-word(16-bit) and word(32-bit)

Bits	Fields	Descriptions
31:30	Reserved	Must be kept at reset value
29	TMURST	TMU reset This bit is set and reset by software. 0: No reset TMU module 1: Reset TMU module
28	SVPWMRST	SVPWM reset This bit is set and reset by software. 0: No reset SVPWM module 1: Reset SVPWM module
27	GTOCRST	GTOC reset This bit is set and reset by software.

		0: No reset GTOC module 1: Reset GTOC module
26	POCRST	POC reset This bit is set and reset by software. 0: No reset POC module 1: Reset POC module
25:24	Reserved	Must be kept at reset value
23	PNRST	GPIO port N reset This bit is set and reset by software. 0: No reset GPIO port N 1: Reset GPIO port N
22	PGRST	GPIO port G reset This bit is set and reset by software. 0: No reset GPIO port G 1: Reset GPIO port G
21	PFRST	GPIO port F reset This bit is set and reset by software. 0: No reset GPIO port F 1: Reset GPIO port F
20	PERST	GPIO port E reset This bit is set and reset by software. 0: No reset GPIO port E 1: Reset GPIO port E
19	PDRST	GPIO port D reset This bit is set and reset by software. 0: No reset GPIO port D 1: Reset GPIO port D
18	PCRST	GPIO port C reset This bit is set and reset by software. 0: No reset GPIO port C 1: Reset GPIO port C
17	PBRST	GPIO port B reset This bit is set and reset by software. 0: No reset GPIO port B 1: Reset GPIO port B
16	PARST	GPIO port A reset This bit is set and reset by software. 0: No reset GPIO port A

		1: Reset GPIO port A
15:7	Reserved	Must be kept at reset value.
6	CRCRST	CRC reset This bit is set and reset by software. 0: No reset CRC module 1: Reset CRC module
5:4	Reserved	Must be kept at reset value
3	DMAMUXRST	DMAMUX reset This bit is set and reset by software. 0: No reset DMAMUX module 1: Reset DMAMUX module
2	Reserved	Must be kept at reset value
1	DMA1RST	DMA1 reset This bit is set and reset by software. 0: No reset DMA1 module 1: Reset DMA1 module
0	DMA0RST	DMA0 reset This bit is set and reset by software. 0: No reset DMA0 module 1: Reset DMA0 module

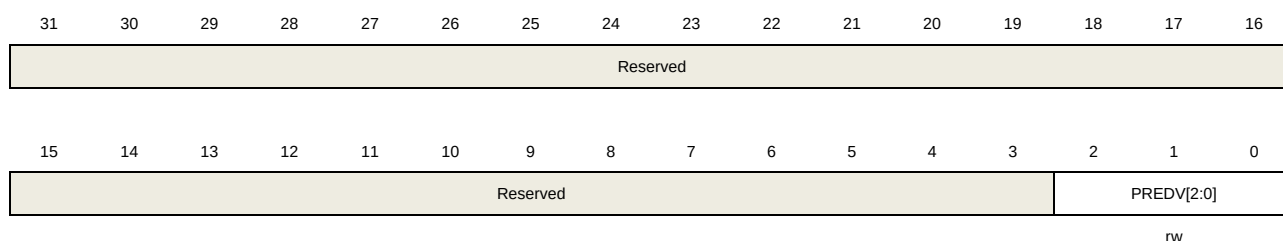
4.3.11 Configuration register 1 (RCU_CFG1)

Address offset: 0x2C

Reset value: 0x0000 0000

Note: Set the SYSCFG.PRCFG0 bit to 1 (write enabled) before rewriting this register.

This register can be accessed by byte(8-bit), half-word(16-bit) and word(32-bit)



Bits	Fields	Descriptions
31:3	Reserved	Must be kept at reset value
2:0	PREDV[2:0]	CK_HXTAL divider previous PLL This bit is set and reset by software. These bits can be written when PLL is disabled.

The CK_HXTAL is divided by (PREDV + 1).

000: Input to PLL not divided

001: Input to PLL divided by 2

010: Input to PLL divided by 3

011: Input to PLL divided by 4

100: Input to PLL divided by 5

101: Input to PLL divided by 6

110: Input to PLL divided by 7

111: Input to PLL divided by 8

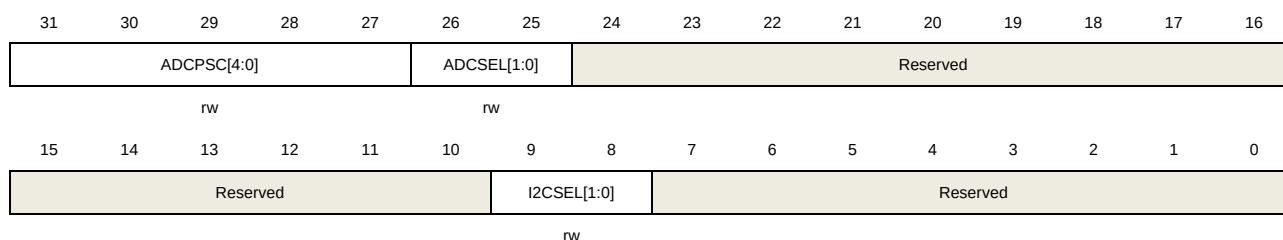
4.3.12 Configuration register 2 (RCU_CFG2)

Address offset: 0x30

Reset value: 0x1800 0000

Note: Set the SYSCFG.PRCFG0 bit to 1 (write enabled) before rewriting this register.

This register can be accessed by byte(8-bit), half-word(16-bit) and word(32-bit)



Bits	Fields	Descriptions
31:27	ADCPSC[4:0]	ADC clock prescaler selection These bits are set and cleared by software. Note: These bits cannot be set to "11111". $CK\_ADC = CK\_ADCPRE / (ADCPSC + 2)$
26:25	ADCSEL[1:0]	CK_ADCPRE clock source selection This bit is set and reset by software. 00: CK_ADCPRE select CK_PCLK2 01: CK_ADCPRE select CK_HXTAL 10: CK_ADCPRE select CK_PLL 11: CK_ADCPRE select CK_IRC32M
24:10	Reserved	Must be kept at reset value
9:8	I2CSEL[1:0]	CK_I2C clock source selection This bit is set and reset by software. 00: CK_I2C select CK_PCLK1 01: CK_I2C select CK_PLL 10: CK_I2C select CK_IRC32M

11: CK_I2C select CK_HXTAL

7:0

Reserved

Must be kept at reset value

5 Clock frequency measurement unit (CFMU)

5.1 Overview

The clock frequency measurement unit provides a clock frequency accuracy measurement function. Count the pulses of the measured clock using the reference clock as the time base, and then determine whether the accuracy of the clock to be measured is within a allowable range through the count result. An interrupt request will be generated when the clock frequency accuracy measurement is completed or the count number of pulses is not within the allowable range.

5.2 Characteristics

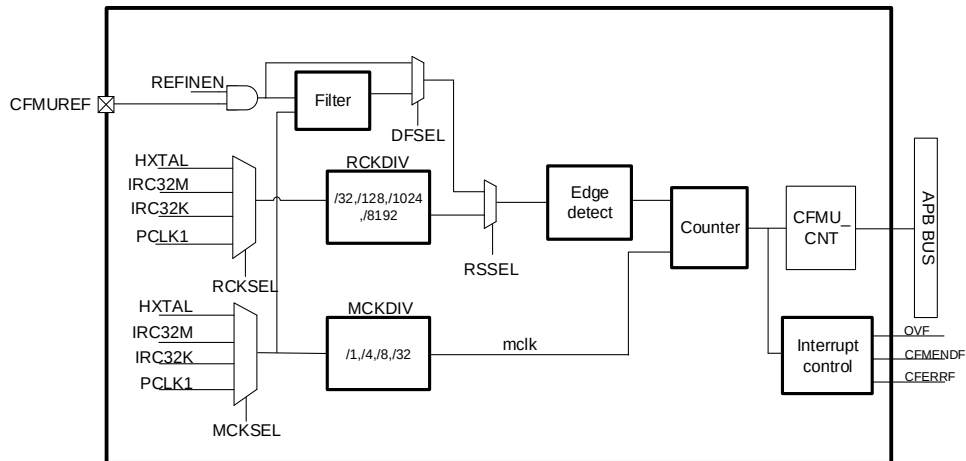
- Clock frequency measurement for different target clocks: HXTAL, IRC32M, IRC32K or PCLK1.
- Selectable reference clocks: external clock input to CFMUREF pin, HXTAL, IRC32M, IRC32K or PCLK1.
- Digital filter function on CFMUREF pin.
- Interrupts and flags to indicate the clock frequency measurement status: clock measurement end, clock frequency error or overflow.

5.3 Function overview

5.3.1 CFMU block diagram

The block diagram of CFMU is shown in [Figure 5-1. Block diagram of CFMU](#). Two clock selector are used for the measurement target and reference clock respectively and there are two corresponding clock dividers. Edge detection function can be configured by EDGSEL bits in CFMU_CTL register. Different interrupt requests can be generated by CFMU in interrupt control unit.

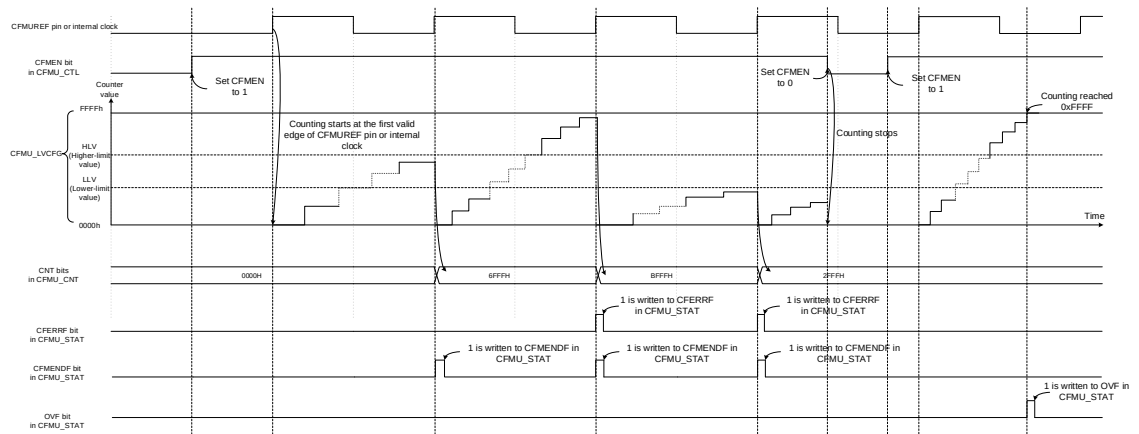
Figure 5-1. Block diagram of CFMU



5.3.2 Clock frequency measurement

The input clock frequency using CFMUREF pin or internal clock (configured by RCKSEL bits and RSSEL bit in CFMU_CTL register) can be measured by clock frequency accuracy measurement unit. [Figure 5-2. Clock frequency accuracy measurement](#) shows the operations of clock frequency accuracy measurement unit.

Figure 5-2. Clock frequency accuracy measurement



Note: When enabling the CFMU for the first time, it is possible that a complete reference clock cycle is not captured. Therefore, it is necessary to disregard the initial CFMU_CNT value and the clock frequency error flag.

5.3.3 Digital filter function on CFMUREF pin

There is a digital filter on CFMUREF pin and this function can be configured by DFSEL bits in CFMU_CTL register.

5.3.4 CFMU interrupts

The CFMU generates three types of interrupt request.

- Frequency error interrupt
- Measurement end interrupt
- Overflow interrupt

When an interrupt source is generated, the corresponding status flag becomes 1. Refer to the [Table 5-1. CFMU interrupt requests](#) for details.

Table 5-1. CFMU interrupt requests

Flag	Description	Clear method	Interrupt enable bit
CFERRF	Clock frequency error flag	Set CFERRCLR bit in CFMU_CTL	CFERRIE
CFMENDF	Clock frequency measurement end flag	Set CFMENDFCLR bit in CFMU_CTL	CFMENDIE
OVF	Counter overflow flag	Set OVFCCLR bit in CFMU_CTL	OVIE

5.4 Register definition

CFMU base address: 0x4000 C800

5.4.1 Control register (CFMU_CTL)

Address offset: 0x00

Reset value: 0x0000 0000

This register can be accessed by word(32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved	OVFCLR	CFMEND FCLR	CFERRC LR	Reserved	OVIE	CFMENDI E	CFERRIE	DFSEL[1:0]	RCKDIV[1:0]	MCKSEL[2:0]	RSSEL				
	rw	rw	rw		rw	rw	rw	rw	rw	rw		rw		rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EDGSEL[1:0]	MCKDIV[1:0]	RCKSEL[2:0]	REFINEN	Reserved											
rw	rw	rw	rw												
															rw

Bits	Fields	Descriptions
31	Reserved	Must be kept at reset value.
30	OVFCLR	Overflow flag clear When 1 is written to this bit, the OVF flag is cleared. This bit is read as 0.
29	CFMENDFCLR	Clock frequency measurement end flag clear When 1 is written to this bit, the CFMENDF flag is cleared. This bit is read as 0.
28	CFERRCLR	Clock frequency error flag clear When 1 is written to this bit, the CFERR flag is cleared. This bit is read as 0.
27	Reserved	Must be kept at reset value.
26	OVIE	Overflow interrupt enable 0: Interrupt disabled 1: Interrupt enabled
25	CFMENDIE	Clock frequency measurement end interrupt enable 0: Interrupt disabled 1: Interrupt enabled
24	CFERRIE	Clock frequency error interrupt enable 0: Interrupt disabled 1: Interrupt enabled
23:22	DFSEL[1:0]	Digital filter select These bits configure the sampling clock for the digital filter. 00: Digital filtering disabled

		01: Select the frequency measuring clock divided by 1 10: Select the frequency measuring clock divided by 4 11: Select the frequency measuring clock divided by 16
21:20	RCKDIV[1:0]	Reference clock frequency divider These bits configure the reference clock frequency division ration. 00: Clock frequency divided by 32 01: Clock frequency divided by 128 10: Clock frequency divided by 1024 11: Clock frequency divided by 8192
19:17	MCKSEL[2:0]	Measurement clock select 000: External HXTAL oscillator clock selected 010: Internal 32MHz RC oscillator clock selected 011: Internal 32K RC oscillator clock selected 100: Peripheral module clock 1 (PCLK1) selected Others: Reserved
16	RSSEL	Reference signal select 0: Select CFMUREF pin input 1: Select internal clock
15:14	EDGSEL[1:0]	Edge select 00: Rising edge selected 01: Falling edge selected 10: Both rising and falling edges selected 11: Reserved
13:12	MCKDIV[1:0]	Measurement clock frequency divider These bits configure the measurement clock frequency division ration. 00: Clock frequency divided by 1 01: Clock frequency divided by 4 10: Clock frequency divided by 8 11: Clock frequency divided by 32
11:9	RCKSEL[2:0]	Reference clock select 000: External HXTAL oscillator clock selected 010: Internal 32MHz RC oscillator clock selected 011: Internal 32K RC oscillator clock selected 100: Peripheral module clock 1 (PCLK1) selected Others: Reserved
8	REFINEN	CFMUREF pin input enable 0: CFMUREF pin input is disabled 1: CFMUREF pin input is enabled
7:1	Reserved	Must be kept at reset value.

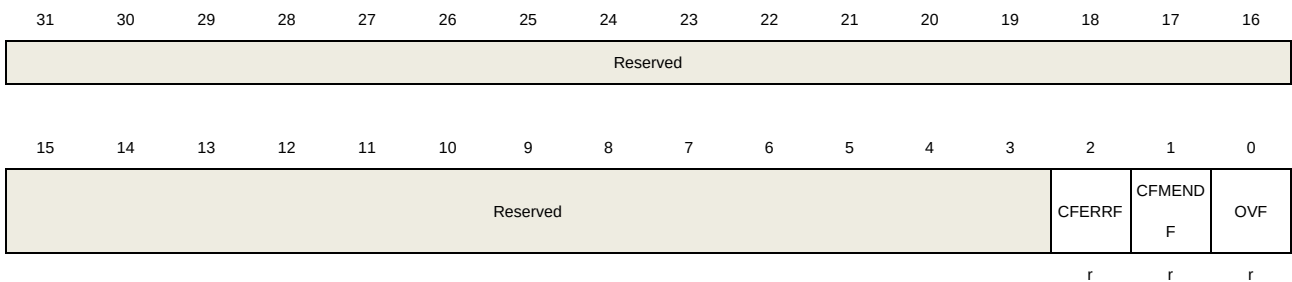
0	CFMEN	Clock frequency measurement enable 0: Clock frequency measurement is disabled 1: Clock frequency measurement is enabled

5.4.2 Status register (CFMU_STAT)

Address offset: 0x04

Reset value: 0x0002 0000

This register can be accessed by word(32-bit)



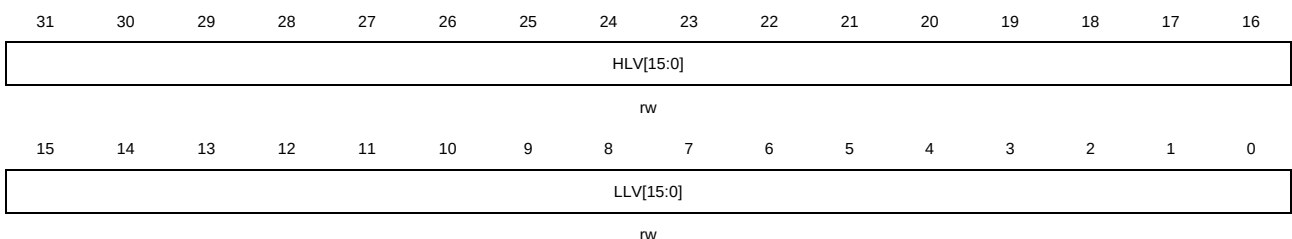
Bits	Fields	Descriptions
31:3	Reserved	Must be kept at reset value.
2	CFERRF	Clock frequency error flag 0: The clock frequency is within the set range 1: The clock frequency is not within the set range
1	CFMENDF	Clock frequency measurement end flag 0: Clock frequency measurement is in progress 1: Clock frequency measurement has ended
0	OVF	Counter overflow flag 0: Counter has not overflowed 1: Counter has overflowed

5.4.3 Limit value configuration register (CFMU_LVCFG)

Address offset: 0x08

Reset value: 0x0000 0000

This register can be accessed by word(32-bit)



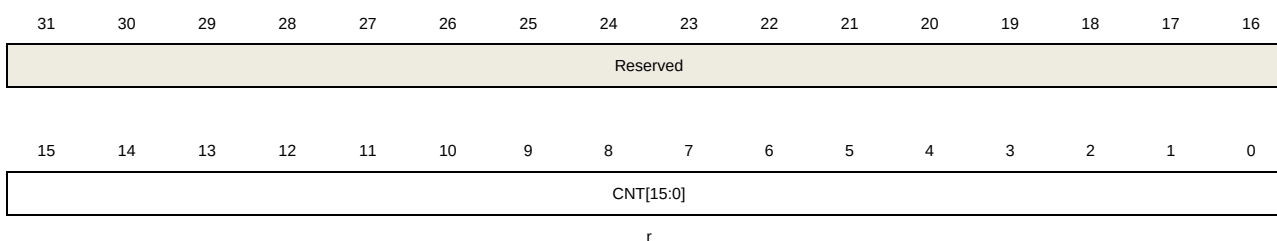
Bits	Fields	Descriptions
31:16	HLV[15:0]	Higher-limit value A frequency error is detected when the clock frequency is greater than this value. These bits can be written when the CFMEN bit is 0.
15:0	LLV[15:0]	Lower-limit value A frequency error is detected when the clock frequency is lower than this value. These bits can be written when the CFMEN bit is 0.

5.4.4 Counter value register (CFMU_CNT)

Address offset: 0x0C

Reset value: 0x0000 0000

This register can be accessed by word(32-bit)



Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CNT[15:0]	Counter value This bit-field indicates the current counter value. It is updated after the CFMENDF bit is set.

6 Interrupt/event controller (EXTI)

6.1 Overview

Cortex®-M33 integrates the Nested Vectored Interrupt Controller (NVIC) for efficient exception and interrupts processing. NVIC facilitates low-latency exception and interrupt handling and power management controls. It's tightly coupled to the processor core. You can read the technical reference manual of Cortex®-M33 for more details about NVIC.

EXTI (interrupt / event controller) contains up to 25 independent edge detectors and generates interrupt requests or events to the processor. The EXTI has three trigger types: rising edge, falling edge and both edges. Each edge detector in the EXTI can be configured and masked independently.

6.2 Characteristics

- Cortex®-M33 system exception.
- Up to 93 maskable peripheral interrupts.
- 4 bits interrupt priority configuration—16 priority levels.
- Efficient interrupt processing.
- Support exception pre-emption and tail-chaining.
- Wake up system from power saving mode.
- Up to 25 independent edge detectors in EXTI.
- Three trigger types: rising, falling and both edges.
- Software interrupt or event trigger.
- Trigger sources configurable.

6.3 Interrupts function overview

The Arm® Cortex®-M33 processor and the Nested Vectored Interrupt Controller (NVIC) prioritize and handle all exceptions in Handler Mode. The processor state is automatically stored to the stack on an exception and automatically restored from the stack at the end of the Interrupt Service Routine (ISR).

The vector is fetched in parallel to the state saving, enabling efficient interrupt entry. The processor supports tail-chaining, which enables back-to-back interrupts to be performed without the overhead of state saving and restoration. The [Table 6-1. NVIC exception types in Cortex®-M33](#) and [Table 6-2. Interrupt vector table](#) list all exception types.

Table 6-1. NVIC exception types in Cortex®-M33

Exception type	Vector number	priority (a)	Vector address	Description
-	0	-	0x0000_0000	Reserved
Reset	1	-3	0x0000_0004	Reset
NMI	2	-2	0x0000_0008	Non maskable interrupt.
HardFault	3	-1	0x0000_000C	All class of fault
MemManage	4	Programmable	0x0000_0010	Memory management
BusFault	5	Programmable	0x0000_0014	Prefetch fault, memory access fault
UsageFault	6	Programmable	0x0000_0018	Undefined instruction or illegal state
-	7-10	-	0x0000_001C - 0x0000_002B	Reserved
SVCall	11	Programmable	0x0000_002C	System service call via SWI instruction
Debug Monitor	12	Programmable	0x0000_0030	Debug Monitor
-	13	-	0x0000_0034	Reserved
PendSV	14	Programmable	0x0000_0038	Pendable request for system service
SysTick	15	Programmable	0x0000_003C	System tick timer

Table 6-2. Interrupt vector table

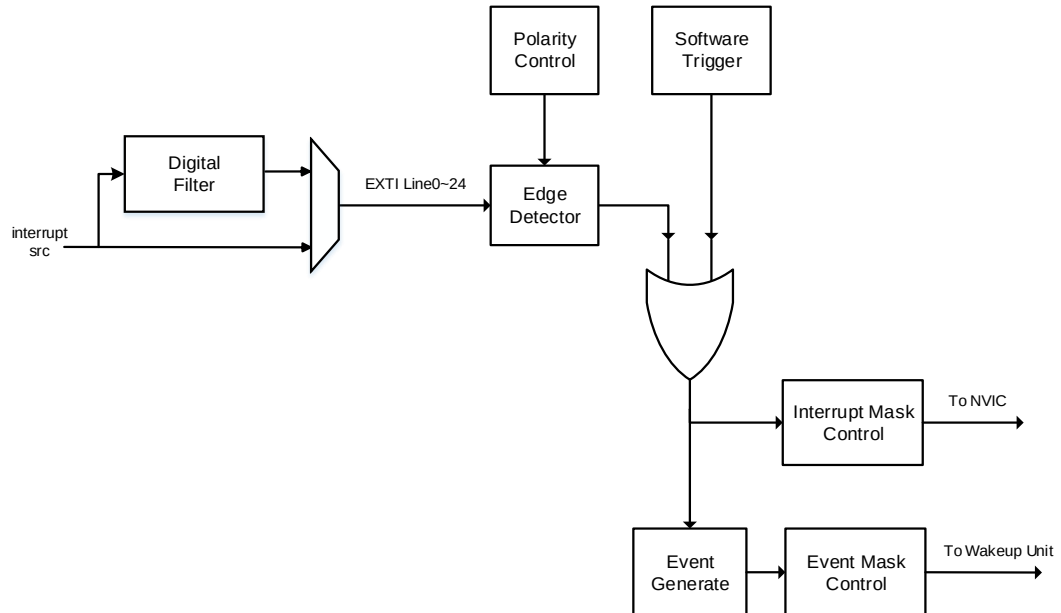
Interrupt number	Vector number	Peripheral interrupt description	Vector address
IRQ 0	16	WWDGT interrupt	0x0000_0040
IRQ 1	17	LVD1 from EXTI interrupt	0x0000_0044
IRQ 2	18	LVD2 from EXTI interrupt	0x0000_0048
IRQ 3	19	reserved	0x0000_004C
IRQ 4	20	FMC global interrupt	0x0000_0050
IRQ 5	21	RCU interrupt	0x0000_0054
IRQ 6	22	EXTI Line0 interrupt	0x0000_0058
IRQ 7	23	EXTI Line1 interrupt	0x0000_005C
IRQ 8	24	EXTI Line2 interrupt	0x0000_0060
IRQ 9	25	EXTI Line3 interrupt	0x0000_0064
IRQ 10	26	EXTI Line4 interrupt	0x0000_0068
IRQ 11	27	DMA0 channel0 global interrupt	0x0000_006C
IRQ 12	28	DMA0 channel1 global interrupt	0x0000_0070
IRQ 13	29	DMA0 channel2 global interrupt	0x0000_0074
IRQ 14	30	DMA0 channel3 global interrupt	0x0000_0078
IRQ 15	31	DMA0 channel4 global interrupt	0x0000_007C
IRQ 16	32	DMA0 channel5 global interrupt	0x0000_0080
IRQ 17	33	reserved	0x0000_0084

Interrupt number	Vector number	Peripheral interrupt description	Vector address
IRQ 18	34	ADC0 global interrupt	0x0000_0088
IRQ 19	35	CAN TX interrupt	0x0000_008C
IRQ 20	36	CAN RX0 interrupt	0x0000_0090
IRQ 21	37	CAN RX1 interrupt	0x0000_0094
IRQ 22	38	CAN EWMC interrupt	0x0000_0098
IRQ 23	39	EXTI line[9:5] interrupts	0x0000_009C
IRQ 24	40	TIMER0 break interrupt	0x0000_00A0
IRQ 25	41	TIMER0 update interrupt	0x0000_00A4
IRQ 26	42	TIMER0 trigger and Channel commutation interrupts	0x0000_00A8
IRQ 27	43	TIMER0 channel capture compare interrupt	0x0000_00AC
IRQ 28	44	TIMER1 global interrupt	0x0000_00B0
IRQ 29	45	TIMER2 global interrupt	0x0000_00B4
IRQ 30	46	GPTIMER0 global interrupt	0x0000_00B8
IRQ 31	47	I2C event interrupt	0x0000_00BC
IRQ 32	48	I2C error interrupt	0x0000_00C0
IRQ 33-34	49-50	reserved	0x0000_00C4- 0x0000_00C8
IRQ 35	51	SPI global interrupt	0x0000_00CC
IRQ 36	52	reserved	0x0000_00D0
IRQ 37	53	UART0 global interrupt	0x0000_00D4
IRQ 38	54	UART1 global interrupt	0x0000_00D8
IRQ 39	55	reserved	0x0000_00DC
IRQ 40	56	EXTI line[15:10] interrupts	0x0000_00E0
IRQ 41-42	57-58	reserved	0x0000_00E4- 0x0000_00E8
IRQ 43	59	TIMER7 break interrupt	0x0000_00EC
IRQ 44	60	TIMER7 update interrupt	0x0000_00F0
IRQ 45	61	TIMER7 trigger and Channel commutation interrupts	0x0000_00F4
IRQ 46	62	TIMER7 channel capture compare interrupt	0x0000_00F8
IRQ 47	63	TMU global interrupt	0x0000_00FC
IRQ 48-49	64-65	reserved	0x0000_0100- 0x0000_0104
IRQ 50	66	GPTIMER1 global interrupt	0x0000_0108
IRQ 51	67	reserved	0x0000_010C
IRQ 52	68	UART2 global interrupt	0x0000_0110
IRQ 53	69	UART3 global interrupt	0x0000_0114

Interrupt number	Vector number	Peripheral interrupt description	Vector address
IRQ 54	70	CPTIMER0 global interrupt	0x0000_0118
IRQ 55	71	CPTIMER1 global interrupt	0x0000_011C
IRQ 56	72	DMA1 channel0 global interrupt	0x0000_0120
IRQ 57	73	DMA1 channel1 global interrupt	0x0000_0124
IRQ 58	74	DMA1 channel2 global interrupt	0x0000_0128
IRQ 59	75	DMA1 channel3 global interrupt	0x0000_012C
IRQ 60	76	DMA1 channel4 global interrupt	0x0000_0130
IRQ 61	77	DMA1 channel5 global interrupt	0x0000_0134
IRQ 62	78	DMAMUX overrun interrupt	0x0000_0138
IRQ 63	79	CPTIMERW global interrupt	0x0000_013C
IRQ 64	80	reserved	0x0000_0140
IRQ 65	81	CFMU interrupt	0x0000_0144
IRQ 66	82	I2C wakeup from EXTI interrupt	0x0000_0148
IRQ 67	83	FWDGT from EXTI interrupt	0x0000_014C
IRQ 68-69	84-85	reserved	0x0000_0150- 0x0000_0154
IRQ 70	86	CMP0 interrupt	0x0000_0158
IRQ 71	87	CMP1 interrupt	0x0000_015C
IRQ 72	88	CMP2 interrupt	0x0000_0160
IRQ 73	89	CMP3 interrupt	0x0000_0164
IRQ 74	90	reserved	0x0000_0168
IRQ 75	91	ADC2 global interrupt	0x0000_016C
IRQ 76	92	reserved	0x0000_0170
IRQ 77	93	POC global interrupt	0x0000_0174
IRQ 79	95	GTOC0 global interrupt	0x0000_017C
IRQ 80	96	GTOC1 global interrupt	0x0000_0180
IRQ 81	97	GTOC2 global interrupt	0x0000_0184
IRQ 82	98	GTOC3 global interrupt	0x0000_0188
IRQ 83-84	99-100	reserved	0x0000_018C- 0x0000_0190
IRQ 85	101	CMP0 from EXTI interrupt	0x0000_0194
IRQ 86	102	CMP1 from EXTI interrupt	0x0000_0198
IRQ 87	103	CMP2 from EXTI interrupt	0x0000_019C
IRQ 88	104	CMP3 from EXTI interrupt	0x0000_01A0
IRQ 89-91	105-107	reserved	0x0000_01A4- 0x0000_01AC
IRQ 92	108	SRAM ECC global interrupt	0x0000_01B0

6.4 External interrupt and event (EXTI) block diagram

Figure 6-1. Block diagram of EXTI



GD32M531 EXTI has a digital filter can reduce noise by filtering the EXTI external source (signals from GPIO pins and NMI pin. That is EXTI line 0 – 15, and line 24). Configure the EXTI_DFEN register to enable or disable the digital filter. After the digital filter is enable, only if the digital filter continuously samples three same level from EXTI external sources, the EXTI trigger signals is considered as effective. The sampling clock of the digital filter can be configured through the EXTI_SCS register.

Note: When the system works in the deep sleep mode or standby mode, the digital filter cannot be used.

6.5 External Interrupt and event function overview

The EXTI contains up to 25 independent edge detectors and generates interrupts request or event to the processor. The EXTI has three trigger types: rising edge, falling edge and both edges. Each edge detector in the EXTI can be configured and masked independently.

The EXTI trigger source includes 16 external lines from GPIO pins and 9 lines from internal modules which refers to [Table 6-3. EXTI source](#) for detail. All GPIO pins can be selected as an EXTI trigger source by configuring SYSCFG_EXTISSx registers (please refer to [System configuration registers](#) section for detail).

EXTI can provide not only interrupts but also event signals to the processor. The Cortex®-M33 processor fully implements the Wait For Interrupt (WFI), Wait For Event (WFE) and the Send Event (SEV) instructions. The Wake-up Interrupt Controller (WIC) enables the processor and NVIC to be put into a very low-power sleep mode leaving the WIC to identify

and prioritize interrupts and event. EXTI can be used to wake up processor and the whole system when some expected event occurs, such as a special GPIO pin toggling.

Hardware trigger

Hardware trigger may be used to detect the voltage change of external or internal signals. The software should follow these steps to use this function:

1. Configure EXTI sources in AFIO module based on application requirement.
2. Configure EXTI_RTEN and EXTI_FTEN to enable the rising or falling detection on related pins. (Software may set both RTENx and FTENx for a pin at the same time to detect both rising and falling changes on this pin).
3. Enable interrupts or events by setting related EXTI_INTEN or EXTI_EVENT bits.
4. EXTI starts to detect changes on the configured pins. The related interrupt or event will be triggered when desired change is detected on these pins. If the interrupt is triggered, the related PDx is set; if the event is triggered, the related PDx is not set. The software should response to the interrupts or events and clear these PDx bits.

Software trigger

Software may also trigger EXTI interrupts or events following these steps:

1. Enable interrupts or events by setting related EXTI_INTEN or EXTI_EVENT bits.
2. Set SWIEVx bits in EXTI_SWIEV register, the related interrupt or event will be triggered immediately. If the interrupt is triggered, the related PDx is set; if the event is triggered, the related PDx is not set. Software should response to these interrupts, and clear related PDx bits.

Table 6-3. EXTI source

EXTI Line Number	Source
0	PA8 / PC10 / PN2 / PC8
1	PA9 / PC11 / PE13 / PC9
2	PA0 / PC12 / PF14 / PG12
3	PF12
4	PA1 / PE14 / PF9 / PF13
5	PD8 / PG14 / PN7
6	PB1 / PC6 / PD2 / PG13
7	PB0 / PC7 / PN5
8	PD4 / PE8 / PF8 / PG15
9	PD5 / PD9 / PE9 / PF11
10	PB2 / PC0 / PD10 / PE10 / PF10
11	PC1 / PD11 / PG11
12	PC2 / PD12 / PE12
13	PC3 / PD13

EXTI Line Number	Source
14	PB14 / PC4 / PD14 / PE11
15	PB15 / PC5
16	LVD0
17	LVD1
18	CMP0
19	CMP1
20	CMP2
21	CMP3
22	FWDGT
23	I2C wakeup
24	NMI (PN2)

6.6 EXTI Register

EXTI base address: 0x4001 0400

6.6.1 Interrupt enable register (EXTI_INTEN)

Address offset: 0x00

Reset value: 0x0100 0000

This register has to be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved							INTEN24	INTEN23	INTEN22	INTEN21	INTEN20	INTEN19	INTEN18	INTEN17	INTEN16
							rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTEN15	INTEN14	INTEN13	INTEN12	INTEN11	INTEN10	INTEN9	INTEN8	INTEN7	INTEN6	INTEN5	INTEN4	INTEN3	INTEN2	INTEN1	INTEN0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:25	Reserved	Must be kept at reset value.
24:0	INTENx	Interrupt enable bit x (x = 0...24) 0: Interrupt from linex is disabled 1: Interrupt from linex is enabled

6.6.2 Event enable register (EXTI_EVEN)

Address offset: 0x04

Reset value: 0x0000 0000

This register has to be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved							EVEN23	EVEN22	EVEN21	EVEN20	EVEN19	EVEN18	EVEN17	EVEN16	
							rw	rw	rw	rw	rw	rw	rw	rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EVEN15	EVEN14	EVEN13	EVEN12	EVEN11	EVEN10	EVEN9	EVEN8	EVEN7	EVEN6	EVEN5	EVEN4	EVEN3	EVEN2	EVEN1	EVEN0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:24	Reserved	Must be kept at reset value.
23:0	EVENx	Event enable bit x (x = 0...23) 0: Event from linex is disabled. 1: Event from linex is enabled.

6.6.3 Rising edge trigger enable register (EXTI_RTEN)

Address offset: 0x08

Reset value: 0x0000 0000

This register has to be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved							RTEN24	RTEN23	RTEN22	RTEN21	RTEN20	RTEN19	RTEN18	RTEN17	RTEN16
							r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTEN15	RTEN14	RTEN13	RTEN12	RTEN11	RTEN10	RTEN9	RTEN8	RTEN7	RTEN6	RTEN5	RTEN4	RTEN3	RTEN2	RTEN1	RTEN0
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

Bits	Fields	Descriptions
31:25	Reserved	Must be kept at reset value.
24:0	RTENx	Rising edge trigger enable x (x = 0...24) 0: Rising edge of linex is invalid 1: Rising edge of linex is valid as an interrupt / event request

6.6.4 Falling edge trigger enable register (EXTI_FTEN)

Address offset: 0x0C

Reset value: 0x0100 0000

This register has to be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved							FTEN24	FTEN23	FTEN22	FTEN21	FTEN20	FTEN19	FTEN18	FTEN17	FTEN16
							r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FTEN15	FTEN14	FTEN13	FTEN12	FTEN11	FTEN10	FTEN9	FTEN8	FTEN7	FTEN6	FTEN5	FTEN4	FTEN3	FTEN2	FTEN1	FTEN0
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

Bits	Fields	Descriptions
31:25	Reserved	Must be kept at reset value.
24:0	FTENx	Falling edge trigger enable x (x = 0...24) 0: Falling edge of linex is invalid 1: Falling edge of linex is valid as an interrupt / event request

6.6.5 Software interrupt event register (EXTI_SWIEV)

Address offset: 0x10

Reset value: 0x0000 0000

This register has to be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								SWIEV23	SWIEV22	SWIEV21	SWIEV20	SWIEV19	SWIEV18	SWIEV17	SWIEV16
								rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SWIEV15	SWIEV14	SWIEV13	SWIEV12	SWIEV11	SWIEV10	SWIEV9	SWIEV8	SWIEV7	SWIEV6	SWIEV5	SWIEV4	SWIEV3	SWIEV2	SWIEV1	SWIEV0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:24	Reserved	Must be kept at reset value.
23:0	SWIEVx	Interrupt / Event software trigger x (x = 0...23) 0: Deactivate the EXTIx software interrupt / event request 1: Activate the EXTIx software interrupt / event request

6.6.6 Pending register (EXTI_PD)

Address offset: 0x14

Reset value: 0xFFFF XXXX

This register has to be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								PD24	PD23	PD22	PD21	PD20	PD19	PD18	PD17
								rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PD15	PD14	PD13	PD12	PD11	PD10	PD9	PD8	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0
rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1

Bits	Fields	Descriptions
31:25	Reserved	Must be kept at reset value.
24:0	PDx	Interrupt pending status x (x = 0...24) 0: EXTI linex is not triggered 1: EXTI linex is triggered. This bit is cleared to 0 by writing 1 to it.

6.6.7 Digital filter enable register (EXTI_DFEN)

Address offset: 0x18

Reset value: 0x0000 0000

This register has to be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								DFEN24	Reserved						
								rw							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

DFEN15	DFEN14	DFEN13	DFEN12	DFEN11	DFEN10	DFEN9	DFEN8	DFEN7	DFEN6	DFEN5	DFEN4	DFEN3	DFEN2	DFEN1	DFEN0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:25	Reserved	Must be kept at reset value.
24	DFENx	Digital filter enable bit x (x = 24) 0: Digital filter from linex is disabled 1: Digital filter from linex is enabled
23:16	Reserved	Must be kept at reset value.
15: 0	DFENx	Digital filter enable bit x (x = 0...15) 0: Digital filter from linex is disabled 1: Digital filter from linex is enabled

6.6.8 Sampling clock select register (EXTI_SCS)

Address offset: 0x1C

Reset value: 0x0000 0000

This register has to be accessed by word(32-bit).

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

Bits	Fields	Descriptions
31:2	Reserved	Must be kept at reset value.
1:0	SCS	Select the sampling clock 00: sampling clock is PCLK 01: sampling clock is PCLK / 8 10: sampling clock is PCLK / 32 11: sampling clock is PCLK / 64.

7 General-purpose and alternate-function I/Os (GPIO and AFIO)

7.1 Overview

There are up to 53 general purpose I/O pins (GPIO), named PA0 ~ PA1, PA9, PB0 ~ PB2, PB14 ~ PB15, PC0 ~ PC12, PD4 ~ PD14, PE8 ~ PE14, PF8 ~ PF14, PG11, PG13, PG15, PN2, PN5, PN6, PN7. for the device to implement logic input / output functions. Each GPIO port has related control and configuration registers to satisfy the requirements of specific applications. The external interrupts on the GPIO pins of the device have related control and configuration registers in the Interrupt / Event Controller Unit (EXTI).

The GPIO ports are pin-shared with other alternative functions (AFs) to obtain maximum flexibility on the package pins. The GPIO pins can be used as alternative functional pins by configuring the corresponding registers regardless of the AF input or output pins.

Each of the GPIO pins can be configured by software as output (push-pull or open-drain), input, peripheral alternate function or analog mode. Each GPIO pin can be configured as pull-up, pull-down or no pull-up / pull-down. All GPIOs are high-current capable except for analog mode.

7.2 Characteristics

- Input / output direction control.
- Schmitt trigger input function enable control.
- Each pin weak pull-up / pull-down function.
- Output push-pull / open drain enable control.
- Output set / reset control.
- External interrupt with programmable trigger edge – using EXTI configuration registers.
- Analog input / output configuration.
- Alternate function input / output configuration.
- Port configuration lock.
- Single cycle toggle output capability.
- Support GPIO output data hold function.

7.3 Function overview

Each of the general-purpose I / O ports can be configured as GPIO inputs, GPIO outputs, AF function or analog mode by GPIO 32-bit configuration registers (GPIOx_CTL). When select AF function, the pad input or output is decided by selected AF function output enable. When the port is output (GPIO output or AFIO output), it can be configured as push-pull or open

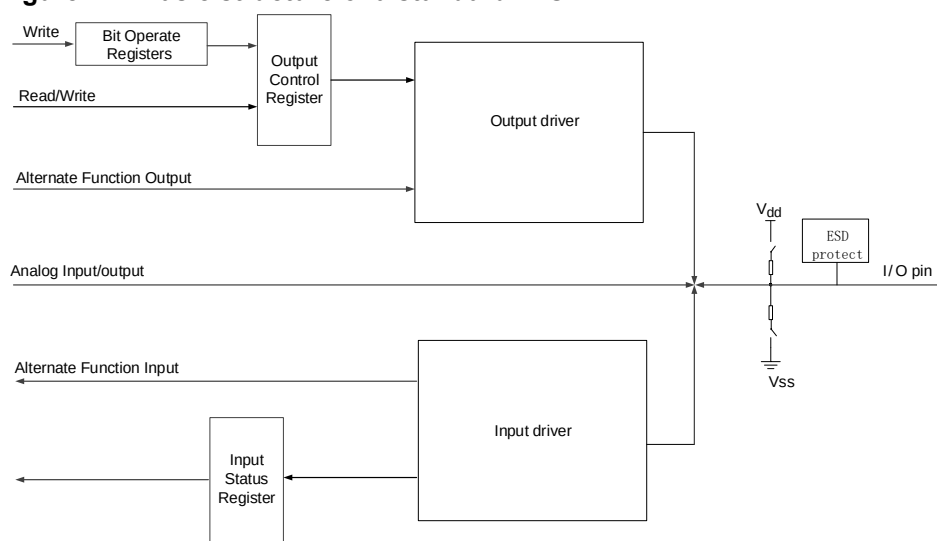
drain mode by GPIO output mode registers (GPIOx_OMODE). And the port max speed can be configured by GPIO output speed registers (GPIOx_OSPD). Each port can be configured as floating (no pull-up and pull-down), pull-up or pull-down function by GPIO pull-up / pull-down registers (GPIOx_PUD).

Table 7-1. GPIO configuration table

PAD TYPE			CTLy	OMy	PUDy
GPIO INPUT	X	Floating	00	X	00
		Pull-up			01
		Pull-down			10
GPIO OUTPUT	Push-pull	Floating	01	0	00
		Pull-up			01
		Pull-down			10
	Open-drain	Floating		1	00
		Pull-up			01
		Pull-down			10
AFIO INPUT	X	Floating	10	X	00
		Pull-up			01
		Pull-down			10
AFIO OUTPUT	Push-pull	Floating	10	0	00
		Pull-up			01
		Pull-down			10
	Open-drain	Floating		1	00
		Pull-up			01
		Pull-down			10
ANALOG	X	X	11	X	XX

Figure 7-1. Basic structure of a standard I / O shows the basic structure of an I / O port bit.

Figure 7-1. Basic structure of a standard I / O



7.3.1 GPIO pin configuration

During or just after the reset period, the alternative functions are all inactive and the GPIO ports are configured into the input floating mode that input disabled without Pull-Up (PU) / Pull-Down (PD) resistors.

Note:In LQFP64, set the PN7FUNC bit to 0 in the GPIOx_FUNCSEL register to enable the TSEL function, and when PN7 is driven high, the PG11-PG15 pins will be used for JTAG/SWD functionality. And the JTAG / Serial-Wired Debug pins are in input PU / PD mode.

PG13: JTDI in PU mode

PG12: JTCK / SWCLK in PD mode

PG14: JTMS / SWDIO in PU mode

PG15: NJRST in PU mode

PG11: JTDO in output mode

Refer to the detailed functions of the debug pins in [14.2 Pin assignment](#).

The GPIO pins can be configured as inputs or outputs. When the GPIO pins are configured as input pins, all GPIO pins have an internal weak pull-up and weak pull-down which can be chosen. And the data on the external pins can be captured at every AHB clock cycle to the port input status register (GPIOx_ISTAT).

When the GPIO pins are configured as output pins, user can configure the speed of the ports. And chooses the output driver mode: Push-Pull or Open-Drain mode. The value of the port output control register (GPIOx_OCTL) is output on the I / O pin.

There is no need to read-then-write when programming the GPIOx_OCTL at bit level, the user can modify only one or several bits in a single atomic AHB write access by programming '1' to the bit operate register (GPIOx_BOP, or for clearing only GPIOx_BC, or for toggle only GPIOx_TG). The other bits will not be affected.

7.3.2 External interrupt / event lines

The port can use external interrupt / event lines only if it is configured in input mode.

7.3.3 Alternate functions (AF)

When the port is configured as AFIO (set CTLY bits to "0b10", which is in GPIOx_CTL registers), the port is used as peripheral alternate functions. Each port has sixteen alternate functions can be configured by GPIO alternate functions selected registers (GPIOx_AFSELY (y = 0,1)). The detail alternate function assignments for each port are in the device datasheet.

7.3.4 Additional functions

Some pins have additional functions, which have priority over the configuration in the standard GPIO registers. When for ADC or DAC additional functions, the port must be configured as

analog mode. When for WKUPx and oscillators additional functions, the port type is set automatically by related PMU and RCU registers. These ports can be used as normal GPIO when the additional functions disabled.

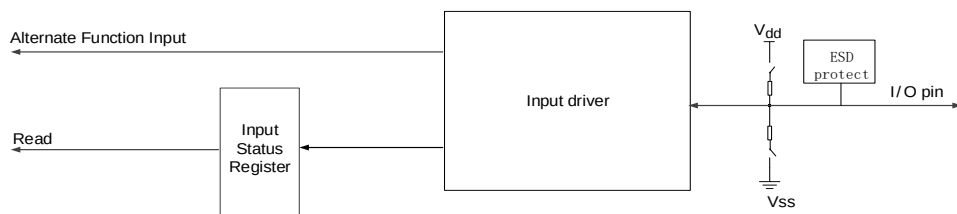
7.3.5 Input configuration

When GPIO pin is configured as input:

- The schmitt trigger input is enabled.
- The weak pull-up and pull-down resistors could be chosen.
- Every AHB clock cycle the data present on the I / O pin is got to the port input status Register.
- Disable the output buffer.

[Figure 7-2. Basic structure of Input configuration](#) shows the input configuration.

Figure 7-2. Basic structure of Input configuration



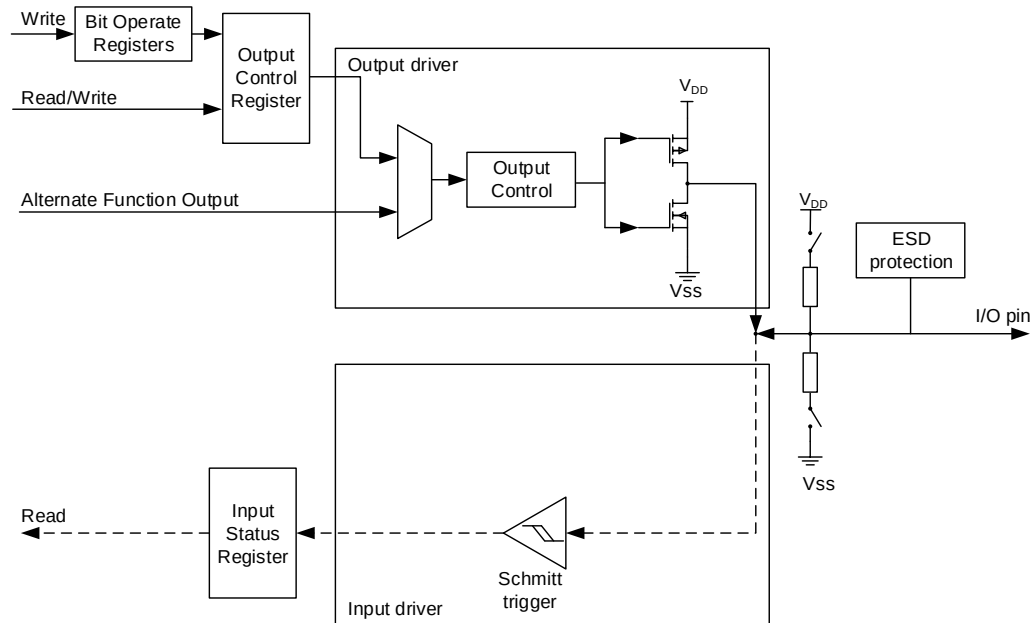
7.3.6 Output configuration

When GPIO pin is configured as output:

- The schmitt trigger input is enabled.
- The weak pull-up and pull-down resistors could be chosen.
- The output buffer is enabled.
- Open Drain Mode: The pad output low level when a “0” in the output control register; while the pad leaves Hi-Z when a “1” in the output control register.
- Push-Pull Mode: The pad output low level when a “0” in the output control register; while the pad output high level when a “1” in the output control register.
- A read access to the port output control register gets the last written value.
- A read access to the port input status register gets the I / O state.

[Figure 7-3. Basic structure of Output configuration](#) shows the output configuration.

Figure 7-3. Basic structure of Output configuration



7.3.7 Analog configuration

When GPIO pin is used as analog configuration:

- The weak pull-up and pull-down resistors are disabled.
- The output buffer is disabled.
- The schmitt trigger input is disabled.
- The port input status register of this I / O port bit is "0".

[Figure 7-4. Basic structure of Analog configuration](#) shows the analog configuration.

Figure 7-4. Basic structure of Analog configuration



7.3.8 Alternate function (AF) configuration

To suit for different device packages, the GPIO supports some alternate functions mapped to some other pins by software.

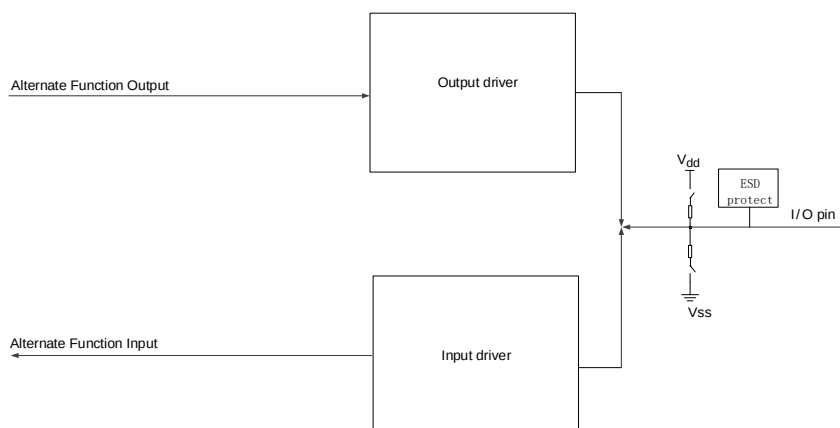
When be configured as alternate function:

- The output buffer is enabled in open-drain or push-pull configuration.
- The output buffer is driven by the peripheral.
- The schmitt trigger input is enabled.
- The weak pull-up and pull-down resistors could be chosen.

- The I / O pin data is stored into the port input status register every AHB clock.
- A read access to the port input status register gets the I / O state.
- A read access to the port output control register gets the last written value.

[Figure 7-5. Basic structure of Alternate function configuration](#) shows the alternate function configuration.

Figure 7-5. Basic structure of Alternate function configuration



7.3.9 GPIO locking function

The locking mechanism allows the IO configuration to be protected. The protected registers are GPIOx_CTL, GPIOx_OMODE, GPIOx_OSPD, GPIOx_PUD, GPIOx_AFSELz (z=0, 1) and GPIOx_HLD. It allows the I / O configuration to be frozen by the 32-bit locking register (GPIOx_LOCK). When the special LOCK sequence has occurred on LKK bit in GPIOx_LOCK register and the LKy bit is set in GPIOx_LOCK register, the corresponding port is locked and the corresponding port configuration cannot be modified until the next reset. It recommended to be used in the configuration of driving a power module.

7.3.10 GPIO single cycle toggle function

GPIO could toggle the I / O output level in single AHB cycle by writing 1 to the corresponding bit of GPIOx_TG register. The output signal frequency could up to the half of the AHB clock.

7.3.11 GPIO pin level hold function

GPIO could hold the pin level by writing 1 to the corresponding bit of GPIOx_HLDCFG register. Once held, the pin level remains unchanged even if the GPIOx_OCTL register is changed. The pin level held by the port will be updated to the value in the GPIOx_OCTL by writing 1 to the HLDUPD bit of GPIOx_HLDCTL register.

7.4 Register definition

GPIOA base address: 0x4800 0000

GPIOB base address: 0x4800 0400

GPIOC base address: 0x4800 0800

GPIOD base address: 0x4800 0C00

GPIOE base address: 0x4800 1000

GPIOF base address: 0x4800 1400

GPIOG base address: 0x4800 1800

GPION base address: 0x4800 4000

7.4.1 Port control register (GPIOx_CTL, x = A..G,N)

Address offset: 0x00

Reset value: 0x0800 0000 for port G

Reset value: 0x0000 2020 for port N

Reset value: 0x0000 0000 for other ports

This register has to be accessed by byte (8-bit), half-word (16-bit) and word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CTL15[1:0]		CTL14[1:0]		CTL13[1:0]		CTL12[1:0]		CTL11[1:0]		CTL10[1:0]		CTL9[1:0]		CTL8[1:0]	
rw		rw		rw		rw		rw		rw		rw		rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CTL7[1:0]		CTL6[1:0]		CTL5[1:0]		CTL4[1:0]		CTL3[1:0]		CTL2[1:0]		CTL1[1:0]		CTL0[1:0]	
rw		rw		rw		rw		rw		rw		rw		rw	

Bits	Fields	Descriptions
31:30	CTL15[1:0]	Pin 15 configuration bits These bits are set and cleared by software. Refer to CTL0[1:0] description
29:28	CTL14[1:0]	Pin 14 configuration bits These bits are set and cleared by software. Refer to CTL0[1:0] description
27:26	CTL13[1:0]	Pin 13 configuration bits These bits are set and cleared by software. Refer to CTL0[1:0] description
25:24	CTL12[1:0]	Pin 12 configuration bits These bits are set and cleared by software.

		Refer to CTL0[1:0] description.
23:22	CTL11[1:0]	Pin 11 configuration bits These bits are set and cleared by software. Refer to CTL0[1:0] description
21:20	CTL10[1:0]	Pin 10 configuration bits These bits are set and cleared by software. Refer to CTL0[1:0] description
19:18	CTL9[1:0]	Pin 9 configuration bits These bits are set and cleared by software. Refer to CTL0[1:0] description
17:16	CTL8[1:0]	Pin 8 configuration bits These bits are set and cleared by software. Refer to CTL0[1:0] description
15:14	CTL7[1:0]	Pin 7 configuration bits These bits are set and cleared by software. Refer to CTL0[1:0] description
13:12	CTL6[1:0]	Pin 6 configuration bits These bits are set and cleared by software. Refer to CTL0[1:0] description
11:10	CTL5[1:0]	Pin 5 configuration bits These bits are set and cleared by software. Refer to CTL0[1:0] description
9:8	CTL4[1:0]	Pin 4 configuration bits These bits are set and cleared by software. Refer to CTL0[1:0] description
7:6	CTL3[1:0]	Pin 3 configuration bits These bits are set and cleared by software. Refer to CTL0[1:0] description
5:4	CTL2[1:0]	Pin 2 configuration bits These bits are set and cleared by software. Refer to CTL0[1:0] description
3:2	CTL1[1:0]	Pin 1 configuration bits These bits are set and cleared by software. Refer to CTL0[1:0] description
1:0	CTL0[1:0]	Pin 0 configuration bits These bits are set and cleared by software. 00: GPIO Input mode (reset value)

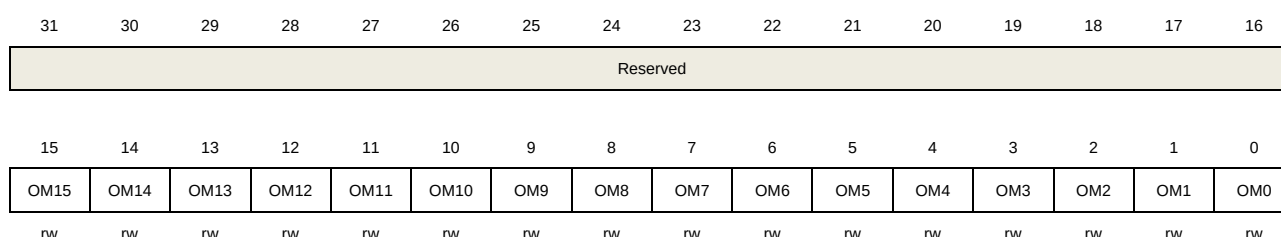
- 01: GPIO output mode
- 10: Alternate function mode
- 11: Analog mode (Input and Output)

7.4.2 Port output mode register (GPIOx_OMODE, x = A..G,N)

Address offset: 0x04

Reset value: 0x0000 0000

This register has to be accessed by byte (8-bit), half-word (16-bit) and word (32-bit).



Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15	OM15	Pin 15 output mode bit These bits are set and cleared by software. Refer to OM0 description
14	OM14	Pin 14 output mode bit These bits are set and cleared by software. Refer to OM0 description
13	OM13	Pin 13 output mode bit These bits are set and cleared by software. Refer to OM0 description
12	OM12	Pin 12 output mode bit These bits are set and cleared by software. Refer to OM0 description
11	OM11	Pin 11 output mode bit These bits are set and cleared by software. Refer to OM0 description
10	OM10	Pin 10 output mode bit These bits are set and cleared by software. Refer to OM0 description
9	OM9	Pin 9 output mode bit These bits are set and cleared by software. Refer to OM0 description

8	OM8	Pin 8 output mode bit These bits are set and cleared by software. Refer to OM0 description
7	OM7	Pin 7 output mode bit These bits are set and cleared by software. Refer to OM0 description
6	OM6	Pin 6 output mode bit These bits are set and cleared by software. Refer to OM0 description
5	OM5	Pin 5 output mode bit These bits are set and cleared by software. Refer to OM0 description
4	OM4	Pin 4 output mode bit These bits are set and cleared by software. Refer to OM0 description.
3	OM3	Pin 3 output mode bit These bits are set and cleared by software. Refer to OM0 description
2	OM2	Pin 2 output mode bit These bits are set and cleared by software. Refer to OM0 description
1	OM1	Pin 1 output mode bit These bits are set and cleared by software. Refer to OM0 description
0	OM0	Pin 0 output mode bit These bits are set and cleared by software. 0: Output push-pull mode (reset value) 1: Output open-drain mode

7.4.3 Port output speed register (GPIOx_OSPD, x = A..G,N)

Address offset: 0x08

Reset value: 0x0000 0000

This register has to be accessed by byte (8-bit), half-word (16-bit) and word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OSPD15[1:0]		OSPD14[1:0]		OSPD13[1:0]		OSPD12[1:0]		OSPD11[1:0]		OSPD10[1:0]		OSPD9[1:0]		OSPD8[1:0]	
rw		rw		rw		rw		rw		rw		rw		rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OSPD7[1:0]		OSPD6[1:0]		OSPD5[1:0]		OSPD4[1:0]		OSPD3[1:0]		OSPD2[1:0]		OSPD1[1:0]		OSPD0[1:0]	

rw rw rw rw rw rw rw rw

Bits	Fields	Descriptions
31:30	OSPD15[1:0]	Pin 15 output max speed bits These bits are set and cleared by software. Refer to OSPD0[1:0] description
29:28	OSPD14[1:0]	Pin 14 output max speed bits These bits are set and cleared by software. Refer to OSPD0[1:0] description
27:26	OSPD13[1:0]	Pin 13 output max speed bits These bits are set and cleared by software. Refer to OSPD0[1:0] description
25:24	OSPD12[1:0]	Pin 12 output max speed bits These bits are set and cleared by software. Refer to OSPD0[1:0] description
23:22	OSPD11[1:0]	Pin 11 output max speed bits These bits are set and cleared by software. Refer to OSPD0[1:0] description.
21:20	OSPD10[1:0]	Pin 10 output max speed bits These bits are set and cleared by software. Refer to OSPD0[1:0] description
19:18	OSPD9[1:0]	Pin 9 output max speed bits These bits are set and cleared by software. Refer to OSPD0[1:0] description
17:16	OSPD8[1:0]	Pin 8 output max speed bits These bits are set and cleared by software. Refer to OSPD0[1:0] description
15:14	OSPD7[1:0]	Pin 7 output max speed bits These bits are set and cleared by software. Refer to OSPD0[1:0] description
13:12	OSPD6[1:0]	Pin 6 output max speed bits These bits are set and cleared by software. Refer to OSPD0[1:0] description
11:10	OSPD5[1:0]	Pin 5 output max speed bits These bits are set and cleared by software. Refer to OSPD0[1:0] description
9:8	OSPD4[1:0]	Pin 4 output max speed bits These bits are set and cleared by software.

		Refer to OSPD0[1:0] description
7:6	OSPD3[1:0]	Pin 3 output max speed bits These bits are set and cleared by software. Refer to OSPD0[1:0] description
5:4	OSPD2[1:0]	Pin 2 output max speed bits These bits are set and cleared by software. Refer to OSPD0[1:0] description
3:2	OSPD1[1:0]	Pin 1 output max speed bits These bits are set and cleared by software. Refer to OSPD0[1:0] description
1:0	OSPD0[1:0]	Pin 0 output max speed bits These bits are set and cleared by software. Output drive capability control GPIO_N5T_HIA type: PD9-PD14 / PE8-13 / PF13 / PG11 00: Low drive(IOL=1mA,25MHz) 01: High drive(IOL=1mA,50MHz) 1x: High current drive(IOL=15mA) GPIO_5T type: PF8 / PF9 / PF10 00: Low drive(IOL=3mA, 25MHz) 01: High dirve(IOL=6mA, 50Mhz) 1x: Reserved GPIO_N5T / GPIO_N5T_WKUP/GPIO_N5T_SE type: others 00: Low drive(IOL=1mA, 25MHz) 01: High drive(IOL=1mA, 50MHz) 1x: Reserved

7.4.4 Port pull-up / pull-down register (GPIOx_PUD, x = A..G,N)

Address offset: 0x0C

Reset value: 0x0000 0000

This register has to be accessed by byte (8-bit), half-word (16-bit) and word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PUD15[1:0]		PUD14[1:0]		PUD13[1:0]		PUD12[1:0]		PUD11[1:0]		PUD10[1:0]		PUD9[1:0]		PUD8[1:0]	
rw		rw		rw		rw		rw		rw		rw		rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PUD7[1:0]		PUD6[1:0]		PUD5[1:0]		PUD4[1:0]		PUD3[1:0]		PUD2[1:0]		PUD1[1:0]		PUD0[1:0]	
rw		rw		rw		rw		rw		rw		rw		rw	

Bits	Fields	Descriptions
31:30	PUD15[1:0]	Pin 15 pull-up or pull-down bits These bits are set and cleared by software.

		Refer to PUD0[1:0] description.
29:28	PUD14[1:0]	Pin 14 pull-up or pull-down bits These bits are set and cleared by software. Refer to PUD0[1:0] description.
27:26	PUD13[1:0]	Pin 13 pull-up or pull-down bits These bits are set and cleared by software. Refer to PUD0[1:0] description.
25:24	PUD12[1:0]	Pin 12 pull-up or pull-down bits These bits are set and cleared by software. Refer to PUD0[1:0] description.
23:22	PUD11[1:0]	Pin 11 pull-up or pull-down bits These bits are set and cleared by software. Refer to PUD0[1:0] description.
21:20	PUD10[1:0]	Pin 10 pull-up or pull-down bits These bits are set and cleared by software. Refer to PUD0[1:0] description.
19:18	PUD9[1:0]	Pin 9 pull-up or pull-down bits These bits are set and cleared by software. Refer to PUD0[1:0] description.
17:16	PUD8[1:0]	Pin 8 pull-up or pull-down bits These bits are set and cleared by software. Refer to PUD0[1:0] description.
15:14	PUD7[1:0]	Pin 7 pull-up or pull-down bits These bits are set and cleared by software. Refer to PUD0[1:0] description.
13:12	PUD6[1:0]	Pin 6 pull-up or pull-down bits These bits are set and cleared by software. Refer to PUD0[1:0] description.
11:10	PUD5[1:0]	Pin 5 pull-up or pull-down bits These bits are set and cleared by software. Refer to PUD0[1:0] description.
9:8	PUD4[1:0]	Pin 4 pull-up or pull-down bits These bits are set and cleared by software. Refer to PUD0[1:0] description.
7:6	PUD3[1:0]	Pin 3 pull-up or pull-down bits These bits are set and cleared by software. Refer to PUD0[1:0] description.

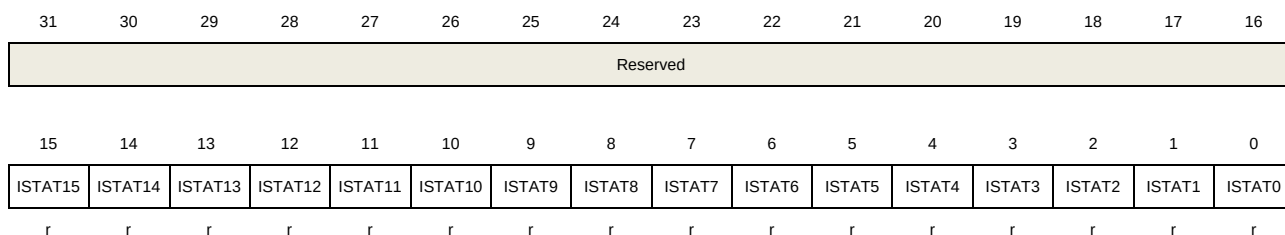
5:4	PUD2[1:0]	Pin 2 pull-up or pull-down bits These bits are set and cleared by software. Refer to PUD0[1:0] description.
3:2	PUD1[1:0]	Pin 1 pull-up or pull-down bits These bits are set and cleared by software. Refer to PUD0[1:0] description.
1:0	PUD0[1:0]	Pin 0 pull-up or pull-down bits These bits are set and cleared by software. 00: Floating mode, no pull-up and pull-down (reset value) 01: With pull-up mode 10: With pull-down mode 11: Reserved

7.4.5 Port input status register (GPIOx_ISTAT, x = A..G,N)

Address offset: 0x10

Reset value: 0x0000 XXXX

This register has to be accessed by word (32-bit).



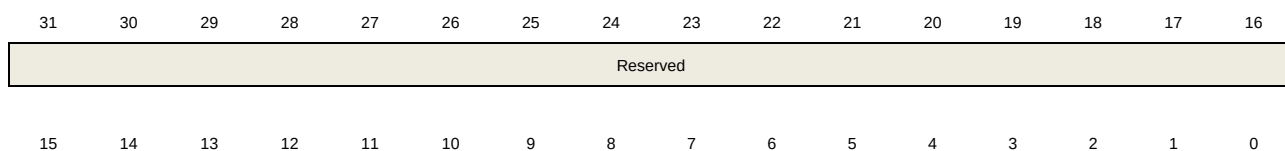
Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	ISTATy	Pin input status (y=0..15) These bits are set and cleared by hardware. 0: Input signal low 1: Input signal high

7.4.6 Port output control register (GPIOx_OCTL, x = A..G,N)

Address offset: 0x14

Reset value: 0x0000 0000

This register has to be accessed by byte (8-bit), half-word (16-bit) and word (32-bit).



OCTL15	OCTL14	OCTL13	OCTL12	OCTL11	OCTL10	OCTL9	OCTL8	OCTL7	OCTL6	OCTL5	OCTL4	OCTL3	OCTL2	OCTL1	OCTL0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	OCTLy	Pin output control (y=0..15) These bits are set and cleared by software. 0: Pin output low 1: Pin output high

7.4.7 Port bit operate register (GPIOx_BOP, x = A..G,N)

Address offset: 0x18

Reset value: 0x0000 0000

This register has to be accessed by byte (8-bit), half-word (16-bit) and word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CR15	CR14	CR13	CR12	CR11	CR10	CR9	CR8	CR7	CR6	CR5	CR4	CR3	CR2	CR1	CR0
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BOP15	BOP14	BOP13	BOP12	BOP11	BOP10	BOP9	BOP8	BOP7	BOP6	BOP5	BOP4	BOP3	BOP2	BOP1	BOP0
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

Bits	Fields	Descriptions
31:16	CRy	Pin Clear bit y(y=0..15) These bits are set and cleared by software. 0: No action on the corresponding OCTLy bit 1: Clear the corresponding OCTLy bit to 0
15:0	BOPy	Pin Set bit y(y=0..15) These bits are set and cleared by software. 0: No action on the corresponding OCTLy bit 1: Set the corresponding OCTLy bit to 1

7.4.8 Port configuration lock register (GPIOx_LOCK, x = A..G,N)

Address offset: 0x1C

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

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

LK15	LK14	LK13	LK12	LK11	LK10	LK9	LK8	LK7	LK6	LK5	LK4	LK3	LK2	LK1	LK0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:17	Reserved	Must be kept at reset value.
16	LKK	Lock sequence key It can only be setted using the Lock Key Writing Sequence. And can always be read. 0: GPIO_LOCK register is not locked and the port configuration is not locked. 1: GPIO_LOCK register is locked until an MCU reset. LOCK key configuration sequence. Write 1→Write 0→Write 1→ Read 0→ Read 1 Note: The value of LK[15:0] must hold during the LOCK Key Writing sequence.
15:0	LKy	Port Lock bit y (y=0..15) These bits are set and cleared by software. 0: The corresponding bit port configuration is not locked 1: The corresponding bit port configuration is locked when LKK bit is "1"

7.4.9 Alternate function selected register 0 (GPIOx_AFSEL0, x = A..G,N)

Address offset: 0x20

Reset value: 0x0000 0000

This register has to be accessed by byte (8-bit), half-word (16-bit) and word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SEL7[3:0]				SEL6[3:0]				SEL5[3:0]				SEL4[3:0]			
rw				rw				rw				rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SEL3[3:0]				SEL2[3:0]				SEL1[3:0]				SEL0[3:0]			
rw				rw				rw				rw			

Bits	Fields	Descriptions
31:28	SEL7[3:0]	Pin 7 alternate function selected These bits are set and cleared by software. Refer to SEL0 [3:0] description.
27:24	SEL6[3:0]	Pin 6 alternate function selected These bits are set and cleared by software. Refer to SEL0 [3:0] description.
23:20	SEL5[3:0]	Pin 5 alternate function selected These bits are set and cleared by software. Refer to SEL0 [3:0] description.
19:16	SEL4[3:0]	Pin 4 alternate function selected

		These bits are set and cleared by software. Refer to SEL0 [3:0] description.
15:12	SEL3[3:0]	Pin 3 alternate function selected These bits are set and cleared by software. Refer to SEL0 [3:0] description.
11:8	SEL2[3:0]	Pin 2 alternate function selected These bits are set and cleared by software. Refer to SEL0 [3:0] description.
7:4	SEL1[3:0]	Pin 1 alternate function selected These bits are set and cleared by software. Refer to SEL0 [3:0] description.
3:0	SEL0[3:0]	Pin 0 alternate function selected These bits are set and cleared by software. 0000: AF0 selected (reset value) 0001: AF1 selected 0010: AF2 selected 0011: AF3 selected ... 1111: AF15 selected

7.4.10 Alternate function selected register 1 (GPIOx_AFSEL1, x = A..G,N)

Address offset: 0x24

Reset value: 0x0000 0000

This register has to be accessed by byte (8-bit), half-word (16-bit) and word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SEL15[3:0]				SEL14[3:0]				SEL13[3:0]				SEL12[3:0]			
rw				rw				rw				rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SEL11[3:0]				SEL10[3:0]				SEL9[3:0]				SEL8[3:0]			
rw				rw				rw				rw			

Bits	Fields	Descriptions
31:28	SEL15[3:0]	Pin 15 alternate function selected These bits are set and cleared by software. Refer to SEL8[3:0] description.
27:24	SEL14[3:0]	Pin 14 alternate function selected These bits are set and cleared by software. Refer to SEL8[3:0] description.
23:20	SEL13[3:0]	Pin 13 alternate function selected

		These bits are set and cleared by software. Refer to SEL8[3:0] description.
19:16	SEL12[3:0]	Pin 12 alternate function selected These bits are set and cleared by software. Refer to SEL8[3:0] description.
15:12	SEL11[3:0]	Pin 11 alternate function selected These bits are set and cleared by software. Refer to SEL8[3:0] description.
11:8	SEL10[3:0]	Pin 10 alternate function selected These bits are set and cleared by software. Refer to SEL8[3:0] description.
7:4	SEL9[3:0]	Pin 9 alternate function selected These bits are set and cleared by software. Refer to SEL8[3:0] description.
3:0	SEL8[3:0]	Pin 8 alternate function selected These bits are set and cleared by software. 0000: AF0 selected (reset value) 0001: AF1 selected 0010: AF2 selected 0011: AF3 selected ... 1111: AF15 selected

7.4.11 Bit clear register (GPIOx_BC, x = A..G,N)

Address offset: 0x28

Reset value: 0x0000 0000

This register has to be accessed by byte (8-bit), half-word (16-bit) and word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CR15	CR14	CR13	CR12	CR11	CR10	CR9	CR8	CR7	CR6	CR5	CR4	CR3	CR2	CR1	CR0
W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CRy	Pin Clear bit y (y=0..15) These bits are set and cleared by software.

0: No action on the corresponding OCTLY bit

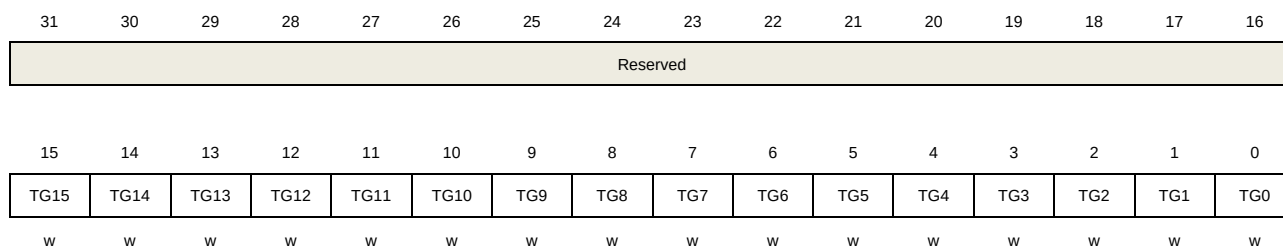
1: Clear the corresponding OCTLY bit

7.4.12 Port bit toggle register (GPIOx_TG, x = A..G,N)

Address offset: 0x2C

Reset value: 0x0000 0000

This register has to be accessed by byte (8-bit), half-word (16-bit) and word (32-bit).



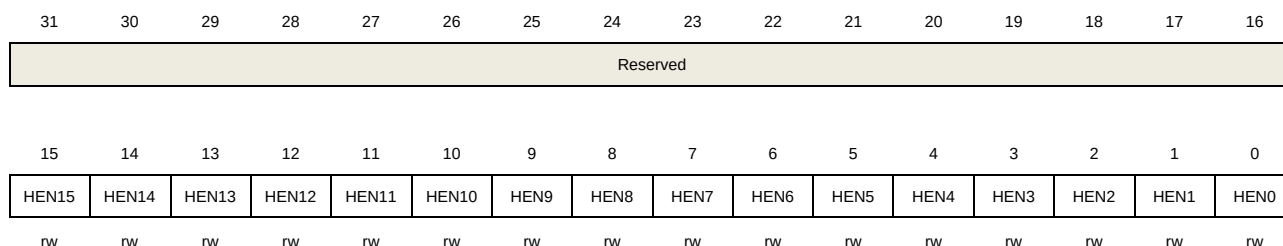
Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	TGy	Pin toggle bit y (y=0..15) These bits are set and cleared by software. 0: No action on the corresponding OCTLY bit 1: Toggle the corresponding OCTLY bit

7.4.13 Pin level hold config register (GPIOx_HLDCFG, x = A..G,N)

Address offset: 0x50

Reset value: 0x0000 0000

This register has to be accessed by byte (8-bit), half-word (16-bit) and word (32-bit).



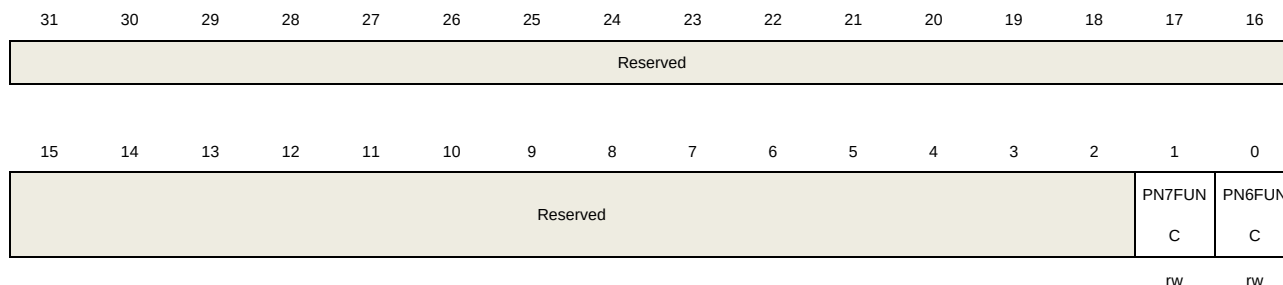
Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	HENy	Pin level hold enable bit y (y=0..15) These bits are set and cleared by software. 0: The pin level hold function is disable 1: The pin level hold function is enable

7.4.14 Pin function select register (GPIOx_FUNCSEL, x = N)

Address offset: 0x60

Reset value: 0x0000 0000

This register has to be accessed by byte (8-bit), half-word (16-bit) and word (32-bit).



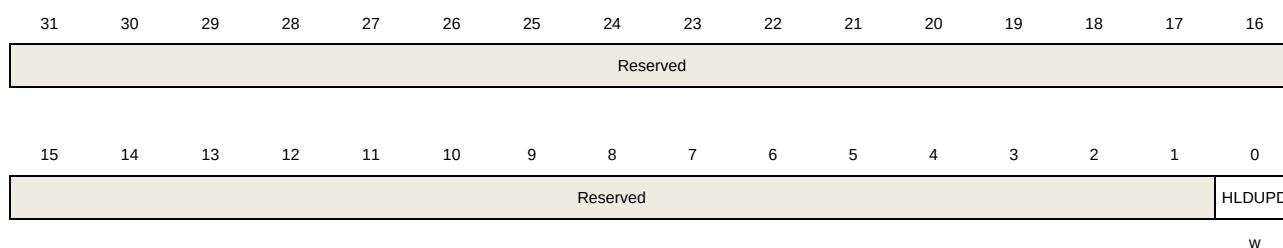
Bits	Fields	Descriptions
31:2	Reserved	Must be kept at reset value.
1	PN7FUNC	PN7 function select bit 0: The TSEL function is selected 1: The general-purpose input/output pin function is selected
0	PN6FUNC	PN6 function select bit 0: The BOOT/One-line function is selected 1: The general-purpose input/output pin function is selected

7.4.15 Pin level hold control register (GPIOx_HLDCTL, x = N)

Address offset: 0x64

Reset value: 0x0000 0000

This register has to be accessed by byte (8-bit), half-word (16-bit) and word (32-bit).



Bits	Fields	Descriptions
31:1	Reserved	Must be kept at reset value.
0	HLDUPD	Pin level held updata bit 0: No effect 1: The pin level held by the port will be updated to the value in the GPIOx_OCTL. Note: This bit only can be set to 1, and then it is automatically cleared by

hardware.

8 Port Output Controller (POC)

8.1 Overview

The POC (Port Output Controller) can be used to disable pin outputs from the TIMER and GPTIMER channels in a variety of situations. The disabled output pins can choose high-impedance state (Hi-Z) output or general purpose I/O (GPIO) control output. These pins can be disabled by the POC module only if they select the TIMER or GPTIMER channels output as alternate function.

8.2 Characteristics

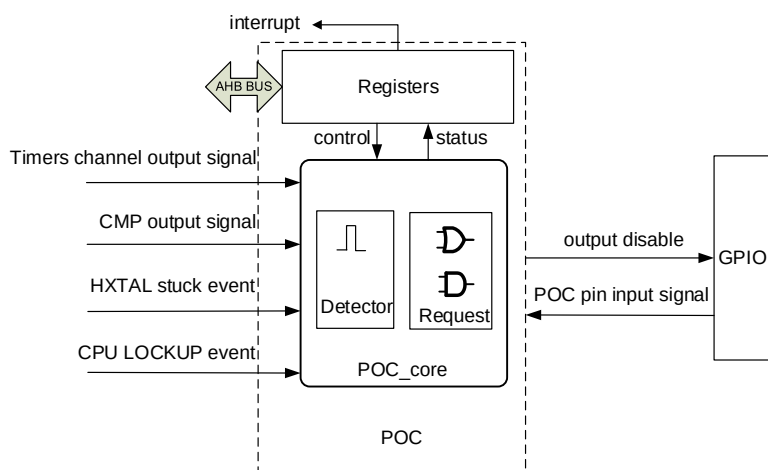
The main features of POC are as follows:

- POC_INn (n=0,1...5) pin input detection:
 - Edge detection.
 - Level detection.
 - Configurable sampling clock frequency.
 - Optional valid sampling times.
 - Input inverting control.
- Request sources for pin disable:
 - From POC_INn (n=0,1...5) pin input detection.
 - From TIMER0/7 complementary PWM channel output detection.
 - From software generation.
 - From HXTAL stuck event.
 - From CPU LOCKUP event.
 - From comparator output detection.
- Selectable target TIMER for pin disable.
- Detection masking function:
 - Mask POC_INn (n=0,1...5) pin input detection.
 - Mask comparator output detection.
- Interrupt generation:
 - Valid signal input of POC_INn (n=0,1...5) pin.
 - Concurrent active-level outputs of TIMER0/7 complementary PWM channels.

8.3 Function overview

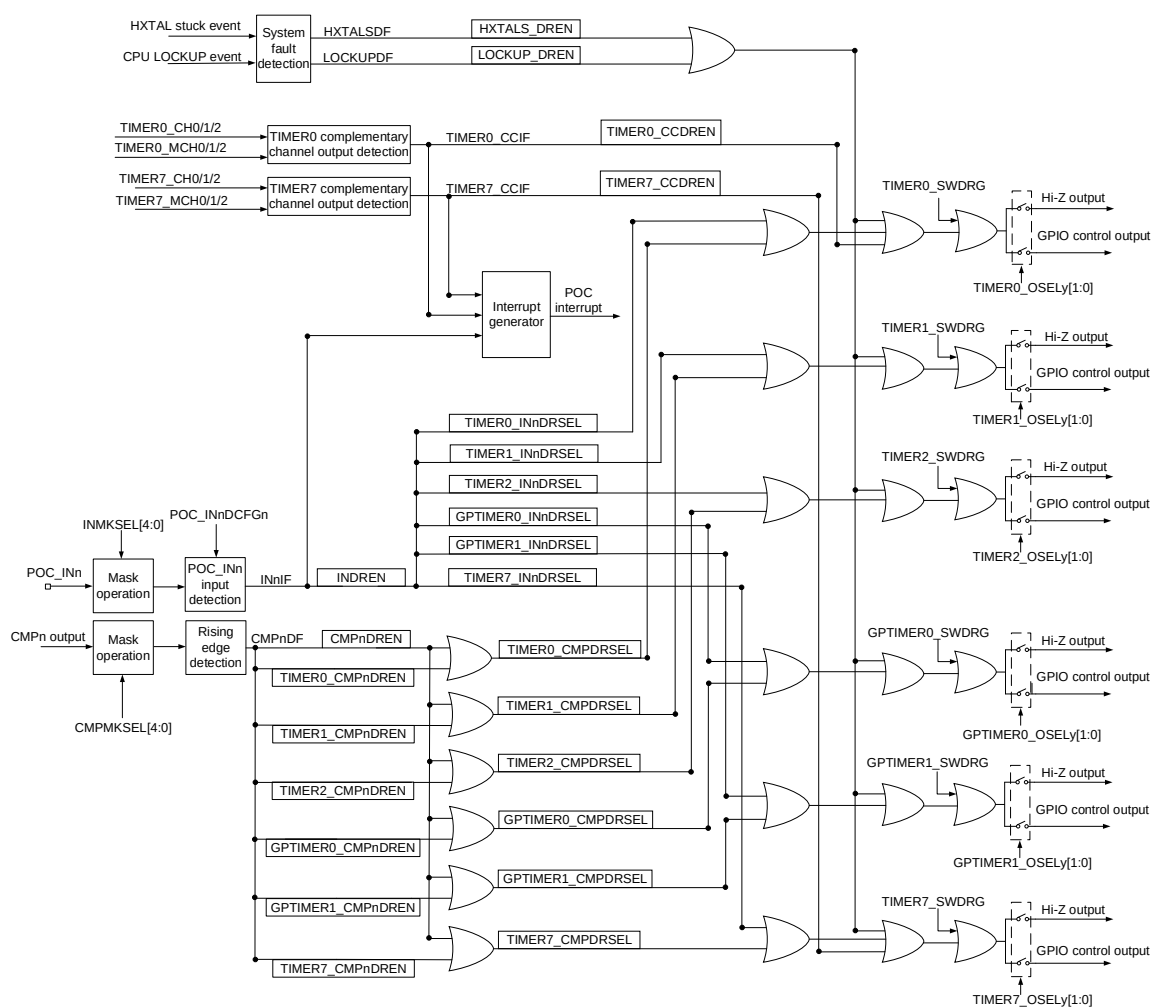
The block diagram of POC is shown below:

Figure 8-1. POC block diagram



The output pin disable block diagram is shown below:

Figure 8-2. Output disable control for TIMER and GPTIMER channel pin



The target timers and channels for output disable is shown below:

Table 8-1. The target timers and channels for output disable

	TIMER				GPTIMER	
Target timers	TIMER0	TIMER7	TIMER1	TIMER2	GPTIMER0	GPTIMER1
Target channels	TIMER0_CH0	TIMER7_CH0	TIMER1_	TIMER2_		
	TIMER0_CH1	TIMER7_CH1	CH0	CH0		
	TIMER0_CH2	TIMER7_CH2	TIMER1_	TIMER2_	GPTIMER0	GPTIMER1
	TIMER0_CH3	TIMER7_CH3	CH1	CH1	_CH0	_CH0
	TIMER0_MCH0	TIMER7_MCH0	TIMER1_	TIMER2_	GPTIMER0	GPTIMER1
	TIMER0_MCH1	TIMER7_MCH1	CH2	CH2	_CH1	_CH1
	TIMER0_MCH2	TIMER7_MCH2	TIMER1_	TIMER2_		
	TIMER0_MCH3	TIMER7_MCH3	CH3	CH3		

8.3.1 POC clock

The clock of the POC which is connected to AHB bus, is synchronous with HCLK.

8.3.2 Output disable function

POC input detection

The POC has six external input pins. The input signal of each POC_INn (n=0,1...5) pin can be used for edge and level detection. When the INPL bit in POC_INnDCFG (n=0,1...5) registers is set to 0, the falling edge or low level is selected as valid detection signals. When the INPL bit is set to 1, rising edge or high level detection is performed.

Note: When configuring POC pin input detection by the software, configure the POC detection first, and then configure the POC pins as the corresponding GPIO alternate function.

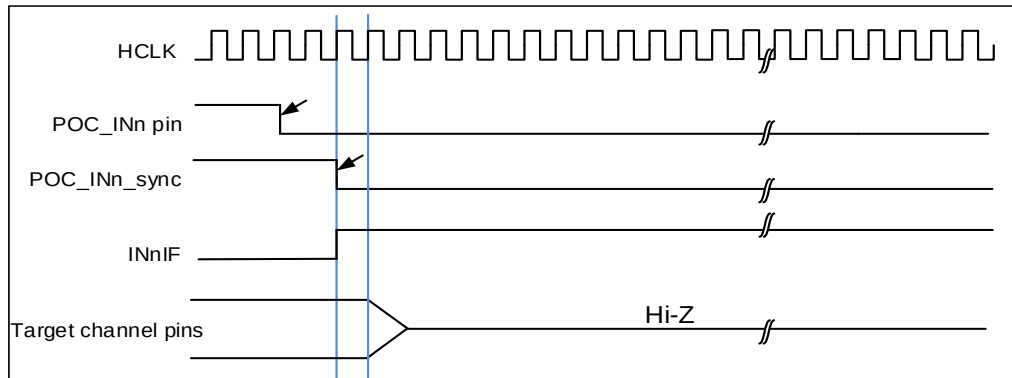
Edge detection mode

The input signal of POC_INn (n=0,1...5) pin is sampled with HCLK. When INDMSEL[3:0] bits in POC_INnDCFG (n=0,1...5) registers is set to 4'b0000, if falling edge (INPL=0) or rising edge (INPL=1) signal comes into POC_INn (n=0,1...5) pin, after the input edge is synchronized, flag bit INnIF (n=0,1...5) in POC_STAT0 register will be set by the hardware, and a disabling request will be generated by POC_INn (n=0,1...5) pin edge detection.

If the disabling request has been enabled by setting the INDREN bit in POC_INnDCFG (n=0,1...5) registers, and the POC_REQSEL0/1 register has been used to select a target timer for the disabling request, the pins connected to the target timer channels will be disabled after a HCLK clock. The disabled pin can become high-impedance state by setting the POC_ODMODE0/1 register.

By clearing the interrupt flag bit INnIF (n=0,1...5), the pins will be recovered from disabled output.

Figure 8-3. POC pins falling edge detection



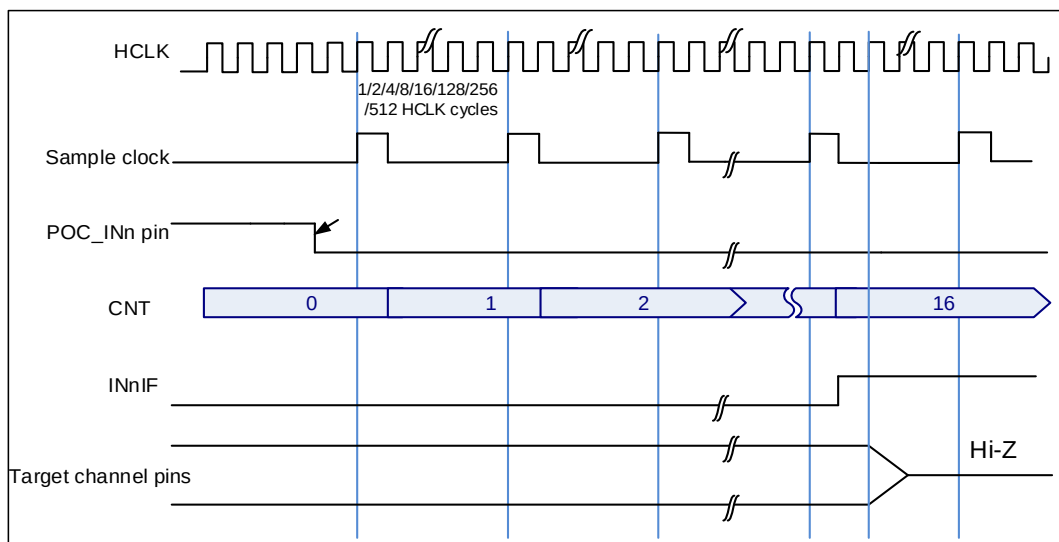
Level detection mode

An appropriate sampling clock frequency can be selected by setting the INDMSEL[3:0] bits in POC_INnDCFG (n=0,1... 5) registers to detect the input level of POC_INn (n=0,1... 5) pin. If the number of times that the low (INPL=0) or high (INPL= 1) level has been continuously sampled at the selected clock frequency reaches the number set by the INDSNUM[3:0] bits in POC_INnDCFG (n=0,1...5) registers, then the flag bit INnIF (n=0,1... 5) will be set by the hardware when the count reaches the specific number. Meanwhile, a disabling request will be generated by POC_INn (n=0,1...5) pin level detection.

Similarly, If the disabling request has been enabled and the target timer has been selected, after one HCLK clock, the output pins connected to the target timer channels are disabled and a high-impedance output can be selected.

Only when a high level signal (INPL=0) or low level signal (INPL=1) is input to POC_INn (n=0,1... 5) pin, writing 0 to INnIF (n=0,1... 5) flag can clear the flag bit. Otherwise, it is invalid that writing 0 to this flag bit. When the flag bit is cleared, the disabled pin can recover from the disabled output.

Figure 8-4. POC pins low-level detection (INDSNUM[3:0]=4'b0000)

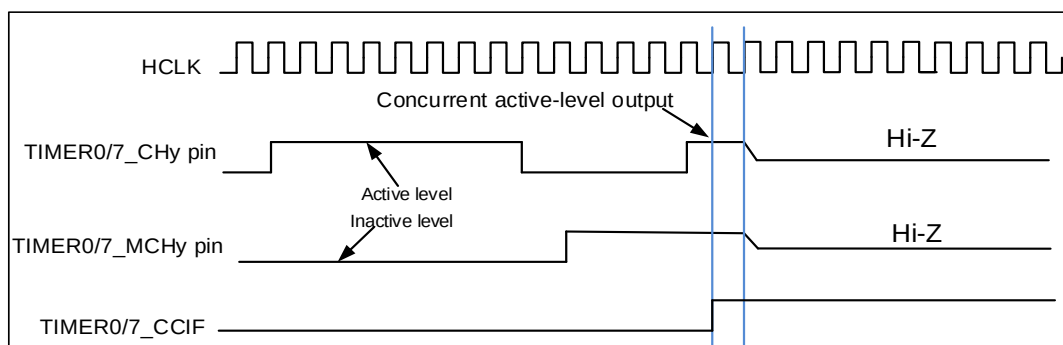


Complementary channel detection

When the TIMER0/7 complementary PWM channels (CH0/1/2 and MCH0/1/2) concurrently output the active level, the flag bit TIMER0/7_CCIF in POC_STAT0 register can be set to 1 by the hardware and a disabling request will be generated by TIMER0/7 concurrent conduction. If the disabling request function has been enabled on by setting TIMER0/7_CCDREN bit in POC_CDCFG0/1 register, the output pins connected to TIMER0/7 all complementary PWM channels (CH0/1/2/3 and MCH0/1/2/3) will be disabled and the disabled pins can be switched to Hi-Z output by setting TIMER0/7_OSELY [1:0] (y=0,1,2,3) bits in POC_ODMODE0 register.

The TIMER0/7_CCIF bit can be cleared by writing 0 to this bit, only after an inactive level output from the TIMER0/7 complementary channel. Otherwise, writing 0 to TIMER0/7_CCIF bit is invalid. After the flag is cleared, the pins will be recovered from disabled output.

Figure 8-5. Complementary PWM channel output detection



Note: The active level of both TIMERx_CHy and TIMERx_MCHy is high.

In addition, TIMER0/7_PSELEN bit in POC_CDCFG0/1 register determines that whether the active polarity of TIMER0/7 complementary PWM channels (CH0/1/2 and MCH0/1/2) is set by the POC related register or by the TIMER0/7 related register. The active polarity setting details can refer to description of TIMER0/7_PSELEN bit in POC_CDCFG0/1 register.

Table 8-2. Complementary PWM channel for output detection

CH	MCH	Interrupt flag
TIMER0_CH0	TIMER0_MCH0	TIMER0_CCIF
TIMER0_CH1	TIMER0_MCH1	
TIMER0_CH2	TIMER0_MCH2	
TIMER7_CH0	TIMER7_MCH0	TIMER7_CCIF
TIMER7_CH1	TIMER7_MCH1	
TIMER7_CH2	TIMER7_MCH2	

Output disable by software

Software can generate a disabling request by setting the GPTIMERx_SWDRG (x=0,1) or TIMERx_SWDRG (x=0,1,2,7) bit in POC_SWDRG register to 1. The disabling request

generated by the software has no request enable bit. Once the disabling request is generated, the pins connected to the corresponding TIMER channels will be disabled. The disabled pins can become Hi-Z state.

For instance, the output pins connected to TIMER1_CH0/1/2/3 can be disabled directly by setting the TIMER1_SWDRG bit to 1.

System fault detection

HXTAL stuck

When HXTAL stuck event occurs, the HXTALSDF bit in POC_STAT0 register can be set to 1 by hardware and a disabling request will be generated. If the disabling request is enabled by setting HXTALS_DREN bit in POC_CTL0 register, all output pins connected to channels in [Table 8-1. The target timers and channels for output disable](#) will be disabled.

CPU LOCKUP

When CPU LOCKUP event occurs, the LOCKUPDF bit in POC_STAT0 register can be set to 1 by hardware and a disabling request will be generated. If the disabling request is enabled by setting LOCKUP_DREN bit in POC_CTL0 register, all output pins connected to channels in [Table 8-1. The target timers and channels for output disable](#) will be disabled.

Comparator output detection

When CMPn (n=0,1,...,3) output rising edge, the detection flag bit CMPnDF (n=0,1,...,3) will be set to 1 by the hardware, and a disabling request will be generated by CMPn (n=0,1,...,3) output detection

There are two ways to enable the disabling request generated by CMPn (n=0,1,...,3) output detection. One way is to set the CMPnDREN (n=0,1,...,3) bit in POC_CTL1 register to enable the disabling request. In this way the disabling request switch can be turned on for output pins connected to all TIMERx (x=0,1,2,7) and GPTIMERx (x=0,1) channels. The other way is to set the TIMERx_CMPnDREN (x=0,1,2,7 n=0,1,...,3) bit or GPTIMERx_CMPnDREN (x=0,1 n=0,1,...,3) in POC_EXTCTL0/1 register to enable the disabling request. In this way, the disable request switch can be turned on for the output pins connected to the corresponding timer channels.

If the CMPn (n=0,1,...,3) disabling request has been enabled and the target timer has been selected, then, the output pins connected to the target timer channels will be disabled, and the disabled pins can be turned into a high impedance state.

By clearing the detection flag bit CMPnDF (n=0,1,...,3), the pins will be recovered from disabled output.

8.3.3 Target timer selection

The disabling request generated by POC_INn (n=0,1... 5) pin input detection and CMPn (n=0,1,...3) output detection can select a timer as disabled target by setting POC_REQSEL0/1 registers. When the disabling request arrives, the output pins connected with the selected target timer channels will be disabled.

It should be noted that some disabling requests have chosen the specific timer by default after a reset.

In addition, a disabling request can select different timers as the target timer, and different disabling requests can also select the same timer as the target timer.

Table 8-3. Disabling request selection for target timer

Disabling request source	Default selection	Non-default selection
POC_IN0	TIMER0	TIMER1, TIMER2, GPTIMER0, GPTIMER1, TIMER7
POC_IN1	TIMER7	TIMER0, TIMER1, TIMER2, GPTIMER0, GPTIMER1
POC_IN2	TIMER1, TIMER2, GPTIMER0, GPTIMER1	TIMER0, TIMER7
POC_IN3	No timer	TIMER0, TIMER1, TIMER2, GPTIMER0, GPTIMER1, TIMER7
POC_IN4	TIMER1	TIMER0, TIMER2, GPTIMER0, GPTIMER1, TIMER7
POC_IN5	TIMER2	TIMER0, TIMER1, GPTIMER0, GPTIMER1, TIMER7
CMPn (n=0,1,...,3)	No timer	TIMER0, TIMER1, TIMER2, GPTIMER0, GPTIMER1, TIMER7

8.3.4 Output switching

When an output pin is disabled, the pin can become high-impedance state if the TIMEx_OSELY [1:0] (x=0,1,2,7 y=0,1,2,3) or GPTIMERx_OSELY [1:0] (x=0,1 y=0,1,2,3) bits of the POC_ODMODE0/1 register is set to 2'b01.

The pin can also output the signal of the corresponding channel if the TIMEx_OSELY [1:0] (x=0,1,2,7 y=0,1,2,3) or GPTIMERx_OSELY [1:0] (x=0,1 y=0,1,2,3) bits is set to 2'b00.

If the TIMEx_OSELY [1:0] (x=0,1,2,7 y=0,1,2,3) or GPTIMERx_OSELY [1:0] (x=0,1 y=0,1,2,3) bits is set to 2'b10, the pin output state is controlled by GPIOx_OCTL (x=A,B,...,H) register in GPIO module.

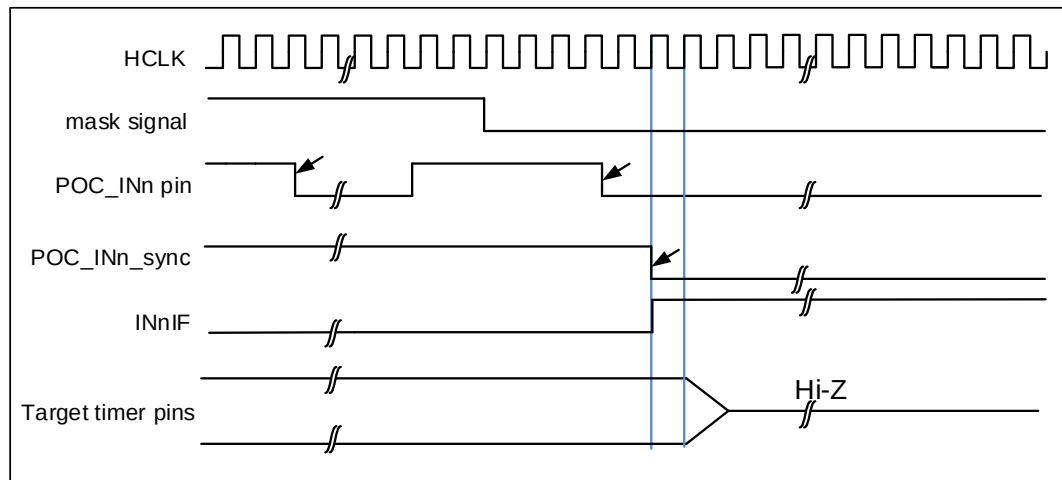
8.3.5 Detection masking function

Detection masking function can be used to mask POC_INn (n=0,1...5) pin input signals and CMPn (n=0,1,...,3) output detection signal.

If the INMKSEL[4:0] bits in the POC_INnDMK (n=0,1...5) register are not set to 4b'0000, when the timer channel signal selected as mask source outputs high level, POC_INn (n=0,1...5) pin input signals will be masked.

Similarly, when the timer channel specified by the CMPMKSEL[4:0] bits in the POC_CMPnDMK (n=0,1...3) register outputs high level, CMPn (n=0,1,...,4) output detection signal will be masked.

Figure 8-6. POC pin falling input detection masking



8.3.6 POC interrupt

The interrupt can be produced on one of the detection:

- POC_INn (n=0,1...5) pin input detection related to interrupt flag INnIF (n=0,1...5).
- TIMER0/7 complementary channels output detection related to interrupt flag TIMER0/7_CCIF.

All above interrupts are mapped into the same interrupt vector.

8.4 Register definition

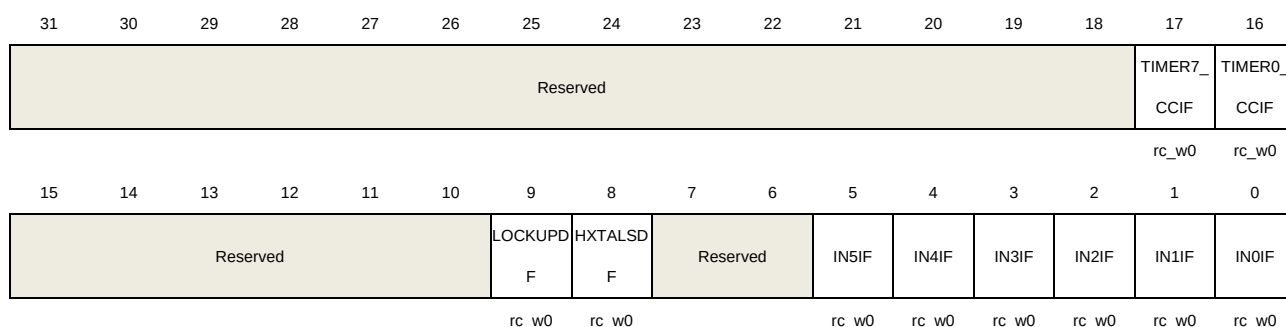
POC base address: 0x4800 4400

8.4.1 Status register 0 (POC_STAT0)

Address offset: 0x00

Reset value: 0x0000 0000

This register has to be accessed by byte(8-bit), half-word(16-bit) and word(32-bit).



Bits	Fields	Descriptions
31:18	Reserved	Must be kept at reset value.
17	TIMER7_CCIF	TIMER7 concurrent conduction interrupt flag This bit shows that whether a disabling request has been generated by the concurrent conduction detection of TIMER7 complementary channels. 0: A disabling request has not been generated 1: A disabling request has been generated Set by hardware when at least one pair of the TIMER7 complementary channels (TIMER7_CH0/1/2 and TIMER7_MCH0/1/2) concurrently output active level. Cleared by software writing 0 to it.
16	TIMER0_CCIF	TIMER0 concurrent conduction interrupt flag This bit shows that whether a disabling request has been generated by the concurrent conduction detection of TIMER0 complementary channels. 0: A disabling request has not been generated 1: A disabling request has been generated Set by hardware when at least one pair of the TIMER0 complementary channels (TIMER0_CH0/1/2 and TIMER0_MCH0/1/2) concurrently output active level. Cleared by software writing 0 to it.
15:10	Reserved	Must be kept at reset value.
9	LOCKUPDF	CPU LOCKUP detection flag This bit shows whether a disabling request has been generated by CPU LOCKUP event detection.

		0: A disabling request has not been generated 1: A disabling request has been generated Set by hardware when the CPU LOCKUP event occurs. Cleared by software writing 0 to it.
8	HXTALSDF	HXTAL stuck detection flag This bit is used to show whether a disabling request has been generated by HXTAL stuck event detection. 0: A disabling request has not been generated 1: A disabling request has been generated Set by hardware when the HXTAL stuck event occurs. Cleared by software writing 0 to it.
7:6	Reserved	Must be kept at reset value.
5	IN5IF	POC_IN5 input interrupt flag This bit shows that whether a disabling request has been generated by POC_IN5 pin input detection. 0: A disabling request has not been generated 1: A disabling request has been generated Set by hardware when the valid signal selected by INDMSEL[3:0] bits of POC_IN5DCFG register inputs POC_IN5 pin. Cleared by software writing 0 to it.
4	IN4IF	POC_IN4 input interrupt flag This bit shows that whether a disabling request has been generated by POC_IN4 pin input detection. 0: A disabling request has not been generated 1: A disabling request has been generated Set by hardware when the valid signal selected by INDMSEL[3:0] bits of POC_IN4DCFG register inputs POC_IN4 pin. Cleared by software writing 0 to it.
3	IN3IF	POC_IN3 input interrupt flag This bit shows that whether a disabling request has been generated by POC_IN3 pin input detection. 0: A disabling request has not been generated 1: A disabling request has been generated Set by hardware when the valid signal selected by INDMSEL[3:0] bits of POC_IN3DCFG register inputs POC_IN3 pin. Cleared by software writing 0 to it.
2	IN2IF	POC_IN2 input interrupt flag This bit shows that whether a disabling request has been generated by POC_IN2 pin input detection. 0: A disabling request has not been generated

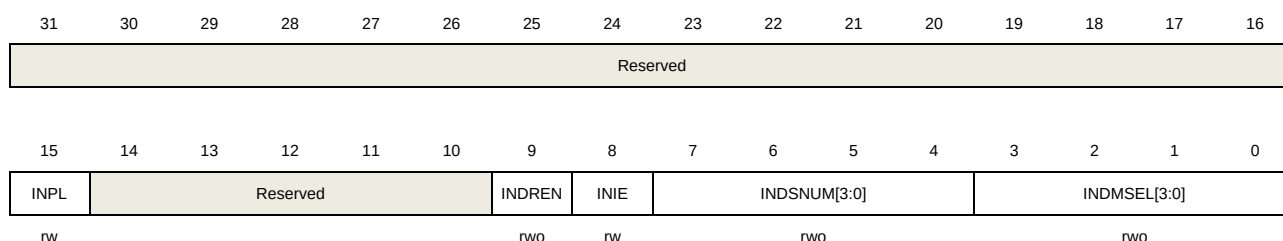
		1: A disabling request has been generated Set by hardware when the valid signal selected by INDMSEL[3:0] bits of POC_IN2DCFG register inputs POC_IN2 pin. Cleared by software writing 0 to it.
1	IN1IF	POC_IN1 input interrupt flag This bit shows that whether a disabling request has been generated by POC_IN1 pin input detection. 0: A disabling request has not been generated 1: A disabling request has been generated Set by hardware when the valid signal selected by INDMSEL[3:0] bits of POC_IN1DCFG register inputs POC_IN1 pin. Cleared by software writing 0 to it.
0	IN0IF	POC_IN0 input interrupt flag This bit shows that whether a disabling request has been generated by POC_IN0 pin input detection. 0: A disabling request has not been generated 1: A disabling request has been generated Set by hardware when the valid signal selected by INDMSEL[3:0] bits of POC_IN0DCFG register inputs POC_IN0 pin. Cleared by software writing 0 to it.

8.4.2 Input n detection configuration register (POC_INnDCFG) (n=0...5)

Address offset: $0x04 + 0x04 * n$

Reset value: 0x0000 0200 for 0x04/0x08, 0x0000 0000 for others

This register has to be accessed by byte(8-bit), half-word(16-bit) and word(32-bit).



Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value
15	INPL	Polarity of POC_INn pin input This bit is used to select the polarity of POC_INn pin input. 0: Input is not inverted 1: Input is inverted
14:10	Reserved	Must be kept at reset value

9	INDREN	POC_INn disabling request enable This bit is used to determine whether the disabling request generated by POC_INn pin input detection is enabled for TIMERx (x=0,1,2,7) and GPTIMERx (x=0,1). 0: The disabling request is disabled 1: The disabling request is enabled If enabled, TIMERx (x=0,1,2,7) channel pin outputs can be disabled when INnIF = 1 and TIMERx_INnDRSEL = 1. If enabled, GPTIMERx (x=0,1) channel pin outputs can be disabled when INnIF = 1 and GPTIMERx_INnDRSEL = 1. Note: This bit-field can be written only once after the reset.
8	INIE	POC_INn input interrupt enable This bit is used to enable or disable interrupt when INnIF = 1. 0: Interrupt disable 1: Interrupt enable
7:4	INDSNUM[3:0]	POC_INn input detection sampling number These bits are used to select the valid sampling number of the POC_INn input detection. 0000: 16 times 0001: 4 times 0010: 8 times 0011: 9 times 0100: 10 times 0101: 11 times 0110: 12 times 0111: 13 times 1000: 14 times 1001: 15 times All other values: reserved. Note: This bit-field can be written only once after the reset.
3:0	INDMSEL[3:0]	POC_INn input detection mode selection These bits are used to select the valid signal and sampling frequency of the POC_INn input detection to generate a disabling request. 0000: Sampling frequency is f_{HCLK} and generate a request on the falling edge (INPL=0) or rising edge (INPL=1). 0001: Sampling frequency is $f_{HCLK}/8$ and generate a request when the low level (INPL=0) or high level (INPL=1) is continuously sampled a certain number of times. 0010: Sampling frequency is $f_{HCLK}/16$ and generate a request when the low level (INPL=0) or high level (INPL=1) is continuously sampled a certain number of times. 0011: Sampling frequency is $f_{HCLK}/128$ and generate a request when the low level (INPL=0) or high level (INPL=1) is continuously sampled a certain number of times. 0100: Sampling frequency is f_{HCLK} and generate a request when the low level (INPL=0) or high level (INPL=1) is continuously sampled a certain number of times.

0101: Sampling frequency is $f_{HCLK}/2$ and generate a request when the low level (INPL=0) or high level (INPL=1) is continuously sampled a certain number of times.

0110: Sampling frequency is $f_{HCLK}/4$ and generate a request when the low level (INPL=0) or high level (INPL=1) is continuously sampled a certain number of times.

0111: Sampling frequency is $f_{HCLK}/256$ and generate a request when the low level (INPL=0) or high level (INPL=1) is continuously sampled a certain number of times.

1000: Sampling frequency is $f_{HCLK}/512$ and generate a request when the low level (INPL=0) or high level (INPL=1) is continuously sampled a certain number of times.

1001: Sampling frequency is $f_{HCLK}/32$ and generate a request when the low level (INPL=0) or high level (INPL=1) is continuously sampled a certain number of times.

1010: Sampling frequency is $f_{HCLK}/64$ and generate a request when the low level (INPL=0) or high level (INPL=1) is continuously sampled a certain number of times.

1011: Sampling frequency is $f_{HCLK}/2$ and generate a request on the falling edge (INPL=0) or rising edge (INPL=1).

1100: Sampling frequency is $f_{HCLK}/4$ and generate a request on the falling edge (INPL=0) or rising edge (INPL=1).

All other values: reserved.

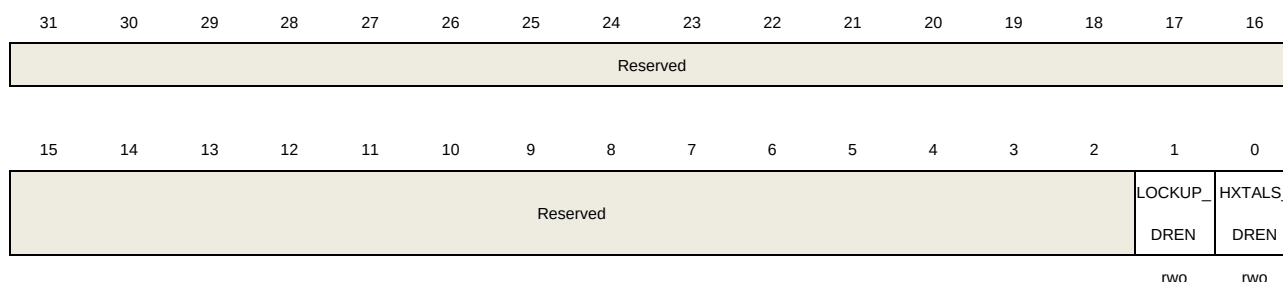
Note: This bit-field can be written only once after the reset.

8.4.3 Control register 0 (POC_CTL0)

Address offset: 0x20

Reset value: 0x0000 0000

This register has to be accessed by byte(8-bit), half-word(16-bit) and word(32-bit).



Bits	Fields	Descriptions
31:2	Reserved	Must be kept at reset value
1	LOCKUP_DREN	CPU LOCKUP disabling request enable This bit is used to determine whether the disabling request generated by CPU LOCKUP event is enabled. 0: The disabling request is disabled 1: The disabling request is enabled If enabled, all POC-related timers channel pin outputs can be disabled when LOCKUPDF = 1. Note: This bit-field can be written only once after the reset.

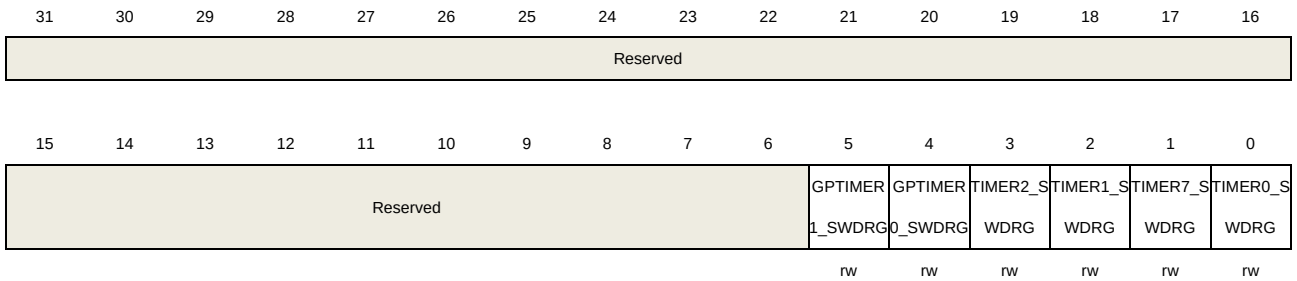
0	HXTALS_DREN	HXTAL stuck disabling request enable This bit is used to determine whether the disabling request generated by HXTAL stuck event is enabled. 0: The disabling request is disabled 1: The disabling request is enabled If enabled, all POC-related timers channel pin outputs can be disabled when HXTALSDF = 1. Note: This bit-field can be written only once after the reset.

8.4.4 Software disabling request generation register (POC_SWDRG)

Address offset: 0x24

Reset value: 0x0000 0000

This register has to be accessed by byte(8-bit), half-word(16-bit) and word(32-bit).



Bits	Fields	Descriptions
31:6	Reserved	Must be kept at reset value.
5	GPTIMER1_SWDRG	Software disabling request generation for GPTIMER1 This bit is used to generate the disabling request by software to disable the GPTIMER1 channel pin outputs. 0: Software does not generate the disabling request 1: Software generate the disabling request
4	GPTIMER0_SWDRG	Software disabling request generation for GPTIMER0 This bit is used to generate the disabling request by software to disable the GPTIMER0 channel pin outputs. 0: Software does not generate the disabling request 1: Software generate the disabling request
3	TIMER2_SWDRG	Software disabling request generation for TIMER2 This bit is used to generate the disabling request by software to disable the TIMER2 channel pin outputs. 0: Software does not generate the disabling request 1: Software generate the disabling request
2	TIMER1_SWDRG	Software disabling request generation for TIMER1 This bit is used to generate the disabling request by software to disable the TIMER1

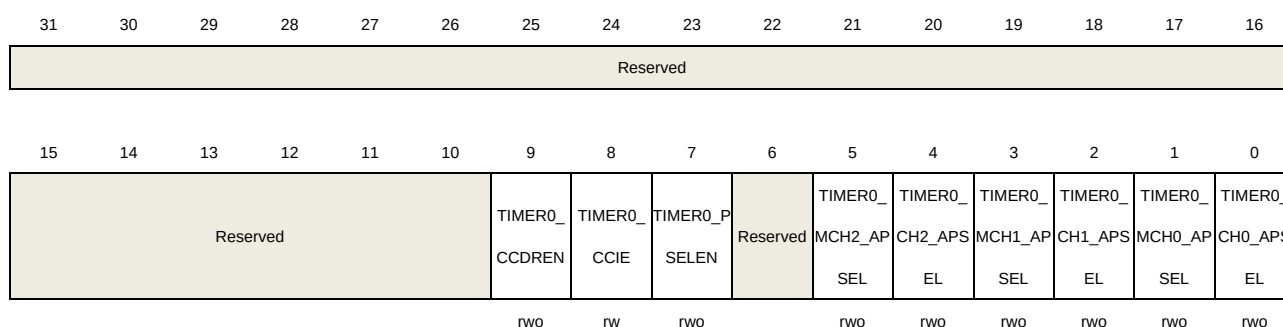
		channel pin outputs. 0: Software does not generate the disabling request 1: Software generate the disabling request
1	TIMER7_SWDRG	Software disabling request generation for TIMER7 This bit is used to generate the disabling request by software to disable the TIMER7 channel pin outputs. 0: Software does not generate the disabling request 1: Software generate the disabling request
0	TIMER0_SWDRG	Software disabling request generation for TIMER0 This bit is used to generate the disabling request by software to disable the TIMER0 channel pin outputs. 0: Software does not generate the disabling request 1: Software generate the disabling request

8.4.5 Complementary detection configuration register 0 (POC_CDCFG0)

Address offset: 0x28

Reset value: 0x0000 0000

This register has to be accessed by byte(8-bit), half-word(16-bit) and word(32-bit).



Bits	Fields	Descriptions
31:10	Reserved	Must be kept at reset value.
9	TIMER0_CCDREN	TIMER0 concurrent conduction disabling request enable This bit is used to determine whether the disabling request generated by TIMER0 concurrent conduction detection is enabled. 0: The disabling request is disabled 1: The disabling request is enabled If enabled, TIMER0 complementary PWM channel pin outputs are disabled when TIMER0_CCIF = 1 Note: This bit-field can be written only once after the reset.
8	TIMER0_CCIE	TIMER0 concurrent conduction interrupt enable This bit is used to enable or disable interrupt when TIMER0_CCIF = 1. 0: Interrupt disable

		1: Interrupt enable
7	TIMER0_PSELEN	TIMER0 polarity selection enable This bit is used to enable the POC_CDCFG0 register to control the TIMER0 channel active polarity selection. 0: Disable 1: Enable If disabled, CHxP (x=0,1,2) and MCHxP (x=0,1,2) in TIMER0_CHCTL2 register specify active polarity for the TIMER0_CH0/1/2 and TIMER0_MCH0/1/2 output. If enabled, TIMER0_CHx_APSEL (x=0,1,2) and TIMER0_MCHx_APSEL (x=0,1,2) specify active polarity for the TIMER0_CH0/1/2 and TIMER0_MCH0/1/2 output. Note: This bit-field can be written only once after the reset.
6	Reserved	Must be kept at reset value.
5	TIMER0_MCH2_APS EL	TIMER0_MCH2 active polarity selection This bit is used to select active polarity for TIMER0_MCH2. 0: High level is active level 1: Low level is active level Note: This bit-field can be written only once after the reset.
4	TIMER0_CH2_APSE L	TIMER0_CH2 active polarity selection This bit is used to select active polarity for TIMER0_CH2. 0: High level is active level 1: Low level is active level Note: This bit-field can be written only once after the reset.
3	TIMER0_MCH1_APS EL	TIMER0_MCH1 active polarity selection This bit is used to select active polarity for TIMER0_MCH1. 0: High level is active level 1: Low level is active level Note: This bit-field can be written only once after the reset.
2	TIMER0_CH1_APSE L	TIMER0_CH1 active polarity selection This bit is used to select active polarity for TIMER0_CH1. 0: High level is active level 1: Low level is active level Note: This bit-field can be written only once after the reset.
1	TIMER0_MCH0_APS EL	TIMER0_MCH0 active polarity selection This bit is used to select active polarity for TIMER0_MCH0. 0: High level is active level 1: Low level is active level Note: This bit-field can be written only once after the reset.
0	TIMER0_CH0_APSE L	TIMER0_CH0 active polarity selection This bit is used to select active polarity for TIMER0_CH0.

0: High level is active level

1: Low level is active level

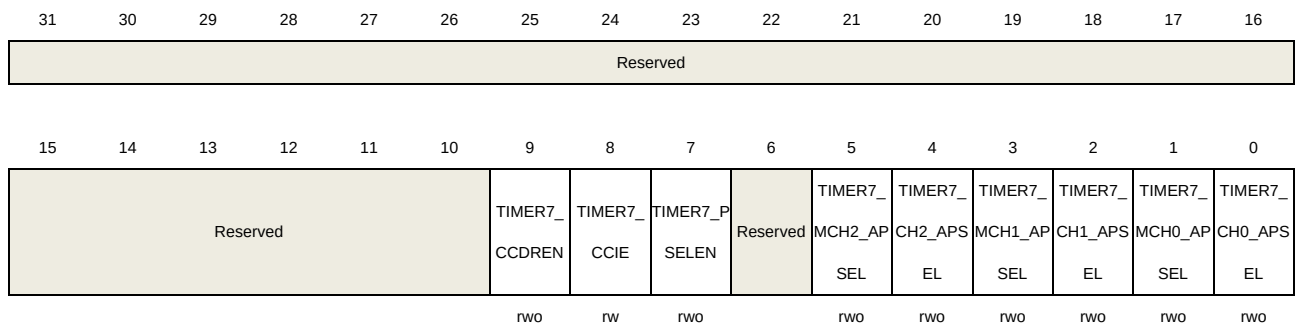
Note: This bit-field can be written only once after the reset.

8.4.6 Complementary detection configuration register 1 (POC_CDCFG1)

Address offset: 0x2C

Reset value: 0x0000 0000

This register has to be accessed by byte(8-bit), half-word(16-bit) and word(32-bit).



Bits	Fields	Descriptions
31:10	Reserved	Must be kept at reset value.
9	TIMER7_CCDREN	TIMER7 concurrent conduction disabling request enable This bit is used to determine whether the disabling request generated by TIMER7 concurrent conduction detection is enabled. 0: The disabling request is disabled 1: The disabling request is enabled If enabled, TIMER7 complementary PWM channel pin outputs are disabled when $TIMER7\_CCIF = 1$ Note: This bit-field can be written only once after the reset.
8	TIMER7_CCIE	TIMER7 concurrent conduction interrupt enable This bit is used to enable or disable interrupt when $TIMER7\_CCIF = 1$. 0: Interrupt disable 1: Interrupt enable
7	TIMER7_PSELEN	TIMER7 polarity selection enable This bit is used to enable the POC_CDCFG1 register to control the TIMER7 active polarity selection. 0: Disable 1: Enable If disabled, CHxP (x=0,1,2) and MCHxP (x=0,1,2) in TIMER7_CHCTL2 register specify active polarity for the TIMER7_CH0/1/2 and TIMER7_MCH0/1/2 output. If enabled, TIMER7_CHx_APSEL (x=0,1,2) and TIMER7_MCHx_APSEL (x=0,1,2) specify active polarity for the TIMER7_CH0/1/2 and TIMER7_MCH0/1/2 output.

Note: This bit-field can be written only once after the reset.

6	Reserved	Must be kept at reset value.
5	TIMER7_MCH2_APS EL	TIMER7_MCH2 active polarity selection This bit is used to select active polarity for TIMER7_MCH2. 0: High level is active level 1: Low level is active level Note: This bit-field can be written only once after the reset.
4	TIMER7_CH2_APSE L	TIMER7_CH2 active polarity selection This bit is used to select active polarity for TIMER7_CH2. 0: High level is active level 1: Low level is active level Note: This bit-field can be written only once after the reset.
3	TIMER7_MCH1_APS EL	TIMER7_MCH1 active polarity selection This bit is used to select active polarity for TIMER7_MCH1. 0: High level is active level 1: Low level is active level Note: This bit-field can be written only once after the reset.
2	TIMER7_CH1_APSE L	TIMER7_CH1 active polarity selection This bit is used to select active polarity for TIMER7_CH1. 0: High level is active level 1: Low level is active level Note: This bit-field can be written only once after the reset.
1	TIMER7_MCH0_APS EL	TIMER7_MCH0 active polarity selection This bit is used to select active polarity for TIMER7_MCH0. 0: High level is active level 1: Low level is active level Note: This bit-field can be written only once after the reset.
0	TIMER7_CH0_APSE L	TIMER7_CH0 active polarity selection This bit is used to select active polarity for TIMER7_CH0. 0: High level is active level 1: Low level is active level Note: This bit-field can be written only once after the reset.

8.4.7 Output disable mode register 0 (POC_ODMODE0)

Address offset: 0x40

Reset value: 0x0000 1515

This register has to be accessed by byte(8-bit), half-word(16-bit) and word(32-bit).

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

Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIMER7_OSEL3[1:0]	TIMER7_OSEL2[1:0]	TIMER7_OSEL1[1:0]	TIMER7_OSEL0[1:0]	TIMER0_OSEL3[1:0]	TIMER0_OSEL2[1:0]	TIMER0_OSEL1[1:0]	TIMER0_OSEL0[1:0]								
rwo	rwo	rwo	rwo	rwo	rwo	rwo	rwo								

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:14	TIMER7_OSEL3[1:0]	Output selection for TIMER7_CH3/ TIMER7_MCH3 pin These bits determine whether to select Hi-Z output or GPIO output for TIMER7_CH3/ TIMER7_MCH3 pin when the pins are disabled. 00: TIMER7_CH3/TIMER7_MCH3 output 01: Hi-Z output 10: GPIO output 11: Reserved Note: This bit-field can be written only once after the reset.
13:12	TIMER7_OSEL2[1:0]	Output selection for TIMER7_CH2/ TIMER7_MCH2 pin These bits determine whether to select Hi-Z output or GPIO output for TIMER7_CH2/ TIMER7_MCH2 pin when the pins are disabled. 00: TIMER7_CH2/TIMER7_MCH2 output 01: Hi-Z output 10: GPIO output 11: Reserved Note: This bit-field can be written only once after the reset. The default value is 2b'01 after the reset.
11:10	TIMER7_OSEL1[1:0]	Output selection for TIMER7_CH1/ TIMER7_MCH1 pin These bits determine whether to select Hi-Z output or GPIO output for TIMER7_CH1/ TIMER7_MCH1 pin when the pins are disabled. 00: TIMER7_CH1/TIMER7_MCH1 output 01: High-impedance output 10: GPIO output 11: Reserved Note: This bit-field can be written only once after the reset. The default value is 2b'01 after the reset.
9:8	TIMER7_OSEL0[1:0]	Output selection for TIMER7_CH0/ TIMER7_MCH0 pin These bits determine whether to select Hi-Z output or GPIO output for TIMER7_CH0/ TIMER7_MCH0 pin when the pins are disabled. 00: TIMER7_CH0/TIMER7_MCH0 output 01: Hi-Z output 10: GPIO output 11: Reserved

Note: This bit-field can be written only once after the reset. The default value is 2b'01 after the reset.

- 7:6 TIMER0_OSEL3[1:0] Output selection for TIMER0_CH3/ TIMER0_MCH3 pin
These bits determine whether to select Hi-Z output or GPIO output for
TIMER0_CH3/ TIMER0_MCH3 pin when the pins are disabled.
00: TIMER0_CH3/ TIMER0_MCH3 output
01: Hi-Z output
10: GPIO output
11: Reserved
Note: This bit-field can be written only once after the reset.
- 5:4 TIMER0_OSEL2[1:0] Output selection for TIMER0_CH2/ TIMER0_MCH2 pin
These bits determine whether to select Hi-Z output or GPIO output for
TIMER0_CH2/ TIMER0_MCH2 pin when the pins are disabled.
00: TIMER0_CH2/TIMER0_MCH2 output
01: Hi-Z output
10: GPIO output
11: Reserved
Note: This bit-field can be written only once after the reset. The default value is
2b'01 after the reset.
- 3:2 TIMER0_OSEL1[1:0] Output selection for TIMER0_CH1/ TIMER0_MCH1 pin
These bits determine whether to select Hi-Z output or GPIO output for
TIMER0_CH1/ TIMER0_MCH1 pin when the pins are disabled.
00: TIMER0_CH1/TIMER0_MCH1 output
01: Hi-Z output
10: GPIO output
11: Reserved
Note: This bit-field can be written only once after the reset. The default value is
2b'01 after the reset.
- 1:0 TIMER0_OSEL0[1:0] Output selection for TIMER0_CH0/ TIMER0_MCH0 pin
These bits determine whether to select Hi-Z output or GPIO output for
TIMER0_CH0/ TIMER0_MCH0 pin when the pins are disabled.
00: TIMER0_CH0/TIMER0_MCH0 output
01: Hi-Z output
10: GPIO output
11: Reserved
Note: This bit-field can be written only once after the reset. The default value is
2b'01 after the reset.

8.4.8 Output disable mode register 1 (POC_ODMODE1)

Address offset: 0x44

Reset value: 0x0000 0000

This register has to be accessed by byte(8-bit), half-word(16-bit) and word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				GPTIMER1_OSEL1[1:0]		GPTIMER1_OSEL0[1:0]		Reserved				GPTIMER0_OSEL1[1:0]		GPTIMER0_OSEL0[1:0]	
rwo				rwo		rwo				rwo		rwo		rwo	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIMER2_OSEL3[1:0]		TIMER2_OSEL2[1:0]		TIMER2_OSEL1[1:0]		TIMER2_OSEL0[1:0]		TIMER1_OSEL3[1:0]		TIMER1_OSEL2[1:0]		TIMER1_OSEL1[1:0]		TIMER1_OSEL0[1:0]	
rwo		rwo		rwo		rwo		rwo		rwo		rwo		rwo	

Bits	Fields	Descriptions
31:28	Reserved	Must be kept at reset value.
27:26	GPTIMER1_OSEL1[1:0]	Output selection for GPTIMER1_CH1 pin These bits determine whether to select Hi-Z output or GPIO output for GPTIMER1_CH1 pin when the pin is disabled. 00: GPTIMER1_CH1 output 01: Hi-Z output 10: GPIO output 11: Reserved Note: This bit-field can be written only once after the reset.
25:24	GPTIMER1_OSEL0[1:0]	Output selection for GPTIMER1_CH0 pin These bits determine whether to select Hi-Z output or GPIO output for GPTIMER1_CH0 pin when the pin is disabled. 00: GPTIMER1_CH0 output 01: Hi-Z output 10: GPIO output 11: Reserved Note: This bit-field can be written only once after the reset.
23:20	Reserved	Must be kept at reset value.
19:18	GPTIMER0_OSEL1[1:0]	Output selection for GPTIMER0_CH1 pin These bits determine whether to select Hi-Z output or GPIO output for GPTIMER0_CH1 pin when the pin is disabled. 00: GPTIMER0_CH1 output 01: Hi-Z output 10: GPIO output 11: Reserved Note: This bit-field can be written only once after the reset.
17:16	GPTIMER0_OSEL0[1:0]	Output selection for GPTIMER0_CH0 pin These bits determine whether to select Hi-Z output or GPIO output for GPTIMER0_CH0 pin when the pin is disabled.

		00: GPTIMER0_CH0 output
		01: Hi-Z output
		10: GPIO output
		11: Reserved
		Note: This bit-field can be written only once after the reset.
15:14	TIMER2_OSEL3[1:0]	Output selection for TIMER2_CH3 pin
		These bits determine whether to select Hi-Z output or GPIO output for TIMER2_CH3 pin when the pin is disabled.
		00: TIMER2_CH3 output
		01: Hi-Z output
		10: GPIO output
		11: Reserved
		Note: This bit-field can be written only once after the reset.
13:12	TIMER2_OSEL2[1:0]	Output selection for TIMER2_CH2 pin
		These bits determine whether to select Hi-Z output or GPIO output for TIMER2_CH2 pin when the pin is disabled.
		00: TIMER2_CH2 output
		01: Hi-Z output
		10: GPIO output
		11: Reserved
		Note: This bit-field can be written only once after the reset.
11:10	TIMER2_OSEL1[1:0]	Output selection for TIMER2_CH1 pin
		These bits determine whether to select Hi-Z output or GPIO output for TIMER2_CH1 pin when the pin is disabled.
		00: TIMER2_CH1 output
		01: Hi-Z output
		10: GPIO output
		11: Reserved
		Note: This bit-field can be written only once after the reset.
9:8	TIMER2_OSEL0[1:0]	Output selection for TIMER2_CH0 pin
		These bits determine whether to select Hi-Z output or GPIO output for TIMER2_CH0 pin when the pin is disabled.
		00: TIMER2_CH0 output
		01: Hi-Z output
		10: GPIO output
		11: Reserved
		Note: This bit-field can be written only once after the reset.
7:6	TIMER1_OSEL3[1:0]	Output selection for TIMER1_CH3 pin
		These bits determine whether to select Hi-Z output or GPIO output for TIMER1_CH3 pin when the pin is disabled.
		00: TIMER1_CH3 output

01: Hi-Z output

10: GPIO output

11: Reserved

Note: This bit-field can be written only once after the reset.

5:4 TIMER1_OSEL2[1:0] Output selection for TIMER1_CH2 pin

These bits determine whether to select Hi-Z output or GPIO output for TIMER1_CH2 pin when the pin is disabled.

00: TIMER1_CH2 output

01: Hi-Z output

10: GPIO output

11: Reserved

Note: This bit-field can be written only once after the reset.

3:2 TIMER1_OSEL1[1:0] Output selection for TIMER1_CH1 pin

These bits determine whether to select Hi-Z output or GPIO output for TIMER1_CH1 pin when the pin is disabled.

00: TIMER1_CH1 output

01: Hi-Z output

10: GPIO output

11: Reserved

Note: This bit-field can be written only once after the reset.

1:0 TIMER1_OSEL0[1:0] Output selection for TIMER1_CH0 pin

These bits determine whether to select Hi-Z output or GPIO output for TIMER1_CH0 pin when the pin is disabled.

00: TIMER1_CH0 output

01: Hi-Z output

10: GPIO output

11: Reserved

Note: This bit-field can be written only once after the reset.

8.4.9 Request selection register 0 (POC_REQSEL0)

Address offset: 0x48

Reset value: 0x0000 0201

This register has to be accessed by byte(8-bit), half-word(16-bit) and word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	TIMER7_	TIMER7_I	TIMER7_I	TIMER7_I	TIMER7_I	TIMER7_I	TIMER7_I	Reserved	TIMER0_	TIMER0_I	TIMER0_I	TIMER0_I	TIMER0_I	TIMER0_I	TIMER0_I
CMPDRS	N5DRSEL	N4DRSEL	N3DRSEL	N2DRSEL	N1DRSEL	N0DRSEL	Reserved	CMPDRS	N5DRSEL	N4DRSEL	N3DRSEL	N2DRSEL	N1DRSEL	N0DRSEL	Reserved
EL								EL							

rWO rWO rWO rWO rWO rWO rWO rWO rWO rWO rWO rWO rWO rWO rWO

Bits	Fields	Descriptions
31:15	Reserved	Must be kept at reset value
14	TIMER7_CMPDRSEL	CMPn (n=0,1,2,3) disabling request select TIMER7
	L	This bit is used to determine whether the disabling request generated by CMPn (n=0,1,2,3) output detection selects TIMER7 as a target timer. 0: TIMER7 is not selected 1: TIMER7 is selected When CMPn generates a disabling request (CMPnDF is set to 1 by hardware), if this bit is set to 1, and CMPnDREN or TIMER7_CMPnDREN also is set to 1, the output pins connected with TIMER7 channels will be disabled. Note: This bit-field can be written only once after the reset.
13	TIMER7_IN5DRSEL	POC_IN5 disabling request select TIMER7
		This bit is used to determine whether the disabling request generated by POC_IN5 pin input detection selects TIMER7 as a target timer. 0: TIMER7 is not selected 1: TIMER7 is selected When POC_IN5 generates a disabling request (IN5IF is set to 1 by hardware), if this bit is set to 1, and INDREN bit in POC_IN5DCFG register also is set to 1, the output pins connected with TIMER7 channels will be disabled. Note: This bit-field can be written only once after the reset.
12	TIMER7_IN4DRSEL	POC_IN4 disabling request select TIMER7
		This bit is used to determine whether the disabling request generated by POC_IN4 pin input detection selects TIMER7 as a target timer. 0: TIMER7 is not selected 1: TIMER7 is selected When POC_IN4 generates a disabling request (IN4IF is set to 1 by hardware), if this bit is set to 1, and INDREN bit in POC_IN4DCFG register also is set to 1, the output pins connected with TIMER7 channels will be disabled. Note: This bit-field can be written only once after the reset.
11	TIMER7_IN3DRSEL	POC_IN3 disabling request select TIMER7
		This bit is used to determine whether the disabling request generated by POC_IN3 pin input detection selects TIMER7 as a target timer. 0: TIMER7 is not selected 1: TIMER7 is selected When POC_IN3 generates a disabling request (IN3IF is set to 1 by hardware), if this bit is set to 1, and INDREN bit in POC_IN3DCFG register also is set to 1, the output pins connected with TIMER7 channels will be disabled. Note: This bit-field can be written only once after the reset.
10	TIMER7_IN2DRSEL	POC_IN2 disabling request select TIMER7

		This bit is used to determine whether the disabling request generated by POC_IN2 pin input detection selects TIMER7 as a target timer. 0: TIMER7 is not selected 1: TIMER7 is selected When POC_IN2 generates a disabling request (IN2IF is set to 1 by hardware), if this bit is set to 1, and INDREN bit in POC_IN2DCFG register also is set to 1, the output pins connected with TIMER7 channels will be disabled. Note: This bit-field can be written only once after the reset.
9	TIMER7_IN1DRSEL	POC_IN1 disabling request select TIMER7 This bit is used to determine whether the disabling request generated by POC_IN1 pin input detection selects TIMER7 as a target timer. 0: TIMER7 is not selected 1: TIMER7 is selected When POC_IN1 generates a disabling request (IN1IF is set to 1 by hardware), if this bit is set to 1, and INDREN bit in POC_IN1DCFG register also is set to 1, the output pins connected with TIMER7 channels will be disabled. Note: This bit-field can be written only once after the reset. The default value is 1 after the reset.
8	TIMER7_IN0DRSEL	POC_IN0 disabling request select TIMER7 This bit is used to determine whether the disabling request generated by POC_IN0 pin input detection selects TIMER7 as a target timer. 0: TIMER7 is not selected 1: TIMER7 is selected When POC_IN0 generates a disabling request (IN0IF is set to 1 by hardware), if this bit is set to 1, and INDREN bit in POC_IN0DCFG register also is set to 1, the output pins connected with TIMER7 channels will be disabled. Note: This bit-field can be written only once after the reset.
7	Reserved	Must be kept at reset value
6	TIMER0_CMPDRSEL	CMPn (n=0,1,2,3) disabling request select TIMER0 This bit is used to determine whether the disabling request generated by CMPn (n=0,1,2,3) output detection selects TIMER0 as a target timer. 0: TIMER0 is not selected 1: TIMER0 is selected When CMPn generates a disabling request (CMPnDF is set to 1 by hardware), if this bit is set to 1, and CMPnDREN or TIMER0_CMPnDREN also is set to 1, the output pins connected with TIMER0 channels will be disabled. Note: This bit-field can be written only once after the reset.
5	TIMER0_IN5DRSEL	POC_IN5 disabling request select TIMER0 This bit is used to determine whether the disabling request generated by POC_IN5 pin input detection selects TIMER0 as a target timer. 0: TIMER0 is not selected

- 1: TIMER0 is selected
When POC_IN5 generates a disabling request (IN5IF is set to 1 by hardware), if this bit is set to 1, and INDREN bit in POC_IN5DCFG register also is set to 1, the output pins connected with TIMER0 channels will be disabled.
Note: This bit-field can be written only once after the reset.
- 4 TIMER0_IN4DRSEL POC_IN4 disabling request select TIMER0
This bit is used to determine whether the disabling request generated by POC_IN4 pin input detection selects TIMER0 as a target timer.
0: TIMER0 is not selected
1: TIMER0 is selected
When POC_IN4 generates a disabling request (IN4IF is set to 1 by hardware), if this bit is set to 1, and INDREN bit in POC_IN4DCFG register also is set to 1, the output pins connected with TIMER0 channels will be disabled.
Note: This bit-field can be written only once after the reset.
- 3 TIMER0_IN3DRSEL POC_IN3 disabling request select TIMER0
This bit is used to determine whether the disabling request generated by POC_IN3 pin input detection selects TIMER0 as a target timer.
0: TIMER0 is not selected
1: TIMER0 is selected
When POC_IN3 generates a disabling request (IN3IF is set to 1 by hardware), if this bit is set to 1, and INDREN bit in POC_IN3DCFG register also is set to 1, the output pins connected with TIMER0 channels will be disabled.
Note: This bit-field can be written only once after the reset.
- 2 TIMER0_IN2DRSEL POC_IN2 disabling request select TIMER0
This bit is used to determine whether the disabling request generated by POC_IN2 pin input detection selects TIMER0 as a target timer.
0: TIMER0 is not selected
1: TIMER0 is selected
When POC_IN2 generates a disabling request (IN2IF is set to 1 by hardware), if this bit is set to 1, and INDREN bit in POC_IN2DCFG register also is set to 1, the output pins connected with TIMER0 channels will be disabled.
Note: This bit-field can be written only once after the reset.
- 1 TIMER0_IN1DRSEL POC_IN1 disabling request select TIMER0
This bit is used to determine whether the disabling request generated by POC_IN1 pin input detection selects TIMER0 as a target timer.
0: TIMER0 is not selected
1: TIMER0 is selected
When POC_IN1 generates a disabling request (IN1IF is set to 1 by hardware), if this bit is set to 1, and INDREN bit in POC_IN1DCFG register also is set to 1, the output pins connected with TIMER0 channels will be disabled.
Note: This bit-field can be written only once after the reset.

0	TIMER0_IN0DRSEL	POC_IN0 disabling request select TIMER0
This bit is used to determine whether the disabling request generated by POC_IN0 pin input detection selects TIMER0 as a target timer. 0: TIMER0 is not selected 1: TIMER0 is selected When POC_IN0 generates a disabling request (IN0IF is set to 1 by hardware), if this bit is set to 1, and INDREN bit in POC_IN0DCFG register also is set to 1, the output pins connected with TIMER0 channels will be disabled. Note: This bit-field can be written only once after the reset. The default value is 1 after the reset.		

8.4.10 Request selection register 1 (POC_REQSEL1)

Address offset: 0x4C

Reset value: 0x0404 2414

This register has to be accessed by byte(8-bit), half-word(16-bit) and word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved	GPTIMER1_CMPDRSEL	GPTIMER1_IN5DRSEL	GPTIMER1_IN4DRSEL	GPTIMER1_IN3DRSEL	GPTIMER1_IN2DRSEL	GPTIMER1_IN1DRSEL	GPTIMER1_IN0DRSEL	Reserved	GPTIMER0_CMPDRSEL	GPTIMER0_IN5DRSEL	GPTIMER0_IN4DRSEL	GPTIMER0_IN3DRSEL	GPTIMER0_IN2DRSEL	GPTIMER0_IN1DRSEL	GPTIMER0_IN0DRSEL
	EL	EL	EL	EL	EL	EL	EL		SEL	EL	EL	EL	EL	EL	EL
	rwo	rwo	rwo	rwo	rwo	rwo	rwo		rwo	rwo	rwo	rwo	rwo	rwo	rwo
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	TIMER2_CMPDRSEL	TIMER2_IN5DRSEL	TIMER2_IN4DRSEL	TIMER2_IN3DRSEL	TIMER2_IN2DRSEL	TIMER2_IN1DRSEL	TIMER2_IN0DRSEL	Reserved	TIMER1_CMPDRSEL	TIMER1_IN5DRSEL	TIMER1_IN4DRSEL	TIMER1_IN3DRSEL	TIMER1_IN2DRSEL	TIMER1_IN1DRSEL	TIMER1_IN0DRSEL
	EL								EL						
	rwo	rwo	rwo	rwo	rwo	rwo	rwo		rwo	rwo	rwo	rwo	rwo	rwo	rwo

Bits	Fields	Descriptions
31	Reserved	Must be kept at reset value
30	GPTIMER1_CMPDRSEL	CMPn (n=0,1,2,3) disabling request select GPTIMER1
	EL	This bit is used to determine whether the disabling request generated by CMPn (n=0,1,2,3) output detection selects GPTIMER1 as a target timer. 0: GPTIMER1 is not selected 1: GPTIMER1 is selected When CMPn generates a disabling request (CMPnDF is set to 1 by hardware), if this bit is set to 1, and CMPnDREN or GPTIMER1_CMPnDREN also is set to 1, the output pins connected with GPTIMER1 channels will be disabled. Note: This bit-field can be written only once after the reset.
29	GPTIMER1_IN5DRSEL	POC_IN5 disabling request select GPTIMER1
	EL	This bit is used to determine whether the disabling request generated by POC_IN5 pin input detection selects GPTIMER1 as a target timer. 0: GPTIMER1 is not selected

		1: GPTIMER1 is selected
		When POC_IN5 generates a disabling request (IN5IF is set to 1 by hardware), if this bit is set to 1, and INDREN bit in POC_IN5DCFG register also is set to 1, the output pins connected with GPTIMER1 channels will be disabled.
		Note: This bit-field can be written only once after the reset.
28	GPTIMER1_IN4DRS	POC_IN4 disabling request select GPTIMER1
	EL	This bit is used to determine whether the disabling request generated by POC_IN4 pin input detection selects GPTIMER1 as a target timer.
		0: GPTIMER1 is not selected
		1: GPTIMER1 is selected
		When POC_IN4 generates a disabling request (IN4IF is set to 1 by hardware), if this bit is set to 1, and INDREN bit in POC_IN4DCFG register also is set to 1, the output pins connected with GPTIMER1 channels will be disabled.
		Note: This bit-field can be written only once after the reset.
27	GPTIMER1_IN3DRS	POC_IN3 disabling request select GPTIMER1
	EL	This bit is used to determine whether the disabling request generated by POC_IN3 pin input detection selects GPTIMER1 as a target timer.
		0: GPTIMER1 is not selected
		1: GPTIMER1 is selected
		When POC_IN3 generates a disabling request (IN3IF is set to 1 by hardware), if this bit is set to 1, and INDREN bit in POC_IN3DCFG register also is set to 1, the output pins connected with GPTIMER1 channels will be disabled.
		Note: This bit-field can be written only once after the reset.
26	GPTIMER1_IN2DRS	POC_IN2 disabling request select GPTIMER1
	EL	This bit is used to determine whether the disabling request generated by POC_IN2 pin input detection selects GPTIMER1 as a target timer.
		0: GPTIMER1 is not selected
		1: GPTIMER1 is selected
		When POC_IN2 generates a disabling request (IN2IF is set to 1 by hardware), if this bit is set to 1, and INDREN bit in POC_IN2DCFG register also is set to 1, the output pins connected with GPTIMER1 channels will be disabled.
		Note: This bit-field can be written only once after the reset. The default value is 1 after the reset.
25	GPTIMER1_IN1DRS	POC_IN1 disabling request select GPTIMER1
	EL	This bit is used to determine whether the disabling request generated by POC_IN1 pin input detection selects GPTIMER1 as a target timer.
		0: GPTIMER1 is not selected
		1: GPTIMER1 is selected
		When POC_IN1 generates a disabling request (IN1IF is set to 1 by hardware), if this bit is set to 1, and INDREN bit in POC_IN1DCFG register also is set to 1, the output pins connected with GPTIMER1 channels will be disabled.

Note: This bit-field can be written only once after the reset.

24	GPTIMER1_IN0DRS EL	POC_IN0 disabling request select GPTIMER1 This bit is used to determine whether the disabling request generated by POC_IN0 pin input detection selects GPTIMER1 as a target timer. 0: GPTIMER1 is not selected 1: GPTIMER1 is selected When POC_IN0 generates a disabling request (IN0IF is set to 1 by hardware), if this bit is set to 1, and INDREN bit in POC_IN0DCFG register also is set to 1, the output pins connected with GPTIMER1 channels will be disabled. Note: This bit-field can be written only once after the reset.
23	Reserved	Must be kept at reset value
22	GPTIMER0_CMPDR SEL	CMPn (n=0,1,2,3) disabling request select GPTIMER0 This bit is used to determine whether the disabling request generated by CMPn (n=0,1,2,3) output detection selects GPTIMER0 as a target timer. 0: GPTIMER0 is not selected 1: GPTIMER0 is selected When CMPn generates a disabling request (CMPnDF is set to 1 by hardware), if this bit is set to 1, and CMPnDREN or GPTIMER0_CMPnDREN also is set to 1, the output pins connected with GPTIMER0 channels will be disabled. Note: This bit-field can be written only once after the reset.
21	GPTIMER0_IN5DRS EL	POC_IN5 disabling request select GPTIMER0 This bit is used to determine whether the disabling request generated by POC_IN5 pin input detection selects GPTIMER0 as a target timer. 0: GPTIMER0 is not selected 1: GPTIMER0 is selected When POC_IN5 generates a disabling request (IN5IF is set to 1 by hardware), if this bit is set to 1, and INDREN bit in POC_IN5DCFG register also is set to 1, the output pins connected with GPTIMER0 channels will be disabled. Note: This bit-field can be written only once after the reset.
20	GPTIMER0_IN4DRS EL	POC_IN4 disabling request select GPTIMER0 This bit is used to determine whether the disabling request generated by POC_IN4 pin input detection selects GPTIMER0 as a target timer. 0: GPTIMER0 is not selected 1: GPTIMER0 is selected When POC_IN4 generates a disabling request (IN4IF is set to 1 by hardware), if this bit is set to 1, and INDREN bit in POC_IN4DCFG register also is set to 1, the output pins connected with GPTIMER0 channels will be disabled. Note: This bit-field can be written only once after the reset.
19	GPTIMER0_IN3DRS EL	POC_IN3 disabling request select GPTIMER0 This bit is used to determine whether the disabling request generated by POC_IN3

		pin input detection selects GPTIMER0 as a target timer. 0: GPTIMER0 is not selected 1: GPTIMER0 is selected When POC_IN3 generates a disabling request (IN3IF is set to 1 by hardware), if this bit is set to 1, and INDREN bit in POC_IN3DCFG register also is set to 1, the output pins connected with GPTIMER0 channels will be disabled. Note: This bit-field can be written only once after the reset.
18	GPTIMER0_IN2DRS EL	POC_IN2 disabling request select GPTIMER0 This bit is used to determine whether the disabling request generated by POC_IN2 pin input detection selects GPTIMER0 as a target timer. 0: GPTIMER0 is not selected 1: GPTIMER0 is selected When POC_IN2 generates a disabling request (IN2IF is set to 1 by hardware), if this bit is set to 1, and INDREN bit in POC_IN2DCFG register also is set to 1, the output pins connected with GPTIMER0 channels will be disabled. Note: This bit-field can be written only once after the reset. The default value is 1 after the reset.
17	GPTIMER0_IN1DRS EL	POC_IN1 disabling request select GPTIMER0 This bit is used to determine whether the disabling request generated by POC_IN1 pin input detection selects GPTIMER0 as a target timer. 0: GPTIMER0 is not selected 1: GPTIMER0 is selected When POC_IN1 generates a disabling request (IN1IF is set to 1 by hardware), if this bit is set to 1, and INDREN bit in POC_IN1DCFG register also is set to 1, the output pins connected with GPTIMER0 channels will be disabled. Note: This bit-field can be written only once after the reset.
16	GPTIMER0_IN0DRS EL	POC_IN0 disabling request select GPTIMER0 This bit is used to determine whether the disabling request generated by POC_IN0 pin input detection selects GPTIMER0 as a target timer. 0: GPTIMER0 is not selected 1: GPTIMER0 is selected When POC_IN0 generates a disabling request (IN0IF is set to 1 by hardware), if this bit is set to 1, and INDREN bit in POC_IN0DCFG register also is set to 1, the output pins connected with GPTIMER0 channels will be disabled. Note: This bit-field can be written only once after the reset.
15	Reserved	Must be kept at reset value
14	TIMER2_CMPDRSE L	CMPn (n=0,1,2,3) disabling request select TIMER2 This bit is used to determine whether the disabling request generated by CMPn (n=0,1,2,3) output detection selects TIMER2 as a target timer. 0: TIMER2 is not selected 1: TIMER2 is selected

When CMPn generates a disabling request (CMPnDF is set to 1 by hardware), if this bit is set to 1, and CMPnDREN or TIMER2_CMPnDREN also is set to 1, the output pins connected with TIMER2 channels will be disabled.

Note: This bit-field can be written only once after the reset.

13	TIMER2_IN5DRSEL	POC_IN5 disabling request select TIMER2 This bit is used to determine whether the disabling request generated by POC_IN5 pin input detection selects TIMER2 as a target timer. 0: TIMER2 is not selected 1: TIMER2 is selected When POC_IN5 generates a disabling request (IN5IF is set to 1 by hardware), if this bit is set to 1, and INDREN bit in POC_IN5DCFG register also is set to 1, the output pins connected with TIMER2 channels will be disabled. Note: This bit-field can be written only once after the reset. The default value is 1 after the reset.
12	TIMER2_IN4DRSEL	POC_IN4 disabling request select TIMER2 This bit is used to determine whether the disabling request generated by POC_IN4 pin input detection selects TIMER2 as a target timer. 0: TIMER2 is not selected 1: TIMER2 is selected When POC_IN4 generates a disabling request (IN4IF is set to 1 by hardware), if this bit is set to 1, and INDREN bit in POC_IN4DCFG register also is set to 1, the output pins connected with TIMER2 channels will be disabled. Note: This bit-field can be written only once after the reset.
11	TIMER2_IN3DRSEL	POC_IN3 disabling request select TIMER2 This bit is used to determine whether the disabling request generated by POC_IN3 pin input detection selects TIMER2 as a target timer. 0: TIMER2 is not selected 1: TIMER2 is selected When POC_IN3 generates a disabling request (IN3IF is set to 1 by hardware), if this bit is set to 1, and INDREN bit in POC_IN3DCFG register also is set to 1, the output pins connected with TIMER2 channels will be disabled. Note: This bit-field can be written only once after the reset.
10	TIMER2_IN2DRSEL	POC_IN2 disabling request select TIMER2 This bit is used to determine whether the disabling request generated by POC_IN2 pin input detection selects TIMER2 as a target timer. 0: TIMER2 is not selected 1: TIMER2 is selected When POC_IN2 generates a disabling request (IN2IF is set to 1 by hardware), if this bit is set to 1, and INDREN bit in POC_IN2DCFG register also is set to 1, the output pins connected with TIMER2 channels will be disabled. Note: This bit-field can be written only once after the reset. The default value is 1

		after the reset.
9	TIMER2_IN1DRSEL	POC_IN1 disabling request select TIMER2 This bit is used to determine whether the disabling request generated by POC_IN1 pin input detection selects TIMER2 as a target timer. 0: TIMER2 is not selected 1: TIMER2 is selected When POC_IN1 generates a disabling request (IN1IF is set to 1 by hardware), if this bit is set to 1, and INDREN bit in POC_IN1DCFG register also is set to 1, the output pins connected with TIMER2 channels will be disabled. Note: This bit-field can be written only once after the reset.
8	TIMER2_IN0DRSEL	POC_IN0 disabling request select TIMER2 This bit is used to determine whether the disabling request generated by POC_IN0 pin input detection selects TIMER2 as a target timer. 0: TIMER2 is not selected 1: TIMER2 is selected When POC_IN0 generates a disabling request (IN0IF is set to 1 by hardware), if this bit is set to 1, and INDREN bit in POC_IN0DCFG register also is set to 1, the output pins connected with TIMER2 channels will be disabled. Note: This bit-field can be written only once after the reset.
7	Reserved	Must be kept at reset value
6	TIMER1_CMPDRSEL	CMPn (n=0,1,2,3) disabling request select TIMER1 This bit is used to determine whether the disabling request generated by CMPn (n=0,1,2,3) output detection selects TIMER1 as a target timer. 0: TIMER1 is not selected 1: TIMER1 is selected When CMPn generates a disabling request (CMPnDF is set to 1 by hardware), if this bit is set to 1, and CMPnDREN or TIMER1_CMPnDREN also is set to 1, the output pins connected with TIMER1 channels will be disabled. Note: This bit-field can be written only once after the reset.
5	TIMER1_IN5DRSEL	POC_IN5 disabling request select TIMER1 This bit is used to determine whether the disabling request generated by POC_IN5 pin input detection selects TIMER1 as a target timer. 0: TIMER1 is not selected 1: TIMER1 is selected When POC_IN5 generates a disabling request (IN5IF is set to 1 by hardware), if this bit is set to 1, and INDREN bit in POC_IN5DCFG register also is set to 1, the output pins connected with TIMER1 channels will be disabled. Note: This bit-field can be written only once after the reset.
4	TIMER1_IN4DRSEL	POC_IN4 disabling request select TIMER1 This bit is used to determine whether the disabling request generated by POC_IN4

- pin input detection selects TIMER1 as a target timer.
0: TIMER1 is not selected
1: TIMER1 is selected
When POC_IN4 generates a disabling request (IN4IF is set to 1 by hardware), if this bit is set to 1, and INDREN bit in POC_IN4DCFG register also is set to 1, the output pins connected with TIMER1 channels will be disabled.
Note: This bit-field can be written only once after the reset. The default value is 1 after the reset.
- 3 TIMER1_IN3DRSEL POC_IN3 disabling request select TIMER1
This bit is used to determine whether the disabling request generated by POC_IN3 pin input detection selects TIMER1 as a target timer.
0: TIMER1 is not selected
1: TIMER1 is selected
When POC_IN3 generates a disabling request (IN3IF is set to 1 by hardware), if this bit is set to 1, and INDREN bit in POC_IN3DCFG register also is set to 1, the output pins connected with TIMER1 channels will be disabled.
Note: This bit-field can be written only once after the reset.
- 2 TIMER1_IN2DRSEL POC_IN2 disabling request select TIMER1
This bit is used to determine whether the disabling request generated by POC_IN2 pin input detection selects TIMER1 as a target timer.
0: TIMER1 is not selected
1: TIMER1 is selected
When POC_IN2 generates a disabling request (IN2IF is set to 1 by hardware), if this bit is set to 1, and INDREN bit in POC_IN2DCFG register also is set to 1, the output pins connected with TIMER1 channels will be disabled.
Note: This bit-field can be written only once after the reset.
- 1 TIMER1_IN1DRSEL POC_IN1 disabling request select TIMER1
This bit is used to determine whether the disabling request generated by POC_IN1 pin input detection selects TIMER1 as a target timer.
0: TIMER1 is not selected
1: TIMER1 is selected
When POC_IN1 generates a disabling request (IN1IF is set to 1 by hardware), if this bit is set to 1, and INDREN bit in POC_IN1DCFG register also is set to 1, the output pins connected with TIMER1 channels will be disabled.
Note: This bit-field can be written only once after the reset.
- 0 TIMER1_IN0DRSEL POC_IN0 disabling request select TIMER1
This bit is used to determine whether the disabling request generated by POC_IN0 pin input detection selects TIMER1 as a target timer.
0: TIMER1 is not selected
1: TIMER1 is selected

When POC_IN0 generates a disabling request (IN0IF is set to 1 by hardware), if this bit is set to 1, and INDREN bit in POC_IN0DCFG register also is set to 1, the output pins connected with TIMER1 channels will be disabled.

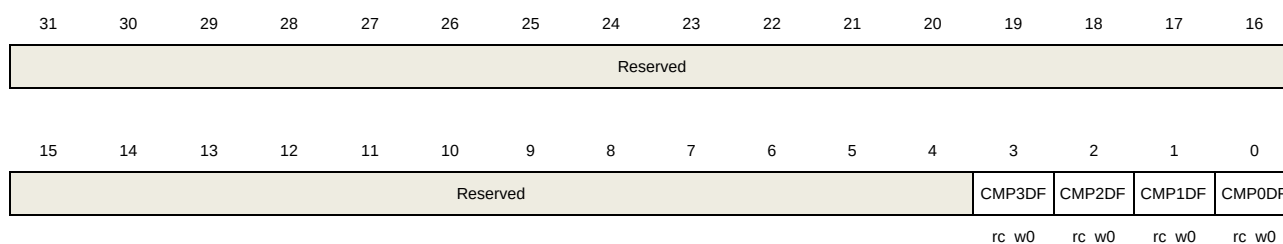
Note: This bit-field can be written only once after the reset.

8.4.11 Status register 1 (POC_STAT1)

Address offset: 0x60

Reset value: 0x0000 0000

This register has to be accessed by byte(8-bit), half-word(16-bit) and word(32-bit).



Bits	Fields	Descriptions
31:4	Reserved	Must be kept at reset value
3	CMP3DF	Comparator 3 detection flag This bit is used to show that whether a disabling request has been generated by comparator 3 output detection. 0: A disabling request has not been generated 1: A disabling request has been generated Set by hardware when the comparator 3 output signal has changed from a low level to a high level. Cleared by software writing 0 to it.
2	CMP2DF	Comparator 2 detection flag This bit is used to show whether a disabling request has been generated by comparator 2 output detection. 0: A disabling request has not been generated 1: A disabling request has been generated Set by hardware when the comparator 2 output signal has changed from a low level to a high level. Cleared by software writing 0 to it.
1	CMP1DF	Comparator 1 detection flag This bit is used to show whether a disabling request has been generated by comparator 1 output detection. 0: A disabling request has not been generated 1: A disabling request has been generated Set by hardware when the comparator 1 output signal has changed from a low level

to a high level.

Cleared by software writing 0 to it.

0 CMP0DF

Comparator 0 detection flag

This bit is used to show whether a disabling request has been generated by comparator 0 output detection.

0: A disabling request has not been generated

1: A disabling request has been generated

Set by hardware when the comparator 0 output signal has changed from a low level to a high level.

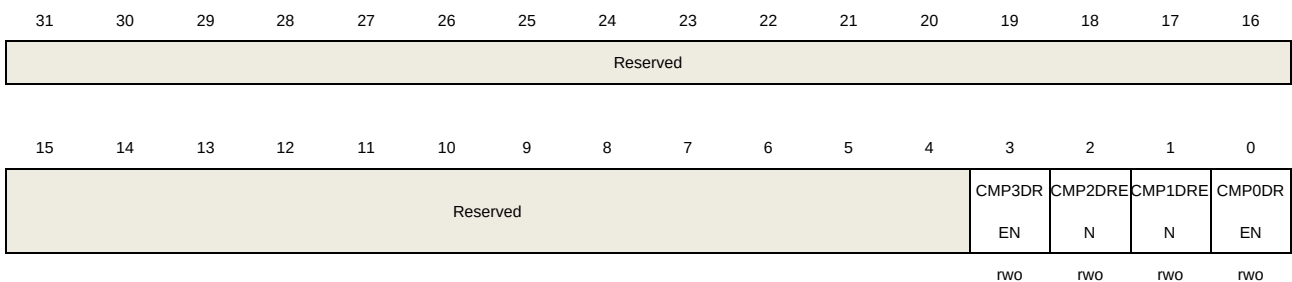
Cleared by software writing 0 to it.

8.4.12 Control register 1 (POC_CTL1)

Address offset: 0x64

Reset value: 0x0000 0000

This register has to be accessed by byte(8-bit), half-word(16-bit) and word(32-bit).



Bits	Fields	Descriptions
31:4	Reserved	Must be kept at reset value
3	CMP3DREN	Comparator 3 disabling request enable This bit is used to determine whether the disabling request generated by comparator 3 output detection is enabled for TIMEx (x=0,1,2,7) and GPTIMEx (x=0,1). 0: The disabling request is disabled 1: The disabling request is enabled If enabled, TIMEx (x=0,1,2,7) channel pin outputs are disabled when CMP3DF = 1 and TIMEx_CMPDRSEL = 1. If enabled, GPTIMEx (x=0,1) channel pin outputs are disabled when CMP3DF = 1 and GPTIMEx_CMPDRSEL = 1. Note: This bit-field can be written only once after the reset.
2	CMP2DREN	Comparator 2 disabling request enable This bit is used to determine whether the disabling request generated by comparator 2 output detection is enabled for TIMEx (x=0,1,2,7) and GPTIMEx (x=0,1). 0: The disabling request is disabled 1: The disabling request is enabled

If enabled, TIMERx (x=0,1,2,7) channel pin outputs are disabled when CMP2DF = 1 and TIMERx_CMPDRSEL = 1.

If enabled, GPTIMERx (x=0,1) channel pin outputs are disabled when CMP2DF = 1 and GPTIMERx_CMPDRSEL = 1.

Note: This bit-field can be written only once after the reset.

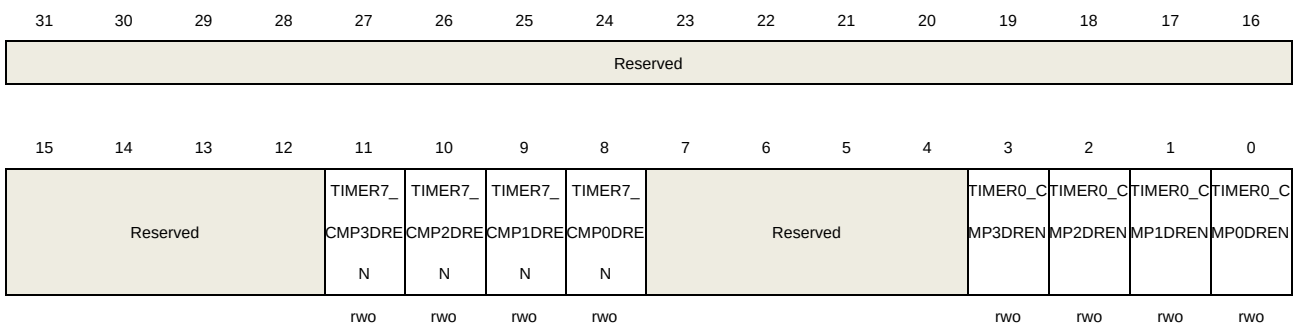
1	CMP1DREN	Comparator 1 disabling request enable This bit is used to determine whether the disabling request generated by comparator 1 output detection is enabled for TIMERx (x=0,1,2,7) and GPTIMERx (x=0,1). 0: The disabling request is disabled 1: The disabling request is enabled If enabled, TIMERx (x=0,1,2,7) channel pin outputs are disabled when CMP1DF = 1 and TIMERx_CMPDRSEL = 1. If enabled, GPTIMERx (x=0,1) channel pin outputs are disabled when CMP1DF = 1 and GPTIMERx_CMPDRSEL = 1. Note: This bit-field can be written only once after the reset.
0	CMP0DREN	Comparator 0 disabling request enable This bit is used to determine whether the disabling request generated by comparator 0 output detection is enabled for TIMERx (x=0,1,2,7) and GPTIMERx (x=0,1). 0: The disabling request is disabled 1: The disabling request is enabled If enabled, TIMERx (x=0,1,2,7) channel pin outputs are disabled when CMP0DF = 1 and TIMERx_CMPDRSEL = 1. If enabled, GPTIMERx (x=0,1) channel pin outputs are disabled when CMP0DF = 1 and GPTIMERx_CMPDRSEL = 1. Note: This bit-field can be written only once after the reset.

8.4.13 Extended control register 0 (POC_EXTCTL0)

Address offset: 0x68

Reset value: 0x0000 0000

This register has to be accessed by byte(8-bit), half-word(16-bit) and word(32-bit).



Bits	Fields	Descriptions

31:12	Reserved	Must be kept at reset value
11	TIMER7_CMP3DRE N	Comparator 3 disabling request enable for TIMER7 This bit is used to determine whether the disabling request generated by comparator 3 output detection is enabled for TIMER7 channel pins. 0: The disabling request is disabled 1: The disabling request is enabled If enabled, TIMER7 channel pin outputs are disabled when CMP3DF = 1 and TIMER7_CMPDRSEL = 1. Note: This bit-field can be written only once after the reset.
10	TIMER7_CMP2DRE N	Comparator 2 disabling request enable for TIMER7 This bit is used to determine whether the disabling request generated by comparator 2 output detection is enabled for TIMER7 channel pins. 0: The disabling request is disabled 1: The disabling request is enabled If enabled, TIMER7 channel pin outputs are disabled when CMP2DF = 1 and TIMER7_CMPDRSEL = 1. Note: This bit-field can be written only once after the reset.
9	TIMER7_CMP1DRE N	Comparator 1 disabling request enable for TIMER7 This bit is used to determine whether the disabling request generated by comparator 1 output detection is enabled for TIMER7 channel pins. 0: The disabling request is disabled 1: The disabling request is enabled If enabled, TIMER7 channel pin outputs are disabled when CMP1DF = 1 and TIMER7_CMPDRSEL = 1. Note: This bit-field can be written only once after the reset.
8	TIMER7_CMP0DRE N	Comparator 0 disabling request enable for TIMER7 This bit is used to determine whether the disabling request generated by comparator 0 output detection is enabled for TIMER7 channel pins. 0: The disabling request is disabled 1: The disabling request is enabled If enabled, TIMER7 channel pin outputs are disabled when CMP0DF = 1 and TIMER7_CMPDRSEL = 1. Note: This bit-field can be written only once after the reset.
7:4	Reserved	Must be kept at reset value
3	TIMER0_CMP3DRE N	Comparator 3 disabling request enable for TIMER0 This bit is used to determine whether the disabling request generated by comparator 3 output detection is enabled for TIMER0 channel pins. 0: The disabling request is disabled 1: The disabling request is enabled If enabled, TIMER0 channel pin outputs are disabled when CMP3DF = 1 and

TIMER0_CMPDRSEL = 1.

Note: This bit-field can be written only once after the reset.

- 2 TIMER0_CMP2DRE Comparator 2 disabling request enable for TIMER0
N This bit is used to determine whether the disabling request generated by comparator 2 output detection is enabled for TIMER0 channel pins.
0: The disabling request is disabled
1: The disabling request is enabled
If enabled, TIMER0 channel pin outputs are disabled when CMP2DF = 1 and TIMER0_CMPDRSEL = 1.
Note: This bit-field can be written only once after the reset.
- 1 TIMER0_CMP1DRE Comparator 1 disabling request enable for TIMER0
N This bit is used to determine whether the disabling request generated by comparator 1 output detection is enabled for TIMER0 channel pins.
0: The disabling request is disabled
1: The disabling request is enabled
If enabled, TIMER0 channel pin outputs are disabled when CMP1DF = 1 and TIMER0_CMPDRSEL = 1.
Note: This bit-field can be written only once after the reset.
- 0 TIMER0_CMP0DRE Comparator 0 disabling request enable for TIMER0
N This bit is used to determine whether the disabling request generated by comparator 0 output detection is enabled for TIMER0 channel pins.
0: The disabling request is disabled
1: The disabling request is enabled
If enabled, TIMER0 channel pin outputs are disabled when CMP0DF = 1 and TIMER0_CMPDRSEL = 1.
Note: This bit-field can be written only once after the reset.

8.4.14 Extended control register 1 (POC_EXTCTL1)

Address offset: 0x6C

Reset value: 0x0000 0000

This register has to be accessed by byte(8-bit), half-word(16-bit) and word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				GPTIMER1_CMP3DRE	GPTIMER1_CMP2DRE	GPTIMER1_CMP1DRE	GPTIMER1_CMP0DRE	Reserved				GPTIMER0_CMP3DRE	GPTIMER0_CMP2DRE	GPTIMER0_CMP1DRE	GPTIMER0_CMP0DRE
				REN	REN	REN	REN					REN	REN	REN	REN
				rwo	rwo	rwo	rwo					rwo	rwo	rwo	rwo
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				TIMER2_CMP3DRE	TIMER2_CMP2DRE	TIMER2_CMP1DRE	TIMER2_CMP0DRE	Reserved				TIMER1_CMP3DRE	TIMER1_CMP2DRE	TIMER1_CMP1DRE	TIMER1_CMP0DRE
				N	N	N	N					N	N	N	N

Bits	Fields	Descriptions
31:28	Reserved	Must be kept at reset value
27	GPTIMER1_CMP3D REN	Comparator 3 disabling request enable for GPTIMER1 This bit is used to determine whether the disabling request generated by comparator 3 output detection is enabled for GPTIMER1 channel pins. 0: The disabling request is disabled 1: The disabling request is enabled If enabled, GPTIMER1 channel pin outputs are disabled when CMP3DF = 1 and GPTIMER1_CMPDRSEL = 1. Note: This bit-field can be written only once after the reset.
26	GPTIMER1_CMP2D REN	Comparator 2 disabling request enable for GPTIMER1 This bit is used to determine whether the disabling request generated by comparator 2 output detection is enabled for GPTIMER1 channel pins. 0: The disabling request is disabled 1: The disabling request is enabled If enabled, GPTIMER1 channel pin outputs are disabled when CMP2DF = 1 and GPTIMER1_CMPDRSEL = 1. Note: This bit-field can be written only once after the reset.
25	GPTIMER1_CMP1D REN	Comparator 1 disabling request enable for GPTIMER1 This bit is used to determine whether the disabling request generated by comparator 1 output detection is enabled for GPTIMER1 channel pins. 0: The disabling request is disabled 1: The disabling request is enabled If enabled, GPTIMER1 channel pin outputs are disabled when CMP1DF = 1 and GPTIMER1_CMPDRSEL = 1. Note: This bit-field can be written only once after the reset.
24	GPTIMER1_CMP0D REN	Comparator 0 disabling request enable for GPTIMER1 This bit is used to determine whether the disabling request generated by comparator 0 output detection is enabled for GPTIMER1 channel pins. 0: The disabling request is disabled 1: The disabling request is enabled If enabled, GPTIMER1 channel pin outputs are disabled when CMP0DF = 1 and GPTIMER1_CMPDRSEL = 1. Note: This bit-field can be written only once after the reset.
23:20	Reserved	Must be kept at reset value
19	GPTIMER0_CMP3D REN	Comparator 3 disabling request enable for GPTIMER0 This bit is used to determine whether the disabling request generated by comparator 3 output detection is enabled for GPTIMER0 channel pins.

		0: The disabling request is disabled 1: The disabling request is enabled If enabled, GPTIMER0 channel pin outputs are disabled when CMP3DF = 1 and GPTIMER0_CMPDRSEL = 1. Note: This bit-field can be written only once after the reset.
18	GPTIMER0_CMP2D REN	Comparator 2 disabling request enable for GPTIMER0 This bit is used to determine whether the disabling request generated by comparator 2 output detection is enabled for GPTIMER0 channel pins. 0: The disabling request is disabled 1: The disabling request is enabled If enabled, GPTIMER0 channel pin outputs are disabled when CMP2DF = 1 and GPTIMER0_CMPDRSEL = 1. Note: This bit-field can be written only once after the reset.
17	GPTIMER0_CMP1D REN	Comparator 1 disabling request enable for GPTIMER0 This bit is used to determine whether the disabling request generated by comparator 1 output detection is enabled for GPTIMER0 channel pins. 0: The disabling request is disabled 1: The disabling request is enabled If enabled, GPTIMER0 channel pin outputs are disabled when CMP1DF = 1 and GPTIMER0_CMPDRSEL = 1. Note: This bit-field can be written only once after the reset.
16	GPTIMER0_CMP0D REN	Comparator 0 disabling request enable for GPTIMER0 This bit is used to determine whether the disabling request generated by comparator 0 output detection is enabled for GPTIMER0 channel pins. 0: The disabling request is disabled 1: The disabling request is enabled If enabled, GPTIMER0 channel pin outputs are disabled when CMP0DF = 1 and GPTIMER0_CMPDRSEL = 1. Note: This bit-field can be written only once after the reset.
15:12	Reserved	Must be kept at reset value
11	TIMER2_CMP3DRE N	Comparator 3 disabling request enable for TIMER2 This bit is used to determine whether the disabling request generated by comparator 3 output detection is enabled for TIMER2 channel pins. 0: The disabling request is disabled 1: The disabling request is enabled If enabled, TIMER2 channel pin outputs are disabled when CMP3DF = 1 and TIMER2_CMPDRSEL = 1. Note: This bit-field can be written only once after the reset.
10	TIMER2_CMP2DRE N	Comparator 2 disabling request enable for TIMER2 This bit is used to determine whether the disabling request generated by comparator

		2 output detection is enabled for TIMER2 channel pins. 0: The disabling request is disabled 1: The disabling request is enabled If enabled, TIMER2 channel pin outputs are disabled when CMP2DF = 1 and TIMER2_CMPDRSEL = 1. Note: This bit-field can be written only once after the reset.
9	TIMER2_CMP1DRE N	Comparator 1 disabling request enable for TIMER2 This bit is used to determine whether the disabling request generated by comparator 1 output detection is enabled for TIMER2 channel pins. 0: The disabling request is disabled 1: The disabling request is enabled If enabled, TIMER2 channel pin outputs are disabled when CMP1DF = 1 and TIMER2_CMPDRSEL = 1. Note: This bit-field can be written only once after the reset.
8	TIMER2_CMP0DRE N	Comparator 0 disabling request enable for TIMER2 This bit is used to determine whether the disabling request generated by comparator 0 output detection is enabled for TIMER2 channel pins. 0: The disabling request is disabled 1: The disabling request is enabled If enabled, TIMER2 channel pin outputs are disabled when CMP0DF = 1 and TIMER2_CMPDRSEL = 1. Note: This bit-field can be written only once after the reset.
7:4	Reserved	Must be kept at reset value
3	TIMER1_CMP3DRE N	Comparator 3 disabling request enable for TIMER1 This bit is used to determine whether the disabling request generated by comparator 3 output detection is enabled for TIMER1 channel pins. 0: The disabling request is disabled 1: The disabling request is enabled If enabled, TIMER1 channel pin outputs are disabled when CMP3DF = 1 and TIMER1_CMPDRSEL = 1. Note: This bit-field can be written only once after the reset.
2	TIMER1_CMP2DRE N	Comparator 2 disabling request enable for TIMER1 This bit is used to determine whether the disabling request generated by comparator 2 output detection is enabled for TIMER1 channel pins. 0: The disabling request is disabled 1: The disabling request is enabled If enabled, TIMER1 channel pin outputs are disabled when CMP2DF = 1 and TIMER1_CMPDRSEL = 1. Note: This bit-field can be written only once after the reset.
1	TIMER1_CMP1DRE	Comparator 1 disabling request enable for TIMER1

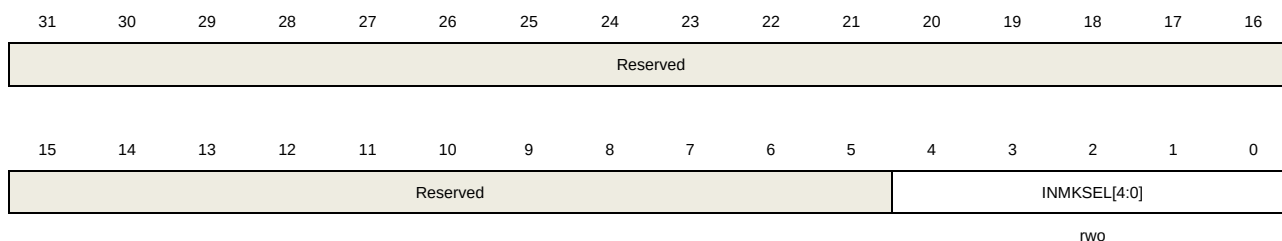
N		This bit is used to determine whether the disabling request generated by comparator 1 output detection is enabled for TIMER1 channel pins. 0: The disabling request is disabled 1: The disabling request is enabled If enabled, TIMER1 channel pin outputs are disabled when CMP1DF = 1 and TIMER1_CMPDRSEL = 1. Note: This bit-field can be written only once after the reset.
0	TIMER1_CMP0DRE	Comparator 0 disabling request enable for TIMER1 N This bit is used to determine whether the disabling request generated by comparator 0 output detection is enabled for TIMER1 channel pins. 0: The disabling request is disabled 1: The disabling request is enabled If enabled, TIMER1 channel pin outputs are disabled when CMP0DF = 1 and TIMER1_CMPDRSEL = 1. Note: This bit-field can be written only once after the reset.

8.4.15 Input n detection mask register (POC_INnDMK) (n=0...5)

Address offset: 0x70 + 0x04 * n

Reset value: 0x0000 0000

This register has to be accessed by byte(8-bit), half-word(16-bit) and word(32-bit).



Bits	Fields	Descriptions
31:5	Reserved	Must be kept at reset value
4:0	INMKSEL[4:0]	POC_INn input detection mask source selection These bits are used to select a timer channel signal to mask POC_INn pin input. When the selected channel signal is high, the input signal from POC_INn pin is masked. 00000: Don't mask 00001: TIMER0_CH0 00010: TIMER0_CH1 00011: TIMER0_CH2 00100: TIMER0_CH3 00101: TIMER0_MCH0 00110: TIMER0_MCH1 00111: TIMER0_MCH2

01000: TIMER0_MCH3
01001: TIMER7_CH0
01010: TIMER7_CH1
01011: TIMER7_CH2
01100: TIMER7_CH3
01101: TIMER7_MCH0
01110: TIMER7_MCH1
01111: TIMER7_MCH2
10000: TIMER7_MCH3
10001: TIMER1_CH0
10010: TIMER1_CH1
10011: TIMER1_CH2
10100: TIMER1_CH3
10101: TIMER2_CH0
10110: TIMER2_CH1
10111: TIMER2_CH2
11000: TIMER2_CH3
11001: GPTIMER0_CH0
11010: GPTIMER0_CH1
11011: GPTIMER1_CH0
11100: GPTIMER1_CH1
Other values are reserved.

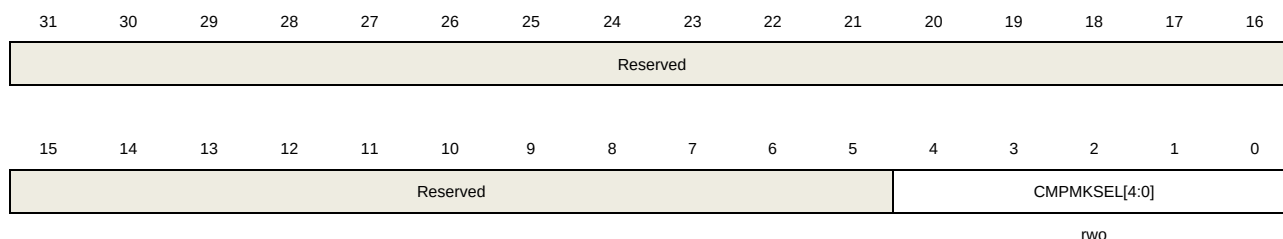
Note: This bit-field can be written only once after the reset.

8.4.16 Comparator n detection mask register (POC_CMPnDMK) (n=0...3)

Address offset: $0x90 + 0x04 * n$

Reset value: 0x0000 0000

This register has to be accessed by byte(8-bit), half-word(16-bit) and word(32-bit).



Bits	Fields	Descriptions
31:5	Reserved	Must be kept at reset value
4:0	CMPMKSEL[4:0]	Comparator n output detection mask source selection These bits are used to select a timer channel signal to mask comparator n output signal. When the selected channel signal is high, the output signal from comparator n is masked.

Please refer to INMKSEL[4:0] descriptions.

Note: This bit-field can be written only once after the reset.

9 GPTIMER Output Controller (GTOC)

9.1 Overview

The GTOC can generate request to close the output pin of GPTIMER. The closed pins can output high-impedance (Hi-Z) state. Output closing request sources can come from different modules on the chip, such as CMP, GPTIMER and so on.

9.2 Characteristics

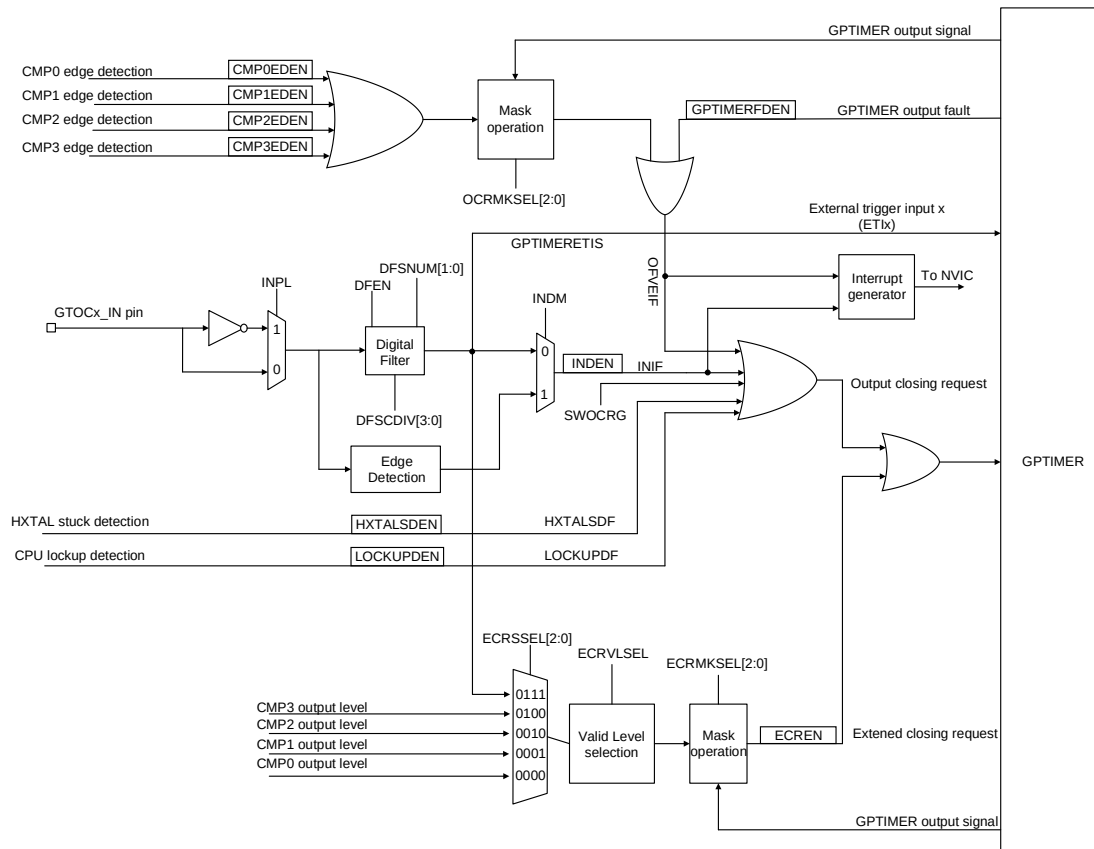
The main features of GTOC are as follows:

- GTOCx_IN (x=0...3) pin input detection:
 - Edge detection.
 - Level detection.
 - Input inverting control.
 - Digital filter
- Request sources for GPTIMER output close:
 - GTOCx_IN (x=0...3) pin input detection.
 - GPTIMER output fault detection (concurrent low or high level output on channel 0 and channel 1)
 - Software request generation.
 - HXTAL stuck event.
 - CPU LOCKUP event.
 - Comparator output detection.
- Request masking function:
 - Output closing request can be masked by GPTIMER output signals.
 - Extended closing request can be masked by GPTIMER output signals.
- Interrupt generation:
 - Valid signal input of GTOCx_IN (x=0...3) pin.
 - GPTIMER output fault or comparator valid edge output.

9.3 Function overview

The block diagram of GTOCx (x=0...3) is shown below:

Figure 9-1. GTOCx (x=0...3) block diagram



The target channels of GTOC is shown below:

Table 9-1. GPTIMER channels for output closing control

	GPTIMER	
Target GPTIMER	GPTIMER0	GPTIMER1
Target channel	GPTIMER0_CH0 GPTIMER0_CH1	GPTIMER1_CH0 GPTIMER1_CH1

9.3.1 GTOC clock

The clock of the GTOC which is connected to AHB bus, is synchronous with HCLK.

9.3.2 Output close function

GTOCx (x=0...3) can generate two kinds of closing requests, one is an output closing request based on flag and software settings, and the other is an extended closing request based on level detection.

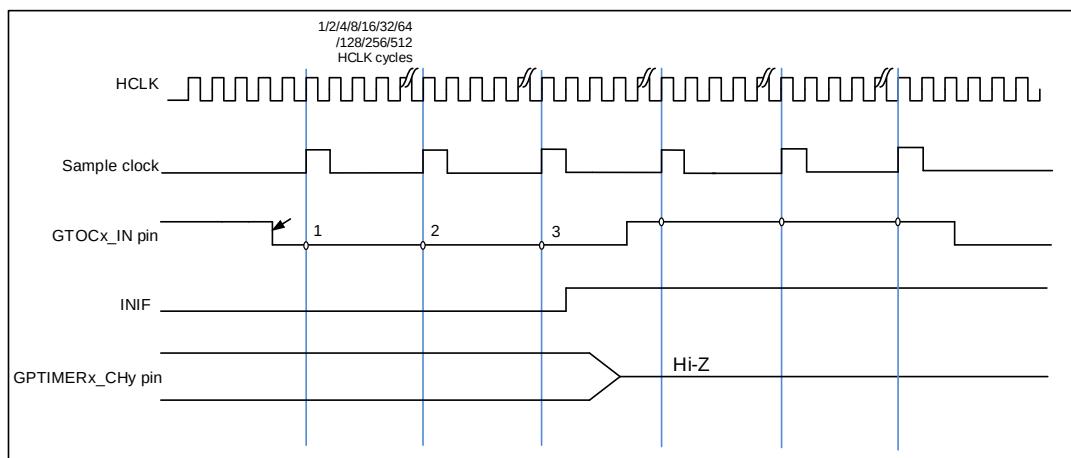
GTOC input detection

Each GTOC has an external input pin. GTOCx ($x=0\dots3$) can perform level or edge detection on GTOCx_IN ($x=0\dots3$) pin input signal, and setting the INDM bit in GTOCx_CFG ($x=0\dots3$) registers can select level detection or edge detection. If GTOCx_IN ($x=0\dots3$) input detection is enabled (INDEN = 1), once a valid edge or level is entered into the GTOCx_IN ($x=0\dots3$) pin, the interrupt flag bit INIF in GTOCx_CFG ($x=0\dots3$) registers is set to 1 by hardware and the output closing request is generated.

By setting the INPL bit in GTOCx_CFG ($x=0\dots3$) registers, the GTOCx_IN ($x=0\dots3$) pin input signal can be inverted. If the INPL bit is set to 0, rising edge and high level detection are performed. Otherwise, falling edge and low level detection are performed.

The digital filtering function is enabled by setting the DFEN bit in GTOCx_CFG ($x=0\dots3$) registers. The DFSNUM[1:0] bits in GTOCx_CFG ($x=0\dots3$) registers can determine the valid number that the same level value is continuously sampled. If the same level is continuously sampled for specific times, then this level will be regarded as the filtered output. The sampling frequency of the filter can be determined by setting the DFSCDIV[3:0] bits in GTOCx_CFG ($x=0\dots3$) registers.

Figure 9-2. GTOCx_IN pin input detection with digital filter (DFSNUM[1:0] = 2'b00, INPL=1, INDM=0)



The software configuration process for GTOC pin input detection is as follows:

1. Configure the GTOC pin for the corresponding GPIO AF function.
2. Configure input polarity bit INPL in the GTOCx_CFG($x=0\dots3$) register.
3. Configure detect mode bit INDM in the GTOCx_CFG($x=0\dots3$) register.
4. If the digital filter function is used, configure the DFSNUM[1:0] bits, DFSCDIV[3:0] bits and enable bit DFEN in the GTOCx_CFG($x=0\dots3$) register.
5. Configure detection enable bit INDEN in the GTOCx_CFG($x=0\dots3$) register.

Note: If GTOC low-level detection is used, the AF function of the corresponding GPIO must be configured and configured before the GTOC detection is enabled.

GTOCx_IN (x=0...3) pin input signal can also be used to generate extended closing request and as an external trigger input for GPTIMER.

Clearing the interrupt flag bit INIF in GTOCx_CFG (x=0...3) registers can stop the output closing request.

GPTIMER output detection

When the GPTIMER channel 0 and channel 1 outputs high or low level at the same time, if the corresponding detection switch is turned on (CHHOUTDET bit or CHLOUTDET bit in GPTIMER module) and GPTIMER output detection function is enabled (GPTIMERFDEN = 1), then the interrupt flag bit OFVEIF in GTOCx_CFG (x=0...3) registers is set to 1 by hardware and the output closing request is generated. Please refer to the GPTIMER chapter for more details.

Clearing the interrupt flag bit OFVEIF in GTOCx_CFG (x=0...3) registers can stop the output closing request.

System fault detection

HXTAL stuck event

When the HXTAL stuck event occurs, if the HXTAL stuck detection enable bit is set to 1 (HXTALSDEN = 1), then the detection flag bit HXTALSDF in GTOCx_CFG (x=0...3) registers is changed to 1 and the output closing request is generated.

Clearing the detection flag bit HXTALSDF in GTOCx_CFG (x=0...3) registers can stop the output closing request.

CPU lockup event

When CPU lockup event is detected, the LOCKUPDF flag in GTOCx_CFG (x=0...3) registers becomes 1 only if the LOCKUPDEN bit in GTOCx_CFG (x=0...3) register is 1. And the output closing request is generated.

Clearing the detection flag bit LOCKUPDF in GTOCx_CFG (x=0...3) registers can stop the output closing request.

Output close by software

By setting SWOCRG bit in GTOCx_CFG (x=0...3) registers to 1, the output closing request can be generated.

Writing 0 to SWOCRG bit in GTOCx_CFG (x=0...3) registers can stop the output closing request.

Comparator output detection

For comparator edge detection, if the CMPn (n=0...3) outputs a valid edge and the corresponding comparator edge detection enable bit is set to 1 (CMPnEDEN = 1), then the interrupt flag bit OFVEIF is set to 1 and the output closing request is generated.

Comparator level detection can generate extended closing requests. The ECRSSEL[2:0] bits can be used to select a CMPn (n=0...3) output signal or GTOCx_IN (x=0...3) pin input signal to generate an extended closing request. If the selected signal matches the valid level set by the ECRVLSSEL bit in GTOCx_ECRCTL (x=0...3) registers, then the extended closing request will be generated. Only when the corresponding extended close enable bit is set to 1 (ECREN = 1), the extended closing request can output to the GPTIMER module.

The output closing request generated by comparator edge detection can be stopped by clearing the interrupt flag bit OFVEIF.

When comparator output level or GTOCx_IN (x=0...3) pin input level mismatches the valid level set by the ECRVLSSEL bit in GTOCx_ECRCTL (x=0...3) registers, the extended closing request can be stopped.

Output close masking function

Output closing request generated by CMPn (n=0...3) edge detection and extended closing request can be masked by GPTIMER channel output signals. When the selected channel signal is high level, the closing request signal is masked.

Setting the OCRMKSEL[2:0] bits in GTOCx_OCRCTL (x=0...3) registers can select the GPTIMER channel signal to mask output closing request generated by CMPn (n=0...3) edge detection.

Setting the ECRMKSEL[2:0] bits in GTOCx_EXTCTL (x=0...3) registers can select the GPTIMER channel signal to mask extended closing request.

9.3.3 GTOC interrupt

The interrupt can be produced on one of the detection:

- GTOCx_IN (x=0...3) pin input detection related to interrupt flag INIF.
- GPTIMER output fault detection related to interrupt flag OFVEIF.
- CMPn (n=0...3) edge detection related to interrupt flag OFVEIF.

Each GTOC has an interrupt. The interrupts of GTOC0, GTOC1, GTOC2 and GTOC3 are mapped into four different interrupt vectors ISR.

9.4 Register definition

GTOC base address: 0x4800 4800

9.4.1 GTOCx configuration register (GTOCx_CFG) (x=0...3)

Address offset: 0x00 + 0x10 * x

Reset value: 0x0000 0000

This register has to be accessed by byte(8-bit), half-word(16-bit) and word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DFSCDIV[1:0]	DFEN	INPL	DFSNUM[1:0]	INDM	DFSCDIV[3:2]	Reserved								SWOCRG	GPTIMER ETIS
rw		rw	rw	rw	rw									rw	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				CMP3EDE	CMP2EDE	CMP1EDE	CMP0EDE	LOCKUPD	HXTALSD	GPTIMER	INDEN	LOCKUPD	HXTALSD	OFVEIF	INIF
				N	N	N	N	EN	EN	FDEN			F	F	
				rw	rw	rw	rw	rw	rw	rw	rw	rc_w0	rc_w0	rc_w0	rc_w0

Bits	Fields	Descriptions
31:30	DFSCDIV[1:0]	GTOCx digital filter sampling clock division DFSCDIV[3:2] and DFSCDIV[1:0] bits are used to select the sampling frequency of digital filter. 0000: Sampling frequency is f_{HCLK} 0001: Sampling frequency is $f_{HCLK}/8$ 0010: Sampling frequency is $f_{HCLK}/32$ 0011: Sampling frequency is $f_{HCLK}/128$ 0100: Sampling frequency is $f_{HCLK}/2$ 0101: Sampling frequency is $f_{HCLK}/4$ 0110: Sampling frequency is $f_{HCLK}/16$ 0111: Sampling frequency is $f_{HCLK}/64$ 1000: Sampling frequency is $f_{HCLK}/256$ 1001: Sampling frequency is $f_{HCLK}/512$ All other values: reserved.
29	DFEN	GTOCx digital filter enable 0: Digital filter disable 1: Digital filter enable
28	INPL	Polarity of GTOCx_IN input 0: Input is not inverted 1: Input is inverted
27:26	DFSNUM[1:0]	GTOCx digital filter sampling number These bit is used to select valid sampling number of the digital filter.

		00: The same value is continuously sampled 3 times. 01: The same value is continuously sampled 4 times. 10: The same value is continuously sampled 5 times. 11: The same value is continuously sampled 6 times.
25	INDM	GTOCx_IN input detection mode 0: Level detection 1: Edge detection Note: When INPL is 0, high level (INDM=0) or rising edge (INDM=1) is valid signal for detection. When INPL is 1, low level (INDM=0) or falling edge (INDM=1) is valid signal for detection.
24:23	DFSCDIV[3:2]	Refer to DFSCDIV[1:0] bits.
22:18	Reserved	Must be kept at reset value.
17	SWOCRG	Software output closing request generation 0: Does not generate output closing request by software 1: Generate output closing request by software
16	GPTIMERETIS	GPTIMER external trigger input x status The GPTIMER external trigger input x is the GTOCx_IN input signal that has passed through inverter and filter. 0: Low level 1: High level Note: This bit is read-only.
15:12	Reserved	Must be kept at reset value.
11	CMP3EDEN	CMP3 valid edge detection enable 0: Disable. 1: Enable. Note: This bit-field can be written only once after the reset.
10	CMP2EDEN	CMP2 valid edge detection enable 0: Disable. 1: Enable. Note: This bit-field can be written only once after the reset.
9	CMP1EDEN	CMP1 valid edge detection enable 0: Disable. 1: Enable. Note: This bit-field can be written only once after the reset.
8	CMP0EDEN	CMP0 valid edge detection enable 0: Disable. 1: Enable. Note: This bit-field can be written only once after the reset.

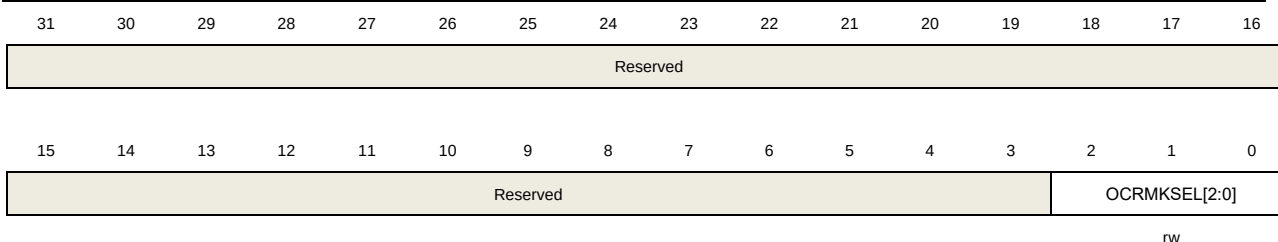
7	LOCKUPDEN	CPU LOCKUP detection enable 0: Disable. 1: Enable. Note: This bit-field can be written only once after the reset.
6	HXTALSDEN	HXTAL stuck detection enable 0: Disable. 1: Enable. Note: This bit-field can be written only once after the reset.
5	GPTIMERFDEN	GPTIMER output fault detection enable 0: Disable. 1: Enable. Note: This bit-field can be written only once after the reset.
4	INDEN	GTOCx_IN input detection enable 0: Disable. 1: Enable. Note: This bit-field can be written only once after the reset.
3	LOCKUPDF	CPU LOCKUP detection flag 0: CPU LOCKUP is not detected. 1: CPU LOCKUP is detected.
2	HXTALSDF	HXTAL stuck detection flag 0: HXTAL stuck is not detected. 1: HXTAL stuck is detected.
1	OFVEIF	GPTIMER output fault or CMP valid edge interrupt flag When a GPTIMER output fault is detected, or a CMP valid edge is detected, the bit is set by hardware. 0: Neither GPTIMER output fault nor CMP valid edge is not detected. 1: GPTIMER output fault or CMP valid edge is detected.
0	INIF	GTOCx_IN input interrupt flag When the GTOCx_IN pin input signal is valid level (INDM=0) or valid edge (INDM=1), the bit is set by hardware. 0: No valid input has been detected. 1: Valid input has been detected.

9.4.2 GTOCx output closing request control register (GTOCx_OCRCTL) (x=0...3)

Address offset: 0x04 + 0x10 * x

Reset value: 0x0000 0000

This register has to be accessed by byte(8-bit), half-word(16-bit) and word(32-bit).



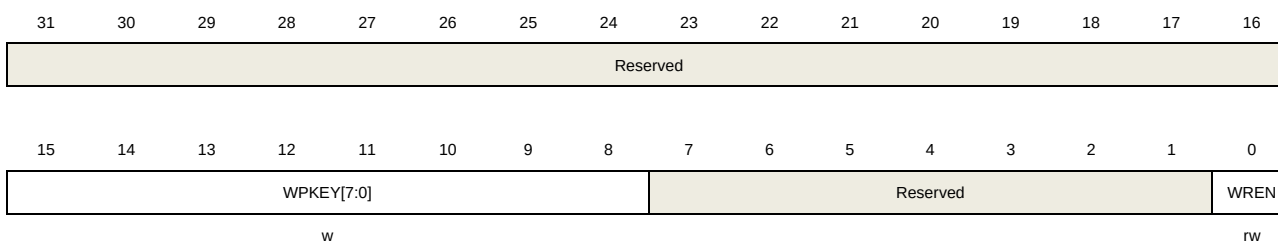
Bits	Fields	Descriptions
31:3	Reserved	Must be kept at reset value
2:0	OCRMKSEL[2:0]	Output closing request mask source selection When the selected output channel is a high level, the output closing request generated by CMP valid edge detection can be masked. 000: Output closing request is not masked. 001: GPTIMER0_CH0. 010: GPTIMER0_CH1. 011: GPTIMER1_CH0. 100: GPTIMER1_CH1. All other values: reserved. Note: This bit-field can be written only once after the reset.

9.4.3 GTOCx write protection register (GTOCx_WP) (x=0...3)

Address offset: 0x08 + 0x10 * x

Reset value: 0x0000 0000

This register has to be accessed by byte(8-bit), half-word(16-bit) and word(32-bit).



Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value
15:8	WPKEY[7:0]	Write protect key These bits determine whether the new value can be written into the GTOCx_WP register. 0xA5 must be written to WPKEY[7:0] while writing other bits, otherwise writing the new value is invalid. These bits are read as 0x00.
7:1	Reserved	Must be kept at reset value
0	WREN	GTOCx_ECRCTL register write enable.

This bit is set to 1 or cleared by software. When it is set to 0, GTOCx_ECRCTL register can be written.

0: GTOCx_ECRCTL register write is enabled.

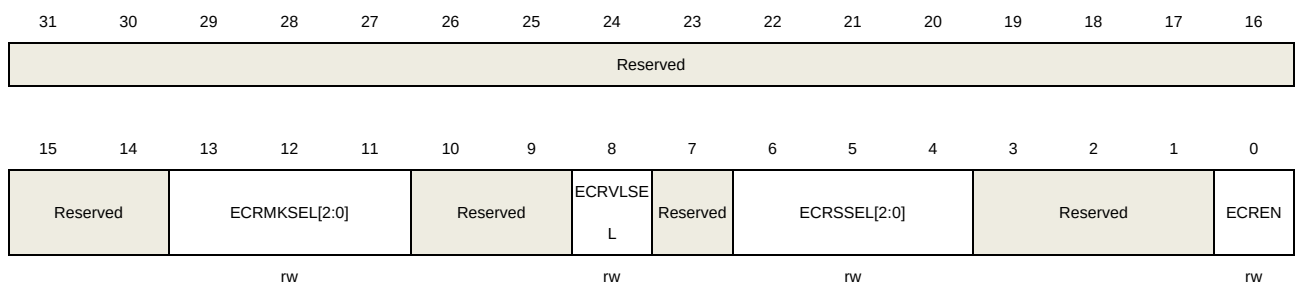
1: GTOCx_ECRCTL register write is disabled.

9.4.4 GTOCx extended closing request control register (GTOCx_ECRCTL) (x=0...3)

Address offset: 0x0C + 0x10 * x

Reset value: 0x0000 0000

This register has to be accessed by byte(8-bit), half-word(16-bit) and word(32-bit).



Bits	Fields	Descriptions
31:14	Reserved	Must be kept at reset value.
13:11	ECRMKSEL[2:0]	Extended closing request mask source selection When the selected output channel is a high level, the extended closing request can be masked. 000: Extended closing request is not masked. 001: GPTIMER0_CH0. 010: GPTIMER0_CH1. 011: GPTIMER1_CH0. 100: GPTIMER1_CH1. All other values: reserved.
10:9	Reserved	Must be kept at reset value.
8	ECRVLSEL	Extended closing request valid level selection When the signal selected by ECRSSEL[2:0] bits become valid level, an extended closing request will be generated. 0: Low level is valid. 1: High level is valid.
7	Reserved	Must be kept at reset value.
6:4	ECRSSEL[2:0]	Extended closing request source selection 000: CMP0 output

		001: CMP1 output 010: CMP2 output 100: CMP3 output 111: GTOCx_IN input All other values: reserved.
3:1	Reserved	Must be kept at reset value.
0	ECREN	Extended closing request enable 0: Disable 1: Enable

10 Cyclic redundancy checks management unit (CRC)

10.1 Overview

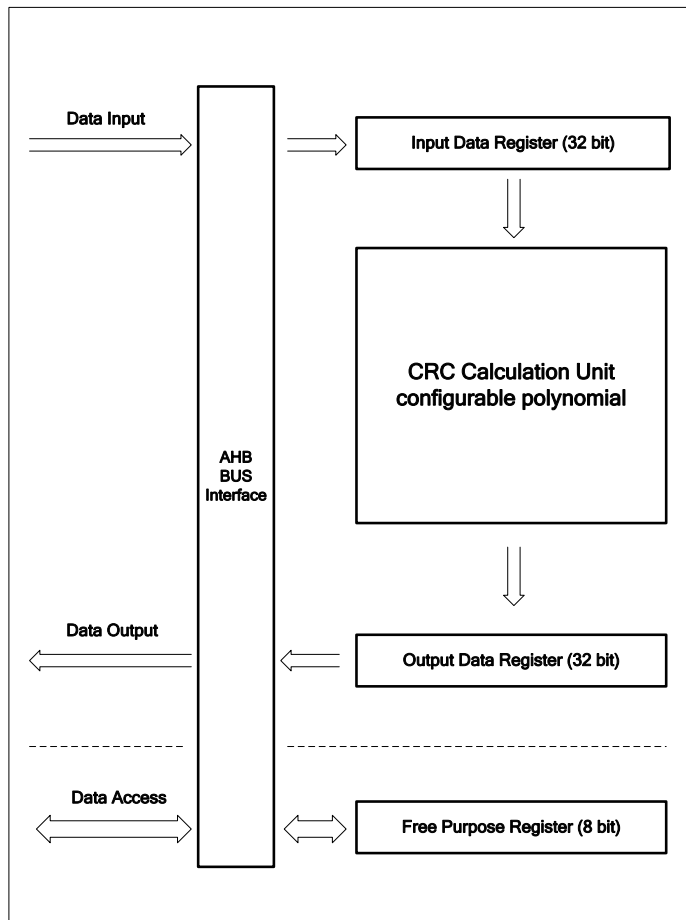
A cyclic redundancy check (CRC) is an error-detecting code commonly used in digital networks and storage devices to detect accidental changes to raw data.

This CRC management unit can be used to calculate 7/8/16/32 bit CRC code within user configurable polynomial.

10.2 Characteristics

- Supports 7/8/16/32 bit data input
- For 7(8)/16/32 bit input data length, the calculation cycles are 1/2/4 AHB clock cycles
- For 32bit input data length, the calculation cycle can be 1 AHB clock cycles (32-bit parallel CRC calculation with two fixed polynomial)
- User configurable polynomial value and size
- After CRC module reset, user can configure initial value
- Free 8-bit register is unrelated to calculation and can be used for any other goals by any other peripheral devices

Figure 10-1. Block diagram of CRC calculation unit



10.3 Function overview

- CRC calculation unit is used to calculate the 32-bit raw data, and CRC_DATA register will receive the raw data and store the calculation result.

If the CRC_DATA register has not been cleared by setting the CRC_CTL register, the new input raw data will be calculated based on the result of previous value of CRC_DATA.

CRC calculation will spend 4/2/1 AHB clock cycles for 32/16/8(7) bit data size. During this period, AHB will not be hanged because of the existence of the 32bit input buffer. For 32bit input data length, the calculation cycle can be 1 AHB clock cycles by configuring PS[1:0]=00 and PAR_32=1 in CRC_CTL.

- This module supplies an 8-bit free register CRC_FDATA.
CRC_FDATA is unrelated to the CRC calculation. Independent read and write operations can be performed at any time.
- Reversible function can reverse the input data and output data.
For input data, 3 reverse types can be selected.

Original data is 0x1A2B3C4D:

1) byte reverse:

32-bit data is divided into 4 groups and reverse implement in group inside. Reversed data: 0x58D43CB2

2) half-word reverse:

32-bit data is divided into 2 groups and reverse implement in group inside. Reversed data: 0xD458B23C

3) word reverse:

32-bit data is divided into 1 groups and reverse implement in group inside. Reversed data: 0xB23CD458

For output data, reverse type is word reverse.

For example: when REV_O=1, calculation result 0x22CC4488 will be converted to 0x11223344..

- User configurable initial calculation data is available.
When RST bit is set or write operation to CRC_IDATA register, the CRC_DATA register will be automatically initialized to the value in CRC_IDATA.
- User configurable polynomial. When 32-bit parallel calculation is select. (PS=2'b00, PAR_32=1), there are only two fixed polynomial.
Depends on PS[1:0] bits, the valid polynomial and output bit width can be selected by user. If the polynomial is less than 32 bit, the high bits of the input data and output data is unavailable. It is strongly recommend resetting the CRC calculation unit before change the PS[1:0] bits、 PAR_32 bit (in CRC_CTL register) or polynomial

10.4 Register definition

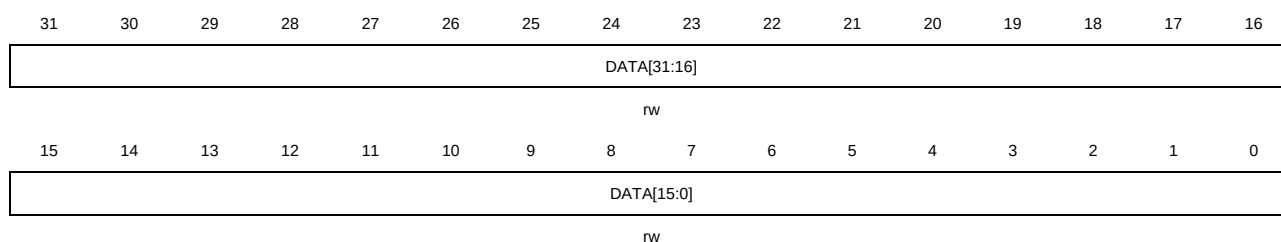
CRC base address: 0x4002 3000

10.4.1 Data register (CRC_DATA)

Address offset: 0x00

Reset value: 0xFFFF FFFF

This register has to be accessed by word (32-bit).



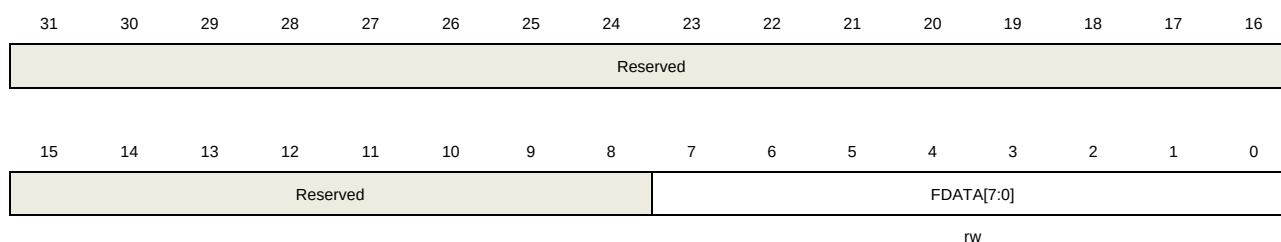
Bits	Fields	Descriptions
31:0	DATA[31:0]	CRC calculation result bits Software writes and reads. This register is used to calculate new data, and the register can be written the new data directly. Write value cannot be read because the read value is the previous CRC calculation result.

10.4.2 Free data register (CRC_FDATA)

Address offset: 0x04

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7:0	FDATA[7:0]	Free data register bits Software writes and reads. These bits are unrelated with CRC calculation. This byte can be used for any goal

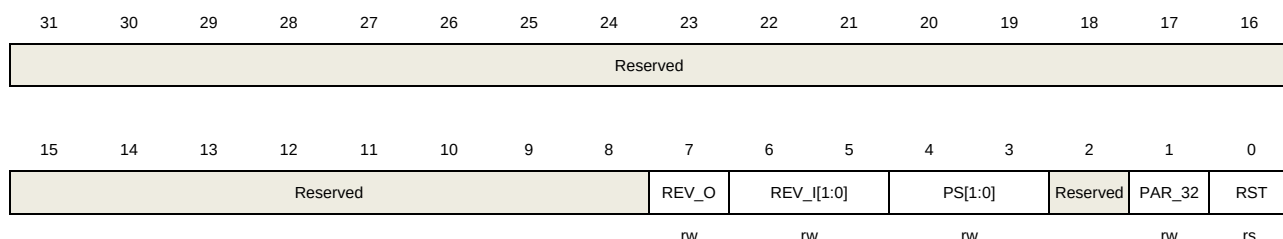
by any other peripheral. The CRC_CTL register will generate no effect to the byte.

10.4.3 Control register (CRC_CTL)

Address offset: 0x08

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7	REV_O	Reverse output data value in bit order 0: Not bit reversed for output data 1: Bit reversed for output data
6:5	REV_I[1:0]	Reverse type for input data 0: Do not use reverse for input data 1: Reverse input data with every 8-bit length 2: Reverse input data with every 16-bit length 3: Reverse input data with whole 32-bit length
4:3	PS[1:0]	Size of polynomial When PAR_32 = 0, using 8-bit CRC parallel calculation unit. 00: 32 bit 01: 16 bit (POLY [15:0] is used for calculation) 10: 8 bit (POLY [7:0] is used for calculation) 11: 7 bit (POLY [6:0] is used for calculation) When PAR_32 = 1, using 32-bit CRC parallel calculation unit with two fixed polynomial. 00: 32-bit CRC parallel calculation with POLY=0x4C11DB7 01: 32-bit CRC parallel calculation with POLY=0x1EDC6F41 10: reserved 11: reserved
2	Reserved	Must be kept at reset value.
1	PAR_32	CRC 32-bit parallel calculation mode. 1: CRC works in 32-bit parallel calculation mode. 0: CRC works in 8-bit parallel calculation mode.

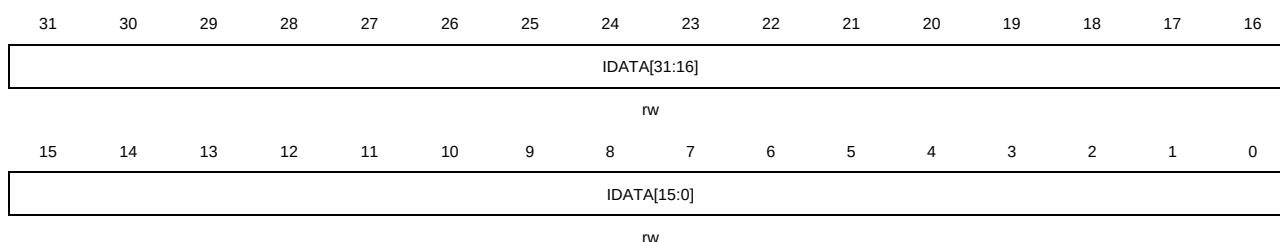
0	RST	Software writes and reads. Set this bit can reset the CRC_DATA and CRC_CTL register. When set, the value of the CRC_DATA register is automatically initialized to the value in the CRC_IDATA register and then RST is automatically cleared by hardware. This bit will take no effect to CRC_FDATA.

10.4.4 Initialization data register (CRC_IDATA)

Address offset: 0x10

Reset value: 0xFFFF FFFF

This register has to be accessed by word (32-bit).



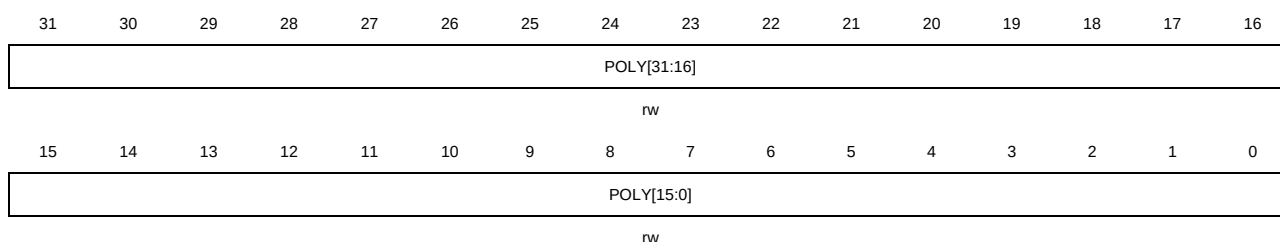
Bits	Fields	Descriptions
31:0	IDATA[31:0]	Configurable initial CRC data value When RST bit in CRC_CTL asserted, CRC_DATA will be programmed to this value.

10.4.5 Polynomial register (CRC_POLY)

Address offset: 0x14

Reset value: 0x04C1 1DB7

This register has to be accessed by word (32-bit).



Bits	Fields	Descriptions
31:0	POLY[31:0]	User configurable polynomial value This value is used together with PS[1:0] bits.

11 Trigonometric Math Unit (TMU)

11.1 Overview

The Trigonometric Math Unit (TMU) is a fully configurable block that execute common trigonometric and arithmetic operations. The TMU can reduce the burden of CPU, and it is usually used in motor control, signal processing and many other applications.

The TMU can be used to calculate total 5 kinds of functions. The input / output data meet q1.31 or q1.15 fixed point format or IEEE754 32-bit single precision floating-point format.

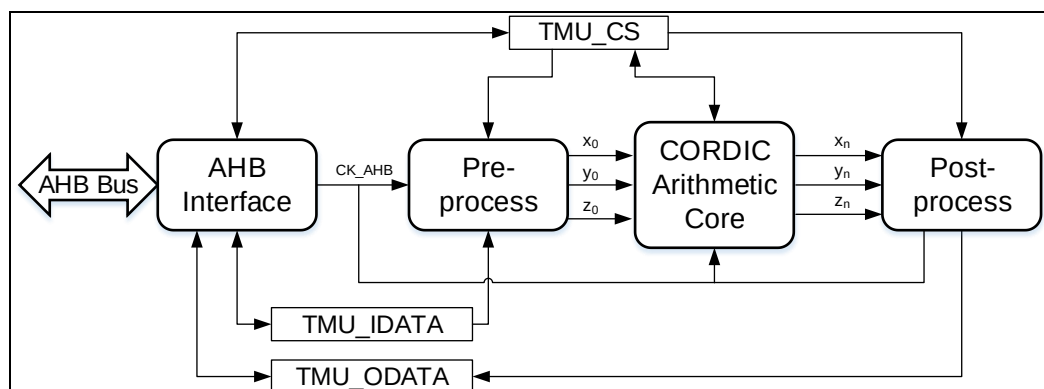
11.2 Characteristics

- 5 kinds of functions.
- Interrupt and DMA requests.
- The fixed point q1.31 / q1.15 format or IEEE754 32-bit single precision floating-point format is configurable.
- Programmable precision.
- CORDIC-algorithm core: circular system and hyperbolic system, rotation pattern and vectoring pattern.
- As an external coprocessor for CPU.

11.3 Block diagram

[Figure 11-1. TMU block diagram](#) provides details of the internal configuration of the TMU.

Figure 11-1. TMU block diagram



The Pre-process module converts the contents in the TMU_IDATA register to obtain the initial data (x_0, y_0, z_0) required by the CORDIC-arithmetic core. The contents of the TMU_IDATA register are in the fixed point q1.31 / q1.15 format or IEEE754 32-bit single precision floating-point format.

After the initial data (x_0, y_0, z_0) is input to the CORDIC-algorithm core, it is iterated and calculated to obtain the (x_n, y_n, z_n) . The CORDIC-algorithm core supports circular system and hyperbolic system, and each system supports rotation pattern and vectoring pattern.

The Post-process module converts and scales the data (x_n, y_n, z_n) and writes the processed results into TMU_ODATA register. The contents of the TMU_ODATA register are in the fixed point q1.31 / q1.15 format or IEEE754 32-bit single precision floating-point format.

11.4 Function overview

11.4.1 Data format and configuration

When the IFLTEN / OFLTEN bit is reset, the input / output data of TMU module is fixed point signed integer format (q1.31 and q1.15 format). When the IFLTEN / OFLTEN bit is set, the input / output data of TMU module is IEEE754 32-bit single precision floating-point format.

In q1.31 format, the 31 bit is sign bit and 0~30 bits are fractional bits. The value range is $[-1, 1 - 2^{-31}]$, corresponding to $[0x80000000, 0x7FFFFFFF]$.

In q1.15 format, the 15 bit is sign bit and 0~14 bits are fractional bits. The value range is $[-1, 1 - 2^{-15}]$, corresponding to $[0x8000, 0x7FFF]$.

In IEEE754 32-bit single precision floating-point format, the value range is shown in [Table 11-3 IEEE754 32-bit single precision floating-point format](#).

The IWIDTH bit in TMU_CS register is used to configure the fixed point format of the input data. Some modes (for example mode 0, $m * \sin(\theta)$) require two input datas, while some modes (for example mode 4, $\sqrt{x}$) require only one input data. The INUM bit in TMU_CS register is used to configure the number of input data. Detailed configuration refer to [Table 11-1. Input data configuration](#).

Note: When the input data is configured in q1.15 format, the TMU_IDATA register only needs to be written once, the first input data in the low half word, the second input data in the high half word. If the mode only needs one input data, only the low half word is used, and the high half word is not used.

Table 11-1. Input data configuration

IWIDTH bit	INUM bit	IFLTEN bit	Data format	Write operation to TMU_IDATA
0	0	0	q1.31 fixed point	Only one write operation
0	1	0	q1.31 fixed point	Two successive write operation
1	0	0	q1.15 fixed point	Only one write operation
1	1	0	q1.15 fixed point	Not available
X	0	1	IEEE754 32-bit single	Only one write operation
X	1	1	precision floating-point	Two successive write operation

The OWIDTH bit in TMU_CS register is used to configure the fixed point format of the output

data. Some modes (for example mode $0, m * \sin(\theta)$) have two output datas, while some modes (for example mode $4, \sqrt{x}$) have only one output data. The ONUM bit in TMU_CS register is used to configure the number of output data. Detailed configuration refer to [Table 11-2. Output data configuration](#).

Note: When the output data is configured in q1.15 format, the TMU_IDATA register only needs to be read once, the first output data in the low half word, the second output data in the high half word. If the mode only needs one output data, only the low half word is used, and the high half word is not used.

Table 11-2. Output data configuration

OWIDTH bit	ONUM bit	OFLTEN bit	Data format	Read operation to TMU_ODATA
0	0	0	q1.31 fixed point	Only one read operation
0	1	0	q1.31 fixed point	Two successive read operation
1	0	0	q1.15 fixed point	Only one read operation
1	1	0	q1.15 fixed point	Not available
X	0	1	IEEE754 32-bit single precision floating-point	Only one read operation
X	1	1		Two successive read operation

11.4.2 Floating-point data format

When the IFLTEN / OFLTEN bit is set, the input / output data of TMU module is IEEE754 32-bit single precision floating-point format, as shown in [Table 11-3 IEEE754 32-bit single precision floating-point format](#).

Table 11-3 IEEE754 32-bit single precision floating-point format

S [31]	E [30:23]	M [22:0]	Value (V)
0	0	0	Zero (V = 0)
1	0	0	Negative Zero (V = -0)
0 + ve 1 - ve	0	non zero	De-normalized ($V = (-1)^s * 2^{(-126)} * (0.M)$)
0 + ve 1 - ve	1 to 254	0 to 0x7FFFFFFF	Normal Range ($V = (-1)^s * 2^{(E-127)} * (1.M)$)
0	254	0x7FFFFFFF	Positive Max (V = +Max)
1	254	0x7FFFFFFF	Negative Max (V = -Max)
0	max=255	0	Positive Infinity (V = +Infinity)
1	max=255	0	Negative Infinity (V = -Infinity)
x	max=255	non zero	Not A Number (V = NaN)

The treatment of the various IEEE floating-point numerical formats for this TMU is given as below:

Negative Zero: All TMU operations generate a positive (S=0, E=0, M=0) zero, never a negative zero if the result of the operation is zero. All TMU operations treat negative zero operations as zero.

De-Normalized Numbers: A de-normalized operand ($E=0$, $M!=0$) input is treated as zero ($E=0$, $M=0$) by all TMU operations.

Not a Number (NaN): An NaN operand ($E=\text{max}$, $M!=0$) input is treated as Infinity ($E=\text{max}$, $M=0$) for all operations.

Overflow: When the IFLTEN and OFLTEN bits are both set, overflow occurs when the input data or operational data exceeds the domain of the corresponding mode. At the same time, the OVRF bit in the TMU_CS register is set to 1.

11.4.3 Mode configuration

The MODE[3:0] bit-field in TMU_CS register is used to configure the mode of the CORDIC-algorithm core. Different modes use different systems (circular or hyperbolic) and different patterns (rotation or vectoring). Detailed configuration refer to [Table 11-4. TMU mode configuration](#). Since the input and output data are in q1.31 or q1.15 format, some modes need to scale the actual input parameters. The FACTOR [2:0] bit-field in the TMU_CS register is used to configure the scaling factor.

Table 11-4. TMU mode configuration

Mode	The first input data	The second input data	The first output data	The second output data	System and Pattern
Mode 0	θ	m	$m \cdot \cos(\theta)$	$m \cdot \sin(\theta)$	Circular, Rotation
Mode 1	θ	m	$m \cdot \sin(\theta)$	$m \cdot \cos(\theta)$	Circular, Rotation
Mode 2	x	y	$\text{atan2}(y, x)$	$\sqrt{x^2 + y^2}$	Circular, Rotation
Mode 3	x	y	$\sqrt{x^2 + y^2}$	$\text{atan2}(y, x)$	Circular, Vectoring
Mode 4	x	None	$\sqrt{x}$	None	Hyperbolic, Vectoring

Mode 0: $m \cdot \cos(\theta)$

Mode 0 calculates the cosine of an angle. This mode takes two input datas and generate two output datas.

When IFLTEN = 1 and OFLTEN = 1, the detailed information refer to [Table 11-5. Mode 0 description, when IFLTEN = 1 and OFLTEN = 1](#).

Table 11-5. Mode 0 description, when IFLTEN = 1 and OFLTEN = 1

Parameter	Range	Description
First input data	$\frac{\theta}{\pi} \in (-2^{24}, 2^{24})$	Overflow when exceed
Second input data	$m \in (-\infty, +\infty)$	Overflow when exceed
First output data	$m \cdot \cos(\theta) \in (-\infty, +\infty)$	Overflow when exceed
Second output data	$m \cdot \sin(\theta) \in (-\infty, +\infty)$	Overflow when exceed

Parameter	Range	Description
FACTOR[2:0]	Not available	keep at reset value 3'b000.

When IFLTEN = 0 or OFLTEN = 0, the detailed information refer to [Table 11-6. Mode 0 description, when IFLTEN = 0 or OFLTEN = 0.](#)

Table 11-6. Mode 0 description, when IFLTEN = 0 or OFLTEN = 0

Parameter	Range	Description
First input data	$\frac{\theta}{\pi} \in [-1,1)$	The angle θ in radians range from $-\pi$ to π . The θ must be divide by π in software to convert it to the range $[-1,1)$, and then it is written to TMU_IDATA register according to the format of q1.31 or q1.15.
Second input data	$m \in [0,1)$	If $0 \leq m < 1$, it is written to TMU_IDATA register according to the format of q1.31 or q1.15. if $m \geq 1$, a scaling must be applied in software to convert it to the range $[-1,1)$, and then it is written to TMU_IDATA register according to the format of q1.31 or q1.15.
First output data	$m * \cos(\theta) \in [-1,1)$	If the previous software has shrunk m , the output data needs to be scaled up to obtain the real result.
Second output data	$m * \sin(\theta) \in [-1,1)$	If the previous software has shrunk m , the output data needs to be scaled up to obtain the real result.
FACTOR[2:0]	Not available	keep at reset value 3'b000.

Note: When IFLTEN = 0 or OFLTEN = 0, if $m > 1$, the software scaling value is optional.

For example, calculating $100 * \cos\left(\frac{\pi}{2}\right)$. When the IFLTEN and OFLTEN bits are reset, and IWIDTH and OWIDTH bits are set, the input and output datas are in q1.15 format. The steps to calculate the function are as follows:

- Software processes the first input parameter $\frac{\pi}{2}$. Angle $\frac{\pi}{2}$ is divided by π : $\frac{\frac{\pi}{2}}{\pi}=0.5$, 0.5 is 0x4000 in q1.15 format. Software processes the first input parameter $\frac{\pi}{2}$. Angle $\frac{\pi}{2}$ is divided by π : $\frac{\frac{\pi}{2}}{\pi}=0.5$, 0.5 is 0x4000 in q1.15 format. Software processes the first input parameter $\frac{\pi}{2}$. Angle $\frac{\pi}{2}$ is divided by π : $\frac{\frac{\pi}{2}}{\pi}=0.5$, 0.5 is 0x4000 in q1.15 format. Software processes the first input parameter $\frac{\pi}{2}$. Angle $\frac{\pi}{2}$ is divided by π : $\frac{\frac{\pi}{2}}{\pi}=0.5$, 0.5 is 0x4000 in q1.15 format.
- Software processes the second input parameter m . Modulus 100 is divided by 128: $\frac{100}{128}=0.78125$. 0.78125 is 0x6400 in q1.15 format. Software processes the second input parameter m . Modulus 100 is divided by 128: $\frac{100}{128}=0.78125$. 0.78125 is 0x6400 in

- q1.15 format. Software processes the second input parameter m . Modulus 100 is divided by 128: $\frac{100}{128}=0.78125$. 0.78125 is 0x6400 in q1.15 format. Software processes the second input parameter m . Modulus 100 is divided by 128: $\frac{100}{128}=0.78125$. 0.78125 is 0x6400 in q1.15 format.
3. The first input data 0x4000 is written into TMU_IDATA. The first input data 0x4000 is written into TMU_IDATA.... The first input data 0x4000 is written into TMU_IDATA. The first input data 0x4000 is written into TMU_IDATA
 4. The second input data 0x6400 is written into TMU_IDATA. Then the TMU calculation starts.. The second input data 0x6400 is written into TMU_IDATA. Then the TMU calculation starts. The second input data 0x6400 is written into TMU_IDATA. Then the TMU calculation starts.. The second input data 0x6400 is written into TMU_IDATA. Then the TMU calculation starts.....
 5. When the ENDF flag is set to 1, reading the TMU_ODATA register can get the first output data: $y_1 = \frac{100}{128} * \cos\left(\frac{\pi}{2}\right)$, reading the TMU_ODATA register again can get the second output data: $y_2 = \frac{100}{128} * \sin\left(\frac{\pi}{2}\right)$. The output data is q1.15 format.
 6. Software processes the result. Since the previous software has shrunk m by 128, the output data needs to be scaled up by 128 to obtain the real result: $100 * \cos\left(\frac{\pi}{2}\right) = 128 * y_1$.

The scaling 128 is used for the input and output data in this example. Of course, other scaling, such as 101, can also be used.

Mode 1: $m * \sin(\theta)$

Mode 1 calculates the sine of an angle. This mode take two input datas and generate two output datas.

When IFLTEN = 1 and OFLTEN = 1, the detailed information refer to [Table 11-7. Mode 1 description, when IFLTEN = 1 and OFLTEN = 1](#).

Table 11-7. Mode 1 description, when IFLTEN = 1 and OFLTEN = 1

Parameter	Range	Description
First input data	$\frac{\theta}{\pi} \in (-2^{24}, 2^{24})$	Overflow when exceed
Second input data	$m \in (-\infty, +\infty)$	Overflow when exceed
First output data	$m * \cos(\theta) \in (-\infty, +\infty)$	Overflow when exceed
Second output data	$m * \sin(\theta) \in (-\infty, +\infty)$	Overflow when exceed
FACTOR[2:0]	Not available	keep at reset value 3'b000.

When IFLTEN = 0 or OFLTEN = 0, the detailed information refer to [Table 11-8. Mode 1 description, when IFLTEN = 0 or OFLTEN = 0](#).

Table 11-8. Mode 1 description, when IFLTEN = 0 or OFLTEN = 0

Parameter	Range	Description
First input data	$\frac{\theta}{\pi} \in [-1, 1)$	The angle θ in radians range from $-\pi$ to π . The θ must be divide by π in software to convert it to the range $[-1, 1)$, and then it is written to TMU_IDATA register according to the format of q1.31 or q1.15.
Second input data	$m \in [0, 1)$	If $0 \leq m < 1$, it is written to TMU_IDATA register according to the format of q1.31 or q1.15. if $m \geq 1$, a scaling must be applied in software to convert it to the range $[-1, 1)$, and then it is written to TMU_IDATA register according to the format of q1.31 or q1.15.
First output data	$m * \sin(\theta) \in [-1, 1)$	If the previous software has shrunk m , the output data needs to be scaled up to obtain the real result.
Second output data	$m * \cos(\theta) \in [-1, 1)$	If the previous software has shrunk m , the output data needs to be scaled up to obtain the real result.
FACTOR[2:0]	Not available	keep at reset value 3'b000.

Note: When IFLTEN = 0 or OFLTEN = 0, if $m > 1$, the software scaling value is optional.

For example, calculating $100 * \cos\left(\frac{\pi}{2}\right)$. When the IFLTEN and OFLTEN bits are reset, and IWIDTH and OWIDTH bits are set, the input and output datas are in q1.15 format. The steps to calculate the function are as follows:

- Software processes the first input parameter $\frac{\pi}{2}$. Angle $\frac{\pi}{2}$ is divided by π : $\frac{\frac{\pi}{2}}{\pi} = 0.5$, 0.5 is 0x4000 in q1.15 format. Software processes the first input parameter $\frac{\pi}{2}$. Angle $\frac{\pi}{2}$ is divided by π : $\frac{\frac{\pi}{2}}{\pi} = 0.5$, 0.5 is 0x4000 in q1.15 format. Software processes the first input parameter $\frac{\pi}{2}$. Angle $\frac{\pi}{2}$ is divided by π : $\frac{\frac{\pi}{2}}{\pi} = 0.5$, 0.5 is 0x4000 in q1.15 format. Software processes the first input parameter $\frac{\pi}{2}$. Angle $\frac{\pi}{2}$ is divided by π : $\frac{\frac{\pi}{2}}{\pi} = 0.5$, 0.5 is 0x4000 in q1.15 format.
- Software processes the second input parameter m . Modulus 100 is divided by 128: $\frac{100}{128} = 0.78125$. 0.78125 is 0x6400 in q1.15 format. Software processes the second input parameter m . Modulus 100 is divided by 128: $\frac{100}{128} = 0.78125$. 0.78125 is 0x6400 in q1.15 format. Software processes the second input parameter m . Modulus 100 is divided by 128: $\frac{100}{128} = 0.78125$. 0.78125 is 0x6400 in q1.15 format. Software processes the second input parameter m . Modulus 100 is divided by 128: $\frac{100}{128} = 0.78125$. 0.78125 is 0x6400 in q1.15 format.

- q1.15 format.
- The first input data 0x4000 is written into TMU_IDATA. The first input data 0x4000 is written into TMU_IDAT. The first input data 0x4000 is written into TMU_IDATA. The first input data 0x4000 is written into TMU_IDAT....
 - The second input data 0x6400 is written into TMU_IDATA. Then the TMU calculation starts. The second input data 0x6400 is written into TMU_IDATA. Then the TMU calculation starts....
 - When the ENDF flag is set to 1, reading the TMU_ODATA register can get the first output data: $y_1 = \frac{100}{128} * \sin\left(\frac{\pi}{2}\right)$, and reading the TMU_ODATA register again can get the second output data: $y_2 = \frac{100}{128} * \cos\left(\frac{\pi}{2}\right)$. The output data is q1.15 format.
 - Software processes the result. Since the previous software has shrunk m by 128, the output data needs to be scaled up by 128 to obtain the real result: $100 * \sin\left(\frac{\pi}{2}\right) = 128 * y_1$.

The scaling 128 is used for the input and output data in this example. Of course, other scaling, such as 101, can also be used.

Mode 2: phase= atan2 (y,x)

Mode 2 calculates the atan2(y,x) of a vector (x,y). This mode takes two input datas and generate two ouput datas.

When IFLTEN = 1 and OFLTEN = 1, the detailed information refer to [Table 11-9. Mode 2 description, when IFLTEN = 1 and OFLTEN = 1](#)

Table 11-9. Mode 2 description, when IFLTEN = 1 and OFLTEN = 1

Parameter	Range	Description
First input data	$2^{-24} < y/x < 2^{24}$	Overflow when exceed
Second input data		
First output data	$\theta \in [-\pi, \pi)$	Angle, Overflow when exceed
Second output data	$m \in [0, +\infty)$	Modulus, $m = \sqrt{x^2 + y^2}$. Overflow when exceed
FACTOR[2:0]	Not available	keep at reset value 3'b000.

When IFLTEN = 0 or OFLTEN = 0, the detailed information refer to [Table 11-10. Mode 2 description, when IFLTEN = 0 or OFLTEN = 0](#)

Table 11-10. Mode 2 description, when IFLTEN = 0 or OFLTEN = 0

Parameter	Range	Description
First input data	$x \in [-1, 1)$	The abscissa value in Cartesian coordinate system. If $x \geq 1$ or $x < -1$, software scaling is required.
Second input data	$y \in [-1, 1)$	The ordinate value in Cartesian coordinate system. If $x \geq 1$ or $x < -1$, software scaling is required.
First output data	$\theta \in [-1, 1)$	Angle, $[-1, 1)$ corresponding $[-\pi, \pi)$. The output data is

Parameter	Range	Description
		multiplied by π to get the real angle value.
Second output data	$m \in [0,1)$	Modulus, $m = \sqrt{x^2+y^2}$. If x and y have been scaled before, the modulus needs to be scaled equally.
FACTOR[2:0]	Not available	keep at reset value 3'b000.

Note: When IFLTEN = 0 or OFLTEN = 0, as long as one of x and y is out of the range $[-1,1)$, or the modulus $\sqrt{x^2+y^2} \geq 1$, x and y need to be scaled at the same scale at the same time to avoid going out of range of data. After scaling x and y in the same scale, the angle corresponding to the coordinates before and after scaling can be kept same.

For example, calculating $\theta = \text{atan}(5,80)$. When the IFLTEN and OFLTEN bits are reset, and IWIDITH and OWIDITH bits are set, the input and output are q1.15 format. The steps to calculate the function are as follows:

1. Software processes the input parameters (5,80). (5,80) is divided by 128. The result is (0.0390625,0.625). The q1.15 format is (0x0500,0x5000).
2. The first input data 0x5000 is written into TMU_IDATA.
3. The second input data 0x0500 is written into TMU_IDATA. Then the TMU calculation starts.
4. When the ENDF flag is set to 1, reading the TMU_ODATA register can get the first output data θ , and reading the TMU_ODATA register again can get the second output data modulus m . The output data is q1.15 format.
5. Software processes the result. The first output data angle θ multiply π to get the real radian. Since the previous software has shrunk the input datas by 128, the second output data m needs to be scaled up by 128 to obtain the real modulus.

The scaling 128 is used for the input datas and the modulus in this example. Of course, other scaling, such as 81, can also be used.

Mode 3: modulus= $\sqrt{x^2+y^2}$

Mode 3 calculates the modulus $\sqrt{x^2+y^2}$ of a vector (x,y) . This mode take two input datas and generate two output datas.

When IFLTEN = 1 and OFLTEN = 1, the detailed information refer to [Table 11-11. Mode 3 description, when IFLTEN = 1 and OFLTEN = 1](#)

Table 11-11. Mode 3 description, when IFLTEN = 1 and OFLTEN = 1

Parameter	Range	Description
First input data	$2^{-24} < y/x < 2^{24}$	Overflow when exceed
Second input data		
First output data	$\theta \in [-\pi, \pi)$	Angle, Overflow when exceed
Second output data	$m \in [0, +\infty)$	Modulus, $m = \sqrt{x^2+y^2}$. Overflow when exceed
FACTOR[2:0]	Not available	keep at reset value 3'b000.

When IFLTEN = 0 or OFLTEN = 0, the detailed information refer to [Table 11-12. Mode 3 description, when IFLTEN = 0 or OFLTEN = 0](#)

Table 11-12. Mode 3 description, when IFLTEN = 0 or OFLTEN = 0

Parameter	Range	Description
First input data	$x \in [-1, 1)$	The abscissa value in Cartesian coordinate system. If $x \geq 1$ or $x < -1$, software scaling is required.
Second input data	$y \in [-1, 1)$	The ordinate value in Cartesian coordinate system. If $x \geq 1$ or $x < -1$, software scaling is required.
First output data	$m \in [0, 1)$	Modulus, $m = \sqrt{x^2 + y^2}$. If x and y have been scaled before, the modulus needs to be scaled equally.
Second output data	$\theta \in [-1, 1)$	Angle, $[-1, 1)$ corresponding $[-\pi, \pi)$. The output data is multiplied by π to get the real angle value.
FACTOR[2:0]	Not available	Keep at reset value 3'b000.

Note: When IFLTEN = 0 or OFLTEN = 0, as long as one of x and y is out of the range $[-1, 1)$, or the modulus $\sqrt{x^2 + y^2} \geq 1$, x and y need to be scaled at the same scale at the same time to avoid going out of range of data. After scaling x and y in the same scale, the angle corresponding to the coordinates before and after scaling can be kept same.

For example, calculating $\sqrt{5^2 + 80^2}$. When the IFLTEN and OFLTEN bits are reset, and IWIDITH and OWIDITH bits are set, the input and output are q1.15 format. The steps to calculate the function are as follows:

1. Software processes the input parameters (5,80). (5,80) is divided by 128. The result is (0.0390625,0.625). The q1.15 format is (0x0500,0x5000).
2. The first input data 0x0500 is written into TMU_IDATA.
3. The second input data 0x5000 is written into TMU_IDATA. Then the TMU calculation starts.
4. When the ENDF flag is set to 1, reading the TMU_ODATA register can get the first output data modulus m , and reading the TMU_ODATA register again can get the second output data θ . The output data is q1.15 format.
5. Software processes the result. Since the previous software has shrunk the input datas by 128, the first output data m needs to be scaled up by 128 to obtain the real modulus. The second output data angle θ multiply π to get the real radian.

The scaling 128 is used for the input datas and the modulus in this example. Of course, other scaling, such as 81, can also be used.

Mode 4: $\sqrt{x}$

Mode 4 calculates calculates the square root of the input parameter x . This mode take one input data and generate one ouput data.

When IFLTEN = 1 and OFLTEN = 1, the detailed information refer to [Table 11-13. Mode 4 description, when IFLTEN = 1 and OFLTEN = 1](#)

Table 11-13. Mode 4 description, when IFLTEN = 1 and OFLTEN = 1

Parameter	Range	Description
Input data	$x \in [0, +\infty)$	Overflow when exceed
Output data	$\sqrt{x} \in [0, +\infty)$	Overflow when exceed
FACTOR[2:0]	Not available	keep at reset value 3'b000

When IFLTEN = 0 or OFLTEN = 0, the detailed information refer to [Table 11-14. Mode 4 description, when IFLTEN = 0 or OFLTEN = 0](#)

Table 11-14. Mode 4 description, when IFLTEN = 0 or OFLTEN = 0

Parameter	Range	Description
Input data	$\frac{x}{2^f} \in [0.027, 0.085]$	$x \in [0.027, 2.34]$, a scaling factor 2^{-f} is applied in software to ensure that $\frac{x}{2^f} < (1 - \frac{1}{2^f})$. Then write $\frac{x}{2^f}$ to TMU_IDATA in q1.15 or q1.31 format.
Output data	$\frac{\sqrt{x}}{2^f} \in [0.04, 1]$	The output data is multiplied 2^f to get the real $\sqrt{x}$.
FACTOR[2:0]	$f \in [0, 2]$	The bit-field FACTOR[2:0] is configured as f

Note: When IFLTEN = 0 or OFLTEN = 0, the scaling factor FACTOR[2:0] needs to be configured.

For example, calculating $\sqrt{2}$. The input and output are q1.15 format. The steps to calculate the function are as follows:

- Software processes the input parameter 2. 2 is divided by 4 ($f = 3'b010$). The result is 0.5 and the q1.15 format is 0x4000.
- The scaling factor $f = 3'b010$ is written into FACTOR[2:0] bit-field in TMU_CS register.
- The input data 0x4000 is written into TMU_IDATA. Then the TMU calculation starts.
- When the ENDF flag is set to 1, reading the TMU_ODATA register can get the output data: $y_1 = \frac{\sqrt{2}}{2^2}$. The output data is q1.15 format.
- Software processes the result. The output data is multiplied 2^2 to get the real value of $\sqrt{2}$.

In order to ensure the accuracy of calculation, it is recommended to use the scaling factors in [Table 11-15. Recommended scaling factors in mode 4](#) for different inputs

Table 11-15. Recommended scaling factors in mode 4

Input Parameter x range	FACTOR[2:0]	Input data x range
$0.027 < x < 0.75$	0	[0.027, 0.75)
$0.75 \leq x < 1.75$	1	[0.375, 0.875)
$1.75 \leq x < 2.341$	2	[0.4375, 0.585)

11.4.4 TMU operation pending

If the TMU operation is ongoing, the further the contents written into TMU_IDATA register will

be suspended. When the TMU operation completes (the result is read and the ENDF flag is cleared), if the number of suspended input data meets the configuration (defined in the suspended TMU_CS register), the TMU module will start a new TMU operation according to the pending configuration and data.

For example, if the configured TMU mode requires two 32-bit input data (IWIDTH = 0, INUM = 1), the TMU operation is started as soon as two input data are written into the TMU_IDATA. If the second input data does not change in the next TMU operation, the INUM bit can be set to 0 at this time. After the previous TMU operation is completed, only one input data is written into TMU_IDATA. The second input data still uses the previous value in the later TMU operation as long as the TMU mode does not change.

Note: After a reset, the second input data is +1 (0x7FFFFFFF).

While an TMU operation is pending, the further contents written into TMU_IDATA or TMU_CS register will cover the original contents. The new TMU contents will be suspended, and the suspension of the original contents will be invalid.

11.4.5 Zero-overhead mode

After a TMU operation starts, the output data register can be read directly. And the bus will automatically insert the waiting cycle before the result is returned. The following steps can be followed:

1. Configure TMU_CS register as needed.
2. Start the TMU operation by written the input data into TMU_IDATA.
3. Configure the next TMU mode as required and write the input data into TMU_IDATA.
4. Read the result from the TMU_ODATA register. The waiting cycles are automatically inserted into the bus. After reading TMU_ODATA operation is completed, the suspended TMU operation will start automatically.
5. Go to step3.

11.4.6 TMU instruction

TMU, as a coprocessor for the M33 core with a coprocessor number of 0, supports the features listed in [Table 11-16. TMU instruction](#).

Table 11-16. TMU instruction

TMU register	TMU number (CRm)	Available instructions	OPCODE	Register access attribute
TMU_CS	0	MCR/MRC	0	rw
TMU_IDATA	1	MCR/MCRR/ MRC/MRRC	0	rw
TMU_ODATA	2	MRC/MRRC	0	r
TMU_IDATA	9	MCR/MCRR	Mode0~4	rw

11.4.7 Interrupt and DMA requests

When ENDF flag is set to 1, if the RIE bit in TMU_CS register is set to 1, an interrupt request is generated. After the ENDF flag is cleared to 0, the interrupt request is also cleared.

If the WDEN bit in TMU_CS register is set to 1 and no TMU operation is pending, DMA request is generated. The number of DMA request is depends on the INUM bit in TMU_CS register. If INUM is 0, only one DMA request is generated. If INUM is 1, two DMA requests are generated.

When ENDF flag is set to 1, if the RDEN bit in TMU_CS register is set to 1, DMA request is generated. The number of DMA request is depends on the ONUM bit in TMU_CS register. If ONUM is 0, only one DMA request is generated. If ONUM is 1, two DMA requests are generated.

11.5 Registers definition

TMU base address: 0x4802 4400

11.5.1 Control and status register (TMU_CS)

Address offset: 0x00

Reset value: 0x0000 0050

This register can be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ENDF	OVRF	OVRIE	Reserved						IWIDTH	OWIDTH	INUM	ONUM	WDEN	RDEN	RIE
r	rc_w1	rw							rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IFLTEN	OFLTEN	Reserved				FACTOR[2:0]				Reserved				MODE[3:0]	
rw	rw					rw								rw	

Bits	Fields	Descriptions
31	ENDF	End of TMU operation flag 0: No TMU operation or TMU operation is ongoing 1: TMU operation ends and the output data has been written into TMU_ODATA register. This bit is set by hardware when TMU operation ends and the output data has been written into TMU_ODATA register. This bit is cleared by reading the TMU_ODATA register (ONUM + 1) times. Note: During this bit is set, no new TMU operation is started.
30	OVRF	The flag of overflow 0: No overflow 1: Overflow

29	OVRIE	Overflow interrupt enable 0: Overflow interrupt disable 1: Overflow interrupt enable
28:23	Reserved	Must be kept at reset value.
22	IWIDTH	Width of input data 0: 32-bit 1: 16-bit This bit decide the data format for input data. If 32-bit is configured, the input data in q1.31 format should be written into TMU_IDATA register. If 16-bit is configured, the input data in q1.15 format should be written into TMU_IDATA register. The first input data is written into the lower half-word of TMU_IDATA register, and the second input data is written into the upper half-word of TMU_IDATA register.
21	OWIDTH	Width of output data 0: 32-bit 1: 16-bit This bit decide the data format for the output data of TMU operation. If 32-bit is configured, the TMU_ODATA contains the output data of TMU operation in q1.31 format. If 16-bit is configured, the TMU_ODATA contains the output data of TMU operation in q1.15 format. The the lower half-word of TMU_IDATA register contains the first output data, and the the upper half-word of TMU_IDATA register contains the second output data.
20	INUM	The number of times that the TMU_IDATA needs to be written 0: One 32-bit write operation. To start a new TMU operation, one 32-bit input data must written into TMU_IDATA register. 1: Two 32-bit write operation. To start a new TMU operation, two 32-bit input data must written into TMU_IDATA register. Note: When the format of input data is q1.15 (IWIDTH=1) and the TMU mode need only one input data (INUM=0), the upper half-word of TMU_IDATA register is unused.
19	ONUM	The number of times that the TMU_ODATA needs to be read 0: One 32-bit read operation. When TMU operation completes, only one 32-bit result is transferred into TMU_ODATA register. Read the TMU_ODATA register once to clear ENDF flag. 1: Two 32-bit read operation. When TMU operation completes, two 32-bit results are transferred into TMU_ODATA register. Read the TMU_ODATA register twice to clear ENDF flag. Note: When OWIDTH=1 (the format of output data is q1.15), only one 32-bit read

		is needed(ONUM=1).
18	WDEN	Enable DMA request to write TMU_IDATA 0: disabled. No DMA request is generated to write TMU_IDATA. 1: enabled. When no TMU operation is pending, the DMA request is generated.
17	RDEN	Enable DMA request to read TMU_ODATA 0: disabled. No DMA request is generated to read TMU_ODATA. 1: enabled. When ENDF is set, the DMA request is generated.
16	RIE	Enable interrupt request to read TMU_ODATA 0: disabled. No interrupt request is generated to read TMU_ODATA. 1: enabled. When ENDF is set, the interrupt request is generated.
15	IFLTEN	Input Floating Point Format Enable 0: Input data support Fixed Point Signed Integer Format Q1.15 or Q1.31 1: Input data support 32-Bit Single Precision Floating-Point Format
14	OFLTEN	Output Floating Point Format Enable 0: output data support Fixed Point Signed Integer Format Q1.15 or Q1.31 1: output data support 32-Bit Single Precision Floating-Point Format
13:11	Reserved	Must be kept at reset value.
10:8	FACTOR[2:0]	Scaling factor This bit-field defines the scaling factor: $2^{\text{FACTOR}[2:0]}$. 000: 2^0 001: 2^1 010: 2^2 ... 110: 2^6 111: 2^7 When the actual input parameter exceeds the specified the input data range $[-1,1)$, it is need to be divide by $2^{\text{FACTOR}[2:0]}$ and the output data is need to be multiplied by $2^{\text{FACTOR}[2:0]}$ to get the actual output result, details as follows: $\text{TMU\_IDATA} = \text{the actual input parameter} / 2^{\text{FACTOR}[2:0]}$ the actual output result = $\text{TMU\_ODATA} * 2^{\text{FACTOR}[2:0]}$. Note: 1. For mode0, mode1, mode2 and mode3, this bit field is recommended to be configured as 3'b000. For mode4, this bit field recommends some configurations for different parameters.
7:4	Reserved	Must be kept at reset value.
3:0	MODE[3:0]	Mode of TMU operation 0000: mode0, $m * \cos(\theta)$ 0001: mode1, $m * \sin(\theta)$ 0010: mode2, $\text{phase} = \tan^2(y,x)$

0011: mode3, modulus = $\sqrt{x^2+y^2}$

0100: mode4, $\sqrt{x}$

0101~1111: reserved

Note:

x,θ: the first input data

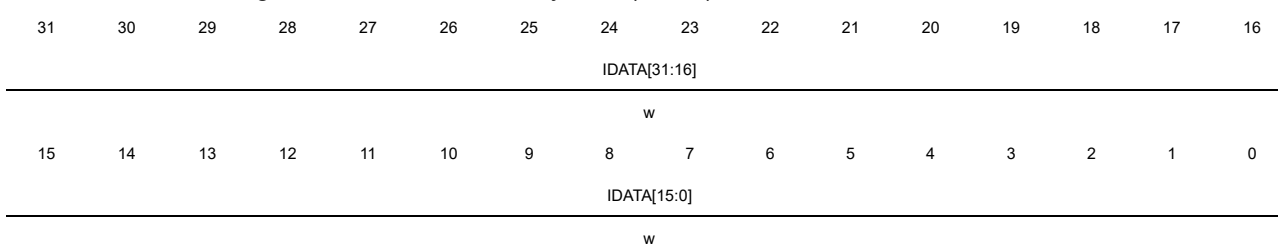
y,m: the second input data

11.5.2 Input data register (TMU_IDATA)

Address offset: 0x04

Reset value: 0xFFFF XXXX

This register can be accessed by word(32-bit).



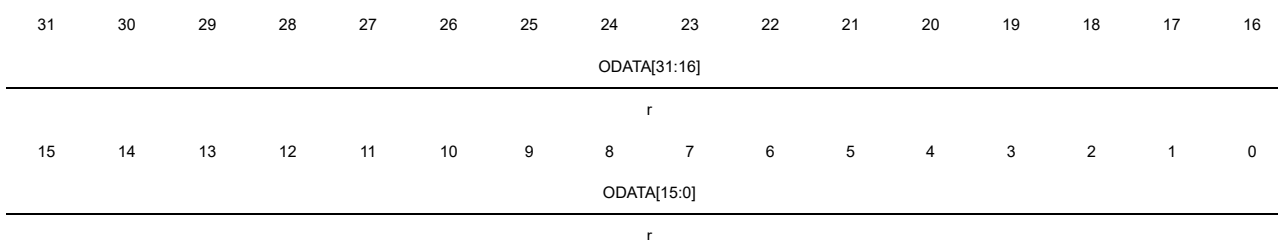
Bits	Fields	Descriptions
31:0	IDATA[31:0]	The input data The input data is written into this register. For details, refer to Table 11-1. Input data configuration. Note: - When no TMU operation is ongoing and the required number of arguments has been written, a new operation will be started automatically. - When the TMU operation is ongoing, the written data is suspended until the end of the TMU operation and the output data is read. During this period, if new input data is written, the previous suspension will be cancelled, and the new data will overwrite the previous data and be suspended.

11.5.3 Output Data register (TMU_ODATA)

Address offset: 0x08

Reset value: 0x0000 0000

This register can be accessed by word(32-bit).



Bits	Fields	Descriptions
31:0	ODATA[31:0]	The output data When TMU operation ends, the result is transferred into this register. For details, refer to Table 11-2. Output data configuration. Note: 1. When the ENDF bit is 1, the result of TMU operation can be obtained by reading the register. 2. When the read operation meeting the configuration is completed, the ENDF bit is cleared.

12 Direct memory access controller (DMA)

12.1 Overview

The direct memory access (DMA) controller provides a hardware method of transferring data between peripherals and/or memory without intervention from the CPU, thereby freeing up bandwidth for other system functions. Data can be quickly moved by DMA between peripherals and memory as well as memory and memory without any CPU actions. There are 12 channels in the DMA controller(6 for DMA0 and 6 for DMA1). Each channel is dedicated to manage memory access requests from one or more peripherals. An arbiter is implemented inside to handle the priority among DMA requests.

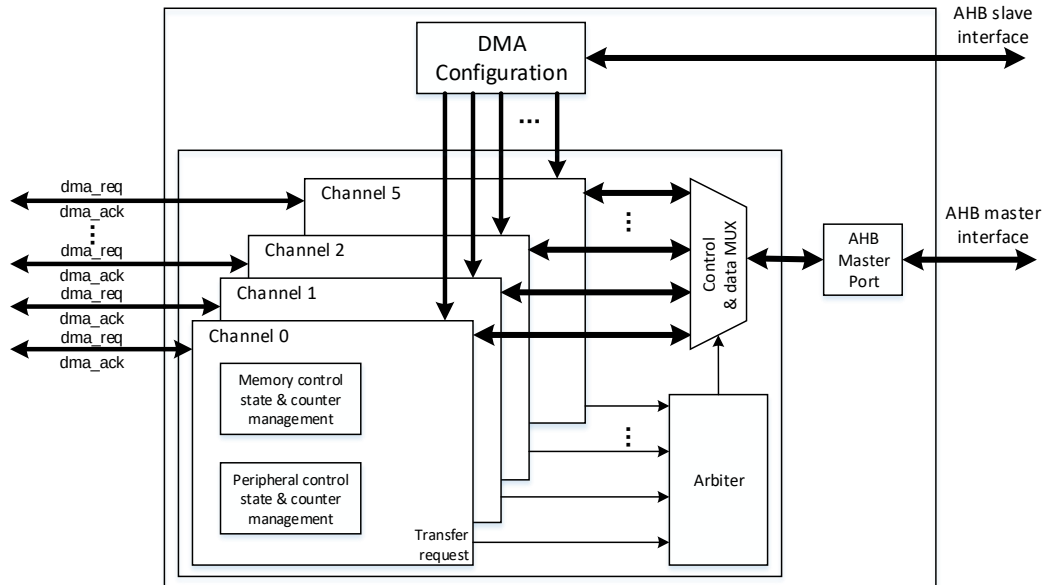
The system bus is shared by the DMA controller and the Cortex®-M33 core. When the DMA and the CPU are targeting the same destination, the DMA access may stop the CPU access to the system bus for some bus cycles. Round-robin scheduling is implemented in the bus matrix to ensure at least half of the system bus bandwidth for the CPU.

12.2 Characteristics

- Programmable length of data to be transferred, max to 65535.
- 12 channels (6 for DMA0 and 6 for DMA1) and each channel are configurable.
- AHB and APB peripherals, FLASH, SRAM can be accessed as source and destination.
- Software DMA channel priority (low, medium, high, ultra high) and hardware DMA channel priority (DMA channel 0 has the highest priority and DMA channel 6 has the lowest priority).
- Support independent 8, 16, 32-bit memory and peripheral transfer.
- Support independent fixed and increasing address generation algorithm of memory and peripheral.
- Support circular transfer mode.
- Support peripheral to memory, memory to peripheral, and memory to memory transfers.
- One separate interrupt per channel with three types of event flags.
- Support interrupt enable and clear.

12.3 Block diagram

Figure 12-1. Block diagram of DMA



As shown in [Figure 12-1. Block diagram of DMA](#), a DMA controller consists of four main parts:

- DMA configuration through AHB slave interface.
- Data transmission through two AHB master interfaces for memory access and peripheral access.
- An arbiter inside to manage multiple peripheral requests coming at the same time.
- Channel management to control address/data selection and data counting.

12.4 Function overview

12.4.1 DMA operation

Each DMA transfer consists of two operations, including the loading of data from the source and the storage of the loaded data to the destination. The source and destination addresses are computed by the DMA controller based on the programmed values in the DMA_CHxPADDR, DMA_CHxMADDR, and DMA_CHxCTL registers. The DMA_CHxCNT register controls how many transfers to be transmitted on the channel. The PWIDTH and MWIDTH bits in the DMA_CHxCTL register determine how many bytes to be transmitted in a transfer.

Suppose DMA_CHxCNT is 4, and both PNAGA and MNAGA are set. The DMA transfer operations for each combination of PWIDTH and MWIDTH are shown in the following [Table 12-1. DMA transfer operation](#).

Table 12-1. DMA transfer operation

Transfer size		Transfer operations	
Source	Destination	Source	Destination
32 bits	32 bits	1: Read B3B2B1B0[31:0] @0x0 2: Read B7B6B5B4[31:0] @0x4 3: Read BBBAB9B8[31:0] @0x8 4: Read BFBEBDBC[31:0] @0xC	1: Write B3B2B1B0[31:0] @0x0 2: Write B7B6B5B4[31:0] @0x4 3: Write BBBAB9B8[31:0] @0x8 4: Write BFBEBDBC[31:0] @0xC
32 bits	16 bits	1: Read B3B2B1B0[31:0] @0x0 2: Read B7B6B5B4[31:0] @0x4 3: Read BBBAB9B8[31:0] @0x8 4: Read BFBEBDBC[31:0] @0xC	1: Write B1B0[15:0] @0x0 2: Write B5B4[15:0] @0x2 3: Write B9B8[15:0] @0x4 4: Write BDBC[15:0] @0x6
32 bits	8 bits	1: Read B3B2B1B0[31:0] @0x0 2: Read B7B6B5B4[31:0] @0x4 3: Read BBBAB9B8[31:0] @0x8 4: Read BFBEBDBC[31:0] @0xC	1: Write B0[7:0] @0x0 2: Write B4[7:0] @0x1 3: Write B8[7:0] @0x2 4: Write BC[7:0] @0x3
16 bits	32 bits	1: Read B1B0[15:0] @0x0 2: Read B3B2[15:0] @0x2 3: Read B5B4[15:0] @0x4 4: Read B7B6[15:0] @0x6	1: Write 0000B1B0[31:0] @0x0 2: Write 0000B3B2[31:0] @0x4 3: Write 0000B5B4[31:0] @0x8 4: Write 0000B7B6[31:0] @0xC
16 bits	16 bits	1: Read B1B0[15:0] @0x0 2: Read B3B2[15:0] @0x2 3: Read B5B4[15:0] @0x4 4: Read B7B6[15:0] @0x6	1: Write B1B0[15:0] @0x0 2: Write B3B2[15:0] @0x2 3: Write B5B4[15:0] @0x4 4: Write B7B6[15:0] @0x6
16 bits	8 bits	1: Read B1B0[15:0] @0x0 2: Read B3B2[15:0] @0x2 3: Read B5B4[15:0] @0x4 4: Read B7B6[15:0] @0x6	1: Write B0[7:0] @0x0 2: Write B2[7:0] @0x1 3: Write B4[7:0] @0x2 4: Write B6[7:0] @0x3
8 bits	32 bits	1: Read B0[7:0] @0x0 2: Read B1[7:0] @0x1 3: Read B2[7:0] @0x2 4: Read B3[7:0] @0x3	1: Write 000000B0[31:0] @0x0 2: Write 000000B1[31:0] @0x4 3: Write 000000B2[31:0] @0x8 4: Write 000000B3[31:0] @0xC
8 bits	16 bits	1: Read B0[7:0] @0x0 2: Read B1[7:0] @0x1 3: Read B2[7:0] @0x2 4: Read B3[7:0] @0x3	1, Write 00B0[15:0] @0x0 2, Write 00B1[15:0] @0x2 3, Write 00B2[15:0] @0x4 4, Write 00B3[15:0] @0x6
8 bits	8 bits	1: Read B0[7:0] @0x0 2: Read B1[7:0] @0x1 3: Read B2[7:0] @0x2 4: Read B3[7:0] @0x3	1, Write B0[7:0] @0x0 2, Write B1[7:0] @0x1 3, Write B2[7:0] @0x2 4, Write B3[7:0] @0x3

The CNT bits in the DMA_CHxCNT register control how many data to be transmitted on the channel and must be configured before enable the CHEN bit in the register. During the transmission, the CNT bits indicate the remaining number of data items to be transferred.

The DMA transmission is disabled by clearing the CHEN bit in the DMA_CHxCTL register.

- If the DMA transmission is not completed when the CHEN bit is cleared, two situations may be occurred when restart this DMA channel:
 - If no register configuration operations of the channel occur before restart the DMA channel, enable CHEN bit ,the DMA will continue to complete the rest of the transmission.
 - If any register configuration operations to DMA_CHxCNT, DMA_CHxPADDR or DMA_CHxMADDR of corresponding channel occur, enable CHEN bit ,the DMA will restart a new transmission.
- If the DMA transmission has been finished when clearing the CHEN bit, enable the DMA channel without any register configuration operation to DMA_CHxCNT, DMA_CHxPADDR or DMA_CHxMADDR of corresponding channel will not launch any DMA transfer.

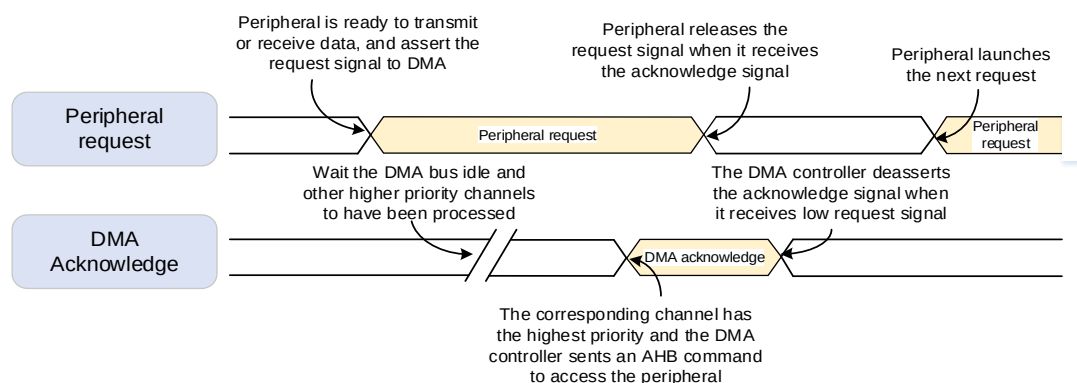
12.4.2 Peripheral handshake

To ensure a well-organized and efficient data transfer, a handshake mechanism is introduced between the DMA and peripherals, including a request signal and an acknowledge signal:

- Request signal asserted by peripheral to DMA controller, indicating that the peripheral is ready to transmit or receive data.
- Acknowledge signal responded by DMA to peripheral, indicating that the DMA controller has initiated an AHB command to access the peripheral.

[Figure 12-2. Handshake mechanism](#) shows how the handshake mechanism works between the DMA controller and peripherals.

Figure 12-2. Handshake mechanism



12.4.3 Arbitration

When two or more requests are received at the same time, the arbiter determines which request is served based on the priorities of channels. There are two-stage priorities, including the software priority and the hardware priority. The arbiter determines which channel is

selected to respond according to the following priority rules:

- Software priority: Four levels, including low, medium, high and ultra-high by configuring the PRIO bits in the DMA_CHxCTL register.
- For channels with equal software priority level, priority is given to the channel with lower channel number.

12.4.4 Address generation

Two kinds of address generation algorithm are implemented independently for memory and peripheral, including the fixed mode and the increased mode. The PNAGA and MNAGA bit in the DMA_CHxCTL register are used to configure the next address generation algorithm of peripheral and memory.

In the fixed mode, the next address is always equal to the base address configured in the base address registers (DMA_CHxPADDR, DMA_CHxMADDR).

In the increasing mode, the next address is equal to the current address plus 1 or 2 or 4, depending on the transfer data width.

12.4.5 Circular mode

Circular mode is implemented to handle continue peripheral requests (for example, ADC scan mode). The circular mode is enabled by setting the CMEN bit in the DMA_CHxCTL register.

In circular mode, the CNT bits are automatically reloaded with the pre-programmed value and the full transfer finish flag is asserted at the end of every DMA transfer. DMA can always respond the peripheral request until the CHEN bit in the DMA_CHxCTL register is cleared.

12.4.6 Memory to memory mode

The memory to memory mode is enabled by setting the M2M bit in the DMA_CHxCTL register. In this mode, the DMA channel can also work without being triggered by a request from a peripheral. The DMA channel starts transferring as soon as it is enabled by setting the CHEN bit in the DMA_CHxCTL register, and completed when the DMA_CHxCNT register reaches zero.

12.4.7 Channel configuration

When starting a new DMA transfer, it is recommended to respect the following steps:

1. Read the CHEN bit and judge whether the channel is enabled or not. If the channel is enabled, clear the CHEN bit by software. When the CHEN bit is read as '0', configuring and starting a new DMA transfer is allowed.
2. Configure the M2M bit and DIR bit in the DMA_CHxCTL register to set the transfer mode.

3. Configure the CMEN bit in the DMA_CHxCTL register to enable/disable the circular mode.
4. Configure the PRIO bits in the DMA_CHxCTL register to set the channel software priority.
5. Configure the memory and peripheral transfer width, memory and peripheral address generation algorithm in the DMA_CHxCTL register.
6. Configure the enable bit for full transfer finish interrupt, half transfer finish interrupt, transfer error interrupt in the DMA_CHxCTL register.
7. Configure the DMA_CHxPADDR register for setting the peripheral base address.
8. Configure the DMA_CHxMADDR register for setting the memory base address.
9. Configure the DMA_CHxCNT register to set the total transfer data number.
10. Configure the CHEN bit with '1' in the DMA_CHxCTL register to enable the channel.

12.4.8 Interrupt

Each DMA channel has a dedicated interrupt. There are three types of interrupt event, including full transfer finish, half transfer finish, and transfer error.

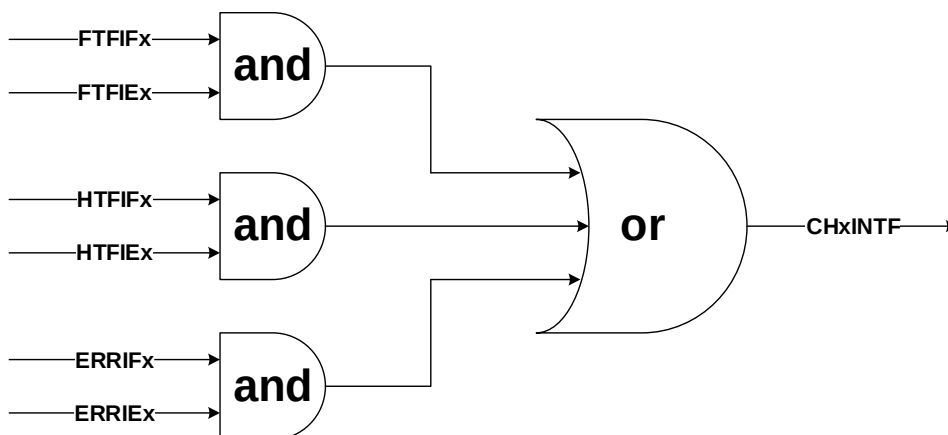
Each interrupt event has a dedicated flag bit in the DMA_INTF register, a dedicated clear bit in the DMA_INTC register, and a dedicated enable bit in the DMA_CHxCTL register. The relationship is described in the following [Table 12-2. interrupt events](#).

Table 12-2. interrupt events

Interrupt event	Flag bit	Clear bit	Enable bit
	DMA_INTF	DMA_INTC	DMA_CHxCTL
Full transfer finish	FTFIF	FTFIFC	FTFIE
Half transfer finish	HTFIF	HTFIFC	HTFIE
Transfer error	ERRIF	ERRIFC	ERRIE

The DMA interrupt logic is shown in the [Figure 12-3. DMA interrupt logic](#), an interrupt can be produced when any type of interrupt event occurs and enabled on the channel.

Figure 12-3. DMA interrupt logic



Note: “x” indicates channel number (x=0...4).

12.4.9 DMA request mapping

The DMA requests of a channel are coming from the AHB/APB peripherals through the corresponding channel output of DMAMUX request multiplexer, refer to [Table 13-3. Request multiplexer input mapping](#).

12.5 Register definition

DMA0 base address: 0x4002 0000

DMA1 base address: 0x4002 0400

12.5.1 Interrupt flag register (DMA_INTF)

Address offset: 0x00

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								ERRIF5	HTFIF5	FTFIF5	GIF5	ERRIF4	HTFIF4	FTFIF4	GIF4
								r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ERRIF3	HTFIF3	FTFIF3	GIF3	ERRIF2	HTFIF2	FTFIF2	GIF2	ERRIF1	HTFIF1	FTFIF1	GIF1	ERRIF0	HTFIF0	FTFIF0	GIF0
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

Bits	Fields	Descriptions
31:20	Reserved	Must be kept at reset value.
23/19/15/11/7	ERRIFx	Error flag of channel x (x=0...5)
	/	Hardware set and software cleared by configuring DMA_INTC register.
	3	0: Transfer error has not occurred on channel x 1: Transfer error has occurred on channel x
22/18/14/10/6	HTFIFx	Half transfer finish flag of channel x (x=0...5)
	/	Hardware set and software cleared by configuring DMA_INTC register.
	2	0: Half number of transfer has not finished on channel x 1: Half number of transfer has finished on channel x
21/17/13/9/5/	FTFIFx	Full Transfer finish flag of channel x (x=0...5)
	1	Hardware set and software cleared by configuring DMA_INTC register.
		0: Transfer has not finished on channel x 1: Transfer has finished on channel x
20/16/12/8/4/	GIFx	Global interrupt flag of channel x (x=0...5)
	0	Hardware set and software cleared by configuring DMA_INTC register.
		0: None of ERRIF, HTFIF or FTFIF occurs on channel x 1: At least one of ERRIF, HTFIF or FTFIF occurs on channel x

12.5.2 Interrupt flag clear register (DMA_INTC)

Address offset: 0x04

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								ERRIFC5	HTFIFC5	FTFIFC5	GIFC5	ERRIFC4	HTFIFC4	FTFIFC4	GIFC4
								w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ERRIFC3	HTFIFC3	FTFIFC3	GIFC3	ERRIFC2	HTFIFC2	FTFIFC2	GIFC2	ERRIFC1	HTFIFC1	FTFIFC1	GIFC1	ERRIFC0	HTFIFC0	FTFIFC0	GIFC0
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

Bits	Fields	Descriptions
31:20	Reserved	Must be kept at reset value.
23/19/15/11/7	ERRIFCx	Clear bit for error flag of channel x (x=0...5)
	/	0: No effect
	3	1: Clear error flag
22/18/14/10/6	HTFIFCx	Clear bit for half transfer finish flag of channel x (x=0...5)
	/	0: No effect
	2	1: Clear half transfer finish flag
21/17/13/9/5	FTFIFCx	Clear bit for full transfer finish flag of channel x (x=0...5)
	1	0: No effect
		1: Clear full transfer finish flag
20/16/12/8/4	GIFCx	Clear global interrupt flag of channel x (x=0...5)
	0	0: No effect
		1: Clear GIFx, ERRIFx, HTFIFx and FTFIFx bits in the DMA_INTF register

12.5.3 Channel x control register (DMA_CHxCTL)

x = 0...5, where x is a channel number

Address offset: 0x08 + 0x14 × x

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	M2M	PRIO[1:0]		MWIDTH[1:0]		PWIDTH[1:0]		MNAGA	PNAGA	CMEN	DIR	ERRIE	HTFIE	FTFIE	CHEN
	rw	rw		rw		rw		rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:15	Reserved	Must be kept at reset value.
14	M2M	Memory to Memory mode Software set and cleared

		0: Disable Memory to Memory mode 1: Enable Memory to Memory mode This bit can not be written when CHEN is '1'.
13:12	PRI0[1:0]	Priority level Software set and cleared 00: Low 01: Medium 10: High 11: Ultra high These bits can not be written when CHEN is '1'.
11:10	MWIDTH[1:0]	Transfer data size of memory Software set and cleared 00: 8-bit 01: 16-bit 10: 32-bit 11: Reserved These bits can not be written when CHEN is '1'.
9:8	PWIDTH[1:0]	Transfer data size of peripheral Software set and cleared 00: 8-bit 01: 16-bit 10: 32-bit 11: Reserved These bits can not be written when CHEN is '1'.
7	MNAGA	Next address generation algorithm of memory Software set and cleared 0: Fixed address mode 1: Increasing address mode This bit can not be written when CHEN is '1'.
6	PNAGA	Next address generation algorithm of peripheral Software set and cleared 0: Fixed address mode 1: Increasing address mode This bit can not be written when CHEN is '1'.
5	CMEN	Circular mode enable Software set and cleared 0: Disable circular mode 1: Enable circular mode This bit can not be written when CHEN is '1'.
4	DIR	Transfer direction

		Software set and cleared 0: Read from peripheral and write to memory 1: Read from memory and write to peripheral This bit can not be written when CHEN is '1'.
3	ERRIE	Enable bit for channel error interrupt Software set and cleared 0: Disable the channel error interrupt 1: Enable the channel error interrupt
2	HTFIE	Enable bit for channel half transfer finish interrupt Software set and cleared 0: Disable channel half transfer finish interrupt 1: Enable channel half transfer finish interrupt
1	FTFIE	Enable bit for channel full transfer finish interrupt Software set and cleared 0: Disable channel full transfer finish interrupt 1: Enable channel full transfer finish interrupt
0	CHEN	Channel enable Software set and cleared 0: Disable channel 1: Enable channel

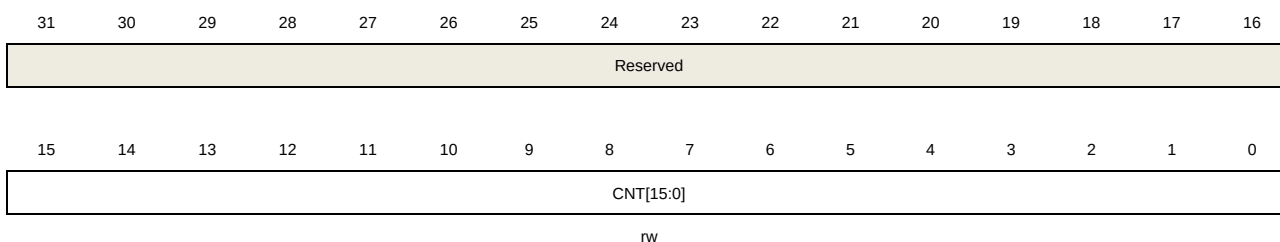
12.5.4 Channel x counter register (DMA_CHxCNT)

$x = 0...5$, where x is a channel number

Address offset: $0x0C + 0x14 \times x$

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CNT[15:0]	Transfer counter These bits can not be written when CHEN in the DMA_CHxCTL register is '1'. This register indicates how many transfers remain. Once the channel is enabled, it is read-only, and decreases after each DMA transfer. If the register is zero, no

transaction can be issued whether the channel is enabled or not. Once the transmission of the channel is complete, the register can be reloaded automatically by the previously programmed value if the channel is configured in circular mode.

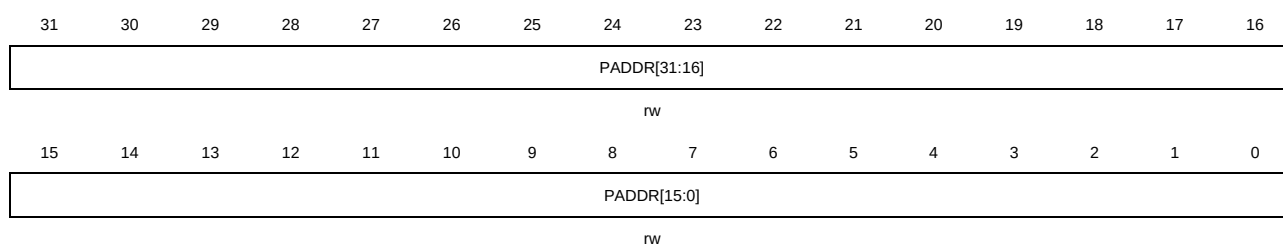
12.5.5 Channel x peripheral base address register (DMA_CHxPADDR)

$x = 0...5$, where x is a channel number

Address offset: $0x10 + 0x14 \times x$

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



Bits	Fields	Descriptions
31:0	PADDR[31:0]	Peripheral base address These bits can not be written when CHEN in the DMA_CHxCTL register is '1'. When PWIDTH is 01 (16-bit), the LSB of these bits is ignored. Access is automatically aligned to a half word address. When PWIDTH is 10 (32-bit), the two LSBs of these bits are ignored. Access is automatically aligned to a word address.

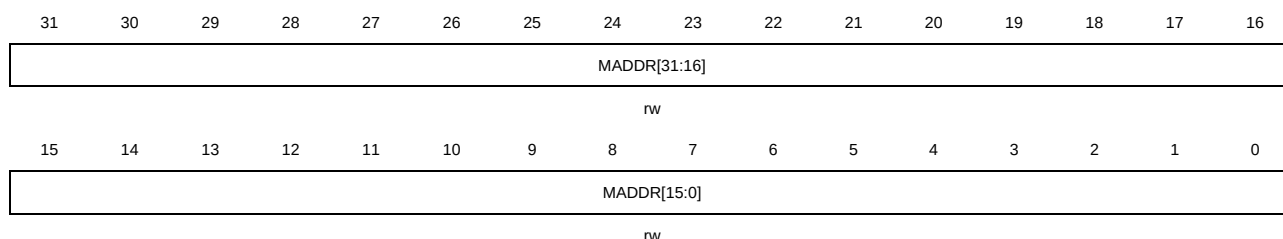
12.5.6 Channel x memory base address register (DMA_CHxMADDR)

$x = 0...5$, where x is a channel number

Address offset: $0x14 + 0x14 \times x$

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



Bits	Fields	Descriptions
31:0	MADDR[31:0]	Memory base address These bits can not be written when CHEN in the DMA_CHxCTL register is '1'.

When MWIDTH in the DMA_CHxCTL register is 01 (16-bit), the LSB of these bits is ignored. Access is automatically aligned to a half word address.

When MWIDTH in the DMA_CHxCTL register is 10 (32-bit), the two LSBs of these bits are ignored. Access is automatically aligned to a word address.

13 DMA request multiplexer (DMAMUX)

13.1 Overview

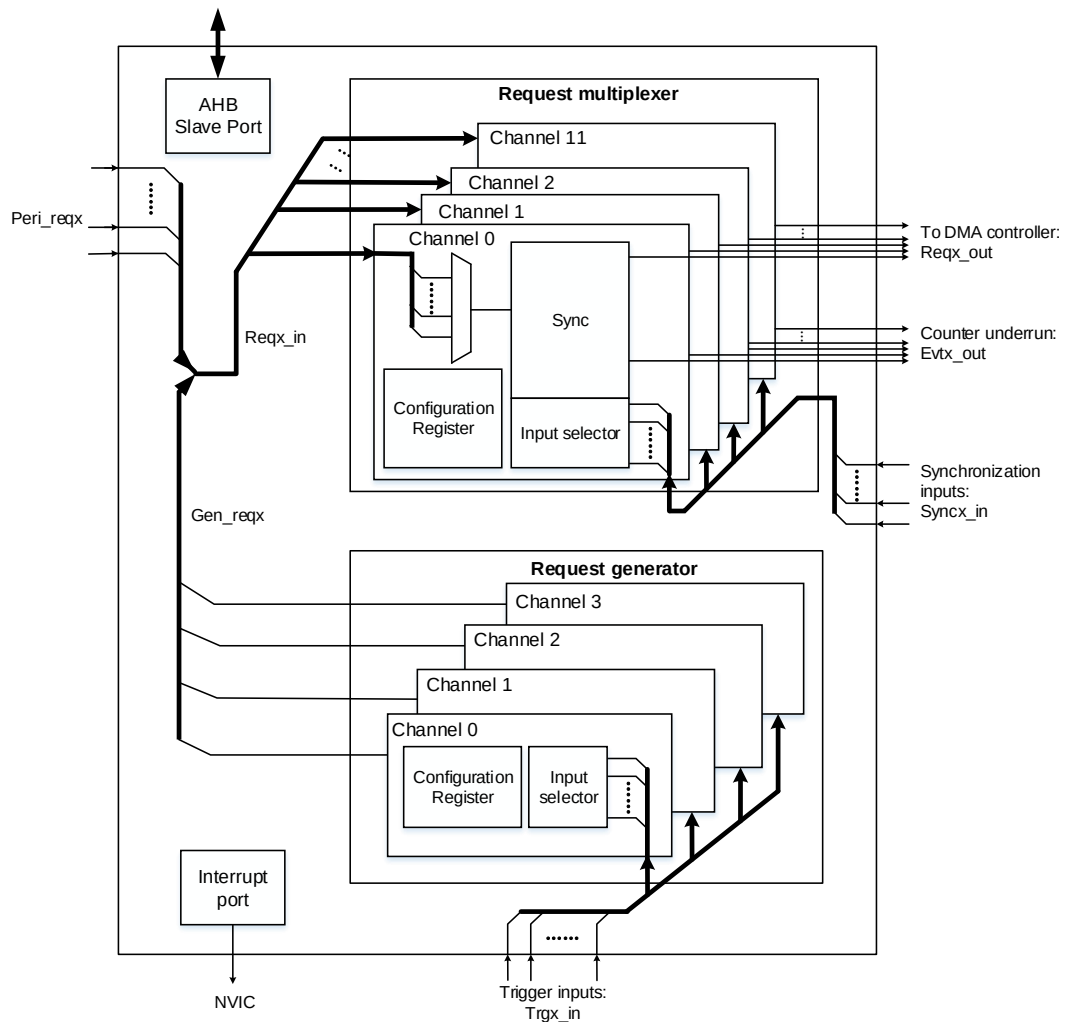
DMAMUX is a transmission scheduler for DMA requests. The DMAMUX request multiplexer is used for routing a DMA request line between the peripherals / generated DMA request (from the DMAMUX request generator) and the DMA controller. Each DMAMUX request multiplexer channel selects a unique DMA request line, unconditionally or synchronously with events from its DMAMUX synchronization inputs. The DMA request is pending until it is served by the DMA controller which generates a DMA acknowledge signal (the DMA request signal is de-asserted).

13.2 Characteristics

- 12 channels for DMAMUX request multiplexer.
- 4 channels for DMAMUX request generator.
- Support 27 trigger inputs.
- Support 27 synchronization inputs.
- Each DMAMUX request generator channel:
 - DMA request trigger input selector
 - DMAMUX request generator counter
 - Trigger overrun flag
- Each DMAMUX request multiplexer channel:
 - 84 input DMA request lines from peripherals
 - Synchronization input selector
 - One DMA request line output
 - One channel event output, for DMA request chaining
 - DMAMUX request multiplexer counter
 - Synchronization overrun flag

13.3 Block diagram

Figure 13-1. Block diagram of DMAMUX



13.4 Function overview

As shown in [Figure 13-1. Block diagram of DMAMUX](#), DMAMUX includes two sub-blocks:

- DMAMUX request multiplexer.

DMAMUX request multiplexer inputs (**Reqx_in**) source from:

- Peripherals (**Peri_reqx**).
- DMAMUX request generator outputs (**Gen_reqx**).

DMAMUX request multiplexer outputs (**Reqx_out**) is connected to channels of DMA controller.

Synchronization inputs (**Syncx_in**) source from internal or external signals.

- DMAMUX request generator.

Trigger inputs (Trgx_in) source from internal or external signals.

13.4.1 DMAMUX signals

Table 13-1. DMAMUX signals

Signal name	Description
Reqx_in	DMAMUX request multiplexer inputs (from peripheral requests and request generator channels)
Peri_reqx	DMAMUX DMA request line inputs from peripherals
Gen_reqx	DMAMUX generated DMA request from request generator
Reqx_out	DMAMUX requests outputs (to DMA controller)
Trgx_in	DMAMUX DMA request triggers inputs (to request generator)
Syncx_in	DMAMUX synchronization inputs (to request multiplexer)
Evtx_out	DMAMUX request multiplexer counter underrun event outputs

13.4.2 DMAMUX request multiplexer

The DMAMUX request multiplexer enables routing a DMA request line between the peripherals / generated DMA request and the DMA controllers of the product. Its component unit is the request multiplexer channels. DMA request lines are connected in parallel to all request multiplexer channels. There is a synchronization unit for each request multiplexer channel. The synchronization inputs are connected in parallel to all synchronization unit of request multiplexer channels. And there is a built-in DMAMUX request multiplexer counter for each request multiplexer channel.

Request multiplexer channel

A DMA request input for the DMAMUX request multiplexer channel x is configured by the MUXID[6:0] bits in the DMAMUX_RM_CHxCFG register, sourced either from the peripherals or from the DMAMUX request generator, the sources can refer to [Table 13-3. Request multiplexer input mapping](#). A DMAMUX request multiplexer channel is connected and dedicated to one single channel of the DMA controller.

Note: The value 0 of MUXID[6:0] bits corresponds to no DMA request line is selected. It is not allowed to configure the same DMA request line (same non-null MUXID[6:0]) to two different request multiplexer channels.

When synchronization mode is disabled

Each time the connected DMAMUX request is served by the DMA controller, the served DMA request is de-asserted, and the built-in DMAMUX request multiplexer counter is decremented. At the request multiplexer counter underrun, the built-in DMAMUX request multiplexer counter is automatically loaded with the value in NBR[4:0] bits of the DMAMUX_RM_CHxCFG register.

If the channel event generation is enabled by setting EVGEN bit, the number of DMA requests before an output event generation is $NBR[4:0] + 1$.

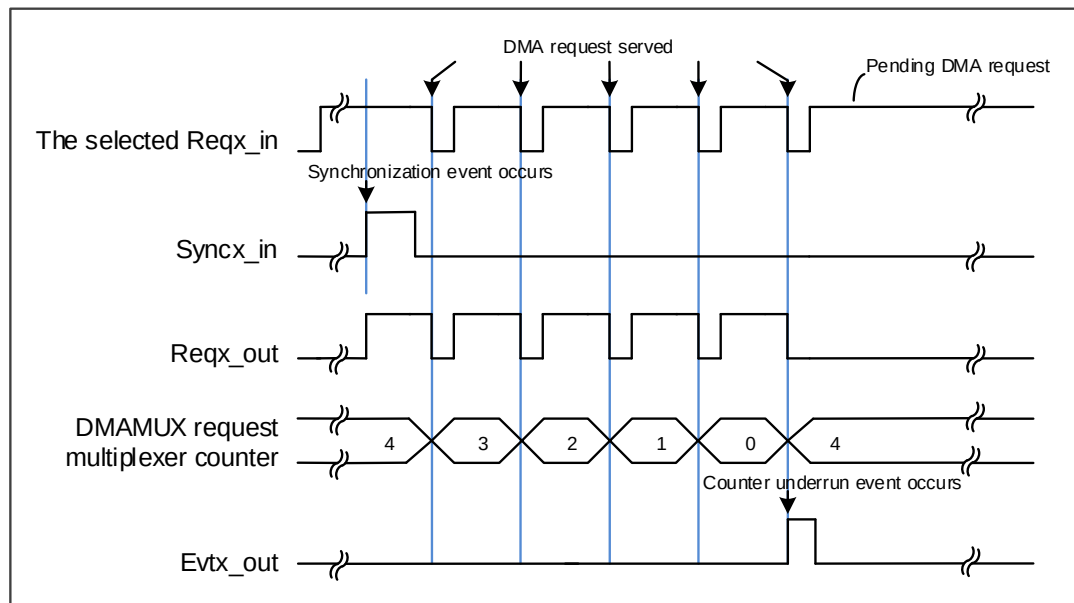
Note: The $NBR[4:0]$ bits value shall only be written by software when both synchronization enable bit SYNCEN and event generation enable EVGEN bit of the corresponding request multiplexer channel x are disabled.

When synchronization mode is enabled

A channel x in synchronization mode, when a rising/falling edge on the selected synchronization input is detected, the pending selected input DMA request line is routed to the multiplexer channel x output. Each time the connected DMAMUX request is served by the DMA controller, the served DMA request is de-asserted, and the built-in DMAMUX request multiplexer counter is decremented. At the request multiplexer counter underrun, the input DMA request line is disconnected from the request multiplexer channel x output, and the built-in DMAMUX request multiplexer counter is automatically loaded with the value in $NBR[4:0]$ bits of the DMAMUX_RM_CHxCFG register. The number of DMA requests transferred to the request multiplexer channel x output following a detected synchronization event is $NBR[4:0] + 1$.

[Figure 13-2. Synchronization mode](#) shows an example when $NBR[4:0]=4$, $SYNCEN=1$, $EVGEN=1$, $SYNCP[1:0]=01$.

Figure 13-2. Synchronization mode



DMAMUX request multiplexer channel x can be synchronized by setting the synchronization enable bit SYNCEN in the DMAMUX_RM_CHxCFG register. The synchronization input is selected by SYNCID[4:0] bits in the DMAMUX_RM_CHxCFG register, the sources can refer to [Table 13-5. Synchronization input mapping](#). The synchronization input valid edge is configured by the SYNCP[1:0] bits of the DMAMUX_RM_CHxCFG register.

Note: If a synchronization input event occurs when there is no pending selected input DMA

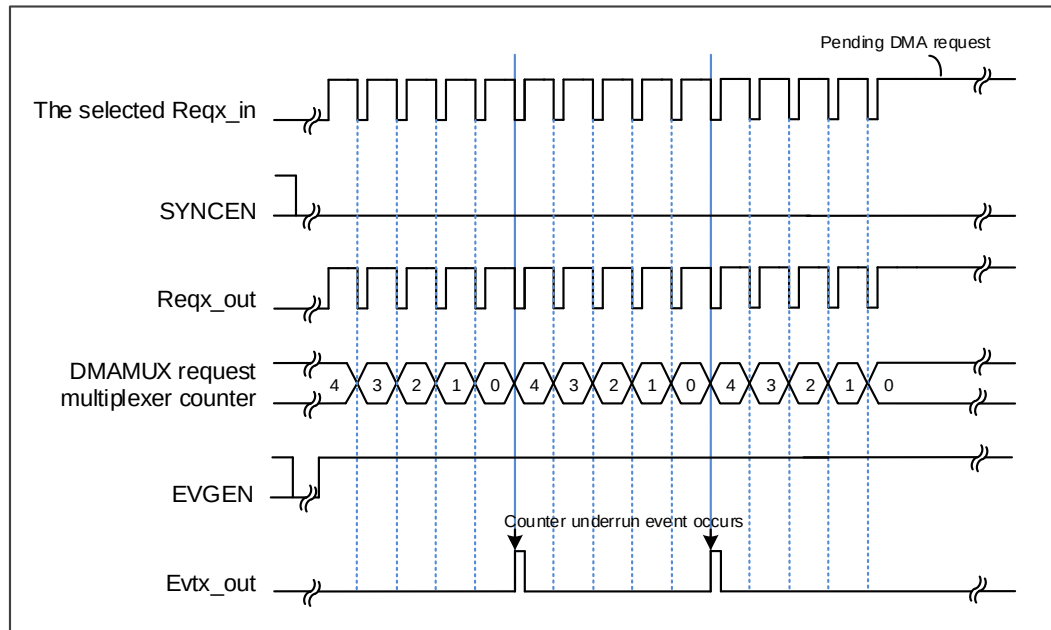
request line, the input event is discarded. The following asserted input request lines will not be routed to the DMAMUX multiplexer channel output until a synchronization input event occurs again.

Channel event generation

Each DMA request line multiplexer channel has an event output called Evtx_out, which is the DMA request multiplexer counter underrun event. Signals dmamux_evt0 ~ dmamux_evt3 can be used for DMA request chaining. If event generation bit EVGEN in the DMAMUX_RM_CHxCFG register is enabled on the channel x output, when its DMA request multiplexer counter is automatically reloaded with the value of the programmed NBR[4:0] field, the multiplexer channel generates a channel event, as a pulse of one AHB clock cycle.

[Figure 13-3. Event generation](#) shows an example when NBR[4:0]=4, SYNCEN=0, EVGEN=1.

Figure 13-3. Event generation



Note: If EVGEN = 1 and NBR[4:0] = 0, an event is generated after each served DMA request.

Synchronization overrun

If a new synchronization event occurs before the built-in DMAMUX request multiplexer counter underrun, the synchronization overrun flag bit SOIFx is set in the DMAMUX_RM_INTF register.

Note: The synchronization mode of request multiplexer channel x shall be disabled by resetting SYNCEN bit in DMAMUX_RM_CHxCFG register at the completion of the use of the related channel of the DMA controller. Otherwise, when a new synchronization event occurs, there will be a synchronization overrun due to the absence of a DMA acknowledge (that is,

no served request) received from the DMA controller.

13.4.3 DMAMUX request generator

The DMAMUX request generator produces DMA requests upon trigger input event. Its component unit is the request generator channels. DMA request trigger inputs are connected in parallel to all request generator channels. And there is a built-in DMAMUX request generator counter for each request generator channel.

The active edge of trigger input events is selected through the RGTP[1:0] bits in DMAMUX_RG_CHxCFG register. The DMA request trigger input for the DMAMUX request generator channel x is selected through the TID[4:0] bits in DMAMUX_RG_CHxCFG register, the sources can refer to [Table 13-4. Trigger input mapping](#). DMAMUX request generator channel x can be enabled by setting RGEN to 1 in DMAMUX_RG_CHxCFG register.

Request generator channel

Upon the trigger input event, the corresponding request generator channel starts generating DMA requests on its output, and the output goes to the input of the DMAMUX request multiplexer. Each time the DMAMUX generated request is served by the connected DMA controller, the served request will be de-asserted, and the built-in DMAMUX request generator counter of the request generator channel is decremented. At the request generator counter underrun, the request generator channel stops generating DMA requests. The built-in DMAMUX request generator counter will be automatically reloaded to its programmed value upon the next trigger input event, the built-in counter is programmed by the NBRG[4:0] bits of the DMAMUX_RG_CHxCFG register.

Note: The number of generated DMA requests after the trigger input event is $NBRG[4:0] + 1$. The NBRG[4:0] value shall only be written by software when the RGEN bit of the corresponding generator channel x is disabled.

Trigger overrun

If a request generator channel x was enabled by RGEN bit, when a new DMA request trigger event for the request generator channel x occurs before the DMAMUX request generator counter underrun, then the request trigger overrun event flag bit TOIFx is set by hardware in the DMAMUX_RG_INTF register.

Note: The request generator channel x shall be disabled by resetting RGEN bit in DMAMUX_RG_CHxCFG register at the completion of the usage of the related channel of the DMA controller. Otherwise, when a new detected trigger event occurs, there will be a trigger overrun due to the absence of an acknowledge (that is, no served request) received from the DMA.

13.4.4 Channel configurations

The following sequence should be followed to configure a DMAMUX channel y and the related DMA channel x:

1. Set and configure the DMA channel x completely, except enabling the channel x.
2. Set and configure the related DMAMUX channel y completely.
3. Configure the CHEN bit with '1' in the DMA_CHxCTL register to enable the DMA channel x.

13.4.5 Interrupt

There are two types of interrupt event, including synchronization overrun event on each DMAMUX request multiplexer channel, and trigger overrun event on each DMAMUX request generator channel.

Each interrupt event has a dedicated flag bit, a dedicated clear bit, and a dedicated enable bit. The relationship is described in the following [Table 13-2. Interrupt events](#).

Table 13-2. Interrupt events

Interrupt event	Flag bit	Clear bit	Enable bit
Synchronization overrun event on DMAMUX request multiplexer channel x	SOIFx	SOIFCx	SOIE
Trigger overrun event on DMAMUX request generator channel y	TOIFy	TOIFCy	TOIE

Trigger overrun interrupt

When the DMAMUX request trigger overrun flag TOIFx is set, and the trigger overrun interrupt is enabled by setting TOIE bit, a trigger overrun interrupt will be generated. The overrun flag TOIFx is reset by writing 1 to the corresponding clear bit of overrun flag TOIFCx in the DMAMUX_RG_INTC register.

Synchronization overrun interrupt

When the synchronization overrun flag SOIFx is set, and the synchronization overrun interrupt is enabled by setting SOIE bit, a synchronization overrun interrupt will be generated. The overrun flag SOIFx is reset by writing 1 to the corresponding clear bit of synchronization overrun flag bit SOIFCx in the DMAMUX_RM_INTC register.

13.4.6 DMAMUX mapping

Request multiplexer input mapping

A DMA request is sourced either from the peripherals or from the DMAMUX request generator, the sources can refer to [Table 13-3. Request multiplexer input mapping](#), configured by the MUXID[6:0] bits in the DMAMUX_RM_CHxCFG register for the DMAMUX request multiplexer channel x.

Table 13-3. Request multiplexer input mapping

Request multiplexer channel input identification MUXID[6:0]	Source
1	Gen_reqx0
2	Gen_reqx1
3	Gen_reqx2
4	Gen_reqx3
5	ADC0_Group_pri1
6	ADC2_Group_pri1
7	ADC0_Group_pri2
8	ADC2_Group_pri2
9	ADC0_Group_pri3
10	ADC2_Group_pri3
11	I2C_RX
12	I2C_TX
13	UART0_RX
14	UART0_TX
15	UART1_RX
16	UART1_TX
17	UART2_RX
18	UART2_TX
19	UART3_RX
20	UART3_TX
21	SPI0_RX
22	SPI0_TX
23	TIMER0_UP
24	TIMER0_CO
25	TIMER0_TI
26	TIMER0_CH0
27	TIMER0_CH1
28	TIMER0_CH2
29	TIMER0_CH3
30	TIMER0_MCH0
31	TIMER0_MCH1
32	TIMER0_MCH2
33	TIMER0_MCH3
34	TIMER7_UP

Request multiplexer channel input identification MUXID[6:0]	Source
35	TIMER7_CO
36	TIMER7_TI
37	TIMER7_CH0
38	TIMER7_CH1
39	TIMER7_CH2
40	TIMER7_CH3
41	TIMER7_MCH0
42	TIMER7_MCH1
43	TIMER7_MCH2
44	TIMER7_MCH3
45	TIMER1_UP
46	TIMER1_TI
47	TIMER1_CH0
48	TIMER1_CH1
49	TIMER1_CH2
50	TIMER1_CH3
51	TIMER2_UP
52	TIMER2_TI
53	TIMER2_CH0
54	TIMER2_CH1
55	TIMER2_CH2
56	TIMER2_CH3
57	TMU_OUTPUT
58	TMU_INPUT
59	GPTIMER0_UP
60	GPTIMER0_TI
61	GPTIMER0_CH0
62	GPTIMER0_CH1
63	GPTIMER1_UP
64	GPTIMER1_TI
65	GPTIMER1_CH0
66	GPTIMER1_CH1
67	EVIC0
68	EVIC1
69	EVIC2
70	EVIC3
71	EVIC4
72	EVIC5
73	EVIC6

Request multiplexer channel input identification MUXID[6:0]	Source
74	EVIC7
75	EVIC8
76	EVIC9
77	ADC0_Group_pri4
78	ADC2_Group_pri4
79	GPTIMER0_CH0COMV_ADD
80	GPTIMER0_CH1COMV_ADD
81	GPTIMER1_CH0COMV_ADD
82	GPTIMER1_CH1COMV_ADD
83	EVIC10
84	EVIC11

Trigger input mapping

The DMA request trigger input for the DMAMUX request generator channel x is selected through the TID[4:0] bits in DMAMUX_RG_CHxCFG register, the sources can refer to [Table 13-4. Trigger input mapping](#).

Table 13-4. Trigger input mapping

Trigger input identification TID[4:0]	Source
0	EXTI_0
1	EXTI_1
2	EXTI_2
3	EXTI_3
4	EXTI_4
5	EXTI_5
6	EXTI_6
7	EXTI_7
8	EXTI_8
9	EXTI_9
10	EXTI_10
11	EXTI_11
12	EXTI_12
13	EXTI_13
14	EXTI_14
15	EXTI_15
16	Evtx_out0
17	Evtx_out1
18	Evtx_out2

Trigger input identification TID[4:0]	Source
19	Evtx_out3
20	Reserved
21	Reserved
22	Reserved
23	Reserved
24	Reserved
25	Reserved
26	Reserved

Synchronization input mapping

The synchronization input is selected by SYNCID[4:0] bits in the DMAMUX_RM_CHxCFG register, the sources can refer to [Table 13-5. Synchronization input mapping](#).

Table 13-5. Synchronization input mapping

Synchronization input identification SYNCID[4:0]	Source
0	EXTI_0
1	EXTI_1
2	EXTI_2
3	EXTI_3
4	EXTI_4
5	EXTI_5
6	EXTI_6
7	EXTI_7
8	EXTI_8
9	EXTI_9
10	EXTI_10
11	EXTI_11
12	EXTI_12
13	EXTI_13
14	EXTI_14
15	EXTI_15
16	Evtx_out0
17	Evtx_out1
18	Evtx_out2
19	Evtx_out3
20	Reserved
21	Reserved
22	Reserved
23	Reserved

Synchronization input identification SYNCID[4:0]		Source
24		Reserved
25		Reserved
26		Reserved

13.5 Register definition

DMAMUX base address: 0x4002 0800

13.5.1 Request multiplexer channel x configuration register (DMAMUX_RM_CHxCFG)

x = 0...11, where x is a channel number

Address offset: 0x00 + 0x04 × x

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved			SYNCID[4:0]				NBR[4:0]				SYNCP[1:0]		SYNCEN		
			rw				rw				rw		rw		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						EVGEN		SOIE	Reserved	MUXID[6:0]					
						rw		rw		rw					

Bits	Fields	Descriptions
31:29	Reserved	Must be kept at reset value.
28:24	SYNCID[4:0]	Synchronization input identification Selects the synchronization input source.
23:19	NBR[4:0]	Number of DMA requests to forward The number of DMA requests to forward to the DMA controller after a synchronization event / before an output event is generated equals to NBR[4:0] + 1. These bits shall only be written when both SYNCEN and EVGEN bits are disabled.
18:17	SYNCP[1:0]	Synchronization input polarity 00: No event detection 01: Rising edge 10: Falling edge 11: Rising and falling edges
16	SYNCEN	Synchronization enable 0: Disable synchronization 1: Enable synchronization
15:10	Reserved	Must be kept at reset value.
9	EVGEN	Event generation enable 0: Disable event generation

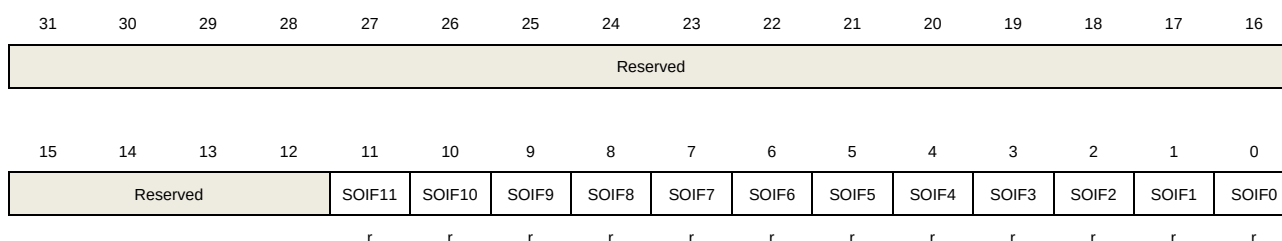
		1: Enable event generation
8	SOIE	Synchronization overrun interrupt enable 0: Disable interrupt 1: Enable interrupt
7	Reserved	Must be kept at reset value.
6:0	MUXID[6:0]	Multiplexer input identification Selects the input DMA request in multiplexer input sources.

13.5.2 Request multiplexer channel interrupt flag register (DMAMUX_RM_INTF)

Address offset: 0x80

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



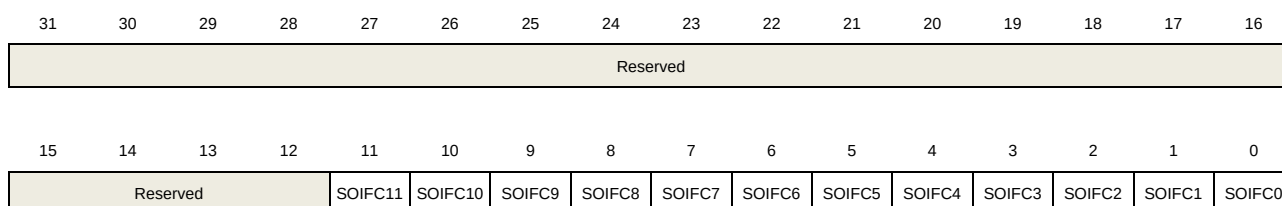
Bits	Fields	Descriptions
31:12	Reserved	Must be kept at reset value.
11:0	SOIFx	Synchronization overrun event flag of request multiplexer channel x (x=0..11) The flag is set when a synchronization event occurs on a DMA request line multiplexer channel x, while the DMA request counter value is lower than NBR[4:0]. The flag is cleared by writing 1 to the corresponding SOIFCx bit in DMAMUX_RM_INTC register.

13.5.3 Request multiplexer channel interrupt flag clear register (DMAMUX_RM_INTC)

Address offset: 0x084

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



Bits	Fields	Descriptions
31:12	Reserved	Must be kept at reset value.
11:0	SOIFCx	Clear bit for synchronization overrun event flag of request multiplexer channel x (x=0..11) Writing 1 clears the corresponding overrun flag SOIFx in the DMAMUX_RM_INTF register.

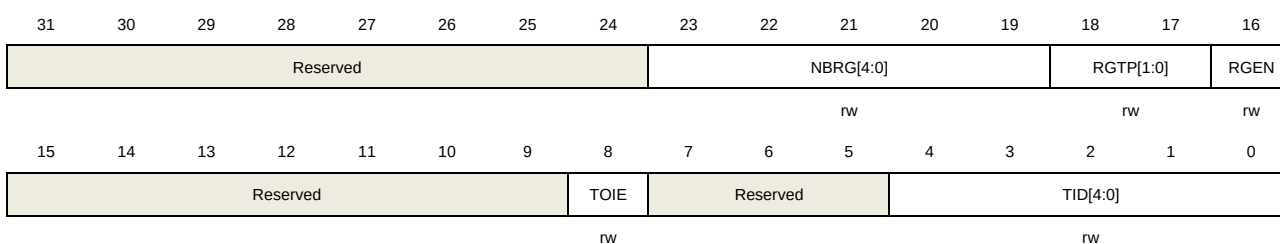
13.5.4 Request generator channel x configuration register (DMAMUX_RG_CHxCFG)

x = 0...3, where x is a channel number

Address offset: 0x100 + 0x04 × x

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



Bits	Fields	Descriptions
31:24	Reserved	Must be kept at reset value.
23:19	NBRG[4:0]	Number of DMA requests to be generated The number of DMA requests to be generated after a trigger event equals to NBRG[4:0] + 1. Note: These bits shall only be written when RGEN bit is disabled.
18:17	RGTP[1:0]	DMA request generator trigger polarity 00: No event trigger detection 01: Rising edge 10: Falling edge 11: Rising and falling edges
16	RGEN	DMA request generator channel x enable 0: Disable DMA request generator channel x 1: Enable DMA request generator channel x
15:9	Reserved	Must be kept at reset value.

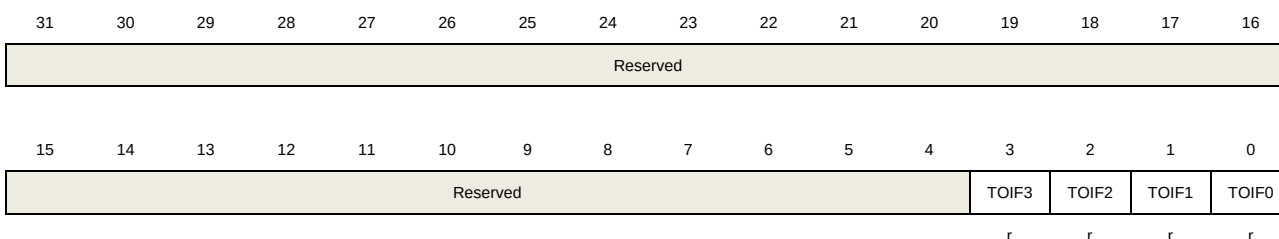
8	TOIE	Trigger overrun interrupt enable 0: Disable interrupt 1: Enable interrupt
7:5	Reserved	Must be kept at reset value.
4:0	TID[4:0]	Trigger input identification Selects the DMA request trigger input source.

13.5.5 Request generator interrupt flag register (DMAMUX_RG_INTF)

Address offset: 0x140

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



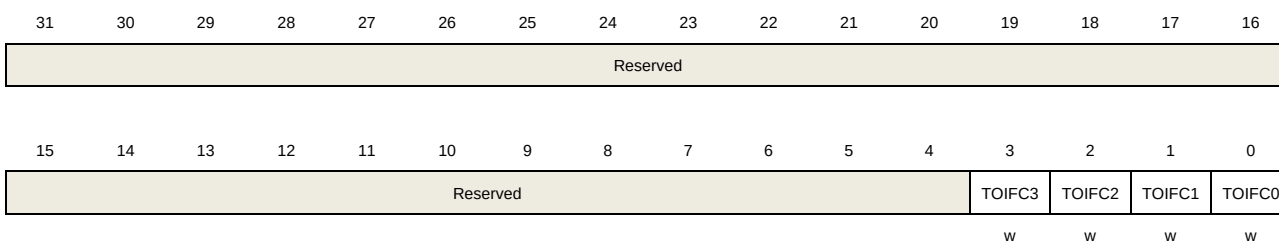
Bits	Fields	Descriptions
31:4	Reserved	Must be kept at reset value.
3:0	TOIFx	Trigger overrun event flag of request generator channel x (x=0..3) The flag is set when a new trigger event occurs on DMA request generator channel x, before the request counter underrun (the internal request counter programmed via the NBRG[4:0] bits of the DMAMUX_RG_CHxCFG register). The flag is cleared by writing 1 to the corresponding TOIFC _x bit in the DMAMUX_RG_INTC register.

13.5.6 Request generator interrupt flag clear register (DMAMUX_RG_INTC)

Address offset: 0x144

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



Bits	Fields	Descriptions
31:4	Reserved	Must be kept at reset value.
3:0	TOIFCx	Clear bit for trigger overrun event flag of request generator channel x (x=0..3) Writing 1 in each bit clears the corresponding overrun flag TOIFx in the DMAMUX_RG_INTF register.

14 Debug (DBG)

14.1 Overview

The GD32M531 series provide a large variety of debug, trace and test features. They are implemented with a standard configuration of the ARM® CoreSight™ module together with a daisy chained standard TAP controller. Debug and trace functions are integrated into the ARM® Cortex®-M33. The debug system supports serial wire debug (SWD), trace functions and standard JTAG debug. The debug system also include one-line interfaces, which is support by GD-Link. The debug and trace functions refer to the following documents.

Cortex®-M33 Technical Reference Manual.

The DBG hold unit helps debugger to debug in power saving mode. When corresponding bit is set, it provides a clock in power saving mode or holds the state for TIMER, I2C, WWDGT, FWDGT or CAN.

14.2 JTAG/SW function overview

Debug capabilities can be accessed by a debug tool via Serial Wire (SW - Debug Port) or JTAG interface (JTAG - Debug Port).

14.2.1 Switch JTAG or SW interface

By default, the JTAG interface is active. The sequence for switching from JTAG to SWD is as following.

Send 50 or more TCK cycles with TMS = 1.
Send the 16-bit sequence on TMS = 1110011110011110 (0xE79E LSB first).
Send 50 or more TCK cycles with TMS = 1.

The sequence for switching from SWD to JTAG is as following.

Send 50 or more TCK cycles with TMS = 1.
Send the 16-bit sequence on TMS = 11100111100111100 (0xE73C LSB first).
Send 50 or more TCK cycles with TMS = 1.

14.2.2 Pin assignment

The JTAG interface provides a 5-pin standard JTAG, known as JTAG clock (JTCK), JTAG mode selection (JTMS), JTAG data input (JTDI), JTAG data output (JTDO) and JTAG reset (NJTRST, active low). The serial wire debug (SWD) provides a 2-pin SW interface, known as SW data input/output (SWDIO) and SW clock (SWCLK). The two SW pins are multiplexed with two of five JTAG pins, which are SWDIO multiplexed with JTMS, SWCLK multiplexed with JTCK. The JTDO is also used as Trace async data output (TRACESWO) when the async trace is enabled.

The pin assignment is as following.

Table 14-1. Pin assignment

Pin (LFQFP64 with PN7 driven high)	Debug interface	Pin (LFQFP48)	Debug interface
PG13	JTDI	-	-
PG12	JTCK/SWCLK	PN2	SWCLK
PG14	JTMS/SWDIO	PG13	SWDIO
PG15	NJTRST	-	-
PG11	JTDO	-	-
BOOT_PN6	one-line	BOOT_PN6	one-line

NOTE: When package is LFQFP48, these pins are dedicated as debug pins only if SWDMOD bit in option byte is 1.

By default, 5-pin standard JTAG debug mode is chosen after reset. User can also use JTAG function without NJTRST pin. (NJTRST tied to 1 by hardware). If it is switched to SW debug mode, the JTDI/JTDO/NJTRST pin are released to other GPIO functions. If JTAG and SW are not used, all 5-pin can be released to other GPIO functions. Please refer to [General-purpose and alternate-function I/Os \(GPIO and AFIO\)](#)

14.2.3 JTAG daisy chained structure

The Cortex®-M33 JTAG TAP is connected to a MCU JTAG TAP. The MCU JTAG IR is of 5-bit width, while the Cortex®-M33 JTAG IR is of 4-bit width. So when JTAG is in IR shift step, it first shifts 5-bit BYPASS instruction (5'b11111) for MCU JTAG and then shifts normal 4-bit instruction for Cortex®-M33 JTAG. Because of the data shift under MCU JTAG BYPASS mode, adding 1 extra bit to the data chain is needed.

The MCU JTAG IDCODE is 0x790007A3.

14.2.4 Debug reset

The JTAG-DP and SW-DP registers are in the Power On Reset domain. The system reset initializes the majority of the Cortex®-M33, excluding NVIC and debug logic, (FPB, DWT, and ITM). The NJTRST reset can reset JTAG TAP controller only. So, debug feature can be performed under system reset. Such as halt-after-reset, it is that the debugger sets halt under system reset, and the core halts immediately after the system reset is released.

14.2.5 JEDEC-106 ID code

The Cortex®-M33 integrates JEDEC-106 ID code, which is located in ROM table and mapped to the address of 0xE00FF000_0xE00FFFFF.

14.3 Debug hold function overview

14.3.1 Debug support for power saving mode

When the STB_HOLD bit in DBG control register (DBG_CTL) is set, and entering the standby mode, the clock of AHB bus and system clock are provided by CK_IRC32M, and the debugger can debug in standby mode. When exiting the standby mode, a system reset is generated.

When the DSLP_HOLD bit in DBG control register (DBG_CTL) is set, and entering the Deep-sleep mode, the clock of AHB bus and system clock are provided by CK_IRC32M, and the debugger can debug in Deep-sleep mode.

When the SLP_HOLD bit in DBG control register (DBG_CTL) is set, and entering the sleep mode, the clock of AHB bus for CPU is not closed, and the debugger can debug in sleep mode.

14.3.2 Debug support for TIMER, I2C, WWDGT, FWDGT and CAN

When the core is halted and the corresponding bit in DBG control register (DBG_CTL) is set, the following events occur.

For TIMER, the timer counters are stopped and held for debugging.

For I2C, SMBUS timeout is held for debugging.

For WWDGT or FWDGT, the counter clock is stopped for debugging.

For CAN, the receive register is stopped counting for debugging.

14.4 Register definition

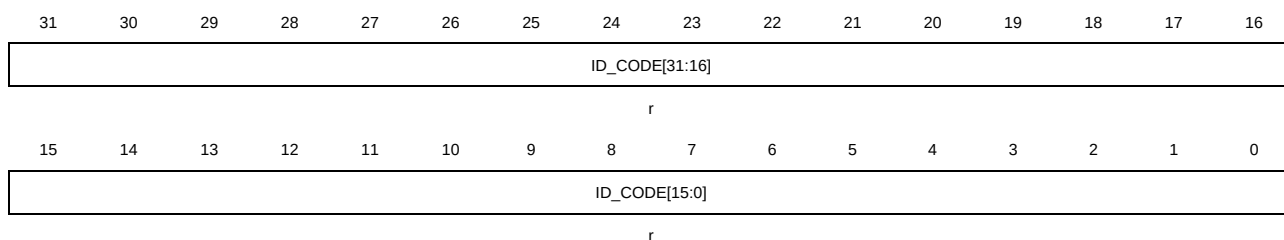
DBG base address: 0xE004 4000

14.4.1 ID code register (DBG_ID)

Address offset: 0x00

Read only

This register has to be accessed by word(32-bit)



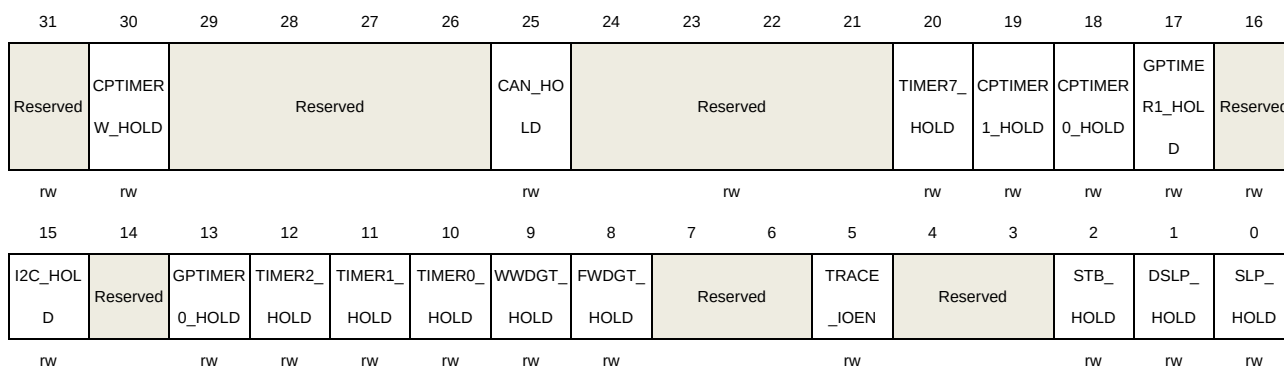
Bits	Fields	Descriptions
31:0	ID_CODE[31:0]	DBG ID code register
These bits can only be read by software. These bits are unchanged constant.		

14.4.2 Control register (DBG_CTL)

Address offset: 0x04

Reset value: 0x0000 0000; power reset only

This register has to be accessed by word(32-bit)



Bits	Fields	Descriptions
31	Reserved	Must be kept at reset value.
30	CPTIMERW_HOLD	CPTIMERW hold bit This bit is set and reset by software. 0: no effect 1: hold the CPTIMERW counter for debugging when the core is halted

29:26	Reserved	Must be kept at reset value.
25	CAN_HOLD	CAN hold bit This bit is set and reset by software. 0: no effect 1: the receive register of CAN stops receiving data when the core is halted
24:21	Reserved	Must be kept at reset value.
20	TIMER7_HOLD	TIMER 7 hold bit This bit is set and reset by software. 0: no effect 1: hold the TIMER 7 counter for debugging when the core is halted
19	CPTIMER1_HOLD	CPTIMER1 hold bit This bit is set and reset by software. 0: no effect 1: hold the CPTIMER1 counter for debugging when the core is halted
18	CPTIMER0_HOLD	CPTIMER0 hold bit This bit is set and reset by software. 0: no effect 1: hold the CPTIMER0 counter for debugging when the core is halted
17	GPTIMER1_HOLD	GPTIMER1 hold bit This bit is set and reset by software. 0: no effect 1: hold the GPTIMER1 counter for debugging when the core is halted
16	Reserved	Must be kept at reset value.
15	I2C_HOLD	I2C hold bit This bit is set and reset by software. 0: no effect 1: hold the I2C status to avoid SMBUS timeout for debugging when the core is halted
14	Reserved	Must be kept at reset value.
13	GPTIMER0_HOLD	GPTIMER 0 hold bit This bit is set and reset by software. 0: no effect 1: hold the GPTIMER 0 counter for debugging when the core is halted
12	TIMER2_HOLD	TIMER 2 hold bit This bit is set and reset by software. 0: no effect 1: hold the TIMER 2 counter for debugging when the core is halted
11	TIMER1_HOLD	TIMER 1 hold bit

		This bit is set and reset by software. 0: no effect 1: hold the TIMER 1 counter for debugging when the core is halted
10	TIMER0_HOLD	TIMER 0 hold bit This bit is set and reset by software. 0: no effect 1: hold the TIMER 0 counter for debugging when the core is halted
9	WWDGT_HOLD	WWDGT hold bit This bit is set and reset by software. 0: no effect 1: hold the WWDGT counter clock for debugging when the core is halted
8	FWDGT_HOLD	FWDGT hold bit This bit is set and reset by software. 0: no effect 1: hold the FWDGT counter clock for debugging when the core is halted
7:6	Reserved	Must be kept at reset value
5	TRACE_IOEN	Trace pin allocation enable This bit is set and reset by software. 0: Trace pin allocation disable 1: Trace pin allocation enable
4:3	Reserved	Must be kept at reset value
2	STB_HOLD	Standby mode hold bit This bit is set and reset by software. 0: no effect 1: In the standby mode, the clock of AHB bus and system clock are provided by CK_IRC32M, users need to select IRC32M as system clock source, a system reset generated when exiting standby mode.
1	DSLP_HOLD	Deep-sleep mode hold bit This bit is set and reset by software. 0: no effect 1: In the Deep-sleep mode, the clock of AHB bus and system clock are provided by CK_IRC32M.
0	SLP_HOLD	Sleep mode hold bit This bit is set and reset by software. 0: no effect 1: In the sleep mode, the clock of AHB is on.

15 Analog-to-digital converter (ADC)

15.1 Overview

Two 12-bit successive approximation analog-to-digital converter module (ADC0/2) are integrated on the MCU chip and work independently.

ADC0 can sample 7 external channels, and ADC2 can sample 10 external channels and 2 internal channels (V_{REFINT} , V_{SENSE}). These selected sampling channels in each ADC module all support a variety of scan modes: Group_pri1 scan once mode, Group_pri1 scan continuous mode, or Groups scan mode. After each channel sampling and conversion, the conversion results can be stored in the corresponding data registers according to the least significant bit (LSB) alignment or the most significant bit (MSB) alignment.

In Group_pri1 mode scan once mode, the selected external channels are converted in ascending channel order with the default channel priority. In Group_pri1 scan continuous mode, the selected external channels are converted in ascending channel order continuously with the default channel priority. In Groups mode scan mode, the selected external channels can be arbitrarily divided into two groups (Group_pri1 & Group_pri2), three groups (Group_pri1 & Group_pri2 & Group_pri3 or Group_pri1 & Group_pri2 & Group_pri4) or four groups (Group_pri1 & Group_pri2 & Group_pri3 & Group_pri4). The channels in each group are converted once in ascending channel order with the default channel priority when the trigger for the corresponding group is acknowledged.

The channel priority is configureable, and the group priority is fixed (Group_pri1 > Group_pri2 > Group_pri3 > Group_pri4). During the channel conversion of low-priority group, the trigger from the high-priority group will be acknowledged, and the ADC aborts the current conversion and starts the channel conversion of high-priority group. After the high-priority group is done, the low-priority group conversion can be resumed from the last aborted channel or the scan start channel. Alternatively, the ADC has the option to skip the conversion of the low-priority group.

In bifurcate trigger mode, only one channel can be selected in each single scan group. The conversion results of the first trigger signal and the second trigger signal will be stored in different data registers.

When the self-diagnosis function is enabled, it is executed once at the beginning of each scan, converting one of the three internally generated diagnostic voltages.

An on-chip hardware oversample scheme improves performances and reduces the computational burden of MCU.

15.2 Characteristics

- High performance.
 - ADC programmable sampling resolution: 12-bit, 10-bit, 8-bit or 6-bit.
 - Programmable A/D sampling time.
 - Programmable Sample and Hold time for sample-and-hold circuit.
 - Programmable disconnection-detection time (precharge and discharge).
 - Data storage mode: the most significant bit (MSB) alignment and the least significant bit (LSB) alignment.
 - DMA support.
- Analog function.
 - 7 external analog inputs for ADC0.
 - 10 external analog inputs channel, an internal reference voltage channel and an internal temperature sensor channel for ADC2.
 - Channel-dedicated sample-and-hold function (three channels for ADC0 only).
 - Analog input disconnection detection assist function (discharge function/precharge function)
- Start-of-conversion can be initiated.
 - Software trigger.
 - Synchronous trigger (from timer).
 - Asynchronous trigger. A/D conversion can be triggered by the external trigger ADC0_TRG pin (for ADC0), ADC2_TRG pin (for ADC2) (independently for two units).
- Operating modes
 - Group_pri1 scan once mode: A/D conversion is performed only once on the selected channel in Group_pri1.
 - Group_pri1 scan continuous mode: A/D conversion is performed repeatedly on the selected channel in Group_pri1.
 - Groups scan mode (group priority control can be enabled or disabled): two groups (Group_pri1 & Group_pri2), three groups (Group_pri1 & Group_pri2 & Group_pri3 or Group_pri1 & Group_pri2 & Group_pri4) or four groups (Group_pri1 & Group_pri2 & Group_pri3 & Group_pri4). A/D conversion is performed only once on the selected channel in every group.
- Conversion result threshold monitor function: analog watchdog A and B.
- Bifurcate trigger mode: The conversion results of the same channel are stored in different data registers.
- Interrupt generation.
 - End of conversion round flag of Group_pri1/Group_pri2/Group_pri3/Group_pri4.
 - The watchdog event (watchdog A channel x compare status or watchdog B compare flag)
 - DMA overflow event of Group_pri1/Group_pri2/Group_pri3/Group_pri4.
- Oversampler: selectable A/D-converted value addition mode or average mode.
 - 32-bit data register.

- Addition count select from 1,2,3,4,16,32,64-time.
- Channel dedicated data register.
- Channel input range: $V_{REFN} \leq V_{IN} \leq V_{REFP}$.

15.3 Pins and internal signals

[Table 15-1. ADC internal signals](#) and [Table 15-2. ADC pins definition](#) give the ADC internal signals and pins description..

Table 15-1. ADC internal signals

Internal signal name	Signal type	Description
V_{REFINT}	Input	Internal voltage reference output voltage (only for ADC2)
V_{SENSE}	Input	Internal temperature sensor output voltage (only for ADC2)

Table 15-2. ADC pins definition

Name	Signal type	ssDescription
V_{DDA}	Input	Analog power supply, analog power supply equals to V_{DD}
V_{SSA}	Input	Analog power supply ground, Ground for analog power supply equals to V_{SS}
V_{REFP}	Input	Analog reference positive, the positive reference voltage for the ADC
V_{REFN}	Input	Analog reference negative, the negative reference voltage for the ADC
$ADCx\_INy$	Input	7 external analog inputs for ADC0 10 external analog inputs for ADC2

Note: In some packages, V_{REFP} and V_{DDA} are connected together, and V_{REFN} and V_{SSA} are connected together.

15.4 Block diagram

Figure 15-1. ADC0 module block diagram

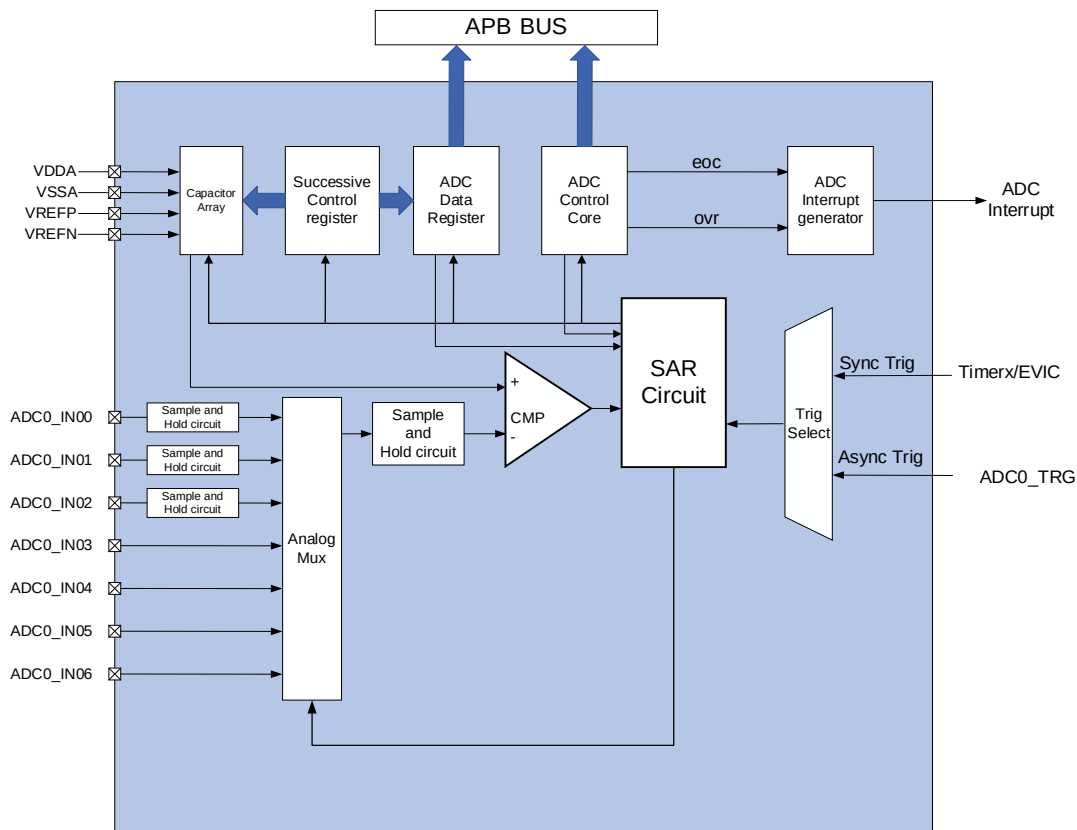
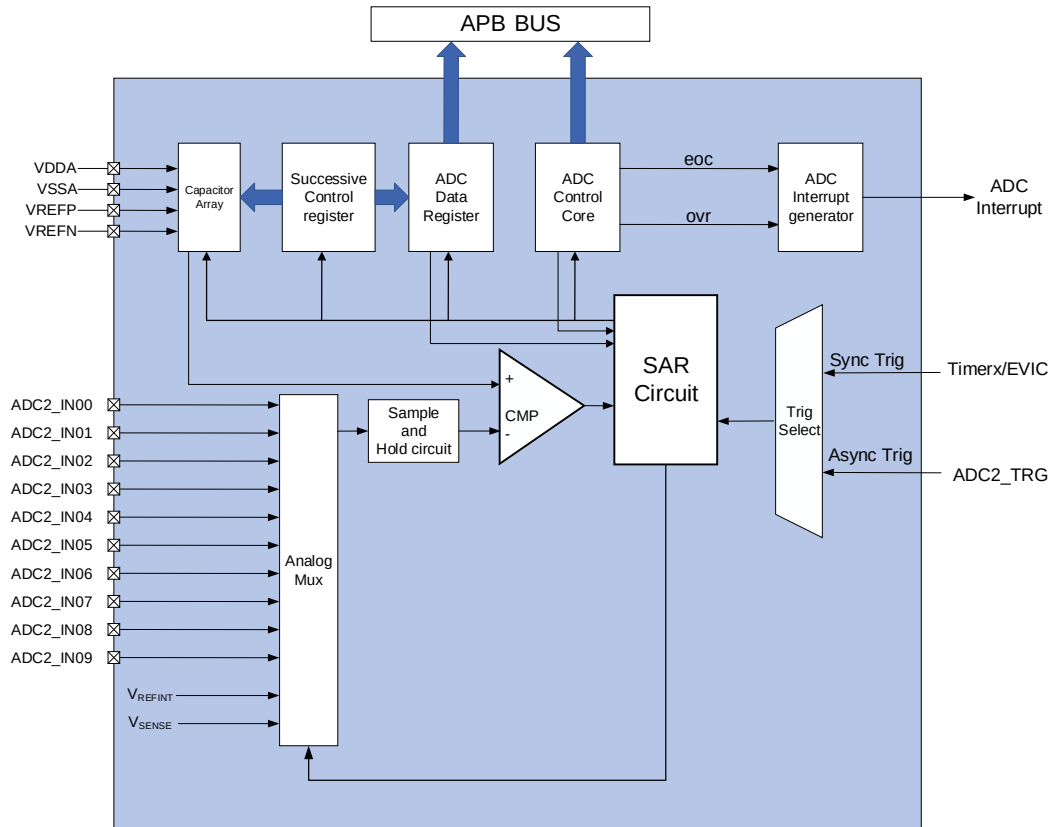


Figure 15-2. ADC2 module block diagram



15.5 Function overview

15.5.1 ADC clock

The CK_ADC clock is provided by the clock controller and has four clock sources (defined by ADCSEL[1:0] bits in the RCU_CFG2 register):

- CK_APB2 (synchronous with the AHB clock)
- CK_PLL (asynchronous with the AHB clock)
- CK_HXTAL (asynchronous with the AHB clock)
- CK_IRC32M (asynchronous with the AHB clock)

ADC clock can be divided and configured by ADCPSC[4:0] in the RCU_CFG2 register. The maximum frequency of CK_ADC is 36MHz.

Note: The maximum operating frequency of the ADC varies with different voltage levels. Please refer to the datasheet for details.

15.5.2 ADC enable

The ADON bit on the ADC_CTL1 register is the enable switch of the ADC module. The ADC module will keep in reset state if this bit is 0. For power saving, when this bit is reset, the analog submodule will be put into power-off mode. After ADC is enabled, you need delay t_{su}

time for sampling, the value of t_{SU} please refer to the device datasheet.

15.5.3 Scanning operation modes

In scan operation mode, the ADC sequentially converts the selected channels based on the specified channel priority, which is defined in the ADC_CHPR $_x$ ($x=0,1$) register.

The ADC supports three scan operation modes, configured by the GSCAN[1:0] bits in the ADC_CTL0 register:

- Group_pri1 scan once mode
- Groups scan mode
- Group_pri1 scan continuous mode

In Group_pri1 scan once mode, the ADC performs a one-time conversion of all selected channels. In Group_pri1 scan continuous mode, the ADC continuously converts the selected channels, and the conversion of all groups can be ended by writing 1 to the SWEND bit in the ADC_CTL0 register. In Groups scan mode, each group converts the selected channels once at their respective synchronous triggers.

When the self-diagnosis is enabled, a self-diagnostic voltage is converted at the beginning of each scan.

In Group_pri1 scan once modes, bifurcate trigger mode can be enabled by setting GP1BIMEN to 1 in ADC_BITRGCTL register. In groups scan mode, the bifurcate trigger function is available for all groups and can be enabled by setting GPxBIMEN to 1 in ADC_BITRGCTL register for Group_prix. With bifurcate trigger mode being enabled, A/D conversion data of a channel selected by the GPxBICHSEL[4:0] bits ADC_BITRGCTL register is duplicated if the conversion is started by the synchronous trigger selected by the GPxTRGSRC[5:0] bits.

The PROCF pin is used to output the status of the PROCF bit (read-only) in the ADC_CTL0.

Extended bifurcate trigger mode indicates a state when two synchronous trigger sources enabled by the GPxTRGSRC[5:0] bits in the ADC_GPCGF0/1 registers in bifurcate mode (GPxTRGSRC[5:0] configured as 6'b000110, 6'b001110, 6'b100101, 6'b101101 or 6'b110011). In extended bifurcate trigger mode, in addition to normal operations in bifurcate trigger mode, A/D conversion data is stored in ADC_GPxBIDATA1 or ADC_GPxBIDATA2 on the trigger type. Bind the conversion result triggered by TRGA to the ADC_GPxBIDATA1 register, and bind the conversion result triggered by TRGB to the ADC_GPxBIDATA2 data register. EOCxF is set only after the conversion triggered by TRGB is complete.

When any of ADC0_IN0/ADC0_IN1/ADC0_IN2 channels is set as a channel-dedicated sample-and-hold circuit by the SH00EN/ SH01EN/ SH02EN in ADC_SHCTL register, the target analog input is sampled and held before the first A/D conversion of each scan.

Note: if a new trigger is input during A/D conversion caused by another trigger, the new trigger is ignored.

15.5.4 Group_pri1 scan once mode

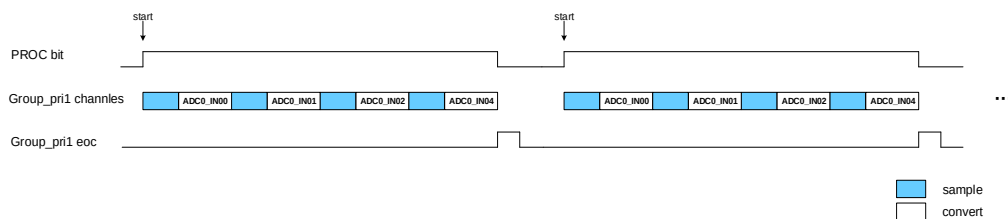
In the following examples, the channel priority registers ADC_CHPR1x are set to default value.

Basic operation (without channel-dedicated sample-and-hold circuits)

In basic operation of Group_pri1 scan once mode, A/D conversion is performed once on the analog input of the specified channels. For example, ADC0_IN0, ADC0_IN1, ADC0_IN2, ADC0_IN4 is selected for ADC0, as shown in [Figure 15-3. Basic scan once operation of ADC0 \(without channel-dedicated sample-and-hold circuits\)](#).

- (1) When the GP1SWST bit in the ADC_EOCCTL register is set to 1 by software, or synchronous or asynchronous trigger input, A/D conversion is performed for these channels selected by the ADC_CHSELx registers, starting from the channel with the smallest number (ADC0_IN0 in this example).
- (2) Each time A/D conversion of a single channel is completed, the A/D conversion result is stored into the corresponding A/D data register.
- (3) When A/D conversion of all the selected channels is completed, EOC1F bit in ADC_EOCCTL register is set to 1. When every the round num ($EOC1RCNT[2:0] + 1$) of end of Group_pri1 conversion comes, this EOC1RF flag will be set to 1 and an interrupt request is generated if the EOC1RIE bit in ADC_EOCCTL is 1.
- (4) The PROC bit in ADC_CTL0 remains 1 (A/D conversion start) during A/D conversion, and is automatically cleared to 0 when A/D conversion of all the selected channels is completed. Then the A/D converter enters a wait state.

Figure 15-3. Basic scan once operation of ADC0 (without channel-dedicated sample-and-hold circuits)



Basic operation (with channel-dedicated sample-and-hold circuits, constant sampling disabled)

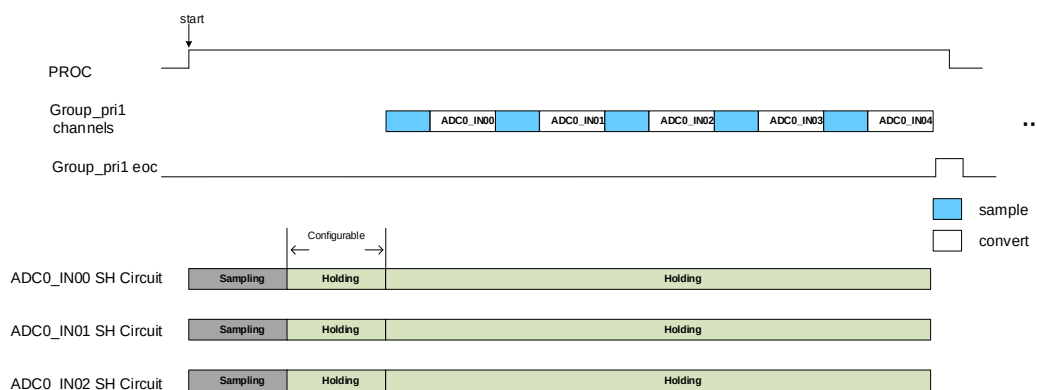
When a channel-dedicated sample-and-hold circuit is used, sample-and-hold operations are performed first, and this is followed by A/D conversion once of the analog inputs on all selected channels. For example, ADC0_IN0, ADC0_IN1, ADC0_IN2, ADC0_IN3, ADC0_IN4 is selected for ADC0, as shown in [Figure 15-4. Basic scan once operation of ADC0 \(with channel-dedicated sample-and-hold circuits, constant sampling disabled\)](#).

The SH00EN/ SH01EN / SH02EN bits in ADC_SHCTL register are used to select the channels for which the channel-dedicated sample-and-hold circuits are to be used.

Clear SHMD bit in ADC_SHCTL register to 0 to disable constant sampling.

- (1) Analog input sampling of all channels for which the channel-dedicated sample-and-hold circuits are to be used is started when the GP1SWST bit in the ADC_EOCCTL register is set to 1 by software, or synchronous or asynchronous trigger input.
- (2) After sample-and-hold operation, A/D conversion is performed for the selected channels, starting from the channel with the smallest number (ADC0_IN0 in this example).
- (3) Each time A/D conversion of a single channel is completed, the result of A/D conversion is stored in the corresponding A/D data register (ADC_CHxDATA).
- (4) When A/D conversion of all the selected channels is completed, EOC1F bit in ADC_EOCCTL register is set to 1. When every the round num ($EOC1RCNT[2:0] + 1$) of end of Group_pri1 conversion comes, this EOC1RF flag will be set to 1 and an interrupt request is generated if the EOC1RIE bit in ADC_EOCCTL is 1.
- (5) The PROC bit in ADC_CTL0 remains 1 (A/D conversion start) during A/D conversion, and is automatically cleared to 0 when A/D conversion of all the selected channels is completed. Then the A/D converter enters a wait state.

Figure 15-4. Basic scan once operation of ADC0 (with channel-dedicated sample-and-hold circuits, constant sampling disabled)



Basic operation (with channel-dedicated sample-and-hold circuits, constant sampling enabled)

When a channel-dedicated sample-and-hold circuit is used, sample-and-hold operations are performed first, and this is followed by A/D conversion once of the analog inputs on all selected channels. For example, ADC0_IN0, ADC0_IN1, ADC0_IN2, ADC0_IN3, ADC0_IN4 is selected for ADC0, as shown in [Figure 15-5. Basic scan once operation of ADC0 \(with channel-dedicated sample-and-hold circuits, constant sampling enabled\).](#)

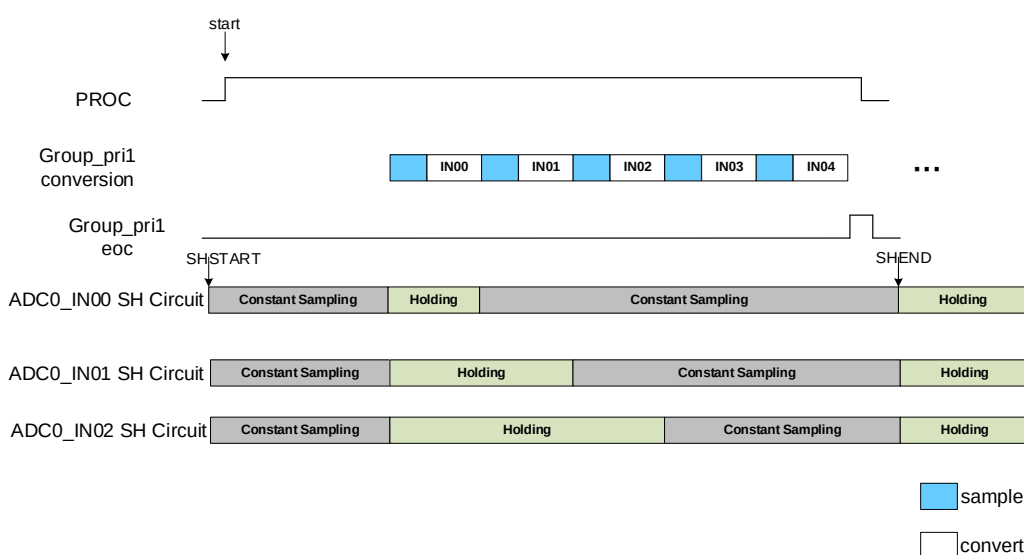
The SH00EN/ SH01EN / SH02EN bits are used to select the channels for which the channel-

dedicated sample-and-hold circuits are to be used.

Set SHMD bit in ADC_SHCTL register to 1 to enable constant sampling.

- (1) Set SHSTART bit in ADC_SHCTL register to 1 to start the selected channel-dedicated Sample-and-Holding circuit working.
- (2) Analog input sampling of all channels for which the channel-dedicated sample-and-hold circuits are to be used is started when the GP1SWST bit in the ADC_EOCCTL register is set to 1 by software, or synchronous or asynchronous trigger input.
- (3) After sample-and-hold operation, A/D conversion is performed for the selected channels, starting from the channel with the smallest number (ADC0_IN0 in this example).
- (4) Each time A/D conversion of a single channel is completed, the result of A/D conversion is stored in the corresponding A/D data register (ADC_CHxDATA).
- (5) When A/D conversion of all the selected channels is completed, EOC1F bit in ADC_EOCCTL register is set to 1. When every the round num (EOC1RCNT[2:0] + 1) of end of Group_pri1 conversion comes, this EOC1RF flag will be set to 1 and an interrupt request is generated if the EOC1RIE bit in ADC_EOCCTL is 1.
- (6) The PROC bit in ADC_CTL0 remains 1 (A/D conversion start) during A/D conversion, and is automatically cleared to 0 when A/D conversion of all the selected channels is completed. Then the A/D converter enters a wait state.
- (7) Set SHEND bit in ADC_SHCTL register to terminate the selected channel-dedicated Sample-and-Holding circuit working.

Figure 15-5. Basic scan once operation of ADC0 (with channel-dedicated sample-and-hold circuits, constant sampling enabled)



Channel selection and self-diagnosis (without channel-dedicated sample-and-hold circuits)

Set SDSEN bit in ADC_CTL0 register to 1 to enable self-diagnosis.

When channels and self-diagnosis are selected, A/D conversion is performed once for the reference voltage supplied to the 12-bit A/D converter, as shown in [Figure 15-6. Channel selection and self-diagnosis in scan once mode of ADC2 \(without channel-dedicated sample-and-hold circuits\)](#). After that, A/D conversion is performed only once on the analog input of the selected channels. In this example, the selected channels are ADC2_IN1, ADC2_IN5, ADC2_IN6, ADC2_IN7, ADC2_IN8.

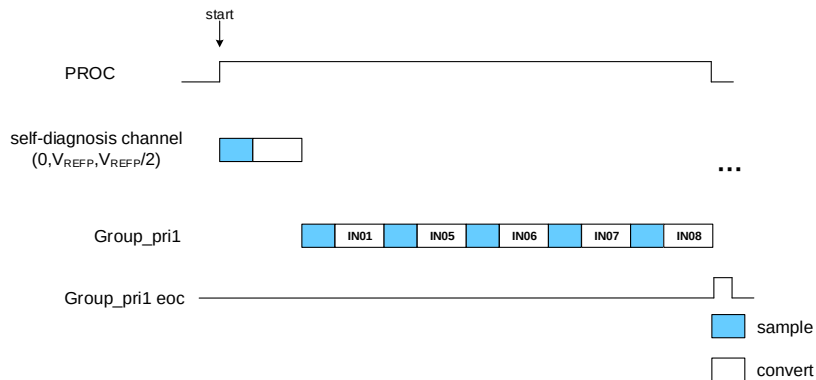
When SDSFIX bit in ADC_CTL0 register is 1, a fix voltage (configure by SDVOG[1:0] in ADC_CTL0 register) is converted for self-diagnosis.

When SDSFIX bit in ADC_CTL0 register is 0, self-diagnosis takes turns converting 0, $V_{REFP}/2$, V_{REFP} for each scan.

- (1) A/D conversion for self-diagnosis is started when the GP1SWST bit in the ADC_EOCCTL register is set to 1 by software, or synchronous or asynchronous trigger input.
- (2) When A/D conversion for self-diagnosis is completed, A/D conversion result is stored into the A/D self-diagnosis data register (ADC_SDDATA), and A/D conversion is performed for the selected channels, starting from the channel with the smallest number (ADC2_IN1 in this example).
- (3) Each time A/D conversion of a single channel is completed, the A/D conversion result is stored into the corresponding A/D data register.
- (4) When A/D conversion of all the selected channels is completed, EOC1F bit in ADC_EOCCTL register is set to 1. When every the round num ($EOC1RCNT[2:0] + 1$) of end of Group_pri1 conversion comes, this EOC1RF flag will be set to 1 and an interrupt request is generated if the EOC1RIE bit in ADC_EOCCTL is 1.
- (5) The PROC bit in ADC_CTL0 remains 1 (A/D conversion start) during A/D conversion, and is automatically cleared to 0 when A/D conversion of all the selected channels is completed. Then the A/D converter enters a wait state.

Figure 15-6. Channel selection and self-diagnosis in scan once mode of ADC2

(without channel-dedicated sample-and-hold circuits)



Channel selection and self-diagnosis (with channel-dedicated sample-and-hold circuits, constant sampling disabled)

Set SDSEN bit in ADC_CTL0 register to 1 to enable self-diagnosis.

When channels and self-diagnosis are selected and a channel-dedicated sample-and-hold circuit is used, sample-and-hold operations are performed first, and A/D conversion is performed once for the reference voltage supplied to the 12-bit A/D converter, as shown in [Figure 15-7. Channel selection and self-diagnosis in scan once mode of ADC0 \(with channel-dedicated sample-and-hold circuits, constant sampling disabled\)](#). After that, A/D conversion is performed only once on the analog input of the selected channels. In this example, the selected channels are ADC0_IN0, ADC0_IN1, ADC0_IN2, ADC0_IN3, ADC0_IN4.

When SDSFIX bit in ADC_CTL0 register is 1, a fix voltage (configure by SDVOG[1:0] in ADC_CTL0 register) is converted for self-diagnosis.

When SDSFIX bit in ADC_CTL0 register is 0, self-diagnosis takes turns converting 0, $V_{REFP}/2$, V_{REFP} for each scan.

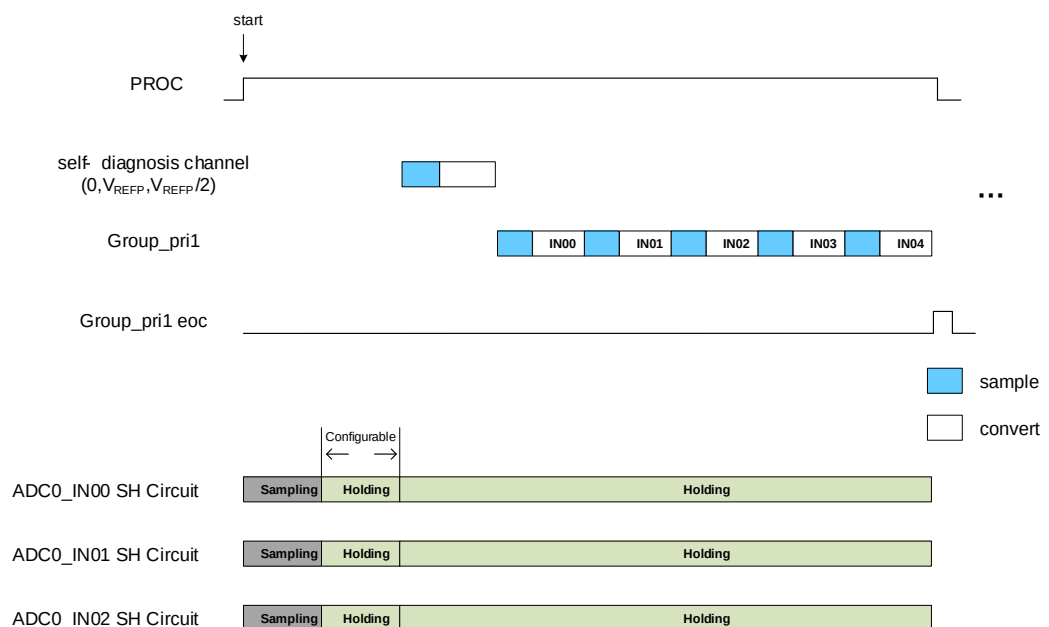
The SH00EN/ SH01EN / SH02EN bits in ADC_SHCTL register are used to select the channels for which the channel-dedicated sample-and-hold circuits are to be used.

Clear SHMD bit in ADC_SHCTL register to 0 to disable constant sampling.

- (1) Analog input sampling of all channels for which the channel-dedicated sample-and-hold circuits are to be used is started when the GP1SWST bit in the ADC_EOCCTL register is set to 1 by software, or synchronous or asynchronous trigger input.
- (2) A/D conversion for self-diagnosis is started after completion of sampling and holding.
- (3) When A/D conversion for self-diagnosis is completed, A/D conversion result is stored into the A/D self-diagnosis data register (ADC_SDDATA), and A/D conversion is performed for the selected channels, starting from the channel with the smallest number (ADC0_IN0 in this example).

- (4) Each time A/D conversion of a single channel is completed, the A/D conversion result is stored into the corresponding A/D data register.
- (5) When A/D conversion of all the selected channels is completed, EOC1F bit in ADC_EOCCTL register is set to 1. When every the round num (EOC1RCNT[2:0] + 1) of end of Group_pri1 conversion comes, this EOC1RF flag will be set to 1 and an interrupt request is generated if the EOC1RIE bit in ADC_EOCCTL is 1.
- (6) The PROC bit in ADC_CTL0 remains 1 (A/D conversion start) during A/D conversion, and is automatically cleared to 0 when A/D conversion of all the selected channels is completed. Then the A/D converter enters a wait state.

Figure 15-7. Channel selection and self-diagnosis in scan once mode of ADC0 (with channel-dedicated sample-and-hold circuits, constant sampling disabled)



Channel selection and self-diagnosis (with channel-dedicated sample-and-hold circuits, constant sampling enabled)

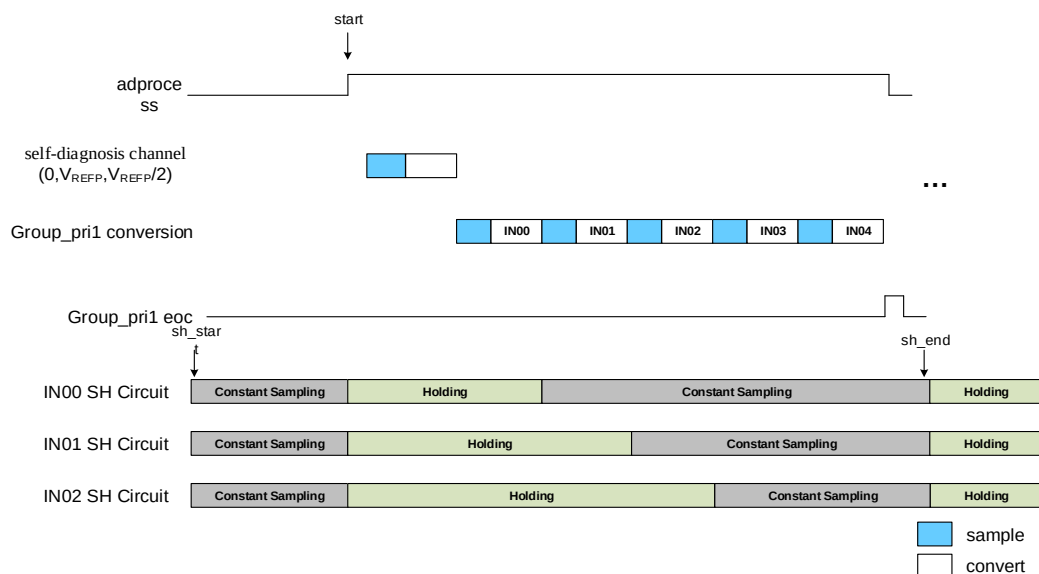
When channels and self-diagnosis are selected and a channel-dedicated sample-and-hold circuit is used, sample-and-hold operations are performed first, and A/D conversion is performed once for the reference voltage supplied to the 12-bit A/D converter, as shown in [Figure 15-8. Channel selection and self-diagnosis in scan once mode of ADC0 \(with channel-dedicated sample-and-hold circuits, constant sampling enabled\)](#). After that, A/D conversion is performed only once on the analog input of the selected channels. In this example, the selected channels are ADC0_IN0, ADC0_IN1, ADC0_IN2, ADC0_IN3, ADC0_IN4.

The SH00EN/ SH01EN / SH02EN bits are used to select the channels for which the channel-dedicated sample-and-hold circuits are to be used.

Set SHMD bit in ADC_SHCTL register to 1 to enable constant sampling.

- (1) Set SHSTART bit in ADC_SHCTL register to 1 to start the selected channel-dedicated Sample-and-Holding circuit working.
- (2) Analog input sampling of all channels for which the channel-dedicated sample-and-hold circuits are to be used is started when the GP1SWST bit in the ADC_EOCCTL register is set to 1 by software, or synchronous or asynchronous trigger input.
- (3) When A/D conversion for self-diagnosis is completed, A/D conversion result is stored into the A/D self-diagnosis data register (ADC_SDDATA), and A/D conversion is performed for the selected channels, starting from the channel with the smallest number (ADC0_IN0 in this example).
- (4) Each time A/D conversion of a single channel is completed, the A/D conversion result is stored into the corresponding A/D data register.
- (5) When A/D conversion of all the selected channels is completed, EOC1F bit in ADC_EOCCTL register is set to 1. When every the round num (EOC1RCNT[2:0] + 1) of end of Group_pri1 conversion comes, this EOC1RF flag will be set to 1 and an interrupt request is generated if the EOC1RIE bit in ADC_EOCCTL is 1.
- (6) The PROC bit in ADC_CTL0 remains 1 (A/D conversion start) during A/D conversion, and is automatically cleared to 0 when A/D conversion of all the selected channels is completed. Then the A/D converter enters a wait state.
- (7) Set SHEND bit in ADC_SHCTL register to terminate the selected channel-dedicated Sample-and-Holding circuit working.

Figure 15-8. Channel selection and self-diagnosis in scan once mode of ADC0 (with channel-dedicated sample-and-hold circuits, constant sampling enabled)



A/D conversion in bifurcate trigger mode

The bifurcate trigger mode of Group_pri x is enabled by setting the GPxBIMEN bit in ADC_BITRGCTL register to 1.

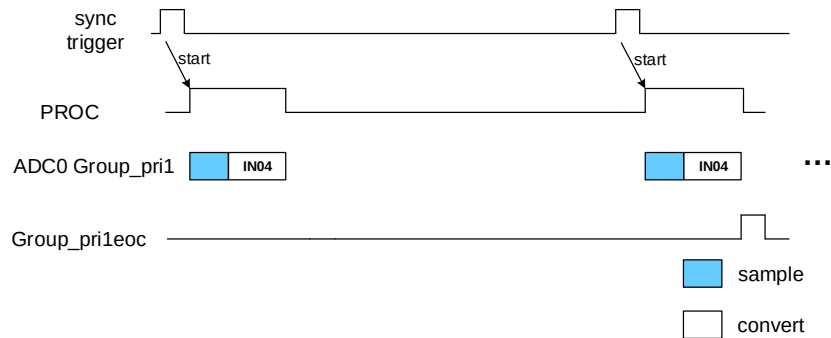
Duplication of A/D conversion data of Group_pri x is enabled by setting the channel numbers to be duplicated to the GPxBICHSEL[4:0] bits in ADC_BITRGCTL register and setting the GPxBIMEN bit in ADC_BITRGCTL register to 1. When the GPxBIMEN bit in ADC_BITRGCTL register is set to 1, channel selection using the ADC_CHSEL0/1 registers are invalid.

In Group_pri x scan once mode with bifurcate trigger mode, single scan operation started by the asynchronous/synchronous trigger bits or software start is performed twice, as shown in [Figure 15-9. A/D conversion in Group_pri1 bifurcate trigger mode of ADC0](#). In this example, the selected channel is ADC0_IN4.

If the synchronous trigger is enable and the GPxTRGSRC[5:0] is configured as 6'b000110, 6'b001110, 6'b100101, 6'b101101 or 6'b110011 then the extended bifurcate mode is automatically enabled.

If bifurcate trigger mode of Group_pri x is enabled, self-diagnosis should be disabled for Group_pri x, and the internal reference voltage and temperature sensor channels should be deselected in Group_pri x.

- (1) When synchronous trigger input, A/D conversion is started on the single channel selected by the GP1BICHSEL[4:0] bits in ADC_BITRGCTL register (ADC0_IN4 in this example), and the PROC bit in ADC_CTL0 register is set to 1 (A/D conversion start).
- (2) When A/D conversion is completed, the A/D conversion result is stored into the corresponding channel data register. If operating in extended bifurcate mode, the result will be stored in ADC_GPxBIDATA1 simultaneously. The PROC bit in ADC_CTL0 register is automatically cleared to 0 and the A/D converter enters a wait state. A scan end interrupt request is not generated irrespective of the EOC1RIE bit in ADC_EOCCTL register setting.
- (3) When the second trigger input, A/D conversion is started on the ADC0_IN4 channel, and the PROC bit in ADC_CTL0 register is set to 1 (A/D conversion start).
- (4) When A/D conversion is completed, the A/D conversion result is stored into Group_pri x the A/D data duplication register (ADC_GP1BIDATA), which is exclusively used in bifurcate trigger mode. If operating in extended bifurcate mode, the result will be stored in ADC_GPxBIDATA2 simultaneously.
- (5) If the EOC1RIE bit in ADC_EOCCTL register is 1 (interrupt generation upon scanning completion enabled), a scan end interrupt request is generated (EOC1RCNT[2:0]=3'b000).
- (6) The PROC bit in ADC_CTL0 register remains 1 during A/D conversion, and is automatically cleared to 0 when A/D conversion is completed. Then the ADC enters a wait state.

Figure 15-9. A/D conversion in Group_pri1 bifurcate trigger mode of ADC0


15.5.5 Group_pri1 scan continuous mode

In the following examples, the channel priority registers ADC_CHPR1x are set to default value.

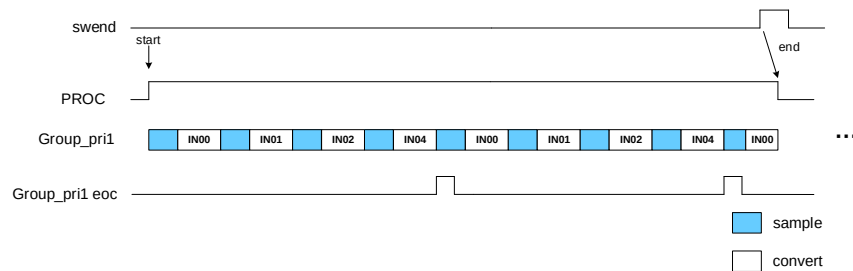
Basic operation (without channel-dedicated sample-and-hold circuits)

In basic operation of Group_pri1 scan continuous mode, A/D conversion is performed continuously on the analog input of the specified channels. For example, ADC0_IN0, ADC0_IN1, ADC0_IN2, ADC0_IN4 is selected for ADC0, as shown in [Figure 15-10. Basic continuous scan operation of ADC0 \(without channel-dedicated sample-and-hold circuits\)](#).

In continuous scan mode, the internal reference voltage and temperature sensor channels should be deselected.

- (1) When the GP1SWST bit in the ADC_EOCCTL register is set to 1 by software, or synchronous or asynchronous trigger input, A/D conversion is performed for these channels selected by the ADC_CHSELx registers, starting from the channel with the smallest number (ADC0_IN0 in this example).
- (2) Each time A/D conversion of a single channel is completed, the A/D conversion result is stored into the corresponding A/D data register.
- (3) When A/D conversion of all the selected channels is completed, EOC1F bit in ADC_EOCCTL register is set to 1. Then the 12-bit A/D converter sequentially starts A/D conversion for these selected channels starting from the channel with the smallest number (ADC0_IN0 in this example). When every the round num (EOC1RCNT[2:0] + 1) of end of Group_pri1 conversion comes, this EOC1RF flag will be set to 1 and an interrupt request is generated if the EOC1RIE bit in ADC_EOCCTL is 1.
- (4) The PORC bit is not automatically cleared to 0 and steps 2 and 3 are repeated as long as the bit remains 1 (A/D conversion start).
- (5) When the SWEND bit in ADC_CTL0 register is set to 1, A/D conversion stops and the A/D converter enters a wait state.

Figure 15-10. Basic continuous scan operation of ADC0 (without channel-dedicated sample-and-hold circuits)



Basic operation (with channel-dedicated sample-and-hold circuits, constant sampling disabled)

When a channel-dedicated sample-and-hold circuit is used, sample-and-hold operations are performed first, and this is followed by A/D conversion once of the analog inputs on all selected channels. Sample-and-Hold -> analog inputs->..., repeats like this. For example, ADC0_IN0, ADC0_IN1, ADC0_IN2, ADC0_IN4 is selected for ADC0, as shown in [Figure 15-11. Basic continuous scan operation of ADC0 \(with channel-dedicated sample-and-hold circuits, constant sampling disabled\)](#).

The SH00EN/ SH01EN / SH02EN bits in ADC_SHCTL register are used to select the channels for which the channel-dedicated sample-and-hold circuits are to be used.

Clear SHMD bit in ADC_SHCTL register to 0 to disable constant sampling.

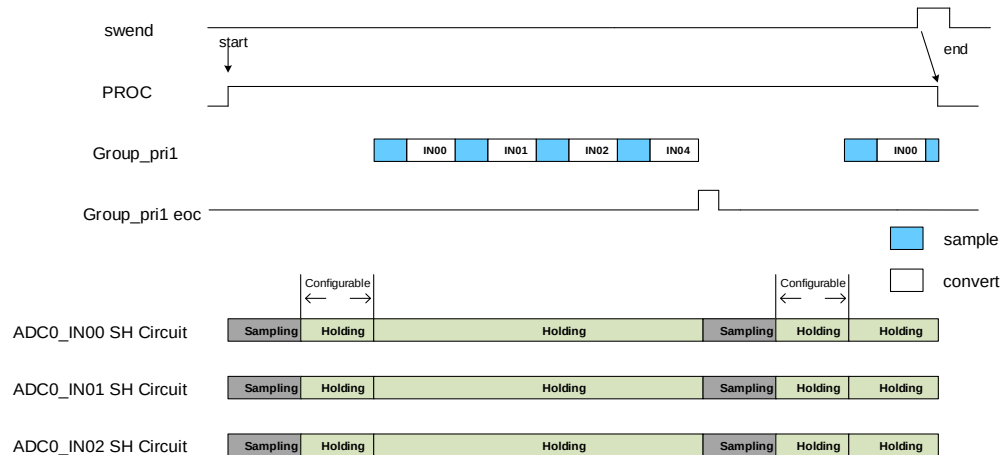
In continuous scan mode, the internal reference voltage and temperature sensor channels should be deselected.

- (1) Analog input sampling of all channels for which the channel-dedicated sample-and-hold circuits are to be used is started when the GP1SWST bit in the ADC_EOCCTL register is set to 1 by software, or synchronous or asynchronous trigger input.
- (2) After sample-and-hold operation, A/D conversion is performed for the selected channels, starting from the channel with the smallest number (ADC0_IN0 in this example).
- (3) Each time A/D conversion of a single channel is completed, the result of A/D conversion is stored in the corresponding A/D data register (ADC_CHxDATA).
- (4) When A/D conversion of all the selected channels is completed, EOC1F bit in ADC_EOCCTL register is set to 1. At the same time, analog input sampling is started for all the channels for which the channel-dedicated sample-and-hold circuits are to be used. When every the round num ($EOC1RCNT[2:0] + 1$) of end of Group_pri1 conversion comes, this EOC1RF flag will be set to 1 and an interrupt request is generated if the EOC1RIE bit in ADC_EOCCTL is 1.
- (5) The PROC bit is not automatically cleared and steps 2 to 4 are repeated as long as the

bit remains 1.

- (6) When the SWEND bit in ADC_CTL0 register is set to 1, A/D conversion stops and the A/D converter enters a wait state.

Figure 15-11. Basic continuous scan operation of ADC0 (with channel-dedicated sample-and-hold circuits, constant sampling disabled)



Basic operation (with channel-dedicated sample-and-hold circuits, constant sampling enabled)

When a channel-dedicated sample-and-hold circuit is used, sample-and-hold operations are performed first, and this is followed by A/D conversion once of the analog inputs on all selected channels. Sample-and-Hold -> analog inputs->..., repeats like this. For example, ADC0_IN0, ADC0_IN2, ADC0_IN3, ADC0_IN4 is selected for ADC0, as shown in [Figure 15-12. Basic continuous scan operation of ADC0 \(with channel-dedicated sample-and-hold circuits, constant sampling enabled\).](#)

The SH00EN/ SH01EN / SH02EN bits in ADC_SHCTL register are used to select the channels for which the channel-dedicated sample-and-hold circuits are to be used.

Clear SHMD bit in ADC_SHCTL register to 1 to enable constant sampling.

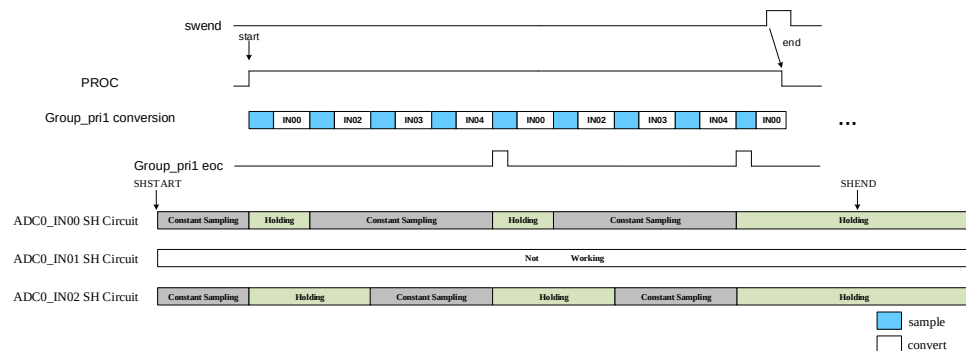
In continuous scan mode, the internal reference voltage and temperature sensor channels should be deselected.

- (1) Set SHSTART bit in ADC_SHCTL register to 1 to start the selected channel-dedicated Sample-and-Holding circuit working.
- (2) Analog input sampling of all channels for which the channel-dedicated sample-and-hold circuits are to be used is started when the GP1SWST bit in the ADC_EOCCTL register is set to 1 by software, or synchronous or asynchronous trigger input.
- (3) A/D conversion is performed for the selected channels, starting from the channel with the smallest number (ADC0_IN0 in this example). Each time A/D conversion of a single channel is completed, the result of A/D conversion is stored in the corresponding A/D

data register (ADC_CHxDATA).

- (4) When A/D conversion of all the selected channels is completed, EOC1F bit in ADC_EOCCTL register is set to 1. At the same time, analog input sampling is started for all the channels for which the channel-dedicated sample-and-hold circuits are to be used. When every the round num (EOC1RCNT[2:0] + 1) of end of Group_pri1 conversion comes, this EOC1RF flag will be set to 1 and an interrupt request is generated if the EOC1RIE bit in ADC_EOCCTL is 1.
- (5) The PROC bit is not automatically cleared and steps 2 to 4 are repeated as long as the bit remains 1.
- (6) When the SWEND bit in ADC_CTL0 register is set to 1, A/D conversion stops and the A/D converter enters a wait state.
- (7) Set SHEND bit in ADC_SHCTL register to terminate the selected channel-dedicated Sample-and-Holding circuit working.

Figure 15-12. Basic continuous scan operation of ADC0 (with channel-dedicated sample-and-hold circuits, constant sampling enabled)



Channel selection and self-diagnosis (without channel-dedicated sample -and- hold circuits)

Set SDSEN bit in ADC_CTL0 register to 1 to enable self-diagnosis.

When channels and self-diagnosis are selected, A/D conversion is performed once for the reference voltage supplied to the 12-bit A/D converter, as shown in [Figure 15-13. Channel selection and self-diagnosis of continuous scan operation of ADC2 \(without channel-dedicated sample-and-hold circuits\)](#). After that, A/D conversion is performed once on the analog input of the selected channels. Self-diagnosis -> analog inputs->..., repeats like this. In this example, the selected channels are ADC2_IN0, ADC2_IN1, ADC2_IN3, ADC2_IN4, ADC2_IN9.

When SDSFIX bit in ADC_CTL0 register is 1, a fix voltage (configure by SDVOG[1:0] in ADC_CTL0 register) is converted for self-diagnosis.

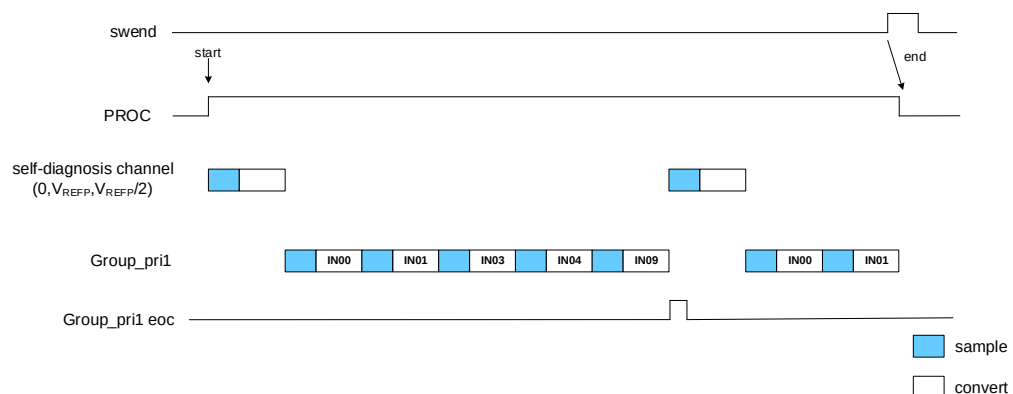
When SDSFIX bit in ADC_CTL0 register is 0, self-diagnosis takes turns converting 0, V_{REFP}

/2, V_{REFP} for each scan.

In continuous scan mode, the internal reference voltage and temperature sensor channels should be deselected.

- (1) When GP1SWST bit in the ADC_EOCCTL register is set to 1 by software, or synchronous or asynchronous trigger input, A/D conversion for self-diagnosis is started first.
- (2) When A/D conversion for self-diagnosis is completed, A/D conversion result is stored into the A/D self-diagnosis data register (ADC_SDDATA), and A/D conversion is performed for the selected channels, starting from the channel with the smallest number (ADC2_IN0 in this example).
- (3) Each time A/D conversion of a single channel is completed, the A/D conversion result is stored into the corresponding A/D data register.
- (4) When A/D conversion of all the selected channels is completed, EOC1F bit in ADC_EOCCTL register is set to 1. At the same time, the 12-bit A/D converter starts A/D conversion for self-diagnosis and then starts A/D conversion on the channels selected by the ADC_CHSEL0/1 registers, starting from the channel with the smallest number (ADC2_IN0 in this example). When every the round num ($EOC1RCNT[2:0] + 1$) of end of Group_pri1 conversion comes, this EOC1RF flag will be set to 1 and an interrupt request is generated if the EOC1RIE bit in ADC_EOCCTL is 1.
- (5) The PROC bit is not automatically cleared and steps 2 to 4 are repeated as long as the bit remains 1.
- (6) When the SWEND bit in ADC_CTL0 register is set to 1, A/D conversion stops and the A/D converter enters a wait state.

Figure 15-13. Channel selection and self-diagnosis of continuous scan operation of ADC2 (without channel-dedicated sample-and-hold circuits)



Channel selection and self-diagnosis (with channel-dedicated sample -and- hold circuits, constant sampling disabled)

Set SDSEN bit in ADC_CTL0 register to 1 to enable self-diagnosis.

When channels and self-diagnosis are selected and a channel-dedicated sample-and-hold circuit is used, sample-and-hold operations are performed first, and A/D conversion is performed once for the reference voltage supplied to the 12-bit A/D converter, as shown in [Figure 15-14. Channel selection and self-diagnosis of continuous scan operation of ADC0 \(with channel-dedicated sample-and-hold circuits, constant sampling disabled\)](#). After that, A/D conversion is performed once on the analog input of the selected channels. Sample-and-hold -> Self-diagnosis -> analog inputs->..., repeats like this. In this example, the selected channels are ADC0_IN0, ADC0_IN1, ADC0_IN2, ADC0_IN3, ADC0_IN4.

When SDSFIX bit in ADC_CTL0 register is 1, a fix voltage (configure by SDVOG[1:0] in ADC_CTL0 register) is converted for self-diagnosis.

When SDSFIX bit in ADC_CTL0 register is 0, self-diagnosis takes turns converting 0, $V_{REFP}/2$, V_{REFP} for each scan.

The SH00EN/ SH01EN / SH02EN bits in ADC_SHCTL register are used to select the channels for which the channel-dedicated sample-and-hold circuits are to be used.

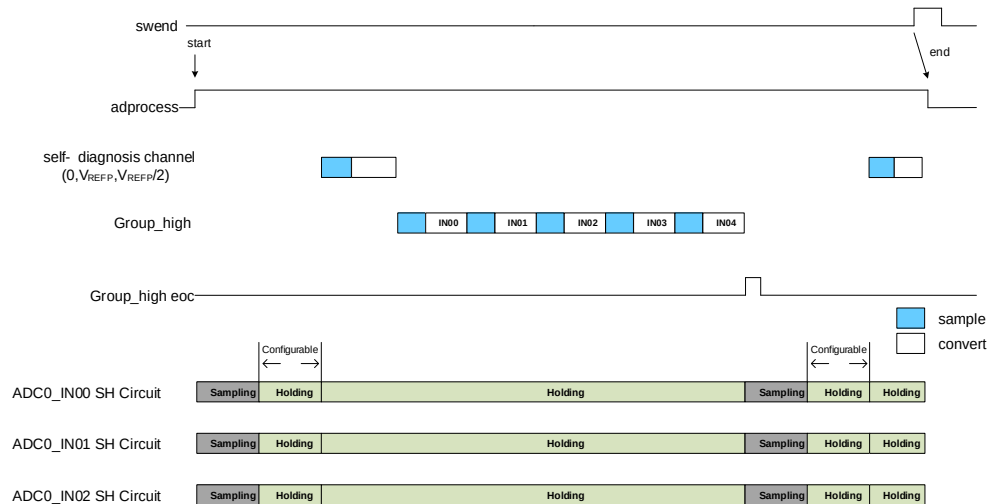
Clear SHMD bit in ADC_SHCTL register to 0 to disable constant sampling.

In continuous scan mode, the internal reference voltage and temperature sensor channels should be deselected.

- (1) Analog input sampling of all channels for which the channel-dedicated sample-and-hold circuits are to be used is started when the GP1SWST bit in the ADC_EOCCTL register is set to 1 by software, or synchronous or asynchronous trigger input.
- (2) A/D conversion for self-diagnosis is started after completion of sampling and holding.
- (3) When A/D conversion for self-diagnosis is completed, A/D conversion result is stored into the A/D self-diagnosis data register (ADC_SDDATA), and A/D conversion is performed for the selected channels, starting from the channel with the smallest number (ADC0_IN0 in this example).
- (4) Each time A/D conversion of a single channel is completed, the A/D conversion result is stored into the corresponding A/D data register.
- (5) When A/D conversion of all the selected channels is completed, EOC1F bit in ADC_EOCCTL register is set to 1. At the same time, analog input sampling is started for all the channels for which the channel-dedicated sample-and-hold circuits are to be used. When every the round num ($EOC1RCNT[2:0] + 1$) of end of Group_pri1 conversion comes, this EOC1RF flag will be set to 1 and an interrupt request is generated if the EOC1RIE bit in ADC_EOCCTL is 1.

- (6) The PROC bit is not automatically cleared and steps 2 to 5 are repeated as long as the bit remains 1.
- (7) When the SWEND bit in ADC_CTL0 register is set to 1, A/D conversion stops and the 12-bit A/D converter enters a wait state.

Figure 15-14. Channel selection and self-diagnosis of continuous scan operation of ADC0 (with channel-dedicated sample-and-hold circuits, constant sampling disabled)



Channel selection and self-diagnosis (with channel-dedicated sample -and- hold circuits, constant sampling enabled)

Set SDSEN bit in ADC_CTL0 register to 1 to enable self-diagnosis.

When channels and self-diagnosis are selected and a channel-dedicated sample-and-hold circuit is used, sample-and-hold operations are performed first, and A/D conversion is performed once for the reference voltage supplied to the 12-bit A/D converter, as shown in [Figure 15-15. Channel selection and self-diagnosis of continuous scan operation of ADC2 \(with channel-dedicated sample-and-hold circuits, constant sampling enabled\)](#). After that, A/D conversion is performed once on the analog input of the selected channels. Sample-and-hold -> Self-diagnosis -> analog inputs->..., repeats like this. In this example, the selected channels are ADC0_IN0, ADC0_IN1, ADC0_IN2, ADC0_IN3, ADC0_IN4.

When SDSFIX bit in ADC_CTL0 register is 1, a fix voltage (configure by SDVOG[1:0] in ADC_CTL0 register) is converted for self-diagnosis.

When SDSFIX bit in ADC_CTL0 register is 0, self-diagnosis takes turns converting 0, $V_{REFP}/2$, V_{REFP} for each scan.

The SH00EN/ SH01EN / SH02EN bits in ADC_SHCTL register are used to select the channels for which the channel-dedicated sample-and-hold circuits are to be used.

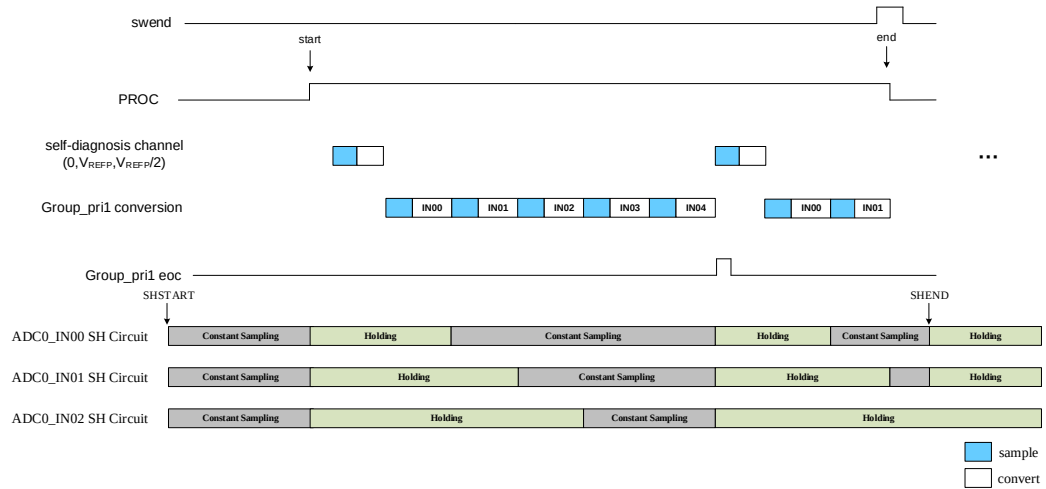
Clear SHMD bit in ADC_SHCTL register to 1 to enable constant sampling.

In continuous scan mode, the internal reference voltage and temperature sensor channels should be deselected.

- (1) Set SHSTART bit in ADC_SHCTL register to 1 to start the selected channel-dedicated Sample-and-Holding circuit working.
- (2) Analog input sampling of all channels for which the channel-dedicated sample-and-hold circuits are to be used is started when the GP1SWST bit in the ADC_EOCCTL register is set to 1 by software, or synchronous or asynchronous trigger input.
- (3) When A/D conversion for self-diagnosis is completed, A/D conversion result is stored into the A/D self-diagnosis data register (ADC_SDDATA), and A/D conversion is performed for the selected channels, starting from the channel with the smallest number (ADC0_IN0 in this example).
- (4) Each time A/D conversion of a single channel is completed, the A/D conversion result is stored into the corresponding A/D data register.
- (5) When A/D conversion of all the selected channels is completed, EOC1F bit in ADC_EOCCTL register is set to 1. At the same time, analog input sampling is started for all the channels for which the channel-dedicated sample-and-hold circuits are to be used. When every the round num ($EOC1RCNT[2:0] + 1$) of end of Group_pri1 conversion comes, this EOC1RF flag will be set to 1 and an interrupt request is generated if the EOC1RIE bit in ADC_EOCCTL is 1.
- (6) The PROC bit is not automatically cleared and steps 2 to 5 are repeated as long as this bit remains 1.
- (7) When the SWEND bit in ADC_CTL0 register is set to 1, A/D conversion stops and the A/D converter enters a wait state.
- (8) Set SHEND bit in ADC_SHCTL register to 1 to terminate the selected channel-dedicated Sample-and-Holding circuit working.

Figure 15-15. Channel selection and self-diagnosis of continuous scan operation of ADC2 (with channel-dedicated sample-and-hold circuits, constant sampling

enabled)



15.5.6 Groups scan mode

Basic operation

Either two (Group_pri1/pri2), three (Group_pri1/pri2/pri3 or Group_pri1/pri2/pri4) or four (Group_pri1/pri2/pri3/pri4) can be selected as the number of the groups to be used in groups scan mode.

In basic operation of groups scan mode, A/D conversion is performed once on the analog inputs of all the specified channels in Group_pri x after scan is started by a synchronous trigger. Scan operation of each group is similar to the scan operation in single scan mode.

Note: software trigger and asynchronous trigger is not recommended to be used for groups scan.

The synchronous triggers of Group_pri x can be selected using the GPxTRGSRC[5:0] in ADC_GPCGF0/1. Different triggers should be used for each group so that scanning of Group_pri1/pri2/pri3/pri4 does not occur simultaneously.

The channels to be scanned are selected using the bits in ADC_CHSEL0/1 registers.

When self-diagnosis is selected in groups scan mode, self-diagnosis is separately executed for Group_pri x.

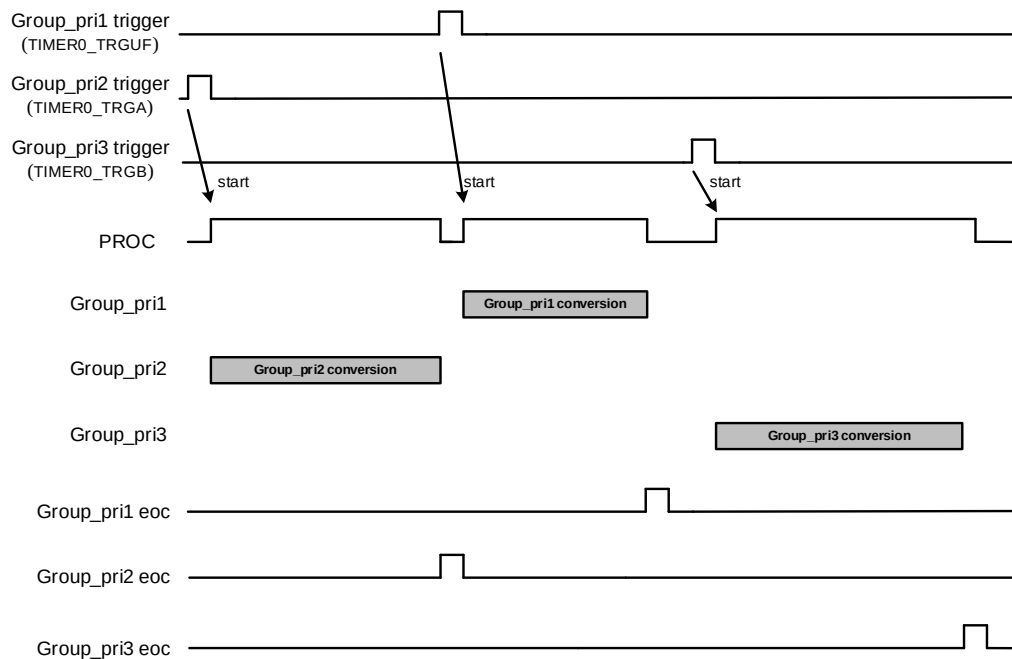
The [Figure 15-16. Groups scan mode basic operation](#) shows basic operation in groups scan mode using a trigger from the TIMER0. The TIMER0_TRGUF, TIMER0_TRGA, and TIMER0_TRGB triggers from the TIMER0 are used to start conversion of Group_pri1/pri2/pri3 respectively.

(1) Scanning of Group_pri1 is started by the TIMER0_TRGUF trigger from the TIMER0.

(2) When Group_pri1 scanning is completed, EOC1F bit in ADC_EOCCTL register is set to

1. When every the round num ($\text{EOC1RCNT}[2:0] + 1$) of end of Group_pri1 conversion comes, this EOC1RF flag will be set to 1 and an interrupt request is generated if the EOC1RIE bit in ADC_EOCCTL is 1.
- (3) Scanning of Group_pri2 is started by the TIMER0_TRGA trigger from the TIMER0.
- (4) When Group_pri2 scanning is completed, EOC2F bit in ADC_EOCCTL register is set to 1. When every the round num ($\text{EOC2RCNT}[2:0] + 1$) of end of Group_pri2 conversion comes, this EOC2RF flag will be set to 1 and an interrupt request is generated if the EOC2RIE bit in ADC_EOCCTL is 1.
- (5) Scanning of Group_pri3 is started by the TIMER0_TRGB trigger from the TIMER0.
- (6) When Group_pri3 scanning is completed, EOC3F bit in ADC_EOCCTL register is set to 1. When every the round num ($\text{EOC3RCNT}[2:0] + 1$) of end of Group_pri3 conversion comes, this EOC3RF flag will be set to 1 and an interrupt request is generated if the EOC3RIE bit in ADC_EOCCTL is 1.

Figure 15-16. Groups scan mode basic operation



A/D conversion in bifurcate trigger mode

Bifurcate trigger mode of Group_pri x is enable by setting the GPxBIMEN bit to 1 in ADC_BITRGCTL register. Bifurcate trigger channel of Group_pri x is set by GPxBICHSEL[4:0] bits in ADC_BITRGCTL register.

When bifurcate trigger mode is selected in groups scan mode, two rounds of single scan operation started by the synchronous trigger are performed for Group_pri x.

The synchronous triggers of Group_pri x can be selected using the GPxTRGSRC[5:0] in

ADC_GPCGF0/1. Different triggers should be used for each group so that scanning of Group_pri1/pri2/pri3/pri4 does not occur simultaneously.

Note: Software trigger and asynchronous trigger is not recommended to be used for groups scan.

If the synchronous trigger is enable and the GPxTRGSRC[5:0] is configured as 6'b000110, 6'b001110, 6'b100101, 6'b101101 or 6'b110011, then the extended bifurcate mode is automatically enabled.

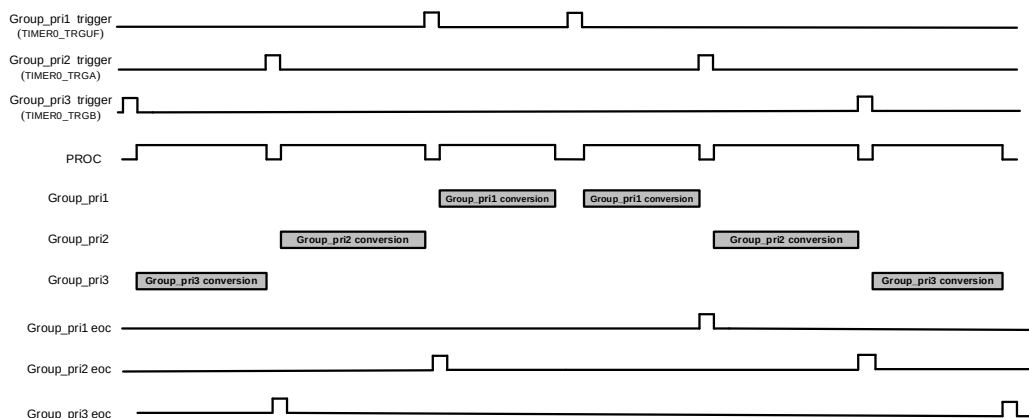
The channel to be scanned is selected using GP4BICHSEL[4:0] bits in ADC_BITRGCTL for Group_pri x.

When bifurcate trigger mode is selected in group scan mode, self-diagnosis cannot be selected.

Basic A/D Conversion in bifurcate trigger mode (without channel-dedicated sample-and-hold circuits)

The [Figure 15-17. Basic A/D conversion in bifurcate trigger mode \(without channel-dedicated sample-and-hold circuits\)](#) shows operation with bifurcate trigger mode enable in Group_pri1 using a synchronous trigger from the TIMER0. The bifurcate trigger mode is not enabled in Group_pri2/pri3. The TIMER0_TRGUF, TIMER0_TRGA, and TIMER0_TRGB triggers from the TIMER0 are assumed to be used to start conversion of Group_pri1/pri2/pri3 respectively.

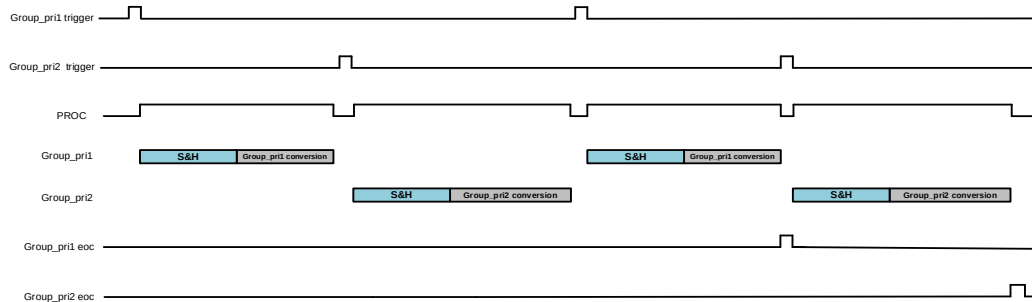
Figure 15-17. Basic A/D conversion in bifurcate trigger mode (without channel-dedicated sample-and-hold circuits)



Basic A/D conversion in bifurcate trigger mode (with channel-dedicated sample-and-hold circuits)

The [Figure 15-18. Basic A/D conversion in bifurcate trigger mode \(with channel-dedicated sample-and-hold circuits\)](#) shows operation with bifurcate trigger mode enable in each group using a synchronous trigger from the TIMER0. The channels using channel-dedicated sample-and-hold circuit are included in Group_pri1/pri2.

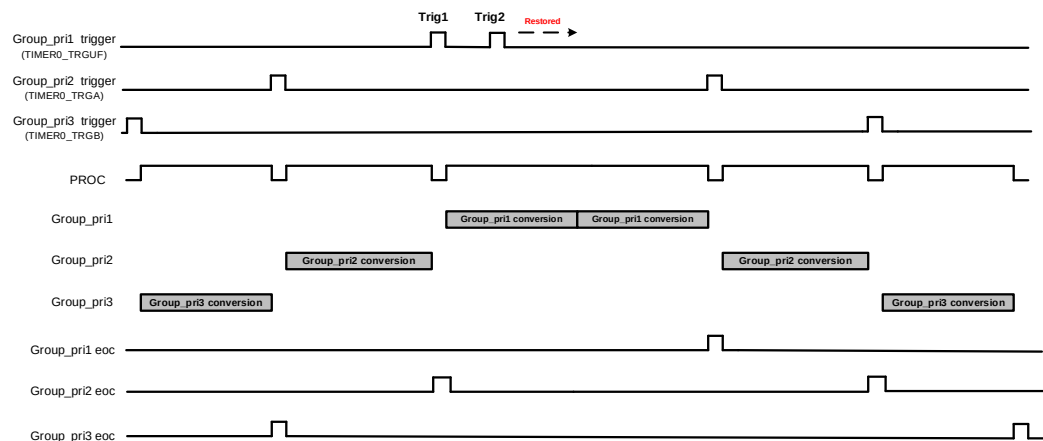
Figure 15-18. Basic A/D conversion in bifurcate trigger mode (with channel-dedicated sample-and-hold circuits)



Basic A/D conversion in bifurcate trigger mode (without channel-dedicated sample-and-hold circuits)

The [Figure 15-19. Basic A/D conversion in bifurcate trigger mode rescan operation \(without channel-dedicated sample-and-hold circuits\)](#) shows operation with bifurcate trigger mode enable in Group_pri1 using a synchronous trigger from the TIMER0. The bifurcate trigger mode is not enabled in Group_pri2/pri3. The TIMER0_TRGUF, TIMER0_TRGA, and TIMER0_TRGB triggers from the TIMER0 are assumed to be used to start conversion of Group_pri1/pri2/pri3 respectively. For Group_pri1, the second trigger comes during the first round A/D conversion started by the first trigger. The second trigger will be restored and the second round A/D conversion will be restarted at the end of the first round A/D conversion if the GP1BITGRS bit in ADC_BITRGCTL register is set to 1 (Otherwise the second trigger will be ignored).

Figure 15-19. Basic A/D conversion in bifurcate trigger mode rescan operation (without channel-dedicated sample-and-hold circuits)

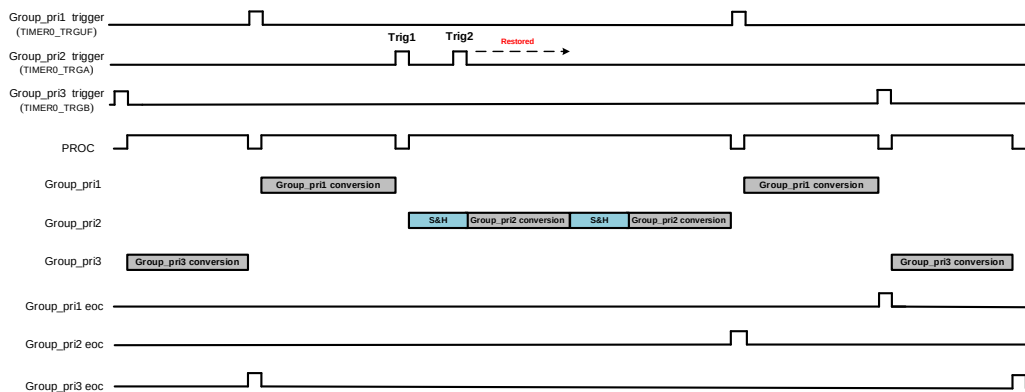


Basic A/D Conversion in bifurcate trigger mode rescan operation (with channel-dedicated sample-and-hold circuits)

The [Figure 15-20. Basic A/D conversion in bifurcate trigger mode rescan operation \(with channel-dedicated sample-and-hold circuits\)](#) shows operation with bifurcate trigger mode enable in Group_pri1/pri2 using a synchronous trigger from the TIMER0. The bifurcate

trigger mode is not enabled in Group_pri3. The TIMER0_TRGUF, TIMER0_TRGA, and TIMER0_TRGB triggers from the TIMER0 are assumed to be used to start conversion of Group_pri1/pri2/pri3 respectively. The channels using channel-dedicated sample-and-hold circuit are included in Group_pri2. For Group_pri2, the second trigger comes during the first round sample-and-holding operation started by the first trigger. The second trigger will be restored and the second round A/D conversion will be restarted at the end of the first round A/D conversion if the GP2BITGRS bit in ADC_BITRGCTL register is set to 1 (Otherwise the second trigger will be ignored).

Figure 15-20. Basic A/D conversion in bifurcate trigger mode rescan operation (with channel-dedicated sample-and-hold circuits)



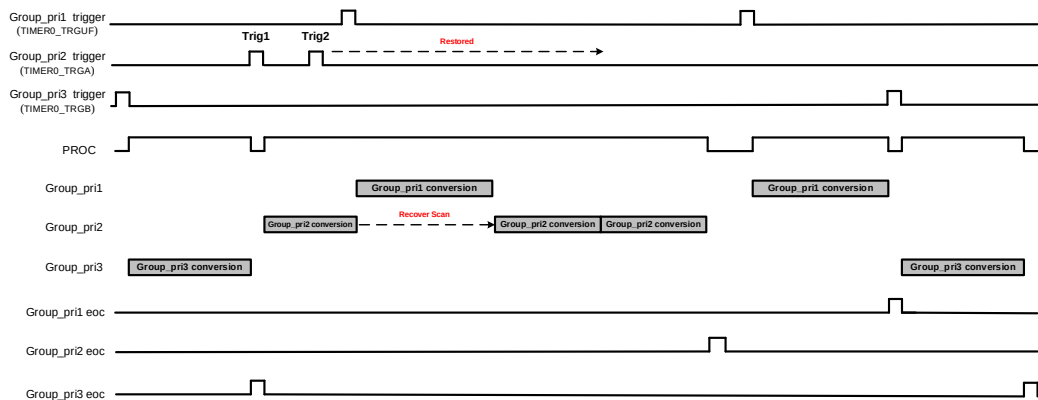
Group rescan operation and group priority control in bifurcate trigger mode (without channel-dedicated sample-and-hold circuits)

The [Figure 15-21. Group rescan operation and group priority control in bifurcate trigger mode \(without channel-dedicated sample-and-hold circuits\)](#) shows operation with bifurcate trigger mode enable in Group_pri1/pri2 using a synchronous trigger from the TIMER0. The bifurcate trigger mode is not enabled in Group_pri3. The TIMER0_TRGUF, TIMER0_TRGA, and TIMER0_TRGB triggers from the TIMER0 are assumed to be used to start conversion of Group_pri1/pri2/pri3 respectively. For Group_pri2, the second trigger comes during the first round A/D conversion started by the first trigger. Before the first round of A/D conversion of Group_pri2 completes, the Group_pri1 trigger comes and breaks the first round of A/D conversion of Group_pri2. After the A/D conversion of Group_pri1 is finished, the first round of A/D conversion of Group_pri2 begins to recover. The second trigger will be restored and the second round A/D conversion will be restarted at the end of the first round A/D conversion of Group_pri2 if the GP2BITGRS bit in ADC_BITRGCTL register is set to 1 (Otherwise the second trigger of Group_pri2 will be ignored).

Note: If group rescan is disabled, the first round of A/D conversion of Group_pri2 will not recover when the A/D conversion of Group_pri1 is finished. The second trigger of Group_pri2 will be equivalent to the trigger of the first round of A/D conversion.

Figure 15-21. Group rescan operation and group priority control in bifurcate trigger

mode (without channel-dedicated sample-and-hold circuits)



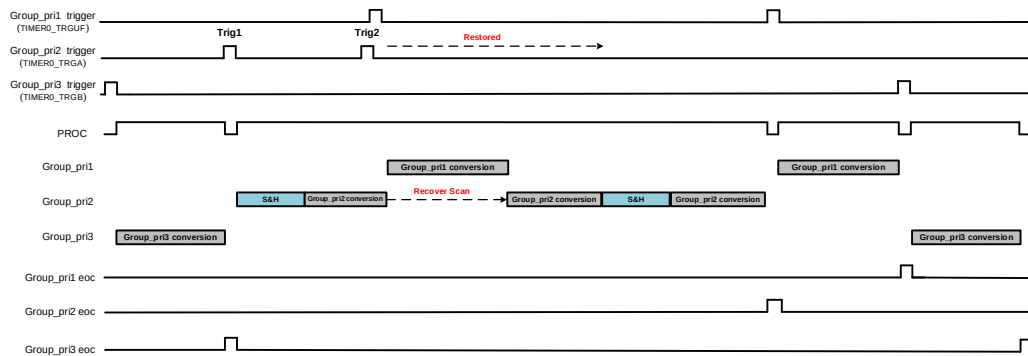
Group rescan operation and group priority control in bifurcate trigger mode (with channel-dedicated sample-and-hold circuits)

The [Figure 15-22. A/D conversion in bifurcate trigger mode rescan operation under group priority control with group rescan enabled \(with channel-dedicated sample-and-hold circuits\)](#) shows operation with bifurcate trigger mode enable in Group_pri1/pri2 using a synchronous trigger from the TIMER0. The bifurcate trigger mode is not enabled in Group_pri3. The TIMER0_TRGUF, TIMER0_TRGA, and TIMER0_TRGB triggers from the TIMER0 are assumed to be used to start conversion of Group_pri1/pri2/pri3 respectively. The channels using channel-dedicated sample-and-hold circuit are included in Group_pri2. For Group_pri2, the second trigger comes during the first round A/D conversion operation started by the first trigger. Before the first round A/D conversion of Group_pri2 completes, the Group_pri1 trigger comes and breaks the first round of A/D conversion of Group_pri2. After the A/D conversion of Group_pri1 is finished, the first round of A/D conversion of Group_pri2 begins to recover. The second trigger will be restored and the second round A/D conversion will be restarted at the end of the first round A/D conversion of Group_pri2 if the GP2BITGRS bit in ADC_BITRGCTL register is set to 1 (Otherwise the second trigger of Group_pri2 will be ignored).

Note: If group rescan is disabled, the first round of A/D conversion of Group_pri2 will not recover when the A/D conversion of Group_pri1 is finished. The second trigger of Group_pri2 will be equivalent to the trigger of the first round of A/D conversion.

Figure 15-22. A/D conversion in bifurcate trigger mode rescan operation under group priority control with group rescan enabled (with channel-dedicated sample-and-hold

circuits)



Operation under group priority control

Setting the GPRIEN bit in ADC_CTL0 register to 1 enable group priority.

The group priority order is Group_pri1 > Group_pri2 > Group_pri3 > Group_pri4. Either two (Group_pri1/pri2), three (Group_pri1/pri2/pri3 or Group_pri1/pri2/pri4) or four (Group_pri1/pri2/pri3/pri4) scan be selected as the number of the groups to be used in group scan mode. If the procedure is not followed, proper scanning operation and data storage are not guaranteed.

In basic operation of group scan mode (GPRIEN = 0), if Group_pri1/pri2/pri3/pri4 is scanning, all other trigger inputs are ignored. If 2 or more trigger inputs come during the same ADC clock period, only the highest priority trigger will start the according group scan, other trigger inputs are ignored.

Under group priority control (GPRIEN = 1), if a priority group trigger is input during A/D conversion for the low-priority group, A/D conversion for the low-priority group is discontinued and A/D conversion for the priority group proceeds.

For any single low-priority group, if the low priority group-dedicated rescan is disabled (the GPxRTEN bit in ADC_GPCGF0/1 is 0), the converter enters a wait state for the low-priority group on completion of the A/D conversion for the priority group. The trigger input for the low-priority group during scanning is ignored.

If the low priority group-dedicated rescan is enabled (GPxRTEN bit in ADC_GPCGF0/1 registers is 1), the converter automatically restarts scanning for the low-priority group after A/D conversion for the priority group. Also, the trigger input for the low-priority group generated during scanning for the priority group is enabled, and the converter automatically restarts scanning for the low-priority group after scanning for the priority group.

When the GPxRTCH bit is 0 while the GPxRTEN bit is 1 in ADC_GPCGF0/1 registers, the converter restarts scanning for the low-priority group from the head of the group. When the GPxRTCH bit is 1, the converter restarts scanning for the low-priority group from the channel on which scanning is discontinued.

When self-diagnosis is used, the converter restarts scanning from the head of the group or

from the discontinued channel according to GPxRTCH setting after self-diagnosis is completed.

[Table 15-3](#), [Table 15-4](#), [Table 15-5](#) and [Table 15-6](#) summarizes operations in response to the input of a trigger during scanning with the low priority group-dedicated rescan setting.

When the LPGCNT bit in ADC_CTL0 is set to 1, scan operations in the lowest-priority group are continuously performed automatically after the first trigger input. Also, the rescan operation is performed automatically ignored of GPxRTEN bit in ADC_GPCGF0/1 registers (lowest priority).

For the trigger settings in groups scan mode, select a synchronous trigger for Group_pri1 using the GP1TRGSRC[5:0] bits, select a synchronous trigger different from that of Group_pri1 for Group_pri2 using the GP2TRGSRC[5:0] bits, and select a synchronous trigger different from those of Group_pri1/2 for Group_pri3 using the GP3TRGSRC[5:0] bits.

Note: When setting the LPGCNT bit in ADC_CTL0 register to 1, It is recommended to disable the synchronous trigger for the lowest priority group.

Furthermore, as targets for scanning, select channels for Group_pri1 using the GP1xxx bits, select channels for Group_pri2 using the GP2xxx bits, select channels for Group_pri3 using the GP3xxx bits, and select channels for Group_pri4 using the GP4xxx bits in ADC_CHSEL0/1 registers.

Table 15-3. Control of scanning operations for groups according to the settings of the GPxRTEN bit (GPRIEN = 1, GP3EN = 1, GP4EN =1)

Scanning Operation	Trigger Input	Scan operation for every group according to GPxRTEN	
		GPxRTEN = 0	GPxRTEN = 1
When A/D conversion for Group_pri1 is in process	trigger for Group_pri1	Trigger input is ineffective.	Trigger input is ineffective.
	trigger for Group_pri2	Trigger input is ineffective. (GP2RTEN = 0)	A/D conversion is performed on Group_pri2 after A/D conversion on Group_pri1 is completed. (GP2RTEN = 1)
	trigger for Group_pri3	Trigger input is ineffective. (GP3RTEN = 0)	A/D conversion is performed on Group_pri3 after A/D conversion on Group_pri1 is completed. (GP3RTEN = 1)
	trigger for Group_pri4	Trigger input is ineffective. (GP4RTEN = 0)	A/D conversion is performed on Group_pri4 after A/D conversion on Group_pri1 is completed. (GP4RTEN = 1)
When A/D conversion for Group_pri2 is in process	trigger for Group_pri1	A/D conversion in process for Group_pri2 discontinued and conversion for Group_pri1 starts.	A/D conversion in process for Group_pri2 is discontinued and conversion for Group_pri1 starts. A/D conversion for Group_pri2 starts

Scanning Operation	Trigger Input	Scan operation for every group according to GPxRTEN	
		GPxRTEN = 0	GPxRTEN = 1
		(GP2RTEN = 0)	after conversion for Group_pri1 is completed. (GP2RTEN = 1)
	trigger for Group_pri2	Trigger input is ineffective. (GP2RTEN = 0)	Trigger input is ineffective. (GP2RTEN = 1)
	trigger for Group_pri3	Trigger input is ineffective. (GP3RTEN = 0)	A/D conversion is performed on Group_pri3 after A/D conversion on Group_pri2 is completed. (GP3RTEN = 1)
	trigger for Group_pri4	Trigger input is ineffective. (GP4RTEN = 0)	A/D conversion is performed on group_pri4 after A/D conversion on Group_pri2 is completed. (GP4RTEN = 1)
When A/D conversion for Group_pri3 is in process	trigger for Group_pri1	A/D conversion in process for Group_pri3 discontinued and conversion for Group_pri1 starts. (GP3RTEN = 0)	A/D conversion in process for Group_pri3 is discontinued and conversion for Group_pri1 starts. A/D conversion for Group_pri3 starts after conversion for Group_pri1 is completed. (GP3RTEN = 1)
	trigger for Group_pri2	A/D conversion in process for Group_pri3 is discontinued and conversion for Group_pri2 starts. (GP3RTEN = 0)	A/D conversion in process for Group_pri3 is discontinued and conversion for Group_pri2 starts. A/D conversion for Group_pri3 starts after conversion for Group_pri2 is completed. (GP3RTEN = 1)
	trigger for Group_pri3	Trigger input is ineffective. (GP3RTEN = 0)	Trigger input is ineffective. (GP3RTEN = 1)
	trigger for Group_pri4	Trigger input is ineffective. (GP4RTEN = 0)	A/D conversion is performed on group_pri4 after A/D conversion on Group_pri3 is completed. (GP4RTEN = 1)
When A/D conversion for Group_pri4 is in process	trigger for Group_pri1	A/D conversion in process for Group_pri4 discontinued and conversion for Group_pri1 starts. (GP4RTEN = 0)	A/D conversion in process for Group_pri4 is discontinued and conversion for Group_pri1 starts. A/D conversion for Group_pri4 starts after conversion for Group_pri1 is completed. (GP4RTEN = 1)
	trigger for Group_pri2	A/D conversion in process for Group_pri4 is discontinued and conversion for Group_pri2 starts.	A/D conversion in process for Group_pri4 is discontinued and conversion for Group_pri2 starts. A/D conversion for Group_pri4 starts

Scanning Operation	Trigger Input	Scan operation for every group according to GPxRTEN	
		GPxRTEN = 0	GPxRTEN = 1
		(GP4RTEN = 0)	after conversion for Group_pri2 is completed. (GP4RTEN = 1)
	trigger for Group_pri3	A/D conversion in process for Group_pri4 is discontinued and conversion for Group_pri3 starts. (GP4RTEN = 0)	A/D conversion in process for Group_pri4 is discontinued and conversion for Group_pri3 starts. A/D conversion for Group_pri4 starts after conversion for Group_pri3 is completed. (GP4RTEN = 1)
	trigger for Group_pri4	Trigger input is ineffective. (GP4RTEN = 0)	Trigger input is ineffective. (GP4RTEN = 1)

Table 15-4. Group priority operation setting and operating mode for Groups_pri1/pri2 (GPRIEN = 1, GP3EN = 0, GP4EN = 0)

Bit Value	Operation
LPGCNT = 0 GP2RTCH = x GP2RTEN = 0	Group priority operation for two groups When a Group_pri1 trigger is input, Group_pri2 scan is completed (not restarted)
LPGCNT = 0 GP2RTCH = 0 GP2RTEN = 1	Group priority operation for two groups After Group_pri2 scan is discontinued, Group_pri2 scan is restarted from the head of the channel selected with the GP2xxx bit in ADC_CHSEL0 register after Group_pri1 scan is completed.
LPGCNT = 0 GP2RTCH = 1 GP2RTEN = 1	Group priority operation for two groups After Group_pri2 scan is discontinued, Group_pri2 scan is restarted from the channel on which scan was discontinued, among the channels selected with the GP2xxx bit in ADC_CHSEL0 register after Group_pri1 scan is completed.
LPGCNT = 1 GP2RTCH = 0 GP2RTEN = x	Group priority operation for two groups Single scan for Group_pri2 is started continuously without start trigger input. After Group_pri2 scan is discontinued, single scan is restarted from the head of the channel selected with the GP2xxx bit in ADC_CHSEL0 register after scan for Group_pri1 is completed.
LPGCNT = 1 GP2RTCH = 1 GP2RTEN = x	Group priority operation for two groups Single scan for Group_pri2 is started continuously without start trigger input. After Group_pri2 scan was discontinued, single scan is restarted from the channel on which scan was discontinued, among the channels selected with the GP2xxx bit in ADC_CHSEL0 register after Group_pri1 scan is completed.

Table 15-5. Group priority operation setting and operating mode for three

Groups_pri1/pri2/pri3 (GPRIEN = 1, GP3EN = 1, GP4EN = 0)

Bit Value	Operation
LPGCNT = x GP2RTCH = x GP2RTEN = 0	Group priority operation for three groups When a Group_pri1 trigger is input, Group_pri2 scan is completed (not restarted)
LPGCNT = x GP2RTCH = 0 GP2RTEN = 1	Group priority operation for three groups After Group_pri2 scan is discontinued, Group_pri2 scan is restarted from the head of the channel selected with the GP2xxx bit in ADC_CHSEL0 register after Group_pri1 scan is completed.
LPGCNT = x GP2RTCH = 1 GP2RTEN = 1	Group priority operation for three groups After Group_pri2 scan is discontinued, Group_pri2 scan is restarted from the channel on which scan was discontinued, among the channels selected with the GP2xxx bit in ADC_CHSEL0 register after Group_pri1 scan is completed.
LPGCNT = 0 GP3RTCH = x GP3RTEN = 0	Group priority operation for three groups When a Group_pri1/pri2 trigger is input, Group_pri3 scan is completed (not restarted)
LPGCNT = 0 GP3RTCH = 0 GP3RTEN = 1	Group priority operation for three groups After Group_pri3 scan is discontinued, Group_pri3 scan is restarted from the head of the channel selected with the GP3xxx bit in ADC_CHSEL1 register after Group_pri1/pri2 scan is completed.
LPGCNT = 0 GP3RTCH = 1 GP3RTEN = 1	Group priority operation for three groups After Group_pri3 scan is discontinued, Group_pri3 scan is restarted from the channel on which scan was discontinued, among the channels selected with the GP3xxx bit in ADC_CHSEL1 register after Group_pri1/pri2 scan is completed.
LPGCNT = 1 GP3RTCH = 0 GP3RTEN = x	Group priority operation for three groups Single scan for Group_pri3 is started continuously without start trigger input. After Group_pri3 scan is discontinued, single scan is restarted from the head of the channel selected with the GP3xxx bit in ADC_CHSEL1 register after scan for Group_pri1/pri2 is completed.
LPGCNT = 1 GP3RTCH = 1 GP3RTEN = x	Group priority operation for three groups Single scan for Group_pri3 is started continuously without start trigger input. After Group_pri3 scan was discontinued, single scan is restarted from the channel on which scan was discontinued, among the channels selected with the GP3xxx bit in ADC_CHSEL1 register after Group_pri1/pri2 scan is completed.

Table 15-6. Group priority operation setting and operating mode for four

Groups(pri1/pri2/pri3/pri4) (GPRIEN = 1, GP3EN = 1, GP4EN =1)

Bit Value	Operation
LPGCNT = x GP2RTCH = x GP2RTEN = 0	Group priority operation for four groups When a Group_pri1 trigger is input, Group_pri2 scan is completed (not restarted)
LPGCNT = x GP2RTCH = 0 GP2RTEN = 1	Group priority operation for four groups After Group_pri2 scan is discontinued, Group_pri2 scan is restarted from the head of the channel selected with the GP2xxx bit in ADC_CHSEL0 register after Group_pri1 scan is completed.
LPGCNT = x GP2RTCH = 1 GP2RTEN = 1	Group priority operation for four groups After Group_pri2 scan is discontinued, Group_pri2 scan is restarted from the channel on which scan was discontinued, among the channels selected with the GP2xxx bit in ADC_CHSEL0 register after Group_pri1 scan is completed.
LPGCNT = x GP3RTCH = x GP3RTEN = 0	Group priority operation for four groups When a Group_pri1/pri2 trigger is input, Group_pri3 scan is completed (not restarted)
LPGCNT = x GP3RTCH = 0 GP3RTEN = 1	Group priority operation for four groups After Group_pri3 scan is discontinued, Group_pri3 scan is restarted from the head of the channel selected with the GP3xxx bit in ADC_CHSEL1 register after Group_pri1/pri2 scan is completed.
LPGCNT = x GP3RTCH = 1 GP3RTEN = 1	Group priority operation for four groups After Group_pri3 scan is discontinued, Group_pri3 scan is restarted from the channel on which scan was discontinued, among the channels selected with the GP3xxx bit in ADC_CHSEL1 register after Group_pri1/pri2 scan is completed.
LPGCNT = 0 GP3RTCH = x GP4RTEN = 0	Group priority operation for four groups When a Group_pri1/pri2/pri3 trigger is input, Group_pri4 scan is completed (not restarted)
LPGCNT = 0 GP3RTCH = 0 GP4RTEN = 1	Group priority operation for four groups After Group_pri4 scan is discontinued, Group_pri4 scan is restarted from the head of the channel selected with the GP4xxx bit in ADC_CHSEL1 register after Group_pri1/pri2/pri3 scan is completed.
LPGCNT = 0 GP3RTCH = 1 GP4RTEN = 1	Group priority operation for four groups After Group_pri4 scan is discontinued, Group_pri4 scan is restarted from the channel on which scan was discontinued, among the channels selected with the GP4xxx bit in ADC_CHSEL1 register after Group_pri1/pri2/pri3 scan is completed.

Bit Value	Operation
LPGCNT = 1 GP3RTCH = 0 GP4RTEN = x	Group priority operation for four groups Single scan for Group_pri4 is started continuously without start trigger input. After Group_pri4 scan is discontinued, single scan is restarted from the head of the channel selected with the GP4xxx bit in ADC_CHSEL1 register after scan for Group_pri1/pri2/pri3 is completed.
LPGCNT = 1 GP3RTCH = 1 GP4RTEN = x	Group priority operation for four groups Single scan for Group_pri4 is started continuously without start trigger input. After Group_pri4 scan was discontinued, single scan is restarted from the channel on which scan was discontinued, among the channels selected with the GP4xxx bit in ADC_CHSEL1 register after Group_pri1/pri2/pri3 scan is completed.

Group priority operation for Group_pri1 & Group_pri2 (GPRIEN = 1, GP3EN = 0, GP4EN = 0)

The following examples show group priority operation in groups scan mode (GP2RTEN = 1, LPGCNT = 0) when ADC0_IN0, ADC0_IN4 is selected for Group_pri1 and ADC0_IN1, ADC0_IN2, ADC0_IN3 are selected for Group_pri2, as shown in [Figure 15-23. Group_pri1 trigger input during Group_pri2 scan, with rescan setting \(GP2RTCH = 1\)](#).

■ Operation example 1: Group_pri1 trigger input during Group_pri2 scan, with rescan setting (GP2RTCH = 1)

- (1) When PROC bit is set to 1 (A/D conversion start) by hardware when external trigger or software start comes, scan for the ADC0_INxx channels selected ADC_CHSEL0/1 registers, starts from the channel with the smallest number.
- (2) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADC_CHxDATA).
- (3) If a Group_pri1 trigger is input during scan for Group_pri2, Group_pri2 scan is discontinued while the PROC bit remains 1, and scan for the ADC0_INxx channels of Group_pri1 selected in the ADC_CHSEL0/1 registers, starts from the channel with the smallest number.

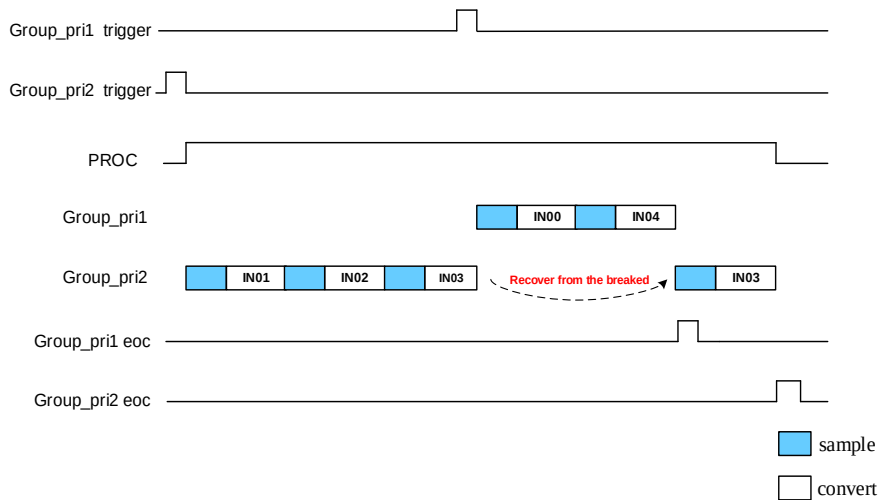
If A/D conversion is not completed when scan was discontinued, the result is not stored in the corresponding A/D data register.

- (4) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADC_CHxDATA).
- (5) When A/D conversion of all the selected channels in the Group_pri1 are completed, EOC1F bit in ADC_EOCCTL register is set to 1. When every the round num (EOC1RCNT[2:0] + 1) of end of Group_pri1 conversion comes, this EOC1RF flag will be set to 1 and an interrupt request is generated if the EOC1RIE bit in ADC_EOCCTL is 1.
- (6) If the GP2RTCH bit is 1 (scan for the group is restarted after having been discontinued

due to group priority operation), scan for the channels of Group_pri2, restarts from the broken channel while the PROC bit remains 1.

- (7) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADC_CHxDATA).
- (8) When A/D conversion of all the selected channels in the Group_pri2 are completed, EOC2F bit in ADC_EOCCTL register is set to 1. When every the round num (EOC2RCNT[2:0] + 1) of end of Group_pri2 conversion comes, this EOC2RF flag will be set to 1 and an interrupt request is generated if the EOC2RIE bit in ADC_EOCCTL is 1.
- (9) The PROC bit is automatically cleared to 0 when scan of all selected channels is completed, and the 12-bit A/D converter enters a wait state.

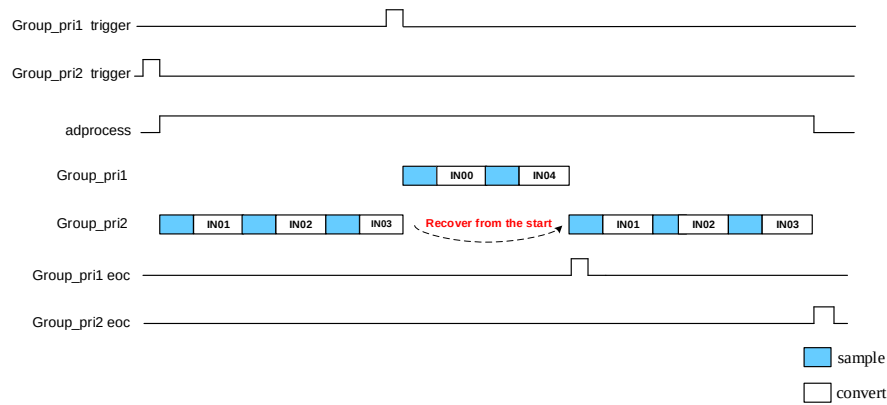
Figure 15-23. Group_pri1 trigger input during Group_pri2 scan, with rescan setting (GP2RTCH = 1)



■ **Operation example 2: Group_pri1 trigger input during Group_pri2 scan, with rescan setting (GP2RTCH = 0)**

Different from example 1, scan for the channels of Group_pri2, restarts from the start channel while the PROC bit remains 1, as shown in [Figure 15-24. Group_pri1 trigger input during Group_pri2 scan, with rescan setting \(GP2RTCH = 0\)](#).

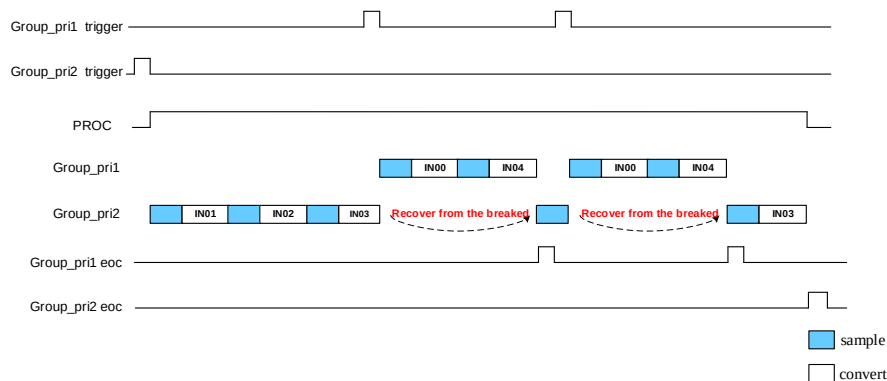
Figure 15-24. Group_pri1 trigger input during Group_pri2 scan, with rescan setting (GP2RTCH = 0)



■ **Operation example 3: Group_pri1 trigger input during Group_pri2 rescan, with rescan setting (GP2RTCH = 1)**

If a Group_pri1 trigger input comes, scan for Group_pri1 starts even while rescan operation is in process. Scan for Group_pri2 starts from the broken channel after scan for Group_pri1 is completed, as shown in [Figure 15-25. Group_pri1 trigger input during Group_pri2 rescan, with rescan setting \(GP2RTCH = 1\)](#).

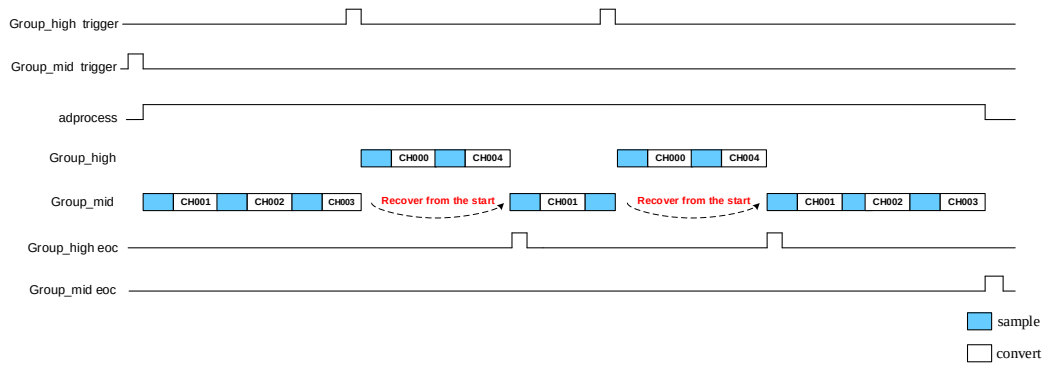
Figure 15-25. Group_pri1 trigger input during Group_pri2 rescan, with rescan setting (GP2RTCH = 1)



■ **Operation example 4: Group_pri1 trigger input during Group_pri2 rescan, with rescan setting (GP2RTCH = 0)**

If a Group_pri1 trigger input comes, scan for Group_pri1 starts even while rescan operation is in process. Scan for Group_pri2 starts from the start channel after scan for Group_pri1 is completed, as shown in [Figure 15-26. Group_pri1 trigger input during Group_pri2 rescan, with rescan setting \(GP2RTCH = 0\)](#).

Figure 15-26. Group_pri1 trigger input during Group_pri2 rescan, with rescan setting (GP2RTCH = 0)



■ Operation example 5: Group_pri2 trigger input during Group_pri1 scan, with rescan setting (GP2RTEN = 1)

The following describes an example when a Group_pri2 trigger input comes during scan operation on Group_pri1 when the GP2RTEN bit is 1 (scan for the group is restarted after having been discontinued due to group priority operation).

If the GP2RTEN bit is 0, all Group_pri2 triggers that are input during scan operation on Group_pri1 are disabled.

- (1) When a Group_pri1 trigger input, the PROC bit is set to 1 (A/D conversion start), scan for the ADC0_INxx channels of Group_pri1 selected in the ADC_CHSEL0/1 registers, starts from the channel with the smallest number.
- (2) If a Group_pri2 trigger is input during scan for Group_pri1, scan for Group_pri2 can be started after Group_pri1 scan completion.
- (3) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADC_CHxDATA).
- (4) When A/D conversion of all the selected channels in the Group_pri1 are completed, EOC1F bit in ADC_EOCCTL register is set to 1. When every the round num (EOC1RCNT[2:0] + 1) of end of Group_pri1 conversion comes, this EOC1RF flag will be set to 1 and an interrupt request is generated if the EOC1RIE bit in ADC_EOCCTL is 1.
- (5) After scan for Group_pri1 is completed, scan for the ADC0_INxx channels of Group_pri2 selected in the ADC_CHSEL0/1 registers, starts from the channel with the smallest number while the PROC bit remains 1.

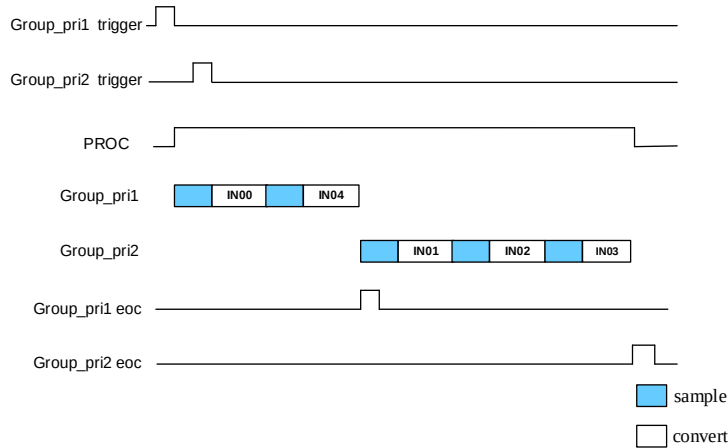
When a Group_pri1 trigger is input during scan for Group_pri2, Group_pri1 scan starts as in [Figure 15-23. Group_pri1 trigger input during Group_pri2 scan, with rescan setting \(GP2RTCH = 1\)](#), and Group_pri2 scan starts after Group_pri1 scan is completed.

- (6) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADC_CHxDATA).
- (7) When A/D conversion of all the selected channels in the Group_pri2 are completed, EOC2F bit in ADC_EOCCTL register is set to 1. When every the round num (EOC2RCNT[2:0] + 1) of end of Group_pri2 conversion comes, this EOC2RF flag will be

set to 1 and an interrupt request is generated if the EOC2RIE bit in ADC_EOCCTL is 1.

- (8) The PROC bit is automatically cleared to 0 when scan of all selected channels is completed, and the 12-bit A/D converter enters a wait state.

Figure 15-27. Group_pri2 trigger input during Group_pri1 scan, with rescan setting (GP2RTEN = 1)



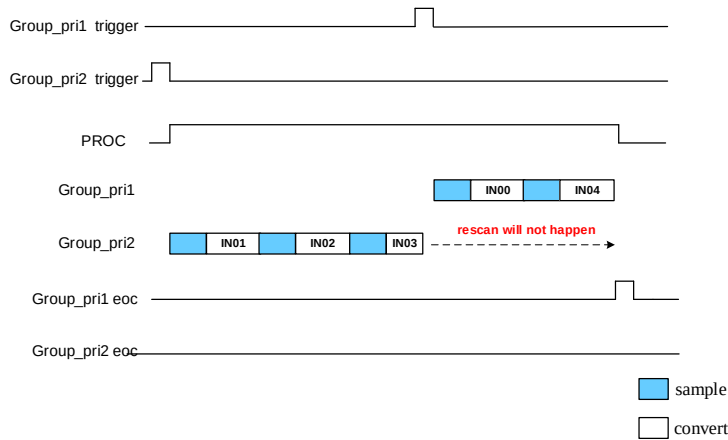
■ **Operation example 6: Group_pri1 trigger input during Group_pri2 scan, without rescan setting (GP2RTEN = 0)**

- (1) When a Group_pri2 trigger input, the PROC bit is set to 1 (A/D conversion start), scan for the ADC0_INxx channels of Group_pri2 selected in the ADC_CHSEL0/1 registers, starts from the channel with the smallest number.
- (2) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADC_CHxDATA).
- (3) If a Group_pri1 trigger is input during scan for Group_pri2, Group_pri2 scan is discontinued while the PROC bit remains 1, and scan for the ADC0_INxx channels of Group_pri1 selected in the ADC_CHSEL0/1 registers, starts from the channel with the smallest number.

If A/D conversion is not completed when scan was discontinued, the result is not stored in the corresponding A/D data register (ADC_CHxDATA).

- (4) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADC_CHxDATA).
- (5) When A/D conversion of all the selected channels in the Group_pri1 are completed, EOC1F bit in ADC_EOCCTL register is set to 1. When every the round num (EOC1RCNT[2:0] + 1) of end of Group_pri1 conversion comes, this EOC1RF flag will be set to 1 and an interrupt request is generated if the EOC1RIE bit in ADC_EOCCTL is 1.
- (6) The PROC bit is automatically cleared to 0 when scan for Group_pri1 is completed, and the 12-bit A/D converter enters a wait state. Scan for Group_pri2 is not started until the next Group_pri2 trigger is input.

Figure 15-28. Group_pri1 trigger input during Group_pri2 scan, without rescan setting (GP2RTEN = 0)



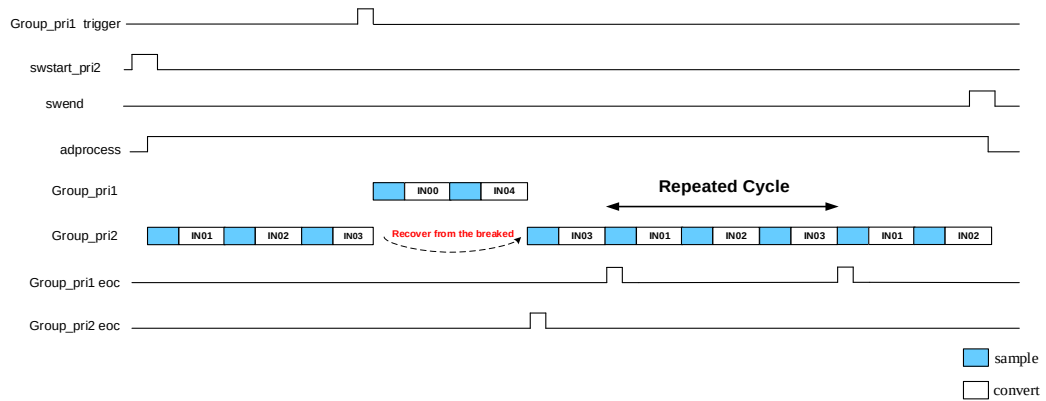
■ **Operation example 7: Continuous single scan operation on Group_pri2(LPGCNT = 1, GP2RTCH = 1)**

- (1) When LPGCNT bit setting to 1, sets the GP2SWST bit to 1 (A/D conversion start by software), scan for the ADC0_INxx channels of Group_pri2 selected in the ADC_CHSEL0/1 registers, starts from the channel with the smallest number.
- (2) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADC_CHxDATA).
- (3) If a Group_pri1 trigger is input during scan for Group_pri2, Group_pri2 scan is discontinued while the PROC bit remains 1, and scan for the ADC0_INxx channels of Group_pri1 selected in the ADC_CHSEL0/1 registers, starts from the channel with the smallest number. If A/D conversion is not completed when scan was discontinued, the result is not stored in the corresponding A/D data register.
- (4) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADC_CHxDATA).
- (5) When A/D conversion of all the selected channels in the Group_pri1 are completed, EOC1F bit in ADC_EOCCTL register is set to 1. When every the round num (EOC1RCNT[2:0] + 1) of end of Group_pri1 conversion comes, this EOC1RF flag will be set to 1 and an interrupt request is generated if the EOC1RIE bit in ADC_EOCCTL is 1.
- (6) If the LPGCNT bit is set to 1 (the lowest-priority group scan continuous), scan for the ADC0_INxx channels of Group_pri2 selected in the ADC_CHSEL0/1 registers, restarts from the channel with the smallest number while the PROC bit remains 1.
- (7) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADC_CHxDATA).
- (8) When A/D conversion of all the selected channels in the Group_pri2 are completed, EOC2F bit in ADC_EOCCTL register is set to 1. When every the round num (EOC2RCNT[2:0] + 1) of end of Group_pri2 conversion comes, this EOC2RF flag will be

set to 1 and an interrupt request is generated if the EOC2RIE bit in ADC_EOCCTL is 1.

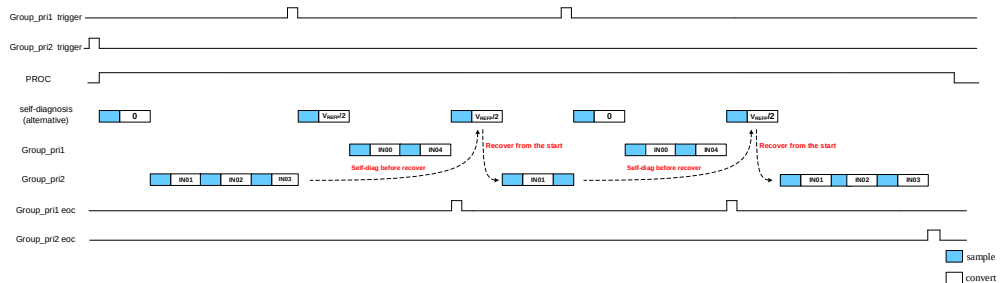
- (9) If the LPGCNT bit is set to 1 (the lowest-priority group scan continuous), scan for the ADC0_INxx channels of Group_pri2 selected in the ADC_CHSEL0/1 registers, restarts from the broken channel while the PROC bit remains 1.

Figure 15-29. Continuous single scan operation on Group_pri2(LPGCNT = 1, GP2RTCH = 1)



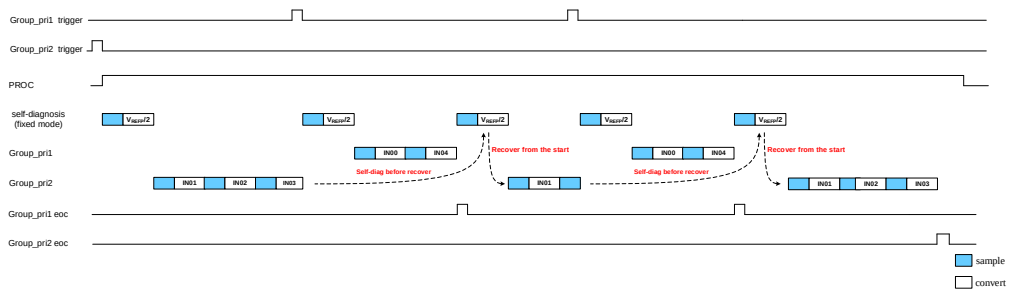
- **Operation example 8: channel selection and alternative self-diagnosis (SDSFIX = 0) with rescan setting (GP2RTCH = 0)**

Figure 15-30. Alternative self-diagnosis (SDSFIX = 0) with rescan setting (GP2RTCH = 0)



- **Operation example 9: channel selection and fixed $V_{REFP}/2$ self-diagnosis (SDSFIX = 1, SDVOG[1:0] = 2'b10) with rescan setting (GP2RTCH = 0)**

Figure 15-31. Fixed self-diagnosis with rescan setting (GP2RTCH = 0)



Group priority operation for Group_pri1 & Group_pri2 & Group_pri3 (GPRIEN = 1, GP3EN = 1, GP4EN = 0)

The following examples show group priority operation in groups scan mode (GP2RTEN = 1, GP2RTCH = 1, GP3RTEN = 1, GP3RTCH = 1) when ADC2_IN3, ADC2_IN8, ADC2_IN9 is selected for Group_pri1, ADC2_IN0, ADC2_IN5, ADC2_IN7 are selected for Group_pri2, and ADC2_IN1, ADC2_IN2, ADC2_IN6 are selected for Group_pri3.

■ Operation example 1: high priority group trigger input during low-priority group scan, with rescan setting (GP2RTCH = 1, GP3RTCH = 1)

- (1) When a Group_pri3 trigger input, the PROC bit is set to 1 (A/D conversion start), scan for the ADC2_INxx channels selected in the ADC_CHSEL0/1 registers, starts from the channel with the smallest number.
- (2) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADC_CHxDATA)
- (3) If a Group_pri2 trigger is input during scan for Group_pri3, Group_pri3 scan is discontinued while the PROC bit remains 1, and scan for the ADC2_INxx channels of Group_pri2 selected in the ADC_CHSEL0/1 registers, starts from the channel with the smallest number. On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADC_CHxDATA)
- (4) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADC_CHxDATA)
- (5) If a Group_pri1 trigger is input during scan for Group_pri2, Group_pri2 scan is discontinued while the PROC bit remains 1, and scan for the ADC2_INxx channels of Group_pri1 selected in the ADC_CHSEL0/1 registers, starts from the channel with the smallest number. If A/D conversion is not completed when scan was discontinued, the result is not stored in the corresponding A/D data register (ADC_CHxDATA).
- (6) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADC_CHxDATA)
- (7) When A/D conversion of all the selected channels in the Group_pri1 are completed, EOC1F bit in ADC_EOCCTL register is set to 1. When every the round num (EOC1RCNT[2:0] + 1) of end of Group_pri1 conversion comes, this EOC1RF flag will be set to 1 and an interrupt request is generated if the EOC1RIE bit in ADC_EOCCTL is 1.
- (8) If the GP2RTEN bit is 1 (scan for the group is restarted after having been discontinued due to group priority operation), scan for the ADC2_INxx channels of Group_pri2 selected in the ADC_CHSEL0/1 registers, restarts from the channel on which A/D conversion was discontinued (GP2RTCH=1) while the PROC bit remains 1.
- (9) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADC_CHxDATA).
- (10) When A/D conversion of all the selected channels in the Group_pri2 are completed,

EOC2F bit in ADC_EOCCTL register is set to 1. When every the round num (EOC2RCNT[2:0] + 1) of end of Group_pri2 conversion comes, this EOC2RF flag will be set to 1 and an interrupt request is generated if the EOC2RIE bit in ADC_EOCCTL is 1.

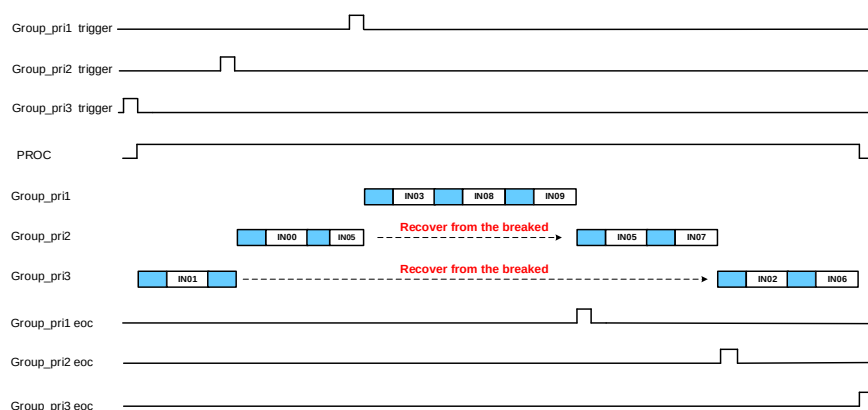
(11) If the GP3RTEN bit is 1, scan for the ADC2_INxx channels of Group_pri3 selected in the ADC_CHSEL0/1 registers, restarts from the channel on which A/D conversion was discontinued (GP3RTCH=1) while the PROC bit remains 1.

(12) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADC_CHxDATA)

(13) When A/D conversion of all the selected channels in the Group_pri3 are completed, EOC3F bit in ADC_EOCCTL register is set to 1. When every the round num (EOC3RCNT[2:0] + 1) of end of Group_pri3 conversion comes, this EOC3RF flag will be set to 1 and an interrupt request is generated if the EOC3RIE bit in ADC_EOCCTL is 1.

(14) The PROC bit is automatically cleared to 0 when scan for Group_pri1 is completed, and the 12-bit A/D converter enters a wait state.

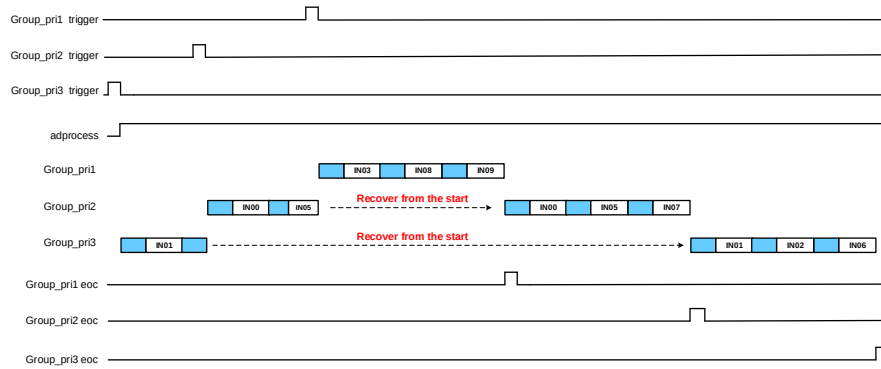
Figure 15-32. High priority group trigger input during low-priority group scan, with rescan setting (GP2RTCH = 1, GP3RTCH = 1)



■ **Operation example 2: high priority group trigger input during low-priority group scan, with rescan setting (GP2RTCH = 0, GP3RTCH = 0)**

Different from example 1, scan for the channels of Group_pri2/3, restarts from the start channel while the PROC bit remains 1, as shown in [Figure 15-33. High priority group trigger input during low-priority group scan, with rescan setting \(GP2RTCH = 0, GP3RTCH = 0\).](#)

Figure 15-33. High priority group trigger input during low-priority group scan, with rescan setting (GP2RTCH = 0, GP3RTCH = 0)

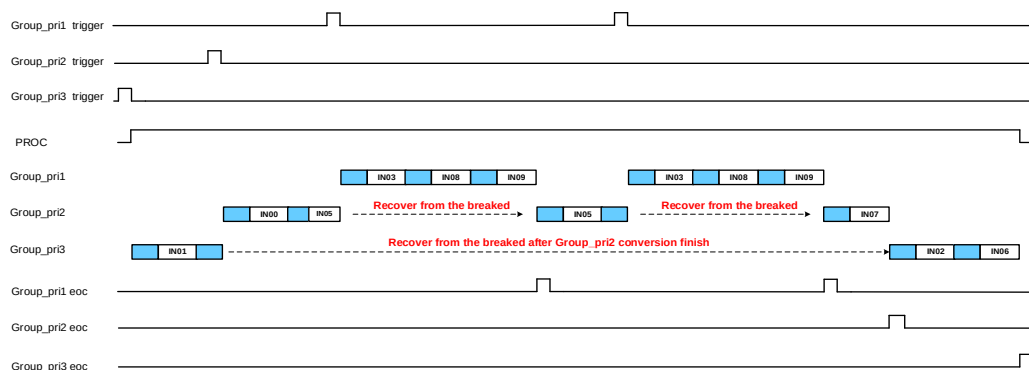


■ **Operation example 3: High priority group trigger input during low-priority group rescan, with rescan setting (GP2RTCH = 1, GP3RTCH = 1)**

If a trigger for the priority groups (Group_pri1/2 for Group_pri3 and Group_pri1 for Group_pri2) is input, scan for the priority group starts even while rescan operation on the low-priority group is in process. After scan for the priority group is completed, scan for the low-priority group is restarted after having been discontinued.

Operations of the PROC bit, storing to the A/D conversion result in the A/D data register (ADC_CHxDATA), and interrupt requests are the same as [Figure 15-34. High priority group trigger input during low-priority group scan, with rescan setting \(GP2RTCH = 1, GP3RTCH = 1\)](#).

Figure 15-34. High priority group trigger input during low-priority group scan, with rescan setting (GP2RTCH = 1, GP3RTCH = 1)



■ **Operation example 4: low-priority group trigger input during priority group scan, with rescan setting (GPxRTEN = 1)**

The following describes an example when a trigger for the low-priority group is input during scan operation on the priority group when the GPxRTEN bit is 1 (scan for the group is restarted after having been discontinued due to group priority operation). If the GPxRTEN bit is 0, all triggers for the low-priority group that are input during scan operation on the priority group are disabled.

- (1) When a Group_pri1 trigger input comes, A/D conversion start (PORC bit setting to 1). Scan for the ADC2_INxx channels of Group_pri1 selected in the ADC_CHSEL0/1

registers, starts from the channel with the smallest number.

- (2) If a Group_pri2 trigger is input during scan for Group_pri1, the trigger of Group_pri2 is suspended.
- (3) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADC_CHxDATA)
- (4) When A/D conversion of all the selected channels in the Group_pri1 are completed, EOC1F bit in ADC_EOCCTL register is set to 1. When every the round num (EOC1RCNT[2:0] + 1) of end of Group_pri1 conversion comes, this EOC1RF flag will be set to 1 and an interrupt request is generated if the EOC1RIE bit in ADC_EOCCTL is 1.
- (5) After scan for Group_pri1 is completed, scan for the ADC2_INxx channels of Group_pri2 selected in the ADC_CHSEL0/1 registers, starts from the channel with the smallest number while the PROC bit remains 1. If the GP2RTCH bit is set to 1 at this time, scan for Group_pri2 starts from the channel on which A/D conversion was discontinued.

When a Group_pri1 trigger is input during scan for Group_pri2, Group_pri1 scan starts as in [Figure 15-23. Group_pri1 trigger input during Group_pri2 scan, with rescan setting \(GP2RTCH = 1\)](#), and Group_pri2 scan starts after Group_pri1 scan is completed.

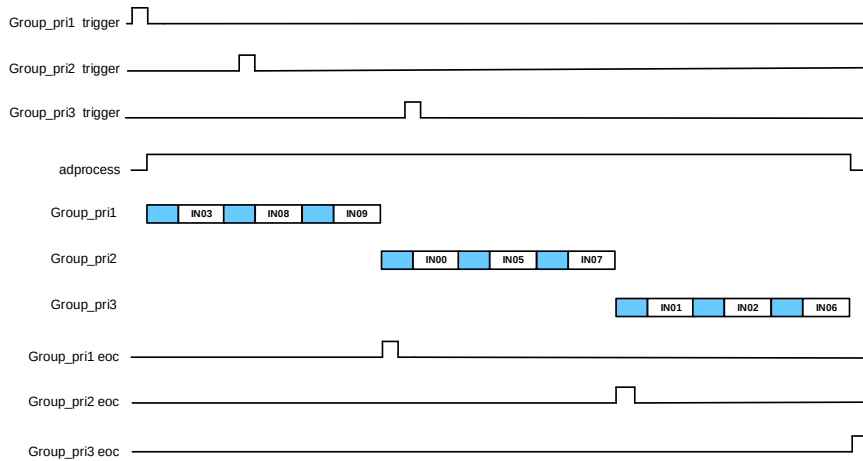
- (6) If a Group_pri3 trigger is input during scan for Group_pri2, the trigger of Group_pri3 is suspended.
- (7) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADC_CHxDATA)
- (8) When A/D conversion of all the selected channels in the Group_pri2 are completed, EOC2F bit in ADC_EOCCTL register is set to 1. When every the round num (EOC2RCNT[2:0] + 1) of end of Group_pri2 conversion comes, this EOC2RF flag will be set to 1 and an interrupt request is generated if the EOC2RIE bit in ADC_EOCCTL is 1.
- (9) After scan for Group_pri2 is completed, scan for the ADC2_INxx channels of Group_pri3 selected in the ADC_CHSEL0/1 registers, starts from the channel with the smallest number while the PROC bit remains 1. If the GP3RTCH bit is set to 1 at this time, scan for Group_pri3 starts from the channel on which A/D conversion was discontinued.

When a Group_pri1 or Group_pri2 trigger is input during scan for Group_pri3, Group_pri1 or Group_pri2 scan starts as in [Figure 15-23. Group_pri1 trigger input during Group_pri2 scan, with rescan setting \(GP2RTCH = 1\)](#), and Group_pri3 scan starts after Group_pri1 or Group_pri2 scan is completed.

- (10) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADC_CHxDATA)
- (11) When A/D conversion of all the selected channels in the Group_pri3 are completed, EOC3F bit in ADC_EOCCTL register is set to 1. When every the round num (EOC3RCNT[2:0] + 1) of end of Group_pri3 conversion comes, this EOC3RF flag will be set to 1 and an interrupt request is generated if the EOC3RIE bit in ADC_EOCCTL is 1.

(12) The PROC bit is automatically cleared to 0 when scan of all selected channels is completed, and the 12-bit A/D converter enters a wait state.

Figure 15-35. Low-priority group trigger input during priority group scan, with rescan setting (GPxRTEN = 1)

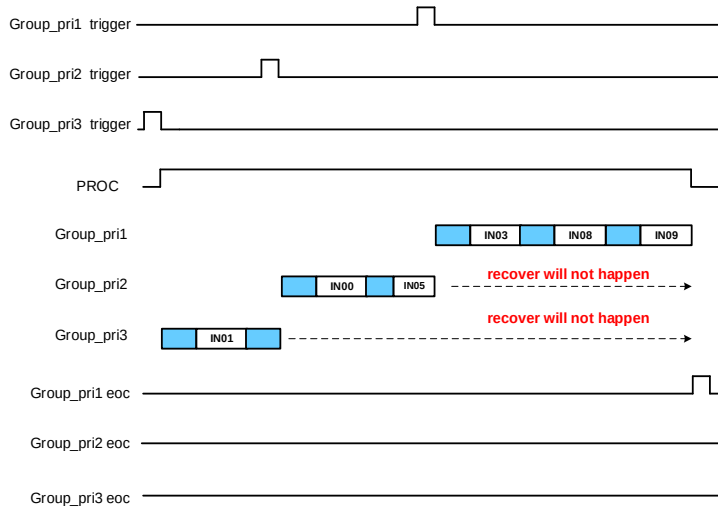


■ **Operation example 5: high priority group trigger input during low-priority group scan, without rescan setting (GPxRTEN = 0)**

- (1) When a Group_pri3 trigger input comes, A/D conversion start (PORC bit setting to 1). Scan for the ADC2_INxx channels selected in the ADC_CHSEL1 registers, starts from the channel with the smallest number.
- (2) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADC_CHXDATA)
- (3) If a Group_pri2 trigger is input during scan for Group_pri3, Group_pri3 scan is discontinued while the PROC bit remains 1, and scan for the ADC2_INxx channels of Group_pri2 selected in the ADC_CHSEL0 registers, starts from the channel with the smallest number. If A/D conversion is not completed when scan was discontinued, the result is not stored in the corresponding A/D data register (ADC_CHXDATA).
- (4) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADC_CHXDATA)
- (5) If a Group_pri1 trigger is input during scan for Group_pri2, Group_pri2 scan is discontinued while the PROC bit remains 1, and scan for the ADC2_INxx channels of Group_pri1 selected in the ADC_CHSEL0 registers, starts from the channel with the smallest number. If A/D conversion is not completed when scan was discontinued, the result is not stored in the corresponding A/D data register (ADC_CHXDATA).
- (6) The PROC bit is automatically cleared to 0 when scan for Group_pri1 is completed, and the 12-bit A/D converter enters a wait state. Scan for Group_pri2 and Group_pri3 is not started until the next trigger corresponding to the group is input.

Figure 15-36. High priority group trigger input during low-priority group scan, without

rescan setting (GPxRTEN = 0)



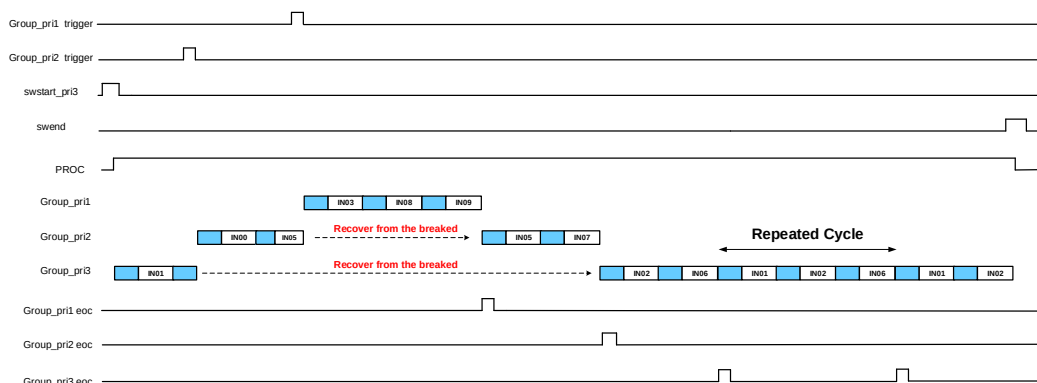
■ Operation example 6: continuous single scan operation on Group_pri3 (GP2RTCH = 1, GP3RTCH = 1)

- (1) When setting LPGCNT bit, sets the GP3SWST bit to 1 (A/D conversion start by software), scan for the ADC2_INxx channels in Group_pri3 selected in the ADC_CHSEL1 registers, starts from the channel with the smallest number.
- (2) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADC_CHXDATA)
- (3) If a Group_pri2 trigger is input during scan for Group_pri3, Group_pri3 scan is discontinued while the PROC bit remains 1, and scan for the ADC2_INxx channels of Group_pri2 selected in the ADC_CHSEL0 registers, starts from the channel with the smallest number. If A/D conversion is not completed when scan was discontinued, the result is not stored in the corresponding A/D data register (ADC_CHXDATA).
- (4) If a Group_pri1 trigger is input during scan for Group_pri2, Group_pri2 scan is discontinued while the PROC bit remains 1, and scan for the ADC2_INxx channels of Group_pri1 selected in the ADC_CHSEL0 registers, starts from the channel with the smallest number. If A/D conversion is not completed when scan was discontinued, the result is not stored in the corresponding A/D data register (ADC_CHXDATA).
- (5) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADC_CHXDATA)
- (6) When A/D conversion of all the selected channels in the Group_pri1 are completed, EOC1F bit in ADC_EOCCTL register is set to 1. When every the round num (EOC1RCNT[2:0] + 1) of end of Group_pri1 conversion comes, this EOC1RF flag will be set to 1 and an interrupt request is generated if the EOC1RIE bit in ADC_EOCCTL is 1.
- (7) If the GP2RTEN bit is 1 (scan for the group is restarted after having been discontinued due to group priority operation), scan for the ADC_INxx channels of Group_pri2 selected

in the ADC_CHSEL0 registers, restarts from the channel with the smallest number while the PROC bit remains 1. If the GP2RTCH bit is set to 1 at this time, scan for Group_pri2 starts from the channel on which A/D conversion was discontinued.

- (8) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADC_CHXDATA)
- (9) When A/D conversion of all the selected channels in the Group_pri2 are completed, EOC2F bit in ADC_EOCCTL register is set to 1. When every the round num (EOC2RCNT[2:0] + 1) of end of Group_pri2 conversion comes, this EOC2RF flag will be set to 1 and an interrupt request is generated if the EOC2RIE bit in ADC_EOCCTL is 1.
- (10) If the LPGCNT bit is 1 (the lowest-priority group scan continuous), scan for the ADC_INxx channels of Group_pri3 selected in the ADC_CHSEL1 register, restarts from the channel with the smallest number while the PROC bit remains 1. If the GP3RTCH bit is set to 1 at this time, scan for Group_pri3 starts from the channel on which A/D conversion was discontinued.
- (11) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADC_CHXDATA)
- (12) When A/D conversion of all the selected channels in the Group_pri3 are completed, EOC3F bit in ADC_EOCCTL register is set to 1. When every the round num (EOC3RCNT[2:0] + 1) of end of Group_pri3 conversion comes, this EOC3RF flag will be set to 1 and an interrupt request is generated if the EOC3RIE bit in ADC_EOCCTL is 1.
- (13) If the LPGCNT bit is set to 1 (the lowest-priority group scan continuous), scan for the ADC2_INxx channels of Group_pri3 elected in the ADC_CHSEL1 register, restarts from the channel with the smallest number while the PROC bit remains 1.

Figure 15-37. Continuous single scan operation on Group_pri3 (GP2RTCH = 1, GP3RTCH = 1)



15.5.7 Configureable channel priority function

The whole channel priority is configured in ADC channel priority registers (ADC_CHPRI0/1). In every single group, the channels selected in ADC_CHSEL0/1 register for the Group_pri

are converted by the order according to the whole channel priority.

For example of ADC2,

- The PRI0[3:0]~ PRI6[3:0] bits in ADC_CHPRI0 register are configured as 5,2,4,6,3,0,1 (ADC2_IN5 > ADC2_IN2 > ADC2_IN4 > ADC2_IN6 > ADC2_IN3 > ADC2_IN0 > ADC2_IN1), as shown in [Table 15-7. Example of channel priority setting in ADC2](#).
- The GP1CH0~GP1CH6 bits in ADC_CHSEL0 register are configured as 1,0,1,0,1,0,0 (ADC2_IN0/2/4 are selected). The GP2CH0~GP2CH6 bits in ADC_CHSEL0 register are configured as 0,1,0,1,0,1,1 (ADC2_IN1/3/5/6 are selected).

The channels conversion order of Group_pri1 is ADC2_IN2, ADC2_IN4, ADC2_IN0 ..., and the channels conversion order of Group_pri2 is ADC2_IN5, ADC2_IN6, ADC2_IN3, ADC2_IN1

Note: Setting the same channel in the ADC_CHPRI0/1 registers is prohibited.

Channels whose priority can be set with this function are analog channels, internal reference voltage channel and temperature sensor channel only. The conversion order for the self-diagnosis is excluded from this function and cannot be changed. The self-diagnosis has the highest priority.

Table 15-7. Example of channel priority setting in ADC2

ADC_CHPRI0 setting		Channel and priority		ADC_CHSEL0 setting	
bits	Value	Channel	Priority	Group_pri1	Group_pri2
PRI0[3:0]	5	ADC2_IN5	High ↓	GP1CH00 = 1	GP2CH00 = 0
PRI1[3:0]	2	ADC2_IN2		GP1CH01 = 0	GP2CH01 = 1
PRI2[3:0]	4	ADC2_IN4		GP1CH02 = 1	GP2CH02 = 0
PRI3[3:0]	6	ADC2_IN6	Low	GP1CH03 = 0	GP2CH03 = 1
PRI4[3:0]	3	ADC2_IN3		GP1CH04 = 1	GP2CH04 = 0
PRI5[3:0]	0	ADC2_IN0		GP1CH05 = 0	GP2CH05 = 1
PRI6[3:0]	1	ADC2_IN1		GP1CH06 = 0	GP2CH06 = 1

Figure 15-38. Actual A/D conversion sequence of Group_pri1 in ADC2

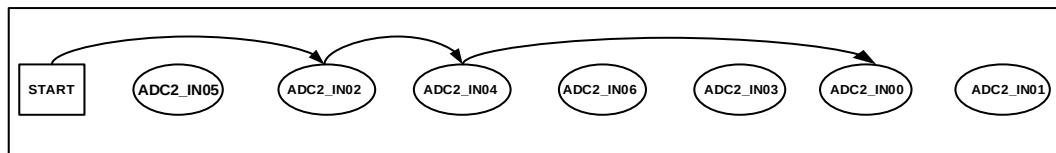
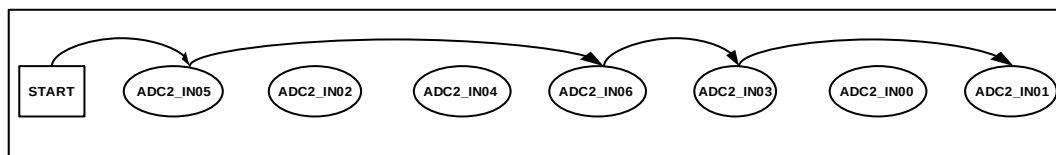


Figure 15-39. Actual A/D conversion sequence of Group_pri2 in ADC2



15.5.8 Conversion result threshold monitor

The ADCx module has two analog watchdog (watchdog A and watchdog B). These two watchdogs allow monitoring of input voltages during A/D conversion and generate interrupts or flags when the input voltage exceeds preset thresholds.

Note: When this function is used, self-diagnosis or bifurcate trigger mode cannot be used.

Analog watchdog A

- **Enable analog watchdog A:** Set the WDAEN bit to 1 in the ADC_WDCTL register to enable analog watchdog A.
- **Operation:** The comparison function compares the reference values set in the ADC_WDATHOLD register with the A/D conversion results.
- **Channel Monitoring:**
For analog watchdog A, the multiple channels selected by the WDACHx bits in the ADC_WDACHCFG register are monitored.
The channel-specific match condition is configured in the WDACHxM bits in the ADC_WDACHCFG register.
- **Flags and Interrupts:**
If the conversion data of one or more channels matches the channel-specific condition setting, the channel-specific compare status signal CHxCMPF bits in the ADC_WDACHSTAT register will be set to 1 by hardware.

An interrupt request is generated if the WDAIE bit in the ADC_WDCTL is 1.

Analog watchdog B

- **Enable analog watchdog B:** Set the WDBEN bit to 1 in the ADC_WDCTL register to enable Watchdog B.
- **Operation:** The comparison function compares the reference values set in the ADC_WDBTHOLD register with the A/D conversion results.
- **Channel Monitoring:**
For analog watchdog B, the channel selected by the WDBCHSEL[3:0] bits in the ADC_WDCTL register is monitored.
The analog watchdog B match condition is configured in the WDBCM bit in the ADC_WDCTL register.
- **Flags and Interrupts:**
If the conversion data of this channel matches the condition setting, the WDBEF bit in the ADC_WDCTL register will be set to 1 by hardware.
An interrupt request is generated if the WDBIE bit in the ADC_WDCTL is 1.

Window comparison function

A window comparison function can also be used (when WINEN in the ADC_WDCTL register is 1), allowing for the comparison of two values.

15.5.9 Data storage mode

When A/D-converted value addition/average mode is disabled or A/D-converted average mode is selected

The alignment of data after conversion or averaging is stored can be specified by DAL bit in the ADC_CTL0 register., which specifies LSB alignment or MSB alignment for the data in ADC_SDDATA, ADC_GPxBIDATA, ADC_GPxBIDATA1, ADC_GPxBIDATA2, ADC_GPxDMAR, ADC_CHxDATA, ADC_TEMPDATA and ADC_VINTDATA. When A/D-converted average mode is selected, 2,3,4,8,16,32,64 additional times can be selected.

Figure 15-40. Data storage mode of 12-bit resolution

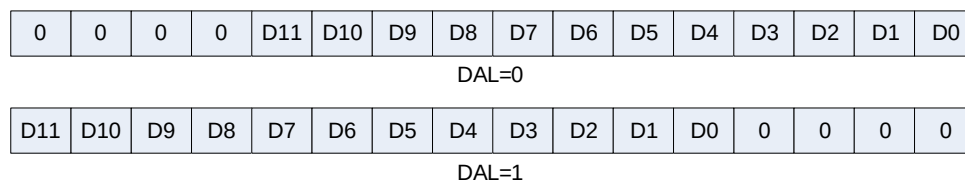


Figure 15-41. Data storage mode of 10-bit resolution

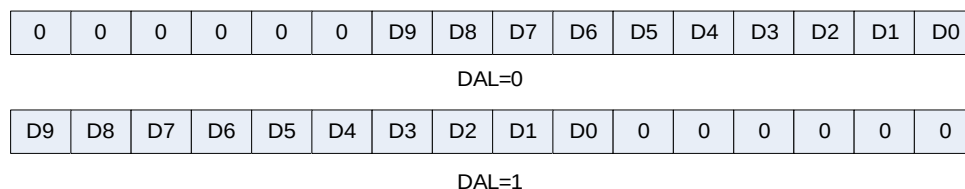


Figure 15-42. Data storage mode of 8-bit resolution

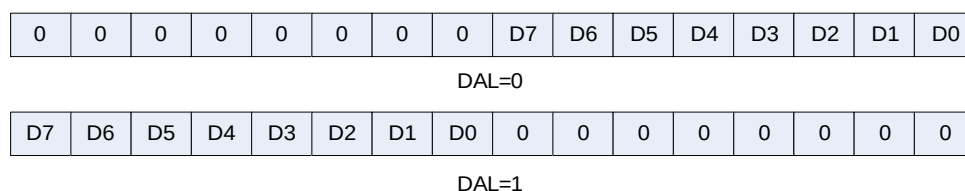
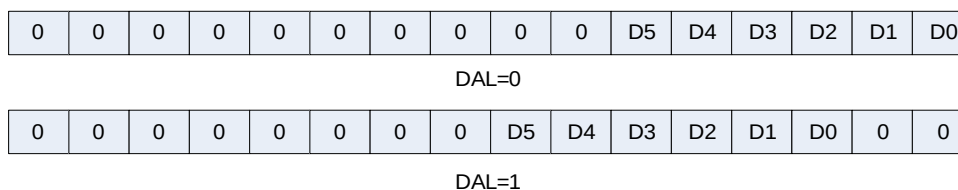


Figure 15-43. Data storage mode of 6-bit resolution


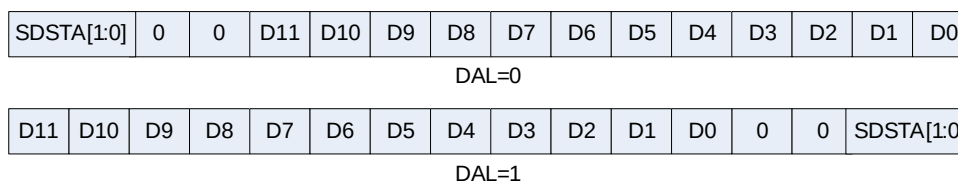
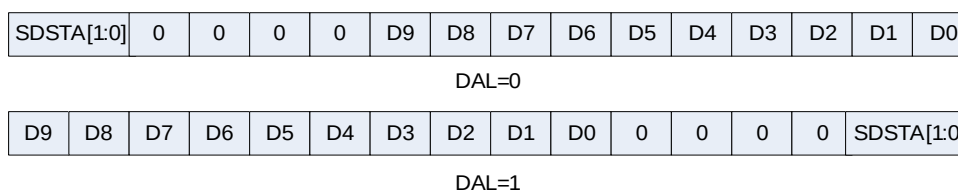
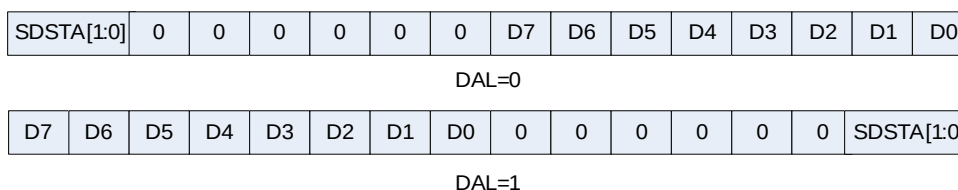
The self-diagnosis data register (ADC_SDDATA) is divided into two parts: the self-diagnosis channel conversion result (SDDATA) and the voltage status of the self-diagnosis conversion (SDSTA[1:0]).

The voltage status of the self-diagnosis conversion (SDSTA[1:0]):

- 2'b00: Self-diagnosis has never been executed since power-on.
- 2'b01: 0V converted by self-diagnosis.
- 2'b10: $V_{REFP}/2$ converted by self-diagnosis.
- 2'b11: V_{REFP} converted by self-diagnosis.

For 12/10/8-bit resolution:

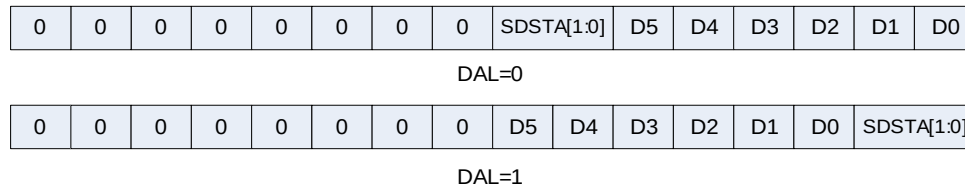
- If DAL=0, SDSTA[1:0] occupies bits [15:14] of the ADC_SDDATA register.
- If DAL=1, SDSTA[1:0] occupies bits [1:0] of the ADC_SDDATA register.

Figure 15-44. Self-Diagnosis Data storage mode of 12-bit resolution

Figure 15-45. Self-Diagnosis Data storage mode of 10-bit resolution

Figure 15-46. Self-Diagnosis Data storage mode of 8-bit resolution


For 6-bit resolution:

- If DAL=0, SDSTA[1:0] occupies bits [7:6] of the ADC_SDDATA register.
- If DAL=1, SDSTA[1:0] occupies bits [1:0] of the ADC_SDDATA register.

Figure 15-47. Self-Diagnosis Data storage mode of 6-bit resolution



When A/D-converted value addition mode is selected

The alignment of data after accumulation by the additional times is stored can be specified by DAL bit in the ADC_CTL0 register., which specifies LSB alignment or MSB alignment for the data in ADC_SDDATA, ADC_GPxBIDATA, ADC_GPxBIDATA1, ADC_GPxBIDATA2, ADC_GPxDMAR, ADC_CHxDATA, ADC_TEMPDATA and ADC_VINTDATA.

Data storage mode of 12-bit resolution

When A/D-converted addition mode is selected, 2,3,4,8,16 additional times can be selected, 32,64 additional times are prohibited.

(1) When 2 times is selected

LSB alignment:

The value added by the A/D-converted value of the same channel is stored in bits 12 to 0.

MSB alignment:

The value added by the A/D-converted value of the same channel is stored in bits 15 to 3.

(2) When 3 or 4 times is selected

LSB alignment:

The value added by the A/D-converted value of the same channel is stored in bits 13 to 0.

MSB alignment:

The value added by the A/D-converted value of the same channel is stored in bits 15 to 2.

(3) When 8 times is selected

LSB alignment:

The value added by the A/D-converted value of the same channel is stored in bits 14 to 0.

MSB alignment:

The value added by the A/D-converted value of the same channel is stored in bits 15 to 1.

(4) When 16 times is selected

LSB alignment:

The value added by the A/D-converted value of the same channel is stored in bits 15 to 0.

MSB alignment:

The value added by the A/D-converted value of the same channel is stored in bits 15 to 0.

Data storage mode of 10-bit resolution

When A/D-Converted addition mode is selected, 2,3,4,8,16,32,64 additional times can be selected.

(1) When 2 times is selected

LSB alignment:

The value added by the A/D-converted value of the same channel is stored in bits 10 to 0.

MSB alignment:

The value added by the A/D-converted value of the same channel is stored in bits 15 to 5.

(2) When 3 or 4 times is selected

LSB alignment:

The value added by the A/D-converted value of the same channel is stored in bits 11 to 0.

MSB alignment:

The value added by the A/D-converted value of the same channel is stored in bits 15 to 4.

(3) When 8 times is selected

LSB alignment:

The value added by the A/D-converted value of the same channel is stored in bits 12 to 0.

MSB alignment:

The value added by the A/D-converted value of the same channel is stored in bits 15 to 3.

(4) When 16 times is selected

LSB alignment:

The value added by the A/D-converted value of the same channel is stored in bits 13 to 0.

MSB alignment:

The value added by the A/D-converted value of the same channel is stored in bits 15 to 2.

(5) When 32 times is selected

LSB alignment:

The value added by the A/D-converted value of the same channel is stored in bits 14 to 0.

MSB alignment:

The value added by the A/D-converted value of the same channel is stored in bits 15 to 1.

(6) When 64 times is selected

LSB alignment:

The value added by the A/D-converted value of the same channel is stored in bits 15 to 0.

MSB alignment:

The value added by the A/D-converted value of the same channel is stored in bits 15 to 0.

Data storage mode of 8-bit resolution

When A/D-Converted addition mode is selected, 2,3,4,8,16,32,64 additional times can be selected.

(1) When 2 times is selected

LSB alignment:

The value added by the A/D-converted value of the same channel is stored in bits 8 to 0.

MSB alignment:

The value added by the A/D-converted value of the same channel is stored in bits 15 to 7.

(2) When 3 or 4 times is selected

LSB alignment:

The value added by the A/D-converted value of the same channel is stored in bits 9 to 0.

MSB alignment:

The value added by the A/D-converted value of the same channel is stored in bits 15 to 6.

(3) When 8 times is selected

LSB alignment:

The value added by the A/D-converted value of the same channel is stored in bits 10 to 0.

MSB alignment:

The value added by the A/D-converted value of the same channel is stored in bits 15 to 5.

(4) When 16 times is selected

LSB alignment:

The value added by the A/D-converted value of the same channel is stored in bits 11 to 0.

MSB alignment:

The value added by the A/D-converted value of the same channel is stored in bits 15 to 4.

(5) When 32 times is selected

LSB alignment:

The value added by the A/D-converted value of the same channel is stored in bits 12 to 0.

MSB alignment:

The value added by the A/D-converted value of the same channel is stored in bits 15 to 3.

(6) When 64 times is selected

LSB alignment:

The value added by the A/D-converted value of the same channel is stored in bits 13 to 0.

MSB alignment:

The value added by the A/D-converted value of the same channel is stored in bits 15 to 2.

Data storage mode of 6-bit resolution

When A/D-Converted addition mode is selected, 2,3,4,8,16,32,64 additional times can be selected.

(1) When 2 times is selected

LSB alignment:

The value added by the A/D-converted value of the same channel is stored in bits 6 to 0.

MSB alignment:

The value added by the A/D-converted value of the same channel is stored in bits 15 to 9.

(2) When 3 or 4 times is selected

LSB alignment:

The value added by the A/D-converted value of the same channel is stored in bits 7 to 0.

MSB alignment:

The value added by the A/D-converted value of the same channel is stored in bits 15 to 8.

(3) When 8 times is selected

LSB alignment:

The value added by the A/D-converted value of the same channel is stored in bits 8 to 0.

MSB alignment:

The value added by the A/D-converted value of the same channel is stored in bits 15 to 7.

(4) When 16 times is selected

LSB alignment:

The value added by the A/D-converted value of the same channel is stored in bits 9 to 0.

MSB alignment:

The value added by the A/D-converted value of the same channel is stored in bits 15 to 6.

(5) When 32 times is selected

LSB alignment:

The value added by the A/D-converted value of the same channel is stored in bits 10 to 0.

MSB alignment:

The value added by the A/D-converted value of the same channel is stored in bits 15 to 5.

(6) When 64 times is selected

LSB alignment:

The value added by the A/D-converted value of the same channel is stored in bits 11 to 0.

MSB alignment:

The value added by the A/D-converted value of the same channel is stored in bits 15 to 4.

15.5.10 Sample time configuration

The number of CK_ADC cycles which is used to sample the input voltage can be specified by the SPTn[7:0] bits in the ADC_SAMPRx(x=0..2) registers. A different sample time can be specified for each channel. For 12-bits resolution, the total sampling and conversion time is "sampling time + 12.5" CK_ADC cycles.

Example: CK_ADC = 36MHz and sample time is 2.5 cycles, the total conversion time is "2.5+12.5" CK_ADC cycles, that means 0.4167us.

15.5.11 External trigger configuration

Conversion trigger requests for Group_prix can be generated by software, synchronous signals, and asynchronous signal.

Software trigger

Setting the GPxSWST bit in the ADC_EOCCTL register to 1 will generate a Group_prix conversion trigger request. When any channel starts conversion, the hardware clears the GPxSWST bit.

Synchronous/asynchronous trigger

Setting the GxTRGEN bit of the ADC_GPCGF0/1 register to 1 enables synchronous and asynchronous trigger request. Setting the GPxTRGTY bit of the ADC_GPCGF0/1 registers to 1 selects asynchronous triggering; otherwise, synchronous triggering is selected, as shown in [Table 15-8. ADC synchronous/asynchronous trigger enable](#).

Table 15-8. ADC synchronous/asynchronous trigger enable

GxTRGEN	GPxTRGTY	Trigger type
0	-	Disable synchronous / asynchronous trigger
1	0	Enable synchronous
1	1	Enable asynchronous

The rising or falling edge of the synchronous or asynchronous signal can trigger the conversion of Group_prix, selected by GxTRGEN[1:0] bits of the ADC_GPCGF0/1 registers, as shown in [Table 15-9. External trigger edge](#).

Table 15-9. External trigger edge

GPxTRGED[1:0]	Trigger edge
00	External trigger disable
01	Rising edge of external trigger enable
10	Falling edge of external trigger enable
11	Rising and falling edge of external trigger enable

The GPxTRGSRC[5:0] bits of the ADC_GPCGF0/1 registers select the synchronous trigger source for Group_prix, as detailed in [Table 15-10. ADC synchronous trigger source](#).

The asynchronous trigger sources for ADC0/2 are fixed (from pins), detailed in [Table 15-11. ADC asynchronous trigger source](#).

Table 15-10. ADC synchronous trigger source

GPxTRGSRC[5:0]	Trigger source	Description
6'b000001	TIMER0_TRGOF	From TIMER
6'b000010	TIMER0_TRGUF	
6'b000011	TIMER0_TRGA	
6'b000100	TIMER0_TRGB	
6'b000101	TIMER0_TRGAB	
6'b000110	TIMER0_TRGAORB	
6'b000111	TIMER0_TRGO	
6'b001001	TIMER7_TRGOF	
6'b001010	TIMER7_TRGUF	
6'b001011	TIMER7_TRGA	
6'b001100	TIMER7_TRGB	
6'b001101	TIMER7_TRGAB	
6'b001110	TIMER7_TRGAORB	
6'b001111	TIMER7_TRGO	

GPxTRGSRC[5:0]	Trigger source	Description
6'b010111	TIMER1_TRGO	
6'b011111	TIMER2_TRGO	
6'b100011	GPTIMER0_TRGA	From GPTIMER
6'b100100	GPTIMER0_TRGB	
6'b100101	GPTIMER0_TRGAB	
6'b101011	GPTIMER1_TRGA	
6'b101100	GPTIMER1_TRGB	
6'b101101	GPTIMER1_TRGAB	
6'b110001	EVSEL0	From EVIC
6'b110010	EVSEL1	
6'b110011	EVSEL0 or EVSEL1	

Table 15-11. ADC asynchronous trigger source

ADCx	Trigger Source	Description
ADC0	ADC0_TRG	From MCU pin
ADC2	ADC2_TRG	From MCU pin

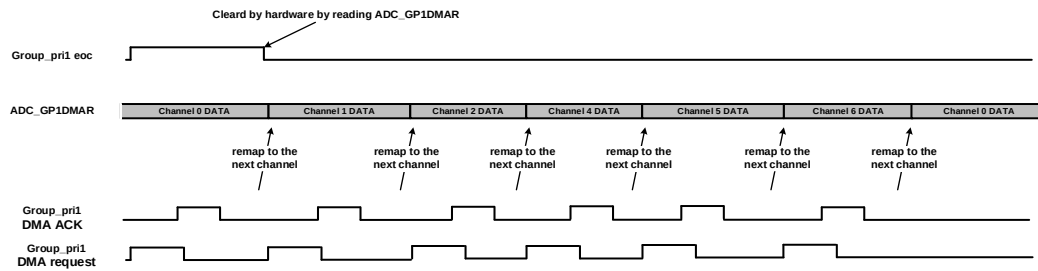
15.5.12 DMA function

The DMA request of Group_pri x, which is enabled by the DMAENx bit in ADC_DMACTL register, is used to transfer data of Group_pri x for conversion of more than one channel. The ADC generates a DMA request of Group_pri x at the end of conversion of Group_pri x. During the request of Group_pri x is high, all the updated data of ADC_CHnDATA which belongs to Group_pri x will be transferred to the destination location which is specified by the user. When a DMA acknowledgement signal of Group_pri x comes, the data from the ADC_GPxDMAR will remap to data of the next channel which belongs to Group_pri x. When all the updated data ADC_CHnDATA of Group_pri x is transferred to the destination location which is specified by the user, the request will be driven low by the ADC.

When GPxDDM in ADC_DMACTL register is set, the according group will continue generating DMA requests after the end of every DMA transfer. It's recommended to set GPxDDM to 1 only when circular mode is enabled in DMA.

[**Figure 15-48. Example of a DMA transfer of Group_pri1 in ADC2**](#) provides an example of ADC2 Group_pri1 DMA transfer behavior, with the Group_pri1 selecting channels ADC2_IN0, ADC2_IN1, ADC2_IN2, ADC2_IN4, ADC2_IN5, and ADC2_IN6.

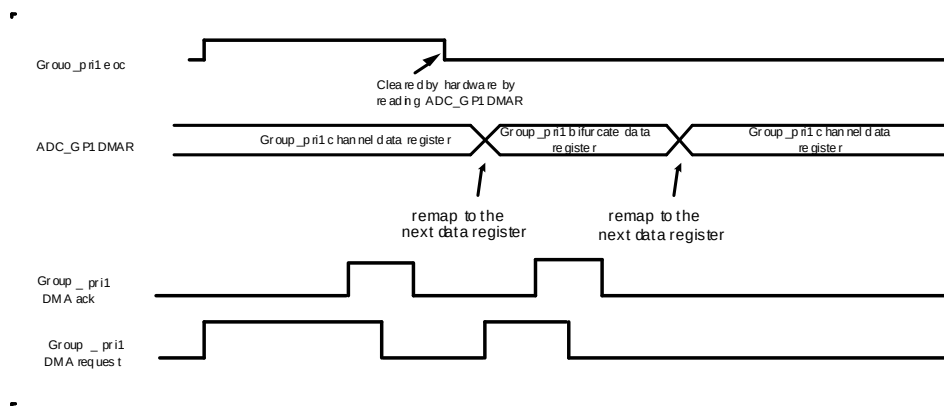
Figure 15-48. Example of a DMA transfer of Group_pri1 in ADC2



When bifurcate trigger function is used in a group (Group_pri x), after eoc the DMA data register will first map to the ADC_CHnDATA register (channel is selected by GPxBICHSEL[4:0] in ADC_BITRGCTL register), then remap to the ADC_GPxBIDATA, as shown in [Figure 15-49. Example of a DMA transfer of Group_pri1 \(bifurcate trigger function enabled\)](#).

Note: When ADC data registers automatic clearing function is used, after remapping operation, the ADC_CHnDATA and the ADC_GPxBIDATA will be cleared accordingly.

Figure 15-49. Example of a DMA transfer of Group_pri1 (bifurcate trigger function enabled)



15.5.13 DMA overflow detection

Overflow Detection

Overflow detection of Group_pri x is enabled when DMAENx and OVRENx in ADC_DMACTL register are set to 1. An overflow event occurs when a single channel of Group_pri x conversion is done before the prior data of ADC_GPxDMAR which maps to the single channel has been transferred to the destination location which is specified by the user, and the GPxOVRF is set to 1 by hardware. If the GPxOVRIE bit in ADC_DMACTL register is set, overflow interrupt will be generated.

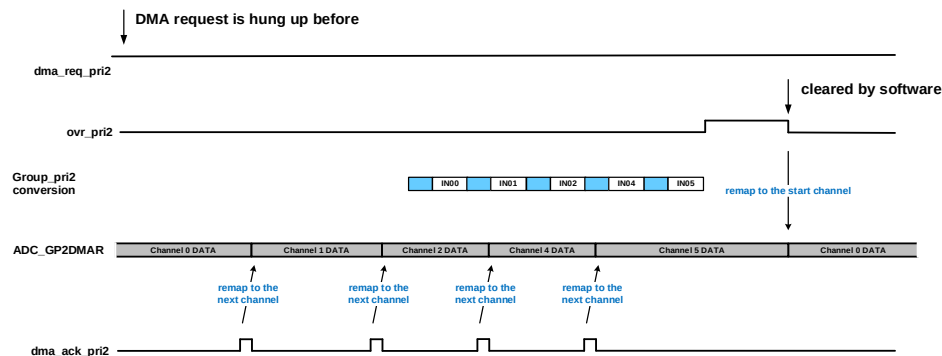
Overflow Handling

It is recommended to reinitialize the DMA module to recover the ADC from the overflow state. To ensure the converted datas of Group_pri x are transferred correctly, the internal state

machine is reset accordingly. The ADC conversion of Group_pri x will be stalled until the GPxOVRF bit in ADC_DMACTL register is cleared by software, and the data of ADC_GPxDMAR register will remap to the data of the start channel of Group_pri x.

Figure 15-50. Example of an overflow event of Group_pri2 in ADC2 provides an example of an overflow event of Group_pri2 in ADC2, with the Group_pri2 selecting channels ADC2_IN0, ADC2_IN1, ADC2_IN2, ADC2_IN4, and ADC2_IN5.

Figure 15-50. Example of an overflow event of Group_pri2 in ADC2



15.5.14 ADC internal channels

Temperature sensor channel

A/D conversion of the temperature sensor channel can be selected for ADC2 only.

The GP1TEMP bit in ADC_CHSEL0 register selects A/D conversion of the temperature sensor channel in Group_pri1 scan once mode.

The GPxTEMP bit in ADC_CHSEL0/1 register selects A/D conversion of the temperature sensor channel of the according Group_pri x (Group_pri1/pri2/pri3/pri4) in groups scan mode, and the temperature sensor channel priority can be set in ADC_CHPRI0/1.

SPTTEMP[7:0] bits in ADC_SAMPR2 register are used to set the the sampling time of temperature sensor channel. If disconnection detection assist is used (CHSEL[4:0] $\neq$ 5'b00000), sampling starts after discharging is completed during A/D conversion of the temperature sensor channel, and the discharging period CHSEL[4:0] are fixed to 5'b11111 automatically. After the conversion is finished the CHSEL[4:0] return to the setting before.

The temperature sensor can be used to measure the ambient temperature of the device. The sensor output voltage can be converted into a digital value by ADC. The sampling time for the temperature sensor is recommended to be set to at least t_{s_temp} μ s (please refer to the datasheet).

To use the temperature sensor:

1. Configure the GPxTEMP bit in ADC_CHSEL0/1 register and the sampling time (t_{s_temp} μ s or longer) for the channel by SPTTEMP[7:0] bits in ADC_SAMPR1 register.
2. Start the ADC conversion by the triggers.

3. Read the temperature data ($V_{\text{temperature}}$) in ADC_TEMPDATA register, and get the temperature with the following equation.

$$\text{Temperature } (^{\circ}\text{C}) = \frac{(V_{\text{temperature}} - V_{-40}) * (-40 - 105)}{V_{-40} - V_{105}} - 40 \quad (15-1)$$

V_{-40} : internal temperature sensor output voltage at -40°C , the typical value and factory calibration value address please refer to the datasheet (TS_CAL1).

V_{105} : internal temperature sensor output voltage at 105°C , the typical value and factory calibration value address please refer to the datasheet (TS_CAL1).

Internal reference voltage channel

A/D conversion of the internal reference voltage channel can be selected for ADC2 only.

The GP1VINT bit selects A/D conversion of the internal reference voltage channel in Group_pri1 scan once mode.

The GPxVINT bit selects A/D conversion of the internal reference voltage channel of the according group (Group_pri1/pri2/pri3/pri4) in groups scan mode, and the internal reference voltage channel priority can be set in ADC_CHPRI0/1.

SPTVINT[7:0] bits in ADC_SAMPR2 register are used to set the the sampling time of internal reference voltage channel. If disconnection detection assist is used ($\text{CHRSEL}[4:0] \neq 5'b00000$), sampling starts after discharging is completed during A/D conversion of the internal reference voltage channel, and the discharging period $\text{CHRSEL}[4:0]$ are fixed to $5'b11111$ automatically. After the conversion is finished the $\text{CHRSEL}[4:0]$ return to the setting before.

The sampling time should be $5 \mu\text{s}$ or longer.

To use the internal reference voltage channel:

1. Set the sampling time to $5 \mu\text{s}$ or longer.
2. After switching to A/D conversion of the internal reference voltage channel, the A/D conversion of the internal reference voltage channel is started.
3. When A/D conversion is completed the conversion result is stored into the A/D internal reference voltage channel data register (ADC_VINTDATA).
4. The PROC bit remains 1 during A/D conversion, and is automatically cleared to 0 upon completion of A/D conversion. Then the 12-bit A/D converter enters a wait state.

15.5.15 Programmable resolution (DRES)

The resolution is configured by programming the DRES[1:0] bits in the ADC_CTL1 register. For applications that do not require high data accuracy, lower resolution allows faster conversion time. The DRES[1:0] bits must only be changed when the ADCON bit is reset. The result of the conversion is always 12 bits wide and any unused LSB bits are read as zeroes. Lower resolution reduces the conversion time needed for the successive approximation steps as shown in [Table 15-12. tCONV timings depending on resolution](#).

Table 15-12. t_{CONV} timings depending on resolution

DRES[1:0] bits	t_{CONV} (ADC clock cycles)	$t_{CONV}(ns)$ at $f_{ADC}=36MHz$	$t_{SMPL}(min)$ (ADC clock cycles)	t_{ADC} (ADC clock cycles)	$t_{ADC}(ns)$ at $f_{ADC}=36MHz$
12	12.5	347.22 ns	2.5	15	416.67 ns
10	10.5	291.67 ns	2.5	13	361.11 ns
8	8.5	236.11 ns	2.5	11	305.56 ns
6	6.5	180.56 ns	2.5	9	250.00 ns

15.5.16 Disconnection detection assist function (precharge and discharge)

This converter incorporates the function to fix the charge for sampling capacitance to the specified state before start of A/D conversion. This function enables disconnection detection in wiring of analog inputs, temperature sensor channel and internal reference voltage channel. If disconnection detection assist is used, disconnection-detection will be operated for every conversion of single channel which does not use Sample-and-Hold circuit. The disconnection detection assist function can be used when SH00EN/SH01EN/SH02EN in ADC_SHCTL register are 0 (bypass the sample-and-hold circuits).

The CHRSEL[4:0] bits in ADC_CTL0 register selects either precharge or discharge and the period of precharge/discharge for the A/D disconnection detection assist function. Setting the CHRSEL[4] bit to 1 allows to select precharge and setting the CHRSEL[4] to 0 allows to select discharge. The period of precharge/discharge can be set with the CHRSEL[3:0] bits. When the CHRSEL[3:0] = 4'b0000 or 4'b0001, the disconnection detection assist function is not effective (disconnection detection is disabled). Except for the case of CHRSEL[3:0] = 0000b or 0001b, the specified value indicates the number of states for the period of precharge/discharge. The CHRSEL[4:0] bits should be set when the PROC bit is 0.

For the internal reference voltage and temperature sensor channel channel, an fix disconnect-detection period (CHRSEL[3:0] = 0xF) is inserted before sampling the channel.

The internal reference voltage and temperature sensor channel can be selected for ADC2 only.

Figure 15-51. Operation of A/D conversion when the disconnection detection assist function is used


15.5.17 Self-Diagnosis Function

The SDSSEN bit in ADC_CTL0 register enables or disables self-diagnosis.

Self-diagnosis is used to detect a failure of the 12-bit A/D converter. Specifically, one of the

internally generated voltage values 0, $V_{REFP}/2$, and the V_{REFP} is converted. When conversion is completed, information on the converted voltage and the conversion result is stored into the self-diagnosis data register (ADC_SDDATA). ADC_SDDATA can then be read out by software to determine whether the conversion result falls within the normal range (normal) or not (abnormal). Self-diagnosis is executed once at the beginning of each scan, and one of the three voltages is converted. When self-diagnosis is selected in groups scan mode, self-diagnosis is separately executed in Group_pri1/Group_pri2/Group_pri3/Group_pri4.

The SDSFIX bit in ADC_CTL0 register selects whether the three voltage values are rotated or the fixed voltage is used in self-diagnosis. Setting SDSFIX bit to 0 allows conversion of the voltages in rotation mode where 0, $V_{REFP}/2$, and the V_{REFP} are converted in this order. When self-diagnosis rotation mode is selected after a reset, self-diagnosis is performed from 0 V. Setting SDSFIX bit to 1 to enable self-diagnosis voltage fixed mode, and the fixed voltage specified by the SDVOG[1:0] bits is converted. In self-diagnosis voltage rotation mode, the self-diagnosis voltage value does not return to 0 when scan conversion is completed. When scan conversion is restarted, therefore, rotation starts at the voltage value following the previous value. If fixed mode is switched to rotation mode, rotation starts at the fixed voltage value.

The SDSEN/SDSFIX/ SDVOG[1:0] in ADC_CTL0 register should be set when the PROC bit is 0.

15.5.18 Sample-and-Hold Function (for ADC0 only)

The SH00EN/ SH01EN/ SH02EN in ADC_SHCTL register select whether to use or not use (bypass) ADC0_IN0 to ADC0_IN2 channel-dedicated sample-and-hold circuits.

The SHSTIME[7:0] bits in ADC_SHCTL register set the sampling time for the channel-dedicated sample-and-hold circuits. If the ADCLK clock is 32 MHz, one state is 31.25 ns (one state is one ADCLK cycle).

In Group_pri1 scan once mode and continuous scan mode, if constant sampling function is not used, the sample initial value is 12 states, the hold initial value is 8 states. If the impedance of analog input signal source is too high to secure sufficient sampling time or if the ADCLK clock is slow, the sampling time can be adjusted. The sampling time must be set to a value that is 4 states or more and is 255 or less. Also, the sampling state setting value should be at least the specified test condition.

The SHSTIME[7:0]/ SHHTIME[3:0] bit in ADC_SHCTL register should be set before the PROC bit is 1.

In constant sampling mode, the circuit working operation is shown in [Group_pri1 scan once mode](#) and [Group_pri1 scan continuous mode](#). In groups scan mode all the channels which enables sampling-and-holding can be selected as a whole for Group_pri x. The channel-dedicated sampling-and-holding circuit works independently. If the last channel with a channel-dedicated sample-and-hold circuit is selected and continuous scanning is performed, securing a period for normal sampling will not be possible in the second and subsequent

rounds of continuous scanning. Adding at least one other channels at the end of this channel is recommended in this case.

In groups scan mode, the sample initial value is 12 states, the hold initial value is 8 states. If the impedance of analog input signal source is too high to secure sufficient sampling time or if the ADCLK clock is slow, the sampling time can be adjusted. The sampling time must be set to a value that is 4 states or more and is 255 or less. Also, the sampling state setting value should be at least the specified test condition.

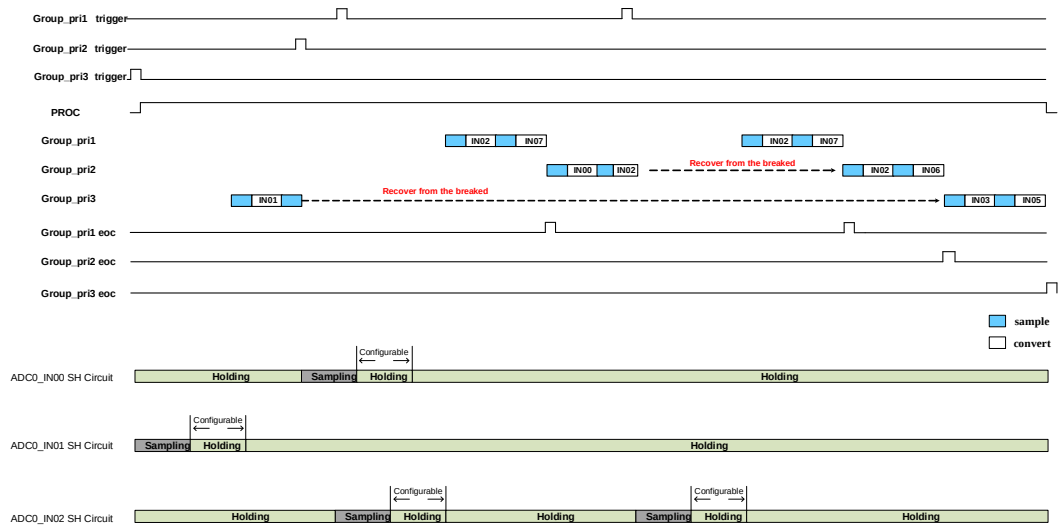
When the trigger for groups which contains the S&H enabled channels comes, the corresponding S&H enabled channels will start sample-and-hold operation immediately no matter whether any of the current higher priority group is working. When all the S&H enabled channels' sample-and-hold operation are finished (the sample-and-hold operation time is set in SHSTIME[7:0]/ SHHTIME[3:0] bit in ADC_SHCTL register), the corresponding Group_pri x starts A/D conversion (If any of the current higher priority group is working, the channels belong to this group will enter a extend holding state to wait all the higher priority groups to finish A/D conversion). The sample-and-hold operation of the S&H enabled channels cannot be breaked by higher priority group trigger even the ADC is under group priority control. The following is an example which the sample-and-hold function is used under group priority control.

Operation example: High priority group trigger input during low-priority group rescan, with rescan setting (GP2RTCH/ GP3RTCH = 1, S&H function enabled channels is selected seperatedly by Group_pri1/ Group_pri2/ Group_pri3)

S&H function enabled channels are ADC0_IN0, ADC0_IN1, ADC0_IN2. Group_pri1 selects ADC0_IN2/IN7, Group_pri2 selects ADC0_IN0/IN2/IN6, Group_pri3 selects ADC0_IN1/IN3/IN5. When the Group_pri3 trigger comes, the ADC0_IN1 SH circuit first starts sample-and-hold operation, when the sample-and-hold operation is done the Group_pri3 starts A/D conversion. When the Group_pri2 trigger comes, the Group_pri3 A/D conversion operation is breaked and ADC0_IN0 SH circuit starts sample-and-hold operation. When the Group_pri1 trigger comes during the sample-and-hold operation of ADC0_IN0 SH circuit, the ADC0_IN2 SH circuit starts sample-and-hold operation and ADC0_IN0 SH circuit continues the unfinished sample-and-hold operation. When the sample-and-hold operation of ADC0_IN0 SH circuit is finished, the ADC0_IN0 SH circuit enters a extend holding period. When ADC0_IN2 SH circuit finishes the sample-and-hold operation, the Group_pri1 starts A/D conversion.

Note: For low priority group which contains SH function enabled channels, if any higher priority groups break the current group operation, the holding and extend holding period of the current group should not exceed 20us, otherwise the SH circuit cannot retain a valid holding voltage due to electric leakage. If bifurcate trigger mode is enabled of Group_pri x, this point should be also noted if the Group_pri x contains SH function enabled channels.

Figure 15-52. High priority group trigger input during low-priority group rescan



15.5.19 On-chip hardware oversample (data accumulation/average)

When the ADDEN bit in ADC_CTL0 register is set to 1, A/D addition conversion function is on for all the channels. The ADDSEL bit in ADC_CTL0 register selects the addition conversion data mode (0 for accumulation data mode, 1 for average data mode).

In accumulation data mode, the every single channel is A/D-converted 2, 3, 4, 8, 16, 32 or 64 consecutive times according to the channel-dedicated addition value setting (the CHyADDT[3:0] bits in ADC_ADDDT0/1 registers) and the sum of the converted values is stored in the data register.

Note: When use accumulation data mode with 12bit conversion resolution, do not choose 32 or 64 consecutive times, otherwise the right conversion result is not guaranteed.

In average data mode, the every single channel is A/D converted 2, 3, 4, 8, 16, 32 or 64 consecutive times according to the channel-dedicated addition value setting and the mean of the converted values is stored in the data register. The use of the average of these results can improve the accuracy of A/D conversion, depending on the types of noise components that are present.

The A/D-converted value accumulation/average mode can be used when A/D conversion of the channel select temperature sensor channel or internal reference voltage (for ADC2 only) channel is selected. Addition times setting is according to TEMPADDT[3:0] / VINTADDT[3:0] bits in ADC_ADDDT1 register.

The A/D addition conversion is disabled for the single channel which channel-dedicated addition value is set to 4'b0000 in ADC_ADDDT0/1 registers.

15.5.20 ADC data registers automatic clearing function

Setting the ACDATA bit in ADC_CTL0 register to 1 will enable automatic clearing data register function. Setting the ASDATA bit in ADC_CTL0 register to 1 will enable automatic setting data

register function.

When ASDATA bit is set, all bits in data register are set to 1 after it has been read by CPU or DMA, including ADC_CHxDATA, ADC_SDDATA, ADC_GPxBIDATA, ADC_GPxBIDATA1, ADC_GPxBIDATA2, ADC_TEMPDATA and ADC_VINTDATA.

When ACDATA bit is set, all bits in data register are cleared to 0 after it has been read by CPU or DMA, including ADC_CHxDATA, ADC_SDDATA, ADC_GPxBIDATA, ADC_GPxBIDATA1, ADC_GPxBIDATA2, ADC_TEMPDATA and ADC_VINTDATA.

Note: Only one of ACDATA or ASDATA bit can be set to enable data automatic clear/setting function (If both set, the automatic setting/ clearing data registers function is invalid).

This function enables detection of update failures of the ADC data registers. After the data registers are read by bus, or the updated data of channels of Group_prix is transferred to the destination location which is specified by the user using DMA, the value of the data registers will be cleared to 0000h if ACDATA bit is set or FFFFh if ASDATA is set. Occurrence of ADC data registers update failure can be determined by simply checking that the read data value is 0000h or FFFFh.

15.5.21 ADC interrupts

The interrupt can be produced on one of the events:

- End of conversion round flag of Group_pri1/Group_pri2/Group_pri3/Group_pri4.
- The analog watchdog event (analog watchdog A channel x compare status or analog watchdog B compare flag).
- DMA overflow event of Group_pri1/Group_pri2/Group_pri3/Group_pri4.

The interrupt of ADC0 is mapped into interrupt vector IRQ[18].

The interrupt of ADC2 is mapped into interrupt vector IRQ[75].

15.6 Register definition

ADC0 base address: 0x4001 2000

ADC2 base address: 0x4001 2400

15.6.1 Software EOC control register (ADC_EOCCTL)

Address offset: 0x00

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EOC4RCNT[2:0]			EOC4RF	EOC4F	EOC4RIE	GP4SWST	EVICG4SEL	EOC3RCNT[2:0]			EOC3RF	EOC3F	EOC3RIE	GP3SWST	EVICG3SEL
rw			rw	rw	rw	rw	rw	rw			rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EOC2RCNT[2:0]			EOC2RF	EOC2F	EOC2RIE	GP2SWST	EVICG2SEL	EOC1RCNT[2:0]			EOC1RF	EOC1F	EOC1RIE	GP1SWST	EVICG1SEL
rw			rw	rw	rw	rw	rw	rw			rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:29	EOC4RCNT[2:0]	End of Group_pri4 conversion round counts Every the round number (EOC4RCNT[2:0] + 1) of end of Group_pri4 conversion comes, the EOC4RF will be set.
28	EOC4RF	End of Group_pri4 conversion round flag Set by hardware when every the round number (EOC4RCNT[2:0] + 1) of end of Group_pri4 conversion comes, this flag will be set to 1. Cleared by software writing 0 to it or by reading the ADC_GP4DMAR register of Group_pri4.
27	EOC4F	End of Group_pri4 conversion flag 0: No end of Group_pri4 conversion 1: End of Group_pri4 conversion Set by hardware at the end the last channel of Group_pri4 conversion. Cleared by software writing 0 to it or by reading the data register of Group_pri4.
26	EOC4RIE	Interrupt enable for EOC4RF 0: EOC4RF interrupt disable 1: EOC4RF interrupt enable
25	GP4SWST	Software start conversion of Group_pri4 Set 1 on this bit starts conversion of Group_pri4. 0: Group_pri4 not started by software 1: Group_pri4 started by software It is set by software and cleared by software or by hardware immediately after

		channel sample starts.
24	EVICG4SEL	EVIC trigger signal select from Group_pri4 0: EVIC trigger signal selects EOC4RF 1: EVIC trigger signal selects EOC4F
23:21	EOC3RCNT[2:0]	End of Group_pri3 conversion round counts Every the round number (EOC3RCNT[2:0] + 1) of end of Group_pri3 conversion comes, the EOC3RF will be set.
20	EOC3RF	End of Group_pri3 conversion round flag Set by hardware when every the round number (EOC3RCNT[2:0] + 1) of end of Group_pri3 conversion comes, this flag will be set to 1. Cleared by software writing 0 to it or by reading the ADC_GP3DMAR register of Group_pri3.
19	EOC3F	End of Group_pri3 conversion flag 0: No end of Group_pri3 conversion 1: End of Group_pri3 conversion Set by hardware at the end the last channel of Group_pri3 conversion. Cleared by software writing 0 to it or by reading the data register of Group_pri3.
18	EOC3RIE	Interrupt enable for EOC3RF 0: EOC3RF interrupt disable 1: EOC3RF interrupt enable
17	GP3SWST	Software start conversion of Group_pri3 Set 1 on this bit starts conversion of Group_pri3. 0: Group_pri3 not started by software 1: Group_pri3 started by software It is set by software and cleared by software or by hardware immediately after channel sample starts.
16	EVICG3SEL	EVIC trigger signal select from Group_pri3 0: EVIC trigger signal is EOC3RF 1: EVIC trigger signal is EOC3F
15:13	EOC2RCNT[2:0]	End of Group_pri2 conversion round counts Every the round number (EOC2RCNT[2:0] + 1) of end of Group_pri2 conversion comes, the EOC2RF will be set.
12	EOC2RF	End of Group_pri2 conversion round flag Set by hardware when every the round number (EOC2RCNT[2:0] + 1) of end of Group_pri2 conversion comes, this flag will be set to 1. Cleared by software writing 0 to it or by reading the ADC_GP2DMAR register of Group_pri2.
11	EOC2F	End of Group_pri2 conversion flag

		0: No end of Group_pri2 conversion 1: End of Group_pri2 conversion Set by hardware at the end the last channel of Group_pri2 conversion. Cleared by software writing 0 to it or by reading the data register of Group_pri2.
10	EOC2RIE	Interrupt enable for EOC2RF 0: EOC2RF interrupt disable 1: EOC2RF interrupt enable
9	GP2SWST	Software start conversion of Group_pri2 Set 1 on this bit starts conversion of Group_pri2. 0: Group_pri2 not started by software 1: Group_pri2 started by software It is set by software and cleared by software or by hardware immediately after channel sample starts.
8	EVICG2SEL	EVIC trigger signal select from Group_pri2 0: EVIC trigger signal is EOC2RF 1: EVIC trigger signal is EOC2F
7:5	EOC1RCNT[2:0]	End of Group_pri1 conversion round counts Every the round number (EOC1RCNT[2:0] + 1) of end of Group_pri1 conversion comes, the EOC1RF will be set.
4	EOC1RF	End of Group_pri1 conversion round flag Set by hardware when every the round number (EOC1RCNT[2:0] + 1) of end of Group_pri1 conversion comes, this flag will be set to 1. Cleared by software writing 0 to it or by reading the ADC_GP4DMAR register of Group_pri1.
3	EOC1F	End of Group_pri1 conversion flag 0: No end of Group_pri1 conversion 1: End of Group_pri1 conversion Set by hardware at the end the last channel of Group_pri1 conversion. Cleared by software writing 0 to it or by reading the data register of Group_pri1.
2	EOC1RIE	Interrupt enable for EOC1RF 0: EOC1RF interrupt disable 1: EOC1RF interrupt enable
1	GP1SWST	Software start conversion of Group_pri1 Set 1 on this bit starts conversion of Group_pri1. 0: Group_pri1 not started by software 1: Group_pri1 started by software It is set by software and cleared by software or by hardware immediately after channel sample starts.
0	EVICG1SEL	EVIC trigger signal select from Group_pri1

0: EVIC trigger signal is EOC1RF

1: EVIC trigger signal is EOC1F

15.6.2 Control register 0 (ADC_CTL0)

Address offset: 0x04

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SWEND	EVICCTL[2:0]			Reserved			GP4EN	GP3EN	ADDSEL	GSCAN[1:0]		PROCF	Reserved		ADDEN
rw	rw						rw	rw	rw	rw		r			rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ASDATA	ACDATA	SDVOG[1:0]		SDSFIX	SDSEN	DAL	CHRSEL[4:0]					GPRIEN	Reserved		LPGCNT
rw	rw	rw		rw	rw	rw	rw					rw			rw

Bits	Fields	Descriptions
31	SWEND	Software end conversion of Group_pri1/pri2/pri3/pri4 Set by software to end conversion of all groups. Cleared by hardware.
30:28	EVICCTL[2:0]	EVIC link signal control 000: An event signal is generated when scanning for Group_pri1 is completed. 001: An event signal is generated when scanning for Group_pri2 is completed. 010: An event signal is generated when scanning for Group_pri3 is completed. 011: An event signal is generated when scanning for Group_pri4 is completed. 100: An event signal is generated when scanning for Group_pri1, Group_pri2, Group_pri3 or Group_pri4 is completed. Other values: Reserved. Settings other than above are prohibited.
27:25	Reserved	Must be kept at reset value.
24	GP4EN	Enable A/D conversion operation for Group_pri4. 0: Group_pri4 is not used 1: Group_pri4 is used
23	GP3EN	Enable A/D conversion operation for Group_pri3. 0: Group_pri3 is not used 1: Group_pri3 is used
22	ADDSEL	A/D conversion data addition mode selection 0: Accumulation data mode 1: Average data mode
21:20	GSCAN[1:0]	Group scan mode The bit can be configured while the PROCF bit is 0.

		00: Group_pri1 scan once mode 01: Groups scan mode 10: Group_pri1 scan continuous mode 11: Reserved Note: When GSCAN[1:0] bits are 2'b01, if the application needs to use two groups, only Group_pri1 and Group_pri2 are used. if the application needs to use three groups, Group_pri1&pri2&pri3 or Group_pri1&pri2&pri4 can be used. if the application needs to use one group, only Group_pri1 is used.
19	PROCF	A/D conversion process flag This bit is read only by software and indicates whether the A/D conversion is in process. 0: A/D conversion is not in process. 1: A/D conversion is in process.
18:17	Reserved	Must be kept at reset value.
16	ADDEN	Addition function enable 0: A/D addition conversion function is disabled. 1: A/D addition conversion function is enabled.
15	ASDATA	Enable automatic setting data registers When this bit is set, all bits in data register are set to 1 after it has been read by CPU or DMA, including ADC_CHxDATA, ADC_SDDATA, ADC_GPxBIDATA, ADC_GPxBIDATA1, ADC_GPxBIDATA2, ADC_TEMPDATA and ADC_VINTDATA. 0: Disable automatic setting data registers 1: Enable automatic setting data registers
14	ACDATA	Enable automatic clearing data registers When this bit is set, all bits in data register are cleared to 0 after it has been read by CPU or DMA, including ADC_CHxDATA, ADC_SDDATA, ADC_GPxBIDATA, ADC_GPxBIDATA1, ADC_GPxBIDATA2, ADC_TEMPDATA and ADC_VINTDATA. 0: Disable automatic clearing data registers 1: Enable automatic clearing data registers
13:12	SDVOG[1:0]	Configure fix voltage for self-diagnosis conversion 00: Reserved 01: 0V 10: $V_{REFP}/2$ 11: V_{REFP}
11	SDSFIX	Enable fix mode for self-diagnosis voltage When this bit is 1, a fix voltage (configure by SDVOG[1:0]) is converted for self-diagnosis. When this bit is 0, if SDSEN is set, self-diagnosis takes turns converting 0, V_{REFP}

		/2, V_{REFP}. The conversion result of self-diagnosis is stored into ADC_SDDATA. The bit can be configured while the PROCF bit is 0. 0: Disable fix voltage mode for self-diagnosis. 1: Enable fix voltage mode for self-diagnosis
10	SDSEN	Enable self-diagnosis When this bit is set, the voltage (loop or fixed: 0, $V_{REFP}/2$ or V_{REFP}) is first converted before start a scan. The bit can be configured while the PROCF bit is 0. 0: Disable self-diagnosis 1: Enable self-diagnosis
9	DAL	Data storage alignment mode 0: LSB alignment 1: MSB alignment
8:4	CHRSEL[4:0]	Select precharge and discharge CHRSEL[4] bit selects either precharge or discharge and CHRSEL[3:0] bits configure the period of precharge/discharge. 00000~00001: Disable precharge and discharge 00000~01111: select discharge, and the period is CHRSEL[3:0] CLK_ADC. 10000~10001: Disable precharge and discharge 10000~11111: select precharge, and the period is CHRSEL[3:0] CLK_ADC
3	GPRIEN	Enable group priority When the bit is set, high priority group can abort the scan of low priority group. The priority order of group is Group_pri1 > Group_pri2 > Group_pri3 > Group_pri4. 0: Disable group priority 1: Enable group priority Note: When the GPRIEN bit is set, the GSCAN[1:0] bits must be set to 2'b01 (groups scan mode).
2:1	Reserved	Must be kept at reset value.
0	LPGCNT	Enable lowest-priority group scan continuous This bit is valid only when GPRIEN is set. 0: The lowest-priority group scan once 1: The lowest-priority group scan continuous Note: For two groups (SCAN[1:0]=2'b01, GP3EN=0, GP4EN=0), the Group_pri1 is the lowest-priority group. For three groups (SCAN[1:0]=2'b01, GP3EN=1, GP4EN=0), the Group_pri3 is the lowest-priority group. For three groups (SCAN[1:0]=2'b01, GP3EN=0, GP4EN=1), the Group_pri4 is the lowest-priority group.

15.6.3 Group config register 0 (ADC_GPCGF0)

Address offset: 0x08

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved	GP2RTEN	GP2RTCH	Reserved	GP2TRGED[1:0]	GP2TRGSRC[5:0]	GP2TRGTY	G2TRGEN								
rw		rw	rw		rw		rw		rw		rw		rw		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	GP1TRGED[1:0]	GP1TRGSRC[5:0]	GP1TRGTY	G1TRGEN											
rw		rw		rw		rw		rw		rw		rw		rw	

Bits	Fields	Descriptions
31:30	Reserved	Must be kept at reset value.
29	GP2RTEN	Enable Group_pri2 rescan This bit is valid only when GPRIEN is set. When this bit is 0, If the lower priority group is aborted by the higher priority group, it not rescanned after the higher priority group scan complete. A trigger for lower priority group generated during higher priority scan is ignored. When this bit is 1, If the lower priority group is aborted by the higher priority group, it rescans after all higher priority group scan complete. A trigger for lower priority group generated during higher priority scan is responded after the higher priority scan complete, and the lower priority group scan is started. 0: Disable group rescan 1: Enable group rescan
28	GP2RTCH	Group_pri2 rescan channel select This bit is valid only when GPRIEN=1, GP2RESTEN=1. The bit can be configured while the PROCF bit is 0. 0: Rescan the lower priority group scanning from initial channel. 1: Rescan the lower priority group scanning from the last aborted channel.
27:26	Reserved	Must be kept at reset value.
25:24	GP2TRGED[1:0]	Trigger edge select of Group_pri2 00: Reserved 01: Rising edge of trigger 01: Falling edge of trigger 11: Rising and falling edge of trigger
23:18	GP2TRGSRC[5:0]	Synchronous trigger source for Group_pri2 000000: Reserved 000001~101000: Refer to Table 15-10. ADC synchronous trigger source. 101001~111111: Reserved

17	GP2TRGTY	Group_pri2 trigger type select 0: Group_pri2 trigger is synchronous 1: Group_pri2 trigger is asynchronous Note: Only one group can enable asynchronous trigger in group_pri1/pri2/pri3/pri4. Asynchronous trigger for two or more groups cannot be enabled simultaneously.
16	G2TRGEN	Enable Group_pri2 synchronous or asynchronous trigger 0: Disable Group_pri2 synchronous or asynchronous trigger 1: Enable Group_pri2 synchronous or asynchronous trigger Note: When this bit is set to 1, software triggering (GP2SWST) is invalid.
15:10	Reserved	Must be kept at reset value.
9:8	GP1TRGED[1:0]	Trigger edge select of Group_pri1 00: Reserved 01: Rising edge of trigger 01: Falling edge of trigger 11: Rising and falling edge of trigger
7:2	GP1TRGSRC[5:0]	Synchronous trigger source for Group_pri1 000000: Reserved 000001~101000: Refer to Table 15-10. ADC synchronous trigger source . 101001~111111: Reserved
1	GP1TRGTY	Group_pri1 trigger type select 0: Group_pri1 trigger is synchronous 1: Group_pri1 trigger is asynchronous Note: Only one group can enable asynchronous trigger in Group_pri1/pri2/pri3/pri4. Asynchronous trigger for two or more groups cannot be enabled simultaneously.
0	G1TRGEN	Enable Group_pri1 synchronous or asynchronous trigger 0: Disable Group_pri1 synchronous or asynchronous trigger 1: Enable Group_pri1 synchronous or asynchronous trigger Note: When this bit is set to 1, software triggering (GP1SWST) is invalid.

15.6.4 Group config register 1 (ADC_GPCGF1)

Address offset: 0x0C

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved	GP4RTEN	GP4RTCH	Reserved	GP4TRGED[1:0]	GP4TRGSRC[5:0]	GP4TRGTY	G4TRGEN								
	rw	rw		rw	rw	rw	rw								
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Reserved	GP3RTEN	GP3RTCH	Reserved	GP3TRGED[1:0]	GP3TRGSRC[5:0]	GP3TRGTY	G3TRGEN
	rw	rw		rw	rw	rw	rw

Bits	Fields	Descriptions
31:30	Reserved	Must be kept at reset value.
29	GP4RTEN	Enable Group_pri4 rescan This bit is valid only when GPRIEN is set. When this bit is 0, If the lower priority group is aborted by the higher priority group, it not rescanned after the higher priority group scan complete. A trigger for lower priority group generated during higher priority scan is ignored. When this bit is 1, If the lower priority group is aborted by the higher priority group, it rescans after all higher priority group scan complete. A trigger for lower priority group generated during higher priority scan is responded after the higher priority scan complete, and the lower priority group scan is started. 0: Disable group rescan 1: Enable group rescan
28	GP4RTCH	Group_pri4 rescan channel select This bit is valid only when GPRIEN=1, GP4RESTEN=1. The bit can be configured while the PROCF bit is 0. 0: Rescan the lower priority group scanning from initial channel. 1: Rescan the lower priority group scanning from the last aborted channel.
27:26	Reserved	Must be kept at reset value.
25:24	GP4TRGED[1:0]	Trigger edge select of Group_pri4 00: Reserved 01: Rising edge of trigger 01: Falling edge of trigger 11: Rising and falling edge of trigger
23:18	GP4TRGSRC[5:0]	Synchronous trigger source for Group_pri4 000000: Reserved 000001~101000: Refer to Table 15-10. ADC synchronous trigger source. 101001~111111: Reserved
17	GP4TRGTY	Group_pri4 trigger type select 0: Group_pri4 trigger is synchronous 1: Group_pri4 trigger is asynchronous Note: Only one group can enable asynchronous trigger in group_pri1/pri2/pri3/pri4. Asynchronous trigger for two or more groups cannot be enabled simultaneously.
16	G4TRGEN	Enable Group_pri4 synchronous or asynchronous trigger 0: Disable Group_pri4 synchronous or asynchronous trigger 1: Enable Group_pri4 synchronous or asynchronous trigger

Note: When this bit is set to 1, software triggering (GP4SWST) is invalid.		
15:14	Reserved	Must be kept at reset value.
13	GP3RTEN	Enable Group_pri3 rescan This bit is valid only when GPRIEN is set. When this bit is 0, If the lower priority group is aborted by the higher priority group, it not rescanned after the higher priority group scan complete. A trigger for lower priority group generated during higher priority scan is ignored. When this bit is 1, If the lower priority group is aborted by the higher priority group, it rescans after all higher priority group scan complete. A trigger for lower priority group generated during higher priority scan is responded after the higher priority scan complete, and the lower priority group scan is started. 0: Disable group rescan 1: Enable group rescan
12	GP3RTCH	Group_pri3 rescan channel select This bit is valid only when GPRIEN=1, GP3RESTEN=1. The bit can be configured while the PROCF bit is 0. 0: Rescan the lower priority group scanning from initial channel. 1: Rescan the lower priority group scanning from the last aborted channel.
11:10	Reserved	Must be kept at reset value.
9:8	GP3TRGED[1:0]	Trigger edge select of Group_pri3 00: Reserved 01: Rising edge of trigger 01: Falling edge of trigger 11: Rising and falling edge of trigger
7:2	GP3TRGSRC[5:0]	Synchronous trigger source for Group_pri3 000000: Reserved 000001~101000: Refer to Table 15-10. ADC synchronous trigger source. 101001~111111: Reserved
1	GP3TRGTY	Group_pri3 trigger type select 0: Group_pri3 trigger is synchronous 1: Group_pri3 trigger is asynchronous Note: Only one group can enable asynchronous trigger in group_pri1/pri2/pri3/pri4. Asynchronous trigger for two or more groups cannot be enabled simultaneously.
0	G3TRGEN	Enable Group_pri3 synchronous or asynchronous trigger 0: Disable Group_pri3 synchronous or asynchronous trigger 1: Enable Group_pri3 synchronous or asynchronous trigger Note: When this bit is set to 1, software triggering (GP3SWST) is invalid.

15.6.5 Channel selection register 0 (ADC_CHSEL0)

Address offset: 0x10

Reset value: 0x0000 0000

This register has to be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				GP2VINT	GP2TEMP	GP2CH9	GP2CH8	GP2CH7	GP2CH6	GP2CH5	GP2CH4	GP2CH3	GP2CH2	GP2CH1	GP2CH0
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				GP1VINT	GP1TEMP	GP1CH9	GP1CH8	GP1CH7	GP1CH6	GP1CH5	GP1CH4	GP1CH3	GP1CH2	GP1CH1	GP1CH0
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:28	Reserved	Must be kept at reset value.
27	GP2VINT	Select internal reference voltage channel in Group_pri2 Refer to GP2CH00. Note: This bit is only used in ADC2.
26	GP2TEMP	Select temperature sensor channel in Group_pri2 Refer to GP2CH00. Note: This bit is only used in ADC2.
25	GP2CH09	Select channel ADCx_IN9 in Group_pri2 Refer to GP2CH00. Note: This bit is only used in ADC2.
24	GP2CH08	Select channel ADCx_IN8 in Group_pri2 Refer to GP2CH00. Note: This bit is only used in ADC2.
23	GP2CH07	Select channel ADCx_IN7 in Group_pri2 Refer to GP2CH00. Note: This bit is only used in ADC2.
22	GP2CH06	Select channel ADCx_IN6 in Group_pri2 Refer to GP2CH00.
21	GP2CH05	Select channel ADCx_IN5 in Group_pri2 Refer to GP2CH00.
20	GP2CH04	Select channel ADCx_IN4 in Group_pri2 Refer to GP2CH00.
19	GP2CH03	Select channel ADCx_IN3 in Group_pri2 Refer to GP2CH00.
18	GP2CH02	Select channel ADCx_IN2 in Group_pri2

		Refer to GP2CH00.
17	GP2CH01	Select channel ADCx_IN1 in Group_pri2 Refer to GP2CH00.
16	GP2CH00	Select channel ADCx_IN0 in Group_pri2 0: Deselect the channel 1: Select the channel Note: 1) When bifurcate trigger mode is enabled,, this bit is invalid. 2) A channel can only be selected by one group.
15:12	Reserved	Must be kept at reset value.
11	GP1VINT	Select internal reference voltage channel in Group_pri1 Refer to GP1CH00. Note: This bit is only used in ADC2.
10	GP1TEMP	Select temperature sensor channel in Group_pri1 Refer to GP1CH00. Note: This bit is only used in ADC2.
9	GP1CH09	Select channel ADCx_IN9 in Group_pri1 Refer to GP1CH00. Note: This bit is only used in ADC2.
8	GP1CH08	Select channel ADCx_IN8 in Group_pri1 Refer to GP1CH00. Note: This bit is only used in ADC2.
7	GP1CH07	Select channel ADCx_IN7 in Group_pri1 Refer to GP1CH00. Note: This bit is only used in ADC2.
6	GP1CH06	Select channel ADCx_IN6 in Group_pri1 Refer to GP1CH00.
5	GP1CH05	Select channel ADCx_IN5 in Group_pri1 Refer to GP1CH00.
4	GP1CH04	Select channel ADCx_IN4 in Group_pri1 Refer to GP1CH00.
3	GP1CH03	Select channel ADCx_IN3 in Group_pri1 Refer to GP1CH00.
2	GP1CH02	Select channel ADCx_IN2 in Group_pri1 Refer to GP1CH00.
1	GP1CH01	Select channel ADCx_IN1 in Group_pri1

Refer to GP1CH00.

0 GP1CH00 Select channel ADCx_IN0 in Group_pri1

0: Deselect the channel

1: Select the channel

Note:

- 1) When When bifurcate trigger mode is enabled, this bit is invalid.
- 2) A channel can only be selected by one group.

15.6.6 Channel selection register 1 (ADC_CHSEL1)

Address offset: 0x14

Reset value: 0x0000 0000

This register has to be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				GP4VINT	GP4TEMP	GP4CH9	GP4CH8	GP4CH7	GP4CH6	GP4CH5	GP4CH4	GP4CH3	GP4CH2	GP4CH1	GP4CH0
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				GP3VINT	GP3TEMP	GP3CH9	GP3CH8	GP3CH7	GP3CH6	GP3CH5	GP3CH4	GP3CH3	GP3CH2	GP3CH1	GP3CH0
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:28	Reserved	Must be kept at reset value.
27	GP4VINT	Select internal reference voltage channel in Group_pri4 Refer to GP4CH00. Note: This bit is only used in ADC2.
26	GP4TEMP	Select temperature sensor channel in Group_pri4 Refer to GP4CH00. Note: This bit is only used in ADC2.
25	GP4CH09	Select channel ADCx_IN9 in Group_pri4 Refer to GP4CH00. Note: This bit is only used in ADC2.
24	GP4CH08	Select channel ADCx_IN8 in Group_pri4 Refer to GP4CH00. Note: This bit is only used in ADC2.
23	GP4CH07	Select channel ADCx_IN7 in Group_pri4 Refer to GP4CH00. Note: This bit is only used in ADC2.
22	GP4CH06	Select channel ADCx_IN6 in Group_pri4 Refer to GP4CH00.

21	GP4CH05	Select channel ADCx_IN5 in Group_pri4 Refer to GP4CH00.
20	GP4CH04	Select channel ADCx_IN4 in Group_pri4 Refer to GP4CH00.
19	GP4CH03	Select channel ADCx_IN3 in Group_pri4 Refer to GP4CH00.
18	GP4CH02	Select channel ADCx_IN2 in Group_pri4 Refer to GP4CH00.
17	GP4CH01	Select channel ADCx_IN1 in Group_pri4 Refer to GP4CH00.
16	GP4CH00	Select channel ADCx_IN0 in Group_pri4 0: Deselect the channel 1: Select the channel Note: 1) When bifurcate trigger mode is enabled, this bit is invalid. 2) A channel can only be selected by one group.
15:12	Reserved	Must be kept at reset value.
11	GP3VINT	Select internal reference voltage channel in Group_pri3 Refer to GP3CH00. Note: This bit is only used in ADC2.
10	GP3TEMP	Select temperature sensor channel in Group_pri3 Refer to GP3CH00. Note: This bit is only used in ADC2.
9	GP3CH09	Select channel ADCx_IN9 in Group_pri3 Refer to GP3CH00. Note: This bit is only used in ADC2.
8	GP3CH08	Select channel ADCx_IN8 in Group_pri3 Refer to GP3CH00. Note: This bit is only used in ADC2.
7	GP3CH07	Select channel ADCx_IN7 in Group_pri3 Refer to GP3CH00. Note: This bit is only used in ADC2.
6	GP3CH06	Select channel ADCx_IN6 in Group_pri3 Refer to GP3CH00.
5	GP3CH05	Select channel ADCx_IN5 in Group_pri3 Refer to GP3CH00.

4	GP3CH04	Select channel ADCx_IN4 in Group_pri3 Refer to GP3CH00.
3	GP3CH03	Select channel ADCx_IN3 in Group_pri3 Refer to GP3CH00.
2	GP3CH02	Select channel ADCx_IN2 in Group_pri3 Refer to GP3CH00.
1	GP3CH01	Select channel ADCx_IN1 in Group_pri3 Refer to GP3CH00.
0	GP3CH00	Select channel ADCx_IN0 in Group_pri3 0: Deselect the channel 1: Select the channel Note: 1) When bifurcate trigger mode is enabled, this bit is invalid. 2) A channel can only be selected by one group.

15.6.7 Sample time register 0 (ADC_SAMPR0)

Address offset: 0x20

Reset value: 0x0202 0202

This register has to be accessed by word(32-bit).



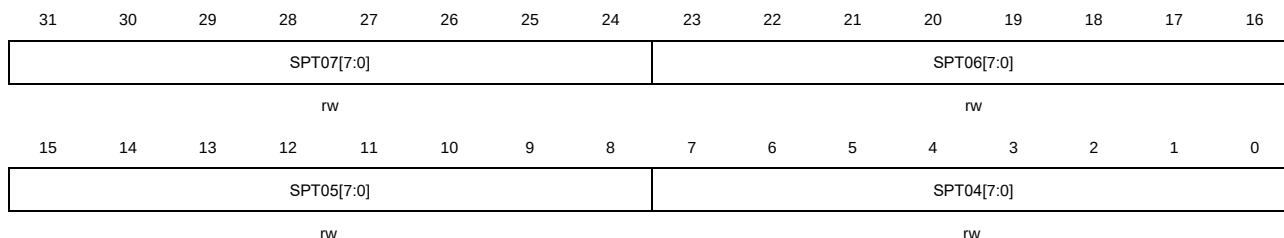
Bits	Fields	Descriptions
31:24	SPT03[7:0]	Channel ADCx_IN3 sampling time Refer to SPT00[7:0] description.
23:16	SPT02[7:0]	Channel ADCx_IN2 sampling time Refer to SPT00[7:0] description.
15:8	SPT01[7:0]	Channel ADCx_IN1 sampling time Refer to SPT00[7:0] description.
7:0	SPT00[7:0]	Channel ADCx_IN0 sampling time The sample time for channel ADCx_IN0 will be (SPT00[7:0] + 0.5) cycles. Note: the minimum value is 0x02.

15.6.8 Sample time register 1 (ADC_SAMPR1)

Address offset: 0x24

Reset value: 0x0202 0202

This register has to be accessed by word(32-bit).



Bits	Fields	Descriptions
31:24	SPT07[7:0]	Channel ADCx_IN7 sampling time Refer to SPT04[7:0] description. Note: These bits are only used in ADC2.
23:16	SPT06[7:0]	Channel ADCx_IN6 sampling time Refer to SPT04[7:0] description.
15:8	SPT05[7:0]	Channel ADCx_IN5 sampling time Refer to SPT04[7:0] description.
7:0	SPT04[7:0]	Channel ADCx_IN4 sampling time The sample time for channel ADCx_IN4 will be (SPT04[7:0] + 0.5) cycles. Note: the minimum value is 0x02.

15.6.9 Sample time register 2 (ADC_SAMPR2)

Address offset: 0x28

Reset value: 0x0202 0202

Note: This register is only used in ADC2.

This register has to be accessed by word(32-bit).



Bits	Fields	Descriptions
31:24	SPTVINT[7:0]	Internal reference voltage channel sampling time

Refer to SPT08[7:0] description.

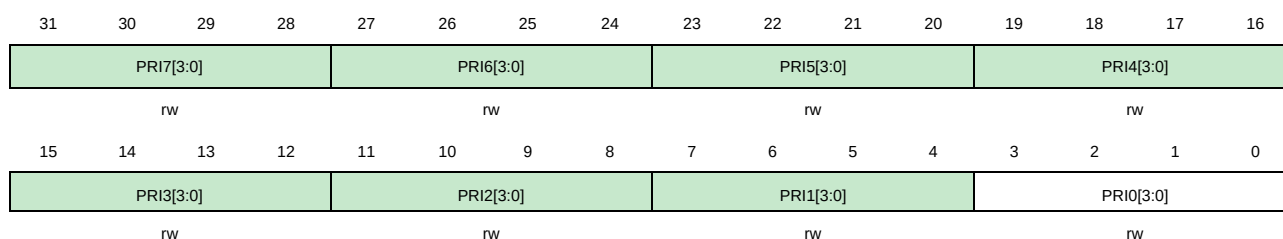
23:16	SPTTEMP[7:0]	Temperature sensor channel sampling time Refer to SPT08[7:0] description.
15:8	SPT09[7:0]	Channel ADCx_IN9 sampling time Refer to SPT08[7:0] description.
7:0	SPT08[7:0]	Channel ADCx_IN8 sampling time The sample time for channel ADCx_IN8 will be (SPT08[7:0] + 0.5) cycles. Note: the minimum value is 0x02.

15.6.10 Channel priority register 0 (ADC_CHPRI0)

Address offset: 0x30

Reset value: 0x7654 3210

This register has to be accessed by word(32-bit).



Bits	Fields	Descriptions
31:28	PRI7[3:0]	The 7th priority of the selected channel Refer to PRI0[3:0]. Note: This bit is only used in ADC2.
27:24	PRI6[3:0]	The 6th priority of the selected channel Refer to PRI0[3:0].
23:20	PRI5[3:0]	The 5th priority of the selected channel Refer to PRI0[3:0].
19:16	PRI4[3:0]	The 4th priority of the selected channel Refer to PRI0[3:0].
15:12	PRI3[3:0]	The 3th priority of the selected channel Refer to PRI0[3:0].
11:8	PRI2[3:0]	The 2nd priority of the selected channel Refer to PRI0[3:0].
7:4	PRI1[3:0]	The 1st priority of the selected channel Refer to PRI0[3:0].
3:0	PRI0[3:0]	The zeroth priority of the selected channel

Set the channel number with a priority of 0.

0000: ADCx_IN0

0001: ADCx_IN1

0010: ADCx_IN2

0011: ADCx_IN3

0100: ADCx_IN4

0101: ADCx_IN5

0110: ADCx_IN6

0111: ADCx_IN7, only used in ADC2.

1000: ADCx_IN8, only used in ADC2.

1001: ADCx_IN9, only used in ADC2.

1101: Temperature sensor channel, only used in ADC2.

1110: Internal reference voltage channel, only used in ADC2.

For example, PRI0[3:0]=4'b0010 indicates that the priority of channel ADCx_IN2 is 0.

Note: Only one priority can be configured for the same channel.

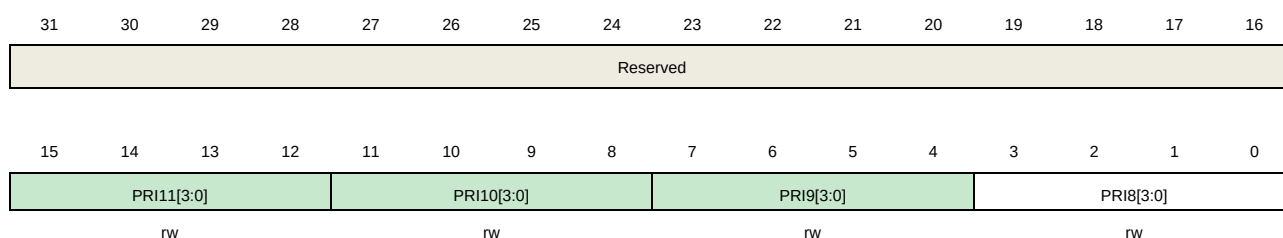
15.6.11 Channel priority register 1 (ADC_CHPRI1)

Address offset: 0x34

Reset value: 0x0000 BA98

Note: This register is only used in ADC2.

This register has to be accessed by word(32-bit).



Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:12	PRI11[3:0]	The 11th priority of the selected channel Refer to PRI8[3:0].
11:8	PRI10[3:0]	The 10th priority of the selected channel Refer to PRI8[3:0].
7:4	PRI9[3:0]	The 9th priority of the selected channel Refer to PRI8[3:0].
3:0	PRI8[3:0]	The 8th priority of the selected channel Set the channel number with a priority of 8.

0000: ADCx_IN0
0001: ADCx_IN1
0010: ADCx_IN2
0011: ADCx_IN3
0100: ADCx_IN4
0101: ADCx_IN5
0110: ADCx_IN6
0111: ADCx_IN7, only used in ADC2.
1000: ADCx_IN8, only used in ADC2.
1001: ADCx_IN9, only used in ADC2.
1101: Temperature sensor channel, only used in ADC2.
1110: Internal reference voltage channel, only used in ADC2.
For example, PRI8[3:0]=4'b0010 indicates that the priority of channel ADCx_IN2 is 8.

Note: Only one priority can be configured for the same channel.

15.6.12 ADC channel addition times register 0 (ADC_ADDT0)

Address offset: 0x40

Reset value: 0x0000 0000

This register has to be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CH7ADDT[3:0]				CH6ADDT[3:0]				CH5ADDT[3:0]				CH4ADDT[3:0]			
rw				rw				rw				rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH3ADDT[3:0]				CH2ADDT[3:0]				CH1ADDT[3:0]				CH0ADDT[3:0]			
rw				rw				rw				rw			

Bits	Fields	Descriptions
31:28	CH7ADDT[3:0]	The total addition conversion times of ADCx_IN7 Refer to CH0ADDT[3:0]. Note: This bit is only used in ADC2.
27:24	CH6ADDT[3:0]	The total addition conversion times of ADCx_IN6 Refer to CH0ADDT[3:0].
23:20	CH5ADDT[3:0]	The total addition conversion times of ADCx_IN5 Refer to CH0ADDT[3:0].
19:16	CH4ADDT[3:0]	The total addition conversion times of ADCx_IN4 Refer to CH0ADDT[3:0].
15:12	CH3ADDT[3:0]	The total addition conversion times of ADCx_IN3 Refer to CH0ADDT[3:0].
11:8	CH2ADDT[3:0]	The total addition conversion times of ADCx_IN2

		Refer to CH0ADDT[3:0].
7:4	CH1ADDT[3:0]	The total addition conversion times of ADCx_IN1 Refer to CH0ADDT[3:0].
3:0	CH0ADDT[3:0]	The total addition conversion times of ADCx_IN0 4'b0000: 1-time conversion (Disable addition function; the same as normal conversion) 4'b0001: 2-time conversion (addition once) 4'b0010: 3-time conversion (addition twice) 4'b0011: 4-time conversion (addition three times) 4'b0100: 8-time conversion (addition 7 times) 4'b0101: 16-time conversion (addition 15 times) 4'b0110: 32-time conversion (addition 31 times) 4'b0111: 64-time conversion (addition 63 times) Others: Reserved.

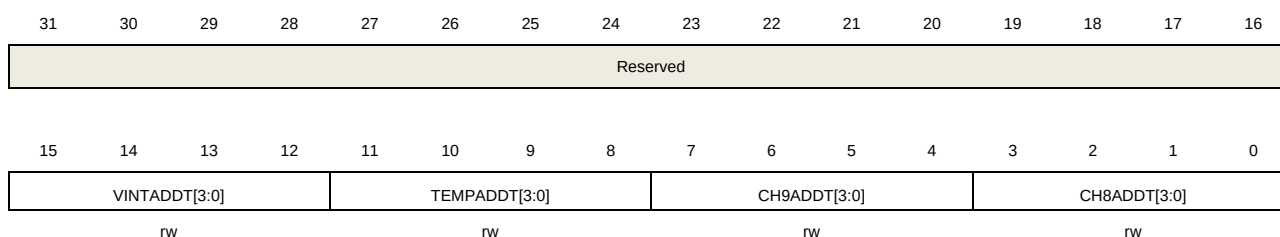
15.6.13 ADC channel addition times register 1 (ADC_ADDT1)

Address offset: 0x44

Reset value: 0x0000 0000

This register is only used in ADC2.

This register has to be accessed by word(32-bit).



Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:12	VINTADDT[3:0]	The total addition conversion times of internal reference voltage channel. Refer to CH8ADDT[3:0].
11:8	TEMPADDT[3:0]	The total addition conversion times of temperature sensor channel Refer to CH8ADDT[3:0].
7:4	CH9ADDT[3:0]	The total addition conversion times of ADCx_IN9 Refer to CH8ADDT[3:0].
3:0	CH8ADDT[3:0]	The total addition conversion times of ADCx_IN8 4'b0000: 1-time conversion (Disable addition function; the same as normal conversion)

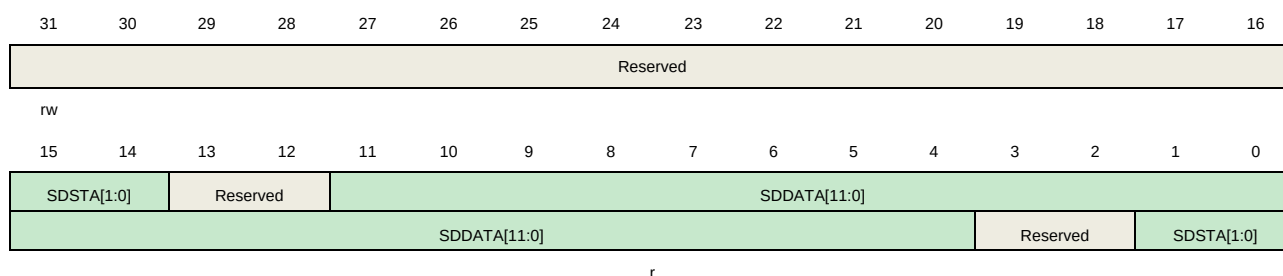
4'b0001: 2-time conversion (addition once)
4'b0010: 3-time conversion (addition twice)
4'b0011: 4-time conversion (addition three times)
4'b0100: 8-time conversion (addition 7 times)
4'b0101: 16-time conversion (addition 15 times)
4'b0110: 32-time conversion (addition 31 times)
4'b0111: 64-time conversion (addition 63 times)
Others: Reserved.

15.6.14 Self-diagnosis data register (ADC_SDDATA)

Address offset: 0x50

Reset value: 0x0000 0000

This register has to be accessed by word(32-bit).



LSB alignment (DAL=0):

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:14	SDSTA[1:0]	Voltage status converted by self-diagnosis 00: Self-diagnosis has never been executed since power-on. 01: 0 converted by self-diagnosis 10: $V_{REFP}/2$ converted by self-diagnosis 11: V_{REFP} converted by self-diagnosis
13:12	Reserved	Must be kept at reset value.
11:0	SDDATA[11:0]	Self-diagnosis channel data These bits are the conversion result of the self-diagnosis channel, which are read only.

MSB alignment (DAL=1):

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:4	SDDATA[11:0]	Self-diagnosis channel data These bits are the conversion result of the self-diagnosis channel, which are read only.

only.

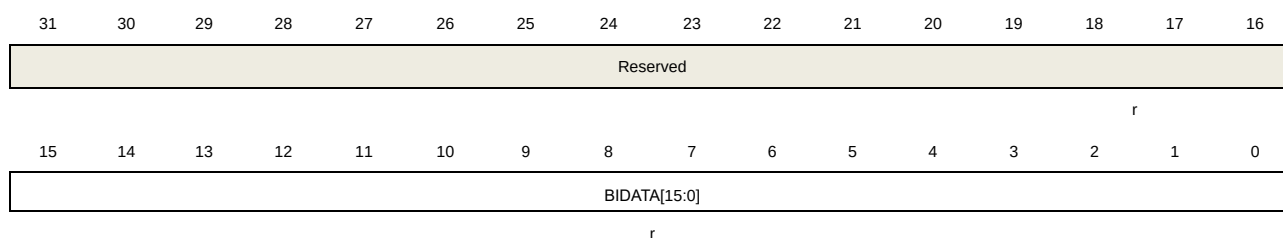
3:2	Reserved	Must be kept at reset value.
1:0	SDSTA[1:0]	Voltage status converted by self-diagnosis 00: Self-diagnosis has never been executed since power-on. 01: 0 converted by self-diagnosis 10: $V_{REFP}/2$ converted by self-diagnosis 11: V_{REFP} converted by self-diagnosis

15.6.15 Group_pri x bifurcate data register (ADC_GPxBIDATA, x=1,2,3,4)

Address offset: $0x60+(x-1)*0xC$

Reset value: 0x0000 0000

This register has to be accessed by word(32-bit).



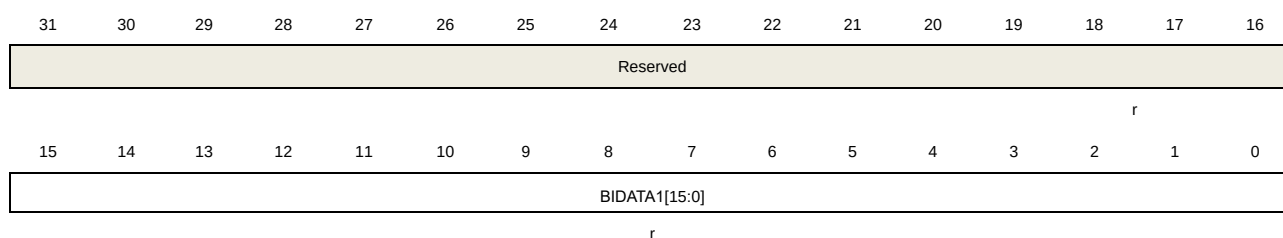
Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	BIDATA[15:0]	The data by the second trigger in bifurcate trigger mode If bifurcate trigger mode is enabled, these bits contain the channel (GPxBICHSEL[4:0]) conversion result by the second trigger, which is read only.

15.6.16 Group_pri x bifurcate data register 1(ADC_GPxBIDATA1, x=1,2,3,4)

Address offset: $0x64+(x-1)*0xC$

Reset value: 0x0000 0000

This register has to be accessed by word(32-bit).



Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.

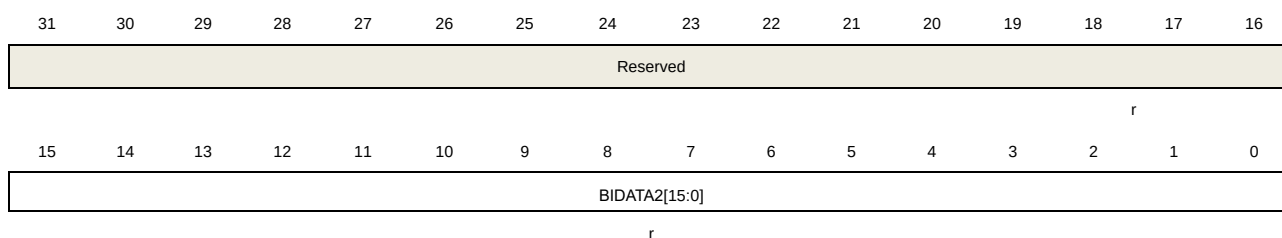
15:0	BIDATA1[15:0]	The data by the TRGA trigger in extended bifurcate trigger mode If bifurcate trigger mode is enabled, these bits contain the channel (GPxBICHSEL[4:0]) conversion result by the second trigger, which is read only.

15.6.17 Group_pri x bifurcate data register 2(ADC_GPxBIDATA2, x=1,2,3,4)

Address offset: $0x68 + (x-1) \times 0xC$

Reset value: 0x0000 0000

This register has to be accessed by word(32-bit).



Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	BIDATA2[15:0]	The data by the TRGB trigger in extended bifurcate trigger mode If bifurcate trigger mode is enabled, these bits contain the channel (GPxBICHSEL[4:0]) conversion result by the second trigger, which is read only.

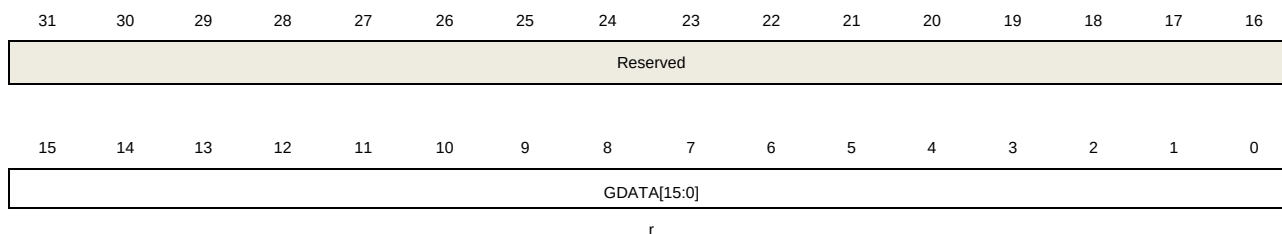
15.6.18 Group_pri x data DMA register (ADC_GPxDMAR, x=1,2,3,4)

Address offset: $0x90 + (x-1) \times 0x4$

Reset value: 0x0000 0000

This register is only recommended to be accessed by DMA. (Not recommended to be accessed by CPU).

This register has to be accessed by word(32-bit).



Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	GDATA[15:0]	The data of Group_prix which can be accessed by DMA. These bits is read only.

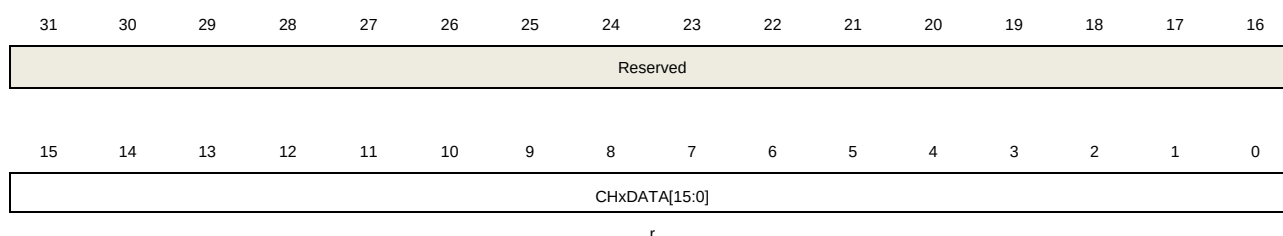
15.6.19 Channel x data register (ADC_CHxDATA)

Address offset: $0xA0 + 0x4 \times x$

Reset value: 0x0000 0000

For ADC0, $x=0..6$. For ADC2, $x=0..9$.

This register has to be accessed by word(32-bit).



Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CHxDATA[15:0]	Channel x data These bits contain the conversion result of channel x, which is read only.

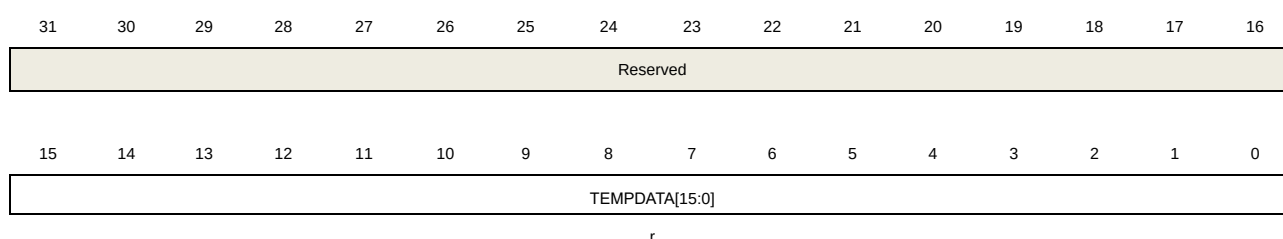
15.6.20 Temperature sensor channel data register (ADC_TEMPDATA)

Address offset: 0xC8

Reset value: 0x0000 0000

This register is only used in ADC2.

This register has to be accessed by word(32-bit)



Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	TEMPDATA[15:0]	Temperature sensor channel data These bits contain the conversion result of the temperature sensor channel, which is read only.

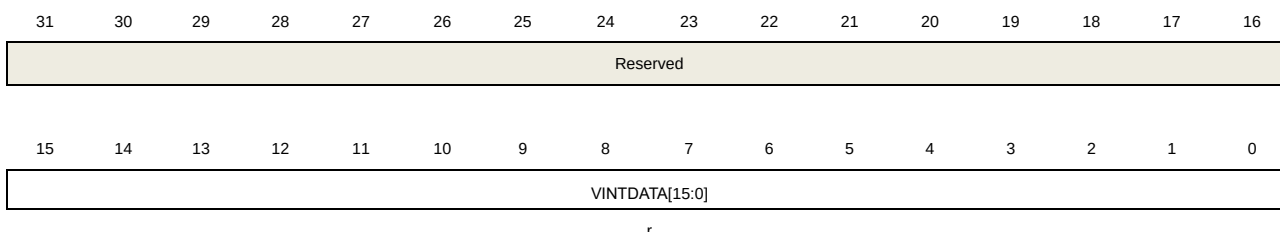
15.6.21 Internal reference voltage channel data register (ADC_VINTDATA)

Address offset: 0xCC

Reset value: 0x0000 0000

This register is only used in ADC2.

This register has to be accessed by word(32-bit).



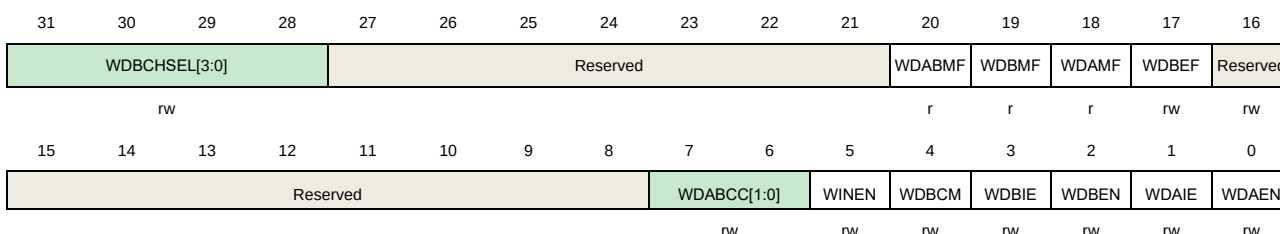
Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	VINTDATA[15:0]	Internal reference voltage channel data These bits contain the conversion result of the Internal reference voltage channel, which is read only.

15.6.22 Watchdog control register (ADC_WDCTL)

Address offset: 0x100

Reset value: 0x0000 0000

This register has to be accessed by word(32-bit).



Bits	Fields	Descriptions
31:28	WDBCHSEL[3:0]	Analog watchdog B channel select 0000: ADCx_IN0 0001: ADCx_IN1 0010: ADCx_IN2 0011: ADCx_IN3 0100: ADCx_IN4 0101: ADCx_IN5 0110: ADCx_IN6 0111: ADCx_IN7, only used in ADC2. 1000: ADCx_IN8, only used in ADC2. 1001: ADCx_IN9, only used in ADC2. 1010: Temperature sensor channel, only used in ADC2.

		1011: Internal reference voltage channel, only used in ADC2. Others: Reserved
27:21	Reserved	Must be kept at reset value.
20	WDABMF	Analog watchdog A/B complex compare monitor flag This bit indicates the combination result. This bit is valid when both watchdog A operation and watchdog B operation are enabled. 0: The complex condition for analog watchdog A/B is not satisfied. 1: The complex condition for analog watchdog A/B is satisfied. Set and cleared by hardware. Readable only by software.
19	WDBMF	Analog watchdog B compare monitor flag 0: The condition for analog watchdog B is not satisfied. 1: The condition for analog watchdog B is satisfied. Set and cleared by hardware. Readable only by software.
18	WDAMF	Analog watchdog A compare monitor flag 0: The condition for analog watchdog A is not satisfied. 1: The condition for analog watchdog A is satisfied. Set and cleared by hardware. Readable only by software.
17	WDBEF	Analog watchdog B event flag 0: The condition for analog watchdog B is not satisfied. 1: The condition for analog watchdog B is satisfied. Set by hardware when the comparison condition for analog watchdog B is satisfied. Cleared by software writing 0 to it.
16:8	Reserved	Must be kept at reset value.
7:6	WDABCC[1:0]	Watchdog A/B complex conditions configuration 00: (Watchdog A condition matched) OR (Watchdog B condition matched) 01: (Watchdog A condition matched) XOR (Watchdog B condition matched) 10: (Watchdog A condition matched) AND (Watchdog B condition matched) 11: Reserved
5	WINEN	Window mode enable 0: Window mode disable. The watchdog A or B compares the low thresholds and A/D-converted value like a comparator. 1: Window mode enable. The watchdog A or B compares between high/low thresholds and A/D-converted value.
4	WDBCM	Watchdog B compare conditions mode When the WINEN is 0, the compare conditions is: 0: WDBLT[15:0] value > A/D-converted value 1: WDBLT[15:0] value < A/D-converted value When the WINEN is 1, the compare conditions is:

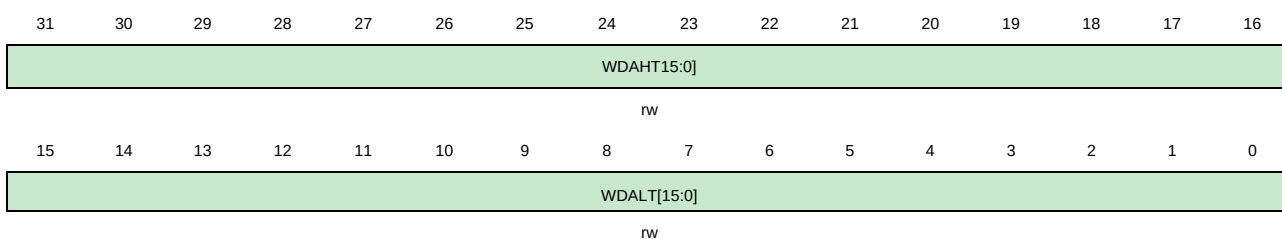
		0: (A/D-converted value < WDBLT[15:0] value) or (WDBHT[15:0] < A/D-converted value)
		1: WDBLT[15:0] value < A/D-converted value < WDBHT[15:0] value.
3	WDBIE	Interrupt enable for WDBEF 0: Watchdog B interrupt disable 1: Watchdog B interrupt enable
2	WDBEN	Watchdog B enable 0: Analog watchdog B disable 1: Analog watchdog B enable
1	WDAIE	Interrupt enable for watchdog A When this bit is 1, an interrupt request will be generated if any bit in ADC_WDACHSTAT register is 1. 0: Watchdog A interrupt disable 1: Watchdog A interrupt enable
0	WDAEN	Watchdog A enable 0: Analog watchdog A disable 1: Analog watchdog A enable

15.6.23 Watchdog A threshold register (ADC_WDATHOLD)

Address offset: 0x104

Reset value: 0xFFFF 0000

This register has to be accessed by word(32-bit).



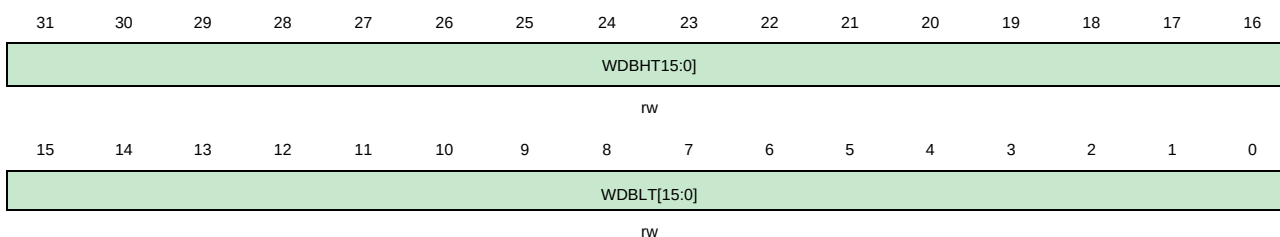
Bits	Fields	Descriptions
31:16	WDAHT[15:0]	High threshold for analog watchdog A These bits define the high threshold for the analog watchdog A.
15:0	WDALT[15:0]	Low threshold for analog watchdog A These bits define the low threshold for the analog watchdog A.

15.6.24 Watchdog B threshold register (ADC_WDBTHOLD)

Address offset: 0x108

Reset value: 0xFFFF 0000

This register has to be accessed by word(32-bit).



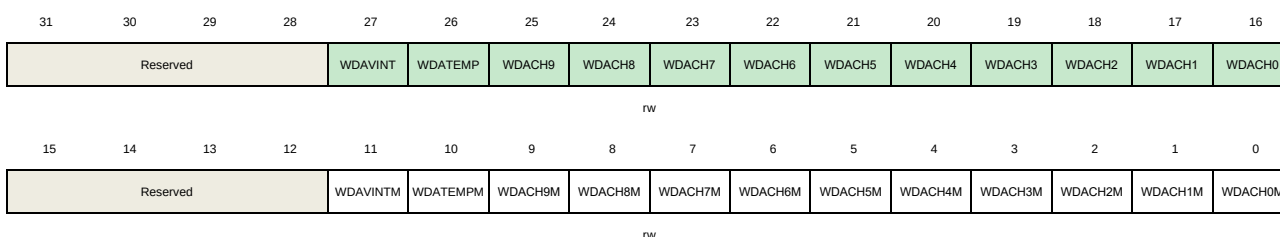
Bits	Fields	Descriptions
31:16	WDBHT[15:0]	High threshold for analog watchdog B These bits define the high threshold for the analog watchdog B.
15:0	WDBLT[15:0]	Low threshold for analog watchdog B These bits define the low threshold for the analog watchdog B.

15.6.25 Watchdog A channel config register (ADC_WDACHCFG)

Address offset: 0x10C

Reset value: 0x0000 0000

This register has to be accessed by word(32-bit).



Bits	Fields	Descriptions
31:28	Reserved	Must be kept at reset value.
27	WDAVINT	Select internal reference voltage channel for watchdog A 0: Deselect the channel 1: Select the channel Note: This bit is only used in ADC2.
26	WDATMP	Select temperature sensor channel for watchdog A 0: Deselect the channel 1: Select the channel Note: This bit is only used in ADC2.
25	WDACH9	Select ADCx_IN9 for watchdog A 0: Deselect the channel 1: Select the channel Note: This bit is only used in ADC2.

24	WDACH8	Select ADCx_IN8 for watchdog A 0: Deselect the channel 1: Select the channel Note: This bit is only used in ADC2.
23	WDACH7	Select ADCx_IN7 for watchdog A 0: Deselect the channel 1: Select the channel Note: This bit is only used in ADC2.
22	WDACH6	Select ADCx_IN6 for watchdog A 0: Deselect the channel 1: Select the channel
21	WDACH5	Select ADCx_IN5 for watchdog A 0: Deselect the channel 1: Select the channel
20	WDACH4	Select ADCx_IN4 for watchdog A 0: Deselect the channel 1: Select the channel
19	WDACH3	Select ADCx_IN3 for watchdog A 0: Deselect the channel 1: Select the channel
18	WDACH2	Select ADCx_IN2 for watchdog A 0: Deselect the channel 1: Select the channel
17	WDACH1	Select ADCx_IN1 for watchdog A 0: Deselect the channel 1: Select the channel
16	WDACH0	Select ADCx_IN0 for watchdog A 0: Deselect the channel 1: Select the channel
15:12	Reserved	Must be kept at reset value.
11	WDAVINTM	Internal reference voltage channel watchdog A compare conditions mode Refer to WDACH0M. Note: This bit is only used in ADC2.
10	WDATMPM	Temperature sensor channel watchdog A compare conditions mode Refer to WDACH0M. Note: This bit is only used in ADC2.
9	WDACH9M	Channel ADCx_IN9 watchdog A compare conditions mode

		Refer to WDACH0M.
		Note: This bit is only used in ADC2.
8	WDACH8M	Channel ADCx_IN8 watchdog A compare conditions mode Refer to WDACH0M. Note: This bit is only used in ADC2.
7	WDACH7M	Channel ADCx_IN7 watchdog A compare conditions mode Refer to WDACH0M. Note: This bit is only used in ADC2.
6	WDACH6M	Channel ADCx_IN6 watchdog A compare conditions mode Refer to WDACH0M.
5	WDACH5M	Channel ADCx_IN5 watchdog A compare conditions mode Refer to WDACH0M.
4	WDACH4M	Channel ADCx_IN4 watchdog A compare conditions mode Refer to WDACH0M.
3	WDACH3M	Channel ADCx_IN3 watchdog A compare conditions mode Refer to WDACH0M.
2	WDACH2M	Channel ADCx_IN2 watchdog A compare conditions mode Refer to WDACH0M.
1	WDACH1M	Channel ADCx_IN1 watchdog A compare conditions mode Refer to WDACH0M.
0	WDACH0M	Channel ADCx_IN0 watchdog A compare conditions mode When the WINEN is 0, the compare conditions is: 0: WDALT[15:0] value > A/D-converted value 1: WDALT[15:0] value < A/D-converted value When the WINEN is 1, the compare conditions is: 0: (A/D-converted value < WDALT[15:0] value) or (WDAHT[15:0] < A/D-converted value) 1: WDALT[15:0] value < A/D-converted value < WDAHT[15:0] value.

15.6.26 Watchdog A channel status register (ADC_WDACHSTAT)

Address offset: 0x110

Reset value: 0x0000 0000

This register has to be accessed by word(32-bit).

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

Reserved	VINTCMPF	TEMPCMPF	CH9CMPF	CH8CMPF	CH7CMPF	CH6CMPF	CH5CMPF	CH4CMPF	CH3CMPF	CH2CMPF	CH1CMPF	CH0CMPF
	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

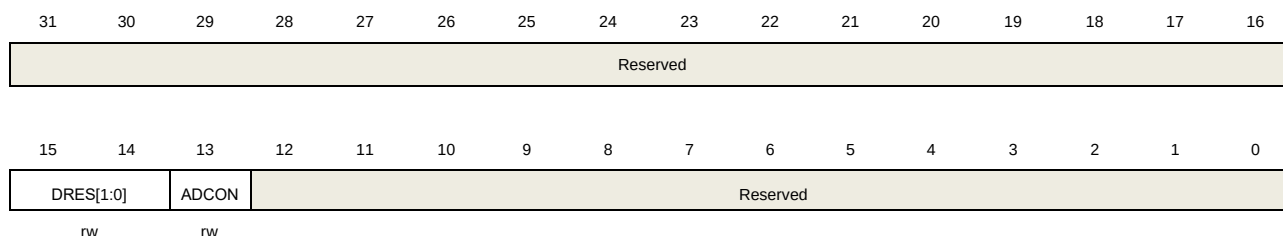
Bits	Fields	Descriptions
31:12	Reserved	Must be kept at reset value.
11	VINTCMPF	Internal reference voltage channel compare status in watchdog A 0: The comparison condition is not satisfied. 1: The comparison condition is satisfied. Note: This bit is only used in ADC2.
10	TEMPCMPF	Temperature sensor channel compare status in watchdog A 0: The comparison condition is not satisfied. 1: The comparison condition is satisfied. Note: This bit is only used in ADC2.
9	CH9CMPF	Channel ADCx_IN9 compare status in watchdog A Refer to CH0CMPF description. Note: This bit is only used in ADC2.
8	CH8CMPF	Channel ADCx_IN8 compare status in watchdog A Refer to CH0CMPF description. Note: This bit is only used in ADC2.
7	CH7CMPF	Channel ADCx_IN7 compare status in watchdog A Refer to CH0CMPF description. Note: This bit is only used in ADC2.
6	CH6CMPF	Channel ADCx_IN6 compare status in watchdog A Refer to CH0CMPF description.
5	CH5CMPF	Channel ADCx_IN5 compare status in watchdog A Refer to CH0CMPF description.
4	CH4CMPF	Channel ADCx_IN4 compare status in watchdog A Refer to CH0CMPF description.
3	CH3CMPF	Channel ADCx_IN3 compare status in watchdog A Refer to CH0CMPF description.
2	CH2CMPF	Channel ADCx_IN2 compare status in watchdog A Refer to CH0CMPF description.
1	CH1CMPF	Channel ADCx_IN1 compare status in watchdog A Refer to CH0CMPF description.
0	CH0CMPF	Channel ADCx_IN0 compare status in watchdog A 0: The comparison condition is not satisfied. 1: The comparison condition is satisfied.

15.6.27 Control register 1 (ADC_CTL1)

Address offset: 0x150

Reset value: 0x0000 1000

This register has to be accessed by word(32-bit).



Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:14	DRES[1:0]	ADC resolution 00: 12-bit 01: 10-bit 10: 8-bit 11: 6-bit
13	ADCON	ADC ON The ADC will be waked up when this bit is changed from low to high and take a stabilization time. When this bit is high and “1” is written to it with other bits of this register unchanged, the conversion will start. 0: ADC disable and power down 1: ADC enable
12:0	Reserved	Must be kept at reset value.

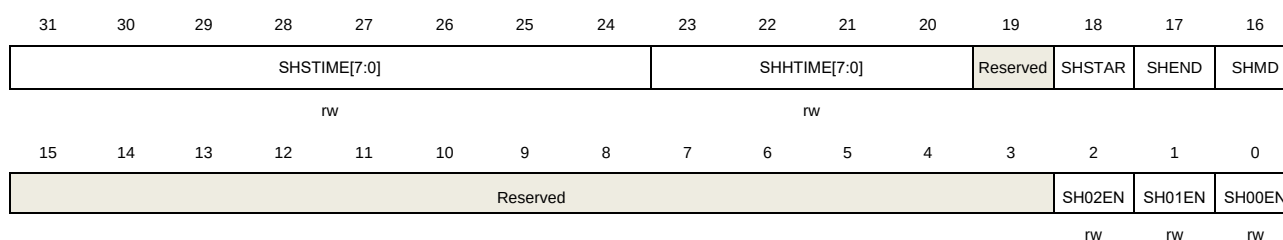
15.6.28 Sample-and-hold control register (ADC_SHCTL)

Address offset: 0x160

Reset value: 0x0880 0000

This register is only used in ADC0.

This register has to be accessed by word(32-bit).



Bits	Fields	Descriptions
31:24	SHSTIME[7:0]	The sample time of sample-and-hold circuit The sample time for channel sample-and-hold circuit will be SHSTIME[7:0] CLK_ADC. When SHMD is 1, the sample time for channel sample-and-hold circuit is controlled by constant sampling mode. Note: The recommended config value of SHSTIME[7:0] should not be less than 8. Otherwise, the accuracy of the sample-and-hold circuit cannot be guaranteed.
23:20	SHHTIME[3:0]	The hold time of sample-and-hold circuit The hold time for channel-dedicated sample-and-hold circuit will be SHHTIME[3:0] CLK_ADC. Note: The recommended config value of SHHTIME[3:0] should not be less than 8. Otherwise, the accuracy of the sample-and-hold circuit cannot be guaranteed.
19	Reserved	Must be kept at reset value.
18	SHSTART	Software start sample-and-hold circuit in constant sampling mode This bit is valid only when SHMD is 1. Set 1 on this bit by software to start sample and hold circuit working. Cleared by hardware after sample and hold circuit working.
17	SHEND	Software end sample-and-hold circuit in constant sampling mode This bit is valid only when SHMD is 1. Set 1 on this bit by software to end sample and hold circuit working. Cleared by hardware after sample and hold circuit stopping.
16	SHMD	Channel-dedicated sample-and-hold circuit operating mode 0: Constant sampling of the channel-dedicated sample-and-hold circuits are disabled. 1: Constant sampling of the channel-dedicated sample-and-hold circuits are enabled.
15:3	Reserved	Must be kept at reset value.
2	SH02EN	Enable channel ADCx_IN2 sample-and-hold circuit The bit enables channel ADCx_IN2 dedicated sample-and-hold circuits. 0: Disable channel-dedicated sample-and-hold circuits. 1: Enable channel-dedicated sample-and-hold circuits.
1	SH01EN	Enable channel ADCx_IN1 sample-and-hold circuit The bit enables channel ADCx_IN1 dedicated sample-and-hold circuits. 0: Disable channel-dedicated sample-and-hold circuits. 1: Enable channel-dedicated sample-and-hold circuits.
0	SH00EN	Enable channel ADCx_IN0 sample-and-hold circuit The bit enables channel ADCx_IN0 dedicated sample-and-hold circuits. 0: Disable channel-dedicated sample-and-hold circuits.

1: Enable channel-dedicated sample-and-hold circuits.

15.6.29 DMA control register (ADC_DMACTL)

Address offset: 0x170

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DMAEN4	GP4DDM	GP4OVRF	OVREN4	GP4OVRIE	Reserved			DMAEN3	GP3DDM	GP3OVRF	OVREN3	GP3OVRIE	Reserved		
rw	rw	rw	rw	rw				rw	rw	rw	rw	rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMAEN2	GP2DDM	GP2OVRF	OVREN2	GP2OVRIE	Reserved			DMAEN1	GP1DDM	GP1OVRF	OVREN1	GP1OVRIE	Reserved		
rw	rw	rw	rw	rw				rw	rw	rw	rw	rw			

Bits	Fields	Descriptions
31	DMAEN4	DMA request enable of Group_pri4 0: DMA request disable 1: DMA request enable
30	GP4DDM	DMA disable mode of Group_pri4 This bit configure the DMA disable mode for single ADC mode 0: The DMA engine is disabled after the end of transfer signal from DMA controller is detected 1: When DMAEN4=1, the DMA engine issues request at end of conversion of Group_pri4
29	GP4OVRF	DMA overflow flag of Group_pri4 0: No overflow 1: DMA overflow Set by hardware when DMA is overflow. Cleared by software writing 0 to it.
28	OVREN4	DMA overflow detect enable of Group_pri4. 0: Disable overflow detect 1: Enable overflow detect
27	GP4OVRIE	DMA overflow detect interrupt enable of Group_pri4 0: Overflow interrupt disable 1: Overflow interrupt enable
26:24	Reserved	Must be kept at reset value
23	DMAEN3	DMA request enable of Group_pri3 0: DMA request disable 1: DMA request enable

22	GP3DDM	DMA disable mode of Group_pri3 This bit configure the DMA disable mode for single ADC mode 0: The DMA engine is disabled after the end of transfer signal from DMA controller is detected 1: When DMAEN3=1, the DMA engine issues request at end of conversion of Group_pri3
21	GP3OVRF	DMA overflow flag of Group_pri3 0: No overflow 1: DMA overflow Set by hardware when DMA is overflow. Cleared by software writing 0 to it.
20	OVREN3	DMA overflow detect enable of Group_pri3 0: Disable overflow detect 1: Enable overflow detect
19	GP3OVRIE	DMA overflow detect interrupt enable of Group_pri3 0: Overflow interrupt disable 1: Overflow interrupt enable
18:16	Reserved	Must be kept at reset value
15	DMAEN2	DMA request enable of Group_pri2 0: DMA request disable 1: DMA request enable
14	GP2DDM	DMA disable mode of Group_pri2 This bit configure the DMA disable mode for single ADC mode 0: The DMA engine is disabled after the end of transfer signal from DMA controller is detected 1: When DMAEN2=1, the DMA engine issues request at end of conversion of Group_pri2
13	GP2OVRF	DMA overflow flag of Group_pri2 0: No overflow 1: DMA overflow Set by hardware when DMA is overflow. Cleared by software writing 0 to it.
12	OVREN2	DMA overflow detect enable of Group_pri2 0: Disable overflow detect 1: Enable overflow detect
11	GP2OVRIE	DMA overflow detect interrupt enable of Group_pri2 0: Overflow interrupt disable 1: Overflow interrupt enable

10:8	Reserved	Must be kept at reset value
7	DMAEN1	DMA request enable of Group_pri1 0: DMA request disable 1: DMA request enable
6	GP1DDM	DMA disable mode of Group_pri1 This bit configure the DMA disable mode for single ADC mode 0: The DMA engine is disabled after the end of transfer signal from DMA controller is detected 1: When DMAEN1=1, the DMA engine issues request at end of conversion of Group_pri1
5	GP1OVRF	DMA overflow flag of Group_pri1 0: No overflow 1: DMA overflow Set by hardware when DMA is overflow. Cleared by software writing 0 to it.
4	OVREN1	DMA overflow detect enable of Group_pri1 0: Disable overflow detect 1: Enable overflow detect
3	GP1OVRIE	DMA overflow detect interrupt enable of Group_pri1 0: Overflow interrupt disable 1: Overflow interrupt enable
2:0	Reserved	Must be kept at reset value

15.6.30 Bifurcate trigger control register (ADC_BITRGCTL)

Address offset: 0x180

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GP4BICHSEL[4:0]				GP4BIMEN	GP4BITGRS	Reserved	GP3BICHSEL[4:0]				GP3BIMEN	GP3BITGRS	Reserved		
rw				rw	rw		rw				rw	rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GP2BICHSEL[4:0]				GP2BIMEN	GP2BITGRS	Reserved	GP1BICHSEL[4:0]				GP1BIMEN	GP1BITGRS	Reserved		
rw				rw	rw		rw				rw	rw			

Bits	Fields	Descriptions
31:27	GP4BICHSEL[4:0]	Select bifurcate trigger channel of Group_pri4 Refer to GP1BICHSEL[4:0]. The bit-fields can be configured while the PROCF bit is 0.

26	GP4BIMEN	Enable bifurcate trigger mode of Group_pri4 The bit can be configured while the PROCF bit is 0. 0: Disable bifurcate trigger mode 1: Enable bifurcate trigger mode Note: 1) When SCAN[1:0] bits are 2'b10, bifurcate trigger mode should disabled. 2) Bifurcate trigger mode should not be used for self-diagnosis, internal reference voltage channel and temperature sensor channel.
25	GP4BITGRS	Enable the next trigger restore of Group_pri4 This bit is only effective while Group_pri4 bifurcate trigger mode is enabled (GP4BIMEN =1). When this bit is set to 1, for the next trigger during the current A/D conversion round, it will restore after the current A/D conversion round complete. Otherwise, the next trigger during the current A/D conversion round will be ignored. 0: The next trigger is ignored. 1: The next trigger is restored.
24	Reserved	Must be kept at reset value.
23:19	GP3BICHSEL[4:0]	Select bifurcate trigger channel of Group_pri3 Refer to GP1BICHSEL[4:0].
18	GP3BIMEN	Enable bifurcate trigger mode of Group_pri3 The bit can be configured while the PROCF bit is 0. 0: Disable bifurcate trigger mode 1: Enable bifurcate trigger mode Note: 1) When GSCAN[1:0] bits are 2'b10, bifurcate trigger mode should disabled. 2) Bifurcate trigger mode should not be used for self-diagnosis, internal reference voltage channel and temperature sensor channel.
17	GP3BITGRS	Enable the next trigger restore of Group_pri3 This bit is only effective while Group_pri3 bifurcate trigger mode is enabled (GP3BIMEN =1). When this bit is set to 1, for the next trigger during the current A/D conversion round, it will restore after the current A/D conversion round complete. Otherwise, the next trigger during the current A/D conversion round will be ignored. 0: The next trigger is ignored. 1: The next trigger is restored.
16	Reserved	Must be kept at reset value.
15:11	GP2BICHSEL[4:0]	Select bifurcate trigger channel of Group_pri2 Refer to GP1BICHSEL[4:0].
10	GP2BIMEN	Enable bifurcate trigger mode of Group_pri2

		The bit can be configured while the PROCF bit is 0. 0: Disable bifurcate trigger mode 1: Enable bifurcate trigger mode Note: 1) When GSCAN[1:0] bits are 2'b10, bifurcate trigger mode should disabled. 2) Bifurcate trigger mode should not be used for self-diagnosis.
9	GP2BITGRS	Enable the next trigger restore of Group_pri2 This bit is only effective while Group_pri2 bifurcate trigger mode is enabled (GP2BIMEN =1). When this bit is set to 1, for the next trigger during the current A/D conversion round, it will restore after the current A/D conversion round complete. Otherwise, the next trigger during the current A/D conversion round will be ignored. 0: The next trigger is ignored. 1: The next trigger is restored..
8	Reserved	Must be kept at reset value.
7:3	GP1BICHSEL[4:0]	Select bifurcate trigger channel of Group_pri1 The bit-field can be configured while the PROCF bit is 0. 00000: ADCx_IN0 00001: ADCx_IN1 00010: ADCx_IN2 00011: ADCx_IN3 00100: ADCx_IN4 00101: ADCx_IN5 00110: ADCx_IN6 00111: ADCx_IN7, only used in ADC2. 01000: ADCx_IN8, only used in ADC2. 01001: ADCx_IN9, only used in ADC2. 01101: Temperature sensor channel, only used in ADC2. 01110: Internal reference voltage channel, only used in ADC2. Others: Reserved Note: 1) These bits are only effective while Group_pri1 bifurcate trigger mode is enabled (GP1BIMEN =1). 2) When Group_pri1 bifurcate mode is enabled (GP1BIMEN =1), the channels selected by the ADC_CHSEL0 and ADC_CHSEL1 registers are invalid for Group_pri1.
2	GP1BIMEN	Enable bifurcate trigger mode of Group_pri1 The bit can be configured while the PROCF bit is 0. 0: Disable bifurcate trigger mode 1: Enable bifurcate trigger mode Note:

		1) When GSCAN[1:0] bits are 2'b10, bifurcate trigger mode should disabled.
		2) Bifurcate trigger mode should not be used for self-diagnosis, internal reference voltage channel and temperature sensor channel.
1	GP1BITGRS	Enable the next trigger restore of Group_pri1 This bit is only effective while Group_pri1 bifurcate trigger mode is enabled (GP1BIMEN =1). When this bit is set to 1, for the next trigger during the current A/D conversion round, it will restore after the current A/D conversion round complete. Otherwise, the next trigger during the current A/D conversion round will be ignored. 0: The next trigger is ignored. 1: The next trigger is restored.
0	Reserved	Must be kept at reset value.

16 Digital-to-analog converter (DAC)

16.1 Overview

The Digital-to-analog converter converts 12-bit digital data to a voltage on the external pins. The digital data can be configured to left-aligned or right-aligned mode. The voltage can be used for comparator reference input voltage and analog output voltage.

The DAC channels can work independently or concurrently.

16.2 Characteristics

The main features of DAC are as follows:

- 12-bit resolution.
- Left or right data alignment.
- Conversion update synchronously.
- Conversion triggered by external triggers.
- Extern voltage reference, V_{REFP} .
- Noise wave generation (LFSR noise mode and Triangle noise mode).
- Two DAC channels in concurrent mode.
- Generating comparator reference input voltage.
- Output can connect to analog output pin.
- DAC conversion can be enabled by EVIC signals.

[Figure 16-1. DAC block diagram](#) and [Table 16-1. DAC I/O description](#) show the block diagram of DAC and the pin description of DAC, respectively.

Figure 16-1. DAC block diagram

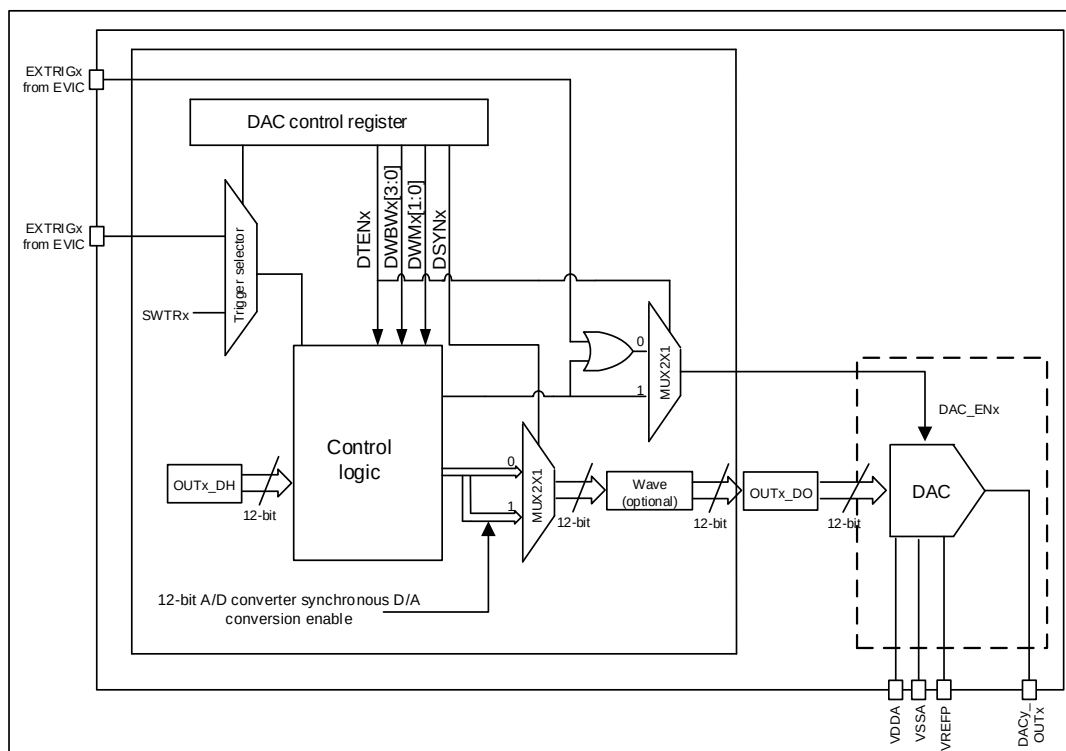


Table 16-1. DAC I/O description

Name	Description	Signal type
V _{DDA}	Analog power supply	Input, analog supply
V _{SSA}	Ground for analog power supply	Input, analog supply ground
V _{REFP}	Positive reference voltage of DAC	Input, analog positive reference
DACy_OUTx	DAC analog output	Analog output signal

The below table details the triggers and outputs of the DAC.

Table 16-2. DAC triggers and outputs summary

	DAC0	
Channel	Channel0	Channel1
DAC outputs connected to I / Os	PD4	PD5
DAC software trigger	•	
EVIC hardware trigger	•	

Note: The GPIO pins should be configured to analog mode before enable the DAC module.

16.3 Function description

16.3.1 DAC enable

The DAC can be turned on when EVIC trigger event occurs or by setting the DENx bit in the DAC_CTL0 register. A t_{WAKEUP} time is needed to startup the analog DAC submodule.

Note: EVIC trigger event enable DAC is only available when the DTENx bit is 1'b0.

16.3.2 DAC data configuration

The 12-bit DAC holding data (OUTx_DH) can be configured by writing any one of the DAC_OUTx_R12DH and DAC_OUTx_L12DH registers.

16.3.3 DAC trigger

The DAC conversion can be triggered by software or rising edge of external trigger source. The DAC external trigger is enabled by setting the DTENx bits in the DAC_CTL0 register. The DAC external triggers are selected by the DTSELx bits in the DAC_CTL0 register, which is shown as [Table 16-3. Triggers of DAC](#).

Table 16-3. Triggers of DAC

DTSELx[1:0]	Trigger Source	Trigger Type
2b'00	EXTRIG	Hardware trigger
2b'01	SWTR	Software trigger
2b'1x	Reserved	Reserved

The EXTRIG signals are generated from event interconnection unit (EVIC), while the software trigger can be generated by setting the SWTRx bits in the DAC_SWT register.

16.3.4 DAC conversion

DAC has two conversion modes, the trigger conversion occurs when DTENx=1'b1 (in DAC_CTL register), and the automatic conversion occurs when DTENx=1'b0. which is shown as [Table 16-4. Conversion mode of DAC](#).

Table 16-4. Conversion mode of DAC

DTENx	DENx	DTSELx[2:0]	Description
1	0	-	No conversion
	1	00	External trigger conversion
		01	Software trigger conversion
		1x	No conversion
0	0	-	Automatic conversion when EVIC trigger event occurs
	1	-	Automatic conversion

Trigger conversion

If the external trigger is enabled by setting the DTENx bit (in DAC_CTL0 register) to 1, and DAC is enable by setting the DENx bit in the DAC_CTL0 register, the DAC holding data (OUTx_DH) is transferred to the DAC output data (DAC_OUTx_DO) register when the selected trigger event happened.

Automatic conversion

When the external trigger is disabled by setting the DTENx bit to 0, and DAC is enable by setting the DENx bit in the DAC_CTL0 register, the transfer is performed automatically.

When external trigger is disable by setting the DTENx bit to 0, and DAC will be enabled when EVIC trigger event occurs, the transfer is performed automatically.

When the DAC holding data (OUTx_DH) is loaded into the DAC_OUTx_DO register, after the time $t_{SETTLING}$ which is determined by the analog output load and the power supply voltage, the analog output is valid.

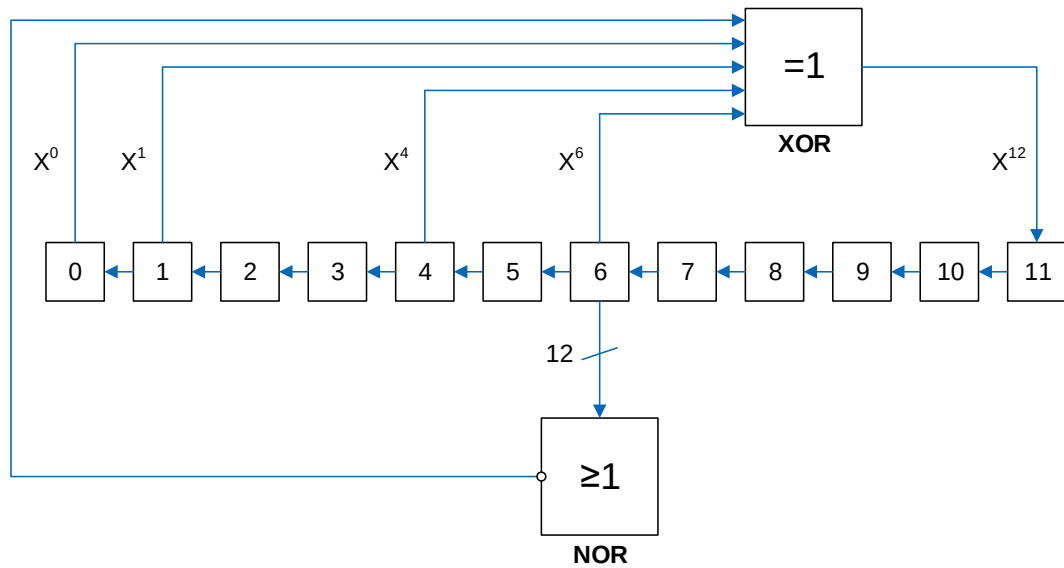
16.3.5 DAC noise wave

There are two methods of adding noise wave to the DAC output data: LFSR noise wave mode and Triangle wave mode. The noise wave mode can be selected by the DWMx bits in the DAC_CTL0 register. The amplitude of the noise can be configured by the DAC noise wave bit width (DWBWx) bits in the DAC_CTL0 register.

LFSR noise wave mode: there is a Linear Feedback Shift Register (LFSR) in the DAC control logic, it controls the LFSR noise signal which is added to the OUTx_DH value, and then the result is stored into the DAC_OUTx_DO register. When the configured DAC noise wave bit width is less than 12, the noise signal equals to the LSB DWBWx bits of the LFSR register, while the MSB bits are masked.

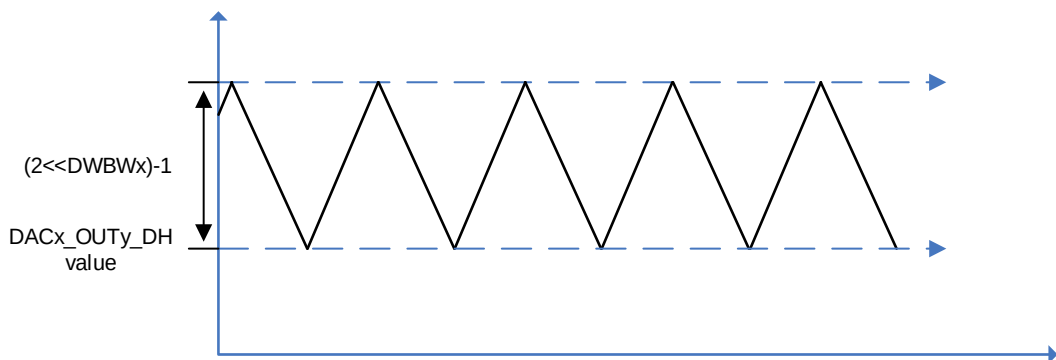
Note: The initial value of X^n is 0xAAA ($n=0$ to 11).

Figure 16-2. DAC LFSR algorithm



Triangle noise mode: a triangle signal is added to the OUTx_DH value, and then the result is stored into the DAC_OUTx_DO register. The minimum value of the triangle signal is 0, while the maximum value of the triangle signal is $(2 \llcorner DWBWx) - 1$.

Figure 16-3. DAC triangle noise wave



16.3.6 DAC output voltage

The following equation determines the analog output voltage on the DAC pin.

$$V_{DAC\_OUT} = V_{REFP} \cdot OUTx\_DO / 4096 \quad (16-1)$$

The digital input is linearly converted to an analog output voltage and its range is 0 to V_{REFP} .

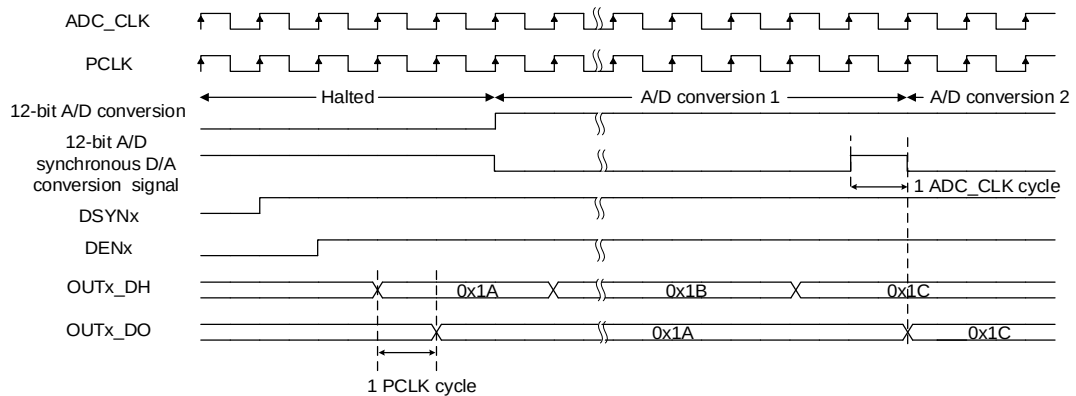
16.3.7 Prevent Interference between D/A and A/D

When D/A conversion starts, an inrush current occurs to the DAC. Since the same analog power supply is shared by the 12-bit D/A converter and 12-bit A/D converter (ADC2), the inrush current may interfere with the proper operation of ADC2.

With the DSYNx bit of the DAC_CTL0 register being 1, even if the data holding registers

(DAC_OUT0_R12DH, DAC_OUT0_L12DH, DAC_OUT1_R12DH, DAC_OUT1_L12DH, DACC_L12DH and DACC_R12DH) is modified during ADC2 operation, D/A conversion does not start immediately but starts synchronously with A/D conversion completion.

Figure 16-4. Prevent Interference between D/A and A/D



16.3.8 DAC concurrent conversion

When the two output channels work at the same time, for maximum bus bandwidth utilization in specific applications, two output channels can be configured in concurrent mode. In concurrent mode, the DAC_OUTx_DH and DAC_OUTx_DO value will be updated at the same time.

There are two concurrent registers that can be used to load the DAC_OUTx_DH value: DACC_R12DH and DACC_L12DH. User just need to access a unique register to realize driving two DAC channels at the same time.

When external trigger is enabled, please ensure both DTENx bits be set, DTSEL0/DTSEL1 bits be same to guarantee the simultaneous trigger.

The noise mode and noise bit width can be configured either the same or different, depending on the application scenario.

16.3.9 DAC low-power modes

Sleep mode

In Sleep mode, DAC can work normally.

Standby mode

In Standby mode, DAC stops working. When exiting from the standby mode, the DAC need to be reinitialized to work again.

16.4 DAC register

DAC0 base address: 0x4000 7400

16.4.1 DACx control register 0 (DAC_CTL0)

Address offset: 0x00

Reset value: 0x0000 0000

This register has to be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DOCMPR EF1	DCSEL1	Reserved	DSYN1	DWBW1[3:0]				DWM1[1:0]		Reserved	DTSEL1[1:0]		DTEN1	Reserved	DEN1
rw	rw		rw	rw				rw			rw		rw		rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DOCMPR EF0	DCSEL0	Reserved	DSYN0	DWBW0[3:0]				DWM0[1:0]		Reserved	DTSEL0[1:0]		DTEN0	Reserved	DEN0
rw	rw		rw	rw				rw			rw		rw		rw

Bits	Fields	Descriptions
31	DOCMREF1	DACx_OUT1 output for CMP reference voltage 0: DACx_OUT1 does not use as a reference voltage for CMP 1: DACx_OUT1 use as a reference voltage for CMP
30	DCSEL1	DACx_OUT1 connect selection 0: DACx_OUT1 is not connected to the external pin. 1: DACx_OUT1 is connected to the external pin.
29	Reserved	Must be kept at reset value
28	DSYN1	DAC converter operation synchronizes with ADC2 converter operation. Prevent interference between DAC and ADC2 conversion. 0: DACx_OUT1 converter operation does not synchronizes with ADC2 1: DACx_OUT1 converter operation synchronizes with ADC2
27:24	DWBW1[3:0]	DACx_OUT1 noise wave bit width These bits specify bit width of the noise wave signal of DACx_OUT1. These bits indicate that unmask LFSR bit [n-1, 0] in LFSR noise mode or the amplitude of the triangle is ((2 << (n-1)) - 1) in triangle noise mode, where n is the bit width of wave. 0000: The bit width of the wave signal is 1 0001: The bit width of the wave signal is 2 0010: The bit width of the wave signal is 3 0011: The bit width of the wave signal is 4 0100: The bit width of the wave signal is 5 0101: The bit width of the wave signal is 6

		0110: The bit width of the wave signal is 7
		0111: The bit width of the wave signal is 8
		1000: The bit width of the wave signal is 9
		1001: The bit width of the wave signal is 10
		1010: The bit width of the wave signal is 11
		≥1011: The bit width of the wave signal is 12
23:22	DWM1[1:0]	DACx_OUT1 noise wave mode These bits specify the mode selection of the noise wave signal of DACx_OUT1 when external trigger of DACx_OUT1 is enabled (DTEN1=1). 00: wave disabled 01: LFSR noise mode 1x: Triangle noise mode
21	Reserved	Must be kept at reset value.
20:19	DTSEL1[1:0]	DACx_OUT1 trigger selection These bits are only used if bit DTEN = 1 and select the external event used to trigger DAC. 00: EXTRIG (external trigger from EVIC). 01: Software trigger 1x: Reserved
18	DTEN1	DACx_OUT1 trigger enable 0: DACx_OUT1 trigger disabled 1: DACx_OUT1 trigger enabled
17	Reserved	Must be kept at reset value
16	DEN1	DACx_OUT1 enable This bit can be set by software or EVIC trigger signals. 0: DACx_OUT1 disabled 1: DACx_OUT1 enabled
15	DOCMPREF0	DACx_OUT0 output for CMP reference voltage 0: DACx_OUT0 does not use as a reference voltage for CMP 1: DACx_OUT0 use as a reference voltage for CMP
14	DCSEL0	DACx_OUT0 connect selection 0: DACx_OUT0 is not connected to the external pin. 1: DACx_OUT0 is connected to the external pin.
13	Reserved	Must be kept at reset value
12	DSYN0	DACx_OUT0 synchronization with ADC2. Prevent interference between DAC and ADC2 conversion. 0: DACx_OUT0 converter operation does not synchronizes with ADC2 1: DACx_OUT0 converter operation synchronizes with ADC2

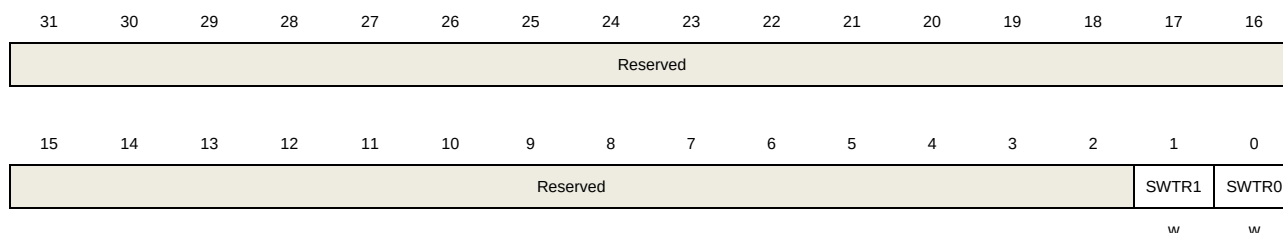
11:8	DWBW0[3:0]	DACx_OUT0 noise wave bit width These bits specify bit width of the noise wave signal of DACx_OUT0. These bits indicate that unmask LFSR bit [n-1, 0] in LFSR noise mode or the amplitude of the triangle is $((2^{n-1})-1)$ in triangle noise mode, where n is the bit width of wave. 0000: The bit width of the wave signal is 1 0001: The bit width of the wave signal is 2 0010: The bit width of the wave signal is 3 0011: The bit width of the wave signal is 4 0100: The bit width of the wave signal is 5 0101: The bit width of the wave signal is 6 0110: The bit width of the wave signal is 7 0111: The bit width of the wave signal is 8 1000: The bit width of the wave signal is 9 1001: The bit width of the wave signal is 10 1010: The bit width of the wave signal is 11 ≥1011: The bit width of the wave signal is 12
7:6	DWM0[1:0]	DACx_OUT0 noise wave mode These bits specify the mode selection of the noise wave signal of DACx_OUT0 when external trigger of DACx_OUT0 is enabled (DTEN0=1). 00: Wave disabled 01: LFSR noise mode 1x: Triangle noise mode
5	Reserved	Must be kept at reset value.
4:3	DTSEL0[1:0]	DACx_OUT0 trigger selection These bits are only used if bit DTEN = 1 and select the external event used to trigger DAC. 00: EXTRIG (external trigger from EVIC). 01: Software trigger 1x: Reserved
2	DTEN0	DACx_OUT0 trigger enable 0: DACx_OUT0 trigger disabled 1: DACx_OUT0 trigger enabled
1	Reserved	Must be kept at reset value
0	DEN0	DACx_OUT0 enable This bit can be set by software or EVIC trigger signals. 0: DACx_OUT0 disabled 1: DACx_OUT0 enabled

16.4.2 DACx software trigger register (DAC_SWT)

Address offset: 0x04

Reset value: 0x0000 0000

This register has to be accessed by word(32-bit).



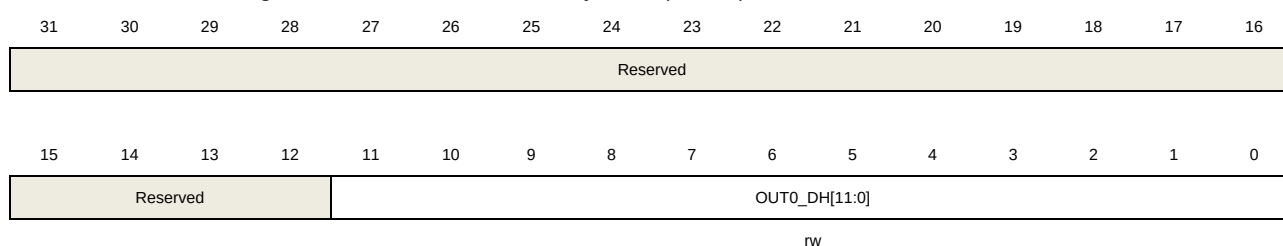
Bits	Fields	Descriptions
31:2	Reserved	Must be kept at reset value.
1	SWTR1	DACx_OUT1 software trigger, cleared by hardware when next clock. 0: Software trigger disabled 1: Software trigger enabled
0	SWTR0	DACx_OUT0 software trigger, cleared by hardware when next clock. 0: Software trigger disabled 1: Software trigger enabled

16.4.3 DACx_OUT0 12-bit right-aligned data holding register (DAC_OUT0_R12DH)

Address offset: 0x08

Reset value: 0x0000 0000

This register has to be accessed by word(32-bit).



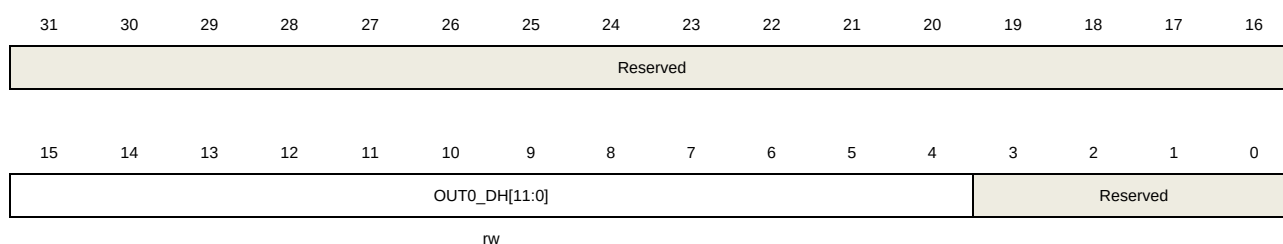
Bits	Fields	Descriptions
31:12	Reserved	Must be kept at reset value.
11:0	OUT0_DH[11:0]	DACx_OUT0 12-bit right-aligned data. These bits specify the data that is to be converted by DACx_OUT0.

16.4.4 DACx_OUT0 12-bit left-aligned data holding register (DAC_OUT0_L12DH)

Address offset: 0x0C

Reset value: 0x0000 0000

This register has to be accessed by word(32-bit).



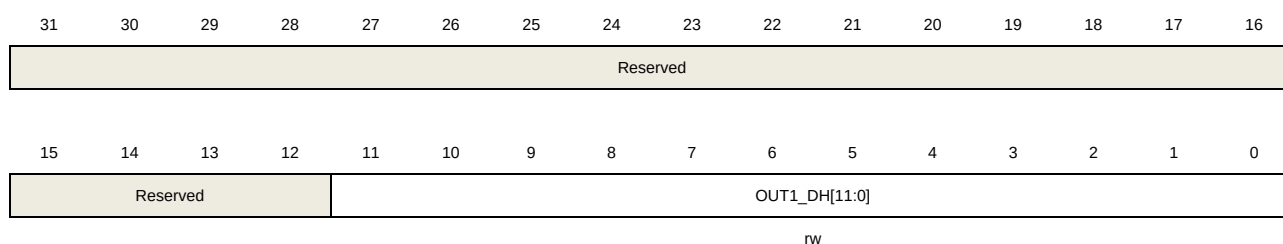
Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:4	OUT0_DH[11:0]	DACx_OUT0 12-bit left-aligned data. These bits specify the data that is to be converted by DACx_OUT0.
3:0	Reserved	Must be kept at reset value.

16.4.5 DACx_OUT1 12-bit right-aligned data holding register (DAC_OUT1_R12DH)

Address offset: 0x14

Reset value: 0x0000 0000

This register has to be accessed by word(32-bit).



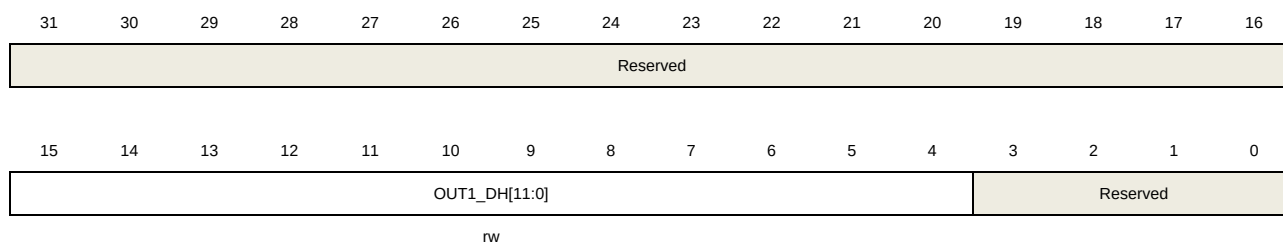
Bits	Fields	Descriptions
31:12	Reserved	Must be kept at reset value.
11:0	OUT1_DH[11:0]	DACx_OUT1 12-bit right-aligned data. These bits specify the data that is to be converted by DACx_OUT1.

16.4.6 DACx_OUT1 12-bit left-aligned data holding register (DAC_OUT1_L12DH)

Address offset: 0x18

Reset value: 0x0000 0000

This register has to be accessed by word(32-bit).



Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:4	OUT1_DH[11:0]	DACx_OUT1 12-bit left-aligned data. These bits specify the data that is to be converted by DACx_OUT1.
3:0	Reserved	Must be kept at reset value.

16.4.7 DACx concurrent mode 12-bit right-aligned data holding register (DACC_R12DH)

Address offset: 0x20

Reset value: 0x0000 0000

This register has to be accessed by word(32-bit).



Bits	Fields	Descriptions
31:28	Reserved	Must be kept at reset value.
27:16	OUT1_DH[11:0]	DACx_OUT1 12-bit right-aligned data These bits specify the data that is to be converted by DACx_OUT1.
15:12	Reserved	Must be kept at reset value.
11:0	OUT0_DH[11:0]	DACx_OUT0 12-bit right-aligned data

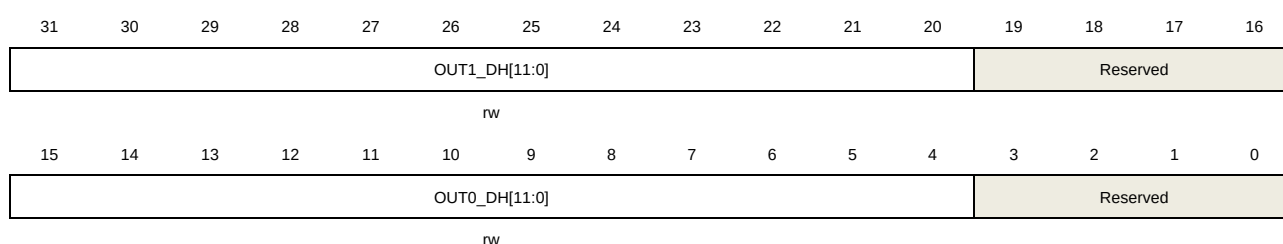
These bits specify the data that is to be converted by DACx_OUT0.

16.4.8 DACx concurrent mode 12-bit left-aligned data holding register (DACC_L12DH)

Address offset: 0x24

Reset value: 0x0000 0000

This register has to be accessed by word(32-bit)



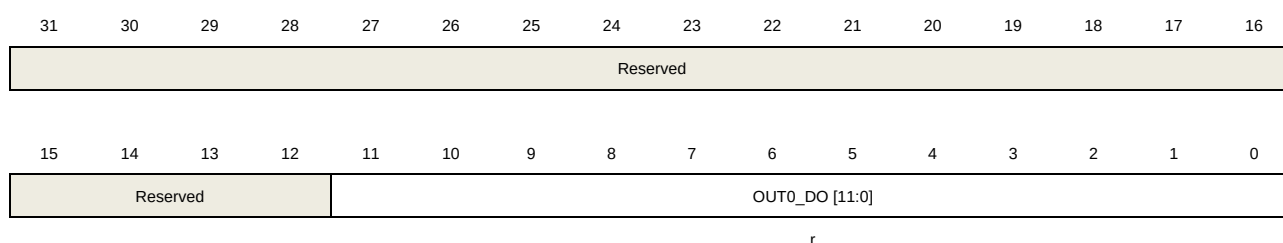
Bits	Fields	Descriptions
31:20	OUT1_DH[11:0]	DACx_OUT1 12-bit left-aligned data These bits specify the data that is to be converted by DACx_OUT1.
19:16	Reserved	Must be kept at reset value.
15:4	OUT0_DH[11:0]	DACx_OUT0 12-bit left-aligned data These bits specify the data that is to be converted by DACx_OUT0.
3:0	Reserved	Must be kept at reset value.

16.4.9 DACx_OUT0 data output register (DAC_OUT0_DO)

Address offset: 0x2C

Reset value: 0x0000 0000

This register has to be accessed by word(32-bit).



Bits	Fields	Descriptions
31:12	Reserved	Must be kept at reset value.
11:0	OUT0_DO [11:0]	DACx_OUT0 12-bit output data These bits, which are read only, storage the data that is being converted by

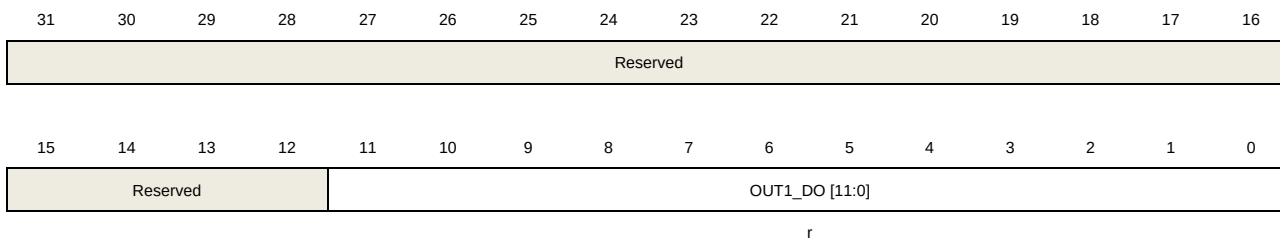
DACx_OUT0.

16.4.10 DACx_OUT1 data output register (DAC_OUT1_DO)

Address offset: 0x30

Reset value: 0x0000 0000

This register has to be accessed by word(32-bit).



Bits	Fields	Descriptions
31:12	Reserved	Must be kept at reset value.
11:0	OUT1_DO [11:0]	DACx_OUT1 12-bit output data These bits, which are read only, storage the data that is being converted by DACx_OUT1.

17 Comparator (CMP)

17.1 Overview

The general purpose comparators can work either standalone (all terminal are available on I/Os) or together with the timers.

It can be used to wake up the MCU from low-power mode by an analog signal, provide a trigger source when an analog signal is in a certain condition, achieve some current control by working together with a PWM output of a timer and the DAC.

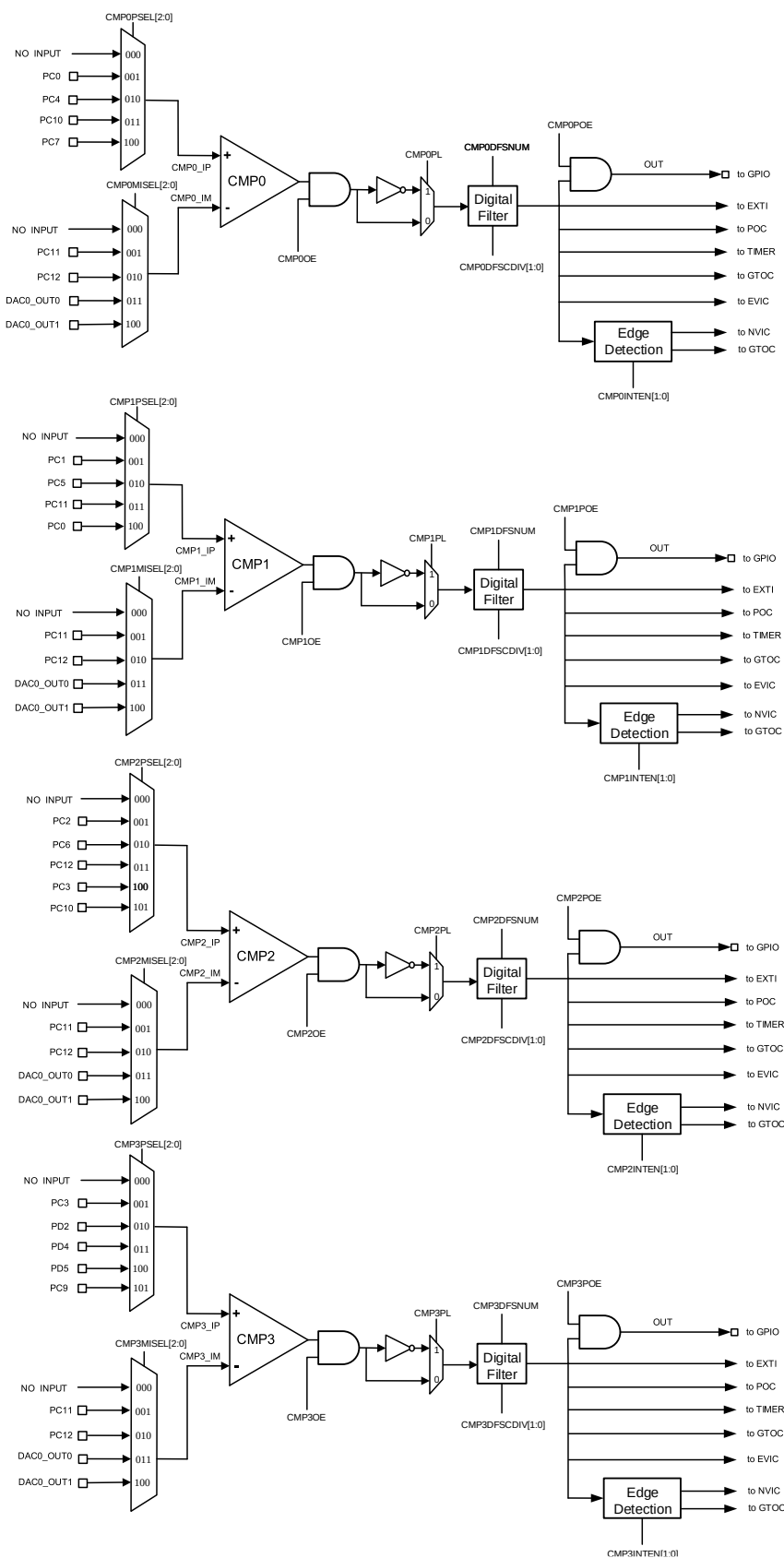
17.2 Characteristic

- Rail-to-rail comparators.
- Configurable hysteresis.
- Configurable analog input source.
 - DAC output.
 - Multiplexed I/O pins.
- Outputs to I/O.
- Outputs to timers for triggering.
- Outputs to EXTI.
- Outputs to NVIC.
- Outputs to POC.
- Outputs to GTOC.
- Outputs to EVIC.
- Digital noise filter.

17.3 Function overview

The block diagram of CMP is shown below:

Figure 17-1. CMP block diagram



17.3.1 CMP clock

The clock of the CMP which is connected to APB bus, is synchronous with PCLK.

17.3.2 CMP I/O configuration

These I/Os must be configured in analog mode in the GPIOs registers before they are selected as CMP inputs.

Considering pin definitions in datasheet, and the CMP output can be connected to corresponding alternate I/Os.

The CMP output can be redirected internally and externally simultaneously. The CMP output connected to polarity selection can be disabled or enabled by setting the CMPxOE bit in the CMPx_CS register. The CMP output connected to external pin can be disabled or enabled by setting the CMPxPOE bit in the CMPx_CS register.

CMP output internally connect to the TIMER and the connections between them are as follows:

- CMP output to the TIMER input channel.
- CMP output to the TIMER break.

In order to work even in deep-sleep mode, the polarity selection logic and the output redirection to the port work independently from PCLK.

[Table 17-1. CMP inputs and outputs summary](#) details the inputs and outputs of the CMP.

Table 17-1. CMP inputs and outputs summary

	CMP0	CMP1	CMP2	CMP3
CMP non inverting inputs connected to I/Os	PC0 PC4 PC10 PC7	PC1 PC5 PC11 PC0	PC2 PC6 PC12 PC3 PC10	PC3 PD2 PD4 PD5 PC9
CMP inverting inputs connected to I/Os	PC11 PC12			
CMP inverting inputs connected to internal signals	DAC0_OUT0 DAC0_OUT1			
CMP outputs connected to I/Os	PA0	PA1	PB2	PB1
CMP outputs connected to EXTI	•			
CMP outputs connected to NVIC	•			
CMP outputs connected to POC	•			

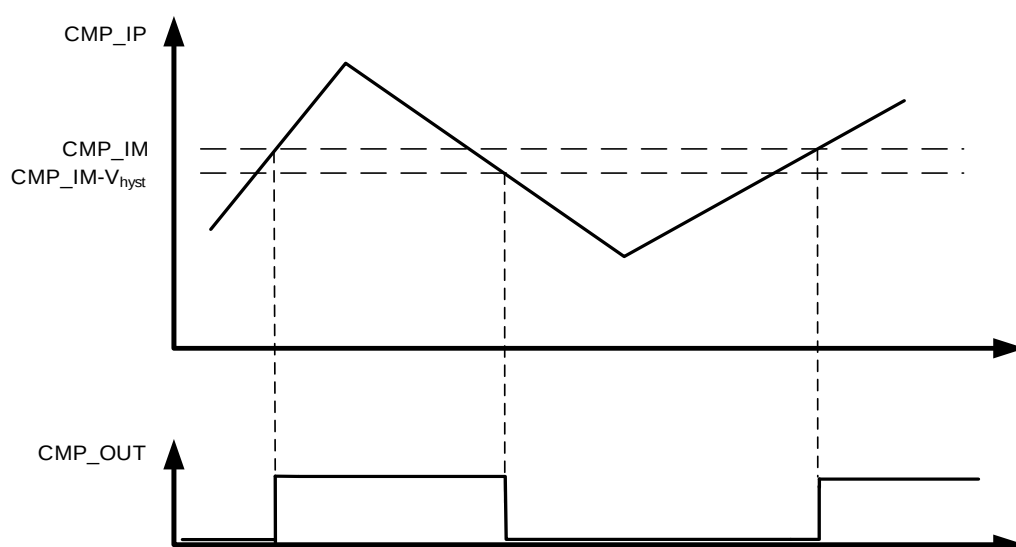
	CMP0	CMP1	CMP2	CMP3
CMP outputs connected to GTOC	•			
CMP outputs connected to EVIC	•			
CMP outputs connected to internal signals	TIMER0、TIMER1、TIMER2、TIMER7			
CMP outputs (motor control protection)	TIMER0 BREAK TIMER7 BREAK			

Note: Refer to the system configuration registers for details on CMP outputs connected to internal signals.

17.3.3 CMP hysteresis

In order to avoid spurious output transitions that caused by the noise signal, a programmable hysteresis is designed to force the hysteresis value by configuring CMPx_CS register. This function could be shut down if it is unnecessary.

Figure 17-2. CMP hysteresis



17.3.4 CMP register write protection

The CMP control and status register CMPx_CS can be protected from writing by setting CMPxLK bit to 1. The CMPx_CS register, including the CMPxLK bit will be read-only, and can only be reset by the MCU reset.

17.3.5 CMP interrupt

The CMP output is connected to the EXTI and the EXTI line is exclusive to CMP. With this function, CMP can generate either interrupt or event which could be used to exit from low-power mode.

The CMP also can generate an interrupt to NVIC. When CMP output a valid edge set by the CMPxINTEN[1:0] bits in the CMPx_CS register, an interrupt request will be generated.

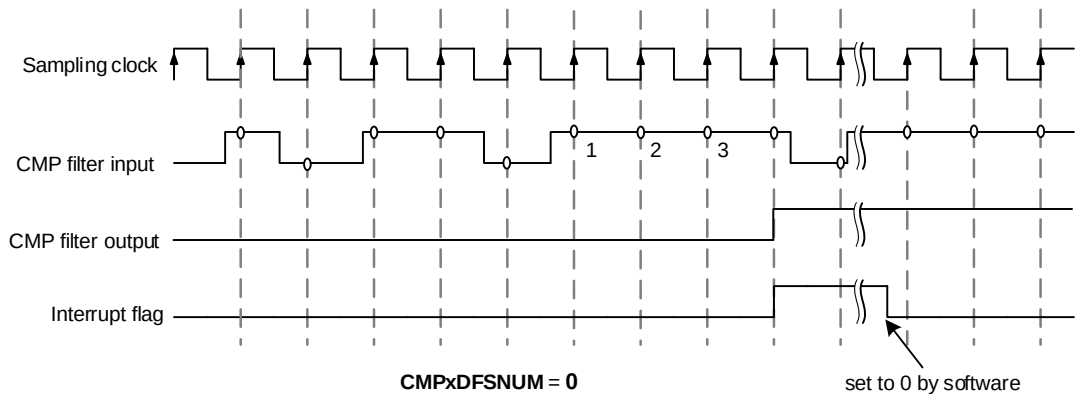
17.3.6 CMP digital filter

The sampling frequency of digital filter can be determined by the CMPxDFSCDIV[1:0] bits in the CMPx_CS register. The valid sampling times that the same level value is continuously sampled can be selected by the CMPxDFSNUM bit in the CMPx_CS register. The features of digital filter are as follows:

- CK_CMP/8, CK_CMP/16 or CK_CMP/32 can be selected as sampling clock.
- The filter function can also be disabled.
- The valid sampling times that the same level value can be three or four.

The CMP output signal will be sampled at every sampling clock, and when the same level value is sampled three or four times in succession, this level is identified as the noise-filtered valid output at the next sampling clock. [Figure 17-3. The CMP output signal filtering](#) shows details of digital filter output function.

Figure 17-3. The CMP output signal filtering



Note: The eliminated noise width can be selected by setting the CMPxDFSCDIV[1:0] bits and CMPxDFSNUM bit in the CMPx_CS register.

17.3.7 CMP low-power mode

Sleep mode

In sleep mode, the CMP can work normally, the device can exit the sleep mode by comparator interrupt/event.

Deep-sleep mode

In deep-sleep mode, the CMP can work normally, the device can exit the deep-sleep mode by EXTI interrupt/event.

Standby mode

In standby mode, the CMP registers are powered down, and must be reinitialized after exiting standby mode.

17.4 Register definition

CMP base address: 0x4001 7C00

17.4.1 CMP status register (CMP_STAT)

Address offset: 0x00

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved												CMP3IF	CMP2IF	CMP1IF	CMP0IF
												r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												CMP3O	CMP2O	CMP1O	CMP0O
												r	r	r	r

Bits	Fields	Descriptions
31:20	Reserved	Must be kept at reset value.
19	CMP3IF	CMP3 interrupt flag 0: No CMP3 output interrupt 1: CMP3 output interrupt Set by hardware when the CMP3 output is a valid edge. Cleared by software writing 1 to CMP3IC bit in the CMP_IFC register.
18	CMP2IF	CMP2 interrupt flag 0: No CMP2 output interrupt 1: CMP2 output interrupt Set by hardware when the CMP2 output is a valid edge. Cleared by software writing 1 to CMP2IC bit in the CMP_IFC register.
17	CMP1IF	CMP1 interrupt flag 0: No CMP1 output interrupt 1: CMP1 output interrupt Set by hardware when the CMP1 output is a valid edge. Cleared by software writing 1 to CMP1IC bit in the CMP_IFC register.
16	CMP0IF	CMP0 interrupt flag 0: No CMP0 output interrupt 1: CMP0 output interrupt Set by hardware when the CMP0 output is a valid edge. Cleared by software writing 1 to CMP0IC bit in the CMP_IFC register.
15:4	Reserved	Must be kept at reset value.

3	CMP3O	CMP3 output state This bit is a copy of CMP3 output state, which is read only. 0: Non-inverting input below inverting input and the output is low 1: Non-inverting input above inverting input and the output is high
2	CMP2O	CMP2 output state This bit is a copy of CMP2 output state, which is read only. 0: Non-inverting input below inverting input and the output is low 1: Non-inverting input above inverting input and the output is high
1	CMP1O	CMP1 output state This bit is a copy of CMP1 output state, which is read only. 0: Non-inverting input below inverting input and the output is low 1: Non-inverting input above inverting input and the output is high
0	CMP0O	CMP0 output state This bit is a copy of CMP0 output state, which is read only. 0: Non-inverting input below inverting input and the output is low 1: Non-inverting input above inverting input and the output is high

17.4.2 CMP interrupt flag clear register (CMP_IFC)

Address offset: 0x04

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved												CMP3IC	CMP2IC	CMP1IC	CMP0IC
												w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															

Bits	Fields	Descriptions
31:20	Reserved	Must be kept at reset value.
19	CMP3IC	CMP3 interrupt flag clear 0: Not clear CMP3 interrupt flag 1: Clear CMP3 interrupt flag
18	CMP2IC	CMP2 interrupt flag clear 0: Not clear CMP2 interrupt flag 1: Clear CMP2 interrupt flag
17	CMP1IC	CMP1 interrupt flag clear 0: Not clear CMP1 interrupt flag 1: Clear CMP1 interrupt flag

16	CMP0IC	CMP0 interrupt flag clear 0: Not clear CMP0 interrupt flag 1: Clear CMP0 interrupt flag

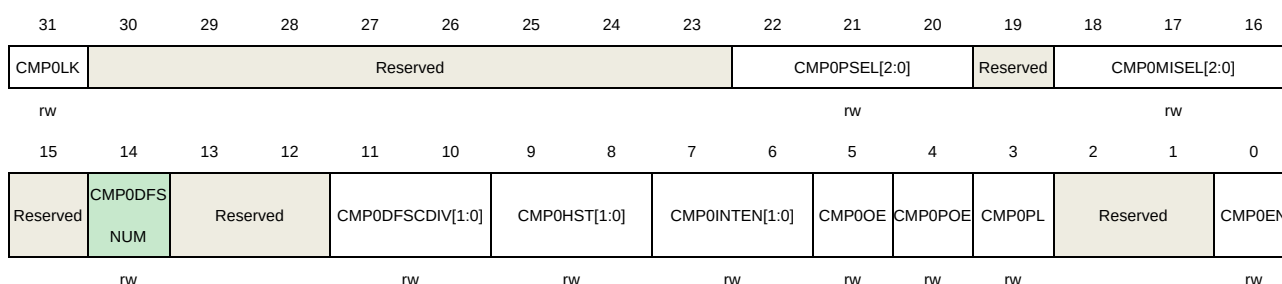
15:0	Reserved	Must be kept at reset value.

17.4.3 CMP0 control/status register (CMP0_CS)

Address offset: 0x0C

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



Bits	Fields	Descriptions
31	CMP0LK	CMP0 lock This bit allows to have all control bits of CMP0 as read-only. It can only be set once by software and cleared by a system reset. 0: CMP0_CS bits are read-write 1: CMP0_CS bits are read-only
30:23	Reserved	Must be kept at reset value.
22:20	CMP0PSEL[2:0]	CMP0_IP input selection These bits are used to select the source connected to the CMP0_IP input of the CMP0. 000: No input 001: PC0 010: PC4 011: PC10 100: PC7 All other values: reserved.
19	Reserved	Must be kept at reset value.
18:16	CMP0MISEL[2:0]	CMP0_IM internal input selection These bits are used to select the internal source connected to the CMP0_IM input of the CMP0. 000: No input 001: PC11

		010: PC12 011: DAC0_OUT0 100: DAC0_OUT1 All other values: reserved.
15	Reserved	Must be kept at reset value.
14	CMP0DFSNUM	CMP0 digital filter sampling number This bit is used to select valid sampling number of the CMP0 digital filter. 0: The same value is continuously sampled 3 times 1: The same value is continuously sampled 4 times
13:12	Reserved	Must be kept at reset value.
11:10	CMP0DFSCDIV[1:0]	CMP0 digital filter sampling clock division These bits are used to select the sampling frequency of digital filter. 00: Digital filter is not used 01: Sampling frequency is $f_{CK\_CMP} / 8$ 10: Sampling frequency is $f_{CK\_CMP} / 16$ 11: Sampling frequency is $f_{CK\_CMP} / 32$
9:8	CMP0HST[1:0]	CMP0 hysteresis These bits are used to control the hysteresis level. 00: No hysteresis 01: Low hysteresis 10: Medium hysteresis 11: High hysteresis
7:6	CMP0INTEN[1:0]	CMP0 interrupt request enable These bits are used to select valid edge of CMP0 output to activate an interrupt request. 00: Interrupt disabled 01: Rising edge 10: Falling edge 11: Both edge Note: These bits also are used to select valid edge for CMP0IF.
5	CMP0OE	CMP0 output enable 0: CMP0 output disabled 1: CMP0 output enabled
4	CMP0POE	CMP0 pin output enable 0: Output to CMP0 pin is disabled 1: Output to CMP0 pin is enabled
3	CMP0PL	Polarity of CMP0 output This bit is used to select the polarity of CMP0 output. 0: Output is not inverted

1: Output is inverted

2:1 Reserved Must be kept at reset value.

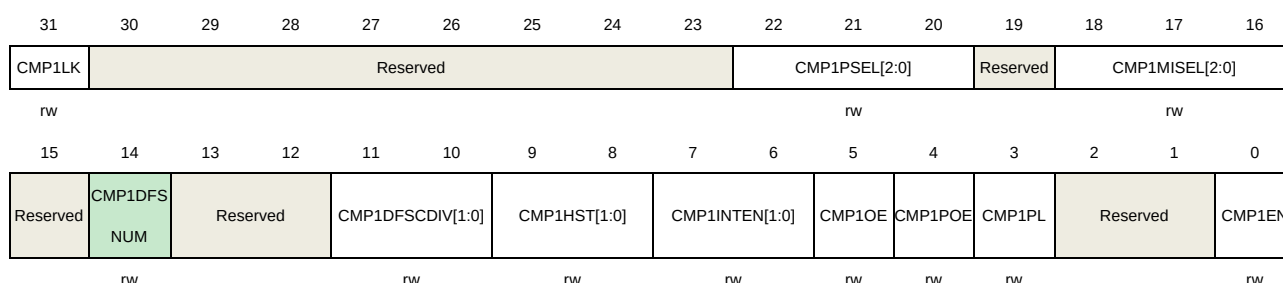
0 CMP0EN CMP0 enable
0: CMP0 disabled
1: CMP0 enabled

17.4.4 CMP1 control/status register (CMP1_CS)

Address offset: 0x10

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



Bits	Fields	Descriptions
31	CMP1LK	CMP1 lock This bit allows to have all control bits of CMP1 as read-only. It can only be set once by software and cleared by a system reset. 0: CMP1_CS bits are read-write 1: CMP1_CS bits are read-only
30:23	Reserved	Must be kept at reset value.
22:20	CMP1PSEL[2:0]	CMP1_IP input selection These bits are used to select the source connected to the CMP1_IP input of the CMP1. 000: No input 001: PC1 010: PC5 011: PC11 100: PC0 All other values: reserved.
19	Reserved	Must be kept at reset value.
18:16	CMP1MISEL[2:0]	CMP1_IM internal input selection These bits are used to select the internal source connected to the CMP1_IM input of the CMP1. 000: No input

		001: PC11 010: PC12 011: DAC0_OUT0 100: DAC0_OUT1 All other values: reserved.
15	Reserved	Must be kept at reset value.
14	CMP1DFSNUM	CMP1 digital filter sampling number This bit is used to select valid sampling number of the CMP1 digital filter. 0: The same value is continuously sampled 3 times 1: The same value is continuously sampled 4 times
13:12	Reserved	Must be kept at reset value.
11:10	CMP1DFSCDIV[1:0]	CMP1 digital filter sampling clock division These bits are used to select the sampling frequency of digital filter. 00: Digital filter is not used 01: Sampling frequency is $f_{CK\_CMP} / 8$ 10: Sampling frequency is $f_{CK\_CMP} / 16$ 11: Sampling frequency is $f_{CK\_CMP} / 32$
9:8	CMP1HST[1:0]	CMP1 hysteresis These bits are used to control the hysteresis level. 00: No hysteresis 01: Low hysteresis 10: Medium hysteresis 11: High hysteresis
7:6	CMP1INTEN[1:0]	CMP1 interrupt request enable These bits are used to select valid edge of CMP1 output to activate an interrupt request. 00: Interrupt disabled 01: Rising edge 10: Falling edge 11: Both edge Note: These bits also are used to select valid edge for CMP1IF.
5	CMP1OE	CMP1 output enable 0: CMP1 output disabled 1: CMP1 output enabled
4	CMP1POE	CMP1 pin output enable 0: Output to CMP1 pin is disabled 1: Output to CMP1 pin is enabled
3	CMP1PL	Polarity of CMP1 output This bit is used to select the polarity of CMP1 output.

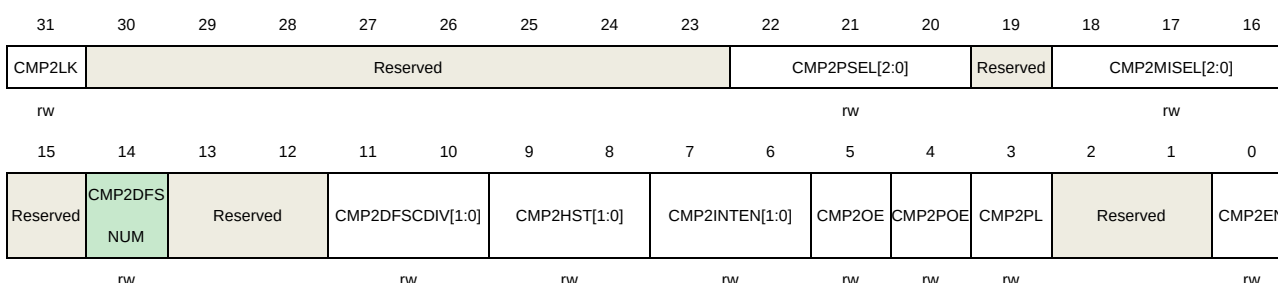
		0: Output is not inverted 1: Output is inverted
2:1	Reserved	Must be kept at reset value.
0	CMP1EN	CMP1 enable 0: CMP1 disabled 1: CMP1 enabled

17.4.5 CMP2 control/status register (CMP2_CS)

Address offset: 0x14

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



Bits	Fields	Descriptions
31	CMP2LK	CMP2 lock This bit allows to have all control bits of CMP2 as read-only. It can only be set once by software and cleared by a system reset. 0: CMP2_CS bits are read-write 1: CMP2_CS bits are read-only
30:23	Reserved	Must be kept at reset value.
22:20	CMP2PSEL[2:0]	CMP2_IP input selection These bits are used to select the source connected to the CMP2_IP input of the CMP2. 000: No input 001: PC2 010: PC6 011: PC12 100: PC3 101: PC10 All other values: reserved.
19	Reserved	Must be kept at reset value.
18:16	CMP2MISEL[2:0]	CMP2_IM internal input selection These bits are used to select the internal source connected to the CMP2_IM input

		of the CMP2. 000: No input 001: PC11 010: PC12 011: DAC0_OUT0 100: DAC0_OUT1 All other values: reserved.
15	Reserved	Must be kept at reset value.
14	CMP2DFSNUM	CMP2 digital filter sampling number This bit is used to select valid sampling number of the CMP2 digital filter. 0: The same value is continuously sampled 3 times 1: The same value is continuously sampled 4 times
13:12	Reserved	Must be kept at reset value.
11:10	CMP2DFSCDIV[1:0]	CMP2 digital filter sampling clock division These bits are used to select the sampling frequency of digital filter. 00: Digital filter is not used 01: Sampling frequency is $f_{CK\_CMP} / 8$ 10: Sampling frequency is $f_{CK\_CMP} / 16$ 11: Sampling frequency is $f_{CK\_CMP} / 32$
9:8	CMP2HST[1:0]	CMP2 hysteresis These bits are used to control the hysteresis level. 00: No hysteresis 01: Low hysteresis 10: Medium hysteresis 11: High hysteresis
7:6	CMP2INTEN[1:0]	CMP2 interrupt request enable These bits are used to select valid edge of CMP2 output to activate an interrupt request. 00: Interrupt disabled 01: Rising edge 10: Falling edge 11: Both edge Note: These bits also are used to select valid edge for CMP2IF.
5	CMP2OE	CMP2 output enable 0: CMP2 output disabled 1: CMP2 output enabled
4	CMP2POE	CMP2 pin output enable 0: Output to CMP2 pin is disabled 1: Output to CMP2 pin is enabled

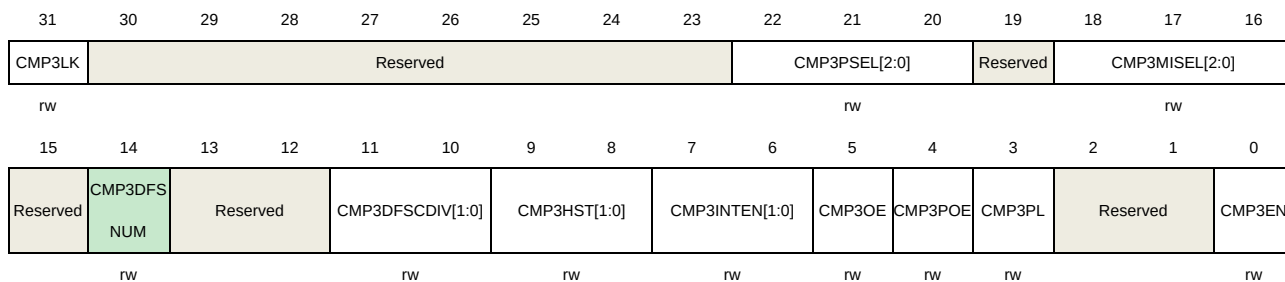
3	CMP2PL	Polarity of CMP2 output This bit is used to select the polarity of CMP2 output. 0: Output is not inverted 1: Output is inverted
2:1	Reserved	Must be kept at reset value.
0	CMP2EN	CMP2 enable 0: CMP2 disabled 1: CMP2 enabled

17.4.6 CMP3 control/status register (CMP3_CS)

Address offset: 0x18

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



Bits	Fields	Descriptions
31	CMP3LK	CMP3 lock This bit allows to have all control bits of CMP3 as read-only. It can only be set once by software and cleared by a system reset. 0: CMP3_CS bits are read-write 1: CMP3_CS bits are read-only
30:23	Reserved	Must be kept at reset value.
22:20	CMP3PSEL[2:0]	CMP3_IP input selection These bits are used to select the source connected to the CMP3_IP input of the CMP3. 000: No input 001: PC3 010: PD2 011: PD4 100: PD5 101: PC9 All other values: reserved.
19	Reserved	Must be kept at reset value.

18:16	CMP3MISEL[2:0]	CMP3_IM internal input selection These bits are used to select the internal source connected to the CMP3_IM input of the CMP3. 000: No input 001: PC11 010: PC12 011: DAC0_OUT0 100: DAC0_OUT1 All other values: reserved.
15	Reserved	Must be kept at reset value.
14	CMP3DFSNUM	CMP3 digital filter sampling number This bit is used to select valid sampling number of the CMP3 digital filter. 0: The same value is continuously sampled 3 times 1: The same value is continuously sampled 4 times
13:12	Reserved	Must be kept at reset value.
11:10	CMP3DFSCDIV[1:0]	CMP3 digital filter sampling clock division These bits are used to select the sampling frequency of digital filter. 00: Digital filter is not used 01: Sampling frequency is $f_{CK\_CMP} / 8$ 10: Sampling frequency is $f_{CK\_CMP} / 16$ 11: Sampling frequency is $f_{CK\_CMP} / 32$
9:8	CMP3HST[1:0]	CMP3 hysteresis These bits are used to control the hysteresis level. 00: No hysteresis 01: Low hysteresis 10: Medium hysteresis 11: High hysteresis
7:6	CMP3INTEN[1:0]	CMP3 interrupt request enable These bits are used to select valid edge of CMP3 output to activate an interrupt request. 00: Interrupt disabled 01: Rising edge 10: Falling edge 11: Both edge Note: These bits also are used to select valid edge for CMP3IF.
5	CMP3OE	CMP3 output enable 0: CMP3 output disabled 1: CMP3 output enabled
4	CMP3POE	CMP3 pin output enable

		0: Output to CMP3 pin is disabled 1: Output to CMP3 pin is enabled
3	CMP3PL	Polarity of CMP3 output This bit is used to select the polarity of CMP3 output. 0: Output is not inverted 1: Output is inverted
2:1	Reserved	Must be kept at reset value.
0	CMP3EN	CMP3 enable 0: CMP3 disabled 1: CMP3 enabled

18 Event interconnection unit (EVIC)

18.1 Overview

The event interconnection unit (EVIC) allows software to select event signals generated by various peripheral modules for a variety of applications. EVIC provides a flexible mechanism for a peripheral to select different event source. Therefore, the different modules are interconnected to each other through EVIC.

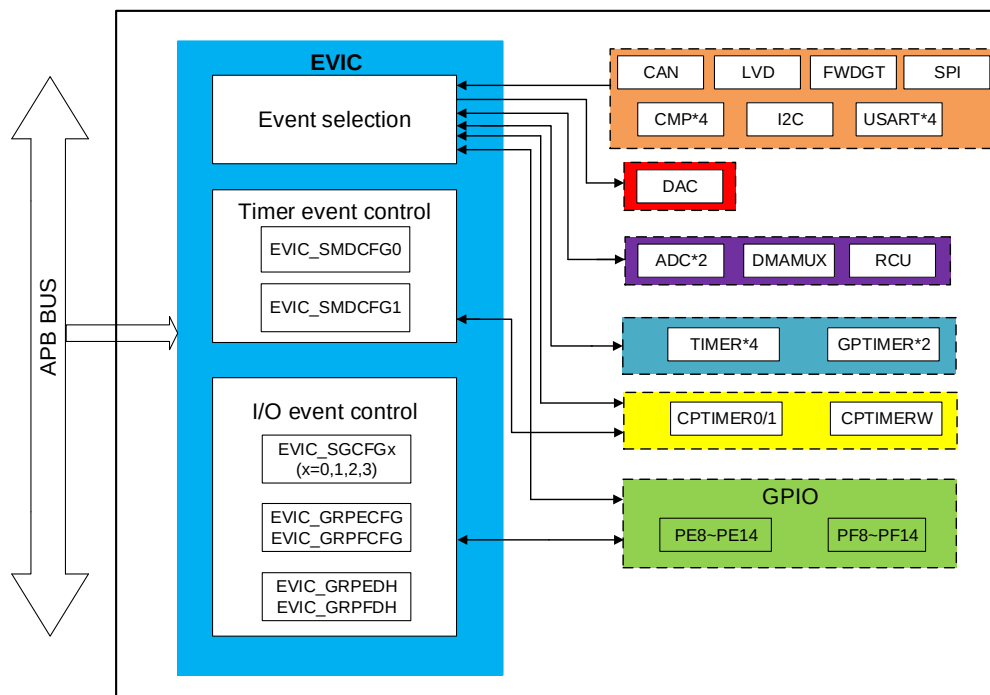
18.2 Characteristics

- Supports different optional event source;
- Peripheral has its own register to select event source;
- Event signal can be generated by external pin input detection or output of other peripherals;
- Event trigger operation for I/O pins status.

18.3 Function overview

The block diagram of EVIC is shown below:

Figure 18-1. EVIC block diagram



18.3.1 Event source selection

The [Table 18-1. Event source bit fields selection](#) gives the event signal from different peripherals.

Table 18-1. Event source bit fields selection

fields	Bits value	Event source selection
EVSELx	0x00	Event signal output is disabled.
	0x01	Software generation event
	0x02	Single I/O input detection 0
	0x03	Single I/O input detection 1
	0x04	Single I/O input detection 2
	0x05	Single I/O input detection 3
	0x06	Group E I/O input detection
	0x07	Group F I/O input detection
	0x08	TIMER0 overflow event
	0x09	TIMER0 underflow event
	0x0a	TIMER0_CH0 compare event
	0x0b	TIMER0_CH1 compare event
	0x0c	TIMER0_CH2 compare event
	0x0d	TIMER0_CH3 compare event
	0x0e	TIMER0_MCH0 compare event
	0x0f	TIMER0_MCH1 compare event
	0x10	TIMER0_MCH2 compare event
	0x11	TIMER0_MCH3 compare event
	0x12	TIMER0 TRGO signal
	0x13	TIMER7 overflow event
	0x14	TIMER7 underflow event
	0x15	TIMER7_CH0 compare event
	0x16	TIMER7_CH1 compare event
	0x17	TIMER7_CH2 compare event
	0x18	TIMER7_CH3 compare event
	0x19	TIMER7_MCH0 compare event
	0x1a	TIMER7_MCH1 compare event
	0x1b	TIMER7_MCH2 compare event
	0x1c	TIMER7_MCH3 compare event
	0x1d	TIMER7 TRGO signal
	0x1e	TIMER1_CH0 compare event
	0x1f	TIMER1_CH1 compare event
	0x20	TIMER1_CH2 compare event
	0x21	TIMER1_CH3 compare event
	0x22	TIMER1 overflow event
	0x23	TIMER1 underflow event

0x24	TIMER1 TRGO signal
0x25	TIMER2_CH0 compare event
0x26	TIMER2_CH1 compare event
0x27	TIMER2_CH2 compare event
0x28	TIMER2_CH3 compare event
0x29	TIMER2 overflow event
0x2a	TIMER2 underflow event
0x2b	TIMER2 TRGO signal
0x2c	CPTIMER0 counter 0 overflow event
0x2d	Reserved
0x2e	CPTIMERW_OCH2 compare match event
0x2f	CPTIMERW_OCH3 compare match event
0x30	CPTIMERW_OCH0 compare match event
0x31	CPTIMERW_OCH1 compare match event
0x32	Reserved
0x33	Reserved
0x34	Reserved
0x35	GPTIMER0_CH0 compare match event
0x36	GPTIMER0_CH1 compare match event
0x37	GPTIMER0_CH0 addition compare match event
0x38	GPTIMER0_CH1 addition compare match event
0x39	GPTIMER0 overflow event
0x3a	GPTIMER0 underflow event
0x3b	GPTIMER0 ADTCV1 match event
0x3c	GPTIMER0 ADTCV2 match event
0x3d	GPTIMER0 end of repeat count event
0x3e	Reserved
0x3f	Reserved
0x40	CMP0 output
0x41	CMP1 output
0x44	CMP2 output
0x43	CMP3 output
0x44	ADC0 group scan complete event
0x45	ADC2 group scan complete event
0x46	LVD1 voltage detection
0x47	LVD2 voltage detection
0x48	FWDGT underflow or refresh
0x49	HXTAL failure event
0x4a	DMA0 channel 0 full transfer finish event
0x4b	DMA0 channel 1 full transfer finish event
0x4c	DMA0 channel 2 full transfer finish event
0x4d	DMA0 channel 3 full transfer finish event

0x4e	DMA0 channel 4 full transfer finish event
0x4f	DMA0 channel 5 full transfer finish event
0x50	DMA1 channel 0 full transfer finish event
0x51	DMA1 channel 1 full transfer finish event
0x52	DMA1 channel 2 full transfer finish event
0x53	DMA1 channel 3 full transfer finish event
0x54	DMA1 channel 4 full transfer finish event
0x55	DMA1 channel 5 full transfer finish event
0x56	CAN transmit interrupt event
0x57	CAN FIFO0 interrupt event
0x58	CAN FIFO1 interrupt event
0x59	CAN EWMC interrupt event
0x5a	Reserved
0x5b	Reserved
0x5c	UART0 error (PERR /FERR /NERR /ORERR)
0x5d	UART0 transmission complete
0x5e	UART1 error (PERR /FERR /NERR /ORERR)
0x5f	UART1 transmission complete
0x60	UART2 error(PERR /FERR /NERR /ORERR)
0x61	UART2 transmission complete
0x65	UART3 error (PERR /FERR /NERR /ORERR)
0x63	UART3 transmission complete
0x65	Reserved
0x65	Reserved
0x66	Reserved
0x67	SPI error
0x68	SPI transfer complete event
0x69	I2C communication error/communication event
0x6a	I2C_RDATA is not empty during receiving
0x6b	I2C transmit interrupt event
0x6c	I2C transfer complete event
0x6d	GPTIMER1_CH0 compare match event
0x6e	GPTIMER1_CH1 compare match event
0x6f	GPTIMER1_CH0 addition compare match event
0x70	GPTIMER1_CH1 addition compare match event
0x71	GPTIMER1 overflow event
0x72	GPTIMER1 underflow event
0x73	GPTIMER1 ADTCV1 match event
0x74	GPTIMER1 ADTCV2 match event
0x75	GPTIMER1 end of repeat count event

18.3.2 Event interconnect for I/O pin

The I/O pin with event interconnect function can generate event signal and subject to event trigger control. These I/O pins can be selected from Port E or Port F.

Single I/O and group I/O

The I/O pins that are used to set an event interconnect can divide into two types: single I/O pin and group I/O pin.

For the former, a single I/O pin is specified by SGIOPORT[1:0] bits and SGIOPIN[2:0] bits of EVIC_SGIOCFGx (x= 0...3) register. The selected single I/O pin can generate events by input detection and select an event signal to trigger the pin output corresponding level.

For the latter, two or more I/O pins from the same port can form a group. The member of the group is called as group I/O pin. Members of the same group can only come from the same GPIO port (port E or port F), so they can be divided into group E and group F. Moreover, based on whether the group members are input or output pins, group E/F can be further divided into input group E/F and output group E/F (Input and output are determined by the GPIO module register). Members of the group E and group F can be determined by configuring EVIC_GRPFCFG and EVIC_GRPFCFG registers. Each group has separate registers for setting input detection condition to generate events. Each group also has separate registers for selecting event signal to trigger pin output and pin input level loading.

If a pin is configured as both a single I/O pin and a group I/O pin, both single I/O and group I/O register can determine the event generation operation, but event trigger operation for the pin is only controlled by group I/O configuration register.

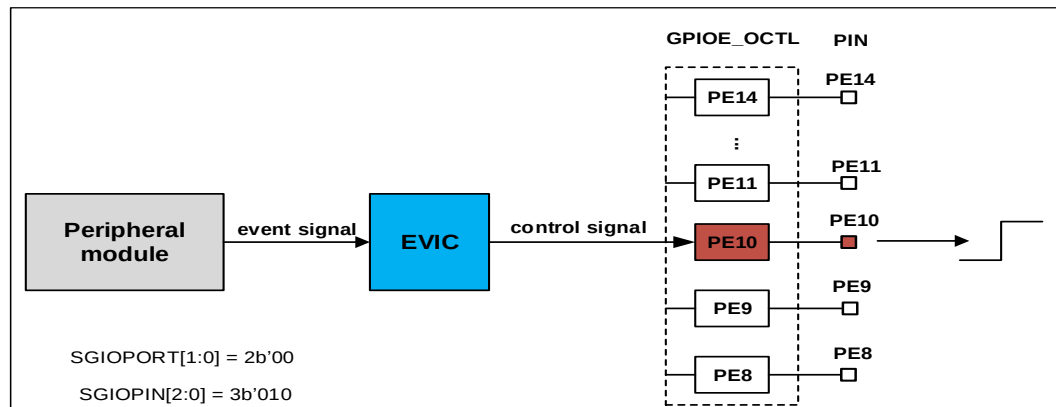
Event generation by single I/O pin

EVIC provides four single I/O input detectors, and can detect up to four single I/O inputs simultaneously. When single I/O pin that selected by EVIC_SGIOCFGx (x= 0...3) register is set as input mode, if a valid edge is detected in the pin input signal, an event signal will be generated. The valid edge can be set by SGIOM[1:0] bits of EVIC_SGIOCFGx (x= 0...3) register.

Single I/O pin output by event trigger

EVIC provides four single I/O controller, and can control up to four single I/O outputs simultaneously. When single I/O pin that specified by EVIC_SGIOCFGx (x= 0...3) register is set as output mode, if the event selected by EVIC_SGIOx (x= 0...3) register occurs, the single I/O pin will output corresponding level as configured by the SGIOM[1:0] bits in EVIC_SGIOCFGx (x= 0...3) register.

Figure 18-2. Example of single I/O pin output



Event generation by input group I/O pin

EVIC provide group E and group F I/O input detector. An event signal is generated by input group when a valid edge of any one of input group I/O pin is detected. The valid edge of input group E or input group F is specified using the GRPEIND[1:0] bits in EVIC_GRPECFG register or GRPFIND[1:0] bits in EVIC_GRPFCFG register.

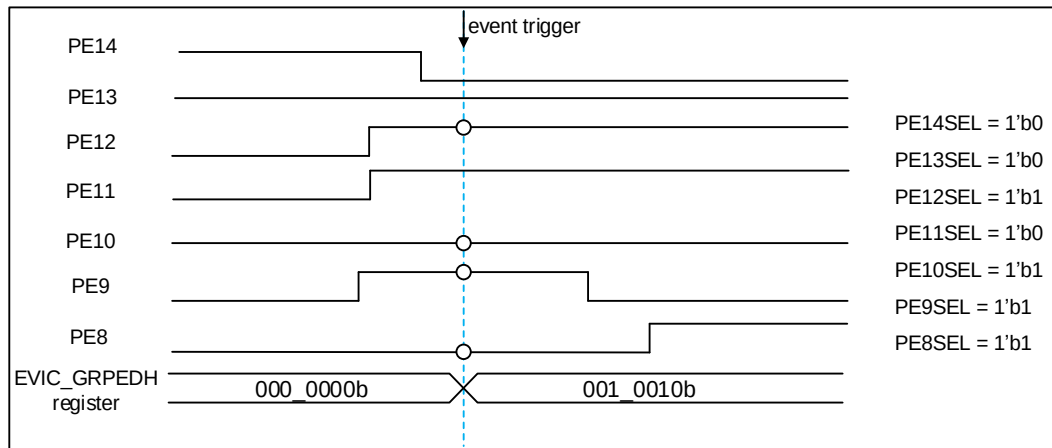
Group I/O pin input loading by event trigger

EVIC provide group E and group F I/O input controller. When the source selected by EVIC INGRPE or EVIC INGRPF register generates event signal, the level of input group I/O pins can be loaded into the corresponding bit of EVIC_GRPEDH or EVIC_GRPFDH register. Values of these bits that do not correspond to input group I/O pins do not change.

When another event signal arrives, if overwriting function is enabled (OVWEN=1'b1 in EVIC_GRPECFG / EVIC_GRPFCFG register), the current level of the input group I/O pins is loaded into the data holding register (EVIC_GRPEDH/EVIC_GRPFDH) to overwrite the previous level value.

if overwriting function is disabled (OVWEN=1'b0 in EVIC_GRPECFG / EVIC_GRPFCFG register), the arrival of another event signal does not overwrite the value of the data holding register unless the previous loaded value has been read. If the previous loaded value is not read, the arrival of the event signal is ignored.

Figure 18-3. Example of group E I/O pin input loading

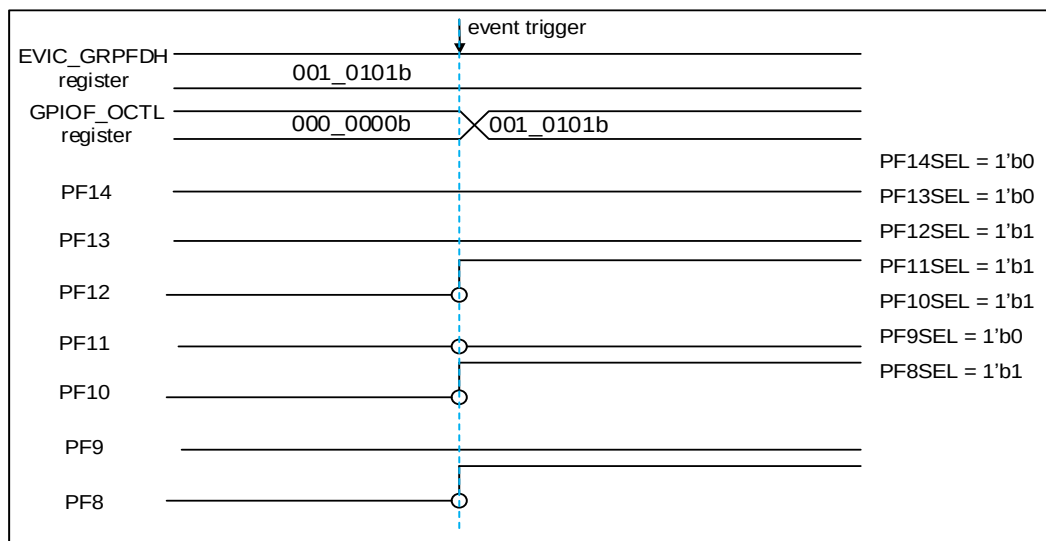


Group I/O pin output by event trigger

EVIC provide group E and group F I/O output controller. When the source selected by EVIC_OUTEGRP or EVIC_OUTFGRP register generates event signal, output group I/O pin will change to corresponding level set by the GRPEOSEL[2:0] bits in EVIC_GRPFCFG register or the GRPFOSEL[2:0] bits in EVIC_GRPFCFG register.

If the GRPEOSEL[2:0] bits or the GRPFOSEL[2:0] bits is 3b'011, when the selected event occurs, the EVIC_GRPEDH or EVIC_GRPFDH register bits value that corresponds only to output group I/O pin will pass into the GPIOE_OCTL or GPIOF_OCTL register corresponding bits, so I/O pins in output group E or F will regard the EVIC_GRPEDH or EVIC_GRPFDH register corresponding bit values as output level at the arrival of event signal.

Figure 18-4. Example of group I/O pin output (GRPEOSEL[2:0] = 3b'011)

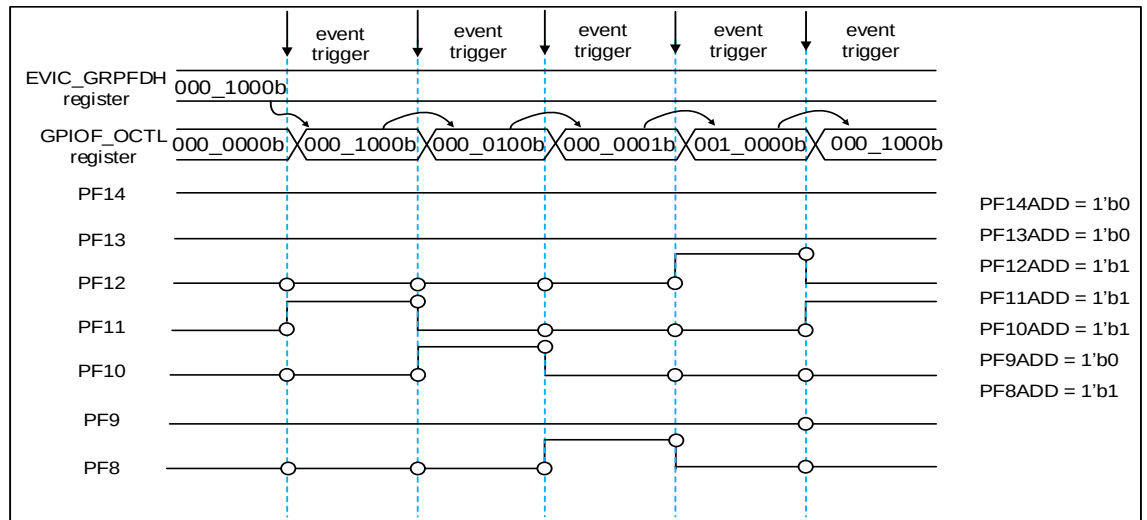


If the GRPEOSEL[2:0] bits or the GRPFOSEL[2:0] bits is 3b'1xx, when an event signal first arrives, the data holding register (EVIC_GRPEDH or EVIC_GRPFDH register) value will transfer to GPIOE_OCTL or GPIOF_OCTL register.

After this, every time an event comes, the value of the GPIOE_OCTL or GPIOF_OCTL register is circularly shifted one bit from MSB to LSB (only the register bit corresponding to the output group I/O pin is shifted).

If a new value is written to the data holding register (EVIC_GRPEDH or EVIC_GRPFDH register) during the shift, the next event trigger will not shift the GPIOE_OCTL or GPIOF_OCTL register but transfer the new value to the GPIOE_OCTL or GPIOF_OCTL register.

Figure 18-5. group output port rotate mode



Note:

1. It is disabled to write to the data holding register (EVIC_GRPEDH or EVIC_GRPFDH register) bits corresponding to the input group I/O pins.
2. If an I/O pin is selected as the output group I/O pin or single I/O output pin, it is disabled to write to the GPIOx_OCTL register bits corresponding to the I/O pin.

18.4 Register definition

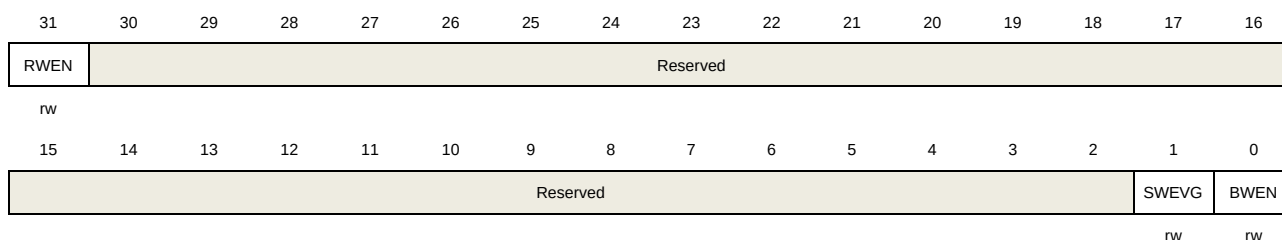
EVIC base address: 0x4001 8400

18.4.1 Software event register (EVIC_SWEV)

Address offset: 0x00

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



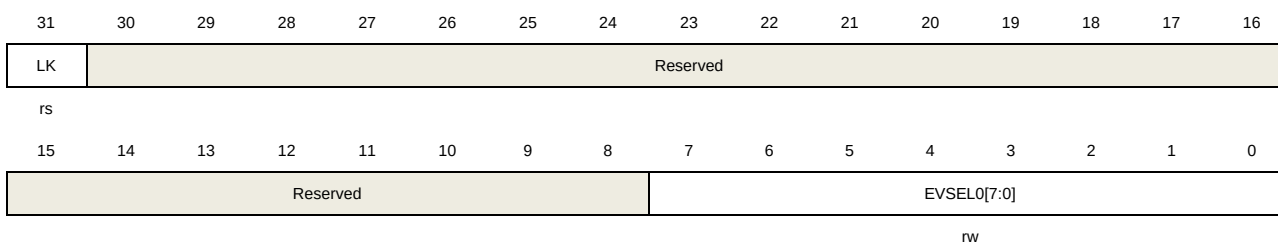
Bits	Fields	Descriptions
31	RWEN	EVIC_SWEV register write enable This bit is set or cleared by software. When it is set to 1, EVIC_SWEV register can be written. 0: EVIC_SWEV register write is disabled. 1: EVIC_SWEV register write is enabled.
30:2	Reserved	Must kept at reset value.
1	SWEVG	Software event generation This bit is set by software to generate an event signal, and it is automatically cleared by hardware after software event generation. 0: No generate a software event. 1: Generate a software event. Note: This bit can be written when BWEN bit and RWEN bit are both 1.
0	BWEN	SWEVG bit write enable This bit is used to enable write access to SWEVG bit. When it is set to 1, SWEVG bit can be written. 0: SWEVG bit write is disabled. 1: SWEVG bit write is enabled. Note: This bit is written when RWEN bit are 1.

18.4.2 Event interconnect for single I/O register x (EVIC_SGIOx) (x=0...3)

Address offset: 0x04 + 0x04 * x

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



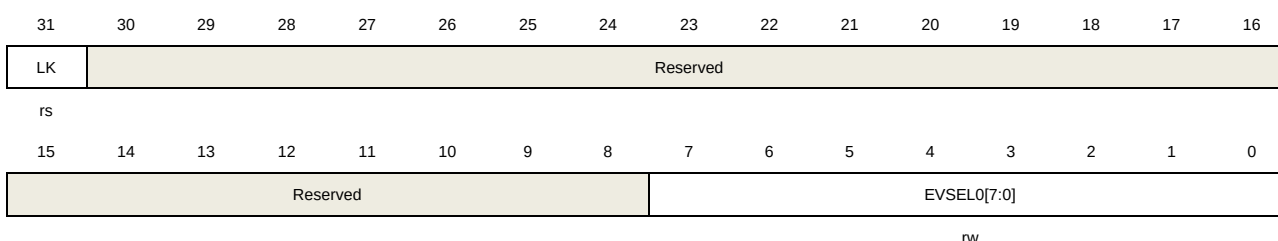
Bits	Fields	Descriptions
31	LK	EVIC_SGIOx register lock. This bit is set by software and cleared only by a system reset. When it is set to 1, EVIC_SGIOx register is read-only. 0: EVIC_SGIOx register is read-write. 1: EVIC_SGIOx register is read-only.
30:8	Reserved	Must kept at reset value.
7:0	EVSEL0[7:0]	Event source selection for single I/O These bits are used to select an event source connected to single I/O output controller x. When the selected event occurs, the single I/O pin selected by EVIC_SGCFGx register will output the specific level. For the detailed event source, please refer to Table 18-1. Event source bit fields selection.

18.4.3 Event interconnect for input group E register (EVIC INGRPE)

Address offset: 0x14

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



Bits	Fields	Descriptions
31	LK	EVIC INGRPE register lock. This bit is set by software and cleared only by a system reset. When it is set to 1, EVIC INGRPE register is read-only. 0: EVIC INGRPE register is read-write. 1: EVIC INGRPE register is read-only.

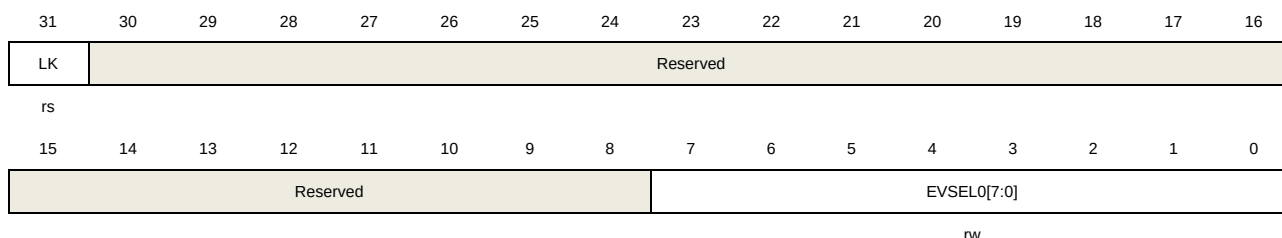
30:8	Reserved	Must kept at reset value.
7:0	EVSEL0[7:0]	Event source selection for group E I/O input These bits are used to select an event source connected to group E I/O input controller. When the selected event occurs, the input level of the group E I/O pins will be loaded to the corresponding bit of the EVIC_GRPEDH register. For the detailed event source, please refer to Table 18-1. Event source bit fields selection.

18.4.4 Event interconnect for input group F register (EVIC INGRPF)

Address offset: 0x18

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



Bits	Fields	Descriptions
31	LK	EVIC INGRPF register lock. This bit is set by software and cleared only by a system reset. When it is set to 1, EVIC INGRPF register is read-only. 0: EVIC INGRPF register is read-write. 1: EVIC INGRPF register is read-only.
30:8	Reserved	Must kept at reset value.
7:0	EVSEL0[7:0]	Event source selection for group F I/O input These bits are used to select an event source connected to group F I/O input controller. When the selected event occurs, the input level of the group F I/O pins will be loaded to the corresponding bit of the EVIC_GRPFDH register. For the detailed event source, please refer to Table 18-1. Event source bit fields selection.

18.4.5 Event interconnect for output group E register (EVIC OUTGRPE)

Address offset: 0x1C

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LK	Reserved														
rs															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								EVSEL0[7:0]							
rw															

Bits	Fields	Descriptions
31	LK	EVIC_OUTGRPE register lock. This bit is set by software and cleared only by a system reset. When it is set to 1, EVIC_OUTGRPE register is read-only. 0: EVIC_OUTGRPE register is read-write. 1: EVIC_OUTGRPE register is read-only.
30:8	Reserved	Must kept at reset value.
7:0	EVSEL0[7:0]	Event source selection for group E I/O output These bits are used to select an event source connected to group E I/O output controller. When the selected event occurs, the group E I/O pins will output the specific level. For the detailed event source, please refer to Table 18-1. Event source bit fields selection.

18.4.6 Event interconnect for output group F register (EVIC_OUTGRPF)

Address offset: 0x20

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LK	Reserved														
rs															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								EVSEL0[7:0]							
rw															

Bits	Fields	Descriptions
31	LK	EVIC_OUTGRPF register lock. This bit is set by software and cleared only by a system reset. When it is set to 1, EVIC_OUTGRPF register is read-only. 0: EVIC_OUTGRPF register is read-write. 1: EVIC_OUTGRPF register is read-only.
30:8	Reserved	Must kept at reset value.
7:0	EVSEL0[7:0]	Event source selection for group F I/O output

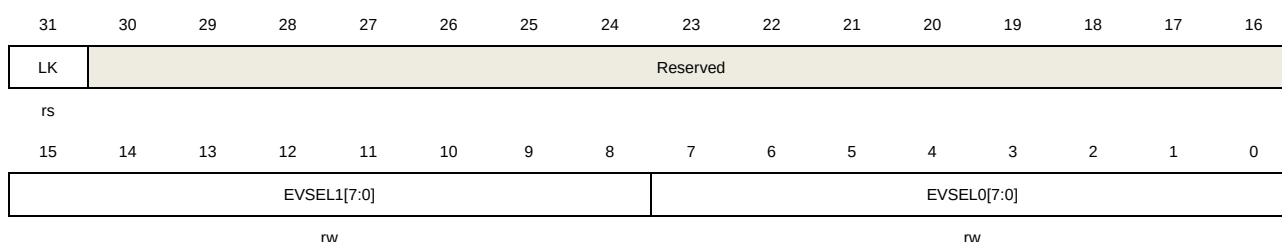
These bits are used to select an event source connected to group F I/O output controller. When the selected event occurs, the group F I/O pins will output the specific level. For the detailed event source, please refer to [Table 18-1. Event source bit fields selection](#).

18.4.7 Event interconnect for DAC0 conversion register (EVIC_DAC0COV)

Address offset: 0x24

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



Bits	Fields	Descriptions
31	LK	EVIC_DAC0COV register lock. This bit is set by software and cleared only by a system reset. When it is set to 1, EVIC_DAC0COV register is read-only. 0: EVIC_DAC0COV register is read-write. 1: EVIC_DAC0COV register is read-only.
30:16	Reserved	Must kept at reset value.
15:8	EVSEL1[7:0]	Event source selection for DAC0_OUT1 conversion These bits are used to select an event source connected to the external trigger of DAC0_OUT1 conversion. For the detailed event source, please refer to Table 18-1. Event source bit fields selection .
7:0	EVSEL0[7:0]	Event source selection for DAC0_OUT0 conversion These bits are used to select an event signal connected to the external trigger of DAC0_OUT0 conversion. For the detailed event source, please refer to Table 18-1. Event source bit fields selection .

18.4.8 Event interconnect for ADC0 conversion register (EVIC_ADC0COV)

Address offset: 0x28

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



LK	Reserved														
rs															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EVSEL1[7:0]								EVSEL0[7:0]							
rw								rw							

Bits	Fields	Descriptions
31	LK	EVIC_ADC0COV register lock. This bit is set by software and cleared only by a system reset. When it is set to 1, EVIC_ADC0COV register is read-only. 0: EVIC_ADC0COV register is read-write. 1: EVIC_ADC0COV register is read-only.
30:16	Reserved	Must kept at reset value.
15:8	EVSEL1[7:0]	Event source selection 1 for ADC0 group conversion These bits are used to select an event source connected to the external trigger 1 of ADC0 group conversion. For the detailed event source, please refer to Table 18-1. Event source bit fields selection .
7:0	EVSEL0[7:0]	Event source selection 0 for ADC0 group conversion These bits are used to select an event source connected to the external trigger 0 of ADC0 group conversion. For the detailed event source, please refer to Table 18-1. Event source bit fields selection .

18.4.9 Event interconnect for ADC2 conversion register (EVIC_ADC2COV)

Address offset: 0x2C

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LK	Reserved														
rs															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EVSEL1[7:0]								EVSEL0[7:0]							
rw								rw							

Bits	Fields	Descriptions
31	LK	EVIC_ADC2COV register lock. This bit is set by software and cleared only by a system reset. When it is set to 1, EVIC_ADC2COV register is read-only. 0: EVIC_ADC2COV register is read-write. 1: EVIC_ADC2COV register is read-only.

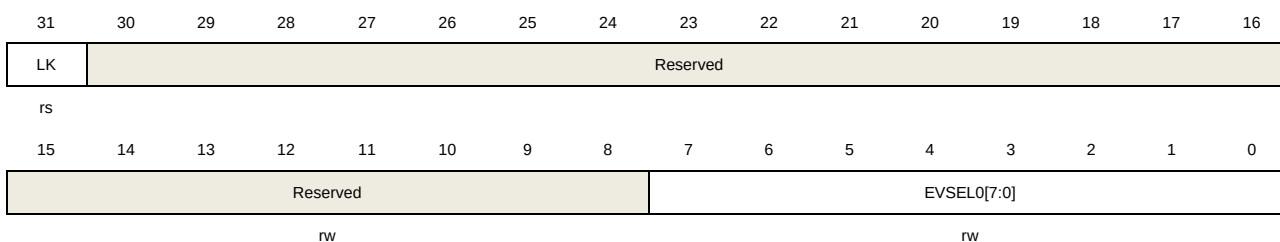
30:16	Reserved	Must kept at reset value.
15:8	EVSEL1[7:0]	Event source selection 1 for ADC2 group conversion These bits are used to select an event source connected to the external trigger 1 of ADC2 group conversion. For the detailed event source, please refer to Table 18-1. Event source bit fields selection .
7:0	EVSEL0[7:0]	Event source selection 0 for ADC2 group conversion These bits are used to select an event source connected to the external trigger 0 of ADC2 group conversion. For the detailed event source, please refer to Table 18-1. Event source bit fields selection .

18.4.10 Event interconnect for RCU register (EVIC_RCU)

Address offset: 0x34

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



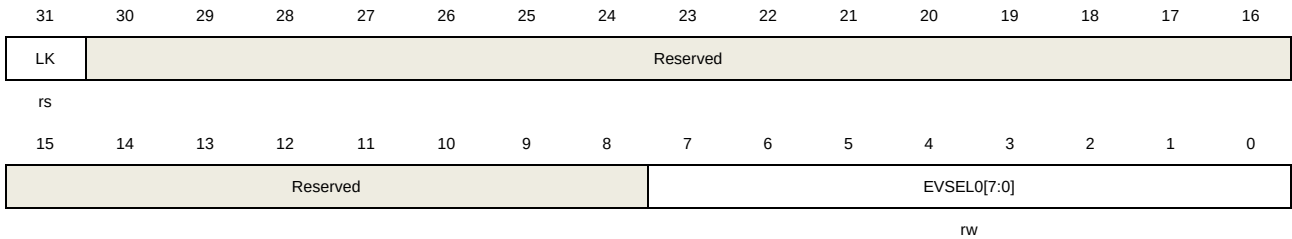
Bits	Fields	Descriptions
31	LK	EVIC_RCU register lock. This bit is set by software and cleared only by a system reset. When it is set to 1, EVIC_RCU register is read-only. 0: EVIC_RCU register is read-write. 1: EVIC_RCU register is read-only.
30:8	Reserved	Must kept at reset value.
7:0	EVSEL0[7:0]	Event source selection for IRC32M switching These bits are used to select an event source connected to the clock switching trigger input. When the selected event occurs, the CK_SYS clock switches to IRC32M. For the detailed event source, please refer to Table 18-1. Event source bit fields selection .

18.4.11 Event interconnect for TIMER0 register (EVIC_TIMER0)

Address offset: 0x38

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



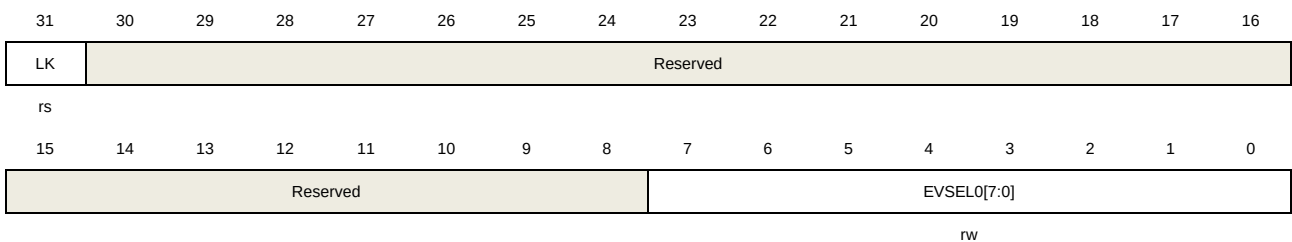
Bits	Fields	Descriptions
31	LK	EVIC_TIMER0 register lock. This bit is set by software and cleared only by a system reset. When it is set to 1, EVIC_TIMER0 register is read-only. 0: EVIC_TIMER0 register is read-write. 1: EVIC_TIMER0 register is read-only.
30:8	Reserved	Must kept at reset value.
7:0	EVSEL0[7:0]	Event source selection for TIMER0 slave mode. These bits are used to select an event source connected to TIMER0 slave mode trigger input. For the detailed configuration, please refer to Table 18-1. Event source bit fields selection.

18.4.12 Event interconnect for TIMER7 register (EVIC_TIMER7)

Address offset: 0x3C

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



Bits	Fields	Descriptions
31	LK	EVIC_TIMER7 register lock. This bit is set by software and cleared only by a system reset. When it is set to 1, EVIC_TIMER7 register is read-only. 0: EVIC_TIMER7 register is read-write. 1: EVIC_TIMER7 register is read-only.
30:8	Reserved	Must kept at reset value.
7:0	EVSEL0[7:0]	Event source selection for TIMER7 slave mode.

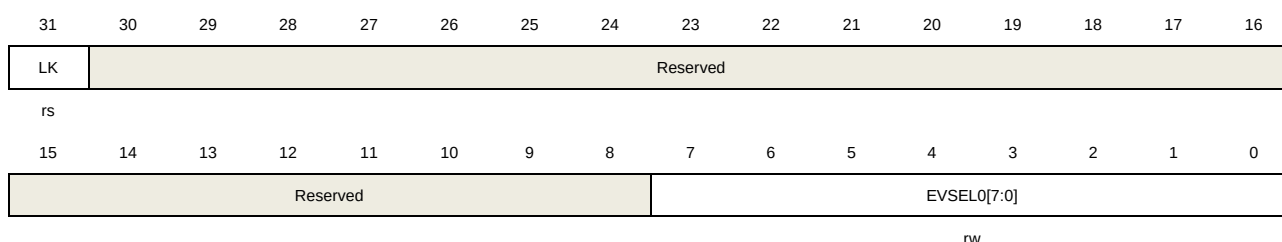
These bits are used to select an event source connected to TIMER7 slave mode trigger input. For the detailed configuration, please refer to [Table 18-1. Event source bit fields selection](#).

18.4.13 Event interconnect for CPTIMER0 register (EVIC_CPTIMER0)

Address offset: 0x40

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



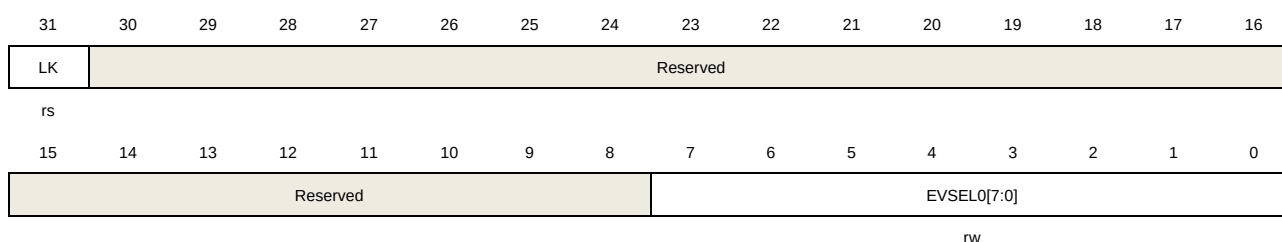
Bits	Fields	Descriptions
31	LK	EVIC_CPTIMER0 register lock. This bit is set by software and cleared only by a system reset. When it is set to 1, EVIC_CPTIMER0 register is read-only. 0: EVIC_CPTIMER0 register is read-write. 1: EVIC_CPTIMER0 register is read-only.
30:8	Reserved	Must kept at reset value.
7:0	EVSEL0[7:0]	Event source selection for CPTIMER0 slave mode These bits are used to select an event source connected to CPTIMER0 slave mode trigger input. For the detailed configuration, please refer to Table 18-1. Event source bit fields selection .

18.4.14 Event interconnect for CPTIMERW register (EVIC_CPTIMERW)

Address offset: 0x44

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



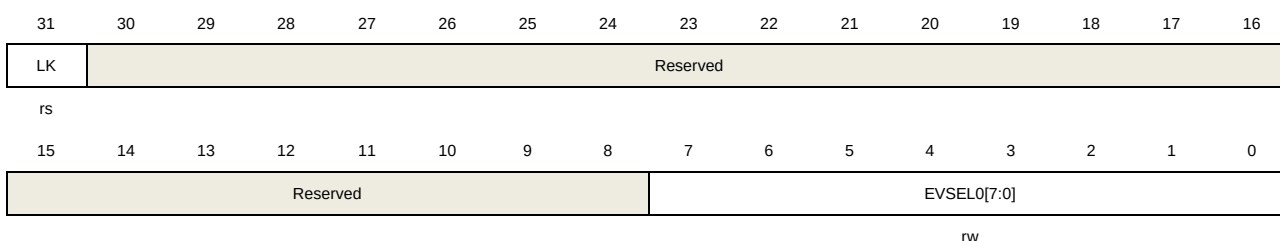
Bits	Fields	Descriptions
31	LK	EVIC_CPTIMERW register lock. This bit is set by software and cleared only by a system reset. When it is set to 1, EVIC_CPTIMERW register is read-only. 0: EVIC_CPTIMERW register is read-write. 1: EVIC_CPTIMERW register is read-only.
30:8	Reserved	Must kept at reset value.
7:0	EVSEL0[7:0]	Event source selection for CPTIMERW slave mode These bits are used to select an event source connected to CPTIMERW slave mode trigger input. For the detailed configuration, please refer to Table 18-1. Event source bit fields selection .

18.4.15 Event interconnect for TIMER1 register (EVIC_TIMER1)

Address offset: 0x48

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



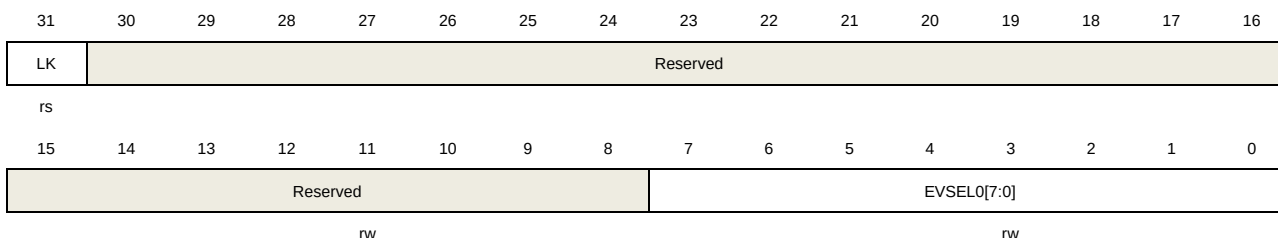
Bits	Fields	Descriptions
31	LK	EVIC_TIMER1 register lock. This bit is set by software and cleared only by a system reset. When it is set to 1, EVIC_TIMER1 register is read-only. 0: EVIC_TIMER1 register is read-write. 1: EVIC_TIMER1 register is read-only.
30:8	Reserved	Must kept at reset value.
7:0	EVSEL0[7:0]	Event input source selection for TIMER1 slave mode trigger These bits are used to select an event source connected to TIMER1 slave mode trigger input. For the detailed configuration, please refer to Table 18-1. Event source bit fields selection .

18.4.16 Event interconnect for TIMER2 register (EVIC_TIMER2)

Address offset: 0x4C

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



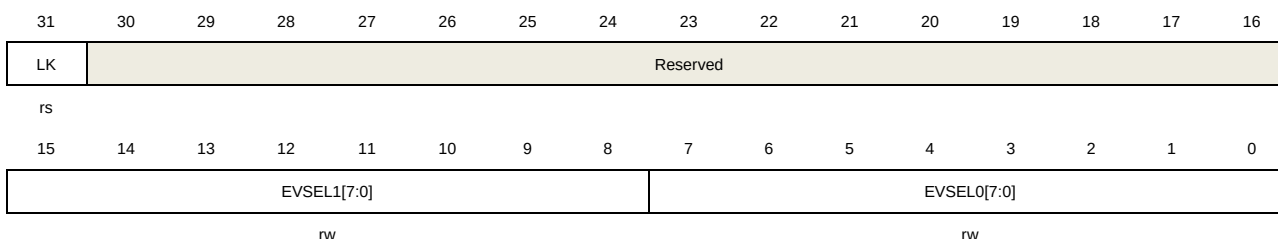
Bits	Fields	Descriptions
31	LK	EVIC_TIMER2 register lock. This bit is set by software and cleared only by a system reset. When it is set to 1, EVIC_TIMER2 register is read-only. 0: EVIC_TIMER2 register is read-write. 1: EVIC_TIMER2 register is read-only.
30:8	Reserved	Must kept at reset value.
7:0	EVSEL0[7:0]	Event source selection for TIMER2 slave mode These bits are used to select an event source connected to TIMER2 slave mode trigger input. For the detailed configuration, please refer to Table 18-1. Event source bit fields selection .

18.4.17 Event interconnect for GPTIMER register 0 (EVIC_GPTIMER_0)

Address offset: 0x50

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



Bits	Fields	Descriptions
31	LK	EVIC_GPTIMER_0 register lock. This bit is set by software and cleared only by a system reset. When it is set to 1, EVIC_GPTIMER_0 register is read-only.

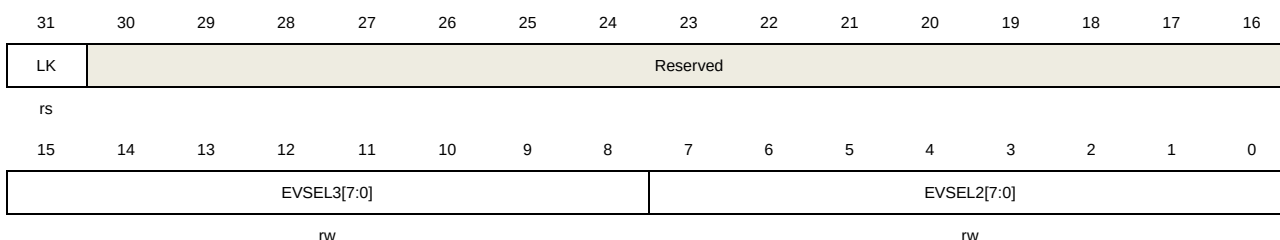
		0: EVIC_GPTIMER_0 register is read-write. 1: EVIC_GPTIMER_0 register is read-only.
30:16	Reserved	Must kept at reset value.
15:8	EVSEL1[7:0]	Event source selection 1 for GPTIMER slave mode These bits are used to select an event source connected to GPTIMER slave mode trigger input 1. For the detailed configuration, please refer to Table 18-1. Event source bit fields selection .
7:0	EVSEL0[7:0]	Event source selection 0 for GPTIMER slave mode These bits are used to select an event source connected to GPTIMER slave mode trigger input 0. For the detailed configuration, please refer to Table 18-1. Event source bit fields selection .

18.4.18 Event interconnect for GPTIMER register 1 (EVIC_GPTIMER_1)

Address offset: 0x54

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



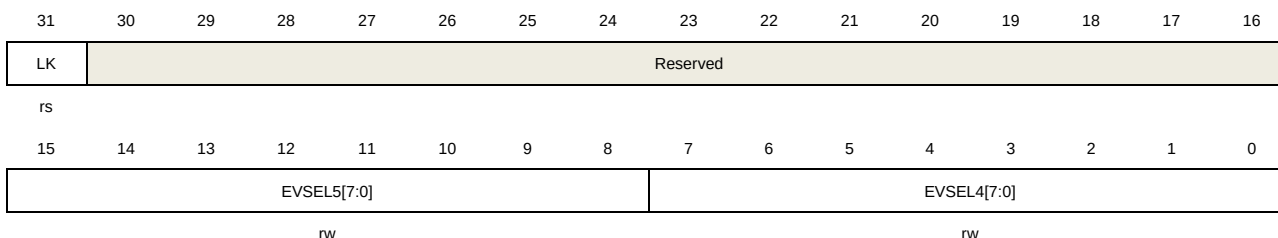
Bits	Fields	Descriptions
31	LK	EVIC_GPTIMER_1 register lock. This bit is set by software and cleared only by a system reset. When it is set to 1, EVIC_GPTIMER_1 register is read-only. 0: EVIC_GPTIMER_1 register is read-write. 1: EVIC_GPTIMER_1 register is read-only.
30:16	Reserved	Must kept at reset value.
15:8	EVSEL3[7:0]	Event source selection 3 for GPTIMER slave mode These bits are used to select an event source connected to GPTIMER slave mode trigger input 3. For the detailed configuration, please refer to Table 18-1. Event source bit fields selection .
7:0	EVSEL2[7:0]	Event source selection 2 for GPTIMER slave mode These bits are used to select an event source connected to GPTIMER slave mode trigger input 2. For the detailed configuration, please refer to Table 18-1. Event source bit fields selection .

18.4.19 Event interconnect for GPTIMER register 2 (EVIC_GPTIMER_2)

Address offset: 0x58

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



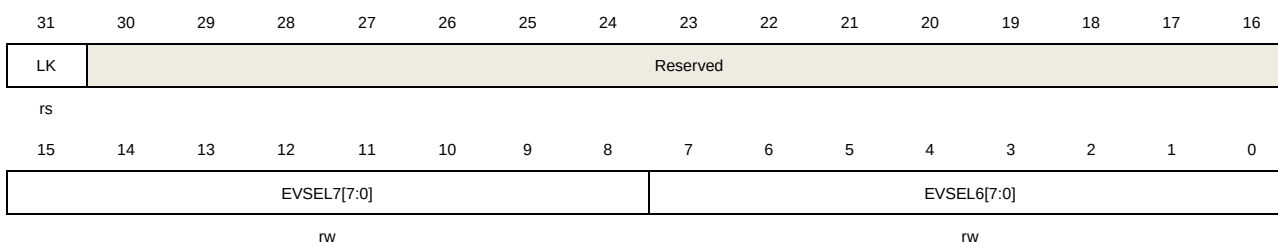
Bits	Fields	Descriptions
31	LK	EVIC_GPTIMER_2 register lock. This bit is set by software and cleared only by a system reset. When it is set to 1, EVIC_GPTIMER_2 register is read-only. 0: EVIC_GPTIMER_2 register is read-write. 1: EVIC_GPTIMER_2 register is read-only.
30:16	Reserved	Must kept at reset value.
15:8	EVSEL5[7:0]	Event source selection 5 for GPTIMER slave mode These bits are used to select an event source connected to GPTIMER slave mode trigger input 5. For the detailed configuration, please refer to Table 18-1. Event source bit fields selection .
7:0	EVSEL4[7:0]	Event source selection 4 for GPTIMER slave mode These bits are used to select an event source connected to GPTIMER slave mode trigger input 4. For the detailed configuration, please refer to Table 18-1. Event source bit fields selection .

18.4.20 Event interconnect for GPTIMER register 3 (EVIC_GPTIMER_3)

Address offset: 0x5C

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



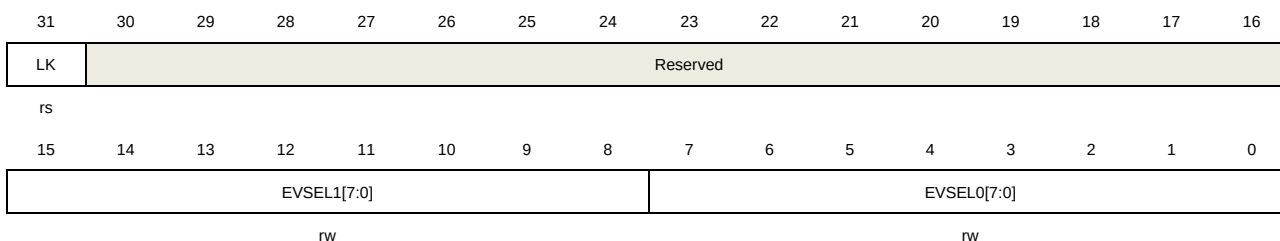
Bits	Fields	Descriptions
31	LK	EVIC_GPTIMER_3 register lock. This bit is set by software and cleared only by a system reset. When it is set to 1, EVIC_GPTIMER_3 register is read-only. 0: EVIC_GPTIMER_3 register is read-write. 1: EVIC_GPTIMER_3 register is read-only.
30:16	Reserved	Must kept at reset value.
15:8	EVSEL7[7:0]	Event source selection 7 for GPTIMER slave mode These bits are used to select an event source connected to GPTIMER slave mode trigger input 7. For the detailed configuration, please refer to Table 18-1. Event source bit fields selection .
7:0	EVSEL6[7:0]	Event source selection 6 for GPTIMER slave mode These bits are used to select an event source connected to GPTIMER slave mode trigger input 6. For the detailed configuration, please refer to Table 18-1. Event source bit fields selection .

18.4.21 Event interconnect for DAC0 enable register (EVIC_DAC0EN)

Address offset: 0x60

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



Bits	Fields	Descriptions
31	LK	EVIC_DAC0EN register lock. This bit is set by software and cleared only by a system reset. When it is set to 1, EVIC_DAC0EN register is read-only. 0: EVIC_DAC0EN register is read-write. 1: EVIC_DAC0EN register is read-only.
30:16	Reserved	Must kept at reset value.
15:8	EVSEL1[7:0]	Event source selection for enabling DAC0_OUT1 These bits are used to select an event source to trigger DAC0_OUT1 enable. For the detailed event source, please refer to Table 18-1. Event source bit fields selection .

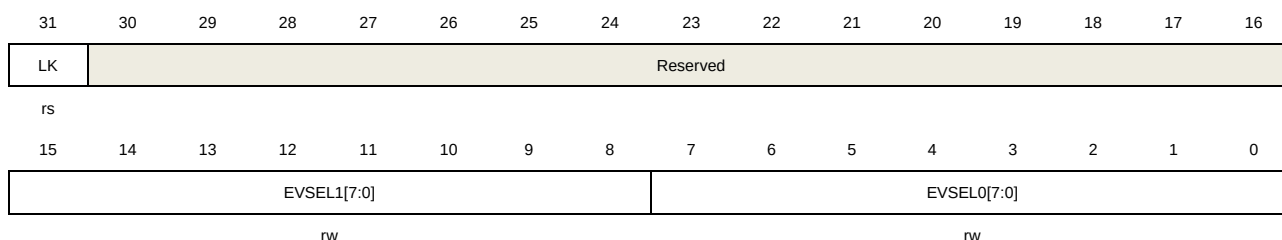
7:0	EVSEL0[7:0]	Event source selection for enabling DAC0_OUT0 These bits are used to select an event source to trigger DAC0_OUT0 enable. For the detailed event source, please refer to Table 18-1. Event source bit fields selection .

18.4.22 Event interconnect for DMAMUX register 0 (EVIC_DMAMUX_0)

Address offset: 0x64

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



Bits	Fields	Descriptions
31	LK	EVIC_DMAMUX_0 register lock. This bit is set by software and cleared only by a system reset. When it is set to 1, EVIC_DMAMUX_0 register is read-only. 0: EVIC_DMAMUX_0 register is read-write. 1: EVIC_DMAMUX_0 register is read-only.
30:16	Reserved	Must kept at reset value.
15:8	EVSEL1[7:0]	Event source selection 1 for DMAMUX These bits are used to select an event source as DMAMUX request multiplexer input 1. For the detailed event source, please refer to Table 18-1. Event source bit fields selection .
7:0	EVSEL0[7:0]	Event source selection 0 for DMAMUX These bits are used to select an event source as DMAMUX request multiplexer input 0. For the detailed event source, please refer to Table 18-1. Event source bit fields selection .

18.4.23 Event interconnect for DMAMUX register 1 (EVIC_DMAMUX_1)

Address offset: 0x68

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



LK	Reserved														
rs															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EVSEL3[7:0]								EVSEL2[7:0]							
rw								rw							

Bits	Fields	Descriptions
31	LK	EVIC_DMAMUX_1 register lock. This bit is set by software and cleared only by a system reset. When it is set to 1, EVIC_DMAMUX_1 register is read-only. 0: EVIC_DMAMUX_1 register is read-write. 1: EVIC_DMAMUX_1 register is read-only.
30:16	Reserved	Must kept at reset value.
15:8	EVSEL3[7:0]	Event source selection 3 for DMAMUX These bits are used to select an event source as DMAMUX request multiplexer input 3. For the detailed event source, please refer to Table 18-1. Event source bit fields selection .
7:0	EVSEL2[7:0]	Event source selection 2 for DMAMUX These bits are used to select an event source as DMAMUX request multiplexer input 2. For the detailed event source, please refer to Table 18-1. Event source bit fields selection .

18.4.24 Event interconnect for DMAMUX register 2 (EVIC_DMAMUX_2)

Address offset: 0x6C

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LK	Reserved														
rs															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EVSEL5[7:0]								EVSEL4[7:0]							
rw								rw							

Bits	Fields	Descriptions
31	LK	EVIC_DMAMUX_2 register lock. This bit is set by software and cleared only by a system reset. When it is set to 1, EVIC_DMAMUX_2 register is read-only. 0: EVIC_DMAMUX_2 register is read-write. 1: EVIC_DMAMUX_2 register is read-only.

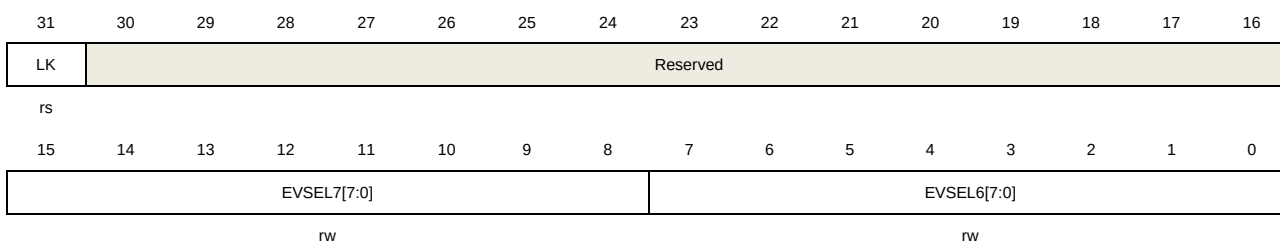
30:16	Reserved	Must kept at reset value.
15:8	EVSEL5[7:0]	Event source selection 5 for DMAMUX These bits are used to select an event source as DMAMUX request multiplexer input 5. For the detailed event source, please refer to Table 18-1. Event source bit fields selection .
7:0	EVSEL4[7:0]	Event source selection 4 for DMAMUX These bits are used to select an event source as DMAMUX request multiplexer input 4. For the detailed event source, please refer to Table 18-1. Event source bit fields selection .

18.4.25 Event interconnect for DMAMUX register 3 (EVIC_DMAMUX_3)

Address offset: 0x70

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



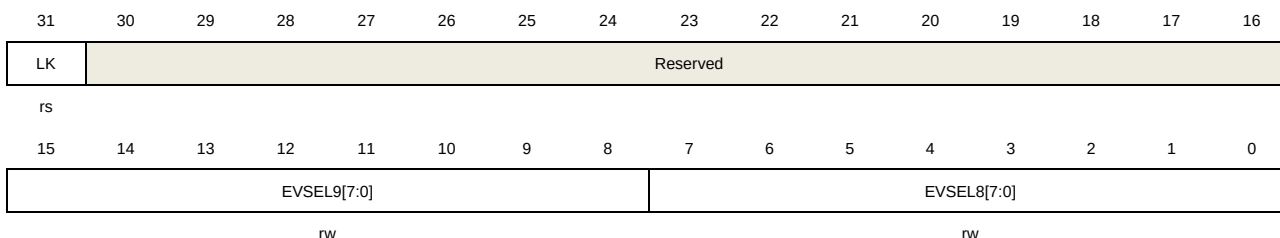
Bits	Fields	Descriptions
31	LK	EVIC_DMAMUX_3 register lock. This bit is set by software and cleared only by a system reset. When it is set to 1, EVIC_DMAMUX_3 register is read-only. 0: EVIC_DMAMUX_3 register is read-write. 1: EVIC_DMAMUX_3 register is read-only.
30:16	Reserved	Must kept at reset value.
15:8	EVSEL7[7:0]	Event source selection 7 for DMAMUX These bits are used to select an event source as DMAMUX request multiplexer input 7. For the detailed event source, please refer to Table 18-1. Event source bit fields selection .
7:0	EVSEL6[7:0]	Event source selection 6 for DMAMUX These bits are used to select an event source as DMAMUX request multiplexer input 6. For the detailed event source, please refer to Table 18-1. Event source bit fields selection .

18.4.26 Event interconnect for DMAMUX register 4 (EVIC_DMAMUX_4)

Address offset: 0x74

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



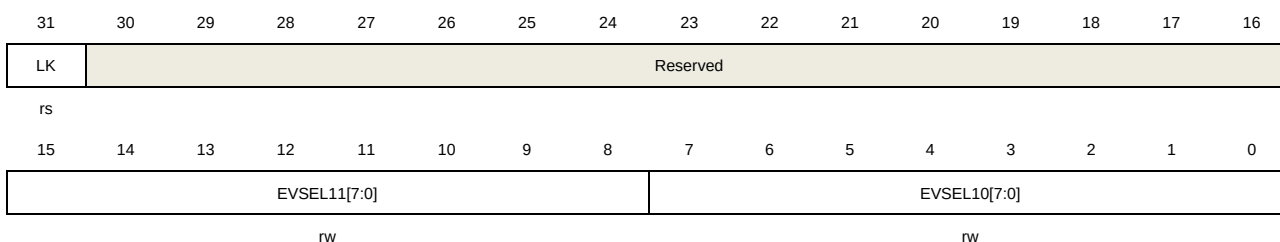
Bits	Fields	Descriptions
31	LK	EVIC_DMAMUX_4 register lock. This bit is set by software and cleared only by a system reset. When it is set to 1, EVIC_DMAMUX_4 register is read-only. 0: EVIC_DMAMUX_4 register is read-write. 1: EVIC_DMAMUX_4 register is read-only.
30:16	Reserved	Must kept at reset value.
15:8	EVSEL9[7:0]	Event source selection 9 for DMAMUX These bits are used to select an event source as DMAMUX request multiplexer input 9. For the detailed event source, please refer to Table 18-1. Event source bit fields selection .
7:0	EVSEL8[7:0]	Event source selection 8 for DMAMUX These bits are used to select an event source as DMAMUX request multiplexer input 8. For the detailed event source, please refer to Table 18-1. Event source bit fields selection .

18.4.27 Event interconnect for DMAMUX register 5 (EVIC_DMAMUX_5)

Address offset: 0x78

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



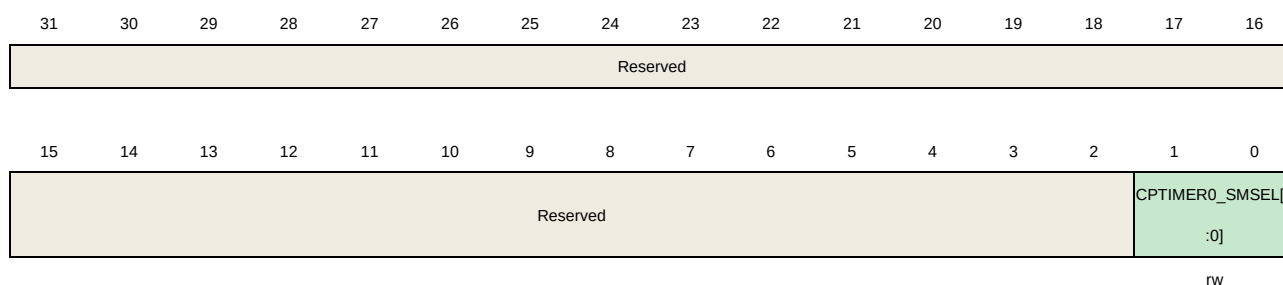
Bits	Fields	Descriptions
31	LK	EVIC_DMAMUX_5 register lock. This bit is set by software and cleared only by a system reset. When it is set to 1, EVIC_DMAMUX_5 register is read-only. 0: EVIC_DMAMUX_5 register is read-write. 1: EVIC_DMAMUX_5 register is read-only.
30:16	Reserved	Must kept at reset value.
15:8	EVSEL11[7:0]	Event source selection 11 for DMAMUX These bits are used to select an event source as DMAMUX request multiplexer input 11. For the detailed event source, please refer to Table 18-1. Event source bit fields selection .
7:0	EVSEL10[7:0]	Event source selection 10 for DMAMUX These bits are used to select an event source as DMAMUX request multiplexer input 10. For the detailed event source, please refer to Table 18-1. Event source bit fields selection .

18.4.28 Slave mode configuration register 0 (EVIC_SMCFG0)

Address offset: 0x7C

Reset value: 0x0000 0003

This register has to be accessed by word (32-bit).



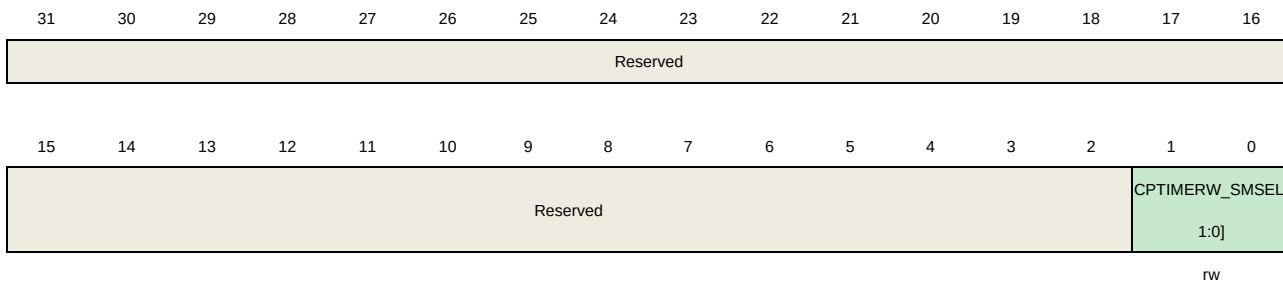
Bits	Fields	Descriptions
31:2	Reserved	Must kept at reset value.
1:0	CPTIMER0_SMSEL[1:0]	CPTIMER0 slave mode selection 00: Counter enable 01: Counter restart 10: Event count 11: Event trigger is disabled

18.4.29 Slave mode configuration register 1 (EVIC_SMCFG1)

Address offset: 0x80

Reset value: 0x0000 0003

This register has to be accessed by word (32-bit).



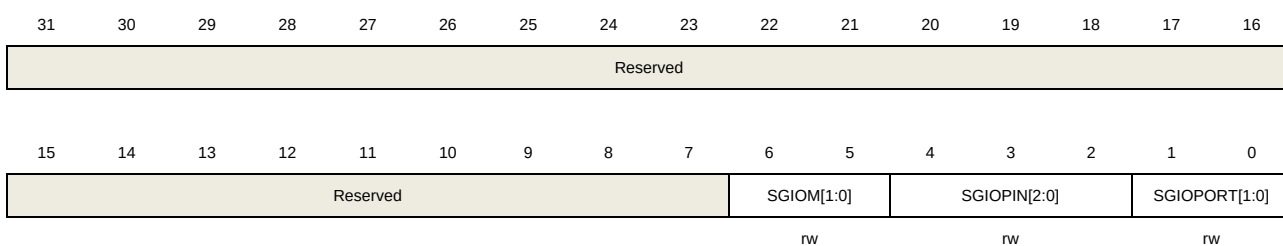
Bits	Fields	Descriptions
31:2	Reserved	Must kept at reset value.
1:0	CPTIMERW_SMSEL[1:0]	CPTIMERW slave mode selection 00: Counter enable 01: Counter restart 10: Event count 11: Event trigger is disabled

18.4.30 Single I/O configuration register x (EVIC_SGIOCFGx) (x=0...3)

Address offset: 0x84 + 0x04 * x (x = 0...3)

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



Bits	Fields	Descriptions
31:7	Reserved	Must kept at reset value.
6:5	SGIOM[1:0]	Single I/O mode As output pin: These bits are used to specify which level output through the selected single I/O pin when an event signal is input to single I/O output controller x. 00: Low level is output 01: High level is output

1x: Inverted level is output

As input pin:

These bits are used to specify which input edge of the selected single I/O pin can generate an event signal.

00: Rising edge

01: Falling edge

1x: Both rising and falling edges

4:2	SGIOPIN[2:0]	Single I/O pin When SGIOPORT[1:0] bits are set to 2'b00, these bits are used to specify which pin from PE8 to PE14 will be selected as a single I/O pin. When SGIOPORT[1:0] bits are set to 2'b01, these bits are used to specify which pin from PF8 to PF14 will be selected as a single I/O pin. 000: PE8 or PF8 001: PE9 or PF9 010: PE10 or PF10 011: PE11 or PF11 100: PE12 or PF12 101: PE13 or PF13 11x: PE14 or PF14
1:0	SGIOPORT[1:0]	Single I/O port 00: Port E 01: Port F 1x: Disabling setting

18.4.31 Group E configuration register (EVIC_GRPECFG)

Address offset: 0x94

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	PE14SEL	PE13SEL	PE12SEL	PE11SEL	PE10SEL	PE9SEL	PE8SEL	Reserved	OVWEN	GRPEOSEL[2:0]			GRPEIND[1:0]		
	rw	rw	rw	rw	rw	rw	rw			rw	rw			rw	

Bits	Fields	Descriptions
31:15	Reserved	Must kept at reset value.
14	PE14SEL	PE14 pin selection 0: PE14 pin is not selected as the member of group E.

		1: PE14 pin is selected as the member of group E.
13	PE13SEL	PE13 pin selection 0: PE13 pin is not selected as the member of group E. 1: PE13 pin is selected as the member of group E.
12	PE12SEL	PE12 pin selection 0: PE12 pin is not selected as the member of group E. 1: PE12 pin is selected as the member of group E.
11	PE11SEL	PE11 pin selection 0: PE11 pin is not selected as the member of group E. 1: PE11 pin is selected as the member of group E.
10	PE10SEL	PE10 pin selection 0: PE10 pin is not selected as the member of group E. 1: PE10 pin is selected as the member of group E.
9	PE9SEL	PE9 pin selection 0: PE9 pin is not selected as the member of group E. 1: PE9 pin is selected as the member of group E.
8	PE8SEL	PE8 pin selection 0: PE8 pin is not selected as the member of group E. 1: PE8 pin is selected as the member of group E.
7:6	Reserved	Must kept at reset value.
5	OVWEN	EVIC_GRPEDH register overwrite enable This bit determines whether the corresponding bits in EVIC_GRPEDH register can obtain new load data when the current load data has not been read. 0: Overwrite disabled 1: Overwrite enabled
4:2	GRPEOSEL[2:0]	Group E output selection These bits are used to specify which level output through the output group E I/O pins when an event signal is input to group E I/O output controller. 000: Low level is output 001: High level is output 010: Inverted level is output 011: EVIC_GRPEDH register data is output 1xx: EVIC_GRPEDH register data is circularly shift output (from MSB to LSB)
1:0	GRPEIND[1:0]	Group E input detection These bits are used to specify which input edge of the input group E I/O pins can generate an event signal from group E I/O input detector. 00: Rising edge 01: Falling edge 1x: Both rising and falling edges

18.4.32 Group E data holding register (EVIC_GRPEDH)

Address offset: 0x98

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

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

Bits	Fields	Descriptions
31:7	Reserved	Must kept at reset value.
6:0	GRPE_DH[6:0]	Group E holding data Output group E I/O pin: if the GRPEOSEL[2:0] bits in EVIC_GPRECFG register is 3b'011 or 3b'1xx, these bits specify the data to be transferred to the GPIOE_OCTL register when an event trigger comes. Input group E I/O pin: these bits specify the data to load the input level value of corresponding pins when an event trigger comes. GRPE_DH[0]: PE8 GRPE_DH[1]: PE9 GRPE_DH[2]: PE10 GRPE_DH[3]: PE11 GRPE_DH[4]: PE12 GRPE_DH[5]: PE13 GRPE_DH[6]: PE14 Note: It is disabled to write to the EVIC_GRPEDH register bit corresponding to the input group E I/O pin.

18.4.33 Group F configuration register (EVIC_GPRFCFG)

Address offset: 0x9C

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	PF14SEL	PF13SEL	PF12SEL	PF11SEL	PF10SEL	PF9SEL	PF8SEL	Reserved	OVWEN	GRPFOSEL[2:0]			GRPFIND[1:0]		
	rw	rw	rw	rw	rw	rw	rw		rw	rw			rw		

Bits	Fields	Descriptions
31:15	Reserved	Must kept at reset value.
14	PF14SEL	PF14 pin selection 0: PF14 pin is not selected as the member of group F. 1: PF14 pin is selected as the member of group F.
13	PF13SEL	PF13 pin selection 0: PF13 pin is not selected as the member of group F. 1: PF13 pin is selected as the member of group F.
12	PF12SEL	PF12 pin selection 0: PF12 pin is not selected as the member of group F. 1: PF12 pin is selected as the member of group F.
11	PF11SEL	PF11 pin selection 0: PF11 pin is not selected as the member of group F. 1: PF11 pin is selected as the member of group F.
10	PF10SEL	PF10 pin selection 0: PF10 pin is not selected as the member of group F. 1: PF10 pin is selected as the member of group F.
9	PF9SEL	PF9 pin selection 0: PF9 pin is not selected as the member of group F. 1: PF9 pin is selected as the member of group F.
8	PF8SEL	PF8 pin selection 0: PF8 pin is not selected as the member of group F. 1: PF8 pin is selected as the member of group F.
7:6	Reserved	Must kept at reset value.
5	OVWEN	EVIC_GRPFDH register overwrite enable This bit determines whether the corresponding bits in EVIC_GRPFDH register can obtain new load data when the current load data has not been read. 0: Overwrite disabled 1: Overwrite enabled
4:2	GRPFOSEL[2:0]	Group F output selection These bits are used to specify which level output through the output group F I/O pins when an event signal is input to group F I/O output controller. 000: Low level is output 001: High level is output 010: Inverted level is output 011: EVIC_GRPFDH register data is output 1xx: EVIC_GRPFDH register data is circularly shift output (from MSB to LSB)
1:0	GRPFIND[1:0]	Group F input detection These bits are used to specify which input edge of the input group F I/O pins can

generate an event signal from group F I/O input detector.

00: Rising edge

01: Falling edge

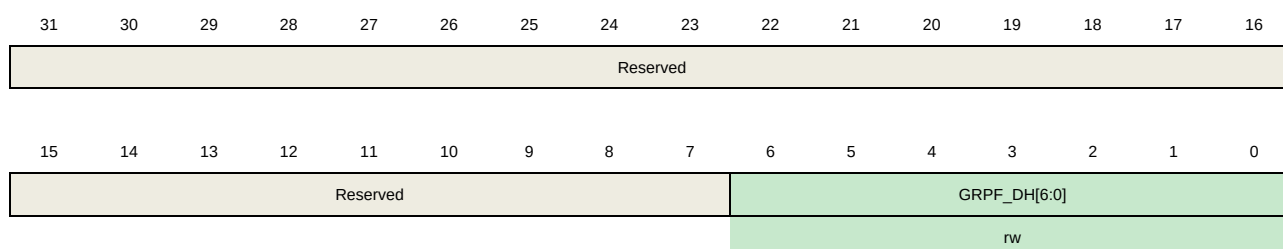
1x: Both rising and falling edges

18.4.34 Group F data holding register (EVIC_GRPFDH)

Address offset: 0x100

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



Bits	Fields	Descriptions
31:7	Reserved	Must kept at reset value.
6:0	GRPF_DH[6:0]	Group F holding data Output group F I/O pin: if the GRPFOSEL[2:0] bits in EVIC_GPRFCFG register is 3b'011 or 3b'1xx, these bits specify the data to be transferred to the GPIOF_OCTL register when an event trigger comes. Input group F I/O pin: these bits specify the data to load the input level value of corresponding pins when an event trigger comes. GRPF_DH[0]: PF8 GRPF_DH[1]: PF9 GRPF_DH[2]: PF10 GRPF_DH[3]: PF11 GRPF_DH[4]: PF12 GRPF_DH[5]: PF13 GRPF_DH[6]: PF14 Note: It is disabled to write to the EVIC_GRPFDH register bit corresponding to the input group F I/O pin.

19 Space vector pulse width modulation (SVPWM)

19.1 Overview

The SVPWM(Space vector pulse width modulation) arithmetic unit is used to control the alternating current induction motor or permanent magnet synchronous motor in the three-phase voltage source inverter power device of special switch trigger sequence and the combination of pulse width. This switch trigger sequence and combination will produce three sinusoidal currents in the stator coil with an electrical angle difference of 120° .

19.2 Characteristics

The SVPWM main features are as follows:

The algorithm includes seven-stage PWM and five-stage PWM.

The three phase PWM wave is generated directly with the voltage space vector.

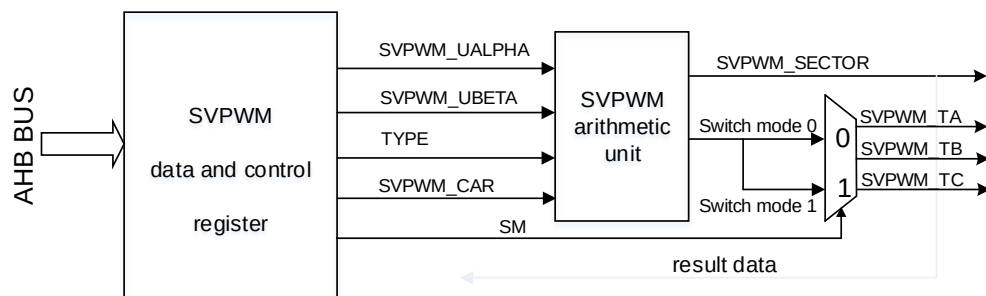
Float point operation.

Support synthetic voltage vector overmodulation processing.

19.2.1 Block diagram

[Figure 19-1. SVPWM module block diagram](#) below provides details on the internal configuration of the SVPWM interface.

Figure 19-1. SVPWM module block diagram



19.3 Function overview

19.3.1 SVPWM clock

The SVPWM clock provided by the clock controller is synchronous with the AHB clock.

19.3.2 Data format configuration

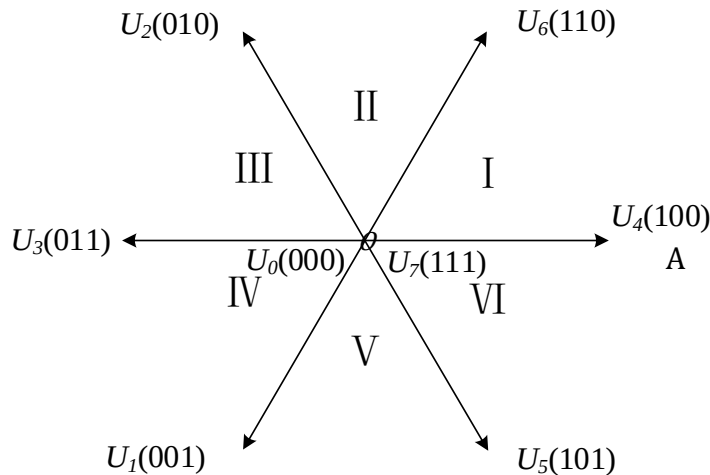
The input data of svpwm module is IEEE 32-Bit Single Precision Floating-Point Format. The SVPWM_UALPHA、SVPWM_UBETA registers are used to configure the input data.

The number of PWM cycles can be configured by setting the SVPWM_CAR register. And the output data of svpwm module are 16-bit unsigned number.

19.3.3 SVPWM principle

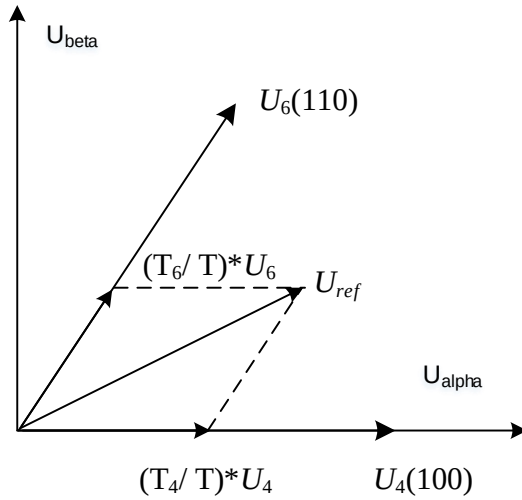
The pulse width modulation signal of 3-phase motor voltage signal is generated by SVPWM algorithm. The pulse width generation process of each phase can be simplified into several first order equations. Each phase output of the three-phase inverter can be one of two states, that is, the output of the inverter can be connected to bus voltage or ground, when bus voltage is connected, it is represented by 1, and when ground is connected it is represented by 0, thus there are eight possible states (000,001,010,011,100,101,110,111). The two states 000 and 111 are regarded as invalid states, because there is no line voltage between any two phases. The remaining six states are represented as vectors with a rotation interval of 60 degrees.

Figure 19-2. SVPWM output the basic voltage vector



The process of space vector modulation allows any space voltage vector to be represented by the sum of the components of two adjacent vectors. Taking sector I as an example, the expected voltage vector U_{ref} is composed of valid vectors $U_4(100)$, $U_6(110)$, and invalid vectors $U_0(000)$, $U_7(111)$.

Figure 19-3. Mean space vector modulation



According to volt-second equilibrium,

$$\begin{cases} U_{ref} * T = U_4 * T_4 + U_6 * T_6 \\ T = T_4 + T_6 + T_0^* \end{cases} \quad (1-1)$$

Where, T is the switching period of SVPWM, T_4 and T_6 are the operation times of valid vectors $U_4(100)$, $U_6(110)$, T_0^* is the operation times of invalid vectors $U_0(000)$, $U_7(111)$.

19.3.4 SVPWM implementation

Firstly, the static 2-axis voltage U_{alpha} and U_{beta} are transformed to the Intermediate voltage variable U_{s1} U_{s2} U_{s3} .

$$\begin{cases} U_{s1} = U_{beta} \\ U_{s2} = (-U_{beta} + \sqrt{3}U_{alpha})/2 \\ U_{s3} = (-U_{beta} - \sqrt{3}U_{alpha})/2 \end{cases} \quad (1-2)$$

Define three variables a,b,c at the same time, if $U_{s1} > 0$, then $a = 1$, otherwise $a = 0$; if $U_{s2} > 0$, then $b = 1$, otherwise $b = 0$; if $U_{s3} > 0$, then $c = 1$, otherwise $c = 0$; Define another variable $N=a+b*2+c*4$, the space is divided into 6 sectors according to different sec.

Additional definition:

$$\begin{cases} X = \sqrt{3}U_{beta} \\ Y = \sqrt{3}(U_{beta} + \sqrt{3}U_{alpha})/2 \\ Z = \sqrt{3}(U_{beta} - \sqrt{3}U_{alpha})/2 \end{cases} \quad (1-3)$$

In sector I, T1 is equal to the operation time T_4 of the vectors $U_4(100)$, T2 is equal to the operation time T_6 of the vectors $U_6(110)$, in sector II, T1 is equal to the operation time T_2 of the vectors $U_2(010)$, T2 is equal to the operation time T_6 of the vectors $U_6(110)$, in sector III, T1 is equal to the operation time T_2 of the vectors $U_2(010)$, T2 is equal to the operation time T_3 of the vectors $U_3(011)$, in sector IV, T1 is equal to the operation time T_1 of the vectors $U_1(001)$, T2 is equal to the operation time T_3 of the vectors $U_3(011)$, in sector V, T1 is equal to the operation time T_1 of the vectors $U_1(001)$, T2 is equal to the

operation time T_5 of the vectors $U_5(101)$, in sector VI, T_1 is equal to the operation time T_4 of the vectors $U_4(100)$, T_2 is equal to the operation time T_5 of the vectors $U_5(110)$.

For the values of T_1 and T_2 in different sectors, the corresponding relationship is as follows:

Table 19-1. The correspondence between the sector and the time

N	1	2	3	4	5	6
sector	II	VI	I	IV	III	V
T_1	Z	Y	-Z	-X	X	-Y
T_2	Y	-X	X	Z	-Y	-Z
T_0^*	$T - T_1 - T_2$	$T - T_1 - T_2$	$T - T_1 - T_2$	$T - T_1 - T_2$	$T - T_1 - T_2$	$T - T_1 - T_2$

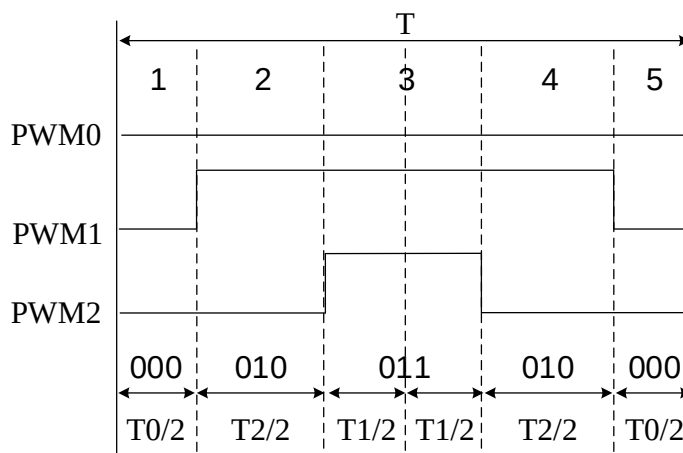
In the PWM period T , the output time of vector T_1 is T_1/T , while the output time of vector T_2 is T_2/T , and the output is invalid vector in the remaining time. The invalid vector can be composed of $U_0(000)$ and is a five-segment SVPWM. It can also be composed of $U_7(111)$ in the middle and $U_0(000)$ on both sides, which is a seven-segment SVPWM.

19.3.5 Five-segment SVPWM

Within a single PWM cycle, the five-stage SVPWM will insert the zero vector in the middle, the torque ripple is large, and the operation will be unstable at low frequency. In each PWM cycle, there is always a phase bridge arm's switching stage does not need to change, but in the seven-stage method, each phase bridge arm needs to be switched once. The switching times of five-stage's bridge arm is $1/3$ less than that of the seven-stage bridge arm, so that five-stage's switch has the lowest power consumption.

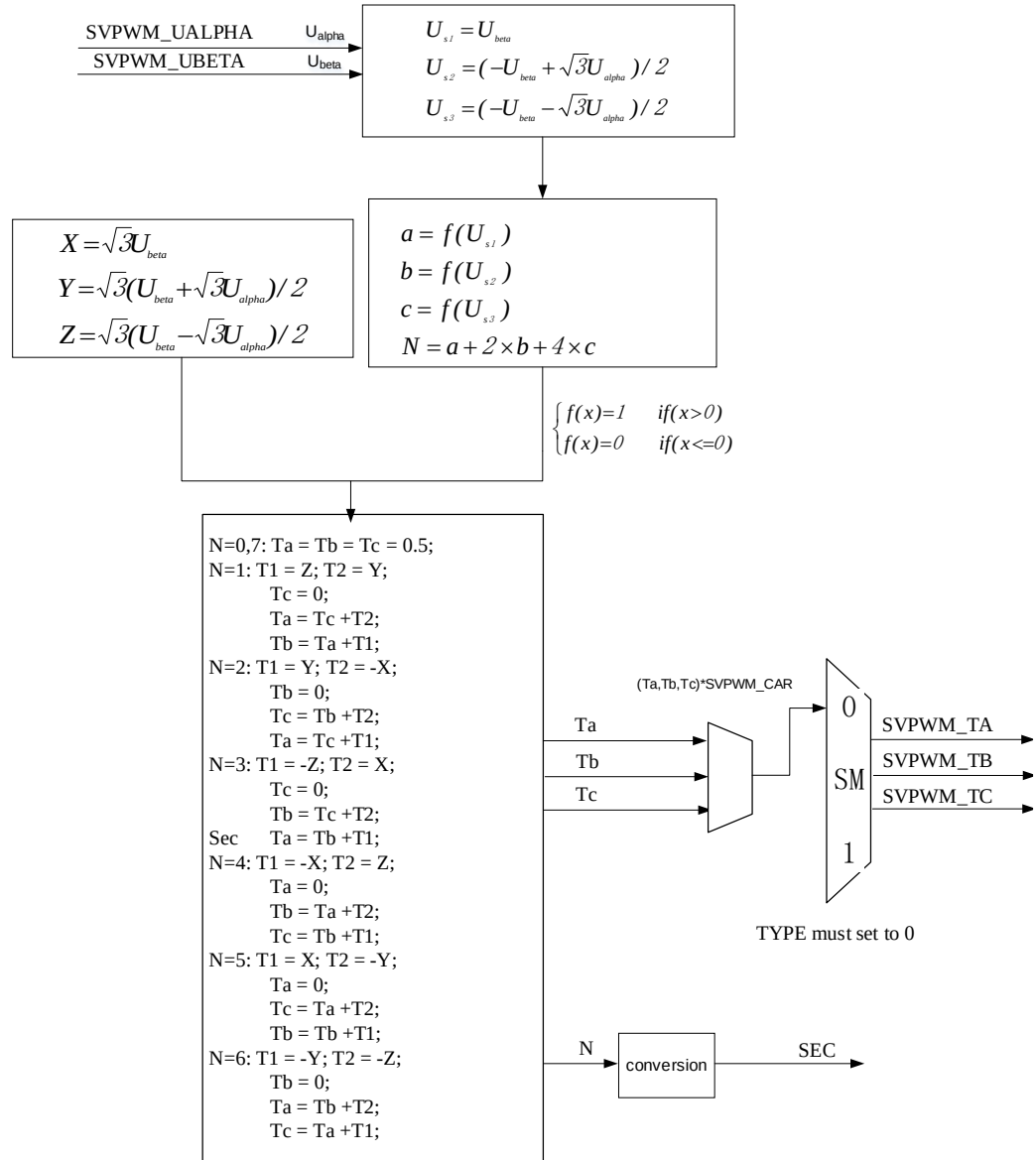
The five-stage SVPWM method can be selected on by setting the TYPE bit in the SVPWM_CTL0 register. The following figure shows the five-stage SVPWM waveform, with the PWM signal configured to align the output center.

Figure 19-4. Five-segment SVPWM waveform



The algorithm of five-segment SVPWM is shown in the figure. T_a , T_b and T_c can be calculated from U_{α} and U_{β} , and the data format of input and output can be set.

Figure 19-5. Five-segment SVPWM algorithm



U_{α} , U_{β} , T1, T2, Ta, Tb, Tc are float data, SECTOR are unsigned 8-digit number. "conversion" is an array, conversion[] = {0,1,5,0,3,2,4,0}. And, the SECTOR is I when SEC is 0, SECTOR is II when SEC is 1, SECTOR is III when SEC is 2, SECTOR is IV when SEC is 3, SECTOR is V when SEC is 4, SECTOR is VI when SEC is 5.

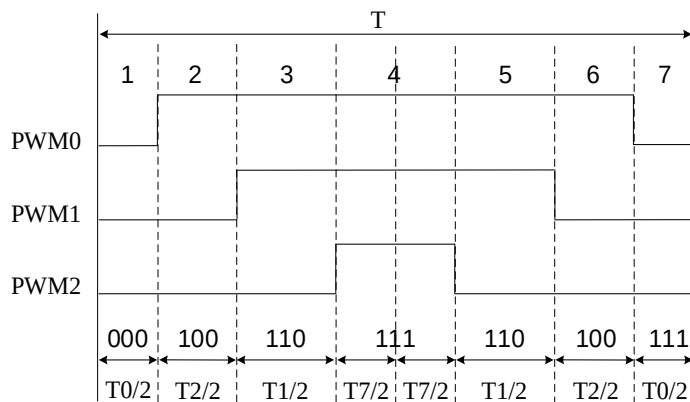
19.3.6 Seven-segment SVPWM

In the seven-stage method, half of the zero vector is inserted in the middle of the PWM cycle, and the other half is inserted on both side of the PWM cycle, which makes the operation of the flux more stable and reduce the motor torque ripple, so that the motor characteristics at low frequency are significantly better than those of the five-stage method, while the characteristics at high frequency are not different.

The seven-stage SVPWM method can be selected on by setting the TYPE bit in the

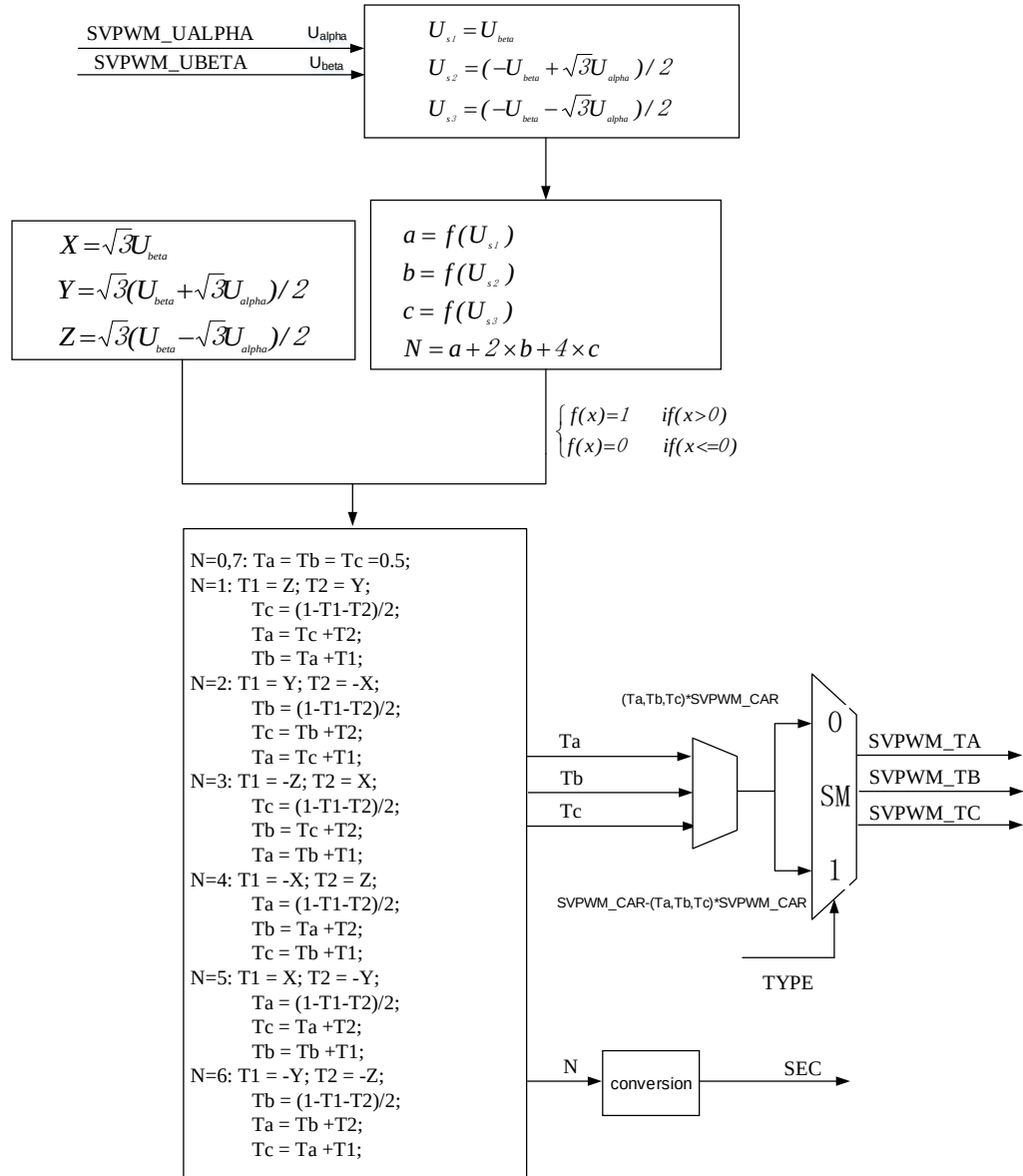
SVPWM_CTL0 register. The following figure shows the seven-stage SVPWM waveform, with the PWM signal configured to align the output center.

Figure 19-6. Seven-segment SVPWM waveform



The algorithm of five-segment SVPWM is shown in the figure. T_a , T_b and T_c can be calculated from U_{α} and U_{β} , and the data format of input and output can be set.

Figure 19-7. Seven-segment SVPWM algorithm



U_{α} , U_{β} , T_1 , T_2 , T_a , T_b , T_c are float data, SECTOR are unsigned 8-digit number. "conversion" is an array, conversion[] = {0,1,5,0,3,2,4,0}. And, the SECTOR is I when SEC is 0, SECTOR is II when SEC is 1, SECTOR is III when SEC is 2, SECTOR is IV when SEC is 3, SECTOR is V when SEC is 4, SECTOR is VI when SEC is 5.

19.3.7 OverModulation processing

Calculation overflow may exist in either five-segment SVPWM or seven-segment SVPWM. When any calculation result of SVPEM_TA, SVPEM_TB, SVPEM_TC is out of range, the overflow flag (OCH, OBH, OAH, OCL, OBL, OAL in SVPWM_STAT register) is set.

19.4 Register definition

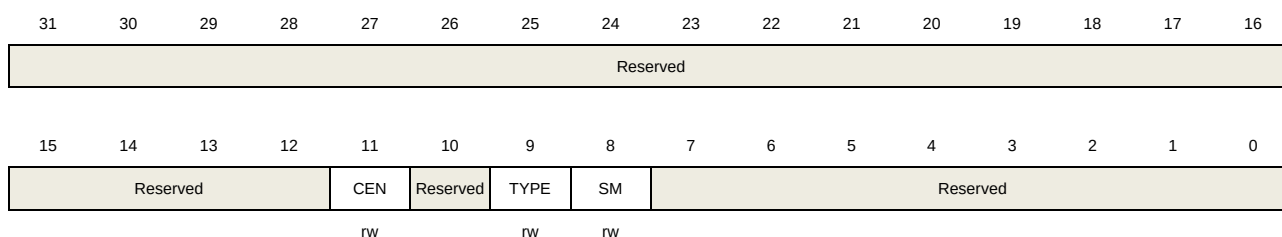
SVPWM base address: 0x4802 4000

19.4.1 Control register 0 (SVPWM_CTL0)

Address offset: 0x00

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



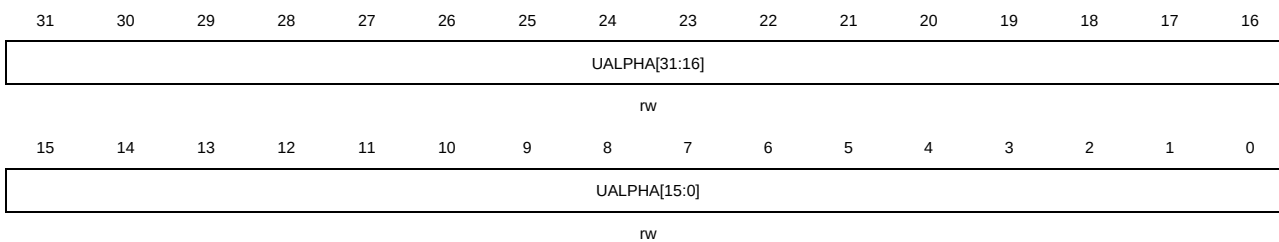
Bits	Fields	Descriptions
31:12	Reserved	Must be kept at reset value.
11	CEN	SVPWM peripheral enable 0: SVPWM operation has finished or not started 1: Start the SVPWM operation. During the operation, the flag is 1. After the operation is complete, the flag is reset by the hardware. Note: This bit is set by software and reset by hardware.
10	Reserved	Must be kept at reset value.
9	TYPE	Five-segment SVPWM and Seven-segment SVPWM select flag bit 0: Five-segment SVPWM 1: Seven-segment SVPWM
8	SM	Switch mode 0: Switch mode 0. 1: Switch mode 1. Note: This bit field cannot be written when the SVPWM is enabled (CEN=1), and can be only used in Seven-segment SVPWM, it must set to 0 in Five-segment SVPWM.
7:0	Reserved	Must be kept at reset value.

19.4.2 Ualpha register (SVPWM_UALPHA)

Address offset: 0x04

Reset value: 0x3F80 0000

This register has to be accessed by word (32-bit).



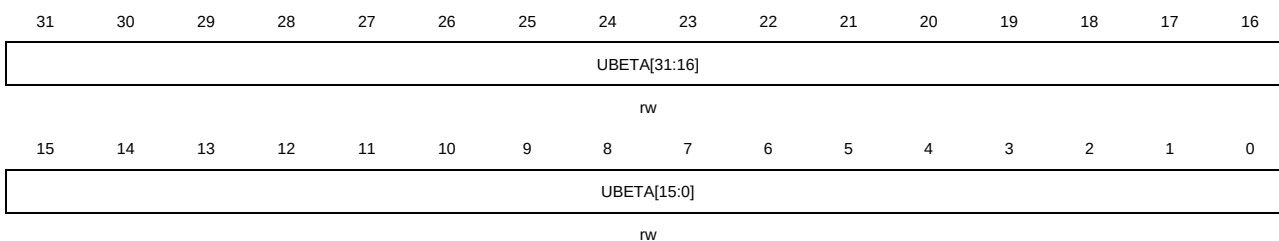
Bits	Fields	Descriptions
31:0	UALPHA	This data represents the UALPHA input to the SVPWM module Note: This bit field cannot be written when the SVPWM is enabled (CEN=1), and UALPHA[31:0] must meet IEEE 32-Bit Single Precision Floating-Point Format.

19.4.3 Ubeta register (SVPWM_UBETA)

Address offset: 0x08

Reset value: 0x3F80 0000

This register has to be accessed by word (32-bit).



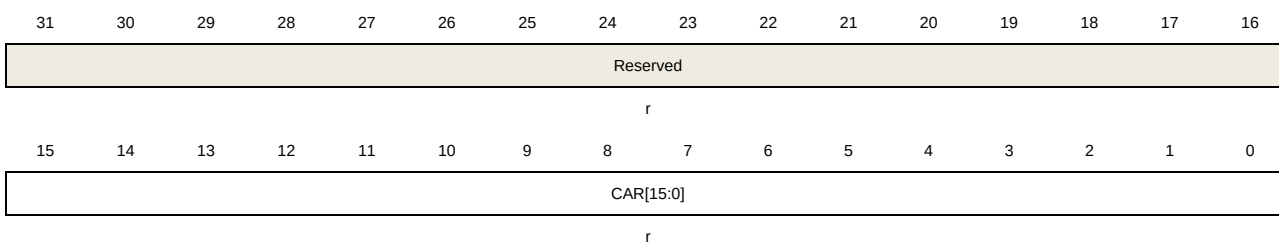
Bits	Fields	Descriptions
31:0	UBETA	This data represents the UBETA input to the SVPWM module Note: This bit field cannot be written when the SVPWM is enabled (CEN=1) and UBETA[31:0] must meet IEEE 32-Bit Single Precision Floating-Point Format.

19.4.4 Counter auto reload register (SVPWM_CAR)

Address offset: 0x0C

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



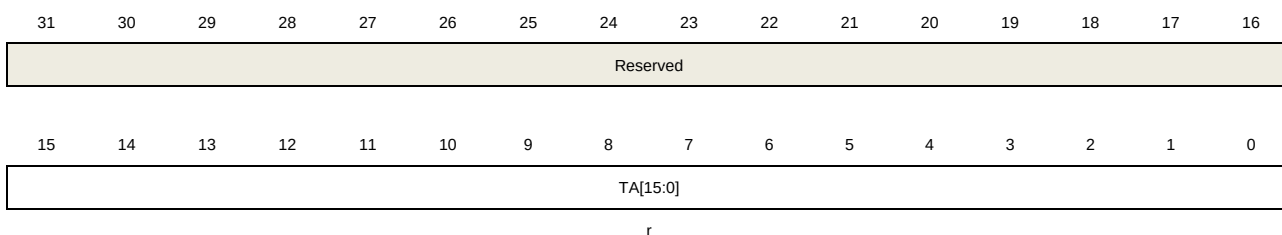
Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CAR[15:0]	This bit-field indicates the period of TIMER. Note: This bit field cannot be written when the SVPWM is enabled (CEN=1), and it is a 16-bit unsigned integer number.

19.4.5 Ta register (SVPWM_TA)

Address offset: 0x10

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



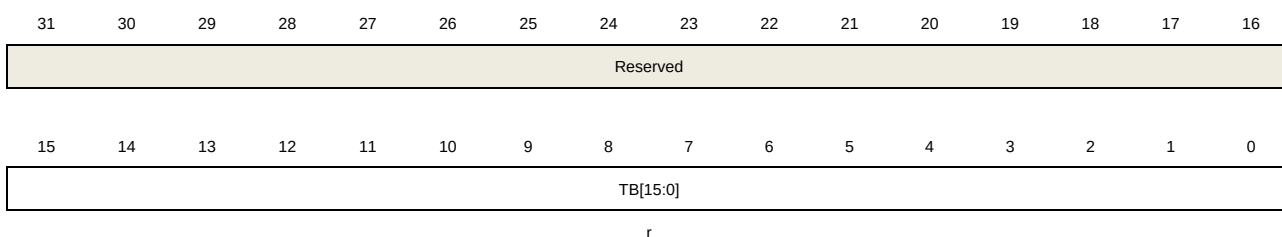
Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	TA	This data represents the TA output to the SVPWM module

19.4.6 Tb register (SVPWM_TB)

Address offset: 0x14

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



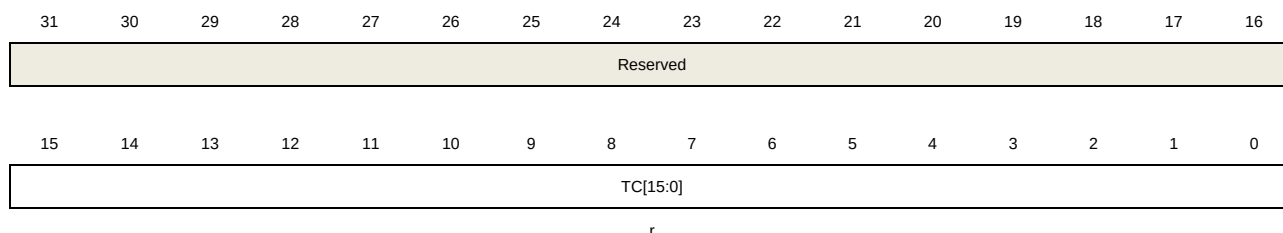
Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	TB	This data represents the TB output to the SVPWM module

19.4.7 Tc register (SVPWM_TC)

Address offset: 0x18

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



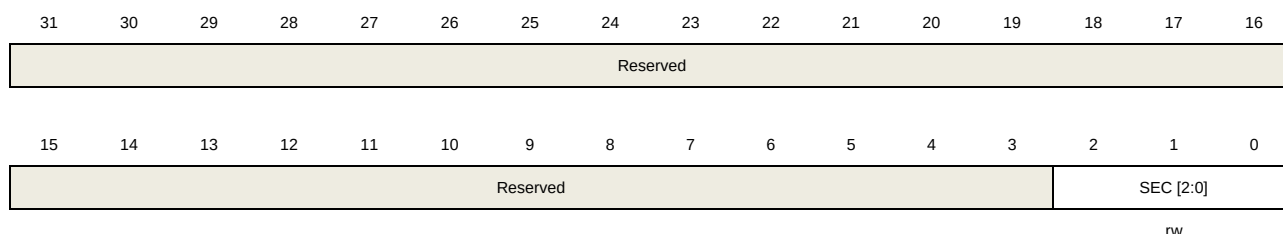
Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	TC	This data represents the TC output to the SVPWM module

19.4.8 Sector register (SVPWM_SEC)

Address offset: 0x1C

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



Bits	Fields	Descriptions
31:3	Reserved	Must be kept at reset value.
2:0	SEC [2:0]	Sector value to the SVPWM module the SECTOR is I when SEC is 0, SECTOR is II when SEC is 1, SECTOR is III when SEC is 2, SECTOR is IV when SEC is 3, SECTOR is V when SEC is 4, SECTOR is VI when SEC is 5.

19.4.9 Status register (SVPWM_STAT)

Address offset: 0x20

Reset value: 0x0000 0000

This register has to be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved													OCH	OBH	OAH
													r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									OSF	Reserved			OCL	OBL	OAL
									r				r	r	r

Bits	Fields	Descriptions
31:19	Reserved	Must be kept at reset value.
18	OCH	The duty cycle of C-phase is greater than the auto reload value (SVPWM_CAR) overflow flag. 0: The result does not overflow 1: The result overflow Note: This bit is set by hardware when a overflow event, update once every calculation is complete.
17	OBH	The duty cycle of B-phase is greater than the auto reload value (SVPWM_CAR) overflow flag. 0: The result does not overflow 1: The result overflow Note: This bit is set by hardware when a overflow event, update once every calculation is complete.
16	OAH	The duty cycle of A-phase is greater than the auto reload value (SVPWM_CAR) overflow flag. 0: The result does not overflow 1: The result overflow Note: This bit is set by hardware when a overflow event, update once every calculation is complete.
15:7	Reserved	Must be kept at reset value.
6	OSF	SVPWM operation status flag bit. This bit is set by hardware when SVPWM algorithm operation is complete. After update SVPWM_UALPHA or SVPWM_UBETA, the hardware automaticly clear zero. 0: The svpwm operation is not complete 1: The svpwm operation is executed successfully
5:3	Reserved	Must be kept at reset value.
2	OCL	The duty cycle of C-phase is less than 0 overflow flag. 0: The result does not overflow 1: The result overflow Note: This bit is set by hardware when a overflow event, update once every

		calculation is complete.
1	OBL	The duty cycle of B-phase is less than 0 overflow flag. 0: The result does not overflow 1: The result overflow Note: This bit is set by hardware when a overflow event, update once every calculation is complete.
0	OAL	The duty cycle of A-phase is less than 0 overflow flag. 0: The result does not overflow 1: The result overflow Note: This bit is set by hardware when a overflow event, update once every calculation is complete.

20 Watchdog timer (WDGT)

The watchdog timer (WDGT) is a hardware timing circuitry that can be used to detect system failures due to software malfunctions. There are two watchdog timer peripherals in the chip: free watchdog timer (FWDGT) and window watchdog timer (WWDGT). They offer a combination of a high safety level, flexibility of use and high timing accuracy. Both watchdog timers are offered to resolve malfunctions of software.

The watchdog timer will generate a reset (or interrupt) when the internal counter reaches a given value. The watchdog timer counter can be stopped while the processor is in the debug mode.

20.1 Free watchdog timer (FWDGT)

20.1.1 Overview

The free watchdog timer (FWDGT) has free clock source (IRC32K). Thereupon the FWDGT can operate even if the main clock fails. It's suitable for the situation that requires an independent environment and lower timing accuracy.

The Free watchdog timer generate a reset or interrupt when the internal down counter reaches 0 or the counter is refreshed when the value of the counter is greater than the window register value. The register write protection function in free watchdog can be enabled to prevent it from changing the configuration unexpectedly.

20.1.2 Characteristics

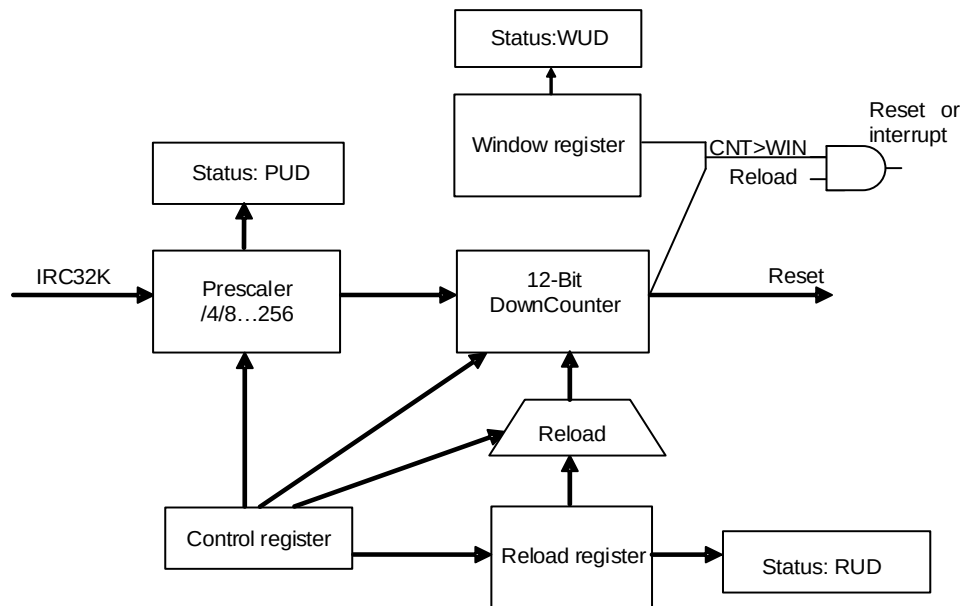
- Free-running 12-bit down counter.
- Generate reset or interrupt in two conditions when FWDGT is enabled:
 - Reset or interrupt when the counter reached 0.
 - The counter is refreshed when the value of the counter is greater than the window register value.
- Free clock source, FWDGT can operate even if the main clock fails such as in standby and Deep-sleep modes.
- Hardware free watchdog bit, automatically start the FWDGT or not when power on.
- FWDGT debug mode, the FWDGT can stop or continue to work in debug mode.
- Configure FWDGSPD_STDBY or FWDGSPD_DPSLP, FWDGT would stop working, or else wake up the device and continue to work, in standby or deep sleep mode.
- Configurable interrupt or reset requests

20.1.3 Function overview

The free watchdog consists of an 8-stage prescaler and a 12-bit down counter. [Figure 20-1.](#)

[Free watchdog block diagram](#) shows the functional block of the free watchdog module.

Figure 20-1. Free watchdog block diagram



The free watchdog is enabled by writing the value (0xCCCC) to the control register (FWDGT_CTL), then the counter starts counting down. When the counter reaches the value (0x000), there will be a reset or interrupt. The RSTREQEN bit in the reset control register (FWDGT_RSTCTL) is configured to enable an output reset request or an output interrupt request.

The counter can be reloaded by writing the value (0xAAAA) to the FWDGT_CTL register at any time. The reload value comes from the FWDGT_RLD register. The software can prevent the watchdog reset or interrupt by reloading the counter before the counter reaches the value (0x000).

Note: After a system reset or interrupt occurs, both reload and enable operations must be executed to restart the watchdog.

By setting the appropriate window in the FWDGT_WND register, the FWDGT can also work as a window watchdog timer. A reset or interrupt will occur if the reload operation is performed while the counter is greater than the value stored in the window register (FWDGT_WND). The default value of the FWDGT_WND is 0x0000 0FFF, so if it is not updated, the window option is disabled. A reload operation is performed in order to reset the downcounter to the FWDGT_RLD value and the prescaler counter to generate the next reload, as soon as the window value is changed.

If the "hardware-enable independent watchdog function" feature is enabled in the option bytes, the watchdog timer is automatically activated at power on. In the reset signal state, the configuration values in the option bytes are reloaded into the registers of the independent watchdog timer (including RLD / PSC / WND). To avoid a system reset or interrupt, the software should reload the counter before it reaches 0x000.

The FWDGT_PSC register, the FWDGT_RLD register and the FWDGT_WND register are write protected. Before writing these registers, the software should write the value (0x5555) to the FWDGT_CTL register. These registers will be protected again by writing any other value to the FWDGT_CTL register. When an update operation of the prescaler register (FWDGT_PSC), the reload value register (FWDGT_RLD) or the window register (FWDGT_WND) is ongoing, the status bits in the FWDGT_STAT register are set.

If the FWDGT_HOLD bit in DBG module is cleared, the FWDGT continues to work even the Cortex®-M33 core halted (Debug mode). The FWDGT stops in Debug mode if the FWDGT_HOLD bit is set.

Table 20-1. Min/max FWDGT timeout period at 32 kHz (IRC32K)

Prescaler divider	PSC[2:0] bits	Min timeout (ms) RLD[11:0]=0x000	Max timeout (ms) RLD[11:0]=0xFFFF
1 / 4	000	0.125	512
1 / 8	001	0.25	1024
1 / 16	010	0.5	2048
1 / 32	011	1.0	4096
1 / 64	100	2.0	8192
1 / 128	101	4.0	16384
1 / 256	110 or 111	8.0	32768

The FWDGT timeout can be more accurate by calibrating the IRC32K.

Note:

1. When after the execution of watchdog reload operation, if the MCU needs enter the deepsleep / standby mode immediately, more than 3 IRC32K clock intervals must be inserted in the middle of reload and deepsleep / standby mode commands by software setting;
2. 7 or more IRC32K clock cycles should be inserted between two consecutive dog feeds.

20.1.4 Register definition

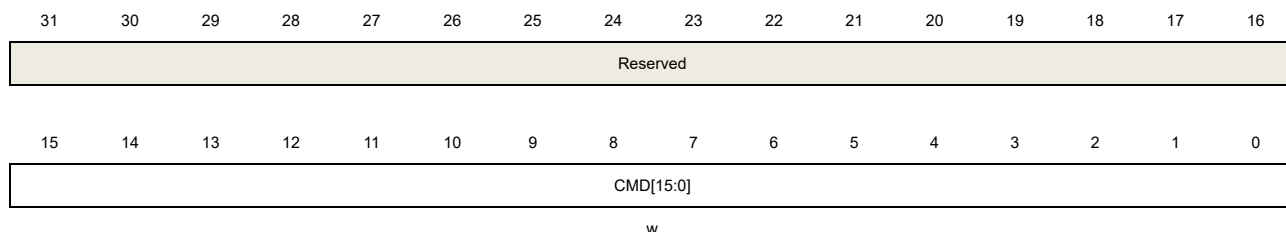
FWDGT base address: 0x4000 3000

Control register (FWDGT_CTL)

Address offset: 0x00

Reset value: 0x0000 0000

This register can be accessed by half-word (16-bit) or word (32-bit) access.



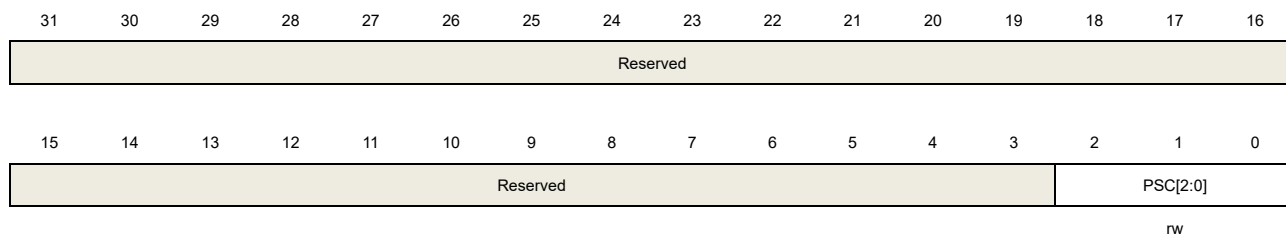
Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CMD[15:0]	Write only. Several different functions are realized by writing these bits with different values. 0x5555: Disable the FWDGT_PSC and FWDGT_RLD write protection. 0xCCCC: Start the free watchdog counter. When the counter reduces to 0, the free watchdog generates a reset or interrupt. 0xAAAA: Reload the counter.

Prescaler register (FWDGT_PSC)

Address offset: 0x04

Reset value: 0x0000 0000

This register can be accessed by half-word (16-bit) or word (32-bit) access.



Bits	Fields	Descriptions
31:3	Reserved	Must be kept at reset value.
2:0	PSC[2:0]	Free watchdog timer prescaler selection. Write 0x5555 to the FWDGT_CTL register before writing these bits. During a write operation to this register, the PUD bit in the FWDGT_STAT register is set and the value read from this register is invalid.

000: 1 / 4
001: 1 / 8
010: 1 / 16
011: 1 / 32
100: 1 / 64
101: 1 / 128
110: 1 / 256
111: 1 / 256

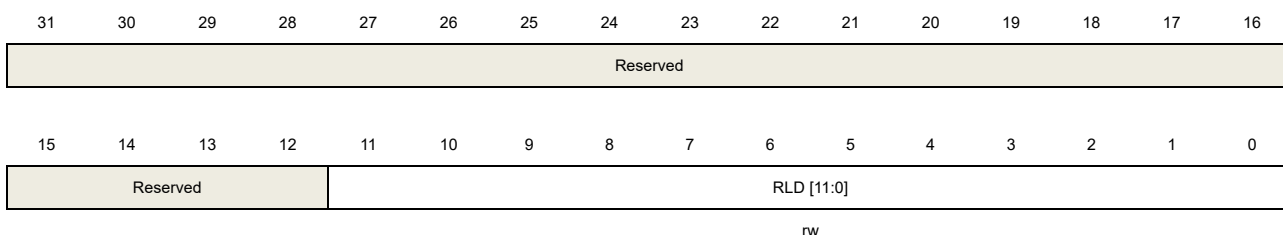
If several prescaler values are used by the application, it is mandatory to wait until PUD bit has been reset before changing the prescaler value. If the prescaler value has been updated, it is not necessary to wait until PUD has been reset before continuing code execution (Before entering low-power mode, it is necessary to wait until PUD is reset).

Reload register (FWDGT_RLD)

Address offset: 0x08

Reset value: 0x0000 0FFF

This register can be accessed by half-word (16-bit) or word (32-bit) access.



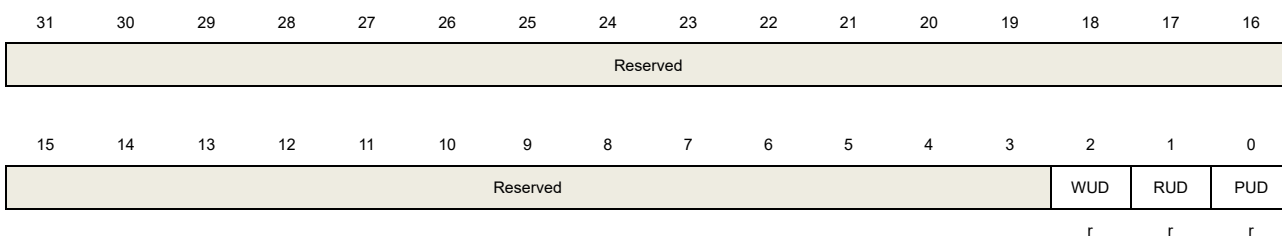
Bits	Fields	Descriptions
31:12	Reserved	Must be kept at reset value.
11:0	RLD[11:0]	Free watchdog timer counter reload value. Write 0xAAAA to the FWDGT_CTL register will reload the FWDGT counter with the RLD value. These bits are write protected. Write 0x5555 to the FWDGT_CTL register before writing these bits. During a write operation to this register, the RUD bit in the FWDGT_STAT register is set and the value read from this register is invalid. If several reload values are used by the application, it is mandatory to wait until RUD bit has been reset before changing the reload value. If the reload value has been updated, it is not necessary to wait until RUD has been reset before continuing code execution (Before entering low-power mode, it is necessary to wait until PUD is reset).

Status register (FWDGT_STAT)

Address offset: 0x0C

Reset value: 0x0000 0000

This register can be accessed by half-word(16-bit) or word(32-bit) access.



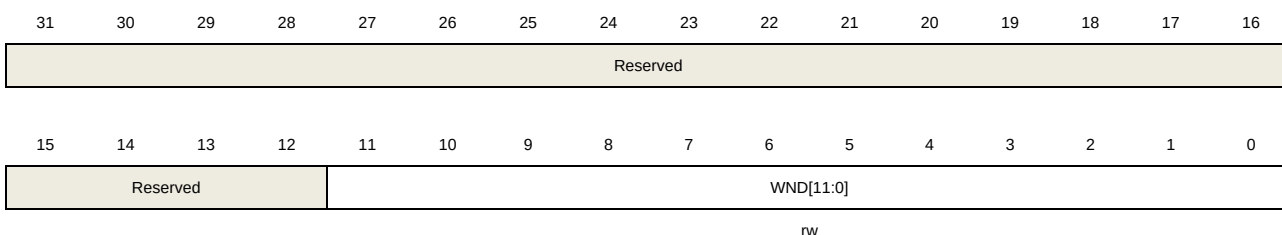
Bits	Fields	Descriptions
31:3	Reserved	Must be kept at reset value.
2	WUD	Watchdog counter window value update When a write operation to FWDGT_WND register ongoing, this bit is set and the value read from FWDGT_WND register is invalid.
1	RUD	Free watchdog timer counter reload value update During a write operation to FWDGT_RLD register, this bit is set and the value read from FWDGT_RLD register is invalid. This bit is reset by hardware after the update operation of the FWDGT_RLD register.
0	PUD	Free watchdog timer prescaler value update During a write operation to FWDGT_PSC register, this bit is set and the value read from FWDGT_PSC register is invalid. This bit is reset by hardware after the update operation of the FWDGT_PSC register.

Window register (FWDGT_WND)

Address offset: 0x10

Reset value: 0x0000 0FFF

This register can be accessed by half-word (16-bit) or word (32-bit).



Bits	Fields	Descriptions
31:12	Reserved	Must be kept at reset value.
11:0	WND[11:0]	Watchdog counter window value. These bits are used to contain the high limit of the window value to be compared to the downcounter. A reset or interrupt will occur if the reload operation is performed while the counter is greater than the value stored in this register. The WUD bit in the FWDGT_STAT register must be reset in order to be able to change the reload value.

These bits are write protected. Write 0x5555 in the FWDGT_CTL register before writing these bits.

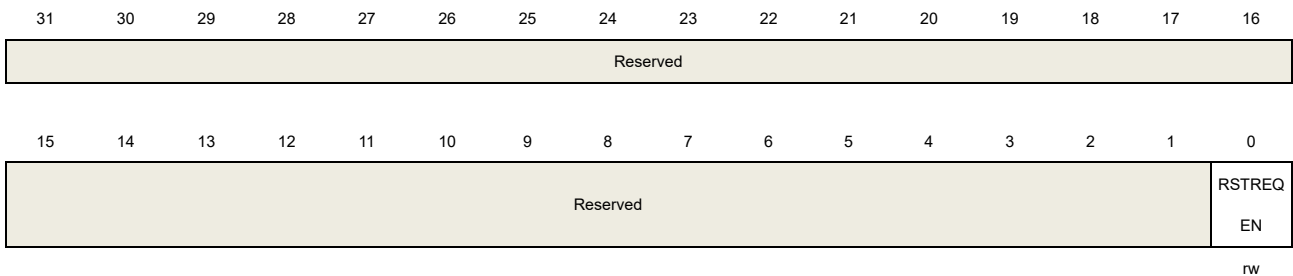
If several window values are used by the application, it is mandatory to wait until WUD bit has been reset before changing the window value. However, after updating the window value it is not necessary to wait until WUD is reset before continuing code execution except in case of low-power mode entry(Before entering low-power mode, it is necessary to wait until WUD is reset).

Reset control register (FWDGT_RSTCTL)

Address offset: 0x14

Reset value: 0x0000 0001

This register can be accessed by half-word(16-bit) or word(32-bit) access.



Bits	Fields	Descriptions
31:1	Reserved	Must be kept at reset value.
0	RSTREQEN	Free watchdog reset request enable 0: Interrupt or Non-maskable interrupt output request is enabled 1: Reset output request is enabled Note: When enabling an interrupt or non-maskable interrupt output request, after the independent watchdog counter generates an interrupt, the control register (FWDGT_CTL) must be reloaded before the independent watchdog counter can be started again.

20.2 Window watchdog timer (WWDGT)

20.2.1 Overview

The window watchdog timer (WWDGT) is used to detect system failures due to software malfunctions. After the window watchdog timer starts, the value of down counter reduces progressively. The watchdog timer causes a reset when the counter reached 0 (the CNT[13:0] = 0). The watchdog timer also causes a reset when the counter is refreshed before the counter reached the window start value (WIN in WWDGT_CFG) or after the counter reached the window end value (set by WEPS in WWDGT_CFG). So the software should refresh the counter in a limited window. The window watchdog timer generates an early wakeup status flag when the counter reaches 0x01 or the counter is written outside the refresh-permitted period. Interrupt occurs if it is enable.

The window watchdog timer clock is prescaled from the APB1 clock. The window watchdog timer is suitable for the situation that requires an accurate timing.

20.2.2 Characteristics

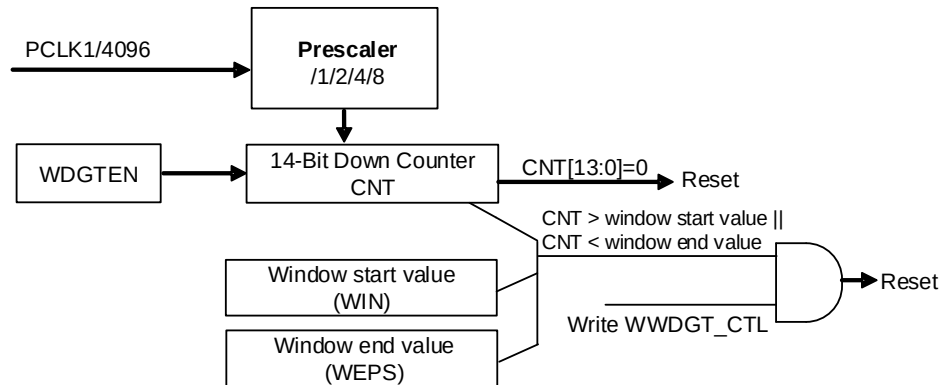
- Programmable free-running 14-bit down counter.
- Two start modes: hardware start mode and software start mode.
 - Hardware start: counting starts automatically after system reset occurs.
 - Software start: counting starts after refresh operation occurs.
- Generate a reset in two conditions when WWDGT is enabled:
 - Reset when the counter reached 0.
 - The counter is refreshed when the value of the counter is greater than the window register value or smaller than the window end value.
- Early wakeup interrupt (EWI): the watchdog is started and the interrupt is enabled, the interrupt occurs when the counter reaches 0x01 or the counter is refreshed out of refresh-permitted period.
- WWDGT debug mode, the WWDGT can stop or continue to work in debug mode.

20.2.3 Function overview

Block diagram

The window watchdog timer block refers to [Figure 20-2. Window watchdog timer block diagram](#).

Figure 20-2. Window watchdog timer block diagram



Software start mode

The software start mode is selected and the WWDGT_CFG register is enabled when the nWWDG_HW bit in option byte is 1.

After a reset signal occurs, configure the WWDGT_CFG register and then refresh the CNT[13:0] in WWDGT_CTL register to start down counter. The counting down speed depends on the APB1 clock and the prescaler value set by PSC[1:0] bits in the WWDGT_CFG register.

If the window watchdog timer reset is enabled (set the WDG TEN bit in the WWDGT_CTL), the watchdog timer causes a reset when the counter reaches 0 (the CNT[13:0] = 0), or the counter is refreshed when the counter is greater than the window start value (set by WIN in WWDGT_CFG) or the counter is refreshed when the counter is smaller than the window end value (set by WEPS in WWDGT_CFG).

If the early wakeup interrupt (EWI) is enabled by setting the EWIE bit in the WWDGT_CFG register, the interrupt will be generated when the counter reaches 0x01 (the CNT[13:0] = 0x01), or the counter is refreshed when the counter is greater than the window start value (set by WIN in WWDGT_CFG) or the counter is refreshed when the counter is smaller than the window end value (set by WEPS in WWDGT_CFG). The software can do something such as communication or data logging in the interrupt service routine (ISR) in order to analyze the reason of software malfunctions or save the important data before resetting the device. Moreover the software can reload the counter in ISR to manage a software system check and so on. In this case, the WWDGT will never generate a WWDGT reset but can be used for other things. The EWI interrupt flag is cleared by writing '0' to the EWIF bit in the WWDGT_STAT register.

Note: In software start mode, CNT in WWDGT_CTL is reset to 0 after an interrupt occurs, the software must reload the CNT value to restart the WWDGT counter.

Hardware start mode

The hardware start mode is selected and the configurations in option byte are enabled and the WWDGT_CFG register is disabled when the nWWDG_HW bit in option byte is 0.

During the reset signal state, the configurations value in option bytes are reload into WWDGT_CFG register (including WIN / WEPS / EWIE / PSC / WDG TEN). In Hardware start mode, WWDGT window start value (WIN) can only be set to four fixed value. When WWDGT_WSPS bits in option byte are set to 0b00, window start value is 0x0FFF (0x3FFF * 25%). When WWDGT_WSPS bits in option byte are set to 0b01, window start value is 0x1FFF (0x3FFF * 50%). When WWDGT_WSPS bits in option byte are set to 0b10, window start value is 0x2FFF (0x3FFF * 75%). When WWDGT_WSPS bits in option byte are set to 0b11, window start value is 0x3FFF (0x3FFF * 100%).

After a reset signal occurs, the down counter starts automatically from the WWDG_CNT[13:0] value in option byte.

If the window watchdog timer reset is enabled (the WWDGT_WDGTEN bit in the option bytes is set), the watchdog timer causes a reset when the counter reaches 0 (the CNT[13:0] = 0), or the counter is refreshed when the counter is greater than the window start value (set by WWDGT_WSPS in the option bytes) or the counter is refreshed when the counter is smaller than the window end value (set by WWDGT_WEPS in the option bytes).

If the early wakeup interrupt (EWI) is enabled (the WWDGT_EWIE bit in the option bytes is set), the interrupt will be generated when the counter reaches 0x01 (the CNT[13:0] = 0x01), or the counter is refreshed when the counter is greater than the window start value (set by WWDGT_WIN_OB in the option bytes) or the counter is refreshed when the counter is smaller than the window end value (set by WWDGT_WEPS in the option bytes). The software can do something such as communication or data logging in the interrupt service routine (ISR) in order to analyze the reason of software malfunctions or save the important data before resetting the device. Moreover the software can reload the counter in ISR to manage a software system check and so on. In this case, the WWDGT will never generate a WWDGT reset but can be used for other things. The EWI interrupt flag is cleared by writing '0' to the EWIF bit in the WWDGT_STAT register.

After a reset or interrupt occurs, the down-counter automatically starts counting down from the value set by the WWDG_CNT[13:0] bits in option byte.

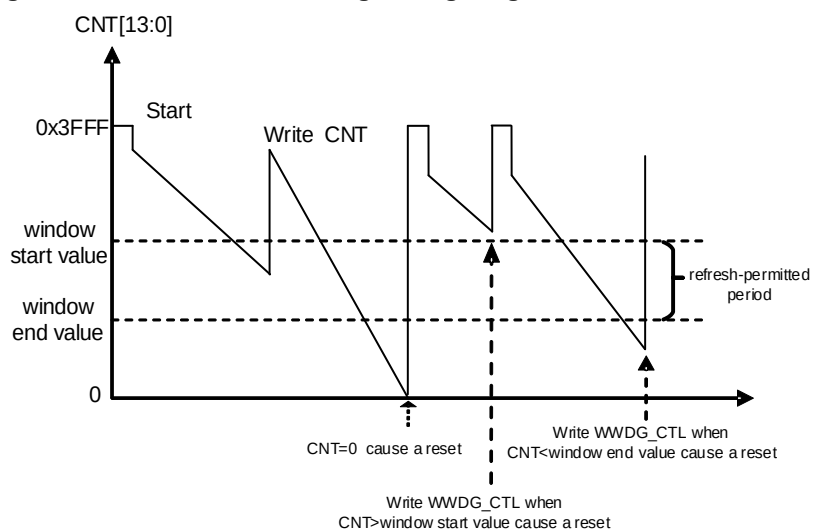
WWDGT_CFG register write operation

The WWDGT_CFG register can only be written once between the reset signal and the first refresh operation. After the refresh operation or writing the WWDGT_CFG register, the WWDGT_CFG register is protected and the protection is released after the WWDGT reset.

Timing and timeout

The WWDGT timing diagram refers to [Figure 20-3. Window watchdog timing diagram](#).

Figure 20-3. Window watchdog timing diagram



Calculate the WWDGT timeout by using the formula below.

$$t_{\text{WWDGT}} = t_{\text{PCLK1}} \times 4096 \times 2^{\text{PSC}} \times (\text{CNT}[13:0]) \quad (\text{ms}) \quad (20-1)$$

where:

t_{WWDGT} : WWDGT timeout

t_{PCLK1} : APB1 clock period measured in ms

The [Table 20-2. Min/max timeout value at 90 MHz \(\$f_{\text{PCLK1}}\$ \)](#) shows the minimum and maximum values of the t_{WWDGT} .

Table 20-2. Min/max timeout value at 90 MHz (f_{PCLK1})

Prescaler divider	PSC[1:0]	Min timeout value CNT[13:0] = 0x01	Max timeout value CNT[13:0] = 0x3FFF
1/1	00	45.5 μs	0.746 s
1/2	01	91.0 μs	1.491 s
1/4	10	182.0 μs	2.982 s
1/8	11	364.1 μs	5.965 s

If the WWDGT_HOLD bit in DBG module is cleared, the WWDGT continues to work even the Cortex®-M33 core halted (Debug mode). While the WWDGT_HOLD bit is set, the WWDGT stops in Debug mode.

20.2.4 Register definition

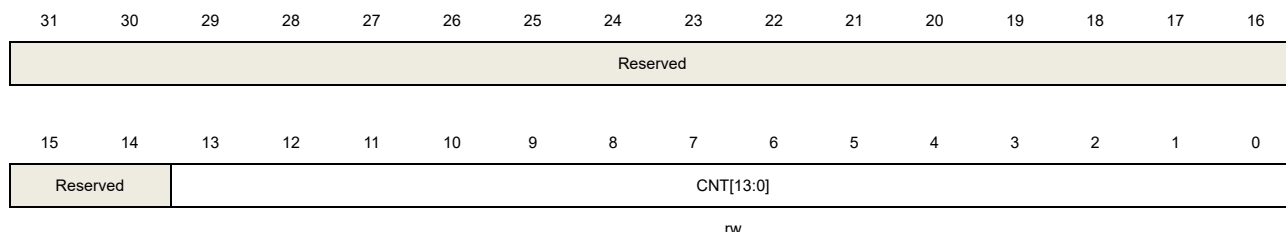
WWDGT base address: 0x4000 2C00

Control register (WWDGT_CTL)

Address offset: 0x00

Reset value: 0x0000 3FFF

This register can be accessed by half-word (16-bit) or word (32-bit).



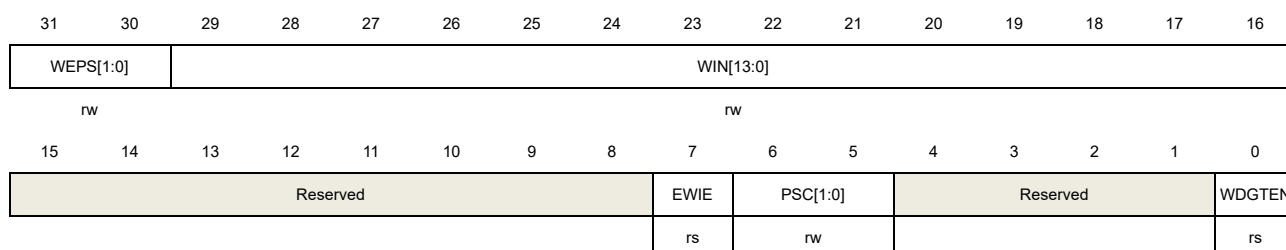
Bits	Fields	Descriptions
31:14	Reserved	Must be kept at reset value.
13:0	CNT[13:0]	The value of the watchdog timer counter. A reset occurs when the value of this counter decreases from 0x01 to 0x0. When the value of this counter is greater than the window starts value (WIN) or smaller than the window end value (WEPS), writing this counter also causes a reset. (In the software-start mode, counting down is started by refreshing CNT value) In hardware start mode, the reset value of CNT is set by option byte. In software start mode, the reset value of CNT is 0x3FFF.

Configuration register (WWDGT_CFG)

Address offset: 0x04

Reset value: 0xFFFF 0000

This register can only be accessed by word (32-bit).



Bits	Fields	Descriptions
31:30	WEPS	Window end position selection. If WDG TEN=1, a reset occurs if the watchdog counter (CNT bits in WWDGT_CTL) is written when the value of the watchdog counter is smaller than the window end position value.

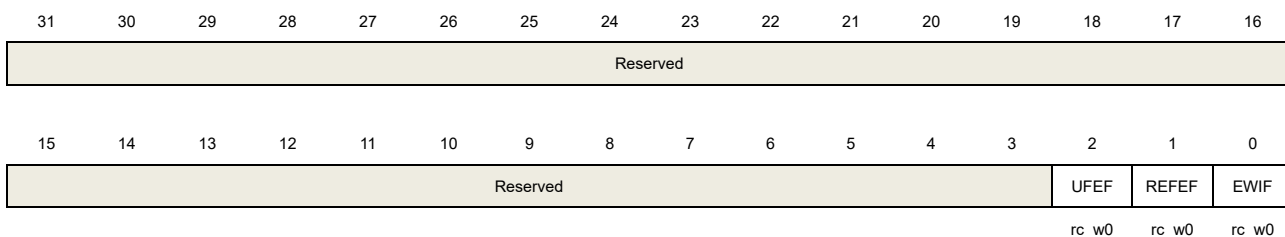
		00: Window end position value is 0x2FFF (75% * 0x3FFF)
		01: Window end position value is 0x1FFF (50% * 0x3FFF)
		10: Window end position value is 0x0FFF (25% * 0x3FFF)
		11: Window end position value is 0x0 (0% * 0x3FFF)
29:16	WIN[13:0]	The window start value. If WDG TEN=1, a reset occurs if the watchdog counter (CNT bits in WWDGT_CTL) is written when the value of the watchdog counter is greater than the window start value.
15:8	Reserved	Must be kept at reset value.
7	EWIE	Early wakeup interrupt enables. If the bit is set, an interrupt occurs when the counter reaches 0x01 or the watchdog counter (CNT bits in WWDGT_CTL) is written outside the refresh-permitted period. It can be cleared by a hardware reset or a software reset by setting the WWDGTRST bit of the RCU module. A write operation of '0' has no effect. EWIE bit is independent of WDG TEN bit. When EWIE is set to 1, WDG TEN must be set to 0.
6:5	PSC[1:0]	Prescaler. The time base of the watchdog timer counter. 00: (PCLK1 / 4096) / 1 01: (PCLK1 / 4096) / 2 10: (PCLK1 / 4096) / 4 11: (PCLK1 / 4096) / 8
4:1	Reserved	Must be kept at reset value.
0	WDGTEN	When WDG TEN=1, window watchdog timer can generate a reset when the counter reaches 0x00 or the watchdog counter (CNT bits in WWDGT_CTL) is written outside the refresh-permitted period. This bit is set by software and cleared by a hardware reset. Writing 0 has no effect. When WDG TEN is set to 1, EWIE must be set to 0. 0: Window watchdog timer don't generate reset 1: Window watchdog timer can generate reset

Status register (WWDGT_STAT)

Address offset: 0x08

Reset value: 0x0000 0000

This register can be accessed by half-word (16-bit) or word (32-bit).



Bits	Fields	Descriptions

31:3	Reserved	Must be kept at reset value.
2	UFEF	Underflow error flag. When the down-counter reaches 0x01, and WWDGT is configured to generate interrupt (WWDGT_EWIE in option byte WWDGT_0 or EWIE in WWDGT_CFG is set), UFEF is set by hardware. This bit is cleared by writing 0 or a reset to WWDG. There is no effect when writing 1.
1	REFEF	Refresh error flag. When the counter (CNT bits in WWDGT_CTL) is written outside the refresh-permitted period, and WWDGT is configured to generate interrupt (WWDGT_EWIE in option byte WWDGT_0 or EWIE in WWDGT_CFG is set), REFEF is set by hardware. This bit is cleared by writing 0 or a reset to WWDGT. There is no effect when writing 1.
0	EWIF	Early wakeup interrupt flag. When the counter reaches 0x01 or the counter (CNT bits in WWDGT_CTL) is written outside the refresh-permitted period, this bit is set by hardware even the interrupt is not enabled (WWDGT_EWIE in option byte WWDGT_0 or EWIE in WWDGT_CFG is cleared). This bit is cleared by writing 0 or a reset to WWDG. There is no effect when writing 1.

21 TIMER (TIMERx)

Table 21-1. Timers (TIMERx) are divided into two sorts

TIMER	TIMER0/7	TIMER1/2
TYPE	Advanced	General-L0
Prescaler	16-bit	16-bit
Counter	16-bit	16-bit
Count mode	UP, DOWN, Center-aligned	UP, DOWN, Center-aligned
Repetition	•	×
Channel Capture/ Compare	8	4
Composite PWM mode	•	×
Asymmetric PWM mode	•	×
Output match pulse select	•	•
Complementary & Dead-time	•	×
Break	•	×
Single Pulse	•	•
Quadrature decoder	•	•
decoder	•	•
Master-slave management	•	•
Inter Connection	• ⁽¹⁾	• ⁽²⁾
Dedicated ADC triggers	•	×
DMA	•	•
Debug Mode	•	•

	TIMERx	ITI0	ITI1	ITI2	ITI3
(1)	TIMER0	TIMER7_TRGO	TIMER1_TRGO	TIMER2_TRGO	EVIC
	TIMER7	TIMER0_TRGO	TIMER1_TRGO	TIMER2_TRGO	EVIC
(2)	TIMER1	TIMER0_TRGO	TIMER7_TRGO	TIMER2_TRGO	EVIC
	TIMER2	TIMER0_TRGO	TIMER7_TRGO	TIMER1_TRGO	EVIC

21.1 Advanced timer (TIMERx, x=0, 7)

21.1.1 Overview

The advanced timer module (TIMER0/7) is an eight-channel timer that supports both input capture and output compare. They can generate PWM signals to control motor or be used for power management applications. The advanced timer has a 16-bit counter that can be used as an unsigned counter.

In addition, the advanced timers can be programmed and be used for counting, their external events can be used to drive other timers.

Timer also includes a dead-time insertion module which is suitable for motor control applications.

Timers are completely independent with each other, but they may be synchronized to provide a larger timer with their counter value increasing in unison.

21.1.2 Characteristics

- Total channel num: 8.
- Counter width: 16 bits.
- Selectable clock source: internal clock, internal trigger, external input, external trigger.
- Multiple counter modes: up counting, down counting and center-aligned counting.
- Quadrature decoder: used for motion tracking and determination of both rotation direction and position.
- Hall sensor function: used for 3-phase motor control.
- Programmable prescaler: 16 bits. The factor can be changed ongoing.
- Each channel is independent and user-configurable: input capture mode, output compare mode, programmable PWM mode, single pulse mode and trigger out.
- Programmable dead time insertion.
- Auto reload function.
- Programmable counter repetition function.
- Break input function.
- Interrupt output or DMA request: update event, trigger event, compare/capture event and break input.
- Daisy chaining of timer module allows a single timer to start multiple timers.
- Timer synchronization allows the selected timers to start counting on the same clock cycle.
- Timer master-slave management.

21.1.3 Block diagram

[Figure 21-1. Advanced timer block diagram](#) provides details of the internal configuration of the advanced timer, and [Table 21-2. Advanced timer channel description](#) introduces the input and output of the channels.

Figure 21-1. Advanced timer block diagram

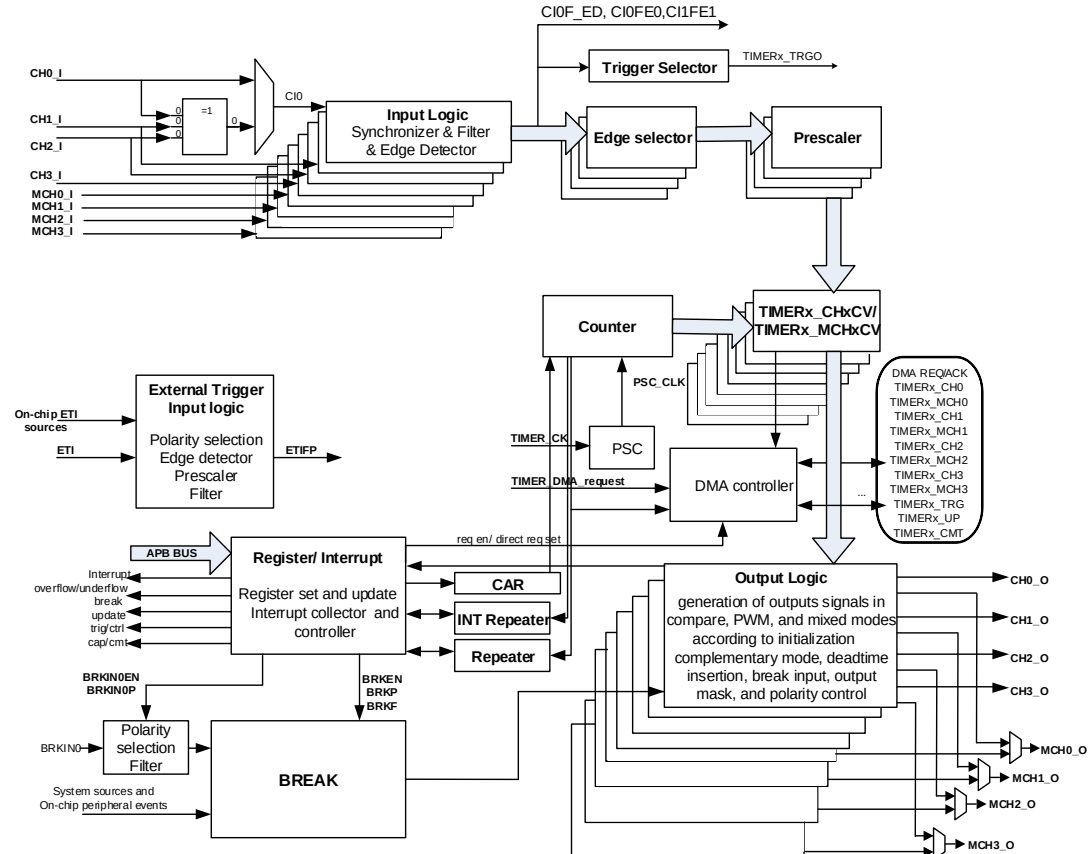


Table 21-2. Advanced timer channel description

Channel name (x=0..3)	MCHxMSEL[1:0]=00 independent mode	MCHxMSEL[1:0]=11 complementary mode
CHx (Channel x)	CHx and MCHx can independently input capture and compare output	only the CHx is valid for input, and the outputs of MCHx and CHx are complementary
MCHx (Multi mode channel x)		

21.1.4 Function overview

Clock selection

The clock source of the advanced timer can be either the CK_TIMER or an alternate clock source controlled by TSCFGy[4:0] (y=0..7,9..12,15) in SYSCFG_TIMERxCFG(x=0,7) registers.

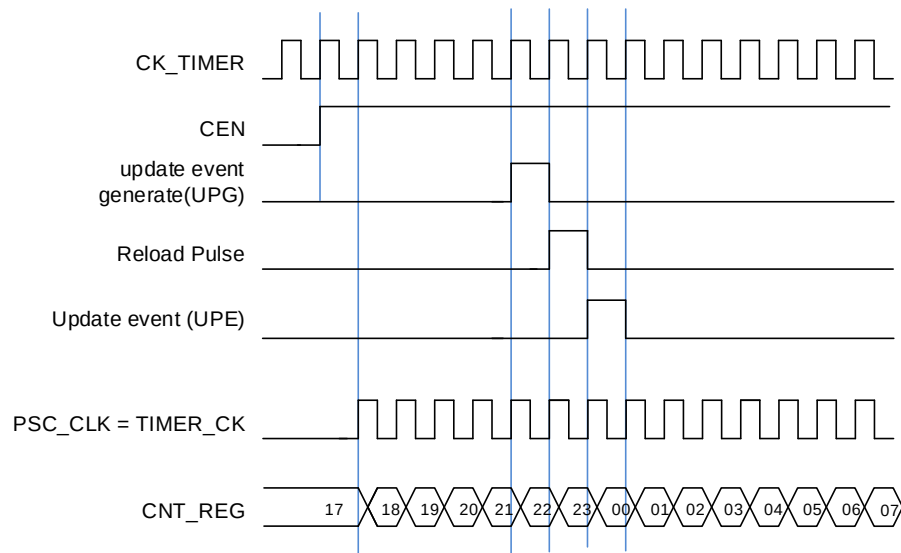
- TSCFGy[4:0] (y=0..7,9..12,15) = 5'b00000 in SYSCFG_TIMERxCFG(x=0,7) registers. Internal clock CK_TIMER is selected as timer clock source which is from module RCU.

The default clock source is the CK_TIMER for driving the counter prescaler when TSCFGy[4:0] (y=0..7,9..12,15) = 5'b00000 in SYSCFG_TIMERxCFG(x=0,7) registers. When the CEN is set, the CK_TIMER will be divided by PSC value to generate PSC_CLK.

In this mode, the TIMER_CK which drives counter's prescaler to count is equal to CK_TIMER which is from RCU module.

If TSCFGy[4:0] (y=0..2,6,9..12) in SYSCFG_TIMERxCFG(x=0,7) registers are setting to a nonzero value, the prescaler is clocked by other clock sources selected in the TSCFGy[4:0] (y=0..2,6,9..12) bit-field, more details will be introduced later. When the TSCFGy[4:0] (y=3,4,5,7) are setting to a nonzero value, the internal clock TIMER_CK is the counter prescaler driving clock source.

Figure 21-2. Normal mode, internal clock divided by 1



- TSCFG6[4:0] are setting to a nonzero value (external clock mode 0). External input pin is selected as timer clock source.

The TIMER_CK, which drives counter's prescaler to count, can be triggered by the event of rising or falling edge on the external pin TIMEx_CHn / TIMEx_MCHn(n=0..3). This mode can be selected by setting TSCFG6[4:0] to 0x5~0x7 and 0x9~0xE.

And, the counter prescaler can also be driven by rising edge on the internal trigger input pin ITI0~ITI4. This mode can be selected by setting TSCFG6[4:0] to 0x1~0x4.

- SMC1= 1'b1 (external clock mode 1). External input ETI is selected as timer clock source.

The TIMER_CK, which drives counter's prescaler to count, can be triggered by the event of rising or falling edge on the external pin ETI. This mode can be selected by setting the SMC1 bit in the TIMEx_SMCFG register to 1. The other way to select the ETI signal as the clock

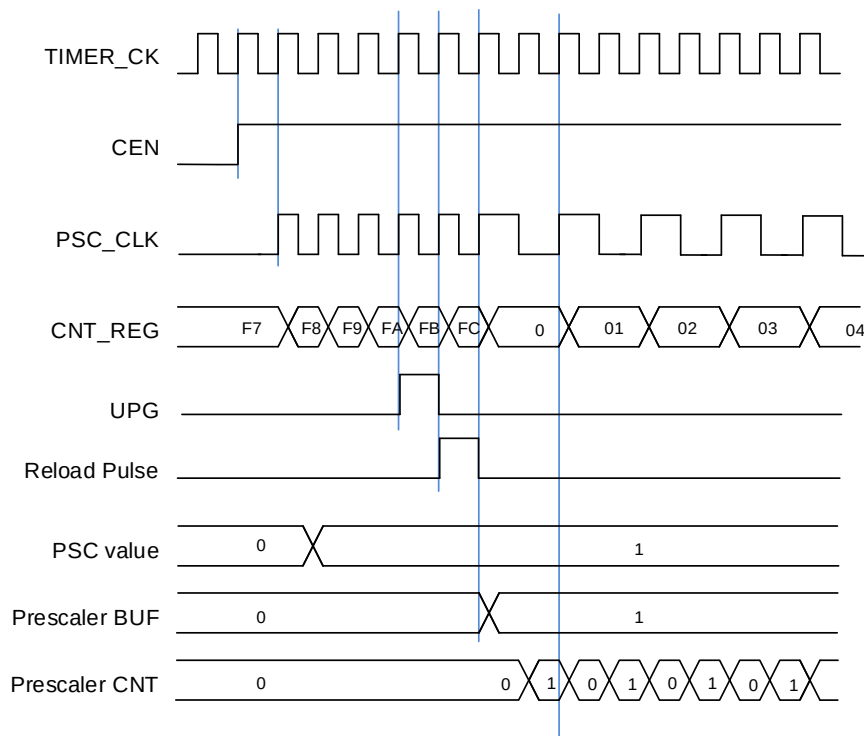
source is setting the TSCFG6[4:0] to 0x8. Note that the ETI signal is derived from the ETI pin sampled by a digital filter. When the ETI signal is selected as the clock source, the trigger controller including the edge detection circuitry will generate a clock pulse on each ETI signal rising edge to clock the counter prescaler.

Note: The ETI signal can be input from an external ETI pin or provide by on-chip peripherals, please refer to ETISEL[1:0] bits in TIMEx_CTL0 register for more details.

Clock prescaler

The prescaler can divide the timer clock (TIMER_CK) to a counter clock (PSC_CLK) by any factor ranging from 1 to 65536. It is controlled by prescaler register (TIMEx_PSC) which can be changed ongoing, but it is adopted at the next update event.

Figure 21-3. Counter timing diagram with prescaler division change from 1 to 2



Up counting mode

In this mode, the counter counts up continuously from 0 to the counter reload value, which is defined in the TIMEx_CAR register, in a count-up direction. Once the counter reaches the counter reload value, the counter restarts from 0. If the repetition counter is set, the update event will be generated after (TIMEx_CREP+1) times of overflow. Otherwise the update event is generated each time when counter overflows. The counting direction bit DIR in the TIMEx_CTL0 register should be set to 0 for the up-counting mode.

Whenever, if the update event software trigger is enabled by setting the UPG bit in the TIMEx_SWEVG register, the counter value will be initialized to 0 and an update event will

be generated.

If the UPDIS bit in TIMEx_CTL0 register is set, the update event is disabled.

When an update event occurs, the prescaler register and repetition counter register are updated. In contrast, when the auto reload register is updated with the preload value is controlled by the shadow register. More details will be presented in the [Shadow register loading function](#).

[Figure 21-4. Timing diagram of up counting mode, PSC=0/2](#) and [Figure 21-5. Timing diagram of up counting mode, change TIMEx_CAR ongoing \(ARSE=0, CAR value updates immediately. ARSE=1, CARUPSEL =2'b00, CAR value updates when an update event occurs.\)](#) show some examples of the counter behavior for different clock prescaler factors when TIMEx_CAR=0x99.

Figure 21-4. Timing diagram of up counting mode, PSC=0/2

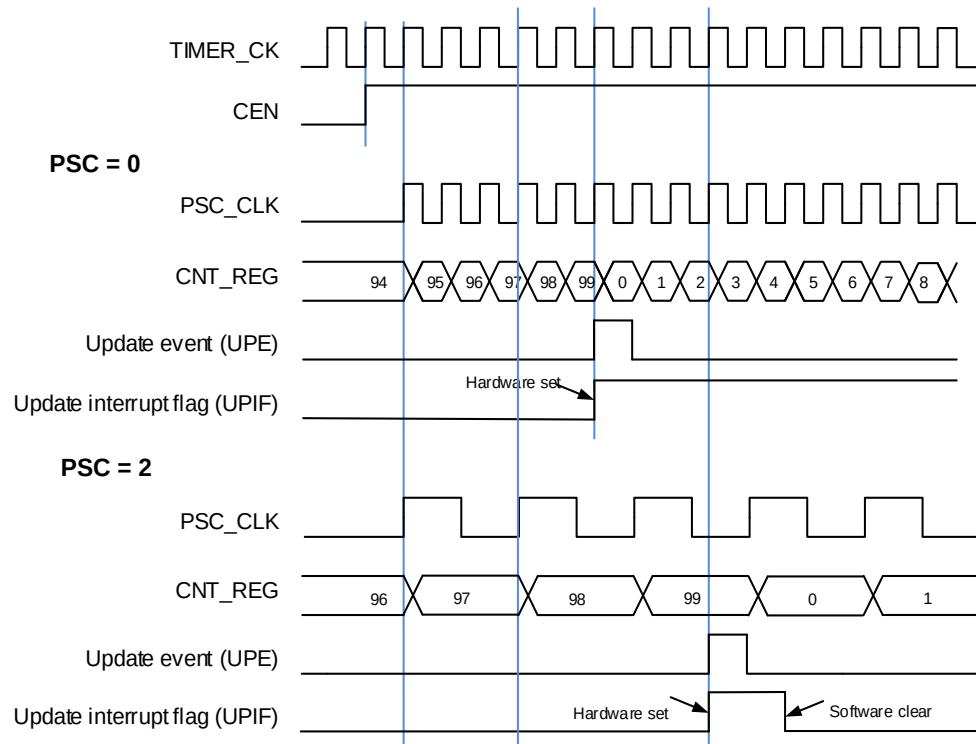
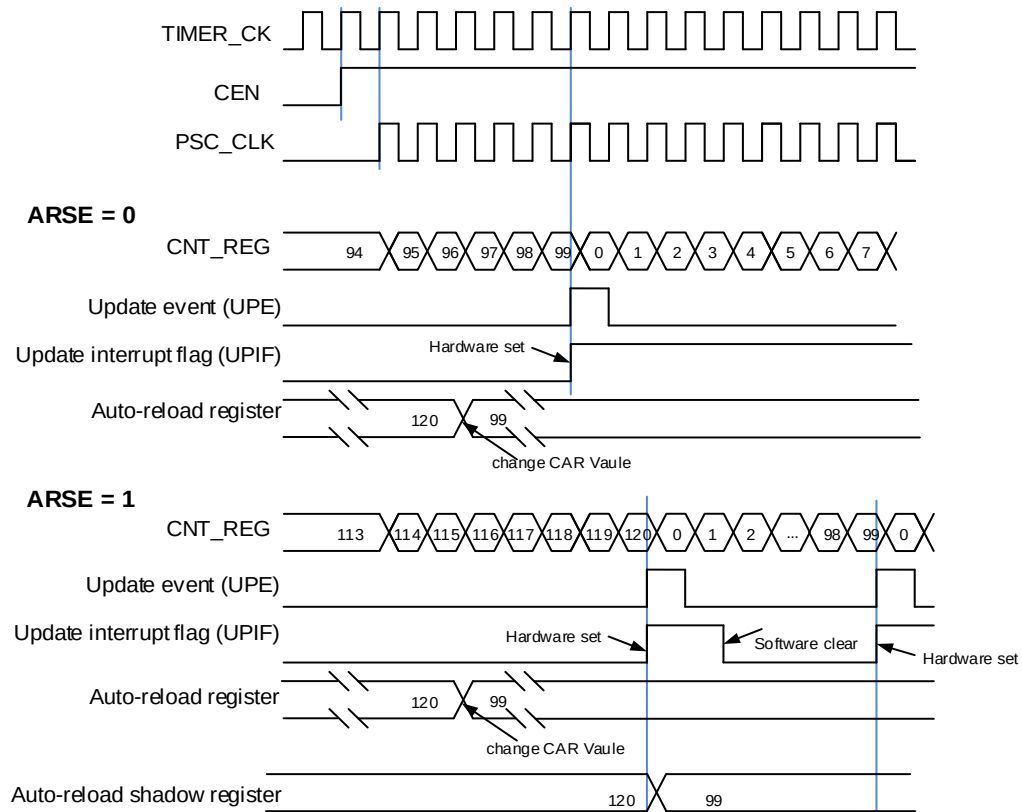


Figure 21-5. Timing diagram of up counting mode, change TIMEx_CAR ongoing (ARSE=0, CAR value updates immediately. ARSE=1, CARUPSEL =2'b00, CAR value

updates when an update event occurs.)



Down counting mode

In this mode, the counter counts down continuously from the counter reload value, which is defined in the `TIMERx_CAR` register, in a count-down direction. Once the counter reaches 0, the counter restarts to count again from the counter reload value. If the repetition counter is set, the update event will be generated after $(\text{TIMERx\_CREP}+1)$ times of underflow. Otherwise, the update event is generated each time when counter underflows. The counting direction bit `DIR` in the `TIMERx_CTL0` register should be set to 1 for the down counting mode.

When the update event is set by the `UPG` bit in the `TIMERx_SWEVG` register, the counter value will be initialized to the counter reload value and an update event will be generated.

If the `UPDIS` bit in `TIMERx_CTL0` register is set, the update event is disabled.

When an update event occurs, the prescaler register and repetition counter register are updated. In contrast, when the auto reload register is updated with the preload value is controlled by configured registers by the shadow register. More details will be presented in the [Shadow register loading function](#).

[Figure 21-6. Timing diagram of down counting mode, \$\text{PSC}=0/2\$](#) and [Figure 21-7. Timing diagram of down counting mode, change `TIMERx\_CAR` ongoing \(\$\text{ARSE}=0\$, `CAR` value updates immediately. \$\text{ARSE}=1\$, `CARUPSEL` = 2'b00, `CAR` value updates when an update event occurs.\)](#) show some examples of the counter behavior in different clock frequencies

when $\text{TIMERx\_CAR} = 0x99$.

Figure 21-6. Timing diagram of down counting mode, $\text{PSC}=0/2$

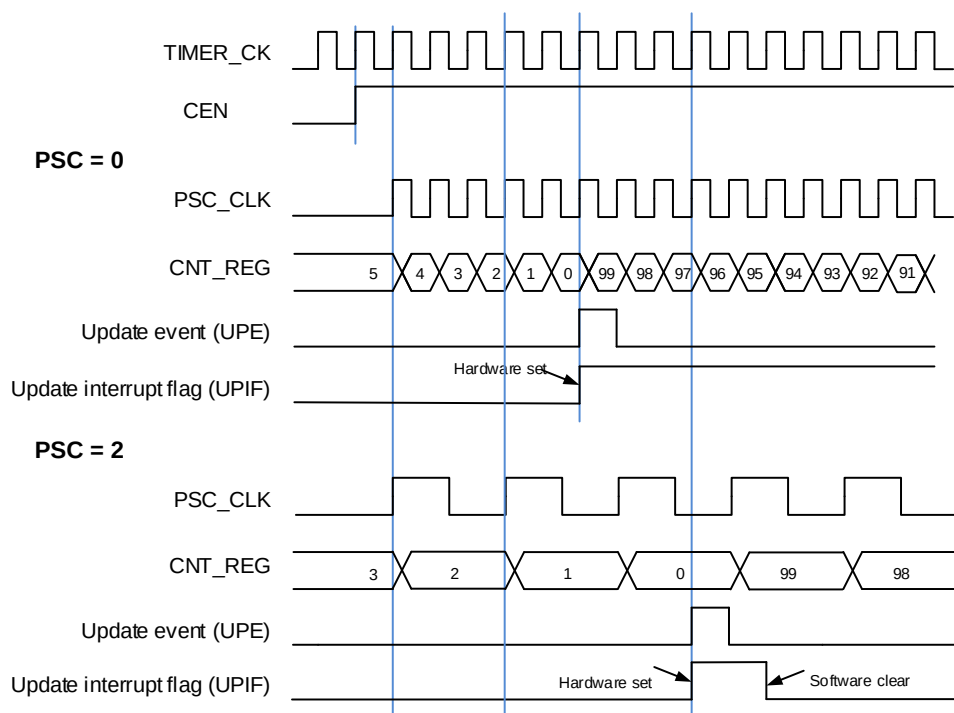
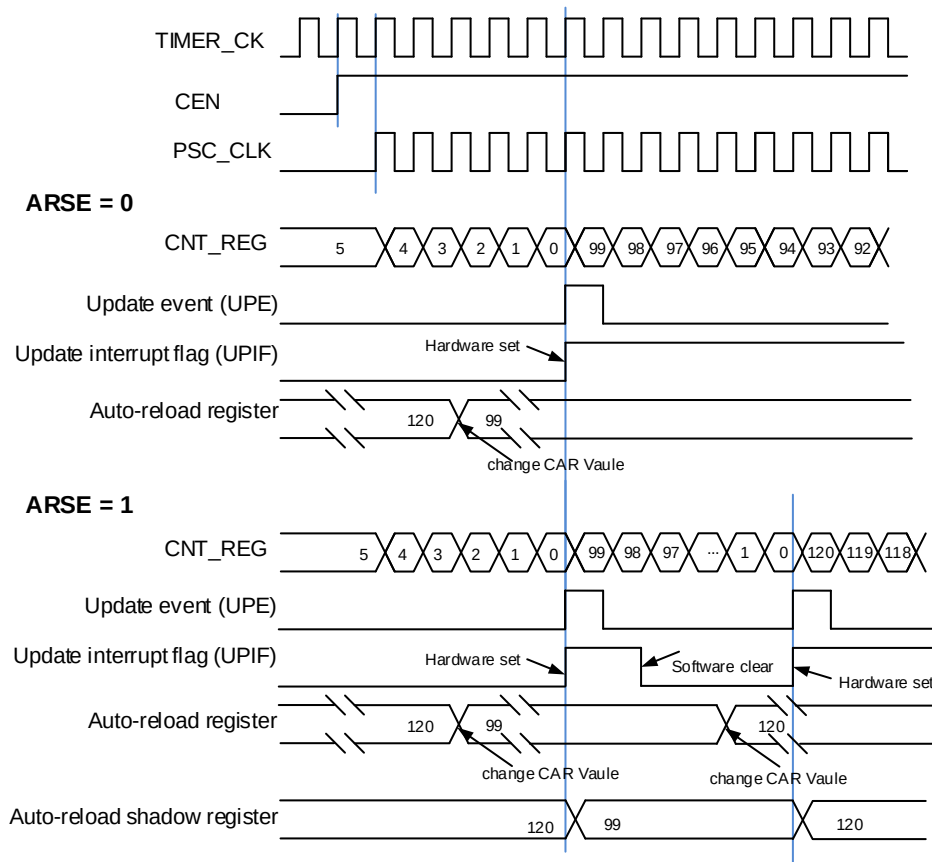


Figure 21-7. Timing diagram of down counting mode, change TIMERx_CAR ongoing ($\text{ARSE}=0$, CAR value updates immediately. $\text{ARSE}=1$, $\text{CARUPSEL} = 2'b00$, CAR value

updates when an update event occurs.)



Center-aligned counting mode

In the center-aligned counting mode, the counter counts up from 0 to the counter reload value and then counts down to 0 alternatively. The timer module generates an overflow event when the counter counts to (TIMERx_CAR-1) in the count-up direction and generates an underflow event when the counter counts to 1 in the count-down direction. The counting direction bit DIR in the TIMERx_CTL0 register is read-only and indicates the counting direction when in the center-aligned counting mode. The counting direction is updated by hardware automatically.

Setting the UPG bit in the TIMERx_SWEVG register will initialize the counter value to 0 and generate an update event irrespective of whether the counter is counting up or down in the center-aligned counting mode.

The UPIF bit in the TIMERx_INTF register will be set to 1 either when an underflow event or an overflow event occurs. While the CHxIF bit is associated with the value of CAM[1:0] in TIMERx_CTL0. In the center-aligned counting mode, dedicated underflow or overflow interrupt can be generated and the OVERFLOWIF or UNDERFLOWIF bit in the TIMERx_INTF register will be set to 1. The details refer to [Figure 21-8. Timing diagram of center-aligned counting mode.](#)

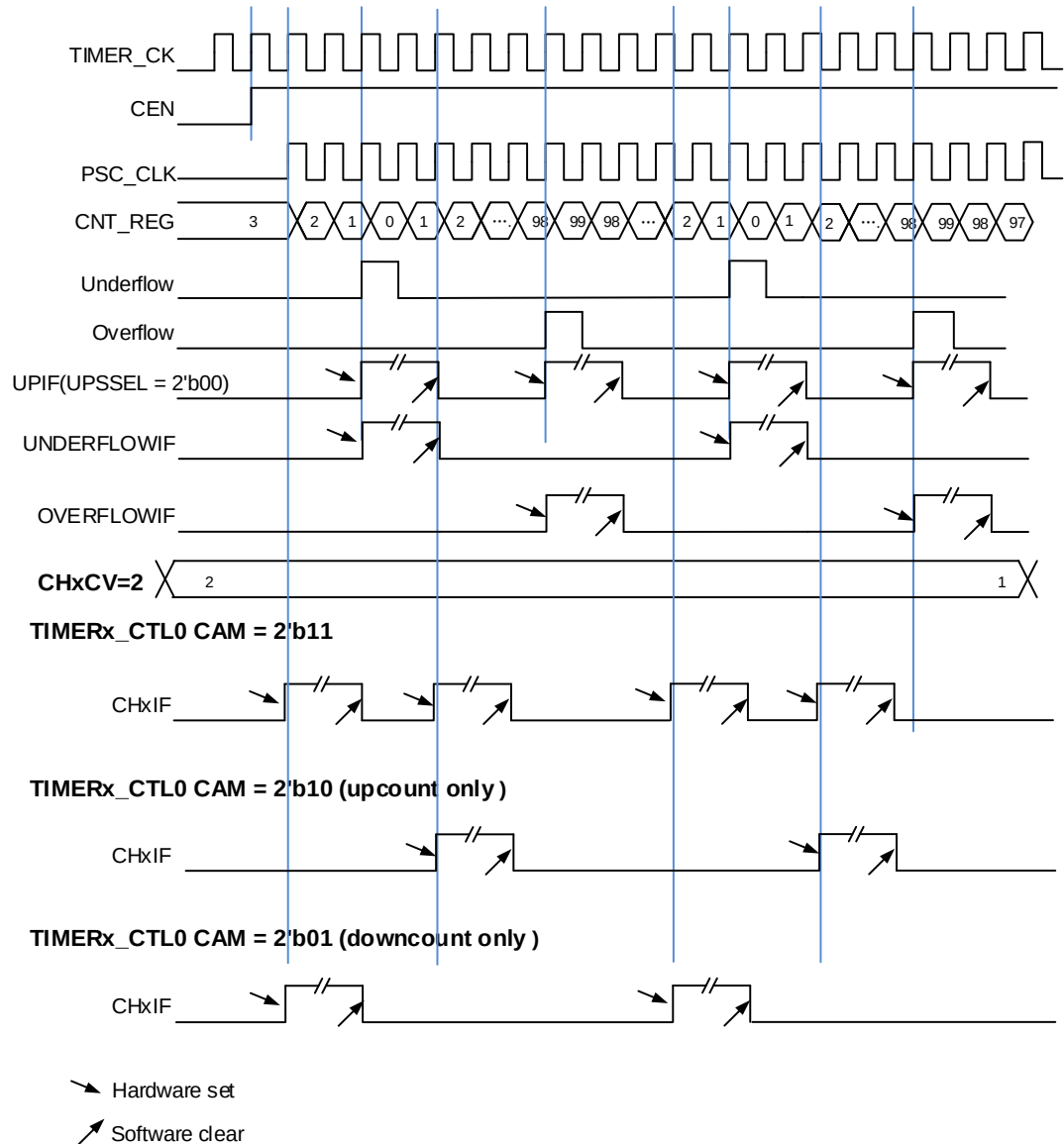
If the UPDIS bit in the TIMERx_CTL0 register is set, the update event is disabled.

When an update event occurs, the prescaler register and repetition counter register are updated. In contrast, when the auto reload register is updated with the preload value is controlled by configured registers by the shadow register. More details will be presented in the [*Shadow register loading function*](#).

In output compare mode or ADC trigger mode, the compare registers and additional compare registers (CHxCV / MCHxCV / CHxCOMVAL_ADD (x=0..3)) used to compare matches, and the two compare registers (TIMERx_ADCCR1 / TIMERx_ADCCR2) dedicated to generating ADC triggers, are also controlled by the shadow register. In the center-aligned counting mode, these registers with shadow registers will select the counter update, the counter overflow, the counter underflow, the counter overflow and underflow, update event or immediate update preload values base on the configured registers by the shadow register. More details will be presented in the [*Shadow register loading function*](#).

[*Figure 21-8. Timing diagram of center-aligned counting mode*](#) shows some examples of the counter behavior when TIMERx_CAR=0x99, TIMERx_PSC=0x0, TIMERx_CREP=0x0.

Figure 21-8. Timing diagram of center-aligned counting mode



Counter repetition

The advance timer has repetition counter `TIMERx_CREP`, which can be used for generating update events and the `TIMERx_IREP`, which can be used for generating overflow or underflow interrupt.

Counter repetition for generating update events

Repetition counter is used to generate the update event or update the timer registers only after a given number (N+1) cycles of the counter, where N is the value of CREP bits in `TIMERx_CREP` register. The repetition counter is decremented at each counter overflow in up counting mode, at each counter underflow in down counting mode.

In center-aligned counting mode, the repetition counter is decremented at each counter overflow, or at each counter underflow, or at each counter overflow and underflow, selected

according to the update source counter flow selection register (UPSSSEL[1:0]).

Setting the UPG bit in the TIMEx_SWEVG register will reload the content of CREP in TIMEx_CREP register and generate an update event.

In center-aligned counting mode, if both the counter overflow and underflow are selected to decrement the repeat counter, for odd values of CREP, the update event occurs either on the overflow or on the underflow depending on when the CREP register was written and when the counter was started. The update event is generated at overflow when the CREP was written before starting the counter and generated at underflow when the CREP was written after starting the counter.

Figure 21-9. Update repetition counter timing diagram of center-aligned counting mode

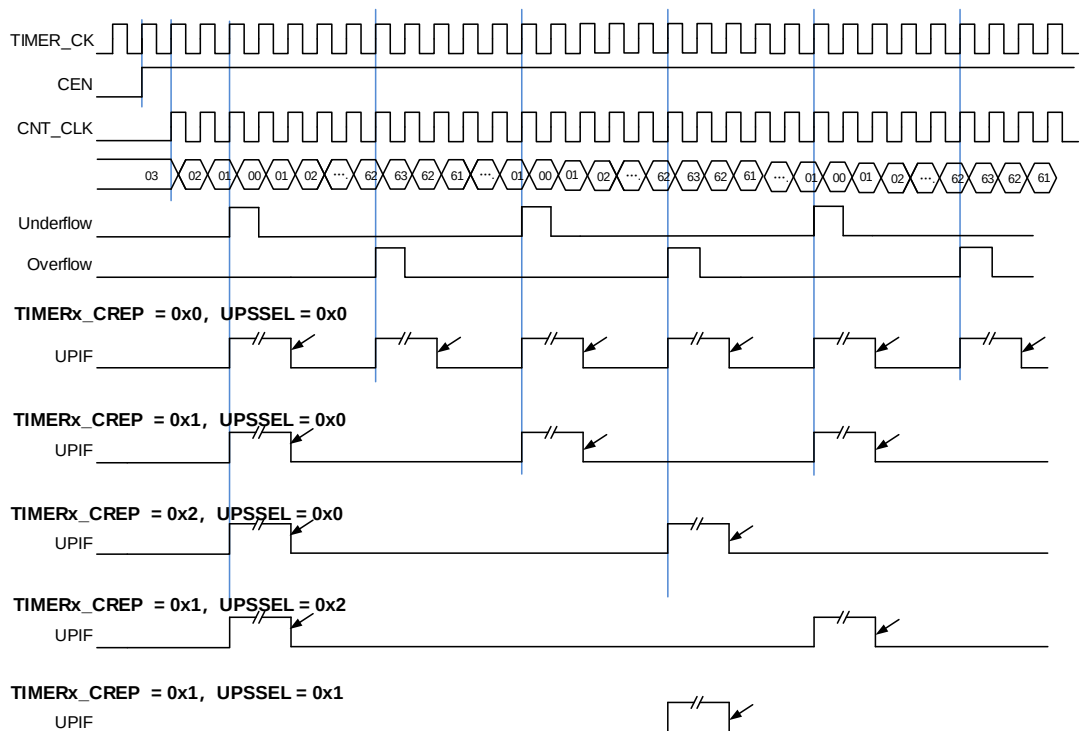


Figure 21-10. Update repetition counter timing diagram of up counting mode

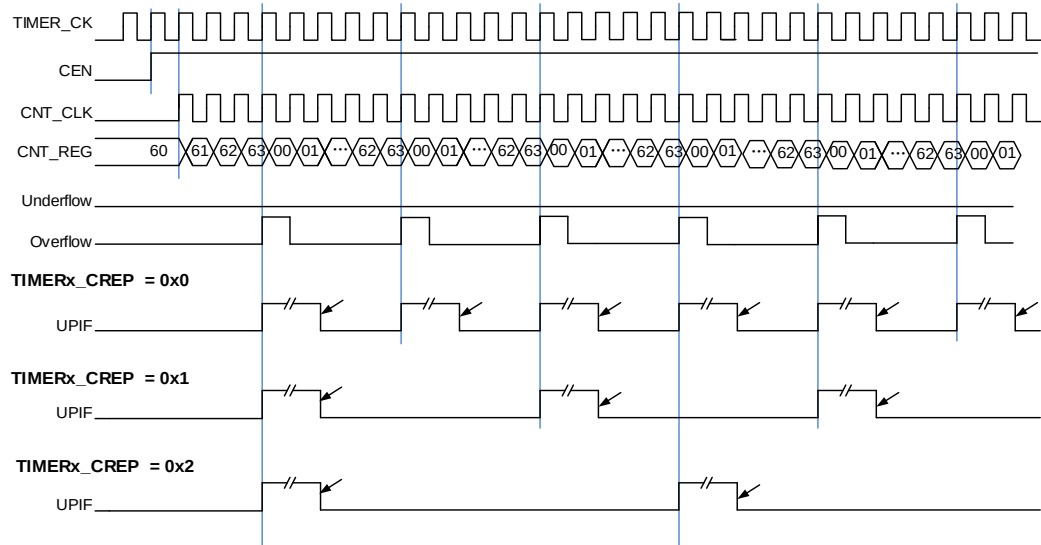
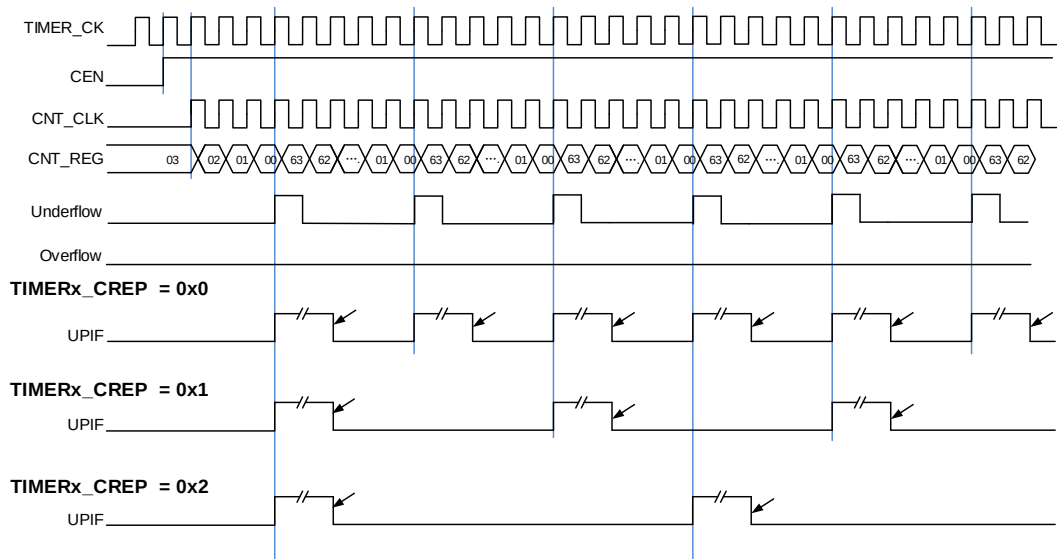


Figure 21-11. Update repetition counter timing diagram of down counting mode



Counter repetition for generating underflow or overflow interrupt

Interrupt repetition counter (TIMERx_IREP) is used to generate the underflow or overflow interrupt only after a given number (N+1) cycles of the counter, where N is the value of IREP bit in TIMERx_IREP register. The interrupt repetition counter is decremented at each counter overflow in up counting mode, at each counter underflow in down counting mode.

In center-aligned counting mode, the interrupt repetition counter is decremented at each counter overflow, or at each counter underflow, or at each counter overflow and underflow, selected according to the interrupt counter flow selection register (INTCOSEL[1:0]).

In center-aligned counting mode, the overflow or underflow interrupt will be determined based on the overflow flag (OVERFLOWIF) or underflow flag (UNDERFLOWIF) respectively.

Figure 21-12. Interrupt repetition counter timing diagram of center-aligned counting mode

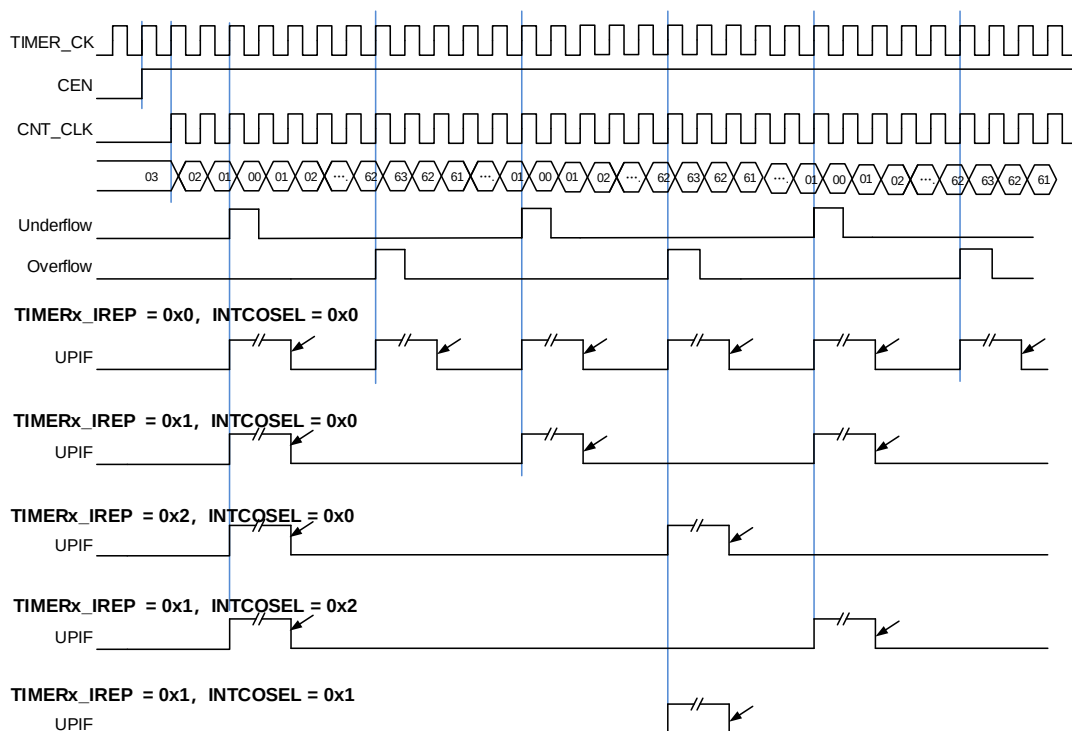


Figure 21-13. Interrupt repetition counter timing diagram of up counting mode

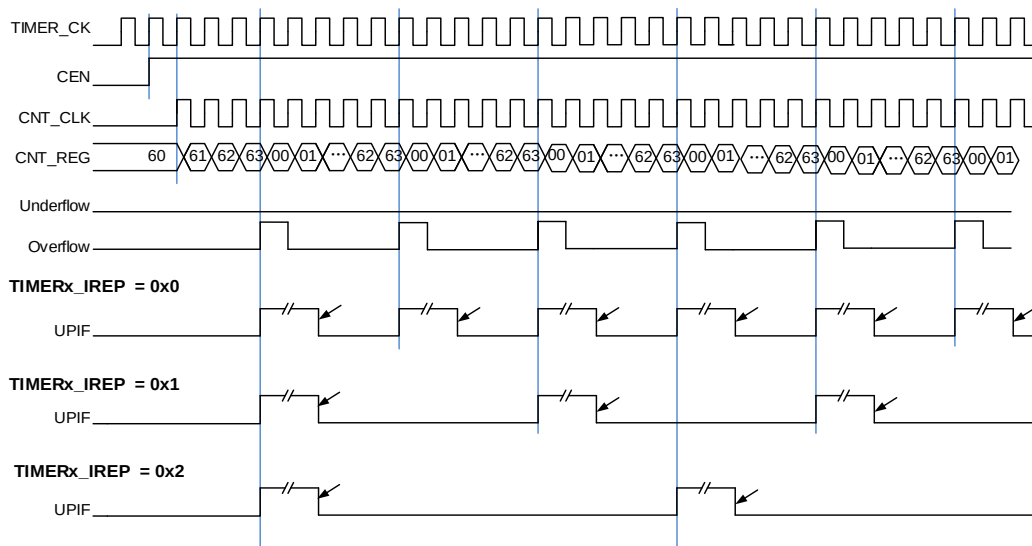
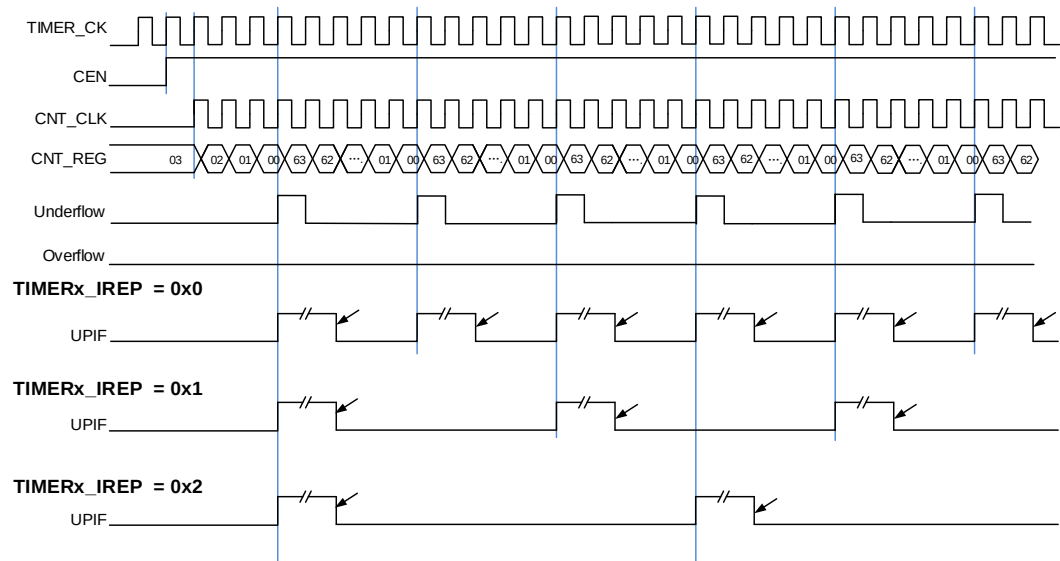


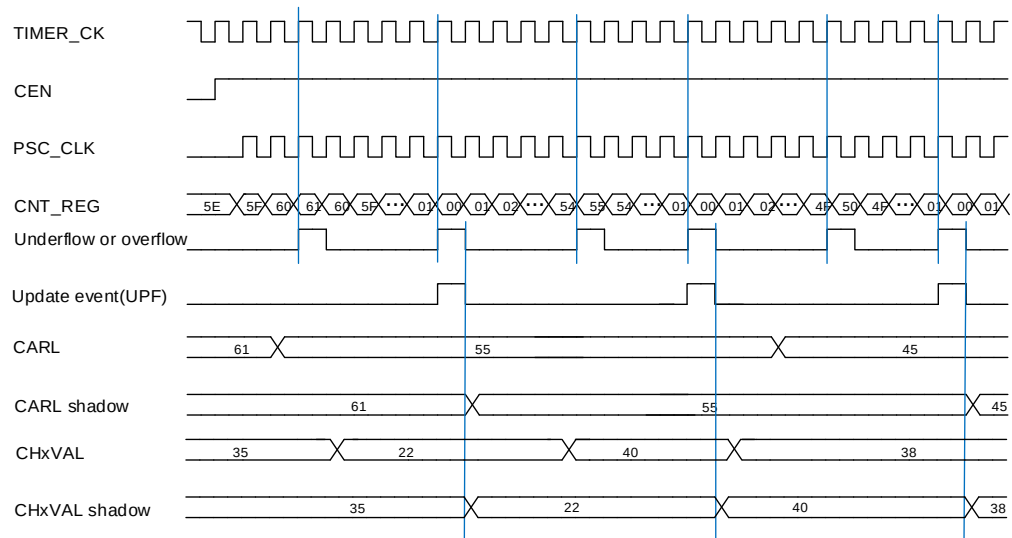
Figure 21-14. Interrupt repetition counter timing diagram of down counting mode



Shadow register loading function

TIMERx_CAR, TIMERx_CHxCV, TIMERx_CHxCOMV_ADD, TIMERx_ADCCR1 and TIMERx_ADCCR2 (x = 0, 7) all have separate shadow function enable bit. When shadow function is disabled, the corresponding register with shadow register function will immediately be overloaded with the newly configured value, and the timer will operate with the newly configured value. Registers with shadow register function also have a separate 2-bit update mode selection in TIMERx_SHUPM0 and TIMERx_SHUPM1 register. When shadow function is enabled, each shadow register can be selected to enable global update function, update the next counter overflow time, underflow update, or both. The global update function can achieve unified updates of selected shadow registers at update event.

Figure 21-15. Timing diagram for updates in global update mode on update events shows CARL and CHxCV (x = 0..3) update timing diagram in global update mode. (ARSE = 1 and CH0COMSEN = 1).

Figure 21-15. Timing diagram for updates in global update mode on update events

Capture/compare channels

The advanced timer has eight independent channels which can be used as capture inputs or compare outputs. Each channel is built around a channel capture compare register including an input stage, a channel controller and an output stage.

When the channels are used for input, channel x and multi mode channel x can perform input capture independently; when the channels are used for comparison output, the channel x and multi mode channel x can output independent and complementary outputs.

Input capture mode

When $MCHxMSEL=2'b00$ (independent mode), channel x and multi mode channel x can perform input capture independently.

Input capture mode allows the channel to perform measurements such as pulse timing, frequency, period, duty cycle and so on. The input stage consists of a digital filter, a channel polarity selection, edge detection and a channel prescaler. When a selected edge occurs on the channel input, the current value of the counter is captured into the $TIMERx\_CHxCV/$ $TIMERx\_MCHxCV(x=0..3)$ registers, at the same time the $CHxIF/$ $MCHxIF(x=0..3)$ bits are set and the channel interrupt is generated if it is enabled when $CHxIE/$ $MCHxIE=1(x=0..3)$.

The diagram illustrates the internal hardware logic of the CH0F0 module. It shows the flow of data and control signals between various components:

- Synchronizer:** Receives CIO and TIMERx_CK signals. It contains two D flip-flops and two comparators.
- Filter:** Receives the output of the Synchronizer and produces a filtered signal.
- Edge Detector:** Receives the filtered signal and produces two edge detection signals (rising and falling edges).
- Edge selector & inverter:** Based on CH0P and MCH0P signals, it selects between the edge detection signals and produces CIOFE0 (Rising/Falling) and CIOF_ED.
- Clock Prescaler:** Receives TIMERx_CK and produces a prescaled clock signal.
- Counter:** Receives the prescaled clock signal and produces a count value (CH0VAL).
- Capture Register (CH0VAL):** Receives the count value from the Counter.
- Prescaler:** Receives the count value and produces a prescaled signal (CH0CAPPSC).
- CH0CAPPSC:** A control register that receives the prescaled signal.
- CH0MS:** A control register that receives the prescaled signal.
- CH0IF and CH0IE:** Status registers that receive the edge detection signals.
- CH0CC_I:** A control register that receives the edge detection signals.
- Capture INT From Other Channels:** Receives the output of the CH0CC_I register and produces the final output signal, TIMERx_CC_INT.

The diagram illustrates the internal architecture of the MCH0F0 timer module. It shows the following components and their interconnections:

- Inputs:** MCIO, TIMERx CK, MCH0F0 Rising/Falling, CIOFEM0, MCI1FEM0, ITS, MCH0APPSC, MCH0MS.
- Processing Blocks:**
 - Synchronizer:** Receives MCIO and TIMERx CK. It contains two D flip-flops (D Q) and two delay elements (triangles).
 - Filter:** Receives the output of the Synchronizer and produces a filtered signal.
 - Edge Detector:** Receives the filtered signal and produces two edge detection signals (rising and falling edges) using AND gates and inverters.
 - Edge selector & inverter:** Receives the edge detection signals and produces MCH0FEM0 Rising/Falling.
 - Based on MCH0FP:** Receives MCH0FEM0 Rising/Falling and produces CIOFEM0.
 - Counter:** Receives the output of the Edge selector & inverter and produces MCH0VAL.
 - Prescaler:** Receives MCH0VAL and produces MCH0APPSC.
 - MCH0MS:** Receives MCH0APPSC and produces MCH0MS.
- Outputs:** TIMERx CC_INT, CIOFEM0, MCH0FEM0 Rising/Falling, MCH0APPSC, MCH0MS.

First, the input signal of channel (C1x/ MC1x) is synchronized to TIMER_CK signal, and then sampled by a digital filter to generate a filtered input signal. Then through the edge detector, the rising or falling edge is detected by configuring CHxP/ MCHxP or MCHxFP bits. The input capture signal can also be selected from the input signal of other channel or the internal trigger signal by configuring CHxMS/ MCHxMS bits. The IC prescaler makes several input events generate one effective capture event. On the capture event, TIMERx_CHxCV/ TIMERx_MCHxCV will store the value of counter.

Step1: Filter configuration (CHxCAPFLT bit in TIMEx CHCTL0 register and MCHxCAPFLT

bit in `TIMERx_MCHCTL0` register).

Based on the input signal and quality of requested signal, configure compatible `CHxCAPFLT` or `MCHxCAPFLT` bit.

Step2: Edge selection (`CHxP` and `MCHxP` bits in `TIMERx_CHCTL2` register, `MCHxFP[1:0]` bits in `TIMERx_MCHCTL2` register).

Rising edge or falling edge, choose one by configuring `CHxP` and `MCHxP` bits or `MCHxFP[1:0]` bits.

Step3: Capture source selection (`CHxMS` bit in `TIMERx_CHCTL0` register, `MCHxMS` bit in `TIMERx_MCHCTL0` register).

As soon as selecting one input capture source by `CHxMS`, the channel must be set to input mode (`CHxMS! = 0x000` or `MCHxMS! = 0x000`) and `TIMERx_CHxCV/` `TIMERx_MCHxCV` cannot be written any more.

Step4: Interrupt enable (`CHxIE` and `CHxDEN` bits, `MCHxIE` and `MCHxDEN` bits in `TIMERx_DMAINTEN`).

Enable the related interrupt to get the interrupt and DMA request.

Step5: Capture enable (`CHxEN` and `MCHxEN` bits in `TIMERx_CHCTL2`).

Result: When the wanted input signal is captured, `TIMERx_CHxCV/` `TIMERx_MCHxCV` will be set by counter's value and `CHxIF/` `MCHxIF` bit is asserted. If the `CHxIF/` `MCHxIF` bit is 1, the `CHxOF/` `MCHxOF` bit will also be asserted. The interrupt and DMA request will be asserted or not based on the configuration of `CHxIE` and `CHxDEN` bits, `MCHxIE` and `MCHxDEN` bits in `TIMERx_DMAINTEN`.

Direct generation: A DMA request or interrupt is generated by setting `CHxG` directly.

The input capture mode can be also used for pulse width measurement from signals on the `TIMERx_CHx` and `TIMERx_MCHx` pins. For example, PWM signal connects to `CI0` input. Select `CI0` as channel 0 capture signals by setting `CH0MS` to 3'b001 in the channel control register (`TIMERx_CHCTL0`) and set capture on rising edge. Select `CI0` as channel 1 capture signal by setting `CH1MS` to 3'b010 in the channel control register (`TIMERx_CHCTL0`) and set capture on falling edge. The counter is set to restart mode and is restarted on channel 0 rising edge. Then the `TIMERx_CH0CV` can measure the PWM period and the `TIMERx_CH1CV` can measure the PWM duty cycle.

■ Output compare mode

[Figure 21-18. Output compare logic \(when `MCHxMSEL = 2'b00, x=0, 1, 2, 3`\)](#) and [Figure 21-19. Output compare logic \(when `MCHxMSEL = 2'b11, x=0,1,2,3`\)](#) show the logic circuit of output compare mode.

Figure 21-18. Output compare logic (when MCHxMSEL = 2'b00, x=0, 1, 2, 3)

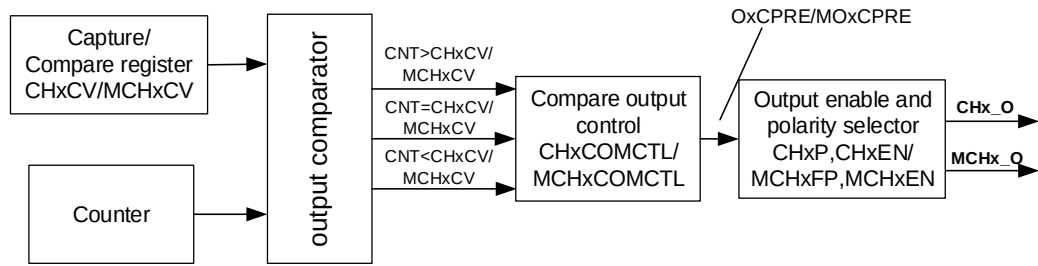
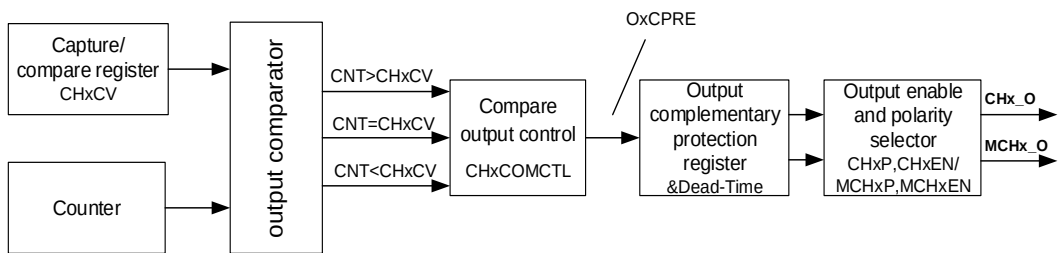


Figure 21-19. Output compare logic (when MCHxMSEL = 2'b11, x=0,1,2,3)



The relationship between the channel output signal CHx_O/MCHx_O and the OxCPRE/MOxCPRE signal (more details refer to [Channel output prepare signal](#)) is described as blew(the active level of OxCPRE is high and the active level of MOxCPRE is high).

- When MCHxMSEL=2'b00 (in TIMERx_CTL2 register), the MCHx_O output is independent from the CHx_O output. The output level of CHx_O depends on OxCPRE signal, CHxP bit and CHxEN bit (please refer to the TIMERx_CHCTL2 register for more details). The output level of MCHx_O depends on MOxCPRE signal, MCHxP[1:0] bits and MCHxEN bit (please refer to the TIMERx_MCHCTL2 and TIMERx_CHCTL2 registers for more details). Please refer to [Figure 21-18. Output compare logic \(when MCHxMSEL = 2'b00, x=0, 1, 2, 3\)](#).
- When MCHxMSEL=2'b11, the MCHx_O output is the inverse of the CHx_O output. The output level of CHx_O/MCHx_O depends on OxCPRE signal, CHxP/ MCHxP bits and CHxEN/MCHxEN bits. Please refer to [Figure 21-19. Output compare logic \(when MCHxMSEL = 2'b11, x=0,1,2,3\)](#).

For examples (the MCHx_O output is independent from the CHx_O output):

- 1) Configure CHxP=0 (the active level of CHx_O is high, the same as OxCPRE), CHxEN=1 (the output of CHx_O is enabled):

If the output of OxCPRE is active(high) level, the output of CHx_O is active(high) level;

If the output of OxCPRE is inactive(low) level, the output of CHx_O is active(low) level.

- 2) Configure MCHxP=1 (the active level of MCHx_O is low, contrary to MOxCPRE), MCHxEN=1 (the output of MCHx_O is enabled):

If the output of MOxCPRE is active(high) level, the output of MCHx_O is active(low) level;

If the output of MOxCPRE is inactive(low) level, the output of MCHx_O is active(high) level.

When MCHxMSEL=2'b11 and CHx_O and MCHx_O are output at the same time, the specific outputs of CHx_O and MCHx_O are related to the relevant bits (ROS, IOS, POE and DTCFG bits) in the TIMERx_CCHP register. Please refer to [Outputs Complementary](#) for more details.

In output compare mode, the TIMERx can generate timed pulses with programmable position, polarity, duration and frequency. When the counter matches the value in the TIMERx_CHxCV/ TIMERx_MCHxCV register of an output compare channel, the channel (n) output can be set, cleared, or toggled based on CHxCOMCTL/ MCHxCOMCTL. When the counter reaches the value in the TIMERx_CHxCV/ TIMERx_MCHxCV register, the CHxIF/ MCHxIF bit will be set and the channel (n) interrupt is generated if CHxIE/ MCHxIE = 1. And the DMA request will be asserted, if CHxDEN/ MCHxDEN =1.

So, the process can be divided into several steps as below:

Step1: Clock Configuration. Such as clock source, clock prescaler and so on.

Step2: Compare mode configuration.

- Set the shadow enable mode by CHxCOMSEN / MCHxCOMSEN .
- Set the output mode (set/clear/toggle) by CHxCOMCTL / MCHxCOMCTL.
- Select the active polarity by CHxP / MCHxP / MCHxFP.
- Enable the output by CHxEN / MCHxEN.

Step3: Interrupt/DMA request enable configuration by CHxIE/ MCHxIE /CHxDEN/ MCHxDEN.

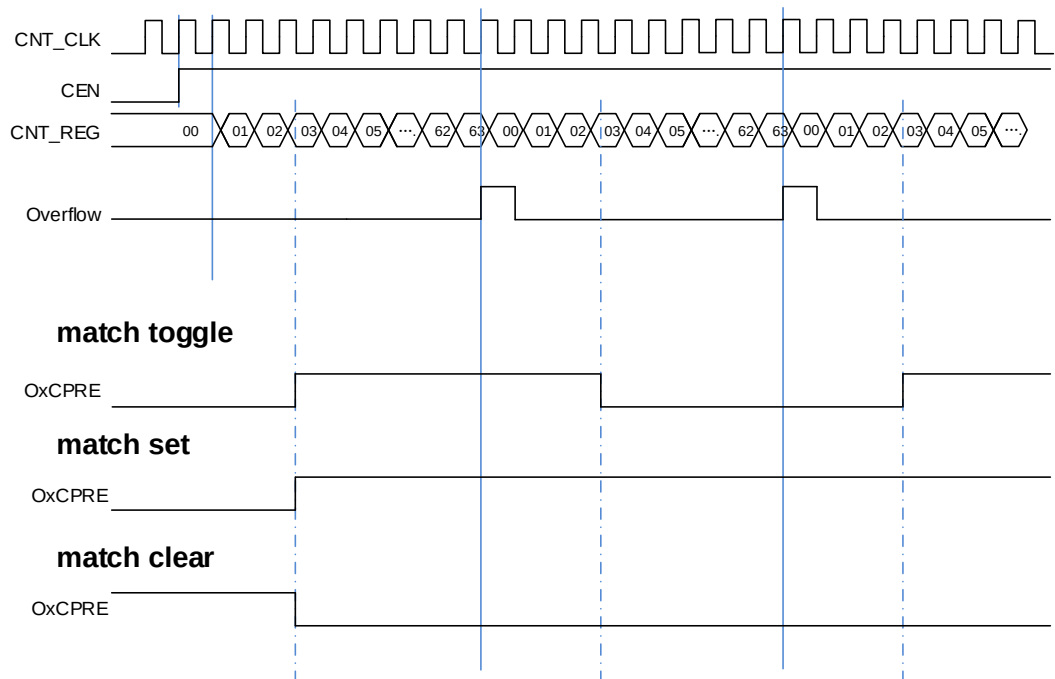
Step4: Compare output timing configuration by TIMERx_CAR and TIMERx_CHxCV/ TIMERx_MCHxCV.

The TIMERx_CHxCV/ TIMERx_MCHxCV can be changed ongoing to meet the expected waveform.

Step5: Start the counter by configuring CEN to 1.

[Figure 21-20. Output-compare in three modes](#) shows the three compare modes: toggle/set/clear. CARL=0x63, CHxVAL=0x3.

Figure 21-20. Output-compare in three modes



PWM mode

In the PWM output mode (by setting the CHxCOMCTL / MCHxCOMCTL bit to 4'b0110 (PWM mode 0) or to 4'b0111 (PWM mode 1)), the channel can generate PWM waveform according to the TIMERx_CAR registers and TIMERx_CHxCV / TIMERx_MCHxCV registers.

Based on the counter mode, PWM can also be divided into EAPWM (Edge-aligned PWM) and CAPWM (Center-aligned PWM).

The EAPWM's period is determined by TIMERx_CAR and the duty cycle is determined by TIMERx_CHxCV / TIMERx_MCHxCV. [Figure 21-21. Timing diagram of EAPWM](#) shows the EAPWM output and interrupts waveform.

The CAPWM's period is determined by 2 * TIMERx_CAR, and the duty cycle is determined by 2 * TIMERx_CHxCV / TIMERx_MCHxCV. [Figure 21-22. Timing diagram of CAPWM](#) shows the CAPWM output and interrupts waveform.

In up counting mode, if the value of TIMERx_CHxCV / TIMERx_MCHxCV is greater than the value of TIMERx_CAR, the output will be always active in PWM mode 0 (CHxCOMCTL / MCHxCOMCTL = 4'b0110). And if the value of TIMERx_CHxCV / TIMERx_MCHxCV is greater than the value of TIMERx_CAR, the output will be always inactive in PWM mode 1 (CHxCOMCTL / MCHxCOMCTL = 4'b0111).

Figure 21-21. Timing diagram of EAPWM

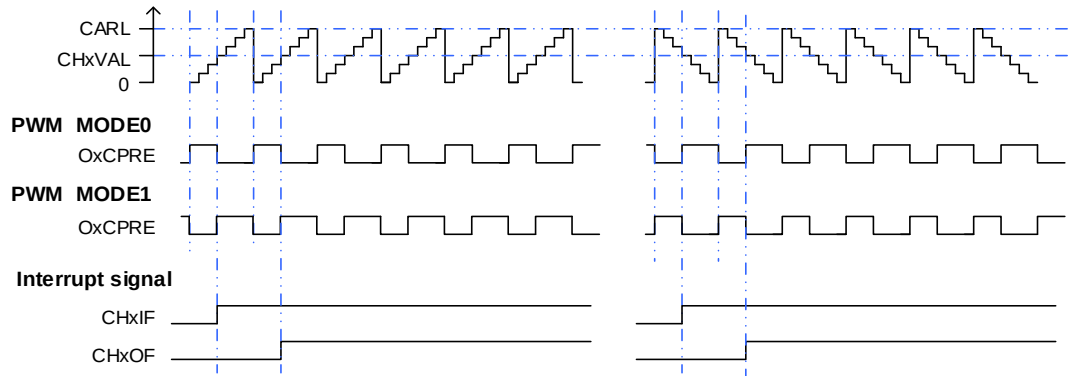
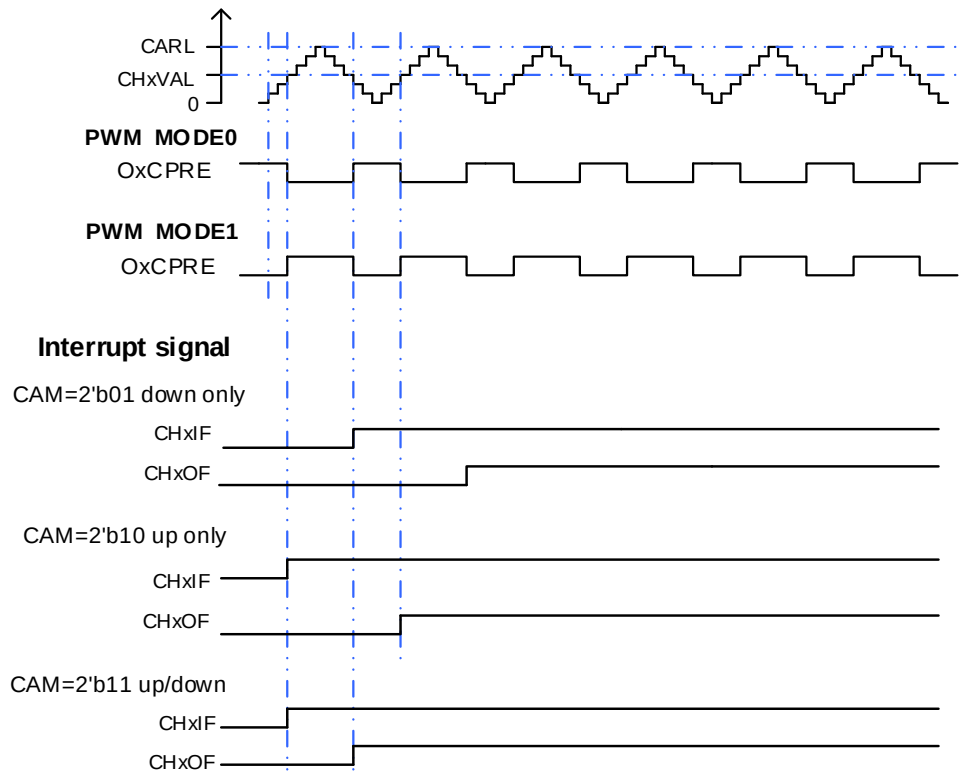


Figure 21-22. Timing diagram of CAPWM



Composite PWM mode

In the Composite PWM mode ($CHxMS[2:0] = 3'b000$ and $CHxCOMCTL = 4'b1000$ or $4'b1001$), the PWM signal output in channel x ($x=0..3$) is composited by $CHxVAL$ and $CHxCOMVAL\_ADD$ bits.

If $CHxCOMCTL = 4'b1000$ (composite PWM mode 0) and $DIR = 1'b0$ (up counting mode), or $CHxCOMCTL = 4'b1001$ (composite PWM mode 1) and $DIR = 1'b1$ (Down counting mode), the channel x output is forced low when the counter matches the value of $CHxVAL$. It is forced high when the counter matches the value of $CHxCOMVAL\_ADD$.

If $CHxCOMCTL = 4'b1001$ (composite PWM mode 1) and $DIR = 1'b0$ (up counting mode), or $CHxCOMCTL = 4'b1000$ (composite PWM mode 0) and $DIR = 1'b1$ (down counting mode)

the channel x output is forced high when the counter matches the value of CHxVAL. It is forced low when the counter matches the value of CHxCOMVAL_ADD.

When CHxVAL or CHxCOMVAL_ADD = 0 / CARL, the channel x output is specially processed, and by setting the CHxPERFOREN bit in the TIMEx_CHCTL2 register, the OxCPRE output is forced to either high or low level (determined according to the selected composite PWM mode).

In the central alignment mode, the composite PWM waveform is the symmetric waveform in the edge alignment mode, and the edge alignment mode is taken as an example for analysis. The PWM period is determined by (CARL + 0x0001) and the PWM pulse width is determined by the following table [Table 21-3.The Composite PWM pulse width](#).

Table 21-3.The Composite PWM pulse width

Condition	Mode	PWM pulse width
CHxVAL < CHxCOMVAL_ADD ≤ CARL	Composite PWM mode 0	(CARL + 0x0001) + (CHxVAL – CHxCOMVAL_ADD)
	Composite PWM mode 1	(CHxCOMVAL_ADD – CHxVAL)
CHxCOMVAL_ADD < CHxVAL ≤ CARL	Composite PWM mode 0	(CHxVAL - CHxCOMVAL_ADD)
	Composite PWM mode 1	(CARL + 0x0001) + (CHxCOMVAL_ADD – CHxVAL)
(CHxVAL = CHxCOMVAL_ADD ≤ CARL) or (CHxVAL > CARL > CHxCOMVAL_ADD)	Composite PWM mode 0 (up counting) or Composite PWM mode 1 (down counting)	100%
	Composite PWM mode 0 (down counting) or Composite PWM mode 1 (up counting)	0%
	Composite PWM mode 0(up counting) or composite PWM mode 1(down counting)	0%
	Composite PWM mode 0(down counting) or composite PWM mode 1(up counting)	100%
CHxCOMVAL_ADD > CARL > CHxVAL		
(CHxVAL > CARL) and (CHxCOMVAL_ADD > CARL)	-	The output of CHx_O is keeping

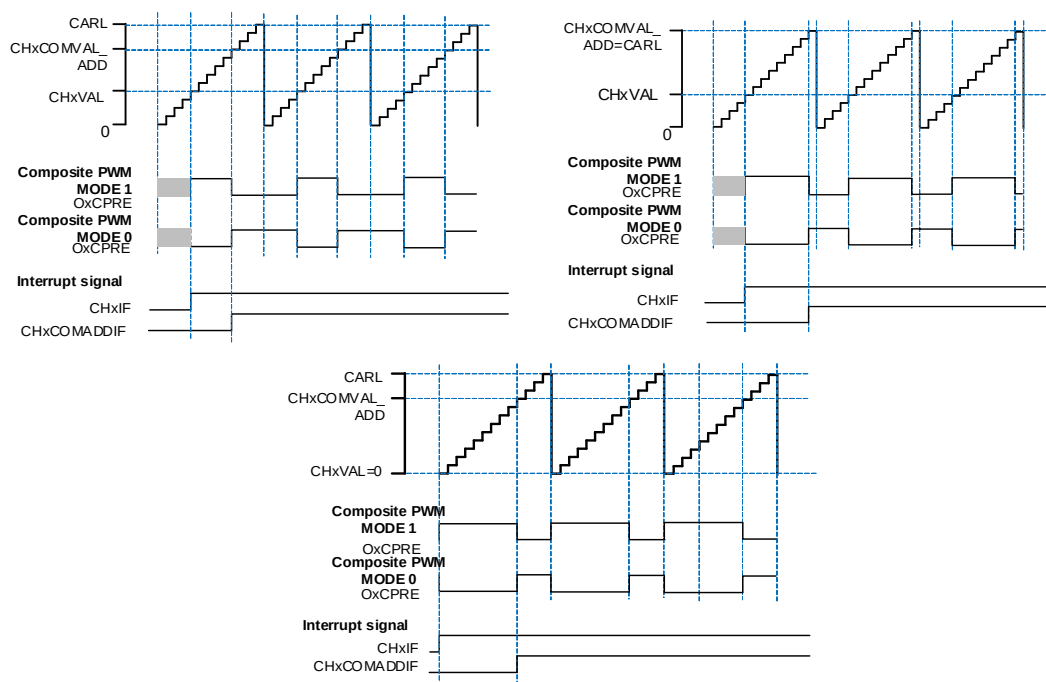
When the counter reaches the value of CHxVAL, the CHxIF bit is set and the channel x interrupt is generated if CHxIE = 1, and the DMA request will be asserted, if CHxDEN=1. When the counter reaches the value of CHxCOMVAL_ADD, the CHxCOMADDIF bit is set (This flag just used in Composite PWM mode and Asymmetric PWM mode, when CHxCOMCTL[3:2] = 2'b10) and the channel x additional compare interrupt is generated if

CHxCOMADDIE = 1 (Only interrupt is generated, no DMA request is generated).

According to the relationship among CHxVAL, CHxCOMVAL_ADD and CARL, it can be divided into four situations:

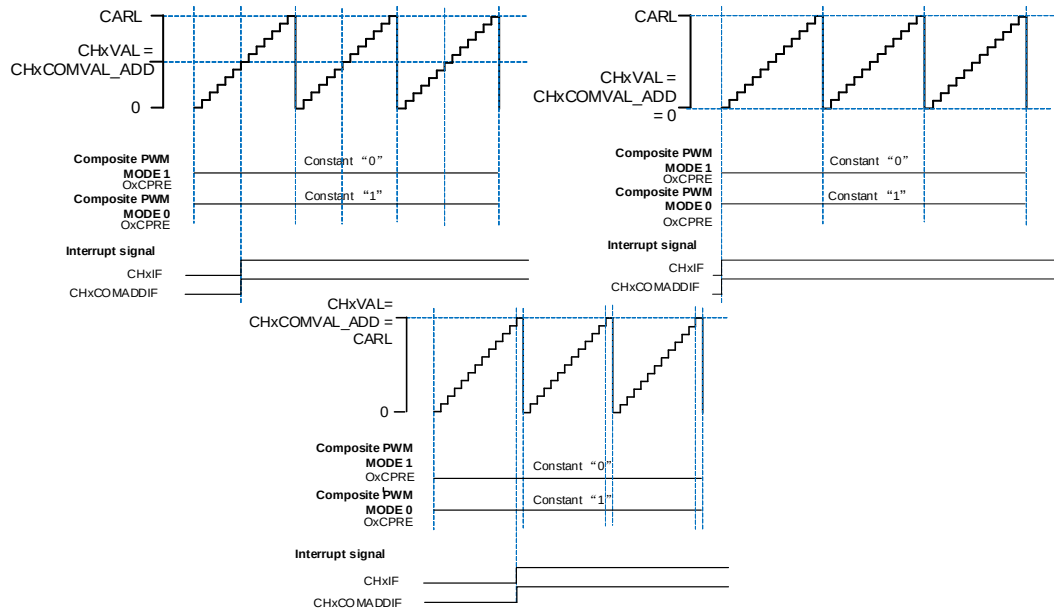
- 1) CHxVAL < CHxCOMVAL_ADD, and the values of CHxVAL and CHxCOMVAL_ADD between 0 and CARL.

Figure 21-23. Channel x output PWM with (CHxVAL < CHxCOMVAL_ADD)



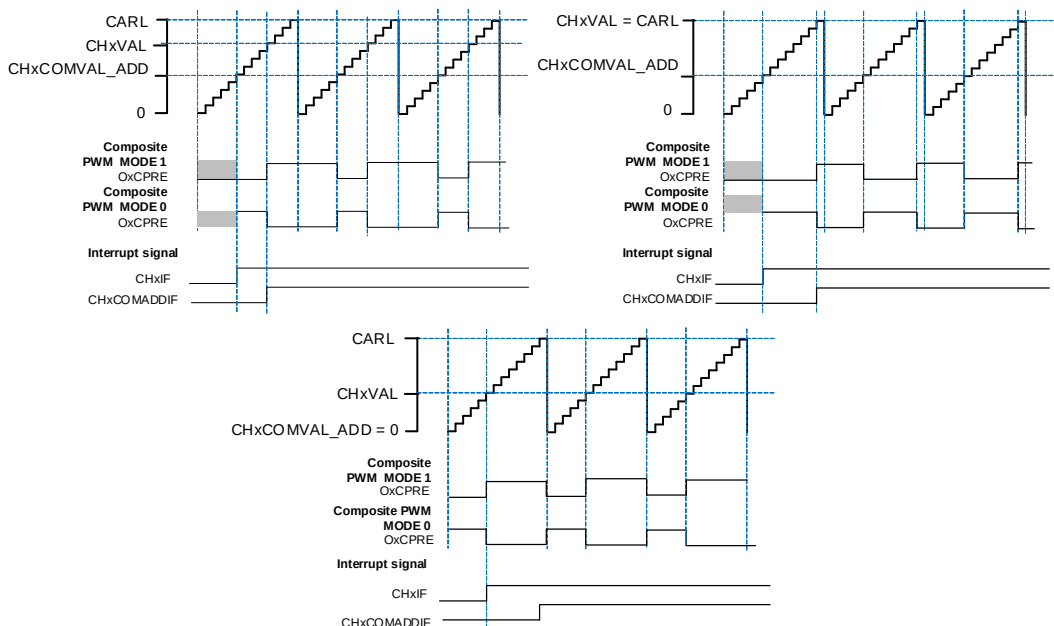
- 2) CHxVAL = CHxCOMVAL_ADD, and the value of CHxVAL and CHxCOMVAL_ADD between 0 and CARL.

Figure 21-24. Channel x output PWM with (CHxVAL = CHxCOMVAL_ADD)



- 3) CHxVAL > CHxCOMVAL_ADD, and the value of CHxVAL and CHxCOMVAL_ADD between 0 and CARL.

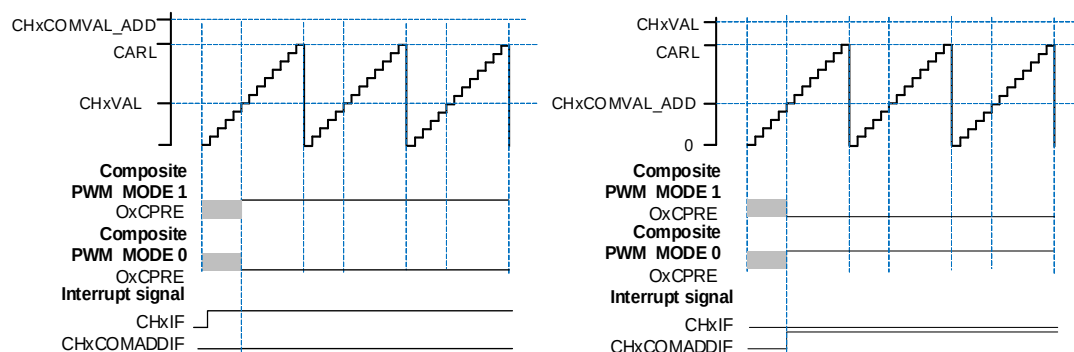
Figure 21-25. Channel x output PWM with (CHxVAL > CHxCOMVAL_ADD)



- 4) One of the value of CHxVAL and CHxCOMVAL_ADD exceeds CARL.

Figure 21-26. Channel x output PWM with CHxVAL or CHxCOMVAL_ADD exceeds

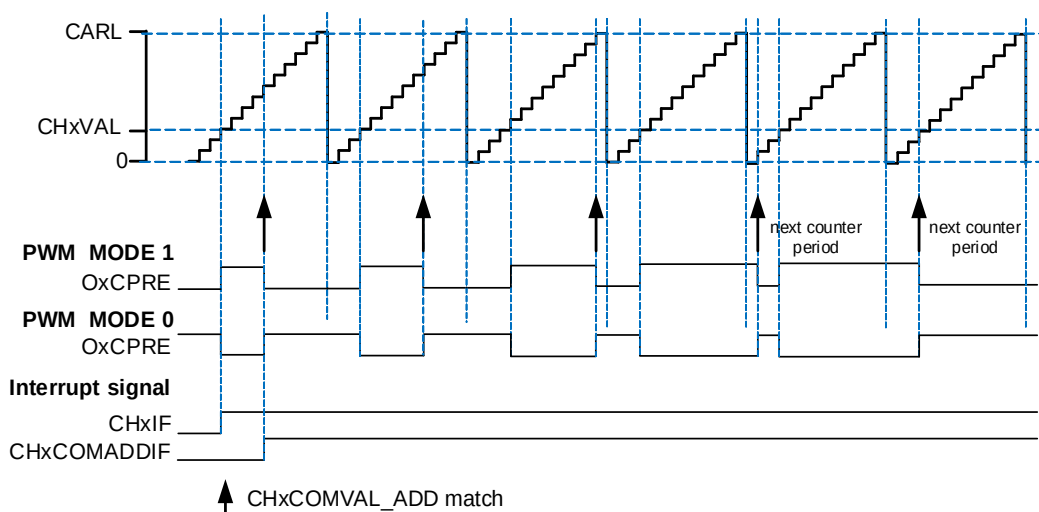
CARL



The composite PWM mode is intended to support the generation of PWM signals where the period is not modified while the signal is being generated, but the duty cycle will be varied. [Figure 21-27. Channel x output PWM duty cycle changing with CHxCOMVAL_ADD](#) shows the the PWM output and interrupts waveform.

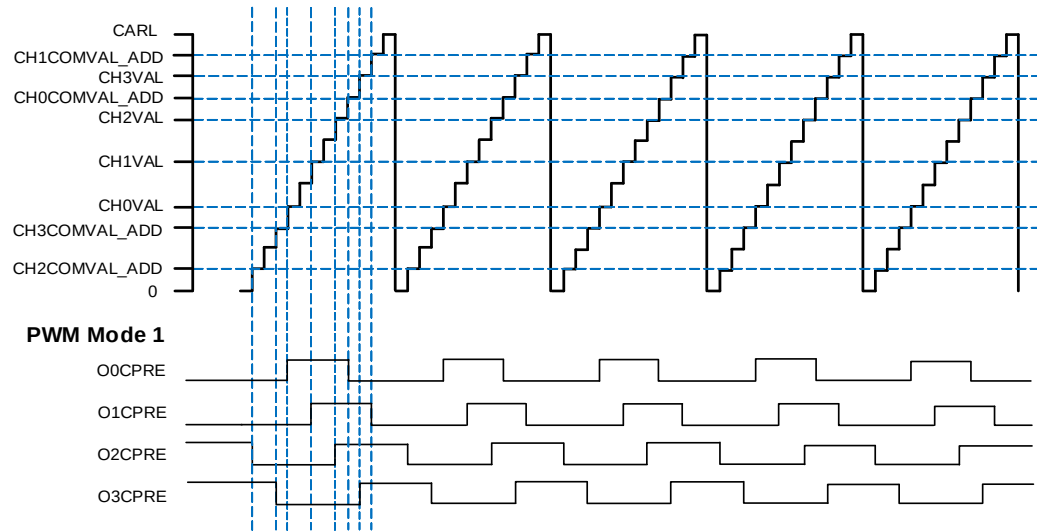
In some cases, the CHxCOMVAL_ADD match can happen on the next counter period (the value of CHxCOMVAL_ADD was written after the counter reaches the value of CHxVAL, and the value of CHxCOMVAL_ADD was less than or equal to the CHxVAL).

Figure 21-27. Channel x output PWM duty cycle changing with CHxCOMVAL_ADD



If more than one channels are configured in composite PWM mode, it is possible to fix an offset for the channel x match edge of each pair with respect to other channels. This behavior is useful in the generation of lighting PWM control signals where it is desirable that edges are not coincident with each other pair to help eliminate noise generation. The CHxVAL register value is the shift of the PWM pulse with respect to the beginning of counter period.

Figure 21-28. Four Channels outputs in Composite PWM mode



Asymmetric PWM mode

Asymmetric PWM mode allows two center-aligned PWM signals to be generated with a programmable phase shift. In the Asymmetric PWM mode ($CHxCOMCTL[3:0]=4'b1010/4'b1011$ and $CHxMS[2:0] = 3'b000$), the PWM signal output in channel x ($x=0..3$) is composited by $CHxVAL$ and $CHxCOMVAL\_ADD$ bits. The Asymmetric PWM mode is only available in the center-aligned counting mode.

If $CHxCOMCTL = 4'b1010$ (Asymmetric PWM mode 0) and $DIR = 1'b0$ (up counting mode), the channel x output is forced low when the counter matches the value of $CHxVAL$.

If $CHxCOMCTL = 4'b1010$ (Asymmetric PWM mode 0) and $DIR = 1'b1$ (down counting mode), the channel x output is forced high when the counter matches the value of $CHxCOMVAL\_ADD$.

If $CHxCOMCTL = 4'b1011$ (Asymmetric PWM mode 1) and $DIR = 1'b0$ (up counting mode), the channel x output is forced high when the counter matches the value of $CHxVAL$.

If $CHxCOMCTL = 4'b1011$ (Asymmetric PWM mode 1) and $DIR = 1'b1$ (down counting mode), the channel x output is forced low when the counter matches the value of $CHxCOMVAL\_ADD$.

Compared to Composite PWM mode, Asymmetric PWM mode is only valid with center-aligned mode, and the $CHxVAL$ is only valid when counting up and $CHxCOMVAL\_ADD$ is only valid when counting down. When $CHxVAL$ or $CHxCOMVAL\_ADD = 0$ / $CARL$, the channel x output is specially processed, and by setting the $CHxPERFOREN$ bit in the $TIMERx\_CHCTL2$ register, the $OxCPRE$ output is forced to either high or low level (determined according to the selected asymmetric PWM mode).

When $CHxVAL = 0$, the channel x output level is low in Asymmetric PWM mode 0. When $CHxVAL = 0$, the channel x output level is high in Asymmetric PWM mode 1.

When $CHxCOMVAL\_ADD = 0$, the channel x output level is high in Asymmetric PWM mode 0. When $CHxCOMVAL\_ADD = 0$, the channel x output level is low in Asymmetric PWM mode 1.

1.

When $CHxVAL = CARL$, the channel x output level is low in Asymmetric PWM mode 0. When $CHxVAL = CARL$, the channel x output level is high in Asymmetric PWM mode 1.

When $CHxCOMVAL\_ADD = CARL$, the channel x output level is high in Asymmetric PWM mode 0. When $CHxCOMVAL\_ADD = CARL$, the channel x output level is low in Asymmetric PWM mode 1.

The PWM period is determined by $2CARL$ and the PWM pulse width is determined by the following table [Table 21-4.The Asymmetric PWM pulse width](#).

Table 21-4.The Asymmetric PWM pulse width

Condition	Mode	PWM pulse width
$CHxVAL < CHxCOMVAL\_ADD \leq CARL$ or $CHxCOMVAL\_ADD < CHxVAL \leq CARL$ or $CHxVAL = CHxCOMVAL\_ADD$	Asymmetric PWM mode 0	$CHxCOMVAL\_ADD + CHxVAL$
	Asymmetric PWM mode 1	$2CARL - CHxCOMVAL\_ADD - CHxVAL$
$CHxVAL > CARL > CHxVAL\_ADD$	Asymmetric PWM mode 0	100%
	Asymmetric PWM mode 1	0%
$CHxCOMVAL\_ADD > CARL > CHxVAL$	Asymmetric PWM mode 0	0%
	Asymmetric PWM mode 1	100%
$(CHxVAL > CARL)$ and $(CHxCOMVAL\_ADD > CARL)$	-	The output of CHx_O is keeping

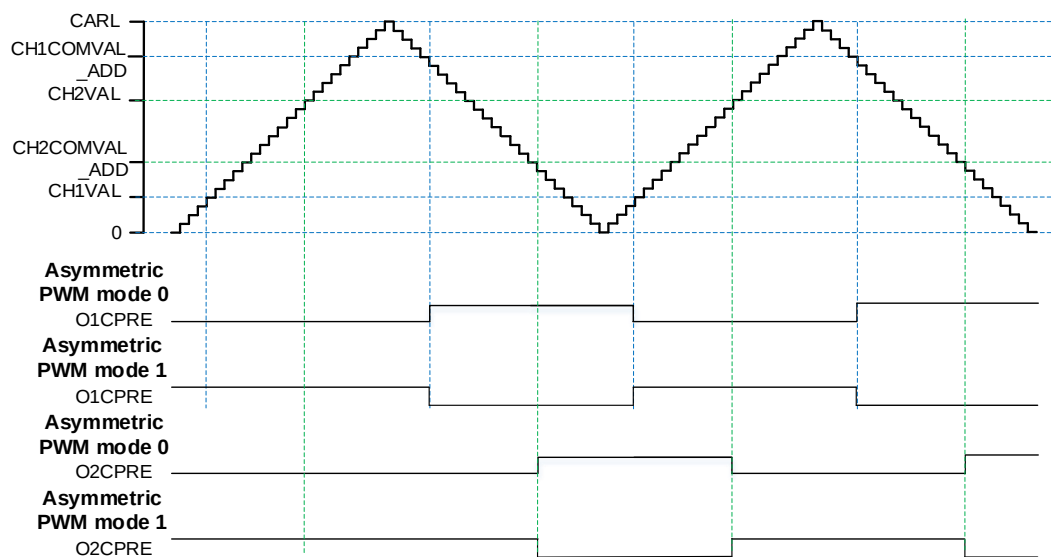
When the counter reaches the value of $CHxVAL$, the $CHxIF$ bit is set and the channel x interrupt is generated if $CHxIE = 1$, and the DMA request will be asserted, if $CHxDEN=1$. When the counter reaches the value of $CHxCOMVAL\_ADD$, the $CHxCOMADDIF$ bit is set (this flag just used in Composite PWM mode and Asymmetric PWM mode, when $CHxCOMCTL[3:2]=2'b10$) and the channel x additional compare interrupt is generated if $CHxCOMADDIE = 1$ (Only interrupt is generated, no DMA request is generated).

According to the relationship among $CHxVAL$, $CHxCOMVAL\_ADD$ and $CARL$, it can be divided into three situations:

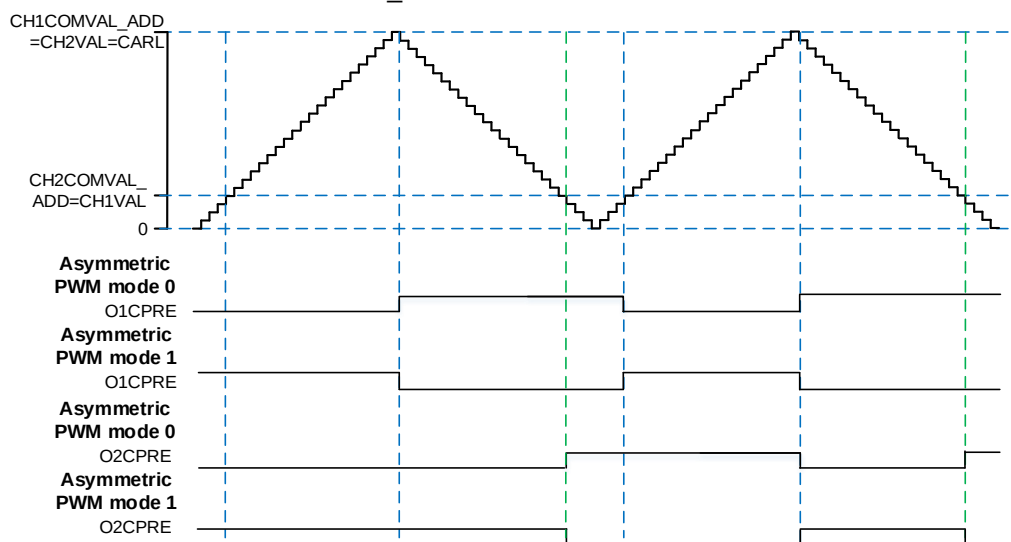
- 1) $CHxVAL < CHxCOMVAL\_ADD$, and the values of $CHxVAL$ and $CHxCOMVAL\_ADD$ between 0 and $CARL$.

Figure 21-29. Channel x output PWM with ($CHxVAL < CHxCOMVAL\_ADD$ / $CHxCOMVAL\_ADD < CHxVAL$)

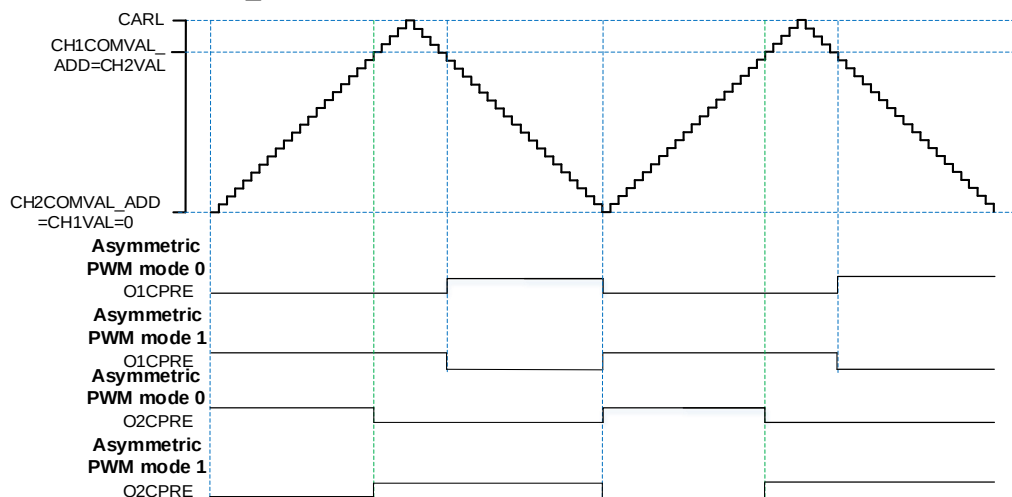
- a. $0 < CH1VAL < CH1COMVAL\_ADD < CARL$ / $0 < CH2COMVAL\_ADD < CH2VAL < CARL$



- b. $0 < CH1VAL < CH1COMVAL\_ADD = CARL / 0 < CH2COMVAL\_ADD < CH2VAL = CARL$,
and $CH1VAL = CH2COMVAL\_ADD$



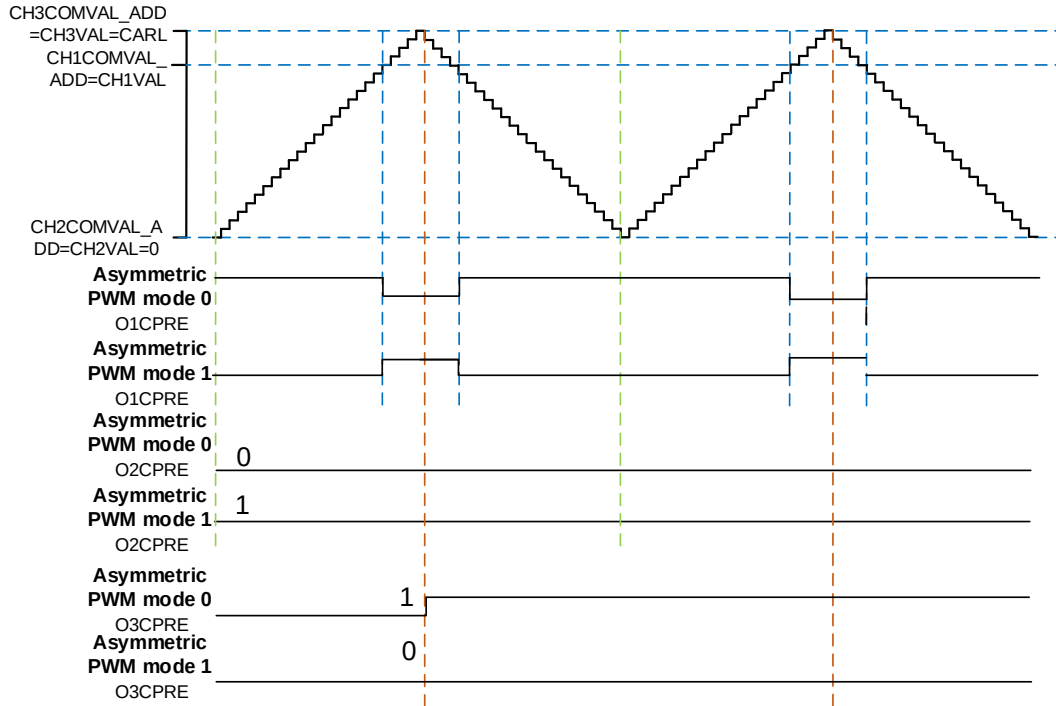
- c. $0 = CH1VAL < CH1COMVAL\_ADD < CARL / 0 = CH2COMVAL\_ADD < CH2VAL < CARL$,
and $CH1COMVAL\_ADD = CH2VAL$



- 2) $CHxVAL = CHxCOMVAL\_ADD$, and the value of $CHxVAL$ and $CHxCOMVAL\_ADD$ between 0 and $CARL$.

Figure 21-30. Channel x output PWM with ($CHxVAL = CHxCOMVAL\_ADD$)

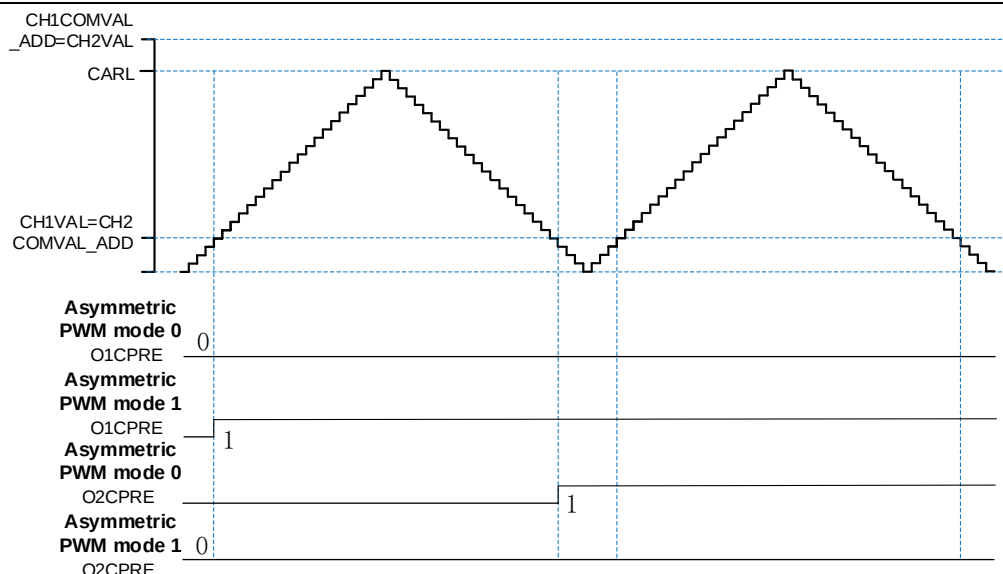
$0 < CH1VAL = CH1COMVAL\_ADD < CARL$ / $CH2VAL = CH2COMVAL\_ADD = 0$ / $CH3VAL = CH3COMVAL\_ADD = CARL$



- 3) One of the value of $CHxVAL$ and $CHxCOMVAL\_ADD$ exceeds $CARL$.

Figure 21-31. Channel x output PWM with $CHxVAL$ or $CHxCOMVAL\_ADD$ exceeds $CARL$

$0 < CH1VAL < CARL < CH1COMVAL\_ADD$ / $0 < CH2COMVAL\_ADD < CARL < CH2VAL$



Output match pulse select

Basing on that $CHx\_O(x=0..3)$ outputs are configured by $CHxCOMCTL[3:0](x=0..3)$ bits when the match events occur, the output signal is configured by $CHxOMPSEL[1:0](x=0..3)$ bit to be normal or a pulse.

When the match events occur, the $CHxOMPSEL[1:0](x=0..3)$ bits are used to select the output of $OxCPRE$ which drives CHx_O :

- $CHxOMPSEL = 2'b00$, the $OxCPRE$ signal is output normally with the configuration of $CHxCOMCTL[3:0]$ bits;
- $CHxOMPSEL = 2'b01$, only the counter is counting up, the $OxCPRE$ signal is output a pulse when the match events occur, and the pulse width is one CK_TIMER clock cycle.
- $CHxOMPSEL = 2'b10$, only the counter is counting down, the $OxCPRE$ signal is output a pulse when the match events occur, and the pulse width is one CK_TIMER clock cycle.
- $CHxOMPSEL = 2'b11$, both the counter is counting up and counting down, the $OxCPRE$ signal is output a pulse when the match events occur, and the pulse width is one CK_TIMER clock cycle.

Figure 21-32. CHx_O output with a pulse in edge-aligned mode (CHxOMPSEL#2'b00)

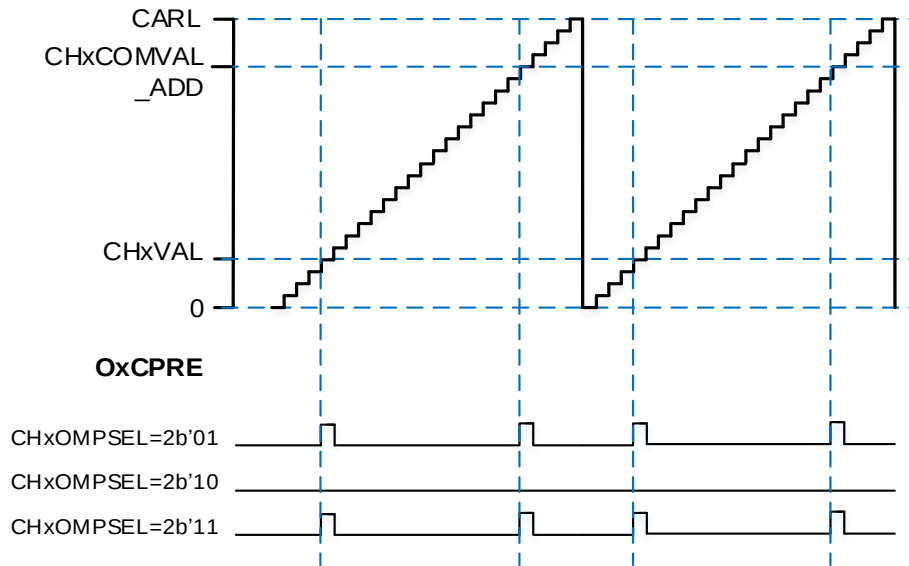
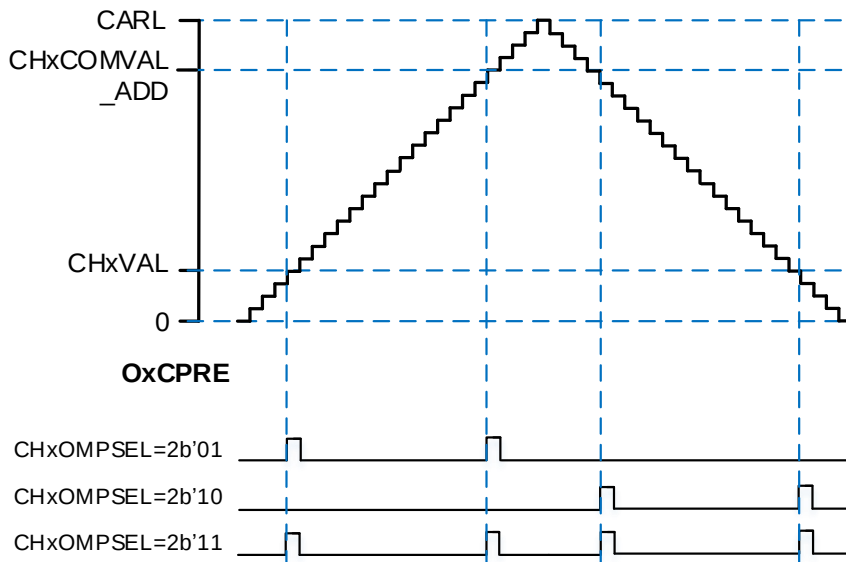


Figure 21-33. CHx_O output with a pulse in center-aligned mode (CHxOMPSEL#2'b00)



Channel output prepare signal

As is shown in [Figure 21-18. Output compare logic \(when MCHxMSEL = 2'b00, x=0, 1, 2, 3\)](#) and [Figure 21-19. Output compare logic \(when MCHxMSEL = 2'b11, x=0,1,2,3\)](#), when TIMEx is configured in compare match output mode, a middle signal named OxCPRE or MOxCPRE (channel x output or multi mode channel x output prepare signal) will be generated before the channel outputs signal.

The OxCPRE and MOxCPRE signal have several types of output function. The OxCPRE signal type is defined by configuring the CHxCOMCTL bit and the MOxCPRE signal type is defined by configuring the MCHxCOMCTL bit.

Take OxCPRE as an example for description below, these include keeping the original level

by configuring the CHxCOMCTL field to 0x00, setting to high by configuring the CHxCOMCTL field to 0x01, setting to low by configuring the CHxCOMCTL field to 0x02 or toggling signal by configuring the CHxCOMCTL field to 0x03 when the counter value matches the content of the TIMERx_CHxCV register.

The PWM mode 0/ PWM mode 1 output is another output type of OxCPRE which is setup by configuring the CHxCOMCTL field to 0x06/0x07. In the two modes, the OxCPRE signal level is changed according to the counting direction and the relationship between the counter value and the TIMERx_CHxCV content. Refer to the definition of relative bit for more details.

The Composite PWM mode 0/ Composite PWM mode 1 output is one output type of OxCPRE which is setup by configuring the CHxCOMCTL field to 0x08/0x09. In the two modes, the OxCPRE signal level is changed according to the counting direction and the relationship between the counter value and the TIMERx_CHxCV / TIMERx_CHxCOMVAL_ADD content. Refer to the definition of relative bit for more details.

The Asymmetric PWM mode 0/ Asymmetric PWM mode 1 output is one output type of OxCPRE which is setup by configuring the CHxCOMCTL field to 0x0a/0x0b. In the two modes, the OxCPRE signal level is changed according to the counting direction and the relationship between the counter value and the TIMERx_CHxCV / TIMERx_CHxCOMVAL_ADD content. Refer to the definition of relative bit for more details.

Another special function of the OxCPRE signal is forced output which can be achieved by configuring the CHxCOMCTL field to 0x04 / 0x05. The output can be forced to an inactive / active level irrespective of the comparison condition between the values of the counter and the TIMERx_CHxCV.

Clear the channel output prepare signal

The OxCPRE and MOxCPRE signal can be cleared by the OCPRE_CLR_INT signal when the CHxCOMCEN or MCHxCOMCEN bit (in TIMERx_CHCTLY/ TIMERx_MCHCTLY register) is set. This function can be used in the compare output modes which configured in CHxCOMCTL[3:0] (without the value of 4'b0100 and 4'b0101) or MCHxCOMCTL[2:0] bit-field (without the value of 3'b100 and 3b101).

The source of the OCPRE_CLR_INT signal can be selected by the OCRC bit in the TIMERx_SMCFG register.

When OCRC is set, the OCPRE_CLR_INT is connected to the OCPRE_CLR input. The OxCPRE / MOxCPRE signal is cleared by the high level of the OCPRE_CLR_INT signal, and restored until the next update event occurs. The OCPRE_CLR inputs can be selected in OCRINSEL[2:0] bit-field in the TIMERx_AFCTL register.

When OCRC is reset, the OCPRE_CLR_INT is connected to the ETIF. The OCPRE_CLR_INT input polarity is configured by ETP bit in TIMERx_SMCFG register. And the SMC1 bit must be set to 0, the ETPSC[1:0] bit-field must be set to 2'b00.

Outputs Complementary

The outputs of CHx_O and MCHx_O have two situations:

- MCHxMSEL=2'b00: The MCHx_O output is independent from the CHx_O output;
- MCHxMSEL=2'b11: The outputs of MCHx_O and CHx_O are complementary and the MCHxOMCTL bits are not used in the generation of the MCHx_O output.

Function of complementary is for a pair of channels, CHx_O and MCHx_O, the two output signals cannot be active at the same time. The TIMERx has 4 pairs of channels, all the four pairs have this function. The complementary signals CHx_O and MCHx_O are controlled by a group of parameters: the CHxEN and MCHxEN bits in the TIMERx_CHCTL2 register, the POEN, ROS and IOS bits in the TIMERx_CCHP register, ISOx and ISOxN bits in the TIMERx_CTL1 register. The output polarity is determined by CHxP and MCHxP bits in the TIMERx_CHCTL2 register.

When the the outputs of CHx_O and MCHx_O are complementary, there are three situations: output enable、output off-state and output disabled. The details are shown in [Table 21-5. Complementary outputs controlled by parameters \(MCHxMSEL =2'b11\).](#)

Table 21-5. Complementary outputs controlled by parameters (MCHxMSEL =2'b11)

Complementary Parameters					Output Status		
POEN	ROS	IOS	CHxEN	MCHxEN	CHx_O	MCHx_O	
0	0/1	0	0	0	CHx_O / CHx_ON = LOW CHx_O / CHx_ON output disable ⁽¹⁾ .		
				1	CHx_O/ CHx_ON output “off-state” ⁽²⁾ :		
			1	0	the CHx_O/ CHx_ON output inactive level firstly: CHx_O = CHxP, CHx_ON = CHxNP; If the clock for deadtime generator is present, after a deadtime: CHx_O = ISOx, CHx_ON = ISOxN. ⁽³⁾		
				1			
		1	x	x	CHx_O/ CHx_ON output “off-state”: the CHx_O/ CHx_ON output inactive level firstly: CHx_O = CHxP, CHx_ON = CHxNP; If the clock for deadtime generator is present, after a deadtime: CHx_O = ISOx, CHx_ON = ISOxN.		
1	0	0/1	0	0	CHx_O/MCHx_O = LOW CHx_O/MCHx_O output disable.		
				1	CHx_O = LOW CHx_O output disable.	MCHx_O=OxCPRE ⊕ ⁽⁴⁾ MCHxP MCHx_O output enable.	
			1	0	CHx_O=OxCPRE ⊕ CHxP CHx_O output enable.	MCHx_O = LOW MCHx_O output disable.	
				1	CHx_O=OxCPRE ⊕ CHxP CHx_O output enable.	MCHx_O=(!OxCPRE) ⁽⁵⁾ ⊕ MCHxP. MCHx_O output enable.	
	1		0	0	CHx_O = CHxP CHx_O output “off-state”.	MCHx_O = MCHxP MCHx_O output “off-state”.	
				1	CHx_O = CHxP CHx_O output “off-state”	MCHx_O=OxCPRE ⊕ MCHxP MCHx_O output enable	
			1	0	CHx_O=OxCPRE ⊕ CHxP CHx_O output enable	MCHx_O = MCHxP MCHx_O output “off-state”.	
				1	CHx_O=OxCPRE ⊕ CHxP CHx_O output enable	MCHx_O=(!OxCPRE) ⊕ MCHxP MCHx_O output enable.	

Note:

(1) output disable: the CHx_O / CHx_ON are disconnected to corresponding pins, the pin is floating with GPIO pull up/down setting which will be Hi-Z if no pull.

(2) “off-state”: CHx_O / CHx_ON output with inactive state (e.g., CHx_O = 0 ⊕ CHxP = CHxP).

(3) See Break mode section for more details.

(4) ⊕: Xor calculate.

(5) (!OxCPRE): the complementary output of the OxCPRE signal.

Dead time insertion

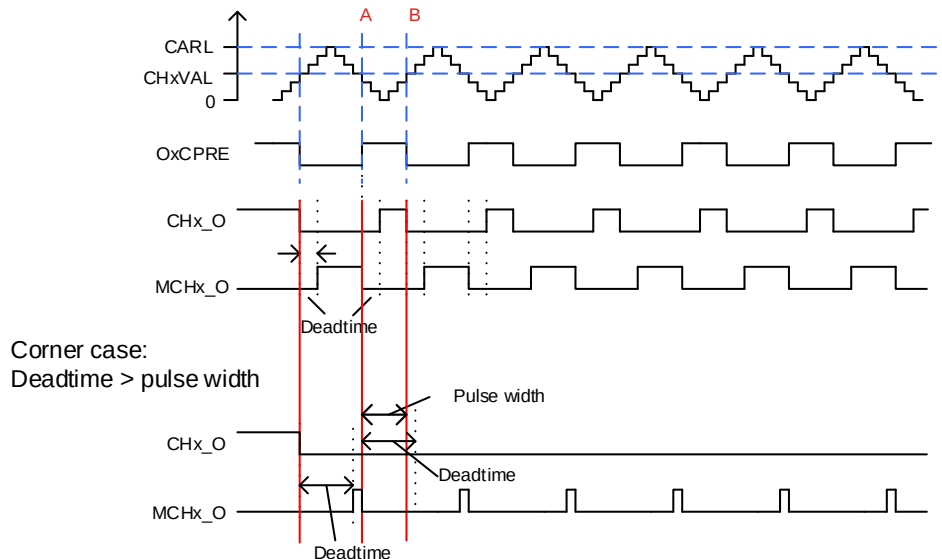
The dead time insertion is enabled when $MCHxMSEL=2'b11$ and both $CHxEN$ and $MCHxEN$ are configured to $1'b1$, it is also necessary to configure $POEN$ to 1. The field named $DTCFG$ defines the dead time delay that can be used for all channels. Refer to the [Complementary channel protection register \(TIMERx_CCHP\)](#) for details about the delay time.

The dead time delay insertion ensures that two complementary signals are not active at the same time.

When the channel x match event ($TIMERx\_CNT = CHxVAL$) occurs, $OxCPRE$ will be toggled in PWM mode 0. At point A in [Figure 21-34. Complementary output with dead time insertion](#), CHx_O signal remains at the low level until the end of the dead time delay, while $MCHx\_O$ signal will be cleared at once. Similarly, at point B when the channelx match event ($TIMERx\_CNT = CHxVAL$) occurs again, $OxCPRE$ is cleared, and CHx_O signal will be cleared at once, while $MCHx\_O$ signal remains at the low level until the end of the dead time delay.

Sometimes, we can see corner cases about the dead time insertion. For example: the dead time delay is greater than or equal to the duty cycle of the CHx_O signal, then the CHx_O signal is always inactive (As shown in [Figure 21-34. Complementary output with dead time insertion](#)).

Figure 21-34. Complementary output with dead time insertion



Break function

In break functions, CHx_O and $MCHx\_O$ are controlled by the $POEN$, $OAEN$, IOS and ROS bits in the $TIMERx\_CCHP$ register, $ISOx$ and $ISOxN$ bits in the $TIMERx\_CTL1$ register.

The $MCHx\_O$ output is the inverse of the CHx_O output when the $MCHxMSEL=2'b11$ (and the $MCHxCOMCTL$ bits are not used in the generation of the $MCHx\_O$ output). In this case,

CHx_O and MCHx_O signals cannot be set to active level at the same time.

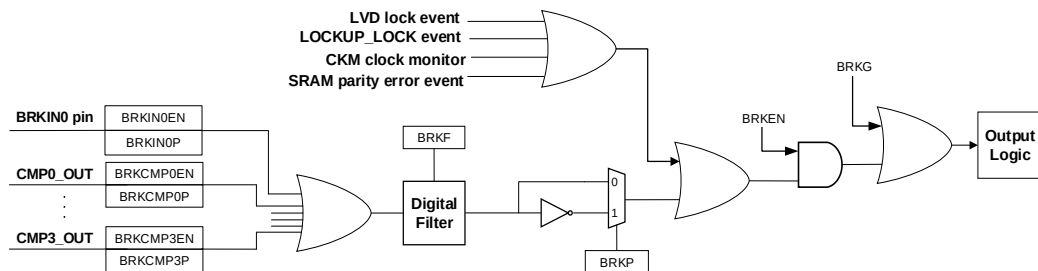
The advanced timers have break function. The break sources are system events and other events. When the break event occurs, the outputs is force at a predefined level (either active or inactive) after a deadtime duration. The break function is enabled by setting the BRKEN bit in the TIMEx_CCHP register. The break input polarity is configured by the BRKP bit in TIMEx_CCHP register.

The break event is the result of logic ORed of all sources. The break functions can handle two types of event sources:

- External sources: coming from external pin BRKIN0 input, and with digital filters and polarity selection;
- Internal sources: system sources(such as: HXTAL stuck event which is generated by Clock Monitor CKM in RCU, LVD lock event, LOCKUP_LOCK event and SRAM parity error event), and on-chip comparator events (the four outputs after digital filters and polarity selection in the four comparators as TIMEx input) .

Break events can also be generated by software using BRKG bit in the TIMEx_SWEVG register. Refer to [Figure 21-35. BREAK function logic diagram.](#)

Figure 21-35. BREAK function logic diagram

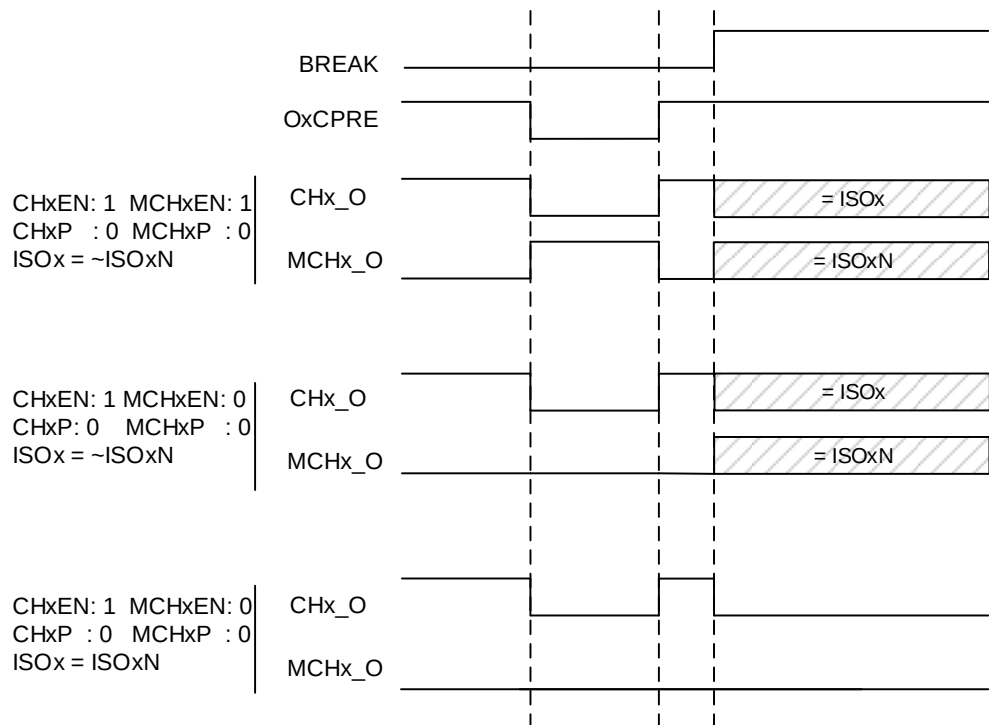


The BRKIN0 pin in the break function diagram is selected from the GPIO, and the BRKCMPPx is from the comparator, and is controlled by BRKCMPPxEN enable bit and the BRKCMPPxP polarity selection. When the system sources are active, the SYSBIF flag is set by hardware. The software waits for the system break sources inactive, and then clears the SYSBIF flag.

When the MCHxMSEL = 2'b11 and a break occurs, the POEN bit is cleared asynchronously. As soon as POEN is 0, the level of the CHx_O and MCHx_O outputs are determined by the ISOx and ISOxN bits in the TIMEx_CTL1 register. If IOS = 0, the timer releases the enable output, otherwise, the enable output remains high. When IOS=1, the output behavior of the channel is shown in [Figure 21-36. Output behavior of the channel in response to BREAK \(the break input high active and IOS=1\).](#) The complementary outputs are first in the reset state, and then the dead time generator is reactivated to drive the outputs with the level programmed in the ISOx and ISOxN bits after a dead time.

Figure 21-36. Output behavior of the channel in response to BREAK (the break input

high active and IOS=1)



When a break occurs, the BRKIF bit in the TIMEx_INTF register will be set. If BRKIE is 1, an interrupt will be generated.

Quadrature decoder

The quadrature decoder function uses two quadrature inputs CI0 and CI1 derived from the TIMEx_CH0 and TIMEx_CH1 pins respectively to interact with each other to generate the counter value.

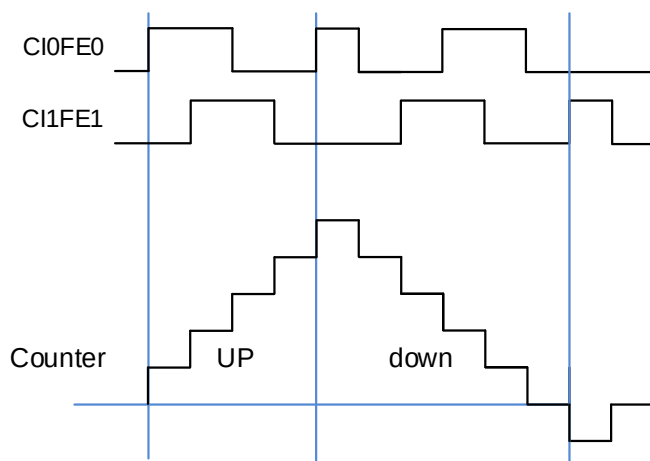
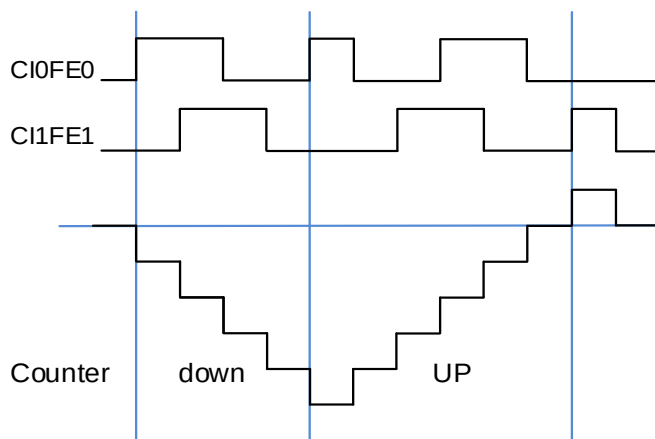
Setting TSCFGy[4:0](y=0..2) != 5'b00000 to select that the counting direction of timer is determined only by the CI0, only by the CI1, or by the CI0 and the CI1.

The DIR bit is modified by hardware automatically during the voltage level change of each direction selection source. The mechanism of changing the counter direction is shown in [Table 21-6. Counting direction in different quadrature decoder signals](#). The CI0FE0 and CI1FE1 are the signals of the CI0 and CI1 after the filtering and polarity selection. The quadrature decoder can be regarded as an external clock with a direction selection. This means that the counter counts continuously from 0 to the counter-reload value. Therefore, users must configure the TIMEx_CAR register before the counter starts to count.

Table 21-6. Counting direction in different quadrature decoder signals

Counting mode	Level	CI0FE0		CI1FE1	
		Rising	Falling	Rising	Falling
Quadrature decoder mode 0 TSCFG0[4:0] != 5'b00000	CI1FE1=1	Down	Up	-	-
	CI1FE1=0	Up	Down	-	-
Quadrature decoder mode 1 TSCFG1[4:0] != 5'b00000	CI0FE0=1	-	-	Up	Down
	CI0FE0=0	-	-	Down	Up
Quadrature decoder mode 2 TSCFG2[4:0] != 5'b00000	CI1FE1=1	Down	Up	X	X
	CI1FE1=0	Up	Down	X	X
	CI0FE0=1	X	X	Up	Down
	CI0FE0=0	X	X	Down	Up

Note: "-" means "no counting"; "X" means impossible. "0" means "low level", "1" means "high level".

Figure 21-37. Example of counter operation in decoder interface mode

Figure 21-38. Example of decoder interface mode with CI0FE0 polarity inverted


Non-quadrature decoder

The non-quadrature decoder function has four modes: non-quadrature decoder mode 0~3, which can be selected by setting TSCFGy[4:0] (y=9, 10, 11, 12) != 5'b00000. There are two

input sources in these four modes: CI0 and CI1, and the CI0FE0 and CI1FE1 are the signals of the CI0 and CI1 after the filtering and polarity selection.

When the non-quadrature decoder mode 0 and non-quadrature decoder mode 1 are enabled, the CI0FE0 is used as the count direction signal and the CI1FE1 signal is used as the count pulse.

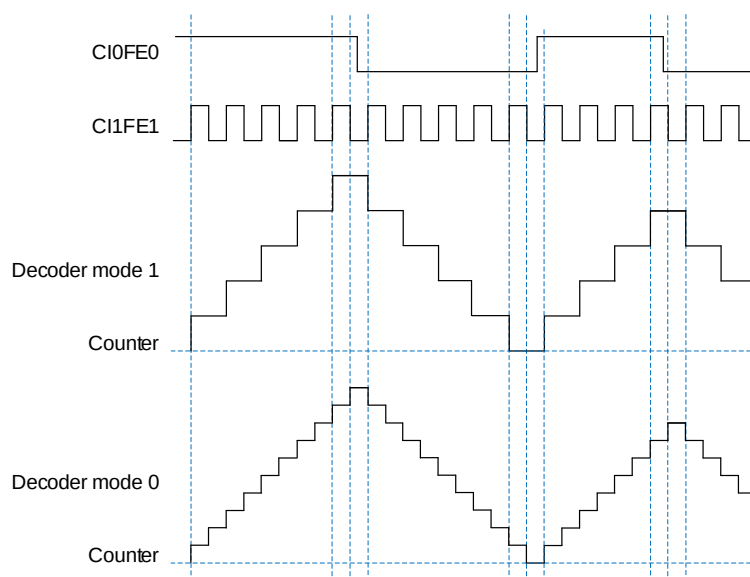
The CH0P is used to configure the count direction selection: When CH0P =0, the counter will count up when the CI0FE0 signal is high and will count down when the CI0FE0 signal is low; When CH0P =1, the counter will count down when the CI0FE0 signal is high and will count up when the CI0FE0 signal is low.

The CH1P is used to configure the count pulse edge selection: In non-quadrature decoder mode 0, the counter will count on both rising and falling edges of the CI1FE1 signal. In non-quadrature decoder mode 1, when CH1P=0, the counter will count on the rising edge of the CI1FE1 signal; When CH1P=1, the counter will count on the falling edge of the CI1FE1 signal. The more details for non-quadrature decoder mode 1 is shown in [Table 21-7. the counter operation in non-quadrature decoder mode 1](#) and [Figure 21-39. Example of counter operation in non-quadrature decoder mode 0 / 1 with CH1P=0](#).

Table 21-7. the counter operation in non-quadrature decoder mode 1

CH1P	level	counter operation
0	CI0FE0 is high	the counter will count up on the rising edge of the CI1FE1 input signal
	CI0FE0 is low	the counter will count down on the rising edge of the CI1FE1 input signal
1	CI0FE0 is high	the counter will count up on the falling edge of the CI1FE1 input signal
	CI0FE0 is low	the counter will count down on the falling edge of the CI1FE1 input signal

Figure 21-39. Example of counter operation in non-quadrature decoder mode 0 / 1 with CH1P=0



The non-quadrature decoder mode 2/3 uses two inputs CI0FE0 and CI1FE1 respectively to interact with each other to generate the counter value, and the counting direction DIR bit modified by hardware automatically.

When in non-quadrature decoder mode 2, the counter will count on both rising and falling edges of CI0FE0 and CI1FE1 signals. And the counting direction determined by the CH0P and CH1P bits.

When in non-quadrature decoder mode 3, the counter will count on rising or falling edge of CI0FE0 and CI1FE1 signals. When CHxP=0, the counter will counter on the high level or the falling edge or the ClxFEx signal; When CHxP=1, the counter will counter on the low level or the rising edge or the ClxFEx signal. The more details for non-quadrature decoder mode 2/3 is shown in [Figure 21-40. Example of counter operation in non-quadrature decoder mode 2 / 3 \(CH0P / CH1P=0\)](#) and [Table 21-8. the counter operation in non-quadrature decoder mode 2 / 3](#).

Figure 21-40. Example of counter operation in non-quadrature decoder mode 2 / 3 (CH0P / CH1P=0)

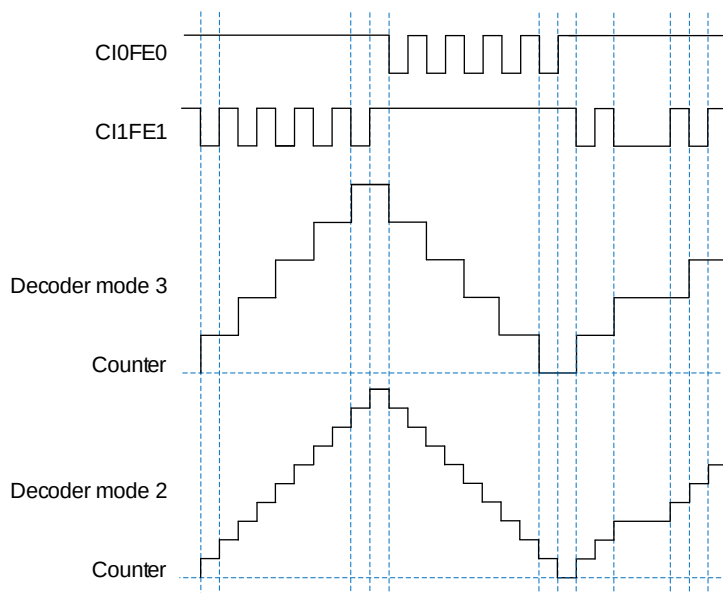


Table 21-8. the counter operation in non-quadrature decoder mode 2 / 3

Counting mode	Polarity	Level	CI0FE0		CI1FE1	
			Rising	Falling	Rising	Falling
non-quadrature decoder mode 2 TSCFG11[4:0] != 5'b00000	CHxP=0 (x=0 or 1)	CI1FE1=1	Down	Down	x	x
		CI1FE1=0	-	-	x	x
		CI0FE0=1	x	x	Up	Up
		CI0FE0=0	x	x	-	-
	CHxP=1 (x=0 or 1)	CI1FE1=1	-	-	x	x
		CI1FE1=0	Down	Down	x	x
		CI0FE0=1	x	x	-	-
		CI0FE0=0	x	x	Up	Up
non-quadrature decoder mode 3 TSCFG12[4:0] != 5'b00000	CHxP=0 (x=0 or 1)	CI1FE1=1	-	Down	x	x
		CI1FE1=0	-	-	x	x
		CI0FE0=1	x	x	-	Up
		CI0FE0=0	x	x	-	-

Counting mode	Polarity	Level	CI0FE0		CI1FE1	
			Rising	Falling	Rising	Falling
	CHxP=1 (x=0 or 1)	CI1FE1=1	-	-	x	x
		CI1FE1=0	Down	-	x	x
		CI0FE0=1	x	x	-	-
		CI0FE0=0	x	x	Up	-

When the counter direction changes in the non-quadrature decoder modes, the DIR bit in `TIMERx_CTL0` register within a change, at the same time the DIRTRANIF bit in `TIMERx_INTF` is set to 1. And the corresponding interrupt is generated if the DIRTRANIE bit in `TIMERx_DMAINTEN` register is set to 1.

Hall sensor function

Hall sensor is generally used to control BLDC motor, the advanced timer supports this function.

[Figure 21-41. Hall sensor is used for BLDC motor](#) shows how to connect the timer and the motor. And two timers are needed. `TIMER_in`(Advanced/General L0 TIMER) is used to accept three rotor position signals of motor from hall sensors.

Each of the 3 hall sensors provides a pulse which is applied to an input capture pin, then both the speed and position of rotor can be calculated by analyzing the hall sensor signals.

By the internal connection function (TRGO-ITIx), `TIMER_in` and `TIMER_out` can be connected. `TIMER_out` will generate PWM signals to control the speed of BLDC motor based on the ITIx. Then, the feedback circuit is finished, you can change the configuration to fit your request.

Because the advanced/general L0 TIMER has the input XOR function, they can be used as the `TIMER_in` timer. And the advanced timer has the functions of complementary output and dead time, so it can be used as the `TIMER_out` timer. Else, based on the timer's internal connection relationship, pair's timers can be selected. For example:

`TIMER_in` (TIMER0) -> `TIMER_out` (TIMER7 ITI0)

`TIMER_in` (TIMER1) -> `TIMER_out` (TIMER0 ITI0)

After appropriate interconnected timers are selected and wires are connected, the timers need to be configured. Some key settings are as follows:

- Enable XOR by setting `TI0S`, then, the change of each input signal will make the `CI0` toggle. `CH0VAL` will record the current value of counter.
- Choose ITIx to trigger commutation by configuring `CCUC` and `CCSE`.
- Configure PWM parameters based on the requests.

Figure 21-41. Hall sensor is used for BLDC motor

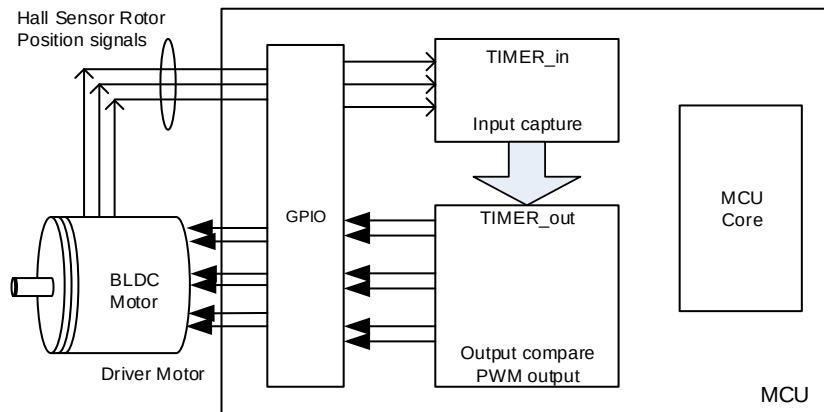
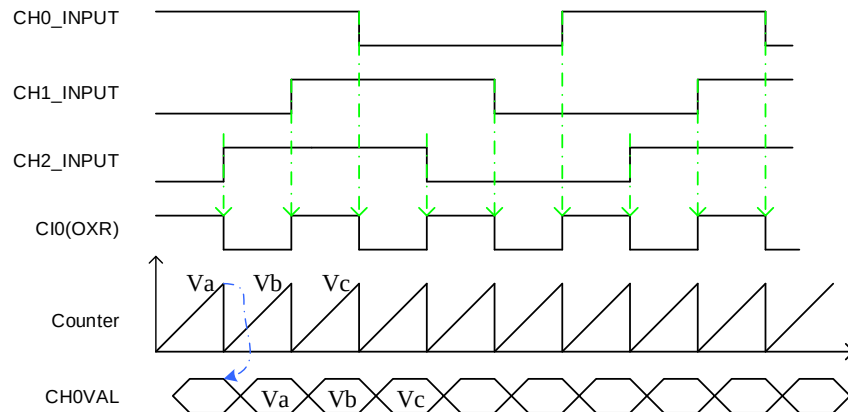
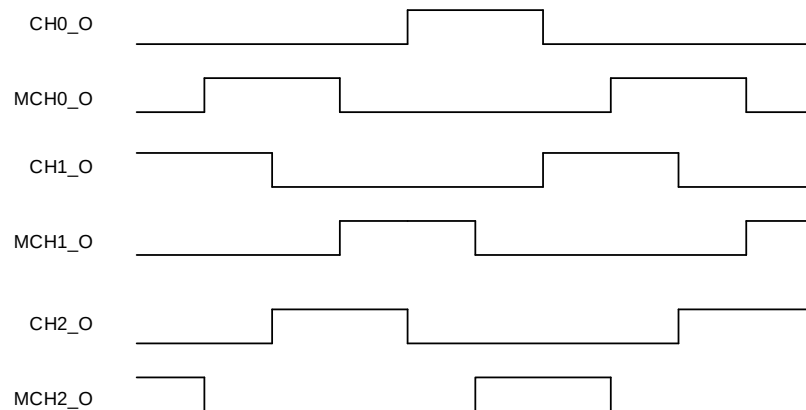


Figure 21-42. Hall sensor timing between two timers

Advanced/General L0 TIMER_in under input capture mode



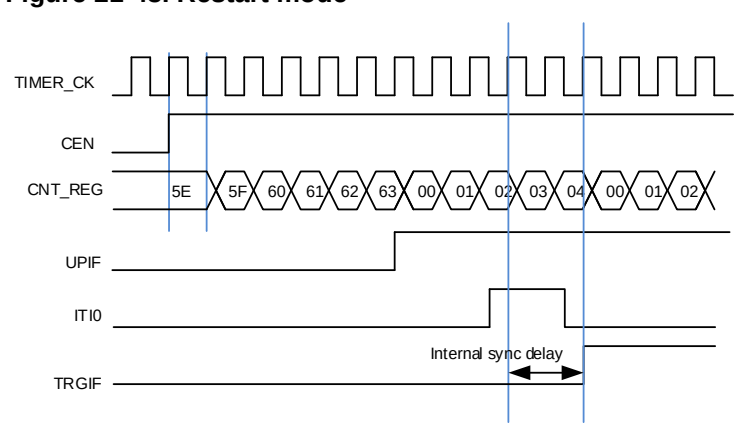
Advanced TIMER_out under output compare mode(PWM with Dead-time)



Master-slave management

The TIMEx can be synchronized with a trigger in several modes including restart mode, pause mode and event mode and so on, which is selected by the TSCFGy[4:0] (y=3..7) in SYSCFG_TIMERxCFG(x=0,7).

Table 21-9. Examples of slave mode

	Mode Selection	Source Selection	Polarity Selection	Filter and Prescaler
LIST	TSCFGy[4:0] y=3: restart mode y=4: pause mode y=5: event mode y=6: external clock mode 0 y=7: restart + event mode	TSCFGy[4:0] 00000: disable 00001: ITI0 00010: ITI1 00011: ITI2 00100: ITI3 00101: CI0F_ED 00110: CI0FE0 00111: CI1FE1 01000: ETIFP 01001: CI2FE2 01010: CI3FE3 01011: MCI0FEM0 01100: MCI1FEM1 01101: MCI2FEM2 01110: MCI3FEM3	If ClxFEx(x=0..3) or MCixFEMx(x=0..3) are selected as the trigger source, configure the CHxP, MCHxP and MCHxFP for the polarity selection and inversion. If ETIFP (the filtered output of external trigger input ETI) is selected as the trigger source, configure the ETP for polarity selection and inversion.	For the ITIx, no filter and prescaler can be used. For the Clx/ MCiFx, filter can be used by configuring CHxCAPFLT/ MCHxCAPFLT, no prescaler can be used. For the ETIFP, filter can be used by configuring ETFC and prescaler can be used by configuring ETPSC.
Exam1	Restart mode The counter will be cleared and restart when a rising edge of trigger input comes.	TSCFG3[4:0] 5'b00001, ITI0 is selected.	For ITI0, no polarity selector can be used.	For the ITI0, no filter and prescaler can be used.
Figure 21-43. Restart mode  The diagram illustrates the restart mode operation. It shows the relationship between the timer clock (TIMER_CK), the counter enable (CEN), the counter register (CNT_REG), the update interrupt flag (UPIF), the trigger input (ITI0), and the trigger input filter (TRGIF). The counter starts at 5E and increments to 04, then resets to 00. A rising edge on TRGIF triggers the reset. The internal sync delay is indicated between the trigger input and the counter reset.				
Exam2	Pause mode The counter will be paused when the trigger input is low, and it will start when the trigger input is high.	TSCFG4[4:0] =5'b00110, CI0FE0 is selected.	TI0S=0 (Non-xor) [MCH0P=0, CH0P=0] CI0FE0 does not invert. The capture event will occur on the rising edge only.	Filter is bypassed in this example.

	Mode Selection	Source Selection	Polarity Selection	Filter and Prescaler
	Figure 21-44. Pause mode			
Exam3	Event mode The counter will start to count when a rising edge of trigger input comes.	TSCFG5[4:0] = 5'b01000, ETIFP is selected.	ETP = 0, the polarity of ETI does not change.	ETPSC = 1, ETI is divided by 2. ETFC = 0, ETI does not filter.
	Figure 21-45. Event mode			

Single pulse mode

Single pulse mode is opposite to the repetitive mode, which can be enabled by setting SPM in `TIMERx_CTL0`. If SPM is set, the counter will be cleared and stopped automatically when the next update event occurs. In order to get a pulse waveform, the `TIMERx` is configured to PWM mode or compare mode by `CHxCOMCTL` or `MCHxCOMCTL` bits.

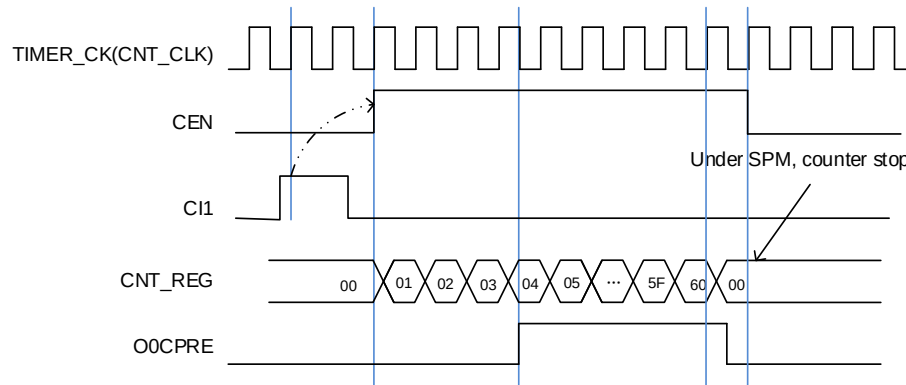
Once the timer is set to the single pulse mode, it is not necessary to configure the timer enable bit `CEN` in the `TIMERx_CTL0` register to 1 to enable the counter. Setting the `CEN` bit to 1 or a trigger signal edge can generate a pulse and then keep the `CEN` bit at a high state until the update event occurs or the `CEN` bit is written to 0 by software. If the `CEN` bit is cleared to 0 by software, the counter will be stopped and its value will be held. If the `CEN` bit is automatically cleared to 0 by a hardware update event, the counter will be reinitialized.

In the single pulse mode, the active edge of trigger which sets the `CEN` bit to 1 will enable the counter. However, there exists several clock delays to perform the comparison result between the counter value and the `TIMERx_CHxCV` value. After a trigger rising occurs in the single pulse mode, the `OxCPRE` signal will immediately be forced to the state which the `OxCPRE`/

MOxCPRE signals will change to, as the compare match event occurs without taking the comparison result into account.

Single pulse mode is also applicable to composite PWM mode and asymmetric PWM mode.

Figure 21-46. Single pulse mode $TIMERx\_CHxCV=0x04$, $TIMERx\_CAR=0x60$



Timers interconnection

The timers can be internally connected for timer chaining or synchronization. This can be implemented by configuring one timer to operate in the master mode while configuring another timer to be in the slave mode. The following figures show several examples of trigger selection for the master mode and slave mode.

Some interconnection examples:

■ TIMER2 as the prescaler for TIMER0

TIMER2 is configured as a prescaler for TIMER0, steps are shown as follows:

1. Configure TIMER2 in master mode and select its update event (UPE) as trigger output (MMC=3'b010 in the `TIMER2_CTL1` register). Then TIMER2 drives a periodic signal on each counter overflow.
2. Configure TIMER2 period (`TIMER2_CAR` register).
3. Configure TIMER0 in external clock mode 0 and select the TIMER2 as TIMER0 input trigger source (`TRCFG6[4:0] = 5b'00011` in the `_SYSCFG_TIMER0CFG1` register).
4. Start TIMER0 by writing '1' to the CEN bit (`TIMER0_CTL0` register).
5. Start TIMER2 by writing '1' to the CEN bit (`TIMER2_CTL0` register).

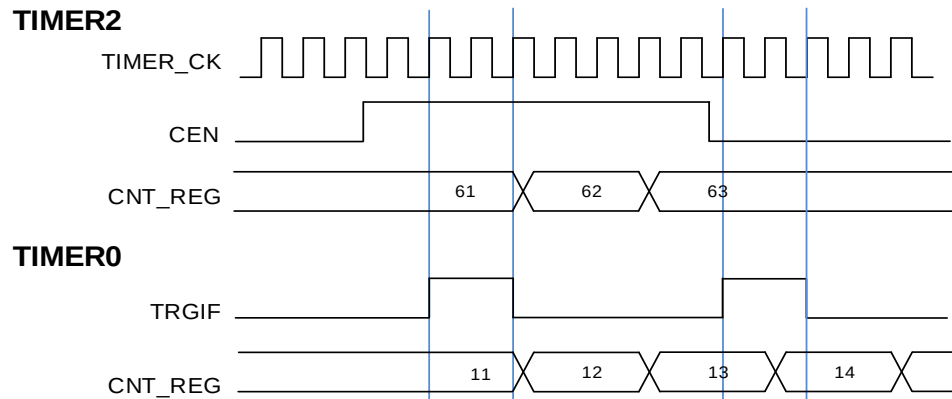
■ Start TIMER0 with TIMER2's enable/update signal

First, enable TIMER0 with the enable signal of TIMER2. Refer to [Figure 21-47. Trigger mode of TIMER0 controlled by enable signal of TIMER2](#). TIMER0 starts counting from its current value with the divided internal clock after being triggered by TIMER2 enable signal output.

When TIMER0 receives the trigger signal, its CEN bit is set automatically and the counter counts until TIMER0 is disabled. Both clock frequency of the counters are divided by 3 from `TIMER_CK` ($f_{PSC\_CLK} = f_{TIMER\_CK} / 3$). Steps are shown as follows:

1. Configure TIMER2 in master mode to send its enable signal as trigger output (MMC=3'b001 in the TIMER2_CTL1 register).
2. Configure TIMER0 in event mode and select the TIMER2 as TIMER0 input trigger source (TRCFG5[4:0] = 5b'00011 in the _SYSCFG_TIMER0CFG0 register).
3. Start TIMER2 by writing 1 to the CEN bit (TIMER2_CTL0 register).

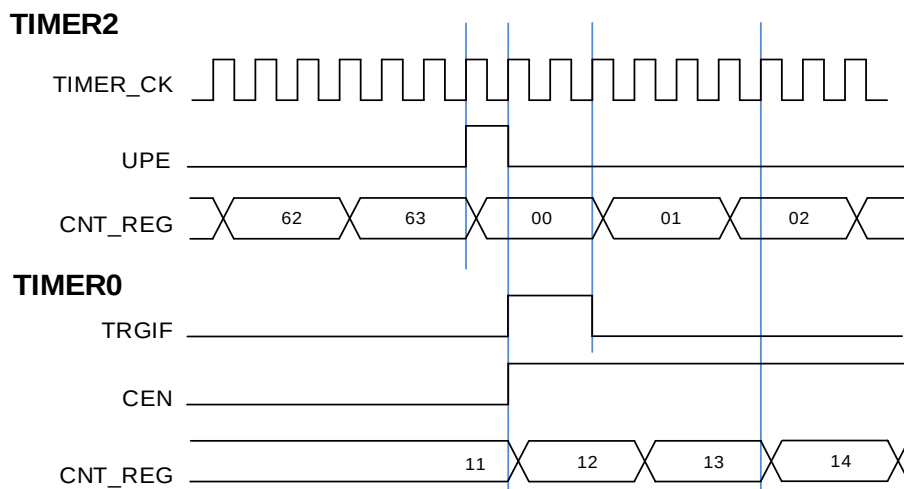
Figure 21-47. Trigger mode of TIMER0 controlled by enable signal of TIMER2



In this example, the update event can also be used as trigger source instead of enable signal. Refer to [Figure 21-48. Trigger mode of TIMER0 controlled by update signal of TIMER2](#). Steps are shown as follows:

1. Configure TIMER2 in master mode to send its update event (UPE) as trigger output (MMC=3'b010 in the TIMER2_CTL1 register).
2. Configure the TIMER2 period (TIMER2_CARL registers).
3. Configure TIMER0 in event mode and select the TIMER2 as TIMER0 input trigger source (TRCFG5[4:0] = 5b'00011 in the _SYSCFG_TIMER0CFG0 register).
4. Start TIMER2 by writing '1' to the CEN bit (TIMER2_CTL0 register).

Figure 21-48. Trigger mode of TIMER0 controlled by update signal of TIMER2

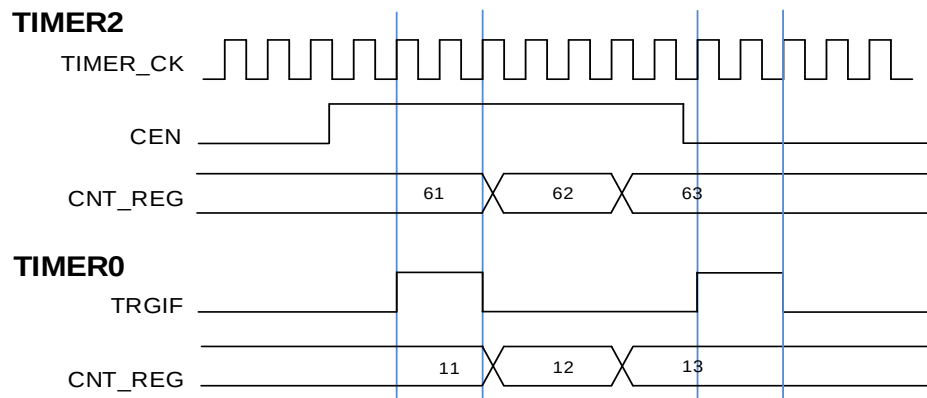


- Enable TIMER0 to count with the enable/O0CPRE signal of TIMER2.

In this example, TIMER0 is enabled with the enable signal of TIMER2. Refer to [Figure 21-49. Pause mode of TIMER0 controlled by enable signal of TIMER2](#). TIMER0 counts with the divided internal clock only when TIMER2 is enabled. Both clock frequency of the counters are divided by 3 from TIMER_CK ($f_{PSC\_CLK} = f_{TIMER\_CK}/3$). Steps are shown as follows:

1. Configure TIMER2 in master mode and output enable signal as trigger output (MMC=3'b001 in the TIMER2_CTL1 register).
2. Configure TIMER0 in pause mode and select the TIMER2 as TIMER0 input trigger source (TRCFG4[4:0] = 5b'00011 in the _SYSCFG_TIMER0CFG0 register).
3. Enable TIMER0 by writing '1' to the CEN bit (TIMER0_CTL0 register).
4. Start TIMER2 by writing '1' to the CEN bit (TIMER2_CTL0 register).
5. Stop TIMER2 by writing '0' to the CEN bit (TIMER2_CTL0 register).

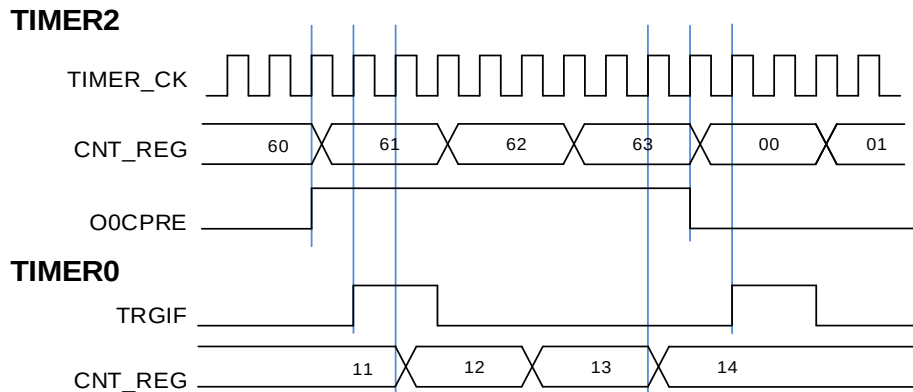
Figure 21-49. Pause mode of TIMER0 controlled by enable signal of TIMER2



In this example, O0CPRE can also be used as trigger source instead of enable signal output. Steps are shown as follows:

1. Configure TIMER2 in master mode and O0CPRE as trigger output (MMC=3'b100 in the TIMER2_CTL1 register).
2. Configure the TIMER2 O0CPRE waveform (TIMER2_CHCTL0 register).
3. Configure TIMER0 in pause mode and select the TIMER2 as TIMER0 input trigger source (TRCFG4[4:0] = 5b'00011 in the _SYSCFG_TIMER0CFG0 register).
4. Enable TIMER0 by writing '1' to the CEN bit (TIMER0_CTL0 register).
5. Start TIMER2 by writing '1' to the CEN bit (TIMER2_CTL0 register).

Figure 21-50. Pause mode of TIMER0 controlled by O0CPRE signal of TIMER2



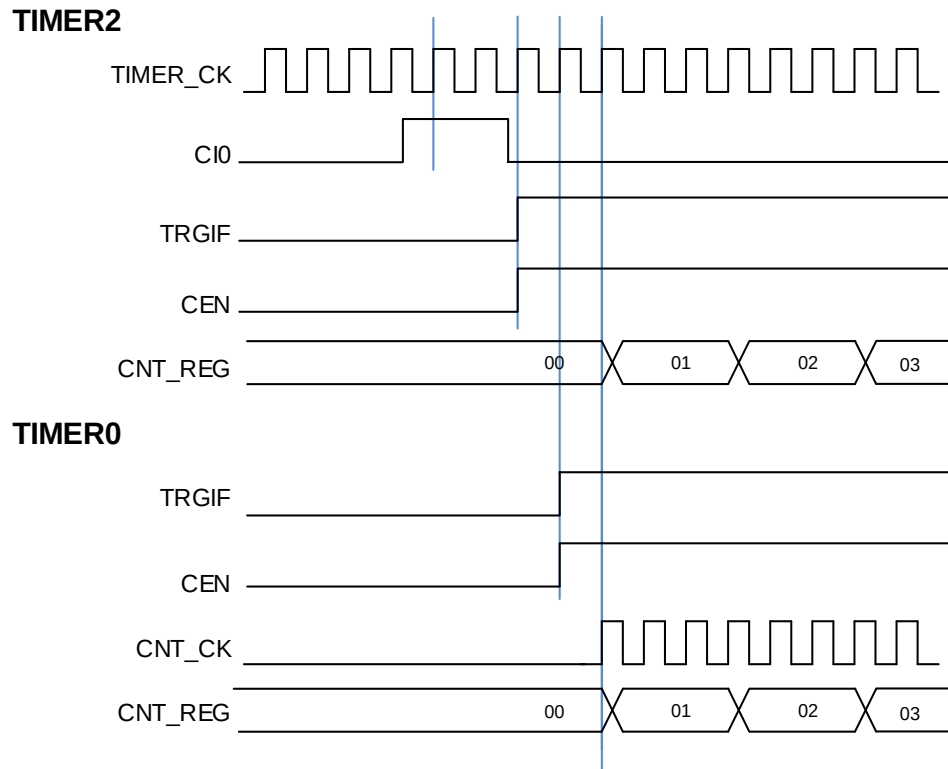
- Using an external trigger to start two timers synchronously.

The start of TIMER0 is triggered by the enable signal of TIMER2, and TIMER2 is triggered by its CI0 input rising edge. To ensure that two timers start synchronously, TIMER2 must be configured in master/slave mode. Steps are shown as follows:

1. Configure TIMER2 in event mode and select the CI0F_ED as TIMER2 input trigger source (TRCFG5[4:0] = 5b'00101 in the _SYSCFG_TIMER2CFG0 register).
2. Configure TIMER2 in master/slave mode by writing MSM=1 (TIMER2_SMCFG register).
3. Configure TIMER0 in event mode and select the TIMER2 as TIMER0 input trigger source (TRCFG5[4:0] = 5b'00011 in the _SYSCFG_TIMER0CFG0 register).

When the CI0 signal of TIMER2 generates a rising edge, two timer counters start counting synchronously with the internal clock and both TRGIF flags are set.

Figure 21-51. Trigger TIMER0 and TIMER2 by the CI0 signal of TIMER2



Trigger event generate and ADC synchronization

Timer can generate triggering events through various internal signals, such as reset, enable, update, or compare events. It can also generate pulses emitted by internal edge detectors.

In addition, specialized ADC comparison events and counter overflow can be used to generate their respective trigger signals and trigger the ADC separately. The following will introduce several ways to trigger ADC separately.

1) Counter overflow triggering mode

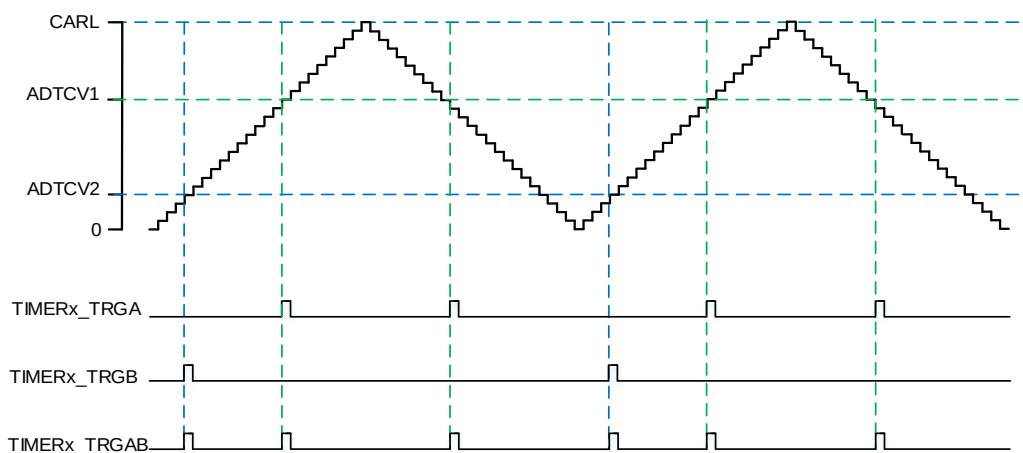
When set ADTSOF bit in `TIMERx_ADCTL` register, counter overflow will generate a separate trigger signal `TIMERx_TRGOF`. When set ADTSUF bit in `TIMERx_ADCTL` register, counter overflow will generate a separate trigger signal `TIMERx_TRGUF`.

2) ADC Compare Match Trigger Mode

The timer has two specialized event triggered comparison registers (`TIMERx_ADCCR1` and `TIMERx_ADCCR2`). When a comparison match occurs, the trigger signals `TIMERx_TRGA` and `TIMERx_TRGB` can be generated to trigger ADC, and they also work together to generate the trigger signal `TIMERx_TRGAB`. The up counting process comparison and matching enable (`ADT1UPGTEN`, `ADT2UPGTEN`) or down counting process comparison and matching enable (`ADT1DWGTEN`, `ADT2DWGTEN`) registers of `TIMERx_TRGA` and `TIMERx_TRGB` can be configured respectively, and the rising edge, falling edge, rising edge and falling edge can be selected respectively to generate trigger signals or not generate trigger signals. As shown in [Figure 21-52. ADC trigger singal operation in ADC compare match trigger](#)

[mode\(ADT1UPGTEN=1、ADT2UPGTEN=1、ADT1DWGTEN=1、ADT2DWGTEN=0\).](#)

Figure 21-52. ADC trigger singal operation in ADC compare match trigger mode(ADT1UPGTEN=1、ADT2UPGTEN=1、ADT1DWGTEN=1、ADT2DWGTEN=0)



Note: In the center-aligned mode, if ADTCVx = 0, one ADC trigger pulse will only be generated when down-count ADC trigger signal enable (ADT1DWGTEN = 1 / ADT2DWGTEN = 1). In the center-aligned mode, if ADTCVx = CARL, one ADC trigger pulse will only be generated when up-count ADC trigger signal enable (ADT1UPGTEN=1 / ADT2UPGTEN=1).

ADTCVx has a corresponding buffer that updates the comparison register based on the relevant configuration of the shadow register loading function.

The table of ADC trigger signals and their corresponding signal generation methods is shown in Table [Table 21-10. TIMER0/TIMER7 ADC trigger signals and their operation](#).

Table 21-10. TIMER0/TIMER7 ADC trigger signals and their operation

ADC Trigger signal	Generation mode
TIMERx_TRGOF	The ADC is triggered by overflow
TIMERx_TRGUF	The ADC is triggered by underflow
TIMERx_TRGA	The ADC trigger compare value 1 (ADTCV1) matches and satisfies the special repeat counter function
TIMERx_TRGB	The ADC trigger compare value 2 (ADTCV2) matches and satisfies the special repeat counter function
TIMERx_TRGAB	The ADC trigger compare value 1 or 2(ADTCV1 or ADTCV2) matches and satisfies the special repeat counter function
TIMERx_TRGAORB	The ADC trigger compare value 1 or 2(ADTCV1 or ADTCV2) matches and satisfies the TIMERx_TRGA, TIMERx_TRGB repeat counter function

Note: In addition to specifically generating ADC trigger signals, TRGO signals generated by reset, enable, update, or compare events, and internal edge detectors can also trigger ADC (according to the MMC register selection, only one trigger signal TRGO can be output).

3) ADC repeat counter function

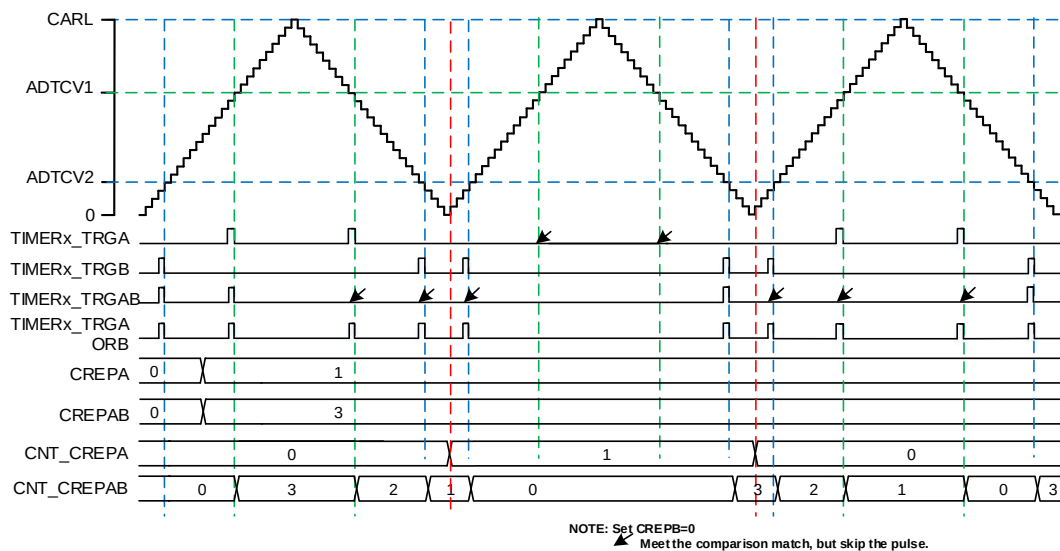
The ADC comparison matching trigger mode has two repeat counter modes, the periodic repeat counter mode, and the comparison matching repeat counter mode.

The periodic repeat counter mode applies the `TIMERx_TRGA` or `TIMERx_TRGB` signal. According to the set 4bit ADC repeat counter value `ADCREPA` and `ADCREPB`, the repeat counter function of the `TIMERx_TRGA` or `TIMERx_TRGB` signal of the ADC is realized respectively. When `TIMERx_TRGA` or `TIMERx_TRGB` is generated each time through the overflow or underflow of the `TIMER`, before the event trigger signal is generated, the counter is continuously decreased until it reaches 0 and takes effect. When the repeat counter is 0, the repeat counter function is not used.

The comparison matching repeat counter pattern is applied to the `TIMERx_TRGAB` signal. According to the set value of the 4bit ADC repeat counter `ADCREPAB`, the repeat counter function of the `TIMERx_TRGAB` signal of the ADC can be realized. According to the valid event-triggered signals `TIMERx_TRGA`, `TIMERx_TRGB`, the repeat counter for the `TIMERx_TRGAB` signal is decremented from the `ADCREPAB` repeat count value until the repeat counter reaches 0, and the event-triggered signal `TIMERx_TRGAB` is generated. When the repeat counter is 0, the repeat counter function is not used.

As is shown in [Figure 21-53](#).

Figure 21-53. ADC repetition counting function in ADC trigger mode(`ADT1UPGTEN =1`、`ADT2UPGTEN =1`、`ADT1DWGTEN =1`、`ADT2DWGTEN =1`)



Note: When change the value of `ADCREPA`, `ADCREPB`, and `ADCREPAB`, the value of the repeat counter is not updated immediately, but only when the corresponding counter reaches 0. If the `ADREPRLD` bit of the `TIMERx_ADCTL` register is set, the value of the repeat counter is updated immediately.

The specific operating steps for ADC synchronization using the ADC repeat counter are as follows:

1. Set the comparison value in the ADC trigger compare value register (`ADTCV1` and

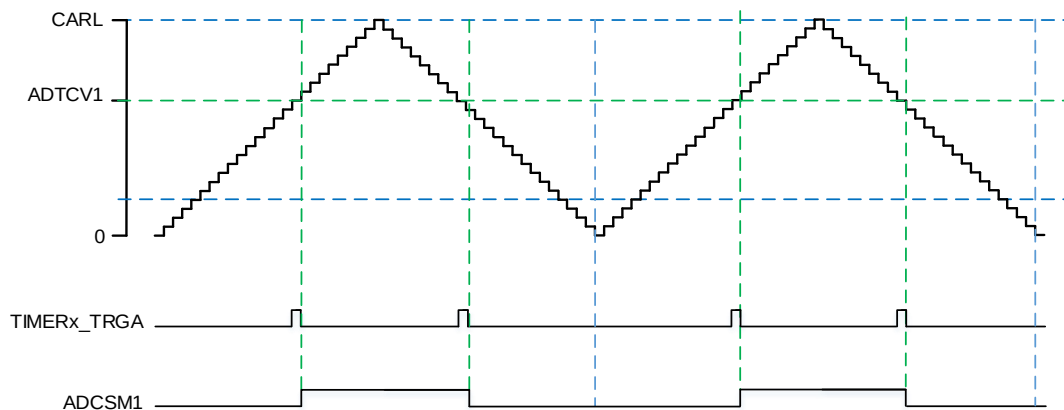
ADTCV2).

2. Set the shadow register loading function related registers and set when the comparison value is updated.
3. Configure the values of ADC repeat counters separately
4. Configure ADCREPA and ADCREPB to decrement when the timer overflow or underflow. Optional repeat counter immediately reload function;
5. Set the edge comparison matching enable (ADT1UPGTEN 、 ADT2UPGTEN 、 ADT1DWGTEN、 ADT2DWGTEN) to enable ADC trigger signal generation.

4) ADC trigger signal monitoring function

Use external pin ADCSM1/2 to monitor the ADC trigger start signal. The ADC trigger signal monitoring selection registers (ADCSM1SEL, ADCSM2SEL) respectively select the ADC trigger signal to be monitored by ADCSM1/2, and ADCSM1/2 outputs it to an external PIN. When the ADC trigger start signal is detected, the level is pulled up, and when the counter after the cycle ends, the level is pulled down. The r update event from the reset mode and software configuration UPG updates the counter, and then the level of the monitoring signal is also low level. As is shown in [Figure 21-54. Timing diagram of ADC trigger synchronization monitor signal \(ADCSM1SEL =3'b010\).](#)

Figure 21-54. Timing diagram of ADC trigger synchronization monitor signal (ADCSM1SEL =3'b010)



Timer DMA mode

Timer DMA mode is the function that configures timer's register by DMA module. The relative registers are TIMERx_DMCFG and TIMERx_DMATB. Corresponding DMA request bit should be asserted to enable DMA request for internal interrupt event. TIMERx will send a request to DMA when the interrupt event occurs. DMA is configured to M2P (memory to peripheral) mode and the address of TIMERx_DMATB is configured to PADDR (peripheral base address), then DMA will access the TIMERx_DMATB. In fact, TIMERx_DMATB register is only a buffer, timer will map the TIMERx_DMATB to an internal register, appointed by the field of DMATA in TIMERx_DMCFG. If the field of DMATC in TIMERx_DMCFG is 0 (1

transfer), the timer sends only one DMA request. While if `TIMERx_DMATC` is not 0, such as 3 (4 transfers), then timer will send 3 more requests to DMA, and DMA will access timer's registers `DMATA+0x4`, `DMATA+0x8` and `DMATA+0xC` at the next 3 accesses to `TIMERx_DMATB`. In a word, one-time DMA internal interrupt event asserts, (`DMATC+1`) times request will be sent by `TIMERx`.

If one more DMA request event occurs, `TIMERx` will repeat the process above.

Output DIR bit

The DIR bit can output on the CH2 and CH3, and this function is enabled by setting `CH2COMCTL[3:0]` or `CH3COMCTL[3:0]` bit-field to 4b'1011 in the `TIMERx_CHCTL1` register.

When the counter works in center-aligned counting mode, this function can be used to indicates the counter direction of the counter. When the counter works in decoder modes, this function can be used to indicates the rotation direction for the external signal.

Backup UPIF bit

The UPIF bit backup function is enabled by setting `UPIFBUEN` in the `TIMERx_CTL0` register. The UPIF and UPIFBU bits are fully synchronized and without latency.

By using this function, the UPIF bit in the `TIMERx_INTF` register will be backed up to the UPIFBU bit in the `TIMERx_CNT` register. This can avoid conflicts when reading the counter and interrupt processing.

Timer debug mode

When the Cortex®-M33 is halted, and the `TIMERx_HOLD` configuration bit in `DBG_CTL` register is set to 1, the `TIMERx` counter stops.

21.1.5 Registers definition (TIMERx, x=0, 7)

TIMER0 base address: 0x4001 2C00

TIMER7 base address: 0x4001 3400

Control register 0 (TIMERx_CTL0)

Address offset: 0x00

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UPSSEL[1:0]		INTCOSEL[1:0]		Reserved		ETISEL[1:0]		Reserved	INTREPR LD	Reserved					
rw		rw				rw			rw						
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				UPIFBUE N	Reserved	CKDIV[1:0]		ARSE	CAM[1:0]		DIR	SPM	UPS	UPDIS	CEN
				rw		rw		rw	rw		rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:30	UPSSEL[1:0]	Update repeat counter overflow source selection 00: A update repeat counter decrement is triggered by a counter overflow / underflow, and a update event is generated when the update repeat counter decreases to 0. 01: A update repeat counter decrement is triggered by a counter overflow, and a update event is generated when the update repeat counter decreases to 0. 10: A update repeat counter decrement is triggered by a counter underflow, and a update event is generated when the update repeat counter decreases to 0. 11: Reserved Note: In a separate count up or count down mode, with 01 or 10 configured, the repeat counter decreases upon overflow (count up is overflow, count down is underflow), and an update event occurs upon overflow when the repeat counter decreases to 0.
29:28	INTCOSEL[1:0]	Interrupt repeat counter overflow source selection 00: An interrupt repeat counter decrement is triggered by a counter overflow / underflow, and an interrupt event is generated when the interrupt repeat counter decreases to 0. 01: An interrupt repeat counter decrement is triggered by a counter overflow, and an interrupt event is generated when the interrupt repeat counter decreases to 0. 10: An interrupt repeat counter decrement is triggered by a counter underflow, and an interrupt event is generated when the interrupt repeat counter decreases to 0. 11: Reserved

Note: In a separate count up or count down mode, with 01 or 10 configured, the repeat counter decreases upon overflow (count up is overflow, count down is underflow), and an update event occurs upon overflow when the repeat counter decreases to 0.

27:26	Reserved	Must be kept at reset value.
25:24	ETISEL[1:0]	Timer external trigger input signal select 00: Select TIMER_ETI0 pin from GPIO. 01: Select TIMER_ETI1 pin from GPIO. 10~11:Reserved
23	Reserved	Must be kept at reset value.
22	INTREPRD	Interrupt repeat counter immediate reload When this bit is set to 1, the interrupt repeat counter is reloaded to the respective repeat count setting value. After the reload is complete, this bit is automatically cleared by hardware.
21:12	Reserved	Must be kept at reset value.
11	UPIFBUEN	UPIF bit backup enable 0: Backup disable. UPIF bit is not backuped to UPIFBU bit in TIMEx_CNT register. 1: Backup enabled. UPIF bit is backuped to UPIFBU bit in TIMEx_CNT register.
10	Reserved	Must be kept at reset value.
9:8	CKDIV[1:0]	Clock division The CKDIV bits can be configured by software to specify division ratio between CK_TIMER (the timer clock) and DTS (the dead time and sampling clock) which is used for the dead time generator and the digital filter. 00: $f_{DTS} = f_{CK\_TIMER}$ 01: $f_{DTS} = f_{CK\_TIMER} / 2$ 10: $f_{DTS} = f_{CK\_TIMER} / 4$ 11: Reserved
7	ARSE	Auto-reload shadow enable 0: The shadow register for TIMEx_CAR register is disabled 1: The shadow register for TIMEx_CAR register is enabled
6:5	CAM[1:0]	Counter align mode selection 00: No center-aligned mode (edge-aligned mode). The direction of the counter is specified by the DIR bit. 01: Center-aligned and counting down assert mode. The counter counts in center-aligned mode and channel is configured in output mode (CHxMS = 3'b000 in TIMEx_CHCTL0 register). Only when the counter is counting down, compare interrupt flag of channels can be set. 10: Center-aligned and counting up assert mode. The counter counts in center-

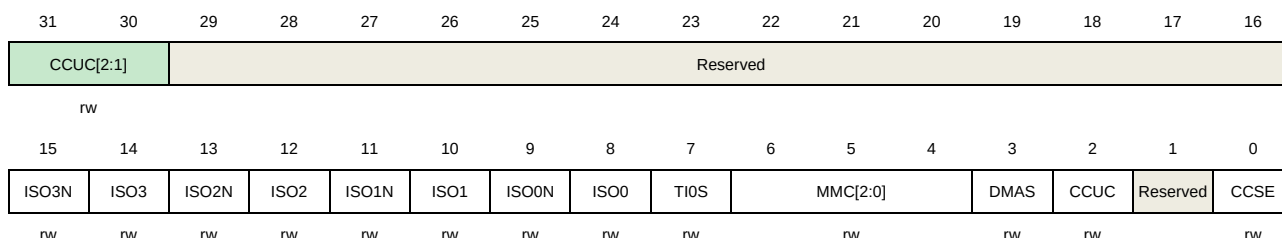
		aligned mode and channel is configured in output mode (CHxMS = 3'b000 in TIMERx_CHCTL0 register). Only when the counter is counting up, compare interrupt flag of channels can be set.
		11: Center-aligned and counting up/down assert mode. The counter counts in center-aligned mode and channel is configured in output mode (CHxMS = 3'b000 in TIMERx_CHCTL0 register). Both when the counter is counting up and counting down, compare interrupt flag of channels can be set.
		After the counter is enabled, these bits cannot be switched from 0x00 to non 0x00.
4	DIR	Direction 0: Count up 1: Count down This bit is read only when the timer is configured in center-aligned mode or decoder mode.
3	SPM	Single pulse mode 0: Single pulse mode is disabled. Counter continues after an update event. 1: Single pulse mode is enabled. The CEN bit is cleared by hardware and the counter stops at next update event.
2	UPS	Update source This bit is used to select the update event sources by software. 0: Any of the following events generates an update interrupt or a DMA request: - The UPG bit is set. - The counter generates an overflow or underflow event. - The slave mode controller generates an update event. 1: Only counter overflow/underflow generates an update interrupt or a DMA request.
1	UPDIS	Update disable This bit is used to enable or disable the update event generation. 0: Update event enable. The update event is generated and the buffered registers are loaded with their preloaded values when one of the following events occurs: - The UPG bit is set. - The counter generates an overflow or underflow event. - The slave mode controller generates an update event. 1: Update event disable. The buffered registers keep their value, while the counter and the prescaler are reinitialized if the UG bit is set or the slave mode controller generates a hardware reset event.
0	CEN	Counter enable 0: Counter disable 1: Counter enable The CEN bit must be set by software when timer works in external clock mode, pause mode or decoder mode. While in event mode, the hardware can set the CEN bit automatically.

Control register 1 (TIMERx_CTL1)

Address offset: 0x04

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



Bits	Fields	Descriptions
31:30	CCUC[2:1]	Commutation control shadow register update control Refer to CCUC[0] description.
29:16	Reserved	Must be kept at reset value.
15	ISO3N	Idle state of multi mode channel 3 complementary output. Refer to ISO0N bit.
14	ISO3	Idle state of channel 3 output. Refer to ISO0 bit.
13	ISO2N	Idle state of multi mode channel 2 complementary output. Refer to ISO0N bit
12	ISO2	Idle state of channel 2 output Refer to ISO0 bit
11	ISO1N	Idle state of multi mode channel 1 complementary output Refer to ISO0N bit
10	ISO1	Idle state of channel 1 output Refer to ISO0 bit
9	ISO0N	Idle state of multi mode channel 0 complementary output 0: When POEN bit is reset, MCH0_O is set low. 1: When POEN bit is reset, MCH0_O is set high. This bit can be modified only when PROT[1:0] bits in TIMERx_CCHP register is 00.
8	ISO0	Idle state of channel 0 output 0: When POEN bit is reset, CH0_O is set low. 1: When POEN bit is reset, CH0_O is set high. The CH0_O output changes after a dead time if MCH0_O is implemented. This bit can be modified only when PROT[1:0] bits in TIMERx_CCHP register is 00.
7	TI0S	Channel 0 trigger input selection

		0: The TIMERx_CH0 pin input is selected as channel 0 trigger input. 1: The result of combinational XOR of TIMERx_CH0, TIMERx_CH1 and TIMERx_CH2 pins is selected as channel 0 trigger input.
6:4	MMC[2:0]	Master mode control These bits control the selection of TRGO signal, which is sent by master timer to slave timer for synchronization function. 000: Reset. When the UPG bit in the TIMERx_SWEVG register is set or a reset is generated by the slave mode controller, a TRGO pulse occurs. And in the latter case, the signal on TRGO is delayed compared to the actual reset. 001: Enable. This mode is used to start several timers at the same time or control a slave timer to be enabled in a period. In this mode, the master mode controller selects the counter enable signal as TRGO. The counter enable signal is set when CEN control bit is set or the trigger input in pause mode is high. There is a delay between the trigger input in pause mode and the TRGO output, except if the master-slave mode is selected. 010: Update. In this mode, the master mode controller selects the update event as TRGO. 011: Capture/compare pulse. In this mode, the master mode controller generates a TRGO pulse when a capture or a compare match occurs in channel 0. 100: Compare. In this mode, the master mode controller selects the O0CPRE signal as TRGO. 101: Compare. In this mode, the master mode controller selects the O1CPRE signal as TRGO. 110: Compare. In this mode, the master mode controller selects the O2CPRE signal as TRGO. 111: Compare. In this mode, the master mode controller selects the O3CPRE signal as TRGO.
3	DMAS	DMA request source selection 0: DMA request of CHx/MCHx is sent when capture/compare event occurs. 1: DMA request of channel CHx/MCHx is sent when update event occurs.
2	CCUC[0]	Commutation control shadow register update control The CCUC[2:1] and CCUC[0] field are used to control the commutation control shadow register update. When the commutation control shadow registers (for CHxEN, MCHxEN and CHxCOMCTL bits) are enabled (CCSE=1), the update control of the shadow registers are shown as below: 000: The shadow registers update when CMTG bit is set. 001: The shadow registers update when CMTG bit is set or a rising edge of TRGI occurs. 100: The shadow registers update when the counter generates an overflow event. 101: The shadow registers update when the counter generates an underflow event. 110: The shadow registers update when the counter generates an overflow or underflow event.

Others: Reserved

When a channel does not have a complementary output, this bit has no effect.

1 Reserved

Must be kept at reset value.

0 CCSE

Commutation control shadow enable

0: The shadow registers (for CHxEN, MCHxEN and CHxCOMCTL bits) are disabled.

1: The shadow registers (for CHxEN, MCHxEN and CHxCOMCTL bits) are enabled.
After these bits have been written, they are updated when commutation event comes.

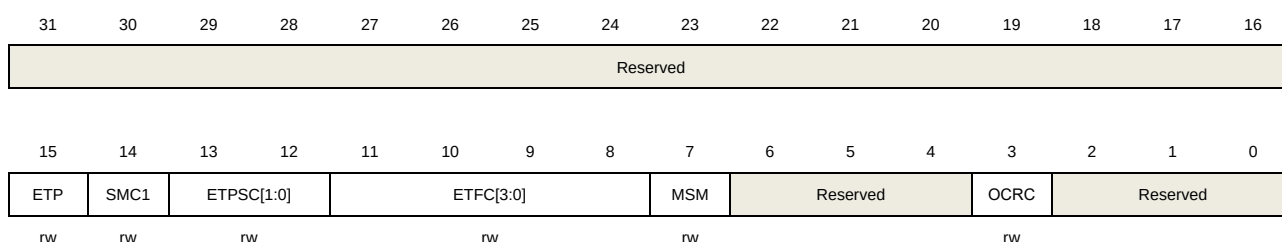
When a channel does not have a complementary output, this bit has no effect.

Slave mode configuration register (TIMERx_SMCFG)

Address offset: 0x08

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15	ETP	External trigger polarity This bit specifies the polarity of ETI signal. 0: ETI is active at high level or rising edge. 1: ETI is active at low level or falling edge.
14	SMC1	Part of slave mode controller is used to enable external clock mode 1 In external clock mode 1, the counter is clocked by any active edge of the ETIFP signal. 0: External clock mode 1 disabled 1: External clock mode 1 enabled It is possible to simultaneously use external clock mode 1 with the restart mode, pause mode or event mode. But the TSCFGy[4:0](y=3,4,5) bits must not be 5b'01000 in this case. The external clock input will be ETIFP if external clock mode 0 and external clock mode 1 are enabled at the same time. Note: External clock mode 0 enable is in TSCFG6[4:0] bit-field in

		SYSCFG_TIMERxCFG1 register.
13:12	ETPSC[1:0]	External trigger prescaler The frequency of external trigger signal ETIFP must not be higher than 1/4 of TIMER_CK frequency. When the frequency of external trigger signal is high, the prescaler can be enabled to reduce ETIFP frequency. 00: Prescaler disabled 01: ETIFP frequency divided by 2 10: ETIFP frequency divided by 4 11: ETIFP frequency divided by 8
11:8	ETFC[3:0]	External trigger filter control An event counter is used in the digital filter, in which a transition on the output occurs after N input events. This bit-field specifies the frequency used to sample ETIFP signal and the length of the digital filter applied to ETIFP. 0000: Filter disabled. $f_{SAMP} = f_{DTS}$, $N=1$. 0001: $f_{SAMP} = f_{CK\_TIMER}$, $N=2$. 0010: $f_{SAMP} = f_{CK\_TIMER}$, $N=4$. 0011: $f_{SAMP} = f_{CK\_TIMER}$, $N=8$. 0100: $f_{SAMP} = f_{DTS}/2$, $N=6$. 0101: $f_{SAMP} = f_{DTS}/2$, $N=8$. 0110: $f_{SAMP} = f_{DTS}/4$, $N=6$. 0111: $f_{SAMP} = f_{DTS}/4$, $N=8$. 1000: $f_{SAMP} = f_{DTS}/8$, $N=6$. 1001: $f_{SAMP} = f_{DTS}/8$, $N=8$. 1010: $f_{SAMP} = f_{DTS}/16$, $N=5$. 1011: $f_{SAMP} = f_{DTS}/16$, $N=6$. 1100: $f_{SAMP} = f_{DTS}/16$, $N=8$. 1101: $f_{SAMP} = f_{DTS}/32$, $N=5$. 1110: $f_{SAMP} = f_{DTS}/32$, $N=6$. 1111: $f_{SAMP} = f_{DTS}/32$, $N=8$.
7	MSM	Master-slave mode This bit can be used to synchronize the selected timers to begin counting at the same time. The TRGI is used as the start event, and through TRGO, timers are connected. 0: Master-slave mode disabled 1: Master-slave mode enabled
6:4	Reserved	Must be kept at reset value.
3	OCRC	OxCPRE/ MOxCPRE clear source selection This bit is used to select the OCREF clear source. 0: OCPRE_CLR_INT is connected to the OCPRE_CLR input 1: OCPRE_CLR_INT is connected to ETIF

2:0 Reserved Must be kept at reset value.

DMA and interrupt enable register (TIMERx_DMAINTEN)

Address offset: 0x0C

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CH3COM ADDIE	CH2COM ADDIE	CH1COM ADDIE	CH0COM ADDIE	MCH3 DEN	MCH2 DEN	MCH1 DEN	MCH0 DEN	MCH3IE	MCH2IE	MCH1IE	MCH0IE	Reserved	FLOWIE	Reserved	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw		rw		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	TRGDEN	CMTDEN	CH3DEN	CH2DEN	CH1DEN	CH0DEN	UPDEN	BRKIE	TRGIE	CMTIE	CH3IE	CH2IE	CH1IE	CH0IE	UPIE
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31	CH3COMADDIE	Channel 3 additional compare interrupt enable 0: Disabled 1: Enabled Note: This bit just used in composite PWM mode and asymmetric PWM mode.
30	CH2COMADDIE	Channel 2 additional compare interrupt enable 0: Disabled 1: Enabled Note: This bit just used in composite PWM mode and asymmetric PWM mode.
29	CH1COMADDIE	Channel 1 additional compare interrupt enable 0: Disabled 1: Enabled Note: This bit just used in composite PWM mode and asymmetric PWM mode.
28	CH0COMADDIE	Channel 0 additional compare interrupt enable 0: Disabled 1: Enabled Note: This bit just used in composite PWM mode and asymmetric PWM mode.
27	MCH3DEN	Multi mode channel 3 capture/compare DMA request enable 0: Disabled 1: Enabled Note: This bit just used for channel input and output independent mode (when MCH3MSEL[1:0] = 2b'00).
26	MCH2DEN	Multi mode channel 2 capture/compare DMA request enable 0: Disabled

		1: Enabled Note: This bit just used for channel input and output independent mode (when MCH2MSEL[1:0] = 2b'00).
25	MCH1DEN	Multi mode channel 1 capture/compare DMA request enable 0: Disabled 1: Enabled Note: This bit just used for channel input and output independent mode (when MMCH1SEL[1:0] = 2b'00).
24	MCH0DEN	Multi mode channel 0 capture/compare DMA request enable 0: Disabled 1: Enabled Note: This bit just used for channel input and output independent mode (when MMCH0SEL[1:0] = 2b'00).
23	MCH3IE	Multi mode channel 3 capture/compare interrupt enable 0: Disabled 1: Enabled Note: This bit just used for channel input and output independent mode (when MCH3MSEL[1:0] = 2b'00).
22	MCH2IE	Multi mode channel 2 capture/compare interrupt enable 0: Disabled 1: Enabled Note: This bit just used for channel input and output independent mode (when MCH2MSEL[1:0] = 2b'00).
21	MCH1IE	Multi mode channel 1 capture/compare interrupt enable 0: Disabled 1: Enabled Note: This bit just used for channel input and output independent mode (when MMCH1SEL[1:0] = 2b'00).
20	MCH0IE	Multi mode channel 0 capture/compare interrupt enable 0: Disabled 1: Enabled Note: This bit just used for channel input and output independent mode (when MMCH0SEL[1:0] = 2b'00).
19	Reserved	Must be kept at reset value.
18	FLOWIE	Overflow or underflow interrupt enable 0: Disabled 1: Enabled
17:15	Reserved	Must be kept at reset value.

14	TRGDEN	Trigger DMA request enable 0: Disabled 1: Enabled
13	CMTDEN	Commutation DMA request enable 0: Disabled 1: Enabled
12	CH3DEN	Channel 3 capture/compare DMA request enable 0: Disabled 1: Enabled
11	CH2DEN	Channel 2 capture/compare DMA request enable 0: Disabled 1: Enabled
10	CH1DEN	Channel 1 capture/compare DMA request enable 0: Disabled 1: Enabled
9	CH0DEN	Channel 0 capture/compare DMA request enable 0: Disabled 1: Enabled
8	UPDEN	Update DMA request enable 0: Disabled 1: Enabled
7	BRKIE	Break interrupt enable 0: Disabled 1: Enabled
6	TRGIE	Trigger interrupt enable 0: Disabled 1: Enabled
5	CMTIE	Commutation interrupt enable 0: Disabled 1: Enabled
4	CH3IE	Channel 3 capture/compare interrupt enable 0: Disabled 1: Enabled
3	CH2IE	Channel 2 capture/compare interrupt enable 0: Disabled 1: Enabled
2	CH1IE	Channel 1 capture/compare interrupt enable

		0: Disabled 1: Enabled
1	CH0IE	Channel 0 capture/compare interrupt enable 0: Disabled 1: Enabled
0	UPIE	Update interrupt enable 0: Disabled 1: Enabled

Interrupt flag register (TIMERx_INTF)

Address offset: 0x10

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CH3COM ADDIF	CH2COM ADDIF	CH1COM ADDIF	CH0COM ADDIF	MCH3OF	MCH2OF	MCH1OF	MCH0OF	MCH3IF	MCH2IF	MCH1IF	MCH0IF	OVERFL OWIF	UNDERF LOWIF	Reserved	
rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		SYSBIF	CH3OF	CH2OF	CH1OF	CH0OF	Reserved	BRKIF	TRGIF	CMTIF	CH3IF	CH2IF	CH1IF	CH0IF	UPIF
		rc_w0	rc_w0	rc_w0	rc_w0	rc_w0		rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0

Bits	Fields	Descriptions
31	CH3COMADDIF	Channel 3 additional compare interrupt flag. Refer to CH0COMADDIF description.
30	CH2COMADDIF	Channel 2 additional compare interrupt flag. Refer to CH0COMADDIF description.
29	CH1COMADDIF	Channel 1 additional compare interrupt flag. Refer to CH0COMADDIF description.
28	CH0COMADDIF	Channel 0 additional compare interrupt flag. This flag is set by hardware and cleared by software. If channel 0 is in output mode, this flag is set when a compare event occurs. 0: No channel 0 output compare interrupt occurred 1: Channel 0 output compare interrupt occurred Note: This flag just used in composite PWM mode and asymmetric PWM mode.
27	MCH3OF	Multi mode channel 3 over capture flag Refer to MCH0OF description.
26	MCH2OF	Multi mode channel 2 over capture flag Refer to MCH0OF description.

25	MCH1OF	Multi mode channel 1 over capture flag Refer to MCH0OF description.
24	MCH0OF	Multi mode channel 0 over capture flag When multi mode channel 0 is configured in input mode, this flag is set by hardware when a capture event occurs while MCH0IF flag has already been set. This flag is cleared by software. 0: No over capture interrupt occurred 1: Over capture interrupt occurred
23	MCH3IF	Multi mode channel 3 capture/compare interrupt flag Refer to MCH0IF description
22	MCH2IF	Multi mode channel 2 capture/compare interrupt flag Refer to MCH0IF description
21	MCH1IF	Multi mode channel 1 capture/compare interrupt flag Refer to MCH0IF description
20	MCH0IF	Multi mode channel 0 capture/compare interrupt flag This flag is set by hardware and cleared by software. If multi mode channel 0 is in input mode, this flag is set when a capture event occurs. If multi mode channel 0 is in output mode, this flag is set when a compare event occurs. If multi mode channel 0 is set to input mode, this bit will be reset by reading TIMEx_MCH0CV. 0: No multi mode channel 0 capture/compare interrupt occurred 1: Multi mode channel 0 capture/compare interrupt occurred
19	OVERFLOWIF	Overflow interrupt flag This flag is set by hardware when an interrupt repetition counter overflow event(counter value =IREP) occurs. It is cleared by software by writing it to '0'. 0: No overflow interrupt occurs 1: Overflow interrupt occurs Note: the overflow or underflow of interrupt occurs by INTCOSEL bits.
18	UNDERFLOWIF	Underflow interrupt flag This flag is set by hardware when an interrupt repetition counter underflow event(counter value =IREP) occurs. It is cleared by software by writing it to '0'. 0: No underflow interrupt occurs 1: Underflow interrupt occurs Note: the overflow or underflow of interrupt occurs by INTCOSEL bits.
17:14	Reserved	Must be kept at reset value.
13	SYSBIF	System source break interrupt flag This flag is set by hardware when the system sources are active, and cleared by software if the system sources are inactive.

		0: No system source break interrupt occurred 1: System source break interrupt occurred Note: When this bit is set, this bit must be cleared by software before the channel outputs are restored.
12	CH3OF	Channel 3 over capture flag Refer to CH0OF description
11	CH2OF	Channel 2 over capture flag Refer to CH0OF description
10	CH1OF	Channel 1 over capture flag Refer to CH0OF description
9	CH0OF	Channel 0 over capture flag When channel 0 is configured in input mode, this flag is set by hardware when a capture event occurs while CH0IF flag has already been set. This flag is cleared by software. 0: No over capture interrupt occurred 1: Over capture interrupt occurred
8	Reserved	Must be kept at reset value.
7	BRKIF	Break interrupt flag This flag is set by hardware when the break input is active, and cleared by software if the break input is not at active level. 0: No active level on break input has been detected. 1: An active level on break input has been detected.
6	TRGIF	Trigger interrupt flag This flag is set by hardware on trigger event and cleared by software. When the slave mode controller is enabled in all modes but pause mode, an active edge of trigger input generates a trigger event. When the slave mode controller is enabled in pause mode, either edge of the trigger input can generate a trigger event. 0: No trigger event occurred 1: Trigger interrupt occurred
5	CMTIF	Channel commutation interrupt flag This flag is set by hardware when the commutation event of channel occurs, and cleared by software. 0: No channel commutation interrupt occurred 1: Channel commutation interrupt occurred
4	CH3IF	Channel 3 capture/compare interrupt flag Refer to CH0IF description
3	CH2IF	Channel 2 capture/compare interrupt flag Refer to CH0IF description

2	CH1IF	Channel 1 capture/compare interrupt flag Refer to CH0IF description
1	CH0IF	Channel 0 capture/compare interrupt flag This flag is set by hardware and cleared by software. If channel 0 is in input mode, this flag is set when a capture event occurs. If channel 0 is in output mode, this flag is set when a compare event occurs. If channel 0 is set to input mode, this bit will be reset by reading TIMEx_CH0CV. 0: No channel 0 interrupt occurred 1: Channel 0 interrupt occurred
0	UPIF	Update interrupt flag This bit is set by hardware when an update event occurs and cleared by software. 0: No update interrupt occurred 1: Update interrupt occurred

Software event generation register (TIMEx_SWEVG)

Address offset: 0x14

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CH3COM ADDG	CH2COM ADDG	CH1COM ADDG	CH0COM ADDG	Reserved				MCH3G	MCH2G	MCH1G	MCH0G	Reserved			
w	w	w	w					w	w	w	w				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								BRKG	TRGG	CMTG	CH3G	CH2G	CH1G	CH0G	UPG
w								w	w	w	w	w	w	w	w

Bits	Fields	Descriptions
31	CH3COMADDG	Channel 3 additional compare event generation. Refer to CH0COMADDG description.
30	CH2COMADDG	Channel 2 additional compare event generation. Refer to CH0COMADDG description.
29	CH1COMADDG	Channel 1 additional compare event generation. Refer to CH0COMADDG description.
28	CH0COMADDG	Channel 0 additional compare event generation. This bit is set by software to generate a compare event in channel 0 additional , it is automatically cleared by hardware. When this bit is set, the CH0COMADDIF flag will be set, and the corresponding interrupt will be sent if enabled. 0: No generate a channel 0 additional compare event

		1: Generate a channel 0 additional compare event
		Note: This bit just used in composite PWM mode and asymmetric PWM mode.
27:24	Reserved	Must be kept at reset value.
23	MCH3G	Multi mode channel 3 capture or compare event generation. Refer to MCH0G description.
22	MCH2G	Multi mode channel 2 capture or compare event generation. Refer to MCH0G description.
21	MCH1G	Multi mode channel 1 capture or compare event generation. Refer to MCH0G description.
20	MCH0G	Multi mode channel 0 capture or compare event generation. This bit is set by software to generate a capture or compare event in multi mode channel 0, it is automatically cleared by hardware. When this bit is set, the MCH0IF flag will be set, and the corresponding interrupt or DMA request will be sent if enabled. In addition, if multi mode channel 0 is configured in input mode, the current value of the counter is captured to TIMEx_MCH0CV register, and the MCH0OF flag is set if the MCH0IF flag has been set. 0: No generate a multi mode channel 0 capture or compare event 1: Generate a multi mode channel 0 capture or compare event
19:8	Reserved	Must be kept at reset value.
7	BRKG	Break event generation This bit is set by software to generate an event and cleared by hardware automatically. When this bit is set, the POEN bit will be cleared and BRKIF flag will be set, related interrupt or DMA transfer can occur if enabled. 0: No generate a break event 1: Generate a break event
6	TRGG	Trigger event generation This bit is set by software and cleared by hardware automatically. When this bit is set, the TRGIF flag in TIMEx_INTF register will be set, related interrupt or DMA transfer can occur if enabled. 0: No generate a trigger event 1: Generate a trigger event
5	CMTG	Channel commutation event generation This bit is set by software and cleared by hardware automatically. When this bit is set, channel's capture/compare control registers (CHxEN, MCHxEN and CHxCOMCTL bits) are updated based on the value of CCSE (in the TIMEx_CTL1). 0: No affect 1: Generate channel commutation update event

4	CH3G	Channel 3 capture or compare event generation Refer to CH0G description
3	CH2G	Channel 2 capture or compare event generation Refer to CH0G description
2	CH1G	Channel 1 capture or compare event generation Refer to CH0G description
1	CH0G	Channel 0 capture or compare event generation This bit is set by software to generate a capture or compare event in channel 0, it is automatically cleared by hardware. When this bit is set, the CH0IF flag will be set, and the corresponding interrupt or DMA request will be sent if enabled. In addition, if channel 0 is configured in input mode, the current value of the counter is captured to TIMEx_CH0CV register, and the CH0OF flag is set if the CH0IF flag has been set. 0: No generate a channel 0 capture or compare event 1: Generate a channel 0 capture or compare event
0	UPG	Update event generation This bit can be set by software, and automatically cleared by hardware. When this bit is set, the counter is cleared if the center-aligned or up counting mode is selected, while in down counting mode it takes the auto-reload value. The prescaler counter is cleared at the same time. 0: No generate an update event 1: Generate an update event

Channel control register 0 (TIMEx_CHCTL0)

Address offset: 0x18

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CH1MS [2]	CH0MS [2]	CH1COM ADDSEN	CH0COM ADDSEN	Reserved			CH1COM CTL[3]	Reserved							CH0COM CTL[3]
		Reserved	Reserved				Reserved								Reserved
		rw	rw	rw	rw				rw						
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH1COM CEN	CH1COMCTL[2:0]			CH1COM SEN	CH1COM FEN	CH1MS[1:0]		CH0COM CEN	CH0COMCTL[2:0]			CH0COM SEN	CH0COM FEN	CH0MS[1:0]	
CH1CAPFLT[3:0]			CH1CAPPSC[1:0]		CH0CAPFLT[3:0]			CH0CAPPSC[1:0]							
rw				rw		rw		rw				rw		rw	

Output compare mode:

Bits	Fields	Descriptions
31	CH1MS[2]	Channel 1 I/O mode selection

		Refer to CH1MS[1:0]description
30	CH0MS[2]	Channel 0 I/O mode selection Refer to CH0MS[1:0] description
29	CH1COMADDSEN	Channel 1 additional compare output shadow enable Refer to CH0COMADDSEN description.
28	CH0COMADDSEN	Channel 0 additional compare output shadow enable When this bit is set, the shadow register of TIMEx_CH0COMV_ADD register which updates at set moments according to the relevant configuration of the shadow register loading function will be enabled. 0: Channel 0 additional compare output shadow disabled 1: Channel 0 additional compare output shadow enabled The PWM mode can be used without validating the shadow register only in single pulse mode (SPM bit in TIMEx_CTL0 register is set). This bit cannot be modified when PROT[1:0] bit-field in TIMEx_CCHP register is 11 and CH0MS bit-field is 000.
27:25	Reserved	Must be kept at reset value.
24	CH1COMCTL[3]	Channel 1 compare output control Refer to CH0COMCTL[2:0] description
23:17	Reserved	Must be kept at reset value.
16	CH0COMCTL[3]	Channel 0 compare output control Refer to CH0COMCTL[2:0] description
15	CH1COMCEN	Channel 1 output compare clear enable Refer to CH0COMCEN description
14:12	CH1COMCTL[2:0]	Channel 1 compare output control Refer to CH0COMCTL[2:0] description
11	CH1COMSEN	Channel 1 output compare shadow enable Refer to CH0COMSEN description
10	CH1COMFEN	Channel 1 output compare fast enable Refer to CH0COMFEN description
9:8	CH1MS[1:0]	Channel 1 mode selection This bit-field specifies the direction of the channel and the input signal selection. The CH1MS[2:0] bit-field is writable only when the channel is not active (When MCH1MSEL[1:0] = 2b'00, the CH1EN bit in TIMEx_CHCTL2 register is reset; when MCH1MSEL[1:0] = 2b'11, the CH1EN and MCH1EN bits in TIMEx_CHCTL2 register are reset). 000: Channel 1 is configured as output. 001: Channel 1 is configured as input, IS1 is connected to CI1FE1.

		010: Channel 1 is configured as input, IS1 is connected to CI0FE1. 011: Channel 1 is configured as input, IS1 is connected to ITS. This mode is working only if an internal trigger input is selected (through TSCFG15[4:0] bit-field in SYSCFG_TIMERxCFG2(x=0,7) register). 100: Channel 1 is configured as input, IS1 is connected to MCI1FE1. 101~111: Reserved.
7	CH0COMCEN	Channel 0 output compare clear enable When this bit is set, the O0CPRE signal is cleared when high level is detected on ETIFP input. 0: Channel 0 output compare clear disabled 1: Channel 0 output compare clear enabled
6:4	CH0COMCTL[2:0]	Channel 0 compare output control The CH0COMCTL[3] and CH0COMCTL[2:0] bit-field control the behavior of O0CPRE which drives CH0_O. The active level of O0CPRE is high, while the active level of CH0_O depends on CH0P bit. Note: When multi mode channel 0 is configured in output mode, and the MCH0MSEL[1:0] = 2b'11, This bit-field controls the behavior of O0CPRE which drives CH0_O and MCH0_O. The active level of O0CPRE is high, while the active level of CH0_O and MCH0_O depends on CH0P and MCH0P bits. 0000: Timing mode. The O0CPRE signal keeps stable, independent of the comparison between the register TIMERx_CH0CV and the counter TIMERx_CNT. 0001: Set the channel output on match. O0CPRE signal is forced high when the counter matches the output compare register TIMERx_CH0CV. 0010: Clear the channel output on match. O0CPRE signal is forced low when the counter matches the output compare register TIMERx_CH0CV. 0011: Toggle on match. O0CPRE toggles when the counter matches the output compare register TIMERx_CH0CV. 0100: Force low. O0CPRE is forced low level. 0101: Force high. O0CPRE is forced high level. 0110: PWM mode 0. When counting up, O0CPRE is active as long as the counter is smaller than TIMERx_CH0CV, otherwise it is inactive. When counting down, O0CPRE is inactive as long as the counter is larger than TIMERx_CH0CV, otherwise it is active. 0111: PWM mode 1. When counting up, O0CPRE is inactive as long as the counter is smaller than TIMERx_CH0CV, otherwise it is active. When counting down, O0CPRE is active as long as the counter is larger than TIMERx_CH0CV, otherwise it is inactive. 1000: Composite PWM mode 0, When counting up, the channel output is inactive when the counter is matched with the value of TIMERx_CH0CV, the channel output is active when the counter matches the value of TIMERx_CH0CV_ADD. When counting down, the channel output is active when the counter is matched with the value of TIMERx_CH0CV, the channel output is inactive when the counter matches

the value of `TIMERx_CH0CV_ADD`.

1001: Composite PWM mode 1. When counting up, the channel output is active when the counter is matched with the value of `TIMERx_CH0CV`, the channel output is inactive when the counter matches the value of `TIMERx_CH0CV_ADD`. When counting down, the channel output is inactive when the counter is matched with the value of `TIMERx_CH0CV`, the channel output is active when the counter matches the value of `TIMERx_CH0CV_ADD`.

1010: Asymmetric PWM mode 0. Only for Center aligned mode, when counting up, the channel output is inactive when the counter is matched with the value of `TIMERx_CH0CV`, when counting down, the channel output is active when the counter is matched with the value of `TIMERx_CH0CV_ADD`.

1011: Asymmetric PWM mode 1. Only for Center aligned mode, when counting up, the channel output is active when the counter is matched with the value of `TIMERx_CH0CV`, when counting down, the channel output is inactive when the counter is matched with the value of `TIMERx_CH0CV_ADD`.

1100~1111: Reserved.

Note: In the composite or asymmetric PWM mode (`CH0COMCTL[3:2]=2'b10` and `CH0MS = 3'b000`), the PWM signal output in channel 0 is composited by `TIMERx_CH0CV` and `TIMERx_CH0COMV_ADD`. Please refer to [Composite PWM mode](#) and [Asymmetric PWM mode](#) for more details.

If configured in PWM mode(PWM mode 0/1、Composite PWM mode0/1、Asymmetric PWM mode0/1), the `O0CPRE` level changes only when the output compare mode switches from “Timing” mode to “PWM” mode or the result of the comparison changes.

When the outputs of `CH0` and `MCH0` are complementary, this bit-field is preloaded. If `CCSE =1`, this bit-field will only be updated when an channel commutation event is generated.

This bit cannot be modified when `PROT[1:0]` bit-field in `TIMERx_CCHP` register is 11 and `CH0MS` bit-field is 000 (compare mode).

3	<code>CH0COMSEN</code>	Channel 0 compare output shadow enable When this bit is set, the shadow register of <code>TIMERx_CH0CV</code> register which updates at set moments according to the relevant configuration of the shadow register loading function will be enabled. 0: Channel 0 output compare shadow disabled 1: Channel 0 output compare shadow enabled The PWM mode can be used without validating the shadow register only in single pulse mode (<code>SPM</code> bit in <code>TIMERx_CTL0</code> register is set). This bit cannot be modified when <code>PROT[1:0]</code> bit-field in <code>TIMERx_CCHP</code> register is 11 and <code>CH0MS</code> bit-field is 000.
2	<code>CH0COMFEN</code>	Channel 0 output compare fast enable 0: Channel 0 output quickly compare disable. Only the valid edge of the comparison result will generate a comparison match, and <code>CH0_O</code> is set to the compare level. The

minimum delay from an edge on the result of the comparison to activate CH0_O output is 5 clock cycles.

1: Channel 0 output quickly compare enable. Both the valid edge of the trigger input signal and the comparison result will generate a comparison match, and CH0_O is set to the compare level. The minimum delay from an edge on the trigger input to activate CH0_O output is 3 clock cycles.

1:0	CH0MS[1:0]	Channel 0 I/O mode selection This bit-field specifies the work mode of the channel and the input signal selection. The CH0MS[2:0] bit-field is writable only when the channel is not active (When MCH0MSEL[1:0] = 2b'00, the CH1EN bit in TIMERx_CHCTL2 register is reset; when MCH0MSEL[1:0] = 2b'11, the CH0EN and MCH0EN bits in TIMERx_CHCTL2 register are reset). 000: Channel 0 is configured as output. 001: Channel 0 is configured as input, IS0 is connected to CI0FE0. 010: Channel 0 is configured as input, IS0 is connected to CI1FE0. 011: Channel 0 is configured as input, IS0 is connected to ITS, this mode is working only if an internal trigger input is selected (through TSCFG15[4:0] bit-field in SYSCFG_TIMERxCFG2(x=0,7) register). 100: Channel 0 is configured as input, IS0 is connected to MCIOFE0. 101~111: Reserved.

Input capture mode:

Bits	Fields	Descriptions
31	CH1MS[2]	Channel 1 I/O mode selection Same as output compare mode.
30	CH0MS[2]	Channel 0 I/O mode selection Same as output compare mode.
29:16	Reserved	Must be kept at reset value.
15:12	CH1CAPFLT[3:0]	Channel 1 input capture filter control Refer to CH0CAPFLT description.
11:10	CH1CAPPSC[1:0]	Channel 1 input capture prescaler Refer to CH0CAPPSC description.
9:8	CH1MS[1:0]	Channel 1 I/O mode selection Same as output compare mode.
7:4	CH0CAPFLT[3:0]	Channel 0 input capture filter control An event counter is used in the digital filter, in which a transition on the output occurs after N input events. This bit-field specifies the frequency used to sample CI0 input signal and the length of the digital filter applied to CI0. 0000: Filter disabled, $f_{SAMP}=f_{DTS}$, $N=1$. 0001: $f_{SAMP}=f_{CK\_TIMER}$, $N=2$.

0010: $f_{SAMP} = f_{CK\_TIMER}$, $N=4$.

0011: $f_{SAMP} = f_{CK\_TIMER}$, $N=8$.

0100: $f_{SAMP} = f_{DTS}/2$, $N=6$.

0101: $f_{SAMP} = f_{DTS}/2$, $N=8$.

0110: $f_{SAMP} = f_{DTS}/4$, $N=6$.

0111: $f_{SAMP} = f_{DTS}/4$, $N=8$.

1000: $f_{SAMP} = f_{DTS}/8$, $N=6$.

1001: $f_{SAMP} = f_{DTS}/8$, $N=8$.

1010: $f_{SAMP} = f_{DTS}/16$, $N=5$.

1011: $f_{SAMP} = f_{DTS}/16$, $N=6$.

1100: $f_{SAMP} = f_{DTS}/16$, $N=8$.

1101: $f_{SAMP} = f_{DTS}/32$, $N=5$.

1110: $f_{SAMP} = f_{DTS}/32$, $N=6$.

1111: $f_{SAMP} = f_{DTS}/32$, $N=8$.

3:2	CH0CAPPSC[1:0]	Channel 0 input capture prescaler This bit-field specifies the factor of the prescaler on channel 0 input. The prescaler is reset when CH0EN bit in TIMEx_CHCTL2 register is cleared. 00: Prescaler disabled, capture is done on each channel input edge. 01: Capture is done every 2 channel input edges. 10: Capture is done every 4 channel input edges. 11: Capture is done every 8 channel input edges.
1:0	CH0MS[1:0]	Channel 0 I/O mode selection Same as output compare mode.

Channel control register 1 (TIMEx_CHCTL1)

Address offset: 0x1C

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CH3MS [2]	CH2MS [2]	CH3COM ADDSEN Reserved	CH2COM ADDSEN Reserved	Reserved			CH3COM CTL[3] Reserved								CH2COM CTL[3] Reserved
rw	rw	rw	rw				rw								rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH3COM CEN	CH3COMCTL[2:0]			CH3COM SEN	CH3COM FEN	CH3MS[1:0]		CH2COM CEN	CH2COMCTL[2:0]			CH2COM SEN	CH2COM FEN	CH2MS[1:0]	
CH3CAPFLT[3:0]				CH3CAPPSC[1:0]				CH2CAPFLT[3:0]				CH2CAPPSC[1:0]			
rw				rw		rw		rw				rw		rw	

Output compare mode:

Bits	Fields	Descriptions
31	CH3MS[2]	Channel 3 I/O mode selection

		Refer to CH3MS[1:0]description.
30	CH2MS[2]	Channel 2 I/O mode selection Refer to CH2MS[1:0] description.
29	CH3COMADDSEN	Channel 3 additional compare output shadow enable Refer to CH2COMADDSEN description.
28	CH2COMADDSEN	Channel 2 additional compare output shadow enable When this bit is set, the shadow register of TIMERx_CH2COMV_ADD register which updates at set moments according to the relevant configuration of the shadow register loading function will be enabled. 0: Channel 2 additional compare shadow disabled 1: Channel 2 additional compare shadow enabled The PWM mode can be used without validating the shadow register only in single pulse mode (SPM bit in TIMERx_CTL0 register is set). This bit cannot be modified when PROT[1:0] bit-field in TIMERx_CCHP register is 11 and CH2MS bit-field is 000.
27:25	Reserved	Must be kept at reset value.
24	CH3COMCTL[3]	Channel 3 compare output control Refer to CH2COMCTL[2:0] description
23:17	Reserved	Must be kept at reset value.
16	CH2COMCTL[3]	Channel 2 compare output control Refer to CH2COMCTL[2:0] description
15	CH3COMCEN	Channel 3 output compare clear enable Refer to CH2COMCEN description
14:12	CH3COMCTL[2:0]	Channel 3 compare output control Refer to CH2COMCTL[2:0] description
11	CH3COMSEN	Channel 3 output compare shadow enable Refer to CH2COMSEN description
10	CH3COMFEN	Channel 3 output compare fast enable Refer to CH0COMFEN description
9:8	CH3MS[1:0]	Channel 3 I/O mode selection This bit-field specifies the direction of the channel and the input signal selection. The CH3MS[2:0] bit-field is writable only when the channel is not active (When MCH3MSEL[1:0] = 2b'00, the CH3EN bit in TIMERx_CHCTL2 register is reset; when MCH3MSEL[1:0] = 2b'11, the CH3EN and MCH3EN bits in TIMERx_CHCTL2 register are reset). 00: Channel 3 is configured as output. 01: Channel 3 is configured as input, IS3 is connected to CI3FE3.

		10: Channel 3 is configured as input, IS3 is connected to CI2FE3.
		11: Channel 3 is configured as input, IS3 is connected to ITS, this mode is working only if an internal trigger input is selected (through TSCFG15[4:0] bit-field in SYSCFG_TIMERxCFG2(x=0,7) register).
		100: Channel 3 is configured as input, IS3 is connected to MCI3FE3.
		101~111: Reserved.
7	CH2COMCEN	Channel 2 output compare clear enable. When this bit is set, the O2CPRE signal is cleared when high level is detected on ETIFP input. 0: Channel 2 output compare clear disabled 1: Channel 2 output compare clear enabled
6:4	CH2COMCTL[2:0]	Channel 2 compare output control Refer to CH0COMCTL[2:0] description.
3	CH2COMSEN	Channel 2 compare output shadow enable Refer to CH0COMSEN description.
2	CH2COMFEN	Channel 2 output compare fast enable Refer to CH0COMFEN description
1:0	CH2MS[1:0]	Channel 2 I/O mode selection This bit-field specifies the work mode of the channel and the input signal selection. The CH2MS[2:0] bit-field is writable only when the channel is not active(When MCH2MSEL[1:0] = 2b'00, the CH2EN bit in TIMERx_CHCTL2 register is reset; when MCH2MSEL[1:0] = 2b'11, the CH2EN and MCH2EN bits in TIMERx_CHCTL2 register are reset). 00: Channel 2 is configured as output. 01: Channel 2 is configured as input, IS2 is connected to CI2FE2. 10: Channel 2 is configured as input, IS2 is connected to CI3FE2. 11: Channel 2 is configured as input, IS2 is connected to ITS. This mode is working only if an internal trigger input is selected (through TSCFG15[4:0] bit-field in SYSCFG_TIMERxCFG2(x=0,7) register). 100: Channel 2 is configured as input, IS2 is connected to MCI2FE2. 101~111: Reserved.

Input capture mode:

Bits	Fields	Descriptions
31	CH3MS[2]	Channel 3 I/O mode selection Same as output compare mode.
30	CH2MS[2]	Channel 2 I/O mode selection Same as output compare mode.
31:16	Reserved	Must be kept at reset value.

15:12	CH3CAPFLT[3:0]	Channel 3 input capture filter control Refer to CH2CAPFLT description.
11:10	CH3CAPPSC[1:0]	Channel 3 input capture prescaler Refer to CH2CAPPSC description.
9:8	CH3MS[1:0]	Channel 3 mode selection Same as output compare mode.
7:4	CH2CAPFLT[3:0]	Channel 2 input capture filter control An event counter is used in the digital filter, in which a transition on the output occurs after N input events. This bit-field specifies the frequency used to sample CI2 input signal and the length of the digital filter applied to CI2. 0000: Filter disabled, $f_{SAMP}=f_{DTS}$, $N=1$. 0001: $f_{SAMP}=f_{CK\_TIMER}$, $N=2$. 0010: $f_{SAMP}=f_{CK\_TIMER}$, $N=4$. 0011: $f_{SAMP}=f_{CK\_TIMER}$, $N=8$. 0100: $f_{SAMP}=f_{DTS}/2$, $N=6$. 0101: $f_{SAMP}=f_{DTS}/2$, $N=8$. 0110: $f_{SAMP}=f_{DTS}/4$, $N=6$. 0111: $f_{SAMP}=f_{DTS}/4$, $N=8$. 1000: $f_{SAMP}=f_{DTS}/8$, $N=6$. 1001: $f_{SAMP}=f_{DTS}/8$, $N=8$. 1010: $f_{SAMP}=f_{DTS}/16$, $N=5$. 1011: $f_{SAMP}=f_{DTS}/16$, $N=6$. 1100: $f_{SAMP}=f_{DTS}/16$, $N=8$. 1101: $f_{SAMP}=f_{DTS}/32$, $N=5$. 1110: $f_{SAMP}=f_{DTS}/32$, $N=6$. 1111: $f_{SAMP}=f_{DTS}/32$, $N=8$.
3:2	CH2CAPPSC[1:0]	Channel 2 input capture prescaler This bit-field specifies the factor of the prescaler on channel 2 input. The prescaler is reset when CH2EN bit in TIMEx_CHCTL2 register is cleared. 00: Prescaler disabled, capture is done on each channel input edge. 01: Capture is done every 2 channel input edges. 10: Capture is done every 4 channel input edges. 11: Capture is done every 8 channel input edges.
1:0	CH2MS[1:0]	Channel 2 mode selection Same as output compare mode.

Channel control register 2 (TIMEx_CHCTL2)

Address offset: 0x20

Reset value: 0x0000 0000

This register can be accessed by half-word(16-bit) or word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CH3PER FOREN	CH2PERF OREN	CH1PERF OREN	CH0PERF OREN	Reserved											
rw	rw	rw	rw												
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MCH3P	MCH3EN	CH3P	CH3EN	MCH2P	MCH2EN	CH2P	CH2EN	MCH1P	MCH1EN	CH1P	CH1EN	MCH0P	MCH0EN	CH0P	CH0EN
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31	CH3PERFOREN	Channel 3 in the composite PWM or asymmetric PWM mode, the period point match moment O3CPRE level setting Refer to CH0PERFOREN description.
30	CH2PERFOREN	Channel 2 in the composite PWM or asymmetric PWM mode, the period point match moment O2CPRE level setting Refer to CH0PERFOREN description.
29	CH1PERFOREN	Channel 1 in the composite PWM or asymmetric PWM mode, the period point match moment O1CPRE level setting Refer to CH0PERFOREN description.
28	CH0PERFOREN	Channel 0 in the composite PWM or asymmetric PWM mode, the period point match moment O0CPRE level setting 0: disable 1: In the composite PWM0 or asymmetric PWM0 mode, the level will be forced to high at the time of period point matching; In the composite PWM1 or asymmetric PWM1 mode, the level at the matching time of the period point is forced to low. Note: In the central alignment mode, the period point is the counter underflow time; In the edge-aligned mode, the period point is the counter flow moment.
27:16	Reserved	Must be kept at reset value.
15	MCH3P	Multi mode channel 3 capture/compare polarity Refer to MCH0P description.
14	MCH3EN	Multi mode channel 3 capture/compare enable Refer to MCH0EN description.
13	CH3P	Channel 3 capture/compare polarity Refer to CH0P description.
12	CH3EN	Channel 3 capture/compare enable Refer to CH0EN description.
11	MCH2P	Multi mode channel 2 output polarity Refer to MCH0P description.
10	MCH2EN	Multi mode channel 2 output enable

		Refer to MCH0EN description.
9	CH2P	Channel 2 capture/compare polarity Refer to CH0P description.
8	CH2EN	Channel 2 capture/compare enable Refer to CH0EN description.
7	MCH1P	Multi mode channel 1 output polarity Refer to MCH0P description.
6	MCH1EN	Multi mode channel 1 output enable Refer to MCH0EN description.
5	CH1P	Channel 1 capture/compare polarity Refer to CH0P description.
4	CH1EN	Channel 1 capture/compare enable Refer to CH0EN description.
3	MCH0P	Multi mode channel 0 output polarity When Multi mode channel 0 is configured in output mode, and the MCH0MSEL[1:0] = 2b'11, this bit specifies the MCH0_O output signal polarity. 0: Multi mode channel 0 output active high 1: Multi mode channel 0 output active low When CH0 is configured in input mode, in conjunction with CH0P, this bit is used to define the polarity of CH0. This bit cannot be modified when PROT[1:0] bit-field in TIMEx_CCHP register is 11 or 10.
2	MCH0EN	Multi mode channel 0 capture/compare enable When multi mode channel 0 is configured in output mode, setting this bit enables MCH0_O signal in active state. When multi mode channel 0 is configured in input mode, setting this bit enables the capture event in multi mode channel 0. 0: Multi mode channel 0 disabled 1: Multi mode channel 0 enabled
1	CH0P	Channel 0 capture/compare polarity When channel 0 is configured in output mode, this bit specifies the output signal polarity. 0: Channel 0 active high 1: Channel 0 active low When channel 0 is configured in input mode, these bits specific the channel 0 input signal's polarity. [MCH0P, CH0P] will select the active trigger or capture polarity for channel 0 input signals. 00: channel 0 input signal's rising edge is the active signal for capture or trigger operation in slave mode. And channel 0 input signal will not be inverted. 01: channel 0 input signal's falling edge is the active signal for capture or trigger

operation in slave mode. And channel 0 input signal will be inverted.

10: Reserved.

11: Noninverted / both channel 0 input signal's edges.

This bit cannot be modified when PROT[1:0] bit-field in TIMEx_CCHP register is 11 or 10.

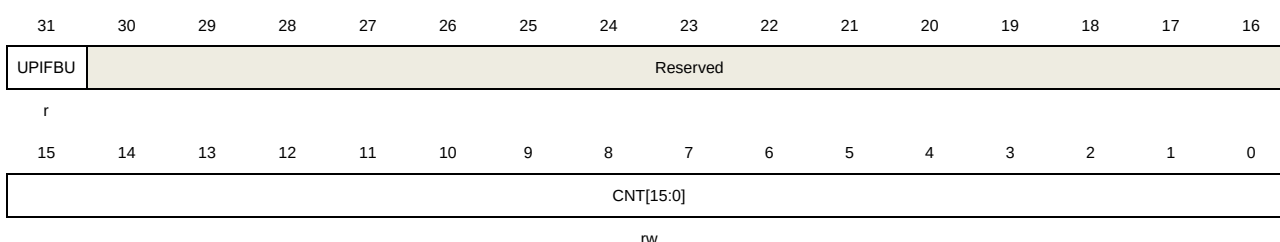
0	CH0EN	Channel 0 capture/compare enable
		When channel 0 is configured in output mode, setting this bit enables CH0_O signal in active state. When channel 0 is configured in input mode, setting this bit enables the capture event in channel 0.
		0: Channel 0 disabled
		1: Channel 0 enabled

Counter register (TIMEx_CNT)

Address offset: 0x24

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



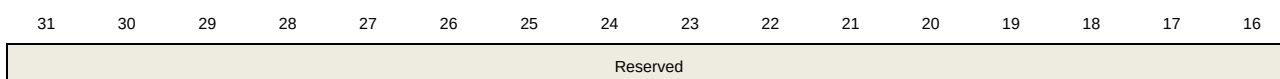
Bits	Fields	Descriptions
31	UPIFBU	UPIF bit backup This bit is a backup of UPIF bit in TIMEx_INTF register and read-only. This bit is only valid when UPIFBUEN = 1. If the UPIFBUEN = 0, this bit is reserved and read the result is 0.
30:16	Reserved	Must be kept at reset value.
15:0	CNT[15:0]	This bit-field indicates the current counter value. Writing to this bit-field can change the value of the counter.

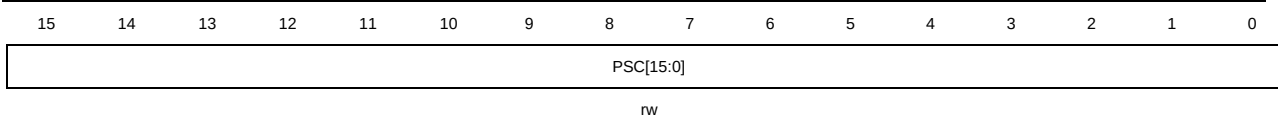
Prescaler register (TIMEx_PSC)

Address offset: 0x28

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).





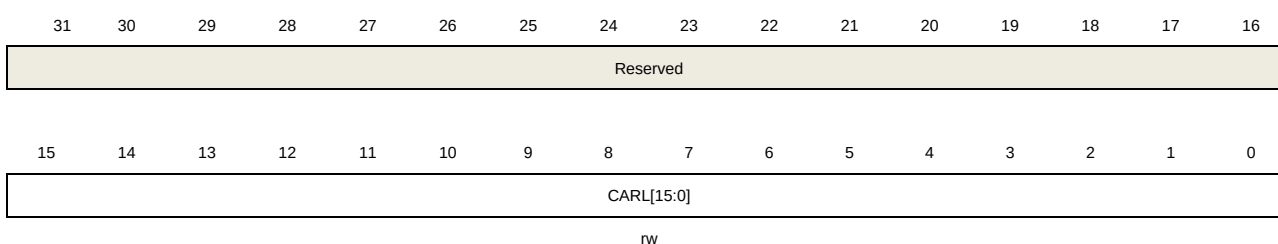
Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	PSC[15:0]	Prescaler value of the counter clock The PSC clock is divided by (PSC+1) to generate the counter clock. The value of this bit-field will be loaded to the corresponding shadow register at every update event.

Counter auto reload register (TIMERx_CAR)

Address offset: 0x2C

Reset value: 0x0000 FFFF

This register has to be accessed by word (32-bit).



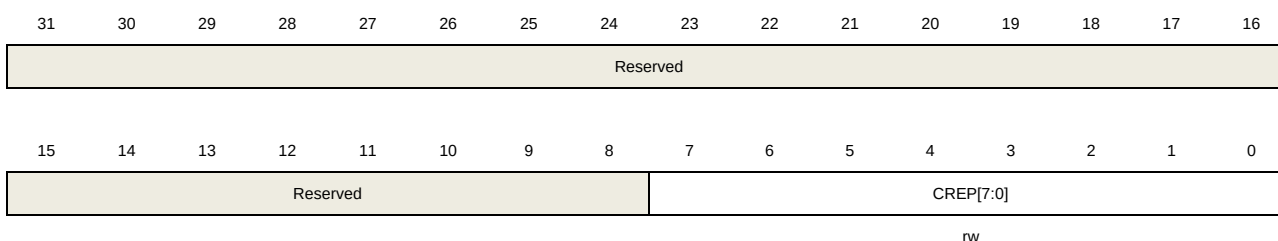
Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CARL[15:0]	Counter auto reload value This bit-field specifies the auto reload value of the counter.

Counter repetition register (TIMERx_CREP)

Address offset: 0x30

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



Bits	Fields	Descriptions

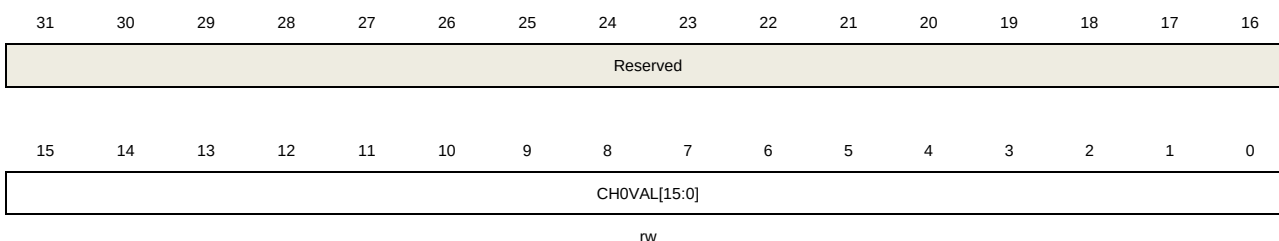
31:8	Reserved	Must be kept at reset value.
7:0	CREP[7:0]	Counter repetition value This bit-field specifies the update event generation rate. Each time the repetition counter counts down to zero, an update event will be generated. The update rate of the shadow registers is also affected by this bit-field when these shadow registers are enabled.

Channel 0 capture/compare value register (TIMERx_CH0CV)

Address offset: 0x34

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



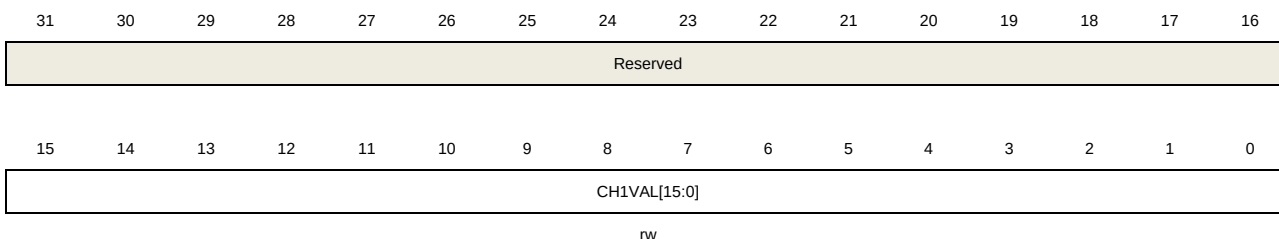
Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CH0VAL[15:0]	Capture/compare value of channel 0 When channel 0 is configured in input mode, these bit-field indicates the counter value at the last capture event. And this bit-field is read-only. When channel 0 is configured in output mode, this bit-field contains value to be compared to the counter. When the corresponding shadow register is enabled, the shadow register updates at the set moment according to the relevant configuration of the shadow register loading function.

Channel 1 capture/compare value register (TIMERx_CH1CV)

Address offset: 0x38

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



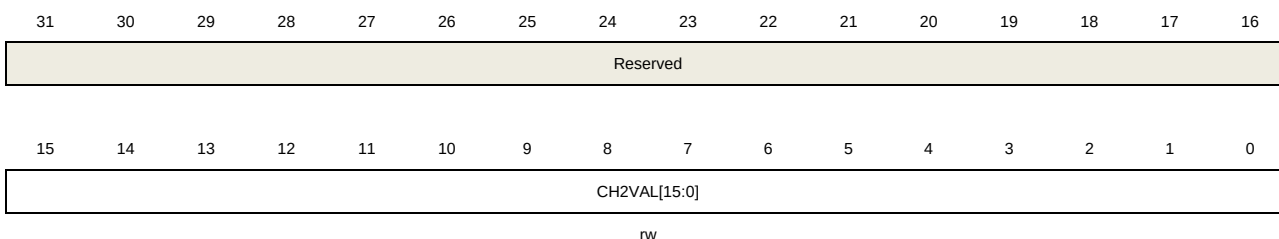
Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CH1VAL[15:0]	Capture/compare value of channel 1 When channel 1 is configured in input mode, these bit-field indicates the counter value at the last capture event. And this bit-field is read-only. When channel 1 is configured in output mode, this bit-field contains value to be compared to the counter. When the corresponding shadow register is enabled, the shadow register updates at the set moment according to the relevant configuration of the shadow register loading function.

Channel 2 capture/compare value register (TIMERx_CH2CV)

Address offset: 0x3C

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



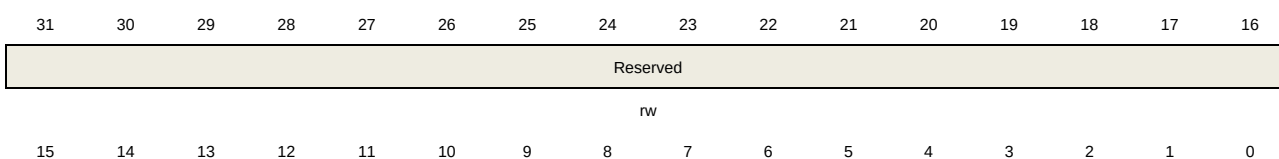
Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CH2VAL[15:0]	Capture/compare value of channel 2 When channel 2 is configured in input mode, these bit-field indicates the counter value at the last capture event. And this bit-field is read-only. When channel 2 is configured in output mode, this bit-field contains value to be compared to the counter. When the corresponding shadow register is enabled, the shadow register updates at the set moment according to the relevant configuration of the shadow register loading function.

Channel 3 capture/compare value register (TIMERx_CH3CV)

Address offset: 0x40

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



CH3VAL[15:0]

rw

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CH3VAL[15:0]	Capture/compare value of channel 3 When channel 3 is configured in input mode, these bit-field indicates the counter value at the last capture event. And this bit-field is read-only. When channel 3 is configured in output mode, this bit-field contains value to be compared to the counter. When the corresponding shadow register is enabled, the shadow register updates at the set moment according to the relevant configuration of the shadow register loading function.

Complementary channel protection register (TIMERx_CCHP)

Address offset: 0x44

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved												BRKF[3:0]			
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
POEN	OAEN	BRKP	BRKEN	ROS	IOS	PROT[1:0]		DTCFG[7:0]							
rw	rw	rw	rw	rw	rw	rw		rw							

Bits	Fields	Descriptions
31:20	Reserved	Must be kept at reset value.
19:16	BRKF[3:0]	BREAK input signal filter An event counter is used in the digital filter, in which a transition on the output occurs after N input events. This bit-field specifies the frequency used to sample BREAK input signal and the length of the digital filter applied to BREAK. 0000: Filter disabled. BREAK act asynchronously, N=1 0001: $f_{SAMP} = f_{CK\_TIMER}$, N=2 0010: $f_{SAMP} = f_{CK\_TIMER}$, N=4 0011: $f_{SAMP} = f_{CK\_TIMER}$, N=8 0100: $f_{SAMP} = f_{DTS}/2$, N=6 0101: $f_{SAMP} = f_{DTS}/2$, N=8 0110: $f_{SAMP} = f_{DTS}/4$, N=6 0111: $f_{SAMP} = f_{DTS}/4$, N=8 1000: $f_{SAMP} = f_{DTS}/8$, N=6 1001: $f_{SAMP} = f_{DTS}/8$, N=8 1010: $f_{SAMP} = f_{DTS}/16$, N=5

		1011: $f_{SAMP} = f_{DTS}/16$, $N=6$ 1100: $f_{SAMP} = f_{DTS}/16$, $N=8$ 1101: $f_{SAMP} = f_{DTS}/32$, $N=5$ 1110: $f_{SAMP} = f_{DTS}/32$, $N=6$ 1111: $f_{SAMP} = f_{DTS}/32$, $N=8$ This bit can be modified only when PROT[1:0] bit-field in TIMERx_CCHP register is 00.
15	POEN	Primary output enable This bit is set by software or automatically set by hardware depending on the OAEN bit. It is cleared asynchronously by hardware as soon as the break input is active. When one of channels is configured in output mode, setting this bit enables the channel outputs (CHx_O and MCHx_O) if the corresponding enable bits (CHxEN, MCHxEN in TIMERx_CHCTL2 register) have been set. 0: Channel outputs are disabled or forced to idle state. 1: Channel outputs are enabled.
14	OAEN	Output automatic enable This bit specifies whether the POEN bit can be set automatically by hardware. 0: POEN cannot be set by hardware. 1: Depending on the automatic output function mode selection (BRKAUMS), POEN can be set by hardware automatically at corresponding counter overflow or underflow moment, if the break input is not active. This bit can be modified only when PROT[1:0] bit-field in TIMERx_CCHP register is 00.
13	BRKP	BREAK input signal polarity This bit specifies the polarity of the BREAK input signal. 0: BREAK input active low 1: BREAK input active high This bit can be modified only when PROT[1:0] bit-field in TIMERx_CCHP register is 00.
12	BRKEN	BREAK input signal enable This bit can be set to enable the BREAK input signal 0: BREAK input disabled 1: BREAK input enabled This bit can be modified only when PROT[1:0] bit-field in TIMERx_CCHP register is 00.
11	ROS	Run mode “off-state” enable When POEN bit is set (Run mode), this bit can be set to enable the “off-state” for the channels which has been configured in output mode. Please refer to Table 21-5. Complementary outputs controlled by parameters (MCHxMSEL =2'b11). 0: “off-state” disabled. If the CHxEN or CHxNEN bit is reset, the corresponding

		channel is output disabled. 1: "off-state" enabled. If the CHxEN or CHxNEN bit is reset, the corresponding channel is "off-state". This bit cannot be modified when PROT [1:0] bit-field in TIMERx_CCHP register is 10 or 11.
10	IOS	Idle mode "off-state" enable When POEN bit is reset (Idle mode), this bit can be set to enable the "off-state" for the channels which has been configured in output mode. Please refer to Table 21-5. Complementary outputs controlled by parameters (MCHxMSEL = 2'b11) . 0: "off-state" disabled. If the CHxEN/CHxNEN bits are both reset, the channels are output disabled. 1: "off-state" enabled. No matter the CHxEN / CHxNEN bits, the channels are "off-state". This bit cannot be modified when PROT [1:0] bit-field in TIMERx_CCHP register is 10 or 11.
9:8	PROT[1:0]	Complementary register protect control This bit-field specifies the write protection property of registers. 00: Protect disabled. No write protection. 01: PROT mode 0. The ISOx/ISOxN bits in TIMERx_CTL1 register, the BRKEN/BRKP/OAEN/DTCFG bits in TIMERx_CCHP register are writing protected. 10: PROT mode 1. In addition to the registers in PROT mode 0, the CHxP/MCHxP bits in TIMERx_CHCTL2 register (if related channel is configured in output mode) , the ROS/IOS bits in TIMERx_CCHP register are writing protected. 11: PROT mode 2. In addition to the registers in PROT mode 1, the CHxCOMCTL/ CHxCOMSEN/ CHxCOMADDSEN/ MCHxCOMCTL/ MCHxCOMSEN bits in TIMERx_CHCTL0/1 and TIMERx_MCHCTL0/1 registers (if the related channel is configured in output) are writing protected. This bit-field can be written only once after the system reset. Once the TIMERx_CCHP register has been written, this bit-field will be writing protected.
7:0	DTCFG[7:0]	Dead time configuration This bit-field controls the value of the dead-time, which is inserted before the output transitions. The relationship between the value of DTCFG and the duration of dead-time is as follow: $DTCFG[7:5] = 3'b0xx: DT\ value = DTCFG[7:0] * t_{DT}, t_{DT} = t_{DTS}$ $DTCFG[7:5] = 3'b10x: DT\ value = (64 + DTCFG[5:0]) * t_{DT}, t_{DT} = t_{DTS} * 2$ $DTCFG[7:5] = 3'b110: DT\ value = (32 + DTCFG[4:0]) * t_{DT}, t_{DT} = t_{DTS} * 8$ $DTCFG[7:5] = 3'b111: DT\ value = (32 + DTCFG[4:0]) * t_{DT}, t_{DT} = t_{DTS} * 16$ This bit can be modified only when PROT[1:0] bit-field in TIMERx_CCHP register is 00.

Multi mode channel control register 0 (TIMERx_MCHCTL0)

Address offset: 0x48

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MCH1 MS[2]	MCH0 MS[2]	Reserved													
rw		rw													
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MCH1CO MCEN	MCH1COMCTL[2:0]		MCH1CO MSEN	MCH1CO MFEN	MCH1MS[1:0]		MCH0CO MCEN	MCH0COMCTL[2:0]		MCH0CO MSEN	MCH0CO MFEN	MCH0MS[1:0]			
MCH1CAPFLT[3:0]			MCH1CAPPSC [1:0]				MCH0CAPFLT[3:0]			MCH0CAPPSC [1:0]					
rw		rw		rw		rw				rw		rw			

Output compare mode:

Bits	Fields	Descriptions
31	MCH1MS[2]	Multi mode channel 1 I/O mode selection Refer to MCH1MS[1:0]description.
30	MCH0MS[2]	Multi mode channel 0 I/O mode selection Refer to MCH0MS[1:0] description.
29:16	Reserved	Must be kept at reset value.
15	MCH1COMCEN	Multi mode channel 1 output compare clear enable. Refer to MCH0COMCEN description.
14:12	MCH1COMCTL [2:0]	Multi mode channel 1 compare output control. Refer to MCH0COMCTL description.
11	MCH1COMSEN	Multi mode channel 1 output compare shadow enable Refer to MCH0COMSEN description.
10	MCH1COMFEN	Multi mode channel 1 output compare fast enable Refer to MCH0COMFEN description
9:8	MCH1MS[1:0]	Multi mode channel 1 I/O mode selection This bit-field specifies the direction of the channel and the input signal selection. This bit-field is writable only when the channel is not active (the MCH1EN bit in TIMERx_CHCTL2 register is reset) 000: Multi mode channel 1 is configured as output. 001: Multi mode channel 1 is configured as input, MIS1 is connected to MCI1FEM1. 010: Multi mode channel 1 is configured as input, MIS1 is connected to MCI0FEM1. 011: Multi mode channel 1 is configured as input, MIS1 is connected to ITS. This mode is working only if an internal trigger input is selected (through TSCFG15[4:0] bit-field in SYSCFG_TIMERxCFG2(x=0,7) register).

		100: Multi mode channel 1 is configured as input, MIS1 is connected to CI1FEM1. 101~111: Reserved.
7	MCH0COMCEN	Multi mode channel 0 output compare clear enable. When this bit is set, the MO0CPRE signal is cleared when high level is detected on ETIFP input. 0: Multi mode channel 0 output compare clear disabled. 1: Multi mode channel 0 output compare clear enabled.
6:4	MCH0COMCTL [2:0]	Multi mode channel 0 output compare control When multi mode channel 0 is configured in output mode, and the MCH0MSEL[1:0] = 2b'00, the bit-field control the behavior of MO0CPRE which drives MCH0_O. The active level of MO0CPRE is high, while the active level of MCH0_O depends on MCH0FP[1:0] bits. Note: When multi mode channel 0 is configured in output mode, and the MCH0MSEL[1:0] = 2b'11, the CH0COMCTL[2:0] and CH0COMCTL[2:0] bit-field controls the behavior of O0CPRE which drives CH0_O and MCH0_O, while the active level of CH0_O and MCH0_O depends on CH0P and MCH0P bits. 000: Timing mode. The MO0CPRE signal keeps stable, independent of the comparison between the register TIMERx_MCH0CV and the counter TIMERx_CNT. 001: Set the channel output on match. MO0CPRE signal is forced high when the counter matches the output compare register TIMERx_MCH0CV. 010: Clear the channel output on match. MO0CPRE signal is forced low when the counter matches the output compare register TIMERx_MCH0CV. 011: Toggle on match. MO0CPRE toggles when the counter matches the output compare register TIMERx_MCH0CV. 100: Force low. MO0CPRE is forced low level. 101: Force high. MO0CPRE is forced high level. 110: PWM mode 0. When counting up, MO0CPRE is active as long as the counter is smaller than TIMERx_MCH0CV, otherwise it is inactive. When counting down, MO0CPRE is inactive as long as the counter is larger than TIMERx_MCH0CV, otherwise it is active. 111: PWM mode 1. When counting up, MO0CPRE is inactive as long as the counter is smaller than TIMERx_MCH0CV, otherwise it is active. When counting down, MO0CPRE is active as long as the counter is larger than TIMERx_MCH0CV, otherwise it is inactive. If configured in PWM mode, the MO0CPRE level changes only when the output compare mode switches from "Timing" mode to "PWM" mode or the result of the comparison changes. When the outputs of CH0 and MCH0 are complementary, this bit-field is preloaded. If CCSE = 1, this bit-field will only be updated when an channel commutation event is generated. This bit cannot be modified when PROT[1:0] bit-field in TIMERx_CCHP register is

		11 and MCH0MS bit-field is 00(compare mode).
3	MCH0COMSEN	Multi mode channel 0 output compare shadow enable When this bit is set, the shadow register of TIMEx_MCH0CV register which updates at set moments according to the relevant configuration of the shadow register loading function will be enabled. 0: Multi mode channel 0 output compare shadow disabled 1: Multi mode channel 0 output compare shadow enabled The PWM mode can be used without validating the shadow register only in single pulse mode (SPM bit in TIMEx_CTL0 register is set). This bit cannot be modified when PROT[1:0] bit-field in TIMEx_CCHP register is 11 and MCH0MS bit-field is 000.
2	MCH0COMFEN	Multi mode channel 0 output compare fast enable 0: Multi mode channel 0 output quickly compare disable.Only the valid edge of the comparison result will generate a comparison match, and MCH0_O is set to the compare level.The minimum delay from an edge on the result of the comparison to activate MCH0_O output is 5 clock cycles. 1: Multi mode channel 0 output quickly compare enable.Both the valid edge of the trigger input signal and the comparison result will generate a comparison match, and MCH0_O is set to the compare level.The minimum delay from an edge on the trigger input to activate MCH0_O output is 3 clock cycles.
1:0	MCH0MS[1:0]	Multi mode channel 0 I/O mode selection This bit-field specifies the work mode of the channel and the input signal selection. This bit-field is writable only when the channel is not active (MCH0EN bit in TIMEx_CHCTL2 register is reset). 000: Multi mode channel 0 is configured as output. 001: Multi mode channel 0 is configured as input, MIS0 is connected to MCIOFEM0. 010: Multi mode channel 0 is configured as input, MIS0 is connected to MCII1FEM0. 011: Multi mode channel 0 is configured as input, MIS0 is connected to ITS, this mode is working only if an internal trigger input is selected (through TSCFG15[4:0] bit-field in SYSCFG_TIMExCFG2(x=0,7) register). 100: Multi mode channel 0 is configured as input, MIS0 is connected to CIOFEM0. 101~111: Reserved.

Input capture mode:

Bits	Fields	Descriptions
31	MCH1MS[2]	Multi mode channel 1 I/O mode selection Refer to MCH1MS[1:0]description.
30	MCH0MS[2]	Multi mode channel 0 I/O mode selection Refer to MCH0MS[1:0] description.
29:16	Reserved	Must be kept at reset value.

15:12	MCH1CAPFLT[3:0]	Multi mode channel 1 input capture filter control. Refer to MCH0CAPFLT description.
11:10	MCH1CAPPSC[1:0]	Multi mode channel 1 input capture prescaler. Refer to MCH0CAPPSC description.
9:8	MCH1MS[1:0]	Multi mode channel 1 I/O mode selection. Same as output compare mode.
7:4	MCH0CAPFLT[3:0]	Multi mode channel 0 input capture filter control. An event counter is used in the digital filter, in which a transition on the output occurs after N input events. This bit-field specifies the frequency used to sample MCI0 input signal and the length of the digital filter applied to MCI0. 0000: Filter disabled, $f_{SAMP}=f_{DTS}$, $N=1$. 0001: $f_{SAMP}=f_{CK\_TIMER}$, $N=2$. 0010: $f_{SAMP}=f_{CK\_TIMER}$, $N=4$. 0011: $f_{SAMP}=f_{CK\_TIMER}$, $N=8$. 0100: $f_{SAMP}=f_{DTS}/2$, $N=6$. 0101: $f_{SAMP}=f_{DTS}/2$, $N=8$. 0110: $f_{SAMP}=f_{DTS}/4$, $N=6$. 0111: $f_{SAMP}=f_{DTS}/4$, $N=8$. 1000: $f_{SAMP}=f_{DTS}/8$, $N=6$. 1001: $f_{SAMP}=f_{DTS}/8$, $N=8$. 1010: $f_{SAMP}=f_{DTS}/16$, $N=5$. 1011: $f_{SAMP}=f_{DTS}/16$, $N=6$. 1100: $f_{SAMP}=f_{DTS}/16$, $N=8$. 1101: $f_{SAMP}=f_{DTS}/32$, $N=5$. 1110: $f_{SAMP}=f_{DTS}/32$, $N=6$. 1111: $f_{SAMP}=f_{DTS}/32$, $N=8$.
3:2	MCH0CAPPSC[1:0]	Multi mode channel 0 input capture prescaler This bit-field specifies the factor of the prescaler on channel 0 input. The prescaler is reset when MCH0EN bit in TIMEx_CHCTL2 register is cleared. 00: Prescaler disabled, capture is done on each channel input edge. 01: Capture is done every 2 channel input edges. 10: Capture is done every 4 channel input edges. 11: Capture is done every 8 channel input edges.
1:0	MCH0MS[1:0]	Multi mode channel 0 I/O mode selection Same as output compare mode

Multi mode channel control register 1 (TIMEx_MCHCTL1)

Address offset: 0x4C

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MCH3MS [2]	MCH2MS [2]	Reserved													
rw	rw	rw													
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MCH3CO MCEN	MCH3COMCTL[2:0]			MCH3CO MSEN	MCH3CO MFEN	MCH3MS[1:0]		MCH2CO MCEN	MCH2COMCTL[2:0]			MCH2CO MSEN	MCH2CO MFEN	MCH2MS[1:0]	
MCH3CAPFLT[3:0]				MCH3CAPPSC[1:0]				MCH2CAPFLT[3:0]			MCH2CAPPSC[1:0]				
rw		rw			rw			rw			rw			rw	

Output compare mode:

Bits	Fields	Descriptions
31	MCH3MS[2]	Multi mode channel 1 I/O mode selection Refer to MCH3MS[1:0]description.
30	MCH2MS[2]	Multi mode channel 0 I/O mode selection Refer to MCH2MS[1:0] description.
29:16	Reserved	Must be kept at reset value.
15	MCH3COMCEN	Multi mode channel 3 output compare clear enable Refer to MCH2COMCEN description
14:12	MCH3COMCTL[2:0]	Multi mode channel 3 compare output control Refer to MCH2COMCTL description
11	MCH3COMSEN	Multi mode channel 3 output compare shadow enable Refer to MCH2COMSEN description
10	MCH3COMFEN	Multi mode channel 3 output compare fast enable Refer to MCH2COMFEN description
9:8	MCH3MS[1:0]	Multi mode channel 3 I/O mode selection This bit-field specifies the direction of the channel and the input signal selection. This bit-field is writable only when the channel is not active (MCH3EN bit in TIMEx_CHCTL2 register is reset). 000: Multi mode channel 3 is configured as output. 01: Multi mode channel 3 is configured as input, MIS3 is connected to MCI3FEM3. 010: Multi mode channel 3 is configured as input, MIS3 is connected to MCI2FEM3. 011: Multi mode channel 3 is configured as input, MIS3 is connected to ITS, this mode is working only if an internal trigger input is selected (through TSCFG15[4:0] bit-field in SYSCFG_TIMExCFG2(x=0,7) register). 100: Multi mode channel 3 is configured as input, MIS3 is connected to CI3FEM3. 101~111: Reserved.
7	MCH2COMCEN	Multi mode channel 2 output compare clear enable. When this bit is set, the MO2CPRE signal is cleared when high level is detected on ETIFP input.

		0: Multi mode channel 2 output compare clear disabled 1: Multi mode channel 2 output compare clear enabled
6:4	MCH2COMCTL[2:0]	Multi mode channel 2 compare output control When multi mode channel 2 is configured in output mode, and the MCH2MSEL[1:0] = 2b'00, this bit-field control the behavior of MO2CPRE which drives MCH2_O. The active level of MO2CPRE is high, while the active level of MCH2_O depends on MCH2FP[1:0] bits. Note: When multi mode channel 2 is configured in output mode, and the MCH2MSEL[1:0] = 2b'11, the CH2COMCTL[2:0] bit-field controls the behavior of O2CPRE which drives CH2_O and MCH2_O, while the active level of CH2_O and MCH2_O depends on CH2P and MCH2P bits. 000: Timing mode. The MO2CPRE signal keeps stable, independent of the comparison between the output compare register TIMERx_MCH2CV and the counter TIMERx_CNT. 001: Set the channel output on match. MO2CPRE signal is forced high when the counter matches the output compare register TIMERx_MCH2CV. 010: Clear the channel output on match. MO2CPRE signal is forced low when the counter matches the output compare register TIMERx_MCH2CV. 011: Toggle on match. MO2CPRE toggles when the counter matches the output compare register TIMERx_MCH2CV. 100: Force low. MO2CPRE is forced low level. 101: Force high. MO2CPRE is forced high level. 110: PWM mode 0. When counting up, MO2CPRE is active as long as the counter is smaller than TIMERx_MCH2CV, otherwise it is inactive. When counting down, MO2CPRE is inactive as long as the counter is larger than TIMERx_MCH2CV, otherwise it is active. 111: PWM mode 1. When counting up, MO2CPRE is inactive as long as the counter is smaller than TIMERx_MCH2CV, otherwise it is active. When counting down, MO2CPRE is active as long as the counter is larger than TIMERx_MCH2CV, otherwise it is inactive. If configured in PWM mode, the MO2CPRE level changes only when the output compare mode switches from "Timing" mode to "PWM" mode or the result of the comparison changes. When the outputs of CH2 and MCH2 are complementary, this bit-field is preloaded. If CCSE =1, this bit-field will only be updated when an channel commutation event is generated. This bit cannot be modified when PROT[1:0] bit-field in TIMERx_CCHP register is 11 and MCH2MS bit-field is 00(compare mode).
3	MCH2COMSEN	Multi mode channel 2 output compare shadow enable When this bit is set, the shadow register of TIMERx_MCH0CV register which updates at set moments according to the relevant configuration of the shadow register loading function will be enabled.

		0: Multi mode channel 2 output compare shadow disabled 1: Multi mode channel 2 output compare shadow enabled The PWM mode can be used without validating the shadow register only in single pulse mode (SPM bit in TIMERx_CTL0 register is set). This bit cannot be modified when PROT[1:0] bit-field in TIMERx_CCHP register is 11 and MCH2MS bit-field is 00.
2	MCH2COMFEN	Multi mode channel 2 output compare fast enable 0: Multi mode channel 2 output quickly compare disable. Only the valid edge of the comparison result will generate a comparison match, and MCH2_O is set to the compare level. The minimum delay from an edge on the result of the comparison to activate MCH2_O output is 5 clock cycles. 1: Multi mode channel 2 output quickly compare enable. Both the valid edge of the trigger input signal and the comparison result will generate a comparison match, and MCH2_O is set to the compare level. The minimum delay from an edge on the trigger input to activate MCH2_O output is 3 clock cycles.
1:0	MCH2MS[1:0]	Multi mode channel 2 I/O mode selection This bit-field specifies the work mode of the channel and the input signal selection. This bit-field is writable only when the channel is not active (MCH2EN bit in TIMERx_CHCTL2 register is reset). 000: Multi mode channel 2 is configured as output. 001: Multi mode channel 2 is configured as input, MIS2 is connected to MCI2FEM2. 010: Multi mode channel 2 is configured as input, MIS2 is connected to MCI3FEM2. 011: Multi mode channel 2 is configured as input, MIS2 is connected to ITS. This mode is working only if an internal trigger input is selected (through TSCFG15[4:0] bit-field in SYSCFG_TIMERxCFG2(x=0,7) register). 100: Multi mode channel 2 is configured as input, MIS2 is connected to CI2FEM2. 101~111: Reserved.

Input capture mode:

Bits	Fields	Descriptions
31	MCH3MS[2]	Multi mode channel 1 I/O mode selection Refer to MCH3MS[1:0] description.
30	MCH2MS[2]	Multi mode channel 0 I/O mode selection Refer to MCH2MS[1:0] description.
29:16	Reserved	Must be kept at reset value.
15:12	MCH3CAPFLT[3:0]	Multi mode channel 3 input capture filter control. Refer to MCH2CAPFLT description.
11:10	MCH3CAPPSC[1:0]	Multi mode channel 3 input capture prescaler. Refer to MCH2CAPPSC description.
9:8	MCH3MS[1:0]	Multi mode channel 3 I/O mode selection.

Same as output compare mode.

7:4	MCH2CAPFLT[3:0]	Multi mode channel 2 input capture filter control. An event counter is used in the digital filter, in which a transition on the output occurs after N input events. This bit-field specifies the frequency used to sample MCI2 input signal and the length of the digital filter applied to MCI2. 0000: Filter disabled, $f_{SAMP}=f_{DTS}$, $N=1$. 0001: $f_{SAMP}=f_{CK\_TIMER}$, $N=2$. 0010: $f_{SAMP}=f_{CK\_TIMER}$, $N=4$. 0011: $f_{SAMP}=f_{CK\_TIMER}$, $N=8$. 0100: $f_{SAMP}=f_{DTS}/2$, $N=6$. 0101: $f_{SAMP}=f_{DTS}/2$, $N=8$. 0110: $f_{SAMP}=f_{DTS}/4$, $N=6$. 0111: $f_{SAMP}=f_{DTS}/4$, $N=8$. 1000: $f_{SAMP}=f_{DTS}/8$, $N=6$. 1001: $f_{SAMP}=f_{DTS}/8$, $N=8$. 1010: $f_{SAMP}=f_{DTS}/16$, $N=5$. 1011: $f_{SAMP}=f_{DTS}/16$, $N=6$. 1100: $f_{SAMP}=f_{DTS}/16$, $N=8$. 1101: $f_{SAMP}=f_{DTS}/32$, $N=5$. 1110: $f_{SAMP}=f_{DTS}/32$, $N=6$. 1111: $f_{SAMP}=f_{DTS}/32$, $N=8$.
3:2	MCH2CAPPSC[1:0]	Multi mode channel 2 input capture prescaler. This bit-field specifies the factor of the prescaler on channel 2 input. The prescaler is reset when MCH2EN bit in TIMEx_CHCTL2 register is cleared. 00: Prescaler disabled, capture is done on each channel input edge. 01: Capture is done every 2 channel input edges. 10: Capture is done every 4 channel input edges. 11: Capture is done every 8 channel input edges.
1:0	MCH2MS[1:0]	Multi mode channel 2 I/O mode selection Same as output compare mode.

Multi mode channel control register 2 (TIMEx_MCHCTL2)

Address offset: 0x50

Reset value: 0x0000 0000

This register can be accessed by half-word(16-bit) or word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								MCH3FP[1:0]		MCH2FP[1:0]		MCH1FP[1:0]		MCH0FP[1:0]	
								rw		rw		rw		rw	

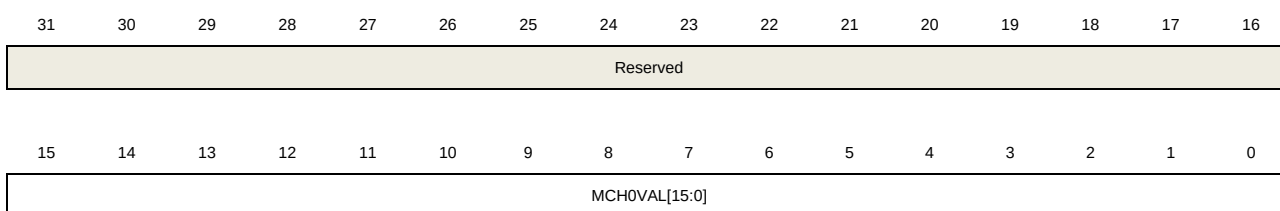
Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7:6	MCH3FP[1:0]	Multi mode channel 3 capture/compare free polarity Refer to MCH0FP[1:0] description.
5:4	MCH2FP[1:0]	Multi mode channel 2 capture/compare free polarity Refer to MCH0FP[1:0] description.
3:2	MCH1FP[1:0]	Multi mode channel 1 capture/compare free polarity Refer to MCH0FP[1:0] description.
1:0	MCH0FP[1:0]	Multi mode channel 0 capture/compare free polarity When multi mode channel 0 is configured in output mode, and the MCH0MSEL[1:0] = 2b'00, these bits specify the multi mode channel 0 output signal polarity. 00: Multi mode channel 0 active high 01: Multi mode channel 0 active low 10: Reserved. 11: Reserved. When multi mode channel 0 is configured in input mode, these bits specify the multi mode channel 0 input signal's polarity. MCH0FP[1:0] will select the active trigger or capture polarity for multi mode channel 0 input signals. 00: Multi mode channel 0 input signal's rising edge is the active signal for capture or trigger operation in slave mode. And multi mode channel 0 input signal will not be inverted. 01: Multi mode channel 0 input signal's falling edge is the active signal for capture or trigger operation in slave mode. And multi mode channel 0 input signal will be inverted. 10: Reserved. 11: Noninverted / both multi mode channel 0 input signal's edges. This bit cannot be modified when PROT[1:0] bit-field in TIMEx_CCHP register is 11 or 10.

Multi mode channel 0 capture/compare value register (TIMEx_MCH0CV)

Address offset: 0x54

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



rw

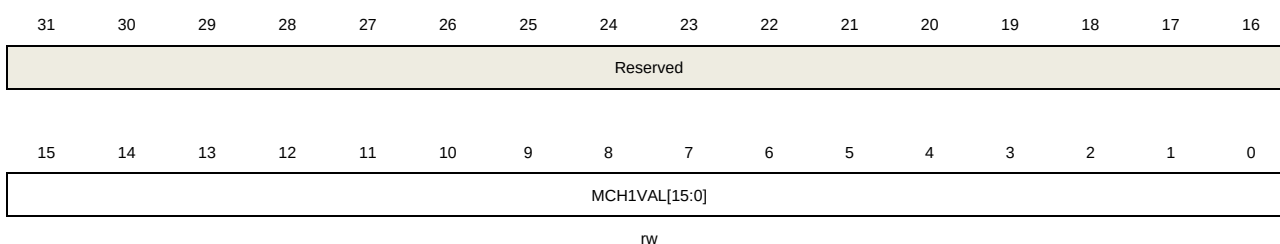
Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	MCH0VAL[15:0]	Capture/compare value of multi mode channel 0. When multi mode channel 0 is configured in input mode, this bit-field indicates the counter value at the last capture event. And this bit-field is read-only. When multi mode channel 0 is configured in output mode, this bit-field contains value to be compared to the counter. When the corresponding shadow register is enabled, the shadow register updates at the set moment according to the relevant configuration of the shadow register loading function.

Multi mode channel 1 capture/compare value register (TIMERx_MCH1CV)

Address offset: 0x58

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



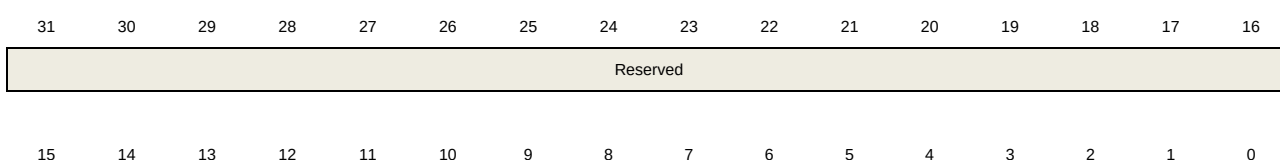
Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	MCH1VAL[15:0]	Capture/compare value of multi mode channel 1. When multi mode channel 1 is configured in input mode, this bit-field indicates the counter value at the last capture event. And this bit-field is read-only. When multi mode channel 1 is configured in output mode, this bit-field contains value to be compared to the counter. When the corresponding shadow register is enabled, the shadow register updates at the set moment according to the relevant configuration of the shadow register loading function.

Multi mode channel 2 capture/compare value register (TIMERx_MCH2CV)

Address offset: 0x5C

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



MCH2VAL[15:0]

rw

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	MCH2VAL[15:0]	Capture/compare value of multi mode channel 2. When multi mode channel 2 is configured in input mode, this bit-field indicates the counter value at the last capture event. And this bit-field is read-only. When multi mode channel 2 is configured in output mode, this bit-field contains value to be compared to the counter. When the corresponding shadow register is enabled, the shadow register updates at the set moment according to the relevant configuration of the shadow register loading function.

Multi mode channel 3 capture/compare value register (TIMERx_MCH3CV)

Address offset: 0x60

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MCH3VAL[15:0]															

rw

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	MCH3VAL[15:0]	Capture/compare value of channel 3. When multi mode channel 3 is configured in input mode, this bit-field indicates the counter value at the last capture event. And this bit-field is read-only. When multi mode channel 3 is configured in output mode, this bit-field contains value to be compared to the counter. When the corresponding shadow register is enabled, the shadow register updates at the set moment according to the relevant configuration of the shadow register loading function.

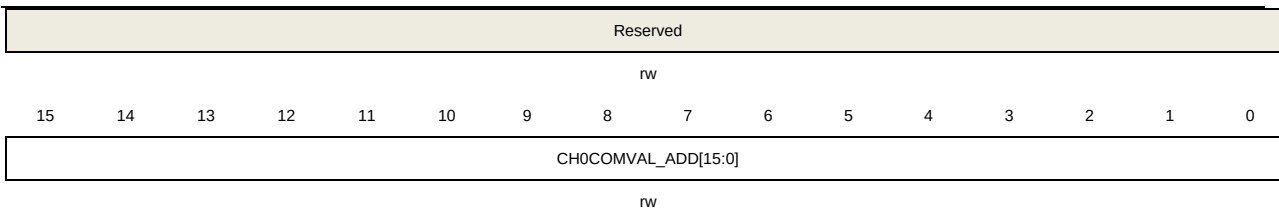
Channel 0 additional compare value register (TIMERx_CH0COMV_ADD)

Address offset: 0x64

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

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



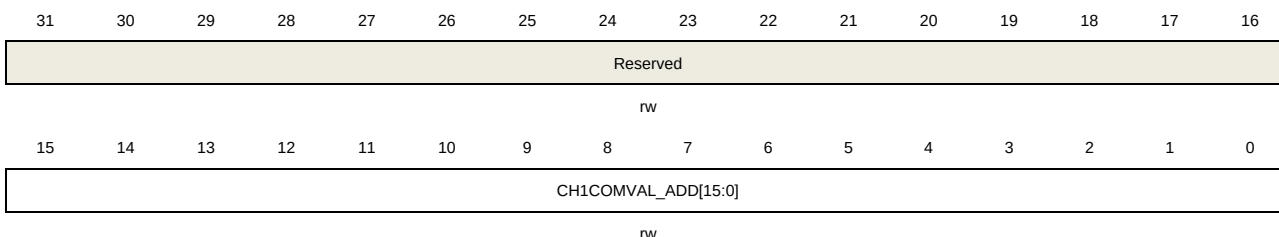
Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CH0COMVAL_ADD [15:0]	Additional compare value of channel 0 When channel 0 is configured in output mode, this bit-field contains value to be compared to the counter. When the corresponding shadow register is enabled, the shadow register updates at the set moment according to the relevant configuration of the shadow register loading function. Note: This register just used in composite PWM mode and asymmetric PWM mode (when CH0COMCTL[3:2] = 2'b10 and CH0MS[2:0] = 3'b000).

Channel 1 additional compare value register (TIMERx_CH1COMV_ADD)

Address offset: 0x68

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



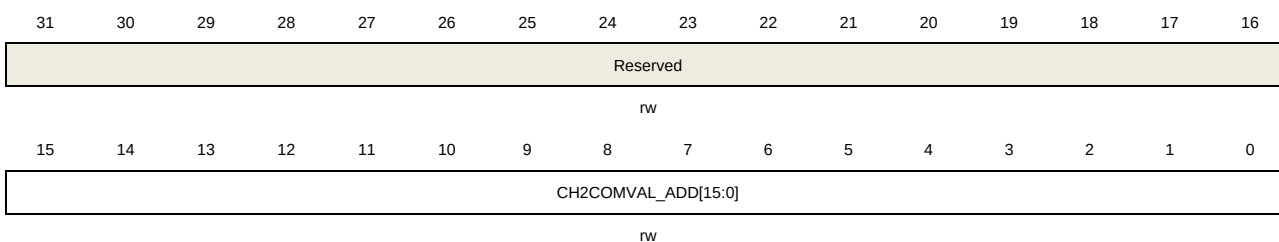
Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CH1COMVAL_ADD [15:0]	Additional compare value of channel 1 When channel 1 is configured in output mode, this bit-field contains value to be compared to the counter. When the corresponding shadow register is enabled, the shadow register updates at the set moment according to the relevant configuration of the shadow register loading function. Note: This register just used in composite PWM mode and asymmetric PWM mode (when CH1COMCTL[3:2] = 2'b10 and CH1MS[2:0] = 3'b000).

Channel 2 additional compare value register (TIMERx_CH2COMV_ADD)

Address offset: 0x6C

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



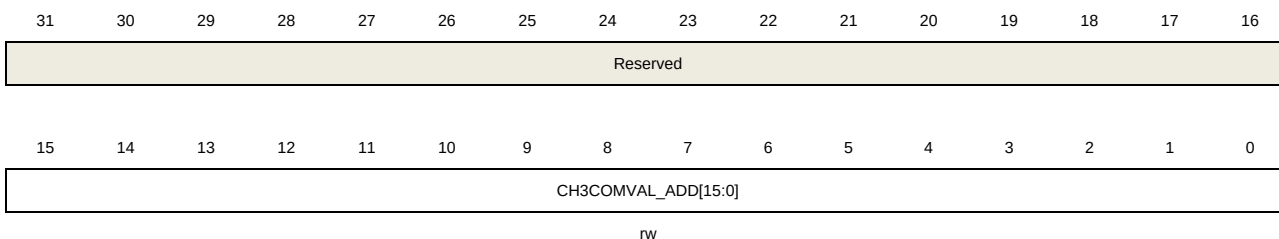
Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CH2COMVAL_ADD [15:0]	Additional compare value of channel 2 When channel 2 is configured in output mode, this bit-field contains value to be compared to the counter. When the corresponding shadow register is enabled, the shadow register updates at the set moment according to the relevant configuration of the shadow register loading function. Note: This register just used in composite PWM mode and asymmetric PWM mode (when CH2COMCTL[3:2] = 2'b10 and CH2MS[2:0] = 3'b000).

Channel 3 additional compare value register (TIMERx_CH3COMV_ADD)

Address offset: 0x70

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CH3COMVAL_ADD [15:0]	Additional compare value of channel 3 When channel 3 is configured in output mode, this bit-field contains value to be compared to the counter. When the corresponding shadow register is enabled, the shadow register updates at the set moment according to the relevant configuration of the shadow register loading function. Note: This register just used in composite PWM mode and asymmetric PWM mode (when CH3COMCTL[3:2] = 2'b10 and CH3MS[2:0] = 3'b000).

Control register 2 (TIMERx_CTL2)

Address offset: 0x74

Reset value: 0x0FF0 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				MCH3MSEL[1:0]		MCH2MSEL[1:0]		MCH1MSEL[1:0]		MCH0MSEL[1:0]		Reserved			
rw				rw		rw		rw		rw					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH3OMPSEL[1:0]		CH2OMPSEL[1:0]		CH1OMPSEL[1:0]		CH0OMPSEL[1:0]		Reserved							
rw		rw		rw		rw									

Bits	Fields	Descriptions
31:28	Reserved	Must be kept at reset value.
27:26	MCH3MSEL[1:0]	Multi mode channel 3 mode select 00: Independent mode, MCH3 is independent of CH3 01: Reserved 10: Reserved 11: Complementary mode, only the CH3 is valid for input, and the outputs of MCH3 and CH3 are complementary
25:24	MCH2MSEL[1:0]	Multi mode channel 2 mode select 00: Independent mode, MCH2 is independent of CH2 01: Reserved 10: Reserved 11: Complementary mode, only the CH2 is valid for input, and the outputs of MCH2 and CH2 are complementary
23:22	MCH1MSEL[1:0]	Multi mode channel 1 mode select 00: Independent mode, MCH1 is independent of CH1 01: Reserved 10: Reserved 11: Complementary mode, only the CH1 is valid for input, and the outputs of MCH1 and CH1 are complementary
21:20	MCH0MSEL[1:0]	Multi mode channel 0 mode select 00: Independent mode, MCH0 is independent of CH0 01: Reserved 10: Reserved 11: Complementary mode, only the CH0 is valid for input, and the outputs of MCH0 and CH0 are complementary
19:16	Reserved	Must be kept at reset value.

15:14	CH3OMPSEL[1:0]	Channel 3 output match pulse select When the match events occur, this bit is used to select the output of O3CPRE which drives CH3_O. 00: The O3CPRE signal is output normally with the configuration of CH3COMCTL[3:0] bits. 01: Only the counter is counting up, the O3CPRE signal is output a pulse when the match events occur, and the pulse width is one CK_TIMER clock cycle. 10: Only the counter is counting down, the O3CPRE signal is output a pulse when the match events occur, and the pulse width is one CK_TIMER clock cycle. 11: Both the counter is counting up and counting down, the O3CPRE signal is output a pulse when the match events occur, and the pulse width is one CK_TIMER clock cycle.
13:12	CH2OMPSEL[1:0]	Channel 2 output match pulse select When the match events occurs, this bit is used to select the output of O2CPRE which drives CH2_O. 00: The O2CPRE signal is output normal with the configuration of CH2COMCTL[3:0] bits. 01: Only when the counter is counting up, the O2CPRE signal is output a pulse when the match events occurs, and the pulse width is one CK_TIMER clock cycle. 10: Only when the counter is counting down, the O2CPRE signal is output a pulse when the match events occurs, and the pulse width is one CK_TIMER clock cycle. 11: Both when the counter is counting up and counting down, the O2CPRE signal is output a pulse when the match events occurs, and the pulse width is one CK_TIMER clock cycle.
11:10	CH1OMPSEL[1:0]	Channel 1 output match pulse select When the match events occurs, this bit is used to select the output of O1CPRE which drives CH1_O. 00: The O1CPRE signal is output normal with the configuration of CH1COMCTL[3:0] bits. 01: Only when the counter is counting up, the O1CPRE signal is output a pulse when the match events occurs, and the pulse width is one CK_TIMER clock cycle. 10: Only when the counter is counting down, the O1CPRE signal is output a pulse when the match events occurs, and the pulse width is one CK_TIMER clock cycle. 11: Both when the counter is counting up and counting down, the O1CPRE signal is output a pulse when the match events occurs, and the pulse width is one CK_TIMER clock cycle.
9:8	CH0OMPSEL[1:0]	Channel 0 output match pulse select When the match events occurs, this bit is used to select the output of O0CPRE which drives CH0_O. 00: The O0CPRE signal is output normal with the configuration of CH0COMCTL[3:0] bits. 01: Only when the counter is counting up, the O0CPRE signal is output a pulse

when the match events occurs, and the pulse width is one CK_TIMER clock cycle.

10: Only when the counter is counting down, the O0CPRE signal is output a pulse when the match events occurs, and the pulse width is one CK_TIMER clock cycle.

11: Both when the counter is counting up and counting down, the O0CPRE signal is output a pulse when the match events occurs, and the pulse width is one CK_TIMER clock cycle.

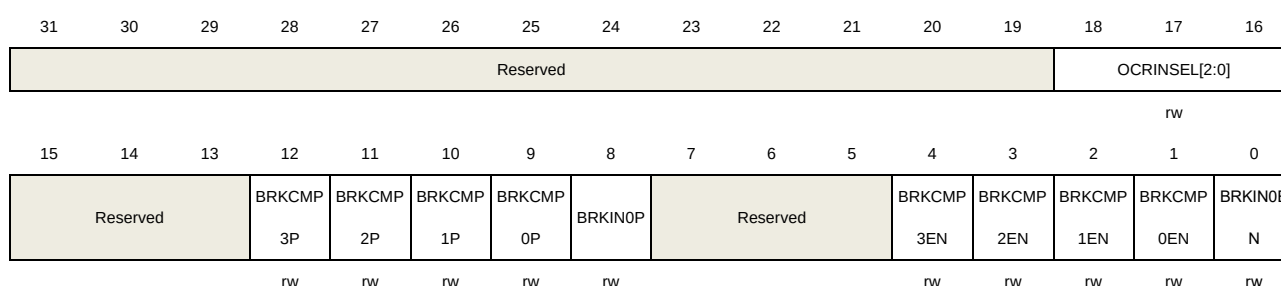
7:0 Reserved Must be kept at reset value.

Alternate function control register (TIMERx_AFCTL)

Address offset: 0x8C

Reset value: 0x0000 0001

This register has to be accessed by word (32-bit).



Bits	Fields	Descriptions
31:19	Reserved	Must be kept at reset value.
18:16	OCRINSEL[2:0]	OCPRE_CLR inputs selection 000: OCPRE_CLR0 (CMP0 output) 001: OCPRE_CLR1 (CMP1 output) 010: OCPRE_CLR2 (CMP2 output) 011: OCPRE_CLR3 (CMP3 output) Note: This bit can be modified only when PROT[1:0] bit-field in TIMERx_CCHP register is 00.
15:13	Reserved	Must be kept at reset value.
12	BRKCOMP3P	BREAK CMP3 input polarity, reference to BRKCOMP0P
11	BRKCOMP2P	BREAK CMP2 input polarity, reference to BRKCOMP0P
10	BRKCOMP1P	BREAK CMP1 input polarity, reference to BRKCOMP0P
9	BRKCOMP0P	This bit specifies the polarity of the CMP0 input signal. This bit is used to configure the CMP0 input polarity, and the specific polarity is determined by this bit and the BRKP bit. 0: CMP0 input signal will not be inverted (BRKP =0, the input signal is active low; BRKP =1, the input signal is active high)

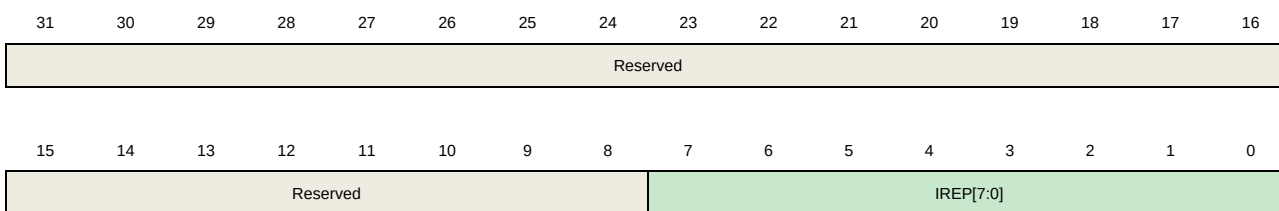
		1: CMP0 input signal will be inverted (BRKP =0, the input signal is active high; BRKP =1, the input signal is active low) Note: This bit can be modified only when PROT[1:0] bit-field in TIMERx_CCHP register is 00.
8	BRKIN0P	BRKIN0 input signal polarity This bit is used to configure the BRKIN0 input polarity, and the specific polarity is determined by this bit and the BRKP bit. 0: BRKIN0 input signal will not be inverted (BRKP =0, the input signal is active low; BRKP =1, the input signal is active high) 1: BRKIN0 input signal will be inverted (BRKP =0, the input signal is active high; BRKP =1, the input signal is active low) Note: This bit can be modified only when PROT[1:0] bit-field in TIMERx_CCHP register is 00.
7:5	Reserved	Must be kept at reset value.
4	BRKCMP3EN	BREAK CMP3 enable, reference to BRKCMP0EN
3	BRKCMP2EN	BREAK CMP2 enable, reference to BRKCMP0EN
2	BRKCMP1EN	BREAK CMP1 enable, reference to BRKCMP0EN
1	BRKCMP0EN	BREAK CMP0 enable 0: CMP0 input disabled 1: CMP0 input enabled Note: This bit can be modified only when PROT[1:0] bit-field in TIMERx_CCHP register is 00.
0	BRKIN0EN	BRKIN0 input signal enable 0: BRKIN0 inputs disabled 1: BRKIN0 inputs enabled Note: This bit can be modified only when PROT[1:0] bit-field in TIMERx_CCHP register is 00.

Interrupt Counter repetition register (TIMERx_IREP)

Address offset: 0x90

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



rw

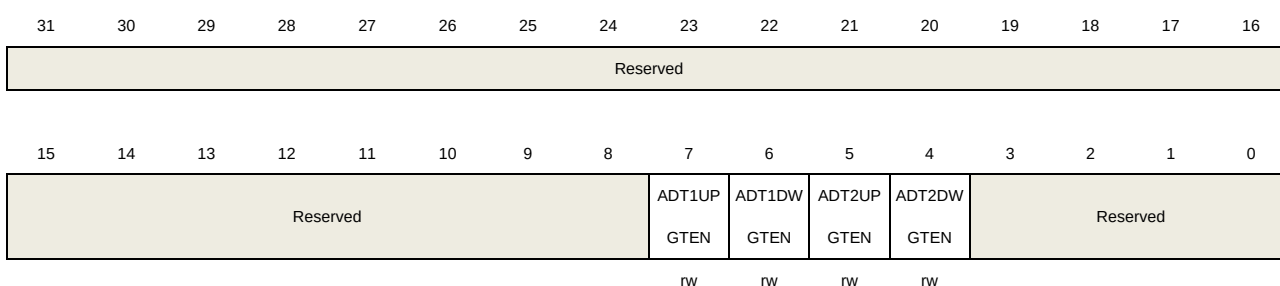
Bits	Fields	Descriptions
31:4	Reserved	must be kept at reset value
3:0	IREF[7:0]	Interrupt counter repetition value This bit-field specifies the overflow or underflow interrupt generation rate. Each time the repetition counter counts down to zero, a flow interrupt will be generated.

ADC conversion request control register (TIMERx_ADCRCTL)

Address offset: 0x94

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



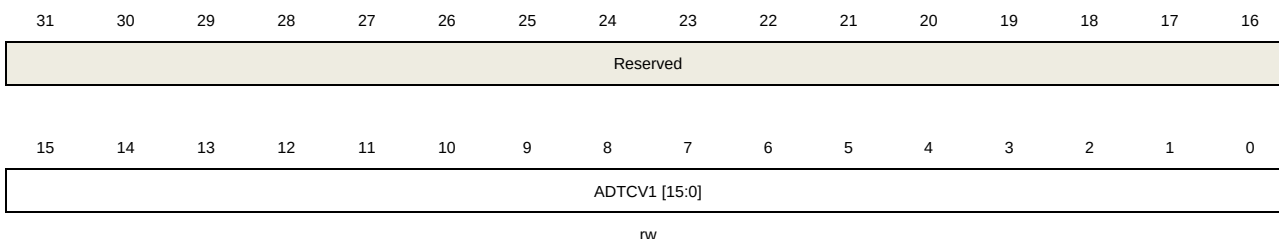
Bits	Fields	Descriptions
31:8	Reserved	must be kept at reset value
7	ADT1UPGTEN	The ADTCV1[15:0] comparison match (count up only) generate trigger enable 0: Disable 1: Enable
6	ADT1DWGTEN	The ADTCV1[15:0] comparison match (count down only) generate trigger enable 0: Disable 1: Enable
5	ADT2UPGTEN	The ADTCV2[15:0] comparison match (count up only) generate trigger enable 0: Disable 1: Enable
4	ADT2DWGTEN	The ADTCV2[15:0] comparison match (count down only) generate trigger enable 0: Disable 1: Enable
3:0	Reserved	must be kept at reset value

ADC trigger compare value register1(TIMERx_ADCCR1)

Address offset: 0x98

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



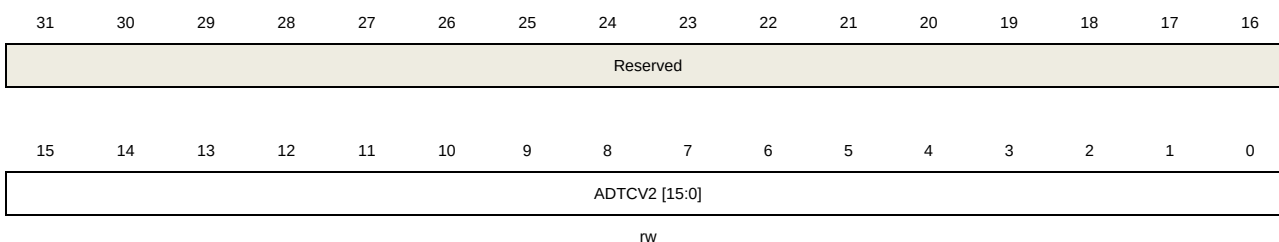
Bits	Fields	Descriptions
31:16	Reserved	must be kept at reset value
15:0	ADTCV1 [15:0]	The ADC trigger compare value 1 These bits contain the value of the comparison with the counter. After the corresponding shadow register is enabled, the register value is updated with the configuration of the shadow register.

ADC trigger compare value register2 (TIMERx_ADCCR2)

Address offset: 0x9C

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



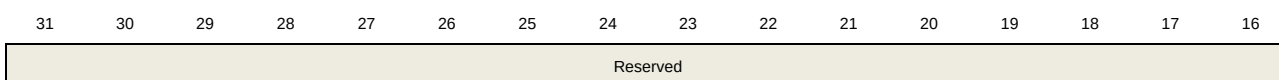
Bits	Fields	Descriptions
31:16	Reserved	must be kept at reset value
15:0	ADTCV2 [15:0]	The ADC trigger compare value 2 These bits contain the value of the comparison with the counter. After the corresponding shadow register is enabled, the register value is updated with the configuration of the shadow register.

ADC counter repetition register (TIMERx_ADCREP)

Address offset: 0xA0

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



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

Bits	Fields	Descriptions
31:9	Reserved	must be kept at reset value
11:8	ADCREPAB[3:0]	ADC TIMEx_TRGAB repeat count setting These bits specify the TIMEx_TRGAB repeat count within the range from 0 to 15. When the comparison trigger generates TIMEx_TRGAB, the counter starts to decline from the these bits count value, until the counter is 0 to generate the event trigger signal TIMEx_TRGAB.
7:4	ADCREPB[3:0]	ADC TIMEx_TRGB repeat count setting These bits specify the TIMEx_TRGB repeat count within the range from 0 to 15. When the comparison trigger generates TIMEx_TRGB, the counter starts to decline from the these bits count value, until the counter is 0 to generate the event trigger signal TIMEx_TRGB.
3:0	ADCREPA[3:0]	ADC TIMEx_TRGA repeat count setting These bits specify the TIMEx_TRGA repeat count within the range from 0 to 15. When the comparison trigger generates TIMEx_TRGA, the counter starts to decline from the these bits count value, until the counter is 0 to generate the event trigger signal TIMEx_TRGA.

ADC trigger control register (TIMEx_ADCTL)

Address offset: 0xAC

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADREPRLD	Reserved					ADTPRE EN2	ADTPRE EN1	ADTSOF	ADTSUF	Reserved				ADCREP BMD	ADCREP AMD
rw						rw	rw	rw	rw					rw	rw

Bits	Fields	Descriptions
31:16	Reserved	must be kept at reset value.
15	ADREPRLD	ADC repeat count reload immediately When this bit is set to 1, the TIMEx_TRGA repeat counter, TIMEx_TRGB repeat counter, and TIMEx_TRGAB repeat counter is reloaded to the respective repeat

		count setting value. After the reload is complete, this bit is automatically cleared by hardware.
14:10	Reserved	must be kept at reset value.
9	ADTPREEN2	ADTCV2[15:0] preload enable 0: Disable ADTCV2[15:0] shadow register 1: Enable ADTCV2[15:0] shadow register Note: this bit also links to the shadow register loading function.
8	ADTPREEN1	ADTCV1[15:0] preload enable 0: Disable ADTCV1[15:0] register shadow 1: Enable ADTCV1[15:0] register shadow Note: this bit also links to the shadow register loading function.
7	ADTSOF	ADC trigger signal TIMERx_TRGOF enable when TIMERx_CNT overflow 0: TIMERx_TRGOF generation by TIMERx_CNT overflow trigger disable 1: TIMERx_TRGOF generation by TIMERx_CNT overflow trigger enable
6	ADTSUF	ADC trigger signal TIMERx_TRGUF enable when TIMERx_CNT underflow 0: TIMERx_TRGUF generation by TIMERx_CNT underflow trigger disable 1: TIMERx_TRGUF generation by TIMERx_CNT underflow trigger enable
5:2	Reserved	must be kept at reset value
1	ADCREPBMD	ADC TIMERx_TRGB repeat counter decrement mode setting. 0: When the timer underflow, TIMERx_TRGB repeat counter will be decremented from ADCREPB, until repeat counter counts to 0, and the trigger signal TIMERx_TRGB occurs. 1: When the timer overflow, TIMERx_TRGB repeat counter will be decremented from ADCREPB, until repeat counter counts to 0, and the trigger signal TIMERx_TRGB occurs.
0	ADCREPAMD	ADC TIMERx_TRGA repeat counter decrement mode setting. 0: When the timer underflow, TIMERx_TRGA repeat counter will be decremented from ADCREPA, until repeat counter counts to 0, and the trigger signal TIMERx_TRGA occurs. 1: When the timer overflow, TIMERx_TRGA repeat counter will be decremented from ADCREPA, until repeat counter counts to 0, and the trigger signal TIMERx_TRGA occurs.

Shadow update mode register 0 (TIMERx_SHUPM0)

Address offset: 0xC4

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

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

CARUPSEL[1:0]	ADCCR2UPSEL[1:0]	ADCCR1UPSEL[1:0]	Reserved	CH3COMV_ADDUPSEL[2:0]	CH3CVUPSEL[2:0]	CH2COMV_ADDUPSEL[2:0]									
rw		rw		rw		rw									
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	CH2CVUPSEL[2:0]		CH1COMV_ADDUPSEL[2:0]		CH1CVUPSEL[2:0]		CH0COMV_ADDUPSEL[2:0]		CH0CVUPSEL[2:0]						
rw		rw		rw		rw		rw							

Bits	Fields	Descriptions
31:30	CARUPSEL[1:0]	The counter auto reload value update mode selection, used to set when the counter auto reload value will be automatically overloaded. 00: Enable global update function. 01: The auto reload value is updated at the next counter overflow, UPG = 1 or master-slave mode (restart, restart + event). 10: The auto reload value is updated at the next counter underflow, UPG = 1 or master-slave mode (restart, restart + event). 11: The auto reload value is updated at the next counter overflow or underflow, UPG = 1 or master-slave mode (restart, restart + event). Note: 1. In the up count mode, the comparison value is updated at the overflow of the counter, while in the down count mode, the comparison value is updated at the underflow of the counter. 2. These bits are controlled by UPDIS bit in the TIMEx_CTL0 register. 3. The global update function is to update uniformly at the update event.
29:28	ADCCR2UPSEL[1:0]	The ADC triggered comparison value 2 update mode selection. Refer to ADCCR1UPSEL[1:0] description
27:26	ADCCR1UPSEL[1:0]	The ADC triggered comparison value 1 update mode selection, used to set when the ADC triggered comparison value is automatically overloaded. 00: Enable global update function. 01: The triggered comparison value is updated at the next counter overflow, UPG = 1 or master-slave mode (restart, restart + event). 10: The triggered comparison value is updated at the next counter underflow, UPG = 1 or master-slave mode (restart, restart + event). 11: The triggered comparison value is updated at the next counter overflow or underflow, UPG = 1 or master-slave mode (restart, restart + event). Note: 1. In the up count mode, the comparison value is updated at the overflow of the counter, while in the down count mode, the comparison value is updated at the underflow of the counter. 2. These bits are controlled by UPDIS bit in the TIMEx_CTL0 register. 3. The global update function is to update uniformly at the update event.
25	Reserved	Must be kept at reset value.
24:22	CH3COMV_ADDUPSEL[2:0]	Channel 3 additional comparison value update mode selection. Refer to CH0COMV_ADDUPSEL[2:0] description
21:19	CH3CVUPSEL[2:0]	Channel 3 capture/compare value update mode selection.

		Refer to CH0CVUPSEL[2:0] description
18:16	CH2COMV_ADDUPSEL[2:0]	Channel 2 additional comparison value update mode selection. Refer to CH0COMV_ADDUPSEL[2:0] description
15	Reserved	Must be kept at reset value.
14:12	CH2CVUPSEL[2:0]	Channel 2 capture/compare value update mode selection. Refer to CH0CVUPSEL[2:0] description
11:9	CH1COMV_ADDUPSEL[2:0]	Channel 1 additional comparison value update mode selection. Refer to CH0COMV_ADDUPSEL[2:0] description
8:6	CH1CVUPSEL[2:0]	Channel 1 capture/compare value update mode selection. Refer to CH0CVUPSEL[2:0] description
5:3	CH0COMV_ADDUPSEL[2:0]	Channel 0 additional comparison value update mode selection, used to set when additional comparison values are automatically overloaded. 000: Enable global update function. 001: The additional comparison value is updated at the next counter overflow, UPG = 1 or master-slave mode (restart, restart + event). 010: The additional comparison value is updated at the next counter underflow, UPG = 1 or master-slave mode (restart, restart + event). 011: The additional comparison value is updated at the next counter overflow or underflow, UPG = 1 or master-slave mode (restart, restart + event). 100: The additional comparison value is updated at the next counting cycle and the matching moment of the channel comparison value. Note: 1. When CH0COMV_ADDUPSEL[2]=1'b0. In the up count mode, the comparison value is updated at the overflow of the counter, while in the down count mode, the comparison value is updated at the underflow of the counter. 2. These bits are controlled by UPDIS bit in the TIMEx_CTL0 register. 3. The global update function is to update uniformly at the update event.
2:0	CH0CVUPSEL[2:0]	Channel 0 capture/compare value update mode selection, used to set when the compare value is automatically overloaded. 000: Enable global update function. 001: The comparison value is updated at the next counter overflow, UPG = 1 or master-slave mode (restart, restart + event). 010: The comparison value is updated at the next counter underflow, UPG = 1 or master-slave mode (restart, restart + event). 011: The comparison value is updated at the next counter overflow or underflow, UPG = 1 or master-slave mode (restart, restart + event). Note: 1. In the up count mode, the comparison value is updated at the overflow of the counter, while in the down count mode, the comparison value is updated at the underflow of the counter. 2. These bits are controlled by UPDIS bit in the TIMEx_CTL0 register.

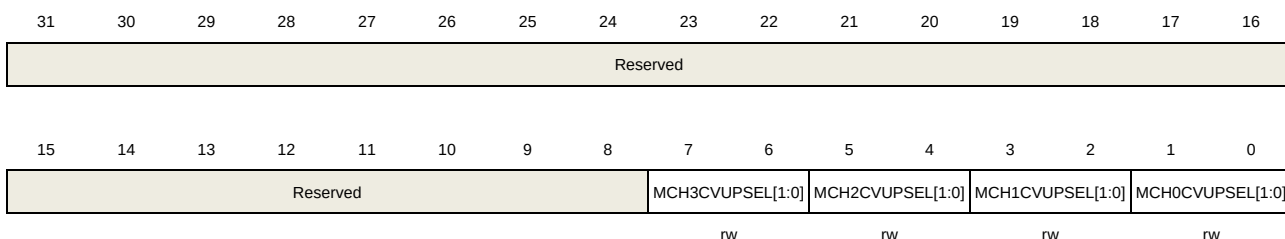
3. The global update function is to update uniformly at the update event.

Shadow update mode register 1 (TIMERx_SHUPM1)

Address offset: 0xC8

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



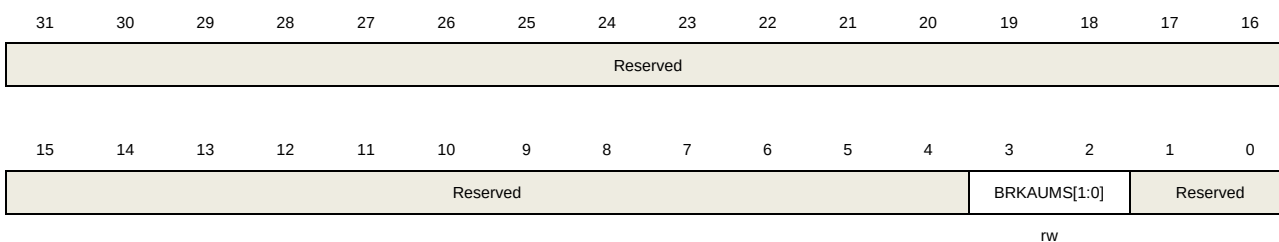
Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7:6	MCH3CVUPSEL [1:0]	Multi mode channel 3 capture/compare value update mode selection. Refer to MCH0CVUPSEL[1:0] description
5:4	MCH2CVUPSEL [1:0]	Multi mode channel 2 capture/compare value update mode selection. Refer to MCH0CVUPSEL[1:0] description
3:2	MCH1CVUPSEL [1:0]	Multi mode channel 1 capture/compare value update mode selection. Refer to MCH0CVUPSEL[1:0] description
1:0	MCH0CVUPSEL [1:0]	Multi mode channel 0 capture/compare value update mode selection, used to set when the compare value is automatically overloaded. 00: Enable global update function. 01: The comparison value is updated at the next counter overflow, UPG = 1 or master-slave mode (restart, restart + event). 10: The comparison value is updated at the next counter underflow, UPG = 1 or master-slave mode (restart, restart + event). 11: The comparison value is updated at the next counter overflow or underflow, UPG = 1 or master-slave mode (restart, restart + event). Note: 1. In the up count mode, the comparison value is updated at the overflow of the counter, while in the down count mode, the comparison value is updated at the underflow of the counter. 2. These bits are controlled by UPDIS bit in the TIMERx_CTL0 register. 3. The global update function is to update uniformly at the update event.

Counter overflow or underflow mode selection register (TIMERx_COUNSEL)

Address offset: 0xCC

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



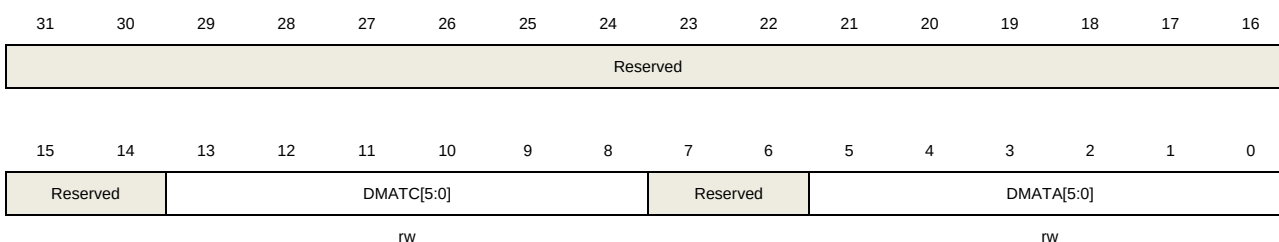
Bits	Fields	Descriptions
31:4	Reserved	Must be kept at reset value.
3:2	BRKAUMS[1:0]	Break automatic output function mode selection. If the break input is invalid, when the poen can be set. 00: POEN bit is set when the next counter overflow or underflow 01: POEN bit is set when the next counter overflow 10: POEN bit is set when the next counter underflow 11: POEN bit is set when the next update event This bit can be modified only when PROT[1:0] bit-field in TIMERx_CCHP register is 00.
1:0	Reserved	Must be kept at reset value.

DMA configuration register (TIMERx_DMACFG)

Address offset: 0xE0

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



Bits	Fields	Descriptions
31:14	Reserved	Must be kept at reset value.
13:8	DMATC[5:0]	DMA transfer count This field defines the times of accessing(R/W) the TIMERx_DMATB register by DMA. 6'b000000: transfer 1 time 6'b000001: transfer 2 times ... 6'b110011: transfer 52 times

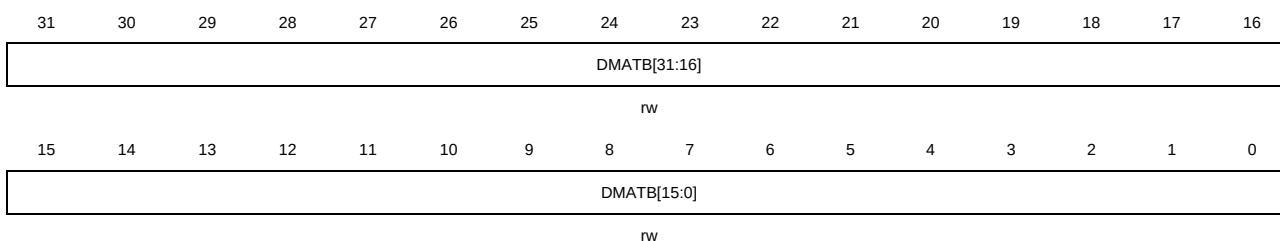
7:6	Reserved	Must be kept at reset value.
5:0	DMATA[5:0]	DMA transfer access start address This field defines the start address of accessing the TIMEx_DMATB register by DMA. When the first access to the TIMEx_DMATB register is done, this bit-field specifies the address just accessed. And then the address of the second access to the TIMEx_DMATB register will be (start address + 0x4). 6'b000000: TIMEx_CTL0 6'b000001: TIMEx_CTL1 ... 6'b110011: TIMEx_COUNSEL In a word: start address = TIMEx_CTL0 + DMATA*4

DMA transfer buffer register (TIMEx_DMATB)

Address offset: 0xE4

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



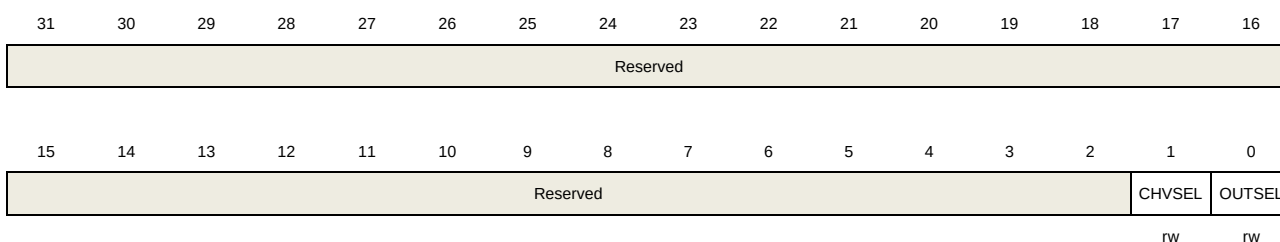
Bits	Fields	Descriptions
31:0	DMATB[31:0]	DMA transfer buffer When a read or write operation is assigned to this register, the register located at the address ranges from (start address) to (start address + transfer count * 4) will be accessed. The transfer count is calculated by hardware, and ranges from 0 to DMATC.

Configuration register (TIMEx_CFG)

Address offset: 0xFC

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



Bits	Fields	Descriptions
31:2	Reserved	Must be kept at reset value.
1	CHVSEL	Write CHxVAL register selection bit This bit-field is set and reset by software. 1: If the value to be written to the CHxVAL register is the same as the value of CHxVAL register, the write access is ignored. 0: No effect.
0	OUTSEL	The output value selection bit This bit-field is set and reset by software. 1: If POEN bit and IOS bit are 0, the output is disabled. 0: No effect.

21.2 General level0 timer (TIMERx, x=1,2)

21.2.1 Overview

The general level0 timer module (TIMER1/2) is a four-channel timer that supports input capture and output compare. They can generate PWM signals to control motor or be used for power management applications. The general level0 timer has a 16-bit counter that can be used as an unsigned counter.

In addition, the general level0 timers can be programmed and be used for counting, their external events can be used to drive other timers.

Timers are completely independent with each other, but they may be synchronized to provide a larger timer with their counter value increasing in unison.

21.2.2 Characteristics

- Total channel num: 4.
- Counter width: 16 bits.
- Selectable clock source: internal clock, internal trigger, external input, external trigger.
- Multiple counter modes: up counting, down counting and center-aligned counting.
- Quadrature decoder: used for motion tracking and determination of both rotation direction and position.
- Hall sensor function: used for 3-phase motor control.
- Programmable prescaler: 16 bits. The factor can be changed ongoing.
- Each channel is user-configurable: input capture mode, output compare mode, programmable PWM mode and single pulse mode.
- Auto reload function.
- Interrupt output or DMA request: update event, trigger event and compare/capture event.
- Daisy chaining of timer module allows a single timer to start multiple timers.
- Timer synchronization allows the selected timers to start counting on the same clock cycle.
- Timer master-slave management.

21.2.3 Block diagram

[Figure 21-55. General Level 0 timer block diagram](#) provides details on the internal configuration of the general level 0 timer.

The block diagram illustrates the internal architecture of the TIMx peripheral. It shows the flow of data from input channels (CH0_I, CH1_I, CH2_I, CH3_I) through the Input Logic (Synchronizer & Filter & Edge Detector) to the Edge selector, Prescaler, Counter, and Output Logic. The diagram also shows the connection to the APB BUS, Register /Interrupt, and the DMA controller. The Output Logic generates signals for CH0_O, CH1_O, CH2_O, and CH3_O. The Register /Interrupt block handles the Register set and update, Interrupt collector and controller, and the CAR (Capture/Compare Register). The DMA controller manages the DMA request and DMA REQ/ACK signals. The Counter block receives the TIMx CK and TIMx DMA request signals and outputs the TIMx_CHxCV (Capture/Compare Value) to the Output Logic. The Prescaler block receives the TIMx TRGO signal and outputs the TIMx CHxPSC (Prescaler) to the Counter. The Edge selector block receives the TIMx CHxPSC signal and outputs the TIMx CHxPSC to the Counter. The Input Logic block receives the TIMx CHxPSC signal and outputs the TIMx CHxPSC to the Edge selector. The External Trigger Input logic block receives the On-chip ETI sources and ETI signals and outputs the ETIFP (External Trigger Input Filter) signal to the Input Logic. The APB BUS is connected to the Register /Interrupt block. The Interrupt signal is generated by the Register /Interrupt block and sent to the Interrupt collector and controller. The Update, Trigger, and Cap/Com signals are sent from the Register /Interrupt block to the Output Logic. The CAR block is connected to the Register /Interrupt block and the Output Logic. The DMA controller is connected to the Counter, Output Logic, and the DMA REQ/ACK signal. The Counter block is connected to the Prescaler, Edge selector, and the TIMx CHxCV signal. The Prescaler block is connected to the Edge selector and the TIMx CHxPSC signal. The Edge selector block is connected to the Input Logic and the TIMx CHxPSC signal. The Input Logic block is connected to the input channels and the TIMx CHxPSC signal. The External Trigger Input logic block is connected to the On-chip ETI sources and ETI signals. The APB BUS is connected to the Register /Interrupt block. The Interrupt signal is generated by the Register /Interrupt block and sent to the Interrupt collector and controller. The Update, Trigger, and Cap/Com signals are sent from the Register /Interrupt block to the Output Logic. The CAR block is connected to the Register /Interrupt block and the Output Logic. The DMA controller is connected to the Counter, Output Logic, and the DMA REQ/ACK signal. The Counter block is connected to the Prescaler, Edge selector, and the TIMx CHxCV signal. The Prescaler block is connected to the Edge selector and the TIMx CHxPSC signal. The Edge selector block is connected to the Input Logic and the TIMx CHxPSC signal. The Input Logic block is connected to the input channels and the TIMx CHxPSC signal. The External Trigger Input logic block is connected to the On-chip ETI sources and ETI signals.

Clock selection

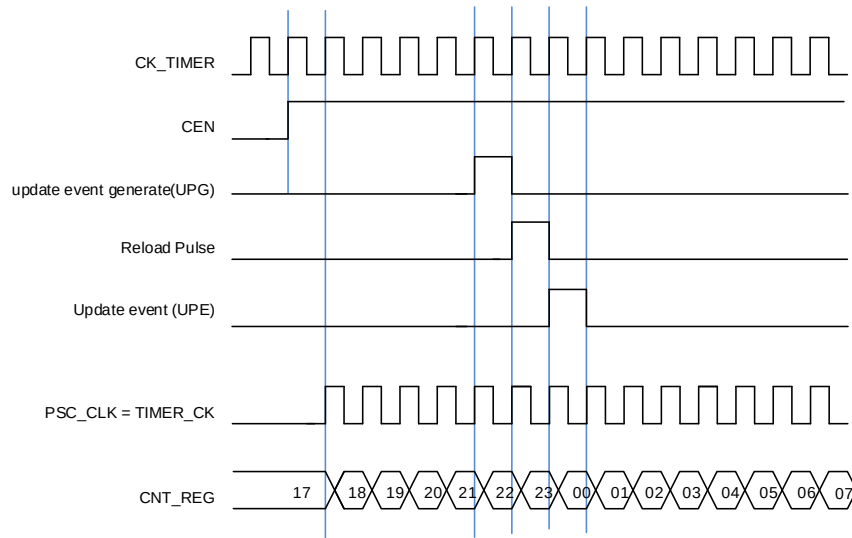
- TSCFGy[4:0] (y=0..7,9..12,15) = 5'b00000 in SYSCFG_TIMERxCFG(x=1,2) registers. Internal clock CK_TIMER is selected as timer clock source which is from module RCU.

The default clock source is the CK_TIMER for driving the counter prescaler when TSCFGy[4:0] (y=0..7,9..12,15) = 5'b00000 in SYSCFG_TIMERxCFG(x=1,2) registers. When the CEN is set, the CK_TIMER will be divided by PSC value to generate PSC_CLK.

In this mode, the `TIMER_CK` which drives counter's prescaler to count is equal to `CK_TIMER` which is from RCU module.

If TSCFGy[4:0] (y=0..2,6,9..12) in SYSCFG_TIMERxCFG(x=0,7) registers are setting to a nonzero value, the prescaler is clocked by other clock sources selected in the TSCFGy[4:0] (y=0..2,6,9..12) bit-field, more details will be introduced later. When the TSCFGy[4:0] (y=3,4,5,7) are setting to a nonzero value, the internal clock TIMER_CK is the counter prescaler driving clock source.

Figure 21-56. Normal mode, internal clock divided by 1



- TSCFG6[4:0] are setting to a nonzero value (external clock mode 0). External input pin is selected as timer clock source.

The TIMER_CLK, which drives counter's prescaler to count, can be triggered by the event of rising or falling edge on the external pin `TIMERx_CHn` ($n=0,1$). This mode can be selected by setting TSCFG6[4:0] to 0x5~0x7 and 0x9~0xE.

And, the counter prescaler can also be driven by rising edge on the internal trigger input pin ITI0~ITI4. This mode can be selected by setting TSCFG6[4:0] to 0x1~0x4.

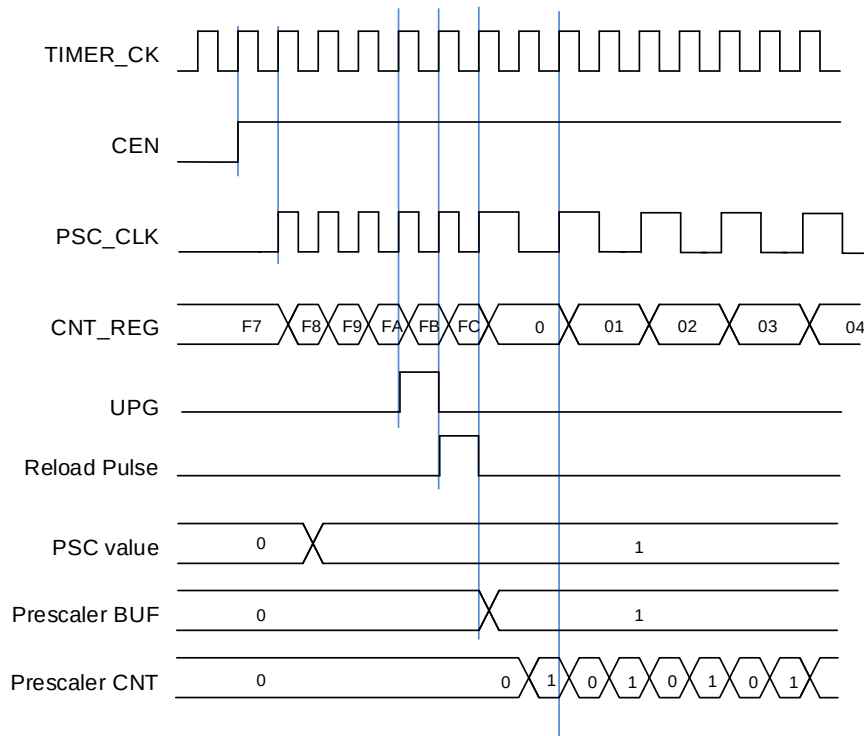
- SMC1= 1'b1 (external clock mode 1). External input ETI is selected as timer clock source.

The TIMER_CLK, which drives counter's prescaler to count, can be triggered by the event of rising or falling edge on the external pin ETI. This mode can be selected by setting the SMC1 bit in the `TIMERx_SMCFG` register to 1. The other way to select the ETI signal as the clock source is setting the TSCFG6[4:0] to 0x8. Note that the ETI signal is derived from the ETI pin sampled by a digital filter. When the ETI signal is selected as the clock source, the trigger controller including the edge detection circuitry will generate a clock pulse on each ETI signal rising edge to clock the counter prescaler.

Note: The ETI signal can be input from an external ETI pin or provide by on-chip peripherals, please refer to ETISEL[1:0] bits in `TIMERx_CTL0` register for more details.

Clock prescaler

The prescaler can divide the timer clock (TIMER_CLK) to a counter clock (PSC_CLK) by any factor ranging from 1 to 65536. It is controlled by prescaler register (`TIMERx_PSC`) which can be changed ongoing, but it is adopted at the next update event.

Figure 21-57. Counter timing diagram with prescaler division change from 1 to 2

Up counting mode

In this mode, the counter counts up continuously from 0 to the counter reload value, which is defined in the `TIMERx_CAR` register, in a count-up direction. Once the counter reaches the counter reload value, the counter restarts from 0. The update event is generated each time when counter overflows. The counting direction bit `DIR` in the `TIMERx_CTL0` register should be set to 0 for the up counting mode.

Whenever, if the update event software trigger is enabled by setting the `UPG` bit in the `TIMERx_SWEVG` register, the counter value will be initialized to 0 and an update event will be generated.

If the `UPDIS` bit in `TIMERx_CTL0` register is set, the update event is disabled.

When an update event occurs, all the registers (auto reload register, prescaler register) are updated.

[Figure 21-58. Timing chart of up counting mode, PSC=0/2](#) and [Figure 21-59. Timing chart of up counting, change `TIMERx\_CAR` ongoing](#) show some examples of the counter behavior for different clock prescaler factor when `TIMERx_CAR=0x99`.

Figure 21-58. Timing chart of up counting mode, PSC=0/2

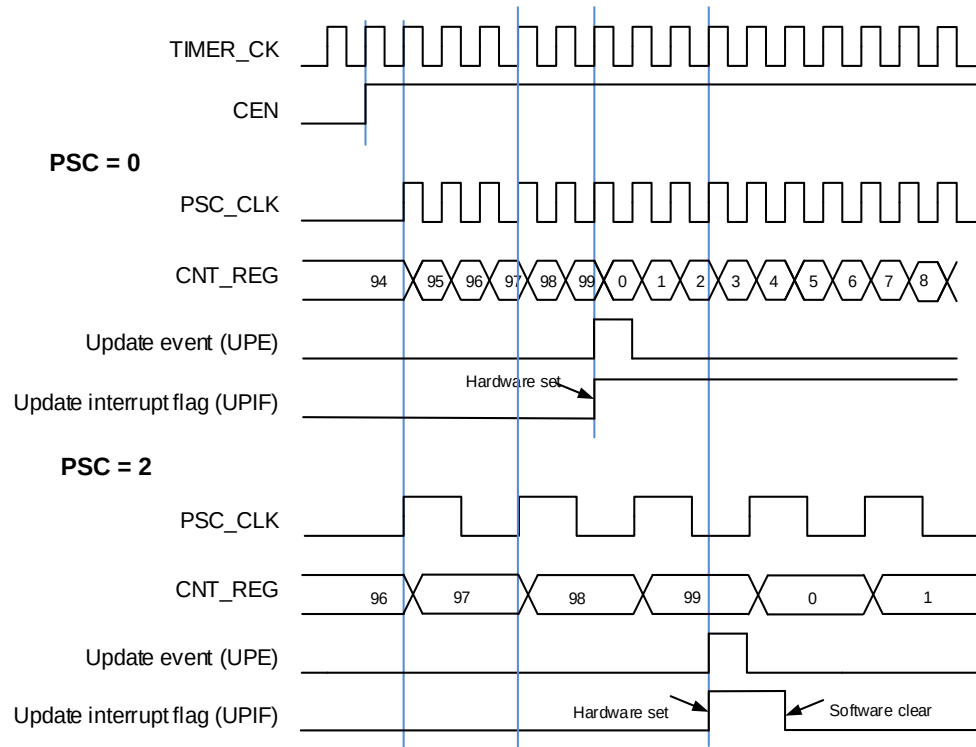
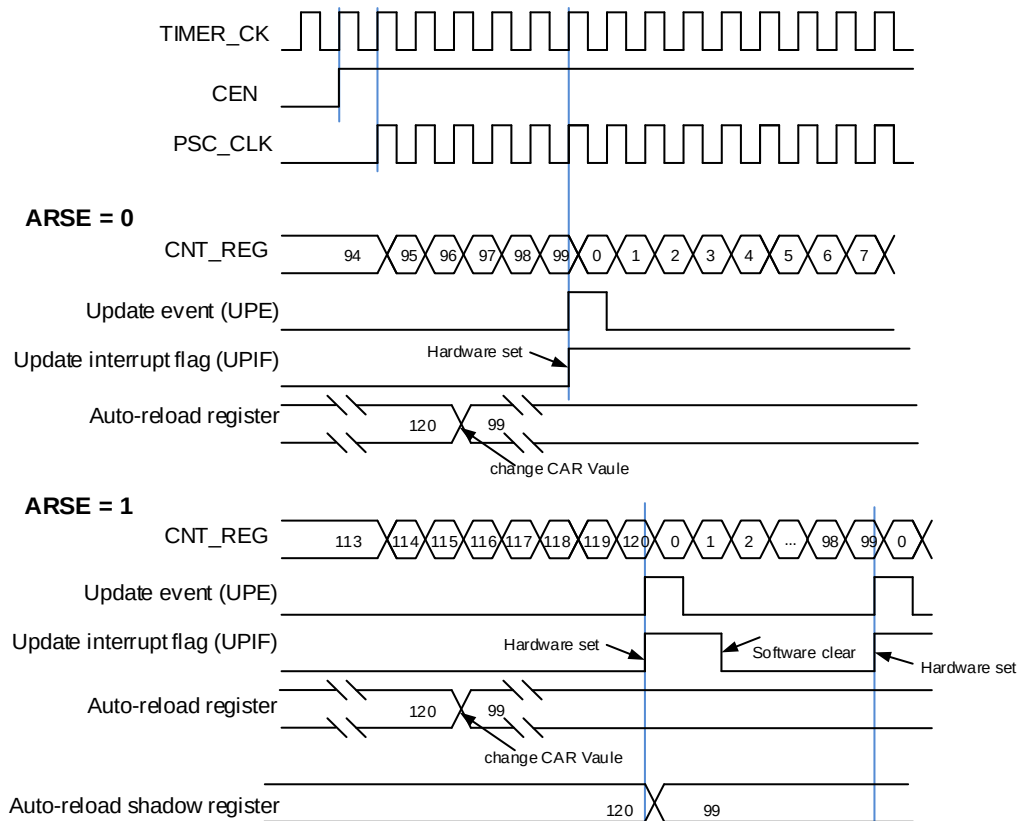


Figure 21-59. Timing chart of up counting, change TIMEx_CAR ongoing



Down counting mode

In this mode, the counter counts down continuously from the counter reload value, which is defined in the `TIMERx_CAR` register, in a count-down direction. Once the counter reaches 0, the counter restarts to count again from the counter reload value. The counting direction bit `DIR` in the `TIMERx_CTL0` register should be set to 1 for the down counting mode.

When the update event is set by the `UPG` bit in the `TIMERx_SWEVG` register, the counter value will be initialized to the counter reload value and an update event will be generated.

If the `UPDIS` bit in `TIMERx_CTL0` register is set, the update event is disabled.

When an update event occurs, all the registers (auto reload register, prescaler register) are updated.

[Figure 21-60. Timing chart of down counting mode, \$PSC=0/2\$](#) and [Figure 21-61. Timing chart of down counting mode, change `TIMERx\_CAR` ongoing](#) show some examples of the counter behavior for different clock frequencies when `TIMERx_CAR = 0x99`.

Figure 21-60. Timing chart of down counting mode, $PSC=0/2$

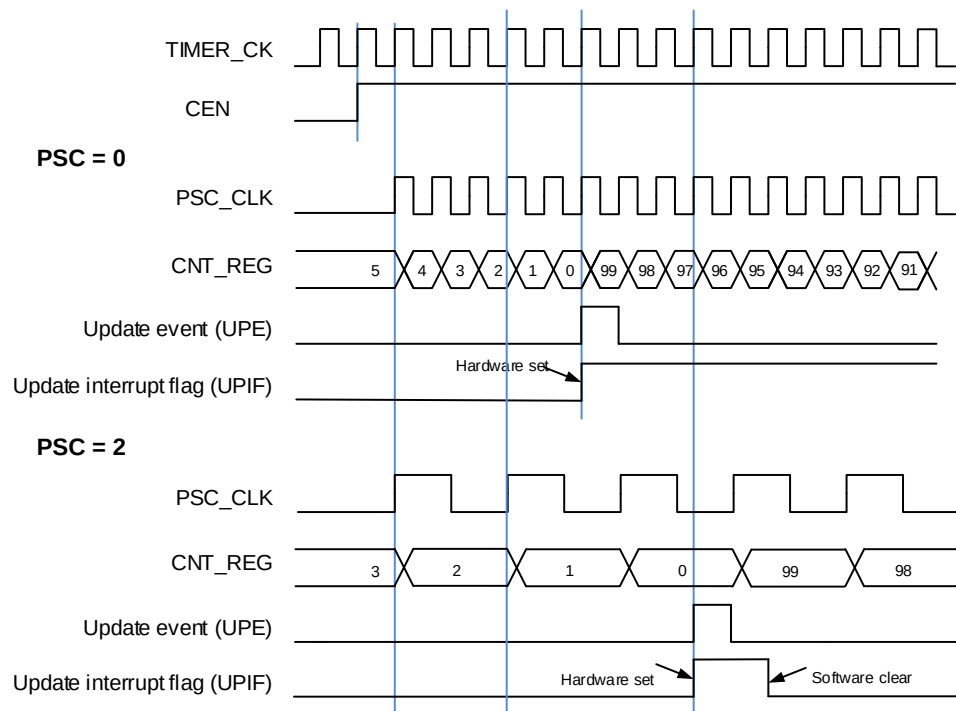
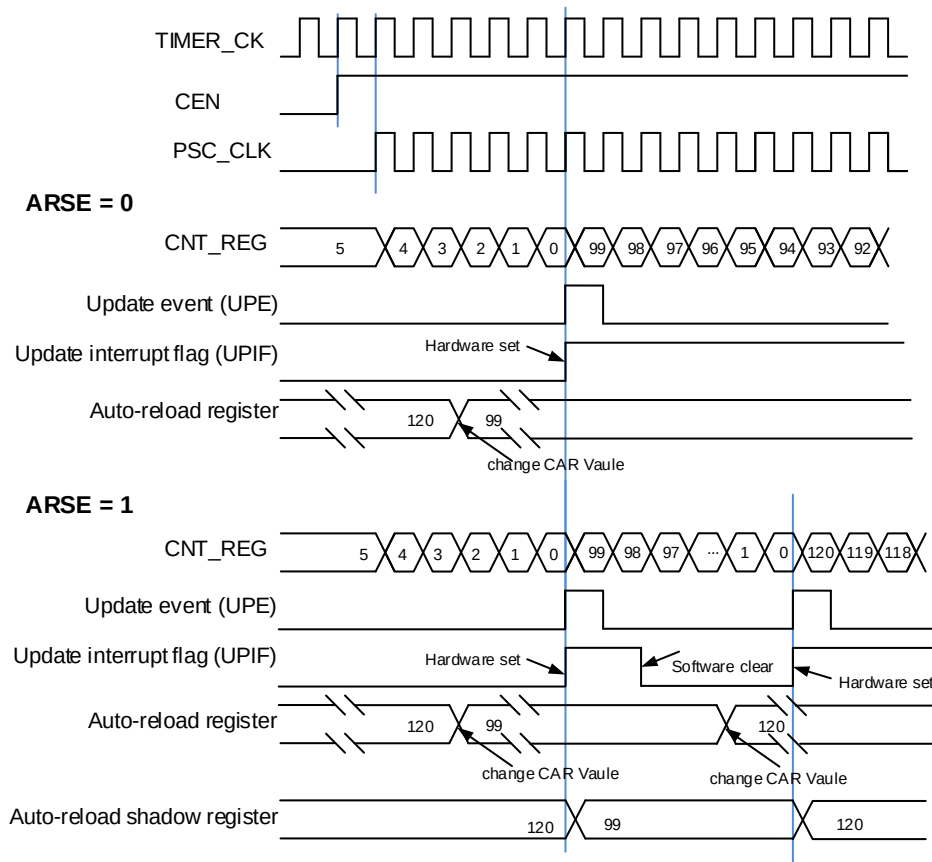


Figure 21-61. Timing chart of down counting mode, change `TIMERx_CAR` ongoing

Center-aligned counting mode

In the center-aligned counting mode, the counter counts up from 0 to the counter reload value and then counts down to 0 alternatively. The timer module generates an overflow event when the counter counts to (`TIMERx_CAR`-1) in the count-up direction and generates an underflow event when the counter counts to 1 in the count-down direction. The counting direction bit `DIR` in the `TIMERx_CTL0` register is read-only and indicates the counting direction when in the center-aligned counting mode. The counting direction is updated by hardware automatically.

Setting the `UPG` bit in the `TIMERx_SWEVG` register will initialize the counter value to 0 and generate an update event irrespective of whether the counter is counting up or down in the center-aligned counting mode.

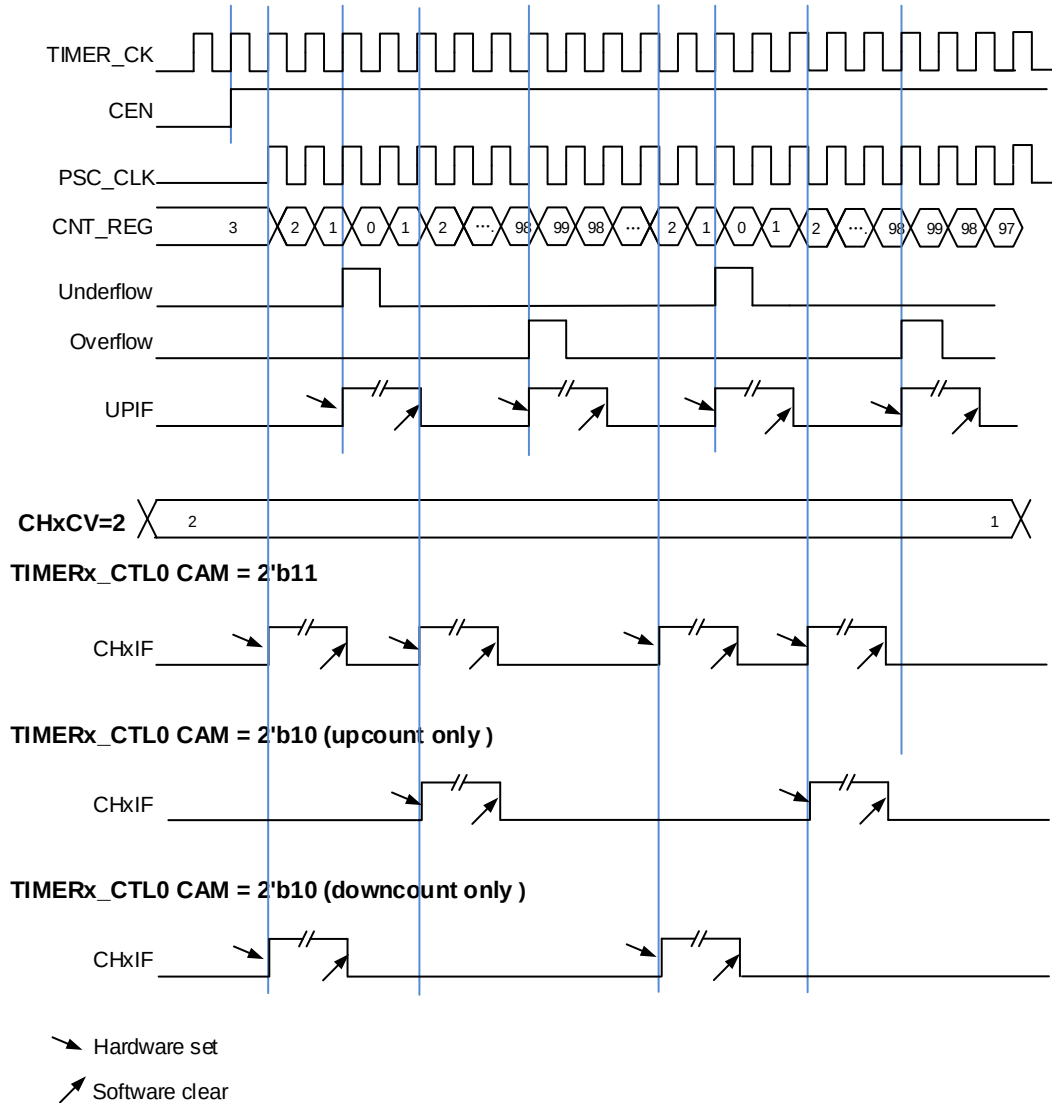
The `UPIF` bit in the `TIMERx_INTF` register will be set to 1 either when an underflow event or an overflow event occurs. While the `CHxIF` bit is associated with the value of `CAM` in `TIMERx_CTL0`. The details refer to [Figure 21-62. Timing chart of center-aligned counting mode](#).

If the `UPDIS` bit in the `TIMERx_CTL0` register is set, the update event is disabled.

When an update event occurs, all the registers (auto-reload register, prescaler register) are

updated. [Figure 21-62. Timing chart of center-aligned counting mode](#) shows the example of the counter behavior when $TIMERx\_CAR=0x99$, $TIMERx\_PSC=0x0$.

Figure 21-62. Timing chart of center-aligned counting mode



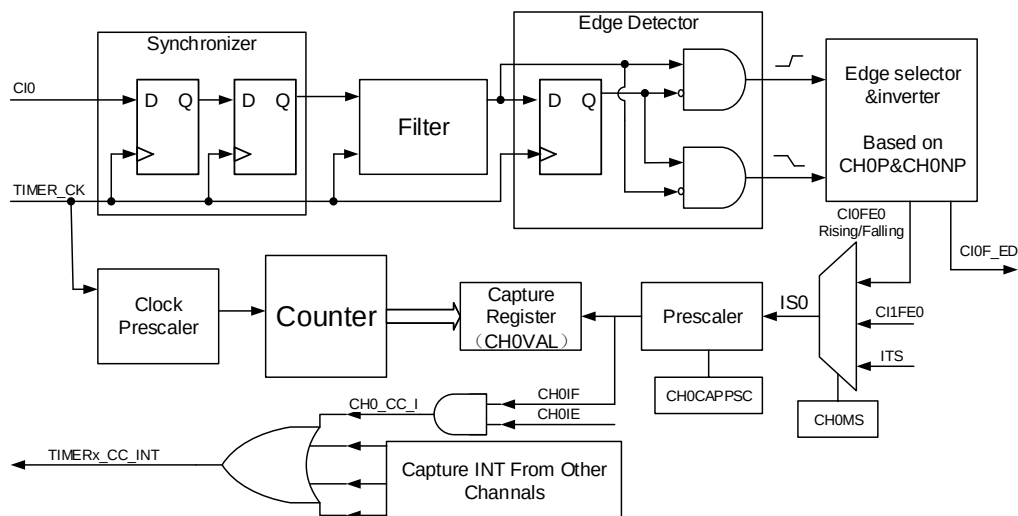
Capture/compare channels

The general level0 timer has four independent channels which can be used as capture inputs or compare match outputs. Each channel is built around a channel capture compare register including an input stage, channel controller and an output stage.

1) Input capture mode

Input capture mode allows the channel to perform measurements such as pulse timing, frequency, period, duty cycle and so on. The input stage consists of a digital filter, a channel polarity selection, edge detection and a channel prescaler. When a selected edge occurs on the channel input, the current value of the counter is captured into the $TIMERx\_CHxCV$ register, at the same time the $CHxIF$ bit is set and the channel interrupt is generated if it is

Figure 21-63. Input capture logic



So, the process can be divided into several steps as below:

Based on the input signal and quality of requested signal, configure compatible CHxCAPFLT.

Rising edge or falling edge, choose one by configuring CHxP and CHxNP bits.

As soon as selecting one input capture source by CHxMS, the channel must be set to input mode (CHxMS! =0x0) and TIMERx CHxCV cannot be written any more.

Enable the related interrupt to get the interrupt and DMA request.

Result: When the wanted input signal is captured, `TIMERx_CHxCV` will be set by counter's value and `CHxIF` is asserted. If the `CHxIF` is 1, the `CHxOF` will also be asserted. The interrupt and DMA request will be asserted or not based on the configuration of `CHxIE` and `CHxDEN`

in `TIMERx_DMAINTEN`.

Direct generation: A DMA request or interrupt is generated by setting `CHxG` directly.

The input capture mode can be also used for pulse width measurement from signals on the `TIMERx_CHx` pins. For example, PWM signal connects to `CI0` input. Select `CI0` as channel 0 capture signals by setting `CH0MS` to 3'b001 in the channel control register (`TIMERx_CHCTL0`) and set capture on rising edge. Select `CI0` as channel 1 capture signal by setting `CH1MS` to 3'b010 in the channel control register (`TIMERx_CHCTL0`) and set capture on falling edge. The counter is set to restart mode and is restarted on channel 0 rising edge. Then the `TIMERx_CH0CV` can measure the PWM period and the `TIMERx_CH1CV` can measure the PWM duty cycle.

2) Output compare mode

Figure 21-64. Output compare logic (x=0,1,2,3)

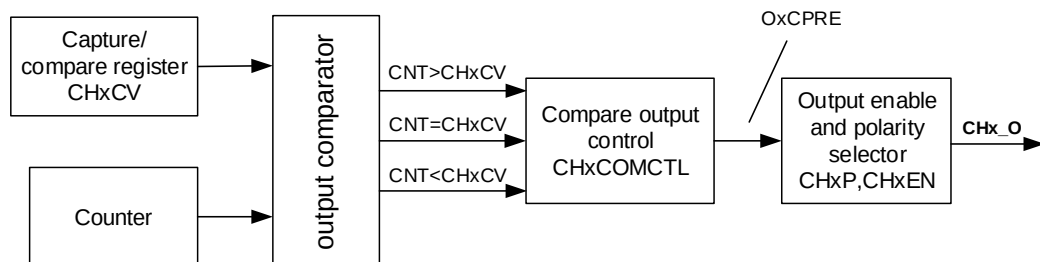


Figure 21-64. Output compare logic (x=0,1,2,3) shows the logic circuit of output compare mode. The relationship between the channel output signal `CHx_O` and the `OxCPRE` signal (more details refer to [Channel output prepare signal](#)) is described as below: The active level of `OxCPRE` is high, the output level of `CHx_O` depends on `OxCPRE` signal, `CHxP` bit and `CH0P` bit (please refer to the `TIMERx_CHCTL2` register for more details). For example, configure `CHxP=0` (the active level of `CHx_O` is high, the same as `OxCPRE`), `CHxEN=1` (the output of `CHx_O` is enabled),

If the output of `OxCPRE` is active(high) level, the output of `CHx_O` is active(high) level;

If the output of `OxCPRE` is inactive(low) level, the output of `CHx_O` is active(low) level.

In output compare mode, the `TIMERx` can generate timed pulses with programmable position, polarity, duration and frequency. When the counter matches the value in the `TIMERx_CHxCV` register of an output compare channel, the channel (n) output can be set, cleared, or toggled based on `CHxCOMCTL`. When the counter reaches the value in the `TIMERx_CHxCV` register, the `CHxIF` bit will be set and the channel (n) interrupt is generated if `CHxIE` = 1. And the DMA request will be asserted, if `CHxDEN`=1.

So, the process can be divided into several steps as below:

Step1: Clock configuration. Such as clock source, clock prescaler and so on.

Step2: Compare mode configuration.

- Set the shadow enable mode by CHxCOMSEN.
- Set the output mode (set/clear/toggle) by CHxCOMCTL.
- Select the active polarity by CHxP.
- Enable the output by CHxEN.

Step3: Interrupt/DMA-request enables configuration by CHxIE/CHxDEN.

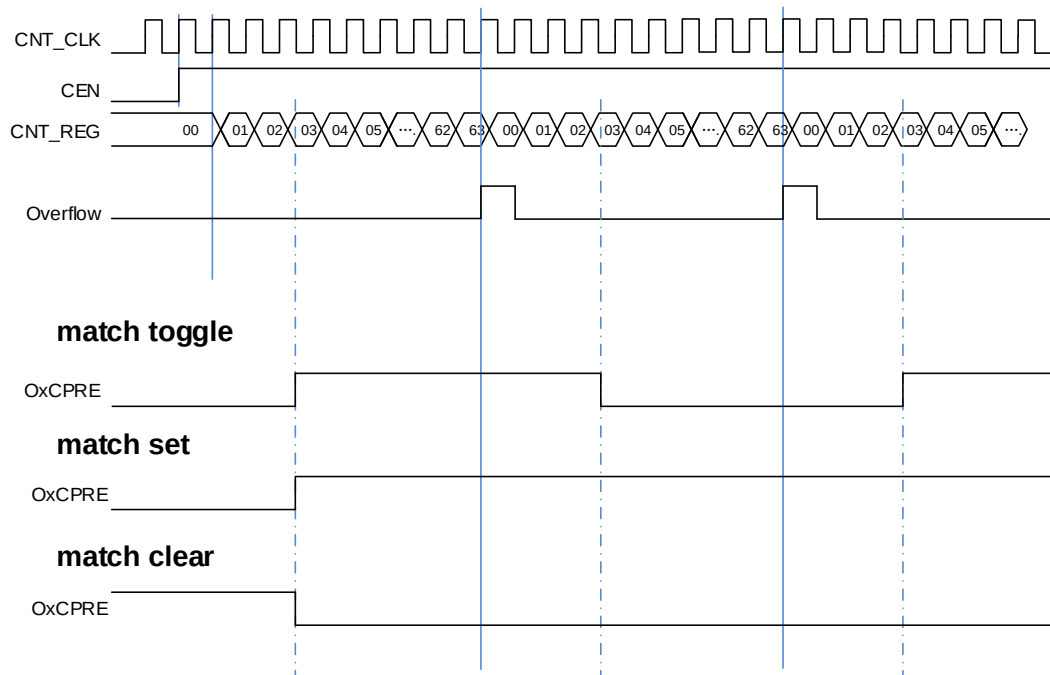
Step4: Compare output timing configuration by TIMERx_CAR and TIMERx_CHxCV.

The TIMERx_CHxCV can be changed ongoing to meet the expected waveform.

Step5: Start the counter by configuring CEN to 1.

Figure 21-65. Output-compare under three modes shows the three compare modes toggle/set/clear. CAR=0x63, CHxVAL=0x3

Figure 21-65. Output-compare under three modes



PWM mode

In the PWM output mode (by setting the CHxCOMCTL bit to 4'b0110 (PWM mode 0) or to 4'b0111 (PWM mode 1)), the channel can generate PWM waveform according to the TIMERx_CAR registers and TIMERx_CHxCV registers.

Based on the counter mode, PWM can also be divided into EAPWM (Edge-aligned PWM) and CAPWM (Center-aligned PWM).

The EAPWM's period is determined by TIMERx_CAR and the duty cycle is determined by TIMERx_CHxCV. [Figure 21-66. Timing chart of EAPWM](#) shows the EAPWM output and interrupts waveform.

The CAPWM period is determined by $2 \times \text{TIMERx\_CAR}$, and duty cycle is determined by $2 \times \text{TIMERx\_CHxCV}$. [Figure 21-67. Timing chart of CAPWM](#) shows the CAPWM output and interrupts waveform.

In up counting mode, if the value of TIMERx_CHxCV is greater than the value of TIMERx_CAR , the output will be always active in PWM mode 0 ($\text{CHxCOMCTL}=4'b0110$). And if the value of TIMERx_CHxCV is greater than the value of TIMERx_CAR , the output will be always inactive in PWM mode 1 ($\text{CHxCOMCTL}=4'b0111$).

Figure 21-66. Timing chart of EAPWM

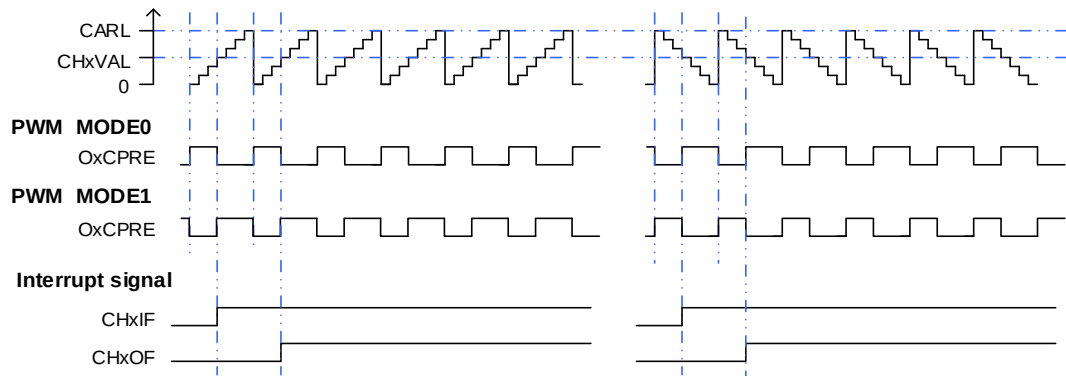
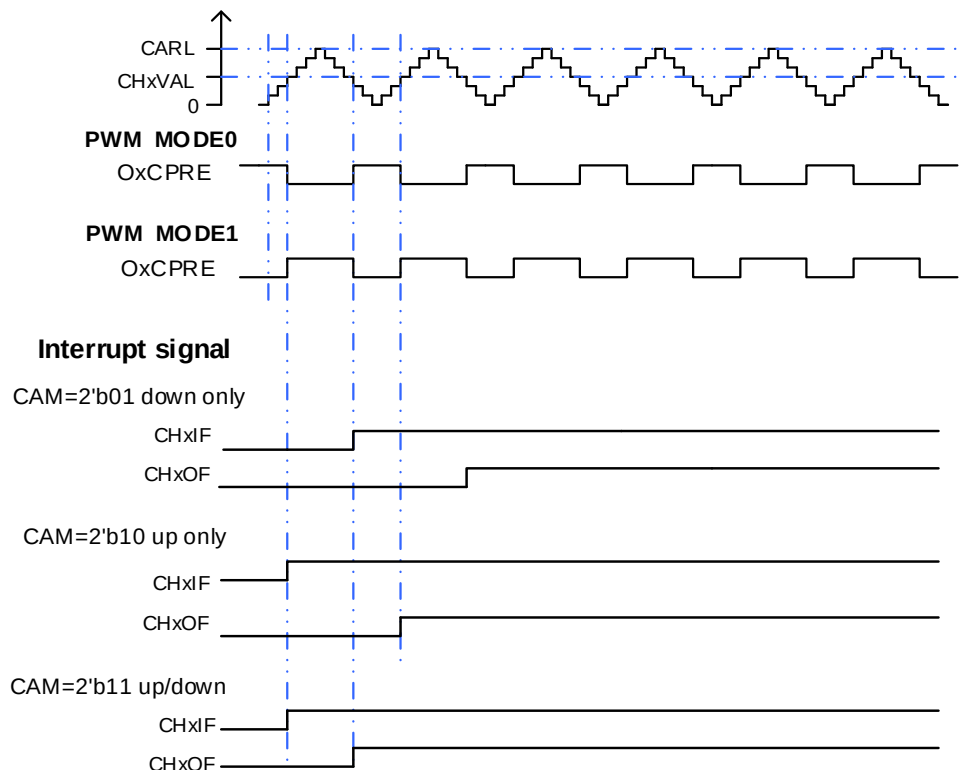


Figure 21-67. Timing chart of CAPWM



Channel output prepare signal

As is shown in [Figure 21-64. Output compare logic \(x=0,1,2,3\)](#), when TIMERx is configured in compare match output mode a middle signal which is OxCPRE signal (Channel x output

prepare signal) will be generated before the channel outputs signal. The OxCPRE signal type is defined by configuring the CHxCOMCTL bit.

The OxCPRE signal has several types of output function. The OxCPRE signal type is defined by configuring the CHxCOMCTL bit. These include keeping the original level by configuring the CHxCOMCTL field to 0x00, setting to high by configuring the CHxCOMCTL field to 0x01, setting to low by configuring the CHxCOMCTL field to 0x02 or toggling signal by configuring the CHxCOMCTL field to 0x03 when the counter value matches the content of the TIMERx_CHxCV register.

The PWM mode 0/PWM mode 1 output is another output type of OxCPRE which is setup by configuring the CHxCOMCTL field to 0x06/0x07. In these modes, the OxCPRE signal level is changed according to the counting direction and the relationship between the counter value and the TIMERx_CHxCV content. Refer to the definition of relative bit for more details.

Another special function of the OxCPRE signal is a forced output which can be achieved by configuring the CHxCOMCTL field to 0x04/0x05. The output can be forced to an inactive/active level irrespective of the comparison condition between the values of the counter and the TIMERx_CHxCV.

Configure the CHxCOMCEN bit to 1 in the TIMERx_CHCTL0 register, the OxCPRE signal can be forced to 0 when the ETIFP signal derived from the external ETI pin is set to a high level. The OxCPRE signal will not return to its active level until the next transition to the active state.

Quadrature decoder

Refer to [Advanced timer \(TIMERx, x=0, 7\)Quadrature decoder](#).

Non-quadrature decoder

Refer to [Advanced timer \(TIMERx, x=0, 7\)Non-quadrature decoder](#).

Hall sensor function

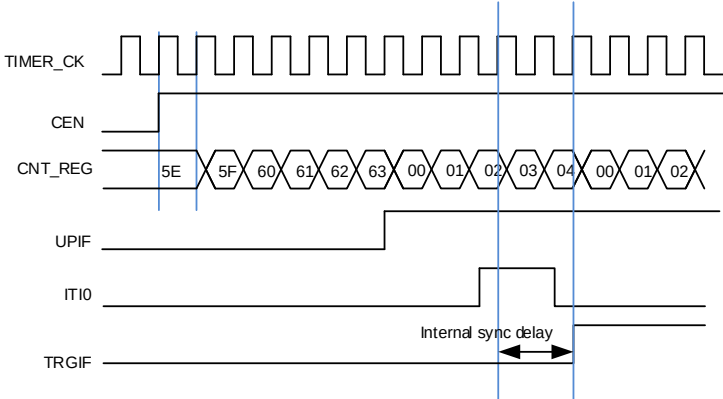
Refer to [Advanced timer \(TIMERx, x=0, 7\)Hall sensor function](#).

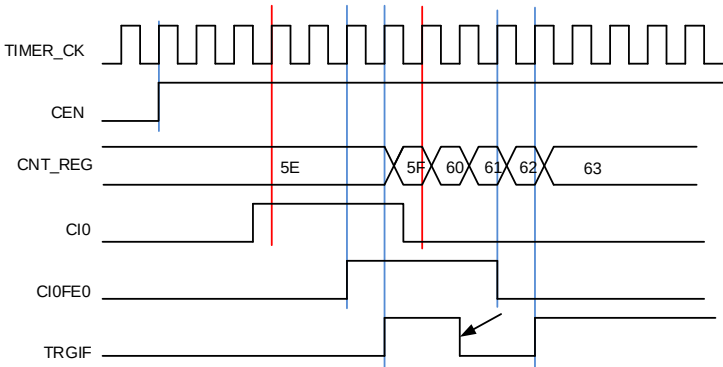
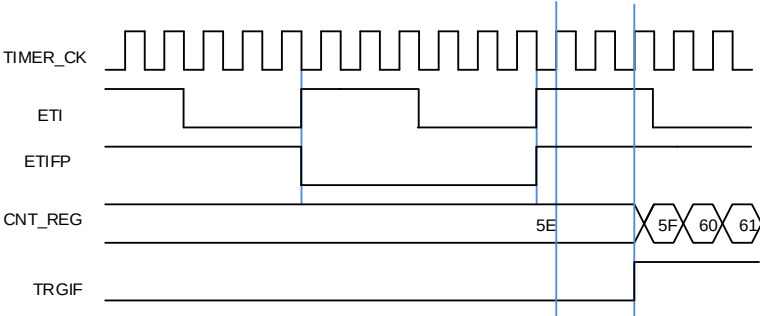
Master-slave management

The TIMERx can be synchronized with a trigger in several modes including restart mode, pause mode and event mode and so on, which is selected by the TSCFGy[4:0] (y=3..7) in SYSCFG_TIMERxCFG(x=1,2).

Table 21-11. Examples of slave mode

	Mode Selection	Source Selection	Polarity Selection	Filter and Prescaler
LIST	TSCFGy[4:0] y=3: restart mode	TSCFGy[4:0] 00000: disable	If CI0FE0 or CI1FE1 is selected as the trigger	For the ITIx, no filter and prescaler can be

	Mode Selection	Source Selection	Polarity Selection	Filter and Prescaler
	y=4: pause mode y=5: event mode y=6: external clock mode 0 y=7: restart + event mode	00001: ITI0 00010: ITI1 00011: ITI2 00100: ITI3 00101: CI0F_ED 00110: CI0FE0 00111: CI1FE1 01000: ETIFP	source, configure the CHxP and CHxNP for the polarity selection and inversion. If ETIFP (the filtered output of external trigger input ETI) is selected as the trigger source, configure the ETP for polarity selection and inversion.	used. For the CIx, filter can be used by configuring CHxCAPFLT, no prescaler can be used. For the ETIFP, filter can be used by configuring ETFC and prescaler can be used by configuring ETPSC.
Exam1	Restart mode The counter will be cleared and restart when a rising edge of trigger input comes.	TSCFG3[4:0] 5'b00001, ITI0 is selected.	For ITI0, no polarity selector can be used.	For the ITI0, no filter and prescaler can be used.
	Figure 21-68. Restart mode 			
Exam2	Pause mode The counter will be paused when the trigger input is low, and it will start when the trigger input is high.	TSCFG4[4:0] =5'b00110, CI0FE0 is selected.	TI0S = 0 (Non-xor) [CH0NP=0, CH0P=0] CI0FE0 does not invert. The capture event will occur on the rising edge only.	Filter is bypassed in this example.

	Mode Selection	Source Selection	Polarity Selection	Filter and Prescaler
	Figure 21-69. Pause mode 			
Exam3	Event mode The counter will start to count when a rising edge of trigger input comes.	TSCFG5[4:0] = 5'b01000, ETIFP is selected.	ETP = 0, the polarity of ETI does not change.	ETPSC = 1, ETI is divided by 2. ETFC = 0, ETI does not filter.
	Figure 21-70. Event mode 			
Exam4	Restart + event mode The counter is reinitialized and started when a rising edge of trigger input comes.			

Single pulse mode

Single pulse mode is opposite to the repetitive mode, which can be enabled by setting SPM in TIMEx_CTL0. If SPM is set, the counter will be cleared and stopped automatically when the next update event occurs. In order to get a pulse waveform, the TIMEx is configured to PWM mode or compare mode by CHxCOMCTL bit.

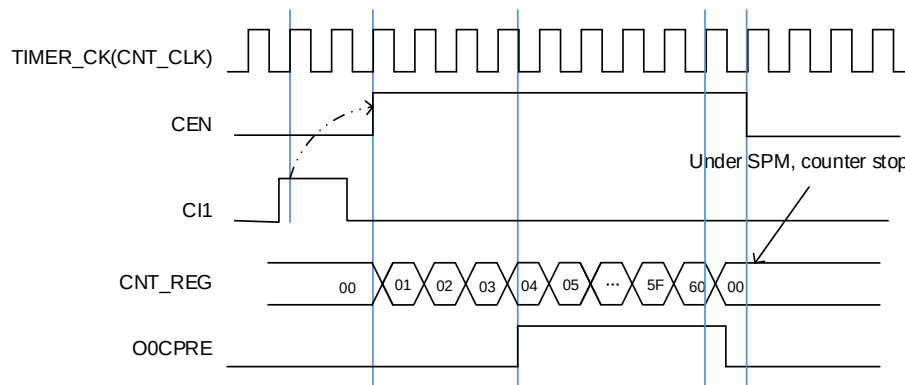
Once the timer is set to the single pulse mode, it is not necessary to configure the timer enable bit CEN in the TIMEx_CTL0 register to 1 to enable the counter. Setting the CEN bit to 1 or a trigger signal edge can generate a pulse and then keep the CEN bit at a high state until the update event occurs or the CEN bit is written to 0 by software. If the CEN bit is cleared to 0 by software, the counter will be stopped and its value will be held. If the CEN bit is automatically cleared to 0 by a hardware update event, the counter will be reinitialized.

In the single pulse mode, the active edge of trigger which sets the CEN bit to 1 will enable the

counter. However, there exists several clock delays to perform the comparison result between the counter value and the `TIMERx_CHxCV` value. After a trigger rising occurs in the single pulse mode, the `OxCPRE` signal will immediately be forced to the state which the `OxCPRE` signal will change to, as the compare match event occurs without taking the comparison result into account.

Single pulse mode is also applicable to composite PWM mode (`CHxCPWMEN` = 1'b1 and `CHxMS[2:0]` = 3'b000).

Figure 21-71. Single pulse mode `TIMERx_CHxCV` = 0x04, `TIMERx_CAR`=0x60



Timers interconnection

Please refer to [Advanced timer \(`TIMERx`, `x=0, 7`\)Timers interconnection](#).

Timer DMA mode

Timer DMA mode is the function that configures timer's register by DMA module. The relative registers are `TIMERx_DMACFG` and `TIMERx_DMATB`. Corresponding DMA request bit should be asserted to enable DMA request for internal interrupt event. `TIMERx` will send a request to DMA when the interrupt event occurs. DMA is configured to M2P (memory to peripheral) mode and the address of `TIMERx_DMATB` is configured to PADDR (peripheral base address), then DMA will access the `TIMERx_DMATB`. In fact, `TIMERx_DMATB` register is only a buffer, timer will map the `TIMERx_DMATB` to an internal register, appointed by the field of `DMATA` in `TIMERx_DMACFG`. If the field of `DMATC` in `TIMERx_DMACFG` is 0 (1 transfer), the timer sends only one DMA request. While if `TIMERx_DMATC` is not 0, such as 3 (4 transfers), then timer will send 3 more requests to DMA, and DMA will access timer's registers `DMATA+0x4`, `DMATA+0x8` and `DMATA+0xC` at the next 3 accesses to `TIMERx_DMATB`. In a word, one-time DMA internal interrupt event asserts, (`DMATC`+1) times request will be sent by `TIMERx`.

If one more DMA request event occurs, `TIMERx` will repeat the process above.

UPIF bit backup

The UPIF bit backup function is enabled by setting UPIFBUEN in the TIMERx_CTL0 register. The UPIF and UPIFBU bits are fully synchronized and without latency.

By using this function, the UPIF bit in the TIMERx_INTF register will be backed up to the UPIFBU bit in the TIMERx_CNT register. This can avoid conflicts when reading the counter and interrupt processing.

Timer debug mode

When the Cortex®-M33 is halted, and the TIMERx_HOLD configuration bit in DBG_CTL register set to 1, the TIMERx counter stops.

21.2.5 Registers definition (TIMERx, x=1,2)

TIMER1 base address: 0x4000 0000

TIMER2 base address: 0x4000 0400

Control register 0 (TIMERx_CTL0)

Address offset: 0x00

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved						ETISEL[1:0]		Reserved							
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				UPIFBUE	Reserved	CKDIV[1:0]		ARSE	CAM[1:0]		DIR	SPM	UPS	UPDIS	CEN
				N											
				rw	rw		rw	rw		rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:26	Reserved	Must be kept at reset value.
25:24	ETISEL[1:0]	Timer external trigger input signal select 00: Select TIMER_ETI0 pin from GPIO. 01: Select TIMER_ETI1 pin from GPIO. 10: Select TIMER_ETI2 pin from GPIO. 11: Select TIMER_ETI3 pin from GPIO.
23:12	Reserved	Must be kept at reset value.
11	UPIFBUE	UPIF bit backup enable 0: Backup disable. UPIF bit is not backed up to UPIFBU bit in TIMERx_CNT register. 1: Backup enabled. UPIF bit is backed up to UPIFBU bit in TIMERx_CNT register.
10	Reserved	Must be kept at reset value.
9:8	CKDIV[1:0]	Clock division The CKDIV bits can be configured by software to specify division ratio between CK_TIMER (the timer clock) and DTS (the dead time and sampling clock) which is used for the dead time generator and the digital filter. 00: $f_{DTS} = f_{CK\_TIMER}$ 01: $f_{DTS} = f_{CK\_TIMER} / 2$ 10: $f_{DTS} = f_{CK\_TIMER} / 4$ 11: Reserved
7	ARSE	Auto-reload shadow enable 0: The shadow register for TIMERx_CAR register is disabled

		1: The shadow register for TIMEx_CAR register is enabled
6:5	CAM[1:0]	Counter align mode selection 00: No center-aligned mode (edge-aligned mode). The direction of the counter is specified by the DIR bit. 01: Center-aligned and counting down assert mode. The counter counts in center-aligned mode and channel is configured in output mode (CHxMS = 2'b00 in TIMEx_CHCTL0 register). Only when the counter is counting down, compare interrupt flag of channels can be set. 10: Center-aligned and counting up assert mode. The counter counts in center-aligned mode and channel is configured in output mode (CHxMS = 2'b00 in TIMEx_CHCTL0 register). Only when the counter is counting up, compare interrupt flag of channels can be set. 11: Center-aligned and counting up/down assert mode. The counter counts in center-aligned mode and channel is configured in output mode (CHxMS = 2'b00 in TIMEx_CHCTL0 register). Both when the counter is counting up and counting down, compare interrupt flag of channels can be set. After the counter is enabled, these bits cannot be switched from 0x00 to non 0x00.
4	DIR	Direction 0: Count up 1: Count down This bit is read only when the timer is configured in center-aligned mode or decoder mode.
3	SPM	Single pulse mode 0: Single pulse mode is disabled. Counter continues after an update event. 1: Single pulse mode is enabled. The CEN bit is cleared by hardware and the counter stops at next update event.
2	UPS	Update source This bit is used to select the update event sources by software. 0: Any of the following events generates an update interrupt or a DMA request: - The UPG bit is set. - The counter generates an overflow or underflow event. - The slave mode controller generates an update event. 1: Only counter overflow/underflow generates an update interrupt or a DMA request.
1	UPDIS	Update disable This bit is used to enable or disable the update event generation. 0: Update event enable. The update event is generated and the buffered registers are loaded with their preloaded values when one of the following events occurs: - The UPG bit is set. - The counter generates an overflow or underflow event. - The slave mode controller generates an update event. 1: Update event disable. The buffered registers keep their value, while the counter

and the prescaler are reinitialized if the UG bit is set or the slave mode controller generates a hardware reset event.

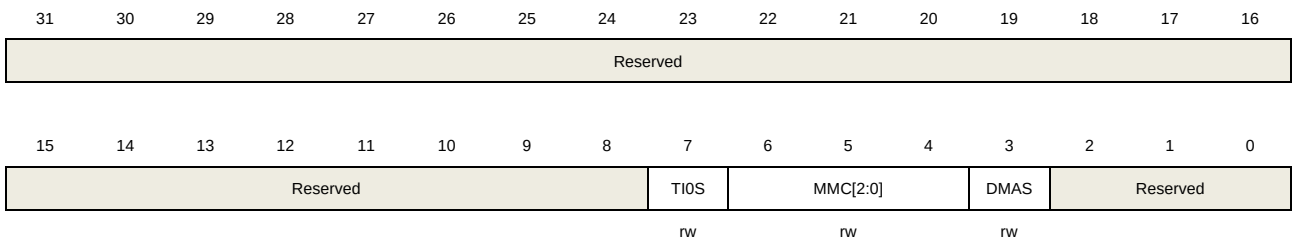
0	CEN	Counter enable 0: Counter disable 1: Counter enable The CEN bit must be set by software when timer works in external clock mode, pause mode or decoder mode. While in event mode, the hardware can set the CEN bit automatically.

Control register 1 (TIMERx_CTL1)

Address offset: 0x04

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7	TI0S	Channel 0 trigger input selection 0: The TIMERx_CH0 pin input is selected as channel 0 trigger input. 1: The result of combinational XOR of TIMERx_CH0, TIMERx_CH1 and TIMERx_CH2 pins is selected as channel 0 trigger input.
6:4	MMC[2:0]	Master mode control These bits control the selection of TRGO signal, which is sent by master timer to slave timer for synchronization function. 000: Reset. When the UPG bit in the TIMERx_SWEVG register is set or a reset is generated by the slave mode controller, a TRGO pulse occurs. And in the latter case, the signal on TRGO is delayed compared to the actual reset. 001: Enable. This mode is used to start several timers at the same time or control a slave timer to be enabled in a period. In this mode, the master mode controller selects the counter enable signal as TRGO. The counter enable signal is set when CEN control bit is set or the trigger input in pause mode is high. There is a delay between the trigger input in pause mode and the TRGO output, except if the master-slave mode is selected. 0010: Update. In this mode, the master mode controller selects the update event as TRGO.

011: Capture/compare pulse. In this mode, the master mode controller generates a TRGO pulse when a capture or a compare match occurs in channel 0.

100: Compare. In this mode, the master mode controller selects the O0CPRE signal as TRGO.

101: Compare. In this mode, the master mode controller selects the O1CPRE signal as TRGO.

110: Compare. In this mode, the master mode controller selects the O2CPRE signal as TRGO.

111: Compare. In this mode, the master mode controller selects the O3CPRE signal as TRGO.

3	DMAS	DMA request source selection 0: DMA request of CHx is sent when capture/compare event occurs. 1: DMA request of channel CHx is sent when update event occurs.
2:0	Reserved	Must be kept at reset value.

Slave mode configuration register (TIMERx_SMCFG)

Address offset: 0x08

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETP	SMC1	ETPSC[1:0]		ETFC[3:0]				MSM	Reserved						
rw	rw	rw		rw				rw							

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15	ETP	External trigger polarity This bit specifies the polarity of ETI signal. 0: ETI is active at high level or rising edge. 1: ETI is active at low level or falling edge.
14	SMC1	Part of slave mode controller is used to enable external clock mode 1 In external clock mode 1, the counter is clocked by any active edge of the ETIFP signal. 0: External clock mode 1 disabled 1: External clock mode 1 enabled It is possible to simultaneously use external clock mode 1 with the restart mode, pause mode or event mode. But the TSCFGy[4:0](y=3,4,5) bits must not be 5b'01000 in this case.

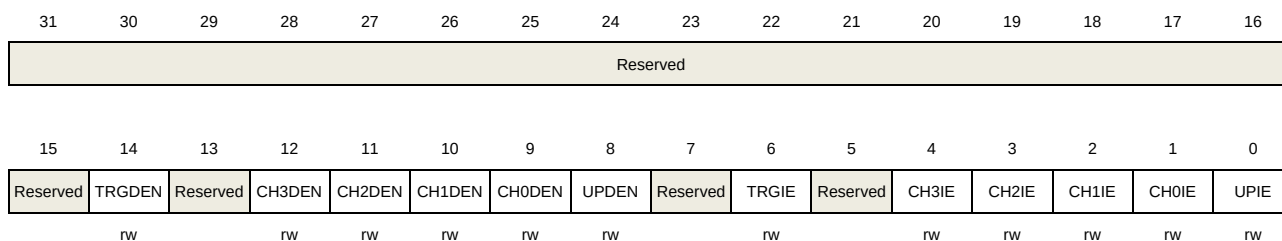
		The external clock input will be ETIFP if external clock mode 0 and external clock mode 1 are enabled at the same time. Note: External clock mode 0 enable is in TSCFG6[4:0] bit-field in SYSCFG_TIMERxCFG1 register.
13:12	ETPSC[1:0]	External trigger prescaler The frequency of external trigger signal ETIFP must not be higher than 1/4 of TIMER_CK frequency. When the frequency of external trigger signal is high, the prescaler can be enabled to reduce ETIFP frequency. 00: Prescaler disabled 01: ETIFP frequency divided by 2 10: ETIFP frequency divided by 4 11: ETIFP frequency divided by 8
11:8	ETFC[3:0]	External trigger filter control An event counter is used in the digital filter, in which a transition on the output occurs after N input events. This bit-field specifies the frequency used to sample ETIFP signal and the length of the digital filter applied to ETIFP. 0000: Filter disabled. $f_{SAMP} = f_{DTS}$, $N=1$. 0001: $f_{SAMP} = f_{CK\_TIMER}$, $N=2$. 0010: $f_{SAMP} = f_{CK\_TIMER}$, $N=4$. 0011: $f_{SAMP} = f_{CK\_TIMER}$, $N=8$. 0100: $f_{SAMP} = f_{DTS}/2$, $N=6$. 0101: $f_{SAMP} = f_{DTS}/2$, $N=8$. 0110: $f_{SAMP} = f_{DTS}/4$, $N=6$. 0111: $f_{SAMP} = f_{DTS}/4$, $N=8$. 1000: $f_{SAMP} = f_{DTS}/8$, $N=6$. 1001: $f_{SAMP} = f_{DTS}/8$, $N=8$. 1010: $f_{SAMP} = f_{DTS}/16$, $N=5$. 1011: $f_{SAMP} = f_{DTS}/16$, $N=6$. 1100: $f_{SAMP} = f_{DTS}/16$, $N=8$. 1101: $f_{SAMP} = f_{DTS}/32$, $N=5$. 1110: $f_{SAMP} = f_{DTS}/32$, $N=6$. 1111: $f_{SAMP} = f_{DTS}/32$, $N=8$.
7	MSM	Master-slave mode This bit can be used to synchronize the selected timers to begin counting at the same time. The TRGI is used as the start event, and through TRGO, timers are connected. 0: Master-slave mode disabled 1: Master-slave mode enabled
6:0	Reserved	Must be kept at reset value.

DMA and interrupt enable register (TIMERx_DMAINTEN)

Address offset: 0x0C

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



Bits	Fields	Descriptions
31:15	Reserved	Must be kept at reset value.
14	TRGDEN	Trigger DMA request enable 0: Disabled 1: Enabled
13	Reserved	Must be kept at reset value.
12	CH3DEN	Channel 3 capture/compare DMA request enable 0: Disabled 1: Enabled
11	CH2DEN	Channel 2 capture/compare DMA request enable 0: Disabled 1: Enabled
10	CH1DEN	Channel 1 capture/compare DMA request enable 0: Disabled 1: Enabled
9	CH0DEN	Channel 0 capture/compare DMA request enable 0: Disabled 1: Enabled
8	UPDEN	Update DMA request enable 0: Disabled 1: Enabled
7	Reserved	Must be kept at reset value.
6	TRGIE	Trigger interrupt enable 0: Disabled 1: Enabled
5	Reserved	Must be kept at reset value.

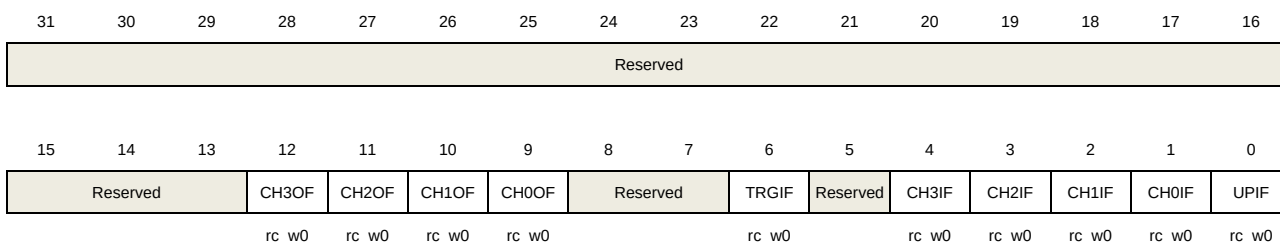
4	CH3IE	Channel 3 capture/compare interrupt enable 0: Disabled 1: Enabled
3	CH2IE	Channel 2 capture/compare interrupt enable 0: Disabled 1: Enabled
2	CH1IE	Channel 1 capture/compare interrupt enable 0: Disabled 1: Enabled
1	CH0IE	Channel 0 capture/compare interrupt enable 0: Disabled 1: Enabled
0	UPIE	Update interrupt enable 0: Disabled 1: Enabled

Interrupt flag register (TIMERx_INTF)

Address offset: 0x10

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



Bits	Fields	Descriptions
31:13	Reserved	Must be kept at reset value.
12	CH3OF	Channel 3 over capture flag Refer to CH0OF description
11	CH2OF	Channel 2 over capture flag Refer to CH0OF description
10	CH1OF	Channel 1 over capture flag Refer to CH0OF description
9	CH0OF	Channel 0 over capture flag When channel 0 is configured in input mode, this flag is set by hardware when a capture event occurs while CH0IF flag has already been set. This flag is cleared by

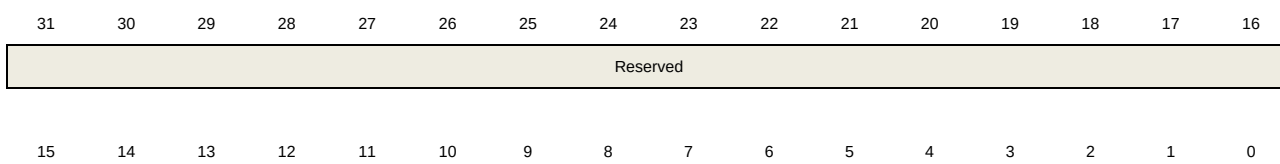
		software. 0: No over capture interrupt occurred 1: Over capture interrupt occurred
8:7	Reserved	Must be kept at reset value.
6	TRGIF	Trigger interrupt flag This flag is set by hardware on trigger event and cleared by software. When the slave mode controller is enabled in all modes but pause mode, an active edge of trigger input generates a trigger event. When the slave mode controller is enabled in pause mode, either edge of the trigger input can generate a trigger event. 0: No trigger event occurred 1: Trigger interrupt occurred
5	Reserved	Must be kept at reset value.
4	CH3IF	Channel 3 capture/compare interrupt flag Refer to CH0IF description
3	CH2IF	Channel 2 capture/compare interrupt flag Refer to CH0IF description
2	CH1IF	Channel 1 capture/compare interrupt flag Refer to CH0IF description
1	CH0IF	Channel 0 capture/compare interrupt flag This flag is set by hardware and cleared by software. If channel 0 is in input mode, this flag is set when a capture event occurs. If channel 0 is in output mode, this flag is set when a compare event occurs. If channel 0 is set to input mode, this bit will be reset by reading TIMEx_CH0CV. 0: No channel 0 interrupt occurred 1: Channel 0 interrupt occurred
0	UPIF	Update interrupt flag This bit is set by hardware when an update event occurs and cleared by software. 0: No update interrupt occurred 1: Update interrupt occurred

Software event generation register (TIMEx_SWEVG)

Address offset: 0x14

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



Reserved	TRGG	Reserved	CH3G	CH2G	CH1G	CH0G	UPG
	W		W	W	W	W	W

Bits	Fields	Descriptions
31:7	Reserved	Must be kept at reset value.
6	TRGG	Trigger event generation This bit is set by software and cleared by hardware automatically. When this bit is set, the TRGIF flag in TIMERx_INTF register will be set, related interrupt or DMA transfer can occur if enabled. 0: No generate a trigger event 1: Generate a trigger event
5	Reserved	Must be kept at reset value.
4	CH3G	Channel 3 capture or compare event generation Refer to CH0G description
3	CH2G	Channel 2 capture or compare event generation Refer to CH0G description
2	CH1G	Channel 1 capture or compare event generation Refer to CH0G description
1	CH0G	Channel 0 capture or compare event generation This bit is set by software to generate a capture or compare event in channel 0, it is automatically cleared by hardware. When this bit is set, the CH0IF flag will be set, and the corresponding interrupt or DMA request will be sent if enabled. In addition, if channel 0 is configured in input mode, the current value of the counter is captured to TIMERx_CH0CV register, and the CH0OF flag is set if the CH0IF flag has been set. 0: No generate a channel 0 capture or compare event 1: Generate a channel 0 capture or compare event
0	UPG	Update event generation This bit can be set by software, and automatically cleared by hardware. When this bit is set, the counter is cleared if the center-aligned or up counting mode is selected, while in down counting mode it takes the auto-reload value. The prescaler counter is cleared at the same time. 0: No generate an update event 1: Generate an update event

Channel control register 0 (TIMERx_CHCTL0)

Address offset: 0x18

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
rw		rw		rw				rw							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH1COM CEN	CH1COMCTL[2:0]			CH1COM SEN	CH1COM FEN	CH1MS[1:0]		CH0COM CEN	CH0COMCTL[2:0]			CH0COM SEN	CH0COM FEN	CH0MS[1:0]	
CH1CAPFLT[3:0]			CH1CAPPSC[1:0]		CH0CAPFLT[3:0]			CH0CAPPSC[1:0]							
rw		rw		rw		rw		rw		rw		rw			

Output compare mode:

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15	CH1COMCEN	Channel 1 output compare clear enable Refer to CH0COMCEN description
14:12	CH1COMCTL[2:0]	Channel 1 compare output control Refer to CH0COMCTL[2:0] description
11	CH1COMSEN	Channel 1 output compare shadow enable Refer to CH0COMSEN description
10	CH1COMFEN	Channel 1 output compare fast enable Refer to CH0COMFEN description
9:8	CH1MS[1:0]	Channel 1 mode selection This bit-field specifies the direction of the channel and the input signal selection. The CH1MS[1:0] bit-field is writable only when the channel is not active (the CH1EN bit in TIMERx_CHCTL2 register is reset). 00: Channel 1 is configured as output. 01: Channel 1 is configured as input, IS1 is connected to CI1FE1. 10: Channel 1 is configured as input, IS1 is connected to CI0FE1. 11: Channel 1 is configured as input, IS1 is connected to ITS. This mode is working only if an internal trigger input is selected (through TSCFG15[4:0] bit-field in SYSCFG_TIMERxCFG2 register).
7	CH0COMCEN	Channel 0 output compare clear enable When this bit is set, the O0CPRE signal is cleared when high level is detected on ETIFP input. 0: Channel 0 output compare clear disabled 1: Channel 0 output compare clear enabled
6:4	CH0COMCTL[2:0]	Channel 0 compare output control The CH0COMCTL[2:0] bit-field control the behavior of O0CPRE which drives CH0_O. The active level of O0CPRE is high, while the active level of CH0_O depends on CH0P bit. 000: Timing mode. The O0CPRE signal keeps stable, independent of the comparison between the register TIMERx_CH0CV and the counter TIMERx_CNT.

		001: Set the channel output on match. O0CPRE signal is forced high when the counter matches the output compare register <code>TIMERx_CH0CV</code>. 010: Clear the channel output on match. O0CPRE signal is forced low when the counter matches the output compare register <code>TIMERx_CH0CV</code>. 011: Toggle on match. O0CPRE toggles when the counter matches the output compare register <code>TIMERx_CH0CV</code>. 100: Force low. O0CPRE is forced low level. 101: Force high. O0CPRE is forced high level. 110: PWM mode 0. When counting up, O0CPRE is active as long as the counter is smaller than <code>TIMERx_CH0CV</code>, otherwise it is inactive. When counting down, O0CPRE is inactive as long as the counter is larger than <code>TIMERx_CH0CV</code>, otherwise it is active. 111: PWM mode 1. When counting up, O0CPRE is inactive as long as the counter is smaller than <code>TIMERx_CH0CV</code>, otherwise it is active. When counting down, O0CPRE is active as long as the counter is larger than <code>TIMERx_CH0CV</code>, otherwise it is inactive. When configured in PWM mode, the O0CPRE level changes only when the output compare mode switches from "Timing" mode to "PWM" mode or when the result of the comparison changes.
3	CH0COMSEN	Channel 0 compare output shadow enable When this bit is set, the shadow register of <code>TIMERx_CH0CV</code> register which updates at each update event will be enabled. 0: Channel 0 output compare shadow disabled 1: Channel 0 output compare shadow enabled The PWM mode can be used without validating the shadow register only in single pulse mode (SPM bit in <code>TIMERx_CTL0</code> register is set).
2	CH0COMFEN	Channel 0 output compare fast enable 0: Channel 0 output quickly compare disable. Only the valid edge of the comparison result will generate a comparison match, and <code>CH0_O</code> is set to the compare level. The minimum delay from an edge on the result of the comparison to activate <code>CH0_O</code> output is 5 clock cycles. 1: Channel 0 output quickly compare enable. Both the valid edge of the trigger input signal and the comparison result will generate a comparison match, and <code>CH0_O</code> is set to the compare level. The minimum delay from an edge on the trigger input to activate <code>CH0_O</code> output is 3 clock cycles.
1:0	CH0MS[1:0]	Channel 0 I/O mode selection This bit-field specifies the work mode of the channel and the input signal selection. The <code>CH0MS[1:0]</code> bit-field is writable only when the channel is not active (the <code>CH0EN</code> bit in <code>TIMERx_CHCTL2</code> register is reset). 000: Channel 0 is configured as output. 001: Channel 0 is configured as input, <code>IS0</code> is connected to <code>CI0FE0</code>. 010: Channel 0 is configured as input, <code>IS0</code> is connected to <code>CI1FE0</code>.

011: Channel 0 is configured as input, IS0 is connected to ITS, this mode is working only if an internal trigger input is selected (through TSCFG15[4:0] bit-field in SYSCFG_TIMERxCFG2(x=1,2) register).

100~111: Reserved.

Input capture mode:

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:12	CH1CAPFLT[3:0]	Channel 1 input capture filter control Refer to CH0CAPFLT description.
11:10	CH1CAPPSC[1:0]	Channel 1 input capture prescaler Refer to CH0CAPPSC description.
9:8	CH1MS[1:0]	Channel 1 I/O mode selection Same as output compare mode.
7:4	CH0CAPFLT[3:0]	Channel 0 input capture filter control An event counter is used in the digital filter, in which a transition on the output occurs after N input events. This bit-field specifies the frequency used to sample CIO input signal and the length of the digital filter applied to CIO. 0000: Filter disabled, $f_{SAMP}=f_{DTS}$, $N=1$. 0001: $f_{SAMP}=f_{CK\_TIMER}$, $N=2$. 0010: $f_{SAMP}=f_{CK\_TIMER}$, $N=4$. 0011: $f_{SAMP}=f_{CK\_TIMER}$, $N=8$. 0100: $f_{SAMP}=f_{DTS}/2$, $N=6$. 0101: $f_{SAMP}=f_{DTS}/2$, $N=8$. 0110: $f_{SAMP}=f_{DTS}/4$, $N=6$. 0111: $f_{SAMP}=f_{DTS}/4$, $N=8$. 1000: $f_{SAMP}=f_{DTS}/8$, $N=6$. 1001: $f_{SAMP}=f_{DTS}/8$, $N=8$. 1010: $f_{SAMP}=f_{DTS}/16$, $N=5$. 1011: $f_{SAMP}=f_{DTS}/16$, $N=6$. 1100: $f_{SAMP}=f_{DTS}/16$, $N=8$. 1101: $f_{SAMP}=f_{DTS}/32$, $N=5$. 1110: $f_{SAMP}=f_{DTS}/32$, $N=6$. 1111: $f_{SAMP}=f_{DTS}/32$, $N=8$.
3:2	CH0CAPPSC[1:0]	Channel 0 input capture prescaler This bit-field specifies the factor of the prescaler on channel 0 input. The prescaler is reset when CH0EN bit in TIMERx_CHCTL2 register is cleared. 00: Prescaler disabled, capture is done on each channel input edge. 01: Capture is done every 2 channel input edges. 10: Capture is done every 4 channel input edges.

11: Capture is done every 8 channel input edges.

1:0 CH0MS[1:0] Channel 0 I/O mode selection
Same as output compare mode.

Channel control register 1 (TIMERx_CHCTL1)

Address offset: 0x1C

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH3COM CEN	CH3COMCTL[2:0]			CH3COM SEN	CH3COM FEN	CH3MS[1:0]		CH2COM CEN	CH2COMCTL[2:0]			CH2COM SEN	CH2COM FEN	CH2MS[1:0]	
CH3CAPFLT[3:0]				CH3CAPPSC[1:0]				CH2CAPFLT[3:0]				CH2CAPPSC[1:0]			
rw				rw		rw		rw				rw		rw	

Output compare mode:

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15	CH3COMCEN	Channel 3 output compare clear enable Refer to CH2COMCEN description
14:12	CH3COMCTL[2:0]	Channel 3 compare output control Refer to CH2COMCTL[2:0] description
11	CH3COMSEN	Channel 3 output compare shadow enable Refer to CH2COMSEN description
10	CH3COMFEN	Channel 3 output compare fast enable Refer to CH0COMFEN description
9:8	CH3MS[1:0]	Channel 3 I/O mode selection This bit-field specifies the direction of the channel and the input signal selection. The CH3MS[1:0] bit-field is writable only when the channel is not active (the CH3EN bit in TIMERx_CHCTL2 register is reset). 00: Channel 3 is configured as output. 01: Channel 3 is configured as input, IS3 is connected to CI3FE3. 10: Channel 3 is configured as input, IS3 is connected to CI2FE3. 11: Channel 3 is configured as input, IS3 is connected to ITS, this mode is working only if an internal trigger input is selected (through TSCFG15[4:0] bit-field in SYSCFG_TIMERxCFG2(x=1,2) register).
7	CH2COMCEN	Channel 2 output compare clear enable. When this bit is set, the O2CPRE signal is cleared when high level is detected on

		ETIFP input. 0: Channel 2 output compare clear disabled 1: Channel 2 output compare clear enabled
6:4	CH2COMCTL[2:0]	Channel 2 compare output control Refer to CH0COMCTL[2:0] description.
3	CH2COMSEN	Channel 2 compare output shadow enable When this bit is set, the shadow register of TIMERx_CH2CV register, which updates at each update event will be enabled. 0: Channel 2 output compare shadow disabled 1: Channel 2 output compare shadow enabled The PWM mode can be used without validating the shadow register only in single pulse mode (SPM bit in TIMERx_CTL0 register is set).
2	CH2COMFEN	Channel 2 output compare fast enable Refer to CH0COMFEN description
1:0	CH2MS[1:0]	Channel 2 I/O mode selection This bit-field specifies the work mode of the channel and the input signal selection. The CH2MS[1:0] bit-field is writable only when the channel is not active (the CH2EN bit in TIMERx_CHCTL2 register is reset). 00: Channel 2 is configured as output. 01: Channel 2 is configured as input, IS2 is connected to CI2FE2. 10: Channel 2 is configured as input, IS2 is connected to CI3FE2. 11: Channel 2 is configured as input, IS2 is connected to ITS. This mode is working only if an internal trigger input is selected (through TSCFG15[4:0] bit-field in SYSCFG_TIMERxCFG2 register).

Input capture mode:

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:12	CH3CAPFLT[3:0]	Channel 3 input capture filter control Refer to CH2CAPFLT description.
11:10	CH3CAPPSC[1:0]	Channel 3 input capture prescaler Refer to CH2CAPPSC description.
9:8	CH3MS[1:0]	Channel 3 mode selection Same as output compare mode.
7:4	CH2CAPFLT[3:0]	Channel 2 input capture filter control An event counter is used in the digital filter, in which a transition on the output occurs after N input events. This bit-field specifies the frequency used to sample CI2 input signal and the length of the digital filter applied to CI2. 0000: Filter disabled, $f_{SAMP}=f_{DTS}$, $N=1$.

0001: $f_{SAMP}=f_{CK\_TIMER}$, $N=2$.
0010: $f_{SAMP}=f_{CK\_TIMER}$, $N=4$.
0011: $f_{SAMP}=f_{CK\_TIMER}$, $N=8$.
0100: $f_{SAMP}=f_{DTS}/2$, $N=6$.
0101: $f_{SAMP}=f_{DTS}/2$, $N=8$.
0110: $f_{SAMP}=f_{DTS}/4$, $N=6$.
0111: $f_{SAMP}=f_{DTS}/4$, $N=8$.
1000: $f_{SAMP}=f_{DTS}/8$, $N=6$.
1001: $f_{SAMP}=f_{DTS}/8$, $N=8$.
1010: $f_{SAMP}=f_{DTS}/16$, $N=5$.
1011: $f_{SAMP}=f_{DTS}/16$, $N=6$.
1100: $f_{SAMP}=f_{DTS}/16$, $N=8$.
1101: $f_{SAMP}=f_{DTS}/32$, $N=5$.
1110: $f_{SAMP}=f_{DTS}/32$, $N=6$.
1111: $f_{SAMP}=f_{DTS}/32$, $N=8$.

3:2	CH2CAPPSC[1:0]	Channel 2 input capture prescaler This bit-field specifies the factor of the prescaler on channel 2 input. The prescaler is reset when CH2EN bit in TIMEx_CHCTL2 register is cleared. 00: Prescaler disabled, capture is done on each channel input edge. 01: Capture is done every 2 channel input edges. 10: Capture is done every 4 channel input edges. 11: Capture is done every 8 channel input edges.
1:0	CH2MS[1:0]	Channel 2 mode selection Same as output compare mode.

Channel control register 2 (TIMEx_CHCTL2)

Address offset: 0x20

Reset value: 0x0000 0000

This register can be accessed by half-word(16-bit) or word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH3NP	Reserved	CH3P	CH3EN	CH2NP	Reserved	CH2P	CH2EN	CH1NP	Reserved	CH1P	CH1EN	CH0NP	Reserved	CH0P	CH0EN
rw		rw	rw	rw		rw	rw	rw		rw	rw	rw		rw	rw

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15	CH3NP	Channel 3 complementary capture polarity Refer to CH0NP description.

14	Reserved	Must be kept at reset value.
13	CH3P	Channel 3 capture/compare function polarity Refer to CH0P description
12	CH3EN	Channel 3 capture/compare function enable Refer to CH0EN description
11	CH2NP	Channel 2 complementary capture polarity Refer to CH0NP description.
10	Reserved	Must be kept at reset value.
9	CH2P	Channel 2 capture/compare function polarity Refer to CH0P description
8	CH2EN	Channel 2 capture/compare function enable Refer to CH0EN description
7	CH1NP	Channel 1 complementary capture polarity Refer to CH0NP description.
6	Reserved	Must be kept at reset value.
5	CH1P	Channel 1 capture/compare function polarity Refer to CH0P description
4	CH1EN	Channel 1 capture/compare function enable Refer to CH0EN description
3	CH0NP	Channel 0 complementary capture polarity When CH0 is configured in input mode, in conjunction with CH0P, this bit is used to define the polarity of CH0.
2	Reserved	Must be kept at reset value.
1	CH0P	Channel 0 capture/compare function polarity When channel 0 is configured in output mode, this bit specifies the output signal polarity. 0: Channel 0 active high 1: Channel 0 active low When channel 0 is configured in input mode, this bit specifies the CIO signal polarity. CH0P will select the active trigger or capture polarity for CIOFE0 or CIOFE1. 00: CIOFE0's rising edge is the active signal for capture or trigger operation in slave mode. And CIOFE0 will not be inverted. 01: CIOFE0's falling edge is the active signal for capture or trigger operation in slave mode. And CIOFE0 will be inverted. 10: Reserved. 11: Noninverted / both CIOFE0's edges.

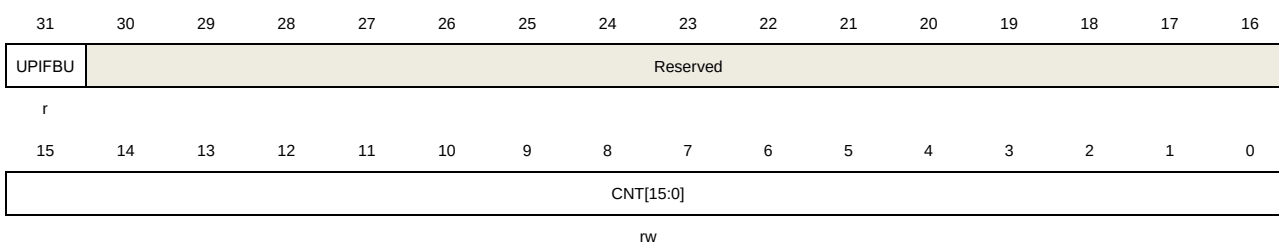
0	CH0EN	Channel 0 capture/compare function enable When channel 0 is configured in output mode, setting this bit enables CH0_O signal in active state. When channel 0 is configured in input mode, setting this bit enables the capture event in channel0. 0: Channel 0 disabled 1: Channel 0 enabled

Counter register (TIMERx_CNT)

Address offset: 0x24

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



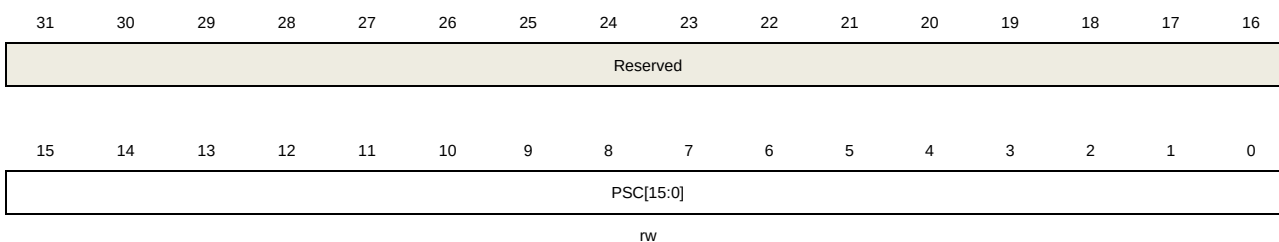
Bits	Fields	Descriptions
31	UPIFBU	UPIF bit backup This bit is a backup of UPIF bit in TIMERx_INTF register and read-only. This bit is only valid when UPIFBUEN = 1. If the UPIFBUEN =0, this bit is reserved and read the result is 0.
30:16	Reserved	Must be kept at reset value.
15:0	CNT[15:0]	This bit-field indicates the current counter value. Writing to this bit-field can change the value of the counter.

Prescaler register (TIMERx_PSC)

Address offset: 0x28

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



Bits	Fields	Descriptions

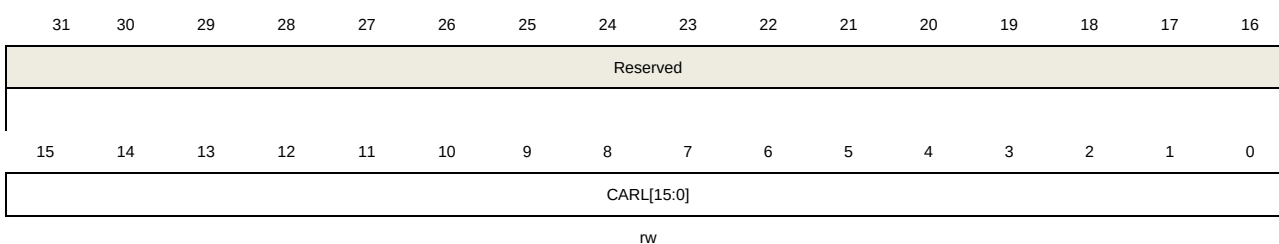
31:16	Reserved	Must be kept at reset value.
15:0	PSC[15:0]	Prescaler value of the counter clock The PSC clock is divided by (PSC+1) to generate the counter clock. The value of this bit-field will be loaded to the corresponding shadow register at every update event.

Counter auto reload register (TIMERx_CAR)

Address offset: 0x2C

Reset value: 0x0000 FFFF

This register has to be accessed by word (32-bit).



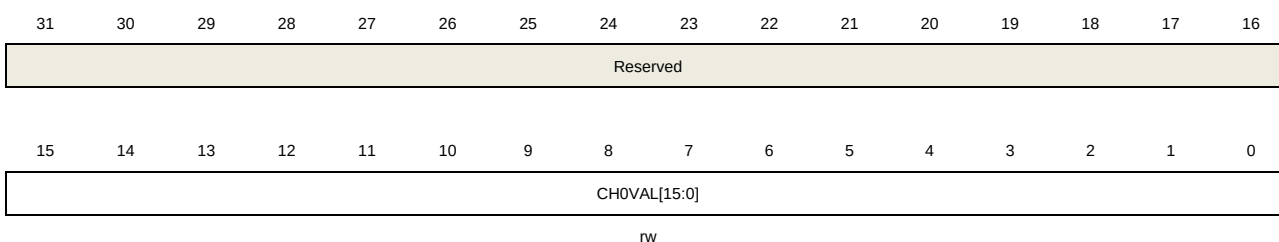
Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CARL[15:0]	Counter auto reload value This bit-field specifies the auto reload value of the counter.

Channel 0 capture/compare value register (TIMERx_CH0CV)

Address offset: 0x34

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CH0VAL[15:0]	Capture/compare value of channel0 When channel 0 is configured in input mode, this bit-field indicates the counter value at the last capture event. And this bit-field is read-only. When channel 0 is configured in output mode, this bit-field contains value to be

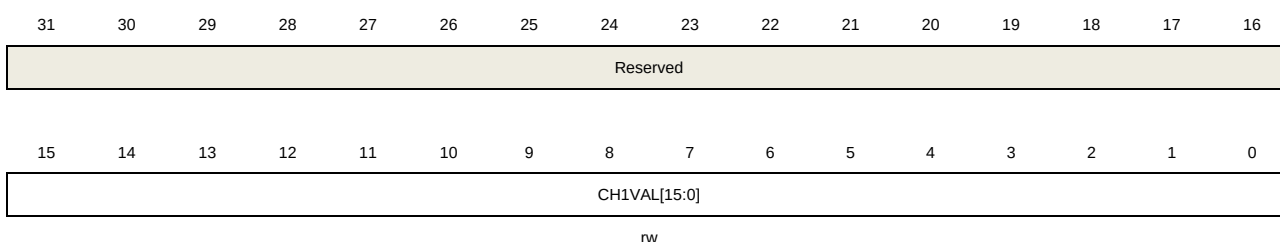
compared to the counter. When the corresponding shadow register is enabled, the shadow register updates by every update event.

Channel 1 capture/compare value register (TIMERx_CH1CV)

Address offset: 0x38

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



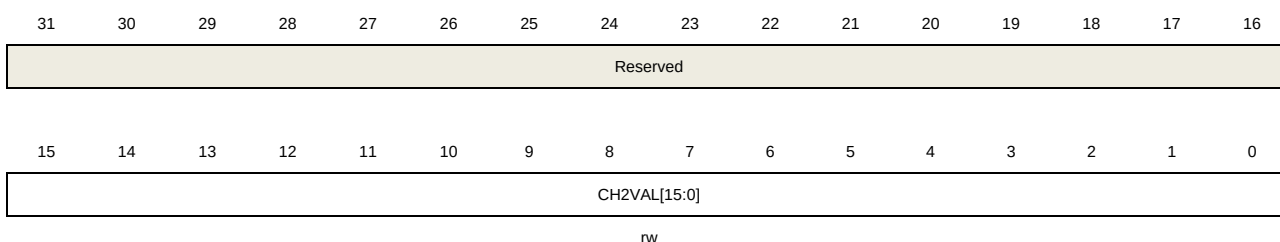
Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CH1VAL[15:0]	Capture/compare value of channel1 When channel 1 is configured in input mode, this bit-field indicates the counter value at the last capture event.And this bit-field is read-only. When channel 1 is configured in output mode, this bit-field contains value to be compared to the counter. When the corresponding shadow register is enabled, the shadow register updates by every update event.

Channel 2 capture/compare value register (TIMERx_CH2CV)

Address offset: 0x3C

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CH2VAL[15:0]	Capture/compare value of channel 2 When channel 2 is configured in input mode, this bit-field indicates the counter value at the last capture event. And this bit-field is read-only.

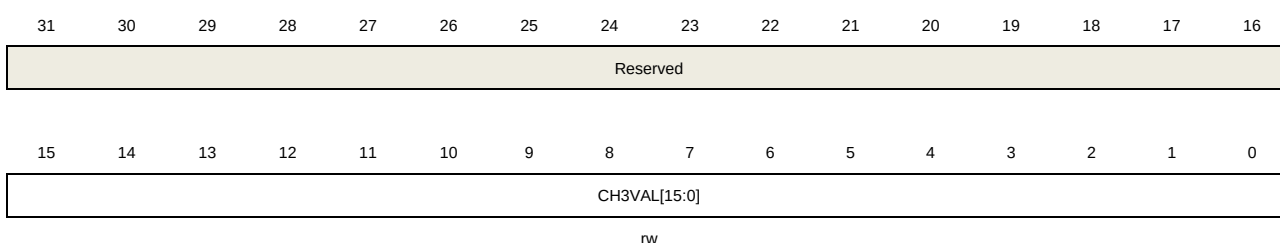
When channel 2 is configured in output mode, this bit-field contains value to be compared to the counter. When the corresponding shadow register is enabled, the shadow register updates by every update event.

Channel 3 capture/compare value register (TIMERx_CH3CV)

Address offset: 0x40

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



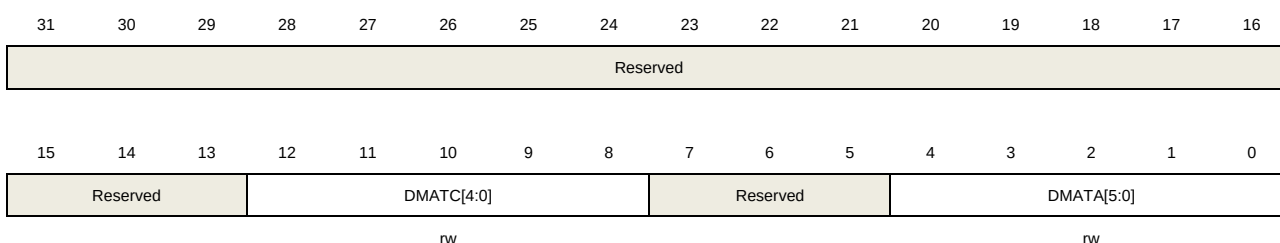
Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CH3VAL[15:0]	Capture/compare value of channel 3 When channel3 is configured in input mode, this bit-field indicates the counter value at the last capture event. And this bit-field is read-only. When channel 3 is configured in output mode, this bit-field contains value to be compared to the counter. When the corresponding shadow register is enabled, the shadow register updates by every update event.

DMA configuration register (TIMERx_DMACFG)

Address offset: 0xE0

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



Bits	Fields	Descriptions
31:13	Reserved	Must be kept at reset value.
12:8	DMATC[4:0]	DMA transfer count This field defines the times of accessing(R/W) the TIMERx_DMATB register by

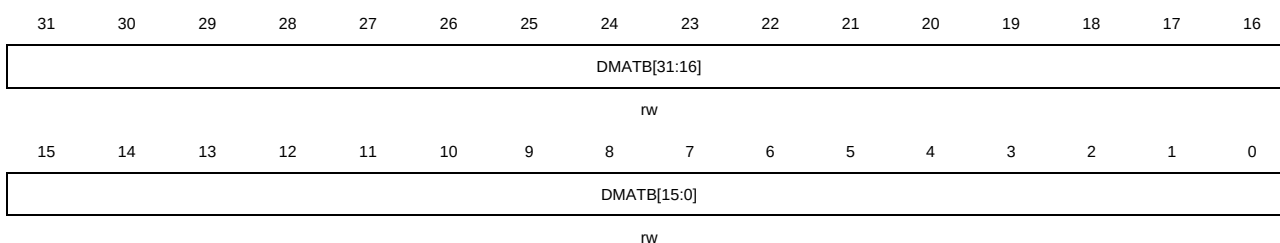
		DMA. 5'b00000: transfer 1 time 5'b00001: transfer 2 times ... 5'b10000: transfer 17 times
7:5	Reserved	Must be kept at reset value.
4:0	DMATA[4:0]	DMA transfer access start address This field defines the start address of accessing the TIMERx_DMATB register by DMA. When the first access to the TIMERx_DMATB register is done, this bit-field specifies the address just accessed. And then the address of the second access to the TIMERx_DMATB register will be (start address + 0x4). 5'b00000: TIMERx_CTL0 5'b00001: TIMERx_CTL1 ... 5'b10000: TIMERx_CH3CV In a word: start address = TIMERx_CTL0 + DMATA*4

DMA transfer buffer register (TIMERx_DMATB)

Address offset: 0xE4

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



Bits	Fields	Descriptions
31:0	DMATB[31:0]	DMA transfer buffer When a read or write operation is assigned to this register, the register located at the address ranges from (start address) to (start address + transfer count * 4) will be accessed. The transfer count is calculated by hardware, and ranges from 0 to DMATC.

Configuration register (TIMERx_CFG)

Address offset: 0xFC

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



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

Bits	Fields	Descriptions
31:2	Reserved	Must be kept at reset value.
1	CHVSEL	Write CHxVAL register selection bit This bit-field is set and reset by software. 1: If the value to be written to the CHxVAL register is the same as the value of CHxVAL register, the write access is ignored. 0: No effect.
0	Reserved	Must be kept at reset value.

22 General purpose timer (GPTIMERx)

22.1 Overview

The general purpose timer module (GPTIMER0/1) is two-channel timer that supports both input capture and output compare. They can generate PWM signals to be used for power factor correction (PFC). The general purpose timer has a 16-bit counter that can be used as an unsigned counter.

In addition, the general purpose timers can be programmed and be used for counting, their external events can be used to drive other timers.

Timers are completely independent with each other, but they may be synchronized to provide a larger timer with their counter value increasing in unison (Through the EVIC function).

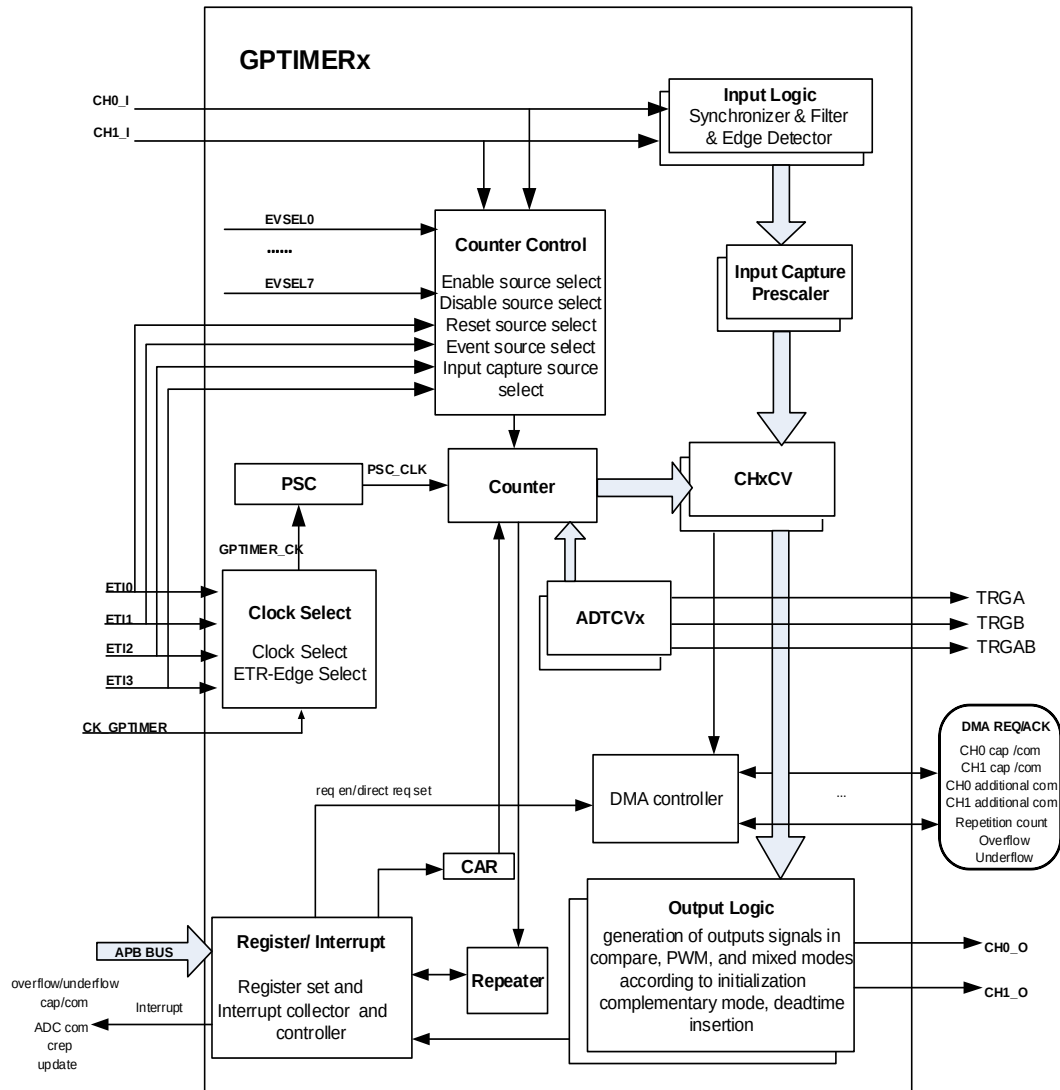
22.2 Characteristics

- Total channel num: 2.
- Counter width: 16 bits.
- Selectable clock source: internal clock, external trigger.
- Multiple counter modes: up counting, down counting and center-aligned counting.
- Programmable prescaler: 16 bits. The factor can be changed ongoing.
- Each channel is independent and user-configurable: input capture mode, output compare mode, programmable PWM mode, single pulse mode.
- Auto reload function.
- Programmable counter repetition function.
- Interrupt output or DMA request: counter reset event, trigger event, compare/capture event.
- Synchronous start/stop/reset of multiple timers can be triggered by EVIC, channel input and ETI input
- Timer synchronization allows selected timers to start counting at the same tick period.
- The ADC triggers the request output and the pin monitoring, and has the repeat skip function.
- Global updates.
- Complementary output and dead-time insertion (asymmetric).
- Work with GTOC to achieve stop output function.
- External pulse width measurement

22.3 Block diagram

[Figure 22-1. General purpose timer block diagram](#) provides details of the internal configuration of the advanced time.

Figure 22-1. General purpose timer block diagram



22.4 Function overview

22.4.1 Clock selection

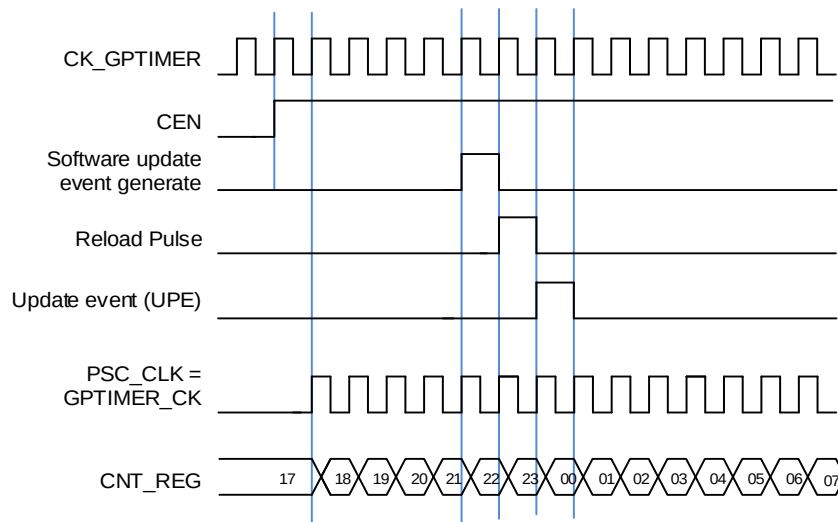
The general purpose timer can be driven by an internal clock source CK_GPTIMER or an external input clock source ETI0/ETI1/ETI2/ETI3 (provided to GPTIMER after filtering and polarity selection via GTOC) and selected by CKSSEL[2:0] in GPTIMERx_CTL0 registers. When an externally input clock source is used, a valid edge can be selected by CKESEL[2:0]

in GPTIMERx_CTL0 registers.

- CKSSEL[2:0] = 3'b000 in GPTIMERx_CTL0 registers. Internal clock CK_GPTIMER is selected as timer clock source which is from module RCU.

When the CEN is set, the CK_GPTIMER will be divided by PSC value to generate PSC_CLK.

Figure 22-2. Normal mode, internal clock divided by 1



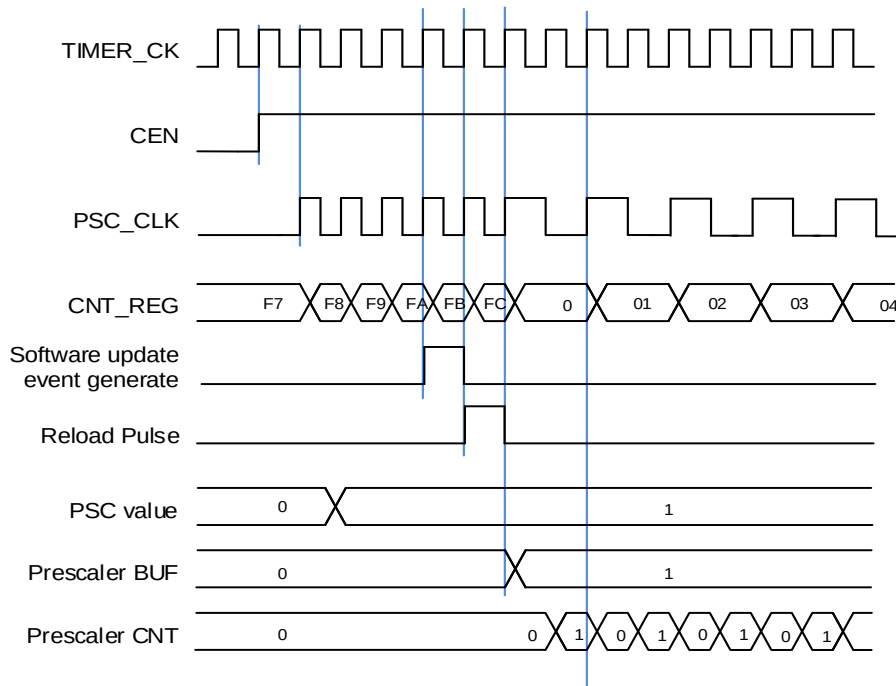
- CKSSEL[2:0] = 3'b001/3'b010/3'b011/3'b100 in GPTIMERx_CTL0 registers. External input pin is selected as timer clock source.

The GPTIMER_CLK, which drives counter's prescaler to count, can be triggered by the event of rising or falling edge on the external pin ETI0/ETI1/ETI2/ETI3.

22.4.2 Clock prescaler

The prescaler can divide the timer clock (GPTIMER_CLK) to a counter clock (PSC_CLK) by any factor ranging from 1 to 65536. It is controlled by prescaler register (GPTIMERx_PSC) which can be changed ongoing, but it is adopted at the next update event.

Figure 22-3. Counter timing diagram with prescaler division change from 1 to 2



22.4.3 Enable/disable/reset the timer counter

The counter is enabled by set CEN bit in GPTIMERx_CTL0 registers, and the counter is disabled by reset CEN bit. The CEN bit can be changed from the following sources:

- Set CEN bit in GPTIMERx_CTL0 registers.
- When SWENCNT = 1 in GPTIMERx_ESSEL registers, set GPTIMERxSWEN bit in GPTIMERx_SWEN registers.
- When SWDISCNT = 1 in GPTIMERx_DSSEL registers, set GPTIMERxSWDIS bit in GPTIMERx_SWDIS registers.
- When the hardware source event selected by GPTIMERx_ESSEL registers occurs(ETI input, EVIC event or channel input).
- When the hardware source event selected by GPTIMERx_DSSEL registers occurs(ETI input, EVIC event or channel input).

The counter can be reset by the following sources:

- Write 0 to GPTIMERx_CNT register.
- When SWRSTCNT = 1 in GPTIMERx_RSSEL registers, set GPTIMERxSWDIS bit in GPTIMERx_SWRST registers.
- When the hardware source event selected by GPTIMERx_RSSEL registers occurs(ETI input, event input source selection signal or channel input).

The register GPTIMERx_CNT is not allowed to be written during counter counting. When the counter is counted or stopped, the register GPTIMERx_SWRST can be written.

Down counting in the edge alignment mode, the value of GPTIMERx_CNT changes to the period value GPTIMERx_CAR after reset, and in other cases, GPTIMERx_CNT changes to 0 after reset.

In event count mode, when the GPTIMERx_SWRST register is written, the counter is reset after a CK_GPTIMER clock.

Counter update priority

The priority of the update action of GPTIMERx_CNT (decreasing from top to bottom):

1. CPU write behavior: write to the CNT register or write to the GPTIMERx_SWRST register.
2. The hardware reset source selected by GPTIMERx_REESEL is reset.
3. Hardware sources selected by GPTIMERx_CUPEVSSEL and GPTIMERx_CDNEVSSEL perform up/down counting;
4. Cycle counting behavior.

TIMER enable priority control:

The CPU writes CEN bit in GPTIMERx_CTL1 registers have the highest priority. GPTIMERxSWEN bit in GPTIMERx_SWEN registers, and GPTIMERxSWDIS bit in GPTIMERx_SWDIS registers with higher priority than the start-stop control of the hardware source.

The CEN does not change when the start and stop events of the hardware source selection occur simultaneously.

When CEN changes, the CPU reads the register, reading the value before the change.

In single pulse mode, the hardware off CEN at the end of the cycle has the highest priority.

22.4.4 Period counting mode

Up counting mode

In this mode, the counter counts up continuously from 0 to the counter reload value, which is defined in the GPTIMERx_CAR register. Once the counter reaches the counter reload value, the counter restarts from 0. If the repetition counter is set, the update event will be generated after (GPTIMERx_CREP+1) times of overflow. Otherwise the update event is generated each time when counter overflows. The counting direction bit DIR in the GPTIMERx_CTL0 register should be set to 0 for the up-counting mode.

When the hardware reset source selected by GPTIMERx_REESEL occurs, or the software reset source is selected and the corresponding bit of the GPTIMERx_SWRST register is set, the counter value will be initialized to 0 and an update event will be generated.

If the UPDIS bit in GPTIMERx_CTL0 register is set, the update event is disabled.

When an update event occurs, the prescaler register and repetition counter register are updated.

[Figure 22-4. Timing diagram of up counting mode, PSC=0/2](#) and [Figure 22-5. Timing diagram of up counting mode, change GPTIMERx_CAR ongoing](#) show some examples of the counter behavior for different clock prescaler factors when GPTIMERx_CAR=0x99.

Figure 22-4. Timing diagram of up counting mode, PSC=0/2

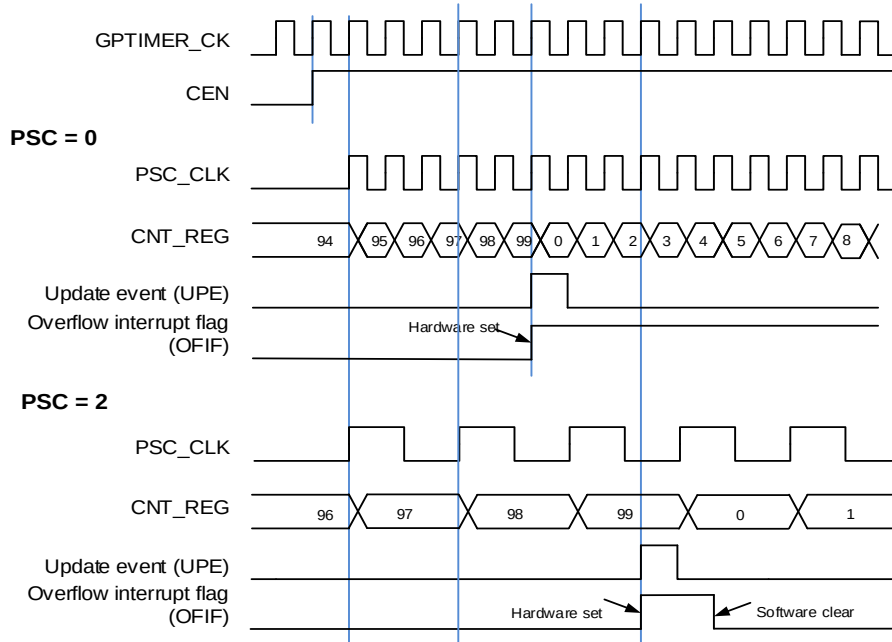
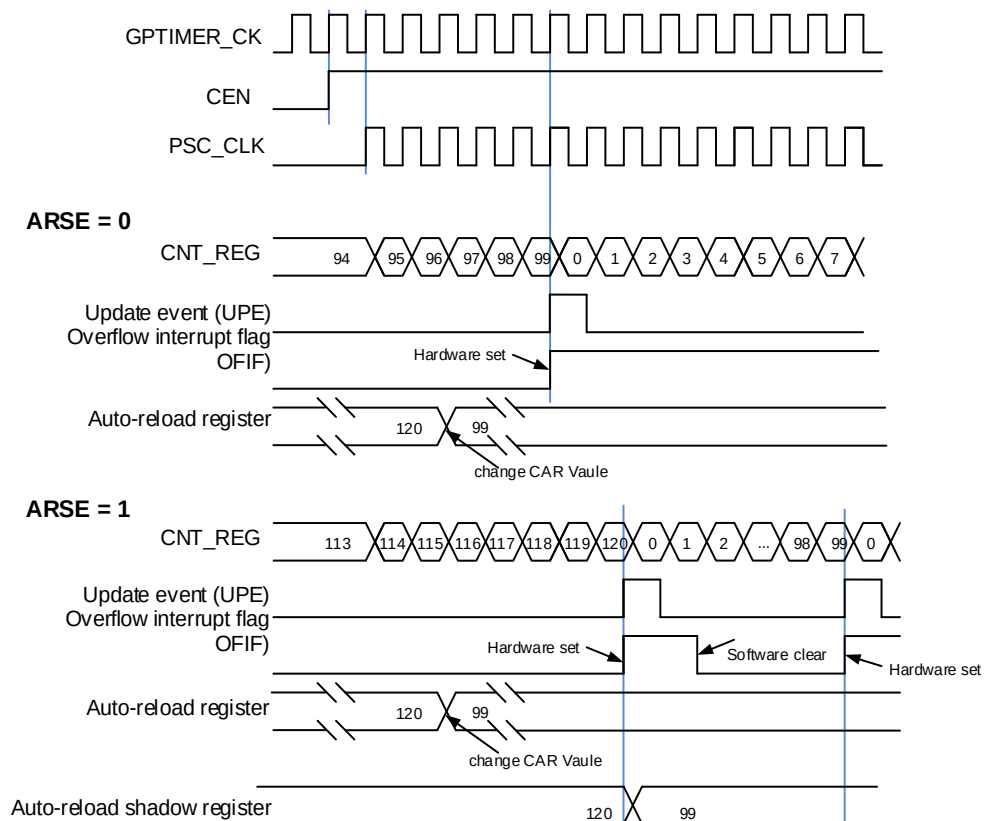


Figure 22-5. Timing diagram of up counting mode, change GPTIMERx_CAR ongoing



Down counting mode

In this mode, the counter counts down continuously from the counter reload value, which is defined in the GPTIMERx_CAR register. Once the counter reaches 0, the counter restarts to count again from the counter reload value. If the repetition counter is set, the update event will be generated after (GPTIMERx_CREP+1) times of underflow. Otherwise, the update event is generated each time when counter underflows. The counting direction bit DIR in the GPTIMERx_CTL0 register should be set to 1 for the down counting mode.

When the hardware reset source selected by GPTIMERx_REESEL occurs, or the software reset source is selected and the corresponding bit of the GPTIMERx_SWRST register is set, the counter value will be initialized to the counter reload value and an update event will be generated.

If the UPDIS bit in GPTIMERx_CTL0 register is set, the update event is disabled.

When an update event occurs, the prescaler register and repetition counter register are updated.

[Figure 22-6. Timing diagram of down counting mode, PSC=0/2](#) and [Figure 22-7. Timing diagram of down counting mode, change GPTIMERx_CAR ongoing](#) show some examples of the counter behavior in different clock frequencies when GPTIMERx_CAR = 0x99.

Figure 22-6. Timing diagram of down counting mode, PSC=0/2

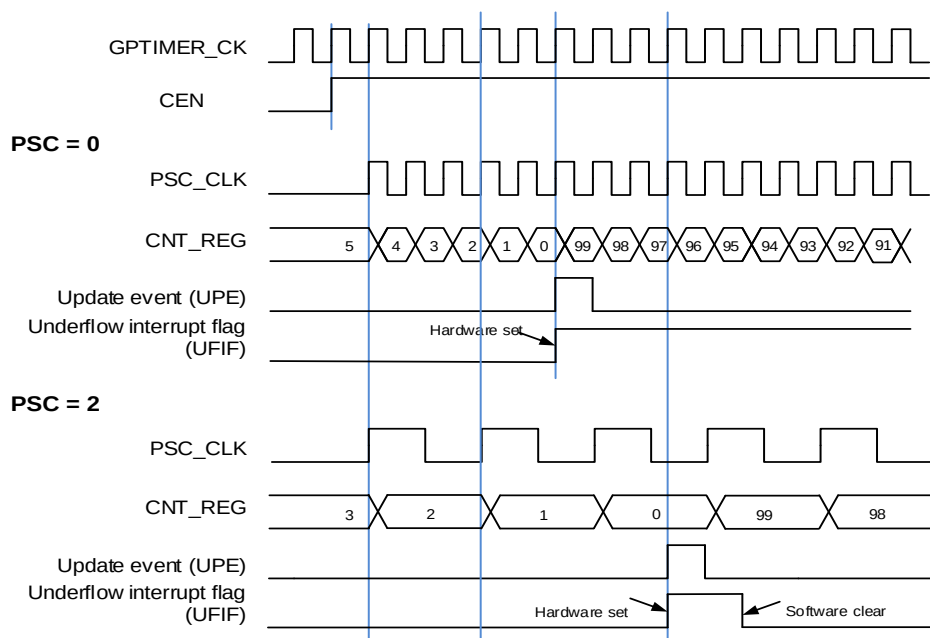
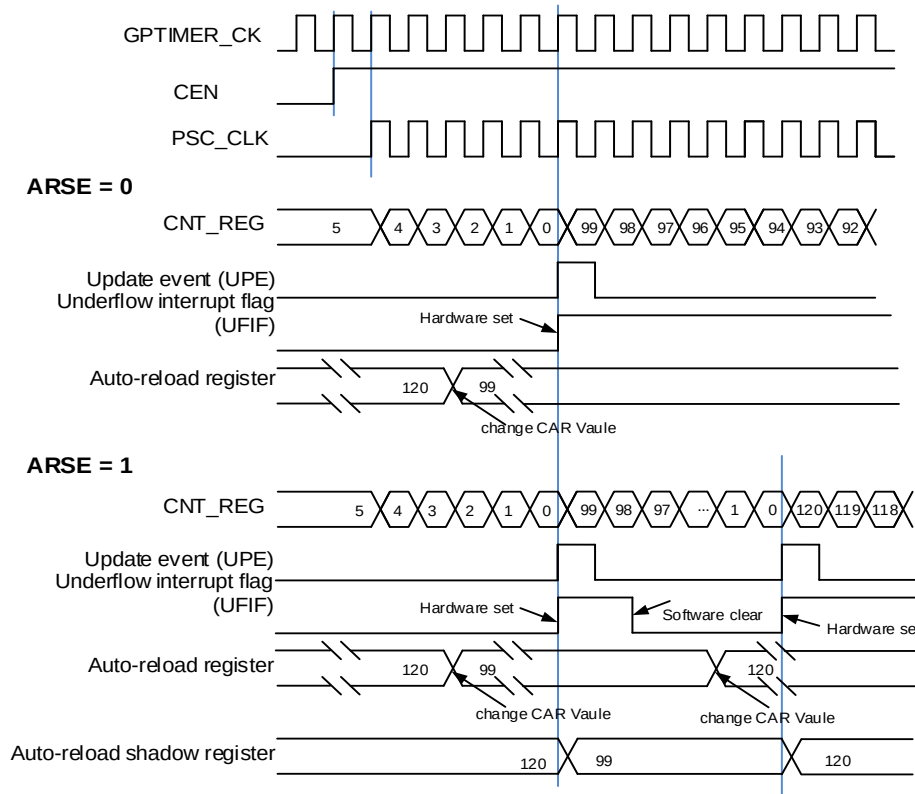


Figure 22-7. Timing diagram of down counting mode, change GPTIMERx_CAR ongoing



Center-aligned counting mode

In the center-aligned counting mode, the counter counts up from 0 to the counter reload value and then counts down to 0 alternatively. The timer module generates an overflow event when the counter counts to (GPTIMERx_CAR-1) in the count-up direction and generates an underflow event when the counter counts to 1 in the count-down direction. The counting direction bit DIR in the GPTIMERx_CTL0 register is read-only and indicates the counting direction when in the center-aligned counting mode. The counting direction is updated by hardware automatically.

When the hardware reset source selected by GPTIMERx_REESEL occurs, or the software reset source is selected and the corresponding bit of the GPTIMERx_SWRST register is set, the counter value will be initialized to 0 and generate an update event irrespective of whether the counter is counting up or down in the center-aligned counting mode.

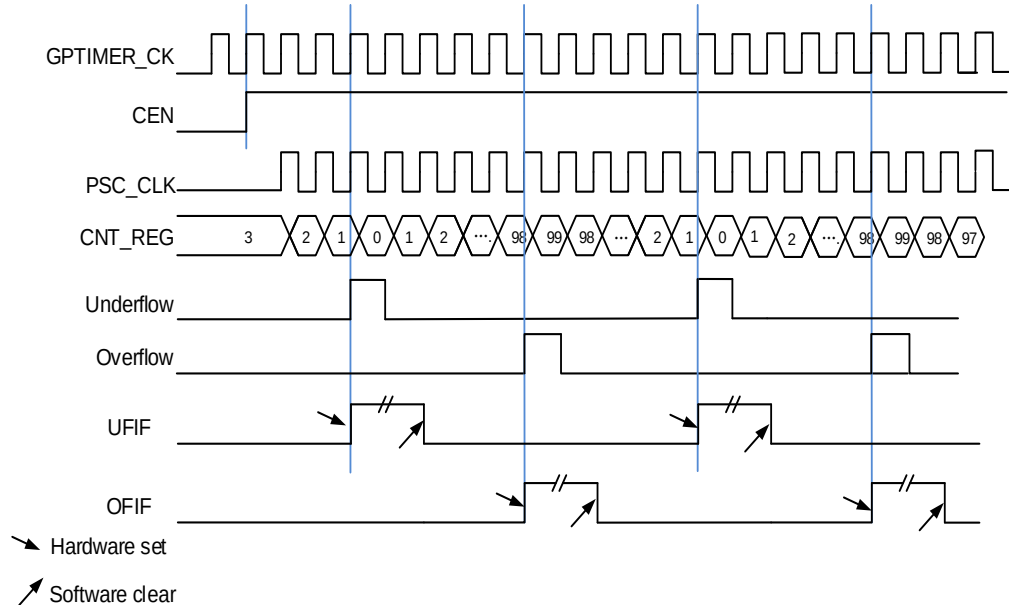
If the UPDIS bit in the GPTIMERx_CTL0 register is set, the update event is disabled.

When an update event occurs, all the registers (repetition counter register, prescaler register) are updated.

In output compare mode or ADC trigger mode, the compare registers and additional compare registers (CHxCV/CHxCOMVAL_ADD(x=0,1)) used to compare matches, and the two compare registers (GPTIMERx_ADCCR1 / GPTIMERx_ADCCR2) dedicated to generating ADC triggers, are also controlled by the global update function. Shadow function details will be presented in the [Shadow register loading function](#).

Figure 22-8. Timing diagram of center-aligned counting mode shows some examples of the counter behavior when `GPTIMERx_CAR=0x99`, `GPTIMERx_PSC=0x0`, `GPTIMERx_CREP=0x0`.

Figure 22-8. Timing diagram of center-aligned counting mode



22.4.5 Event counting mode

The counter can count the events for the `GPTIMERx_CUPEVSSEL` and `GPTIMERx_CDNEVSSEL` register selected hardware source. The clock source selected by `CKSSEL[2:0]` bits in the `GPTIMERx_CTL0` register and the counting direction selected by `DIR` bit in the `GPTIMERx_CTL0` is invalid. The value of the counter remains unchanged if both the up-count source and down-count source selection events occur.

Overflow events and underflow events are generated in the same way as cycle counting.

The `CEN` bit in the `GPTIMERx_CTL0` is set to start event counting via the hardware source, and the `TIMER` will be started after a `CK_GPTIMER` cycle.

In event-count mode, `DIR` is controlled by the hardware and is set to 0 when an up-count event occurs and is set to 1 when the next down-count event occurs. When the update event occurs in event count mode, the value of `GPTIMERx_CNT` is reset according to the current `DIR` value.

External pulse width measurement

The `TIMER` can measure the pulse width of the channel input or the `ETI` input. The pulse to be measured (high / low level, count direction) is selected by `INTLCUPS[3:0]` bits in the `GPTIMERx_CUPEVSSEL` register or `INTLCDWS[3:0]` bits in the `GPTIMERx_CDNEVSSEL` register.

Up counting and down counting pulses prohibit the selection of the same.

In this mode, the clock source is selected by CKSSEL[2:0] bits in the GPTIMERx_CTL0 register, the counting period is set by GPTIMERx_CAR, and the initial value of the counter is set by GPTIMERx_CNT.

22.4.6 count direction

The direction of the counter is specified by configuring DIR bit in the GPTIMERx_CTL0 register.

In edge-aligned mode, if DIR is changed during counting, the counting direction will change on either overflow (when counting up) or underflow (when counting down). When the FDIR bit in the GPTIMERx_CTL0 register is reset, if DIR is changed when counting stops, the direction of the count is changed on the overflow / underflow after counting is restarted. When the FDIR bit in the GPTIMERx_CTL0 register is set, the counting direction is changed as soon as counting is restarted.

In center-aligned mode, if DIR is changed during counting, the counting direction will not change. When the FDIR bit in the GPTIMERx_CTL0 register is reset, if DIR is changed when counting stops, the counting direction will not change. When the FDIR bit in the GPTIMERx_CTL0 register is set, the counting direction is changed as soon as counting is restarted.

When changing from edge-aligned mode to center-aligned mode, the count direction is forced to be up only when the counter reaches 0, the count direction is forced to be down only when the counter reaches period value.

22.4.7 Counter repetition

Repetition counter is used to generate the update event or update the timer registers only after a given number (N+1) cycles of the counter, where N is the value of CREP bits in GPTIMERx_CREP register. The repetition counter is decremented at each counter overflow in up counting mode, at each counter underflow in down counting mode, at each counter underflow in center-aligned counting mode. When the CREPCNT value of the GPTIMERx_CREP register is decreased to 0, the RCENDIF bit in the GPTIMERx_INTF register will be set, and if the corresponding enable bit in the GPTIMERx_DMAINTEN register is set, the corresponding interrupt signal is generated.

CREPCNT keeps the current value when the repeat counter is enabled and the GPTIMER counter stops, and CREPCNT continues decreasing from the current value when the counter is restarted.

Figure 22-9. Repetition counter timing diagram of center-aligned counting mode

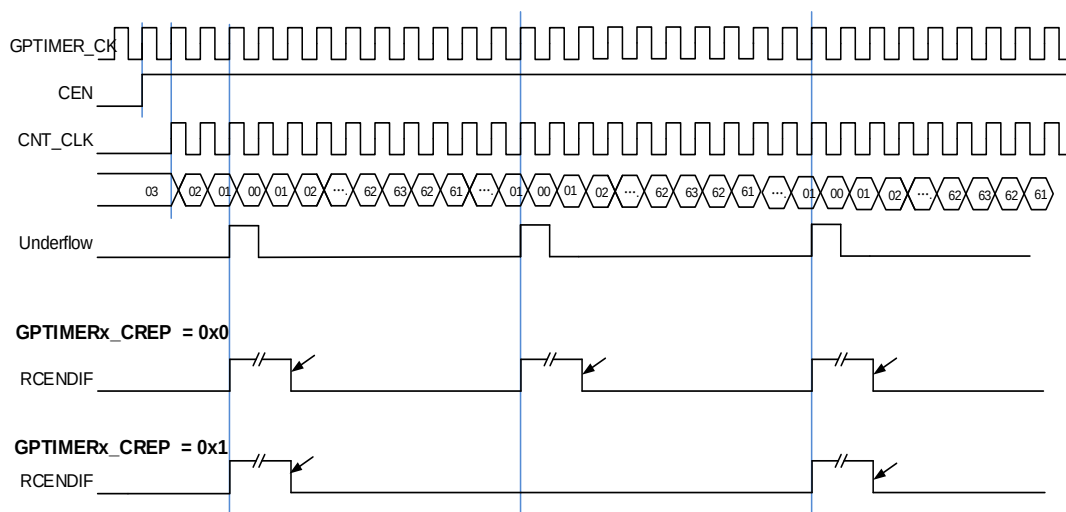


Figure 22-10. Repetition counter timing diagram of up counting mode

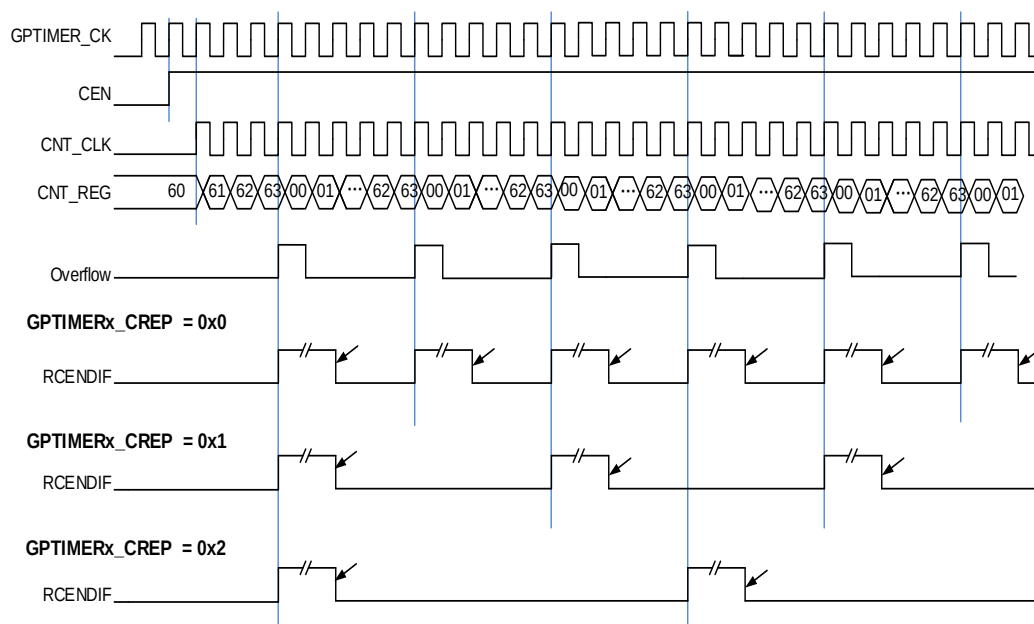
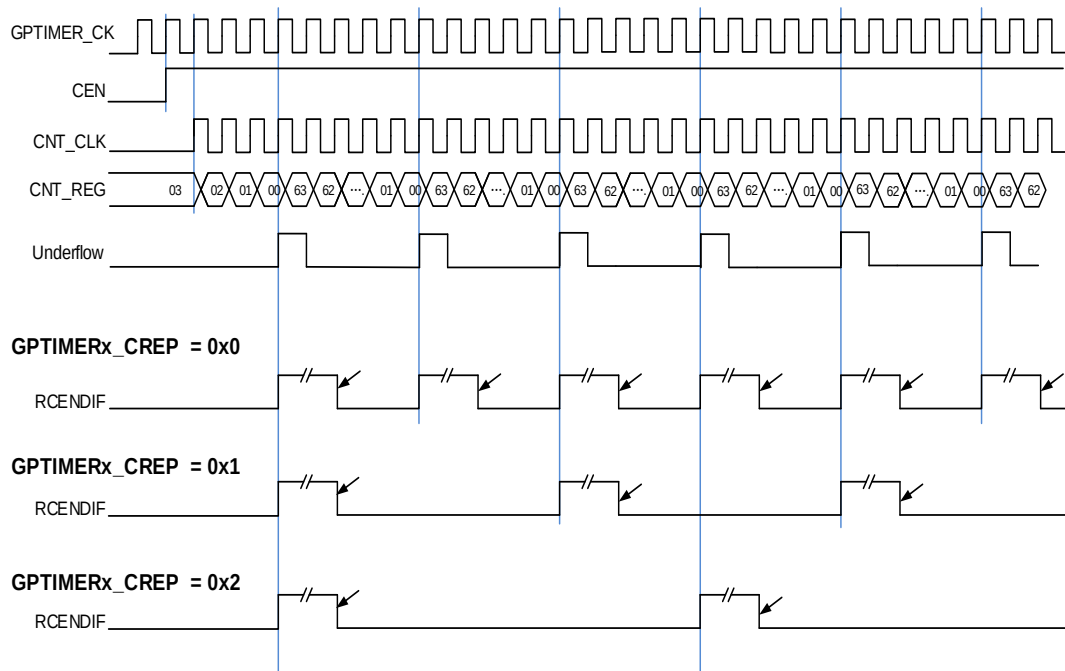


Figure 22-11. Repetition counter timing diagram of down counting mode

22.4.8 Shadow register loading function

GPTIMERx_CAR, GPTIMERx_CHxCV, GPTIMERx_CHxCOMV_ADD, GPTIMERx_ADCCR1 and GPTIMERx_ADCCR2 (x = 0, 1) all have separate shadow function enable bit. When shadow function is disabled, the corresponding register with shadow register function will immediately be overloaded with the newly configured value, and the timer will operate with the newly configured value. Registers with shadow register function also have a separate 2-bit update mode selection in GPTIMERx_UPSSEL register. When shadow function is enabled, each shadow register can be selected to enable global update function, overflow update, underflow update, or both overflow update and underflow update. The global update function can achieve unified updates of selected shadow registers at the set time.

Global one-time reload mode

If GLBUPSEL[1:0] $\neq$ 2'b11 in the GPTIMERx_UPSSEL register, the global one-time reload mode enabled. Selecting a uniform time to reload the register value by GLBUPSEL[1:0] bits.

As is shown in [Figure 22-12. Timing diagram for updates in non-global one-time reload mode on update events](#), CAR does not use the global one-time reload mode (CARUPSEL[1:0] $\neq$ 2'b00), and enables automatic overloading of shadow registers (ARSE=1). When CARUPSEL[1:0] = 2'b01, CAR overloads at the next overflow time, CARUPSEL[1:0] = 2'b10 CAR is overloaded at the next underflow moment, and CARUPSEL[1:0] = 2'b11 CAR is overloaded at the next overflow or underflow moment.

Figure 22-12. Timing diagram for updates in non-global one-time reload mode on

update events

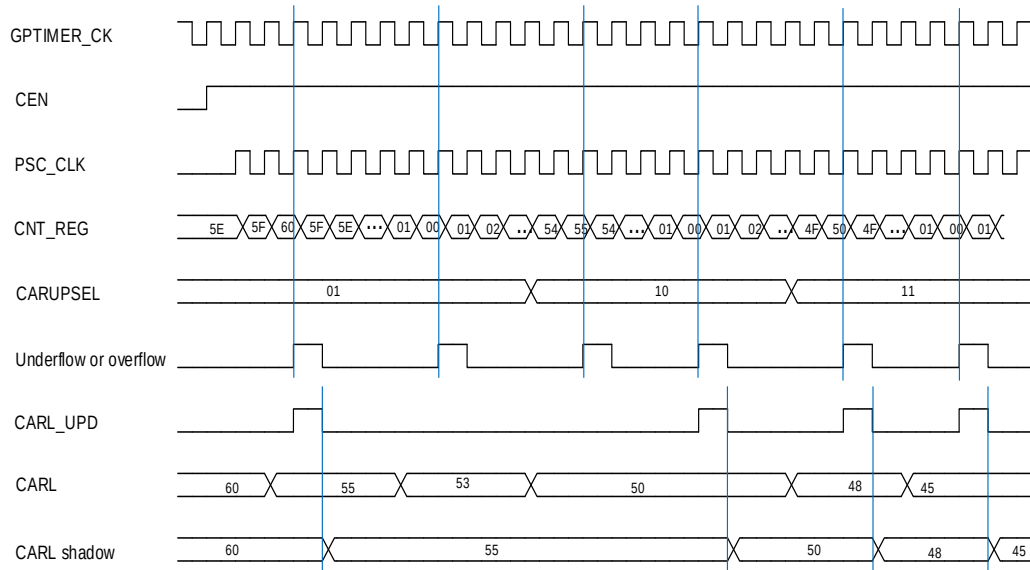


Figure 22-13. Timing diagram for updates in global one-time reload mode on update events ($GLBUPSEL[1:0] \neq 2'b11$) shows CARL and CHxCV(x = 0..1) update timing diagram in global one-time reload mode. (CARUPSEL[1:0] = 2'b00, CHxCVUPSEL[1:0] = 2'b00, ARSE = 1 and CHxCOMSEN = 1).

Figure 22-13. Timing diagram for updates in global one-time reload mode on update events ($GLBUPSEL[1:0] \neq 2'b11$)

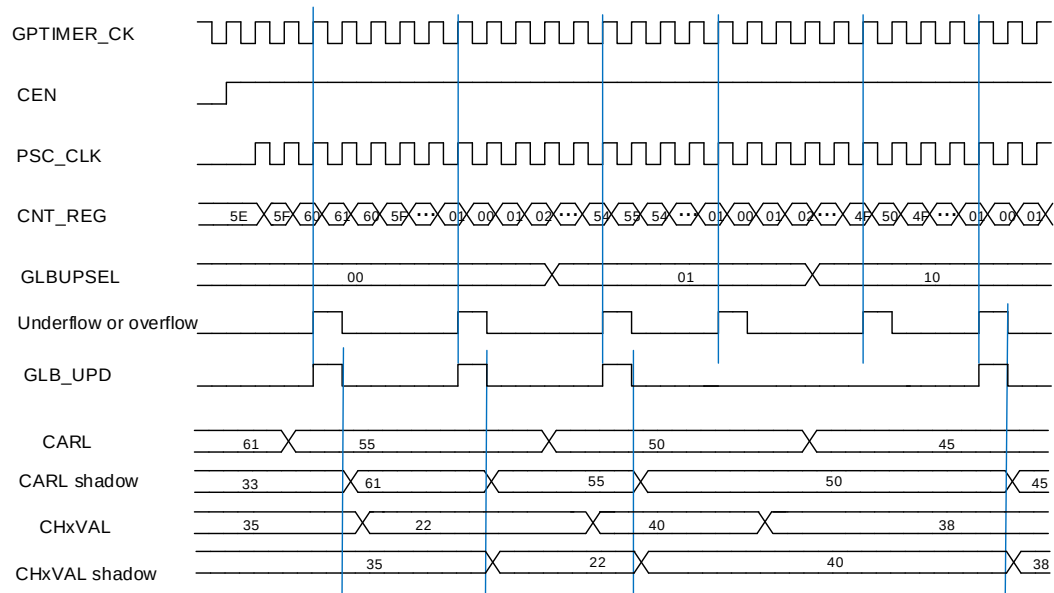
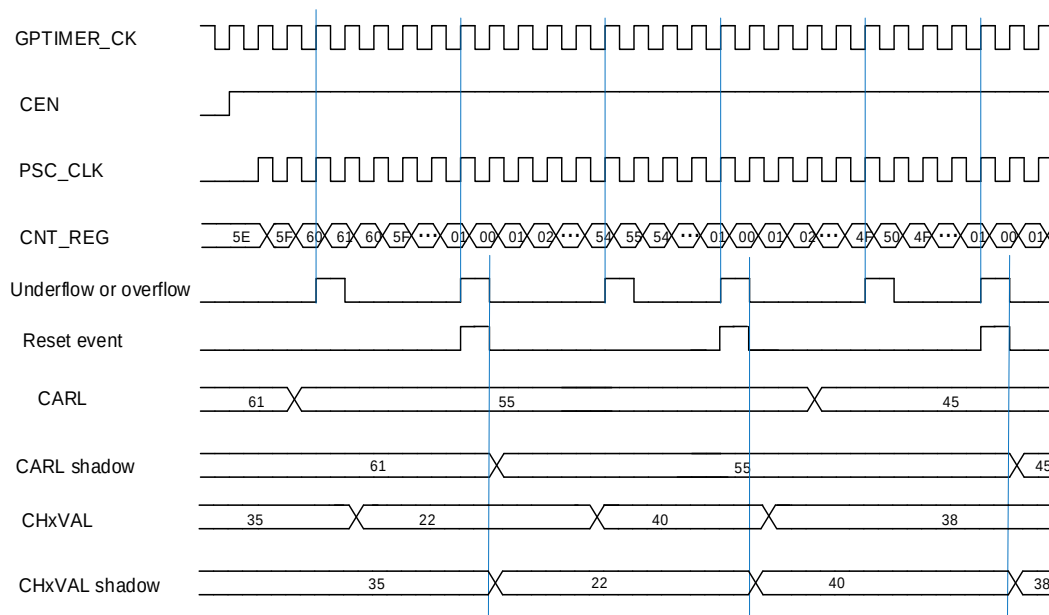


Figure 22-14. Timing diagram for updates in global one-time reload mode on update events ($GLBUPSEL[1:0] = 2'b11$) shows CARL and CHxCV(x = 0..1) update timing diagram in enable global update mode and disable global one-time reload mode, CARL and CHxCV(x = 0..1) are reloaded at the next counter reset event. (CARUPSEL[1:0] = 2'b00, CHxCVUPSEL[1:0] = 2'b00, ARSE = 1 and CHxCOMSEN = 1).

Figure 22-14. Timing diagram for updates in global one-time reload mode on update events (GLBUPSEL[1:0] = 2'b11)



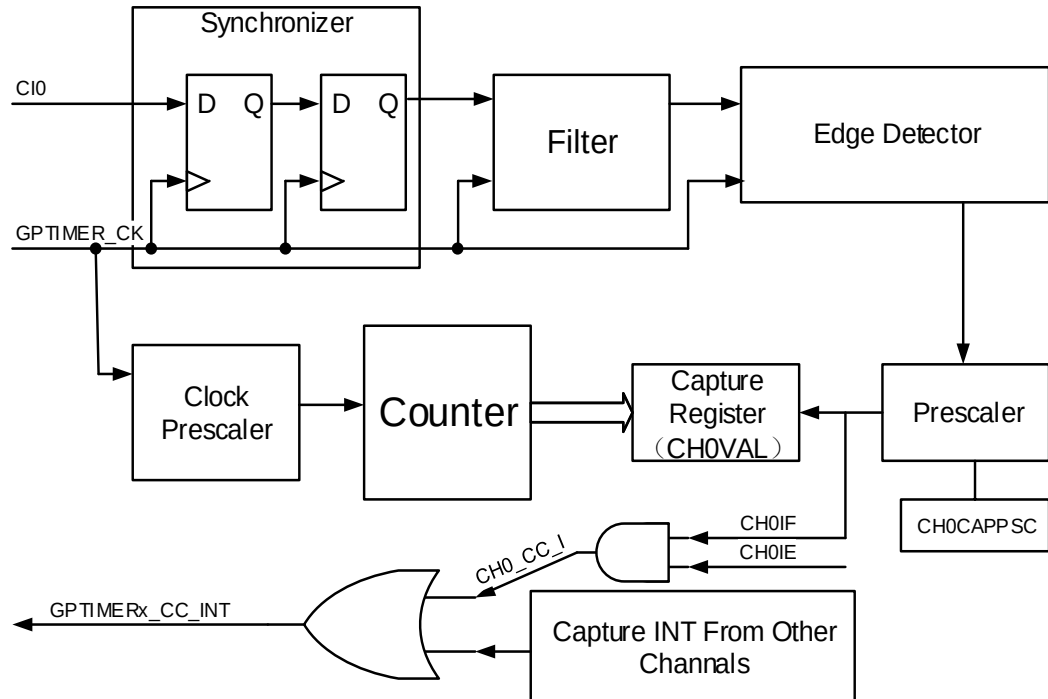
22.4.9 Capture/compare channels

The general purpose timer has 2 independent channels which can be used as capture inputs or compare outputs. Each channel is built around a channel capture compare register including an input stage, a channel controller and an output stage.

Input capture mode

Input capture mode allows the channel to perform measurements such as pulse timing, frequency, period, duty cycle and so on. The input stage consists of a digital filter, edge detection and a channel prescaler. When a selected edge occurs on the channel input, the current value of the counter is captured into the GPTIMERx_CHxCV (x=0,1) registers, at the same time the CHxIF (x=0,1) bit is set and the channel interrupt is generated if it is enabled when CHxIE = 1(x=0,1).

Figure 22-15. Input capture logic for channel 0



The channel input signal source is selected by the GPTIMERx_CH0CSSEL and GPTIMERx_CH1CSSEL registers, including the external input signal ETI0/1/2/3, the channel input signal channel 0/1, and the EVIC event source. ETI and channel can select effective edges (rising edge, falling edge, rising / falling edge).

The input capture prescaler makes several input events generate one effective capture event. On the capture event, GPTIMERx_CHxCV will store the value of counter.

So, the process can be divided into several steps as below:

Step1: Filter configuration (CHxCAPFLT[3:0] bits in GPTIMERx_CHnCTL(x=0,1 and n=0,1) register). Based on the input signal and quality of requested signal, configure CHxCAPFLT[3:0].

Step2: Capture source selection (GPTIMERx_CHnCSSEL register). As soon as selecting one input capture source by GPTIMERx_CHnCSSEL register, the channel must be set to input mode and GPTIMERx_CHxCV cannot be written any more.

Step3: Interrupt enable (CHxIE and CHxDEN bits in GPTIMERx_DMAINTEN). Enable the related interrupt to get the interrupt and DMA request.

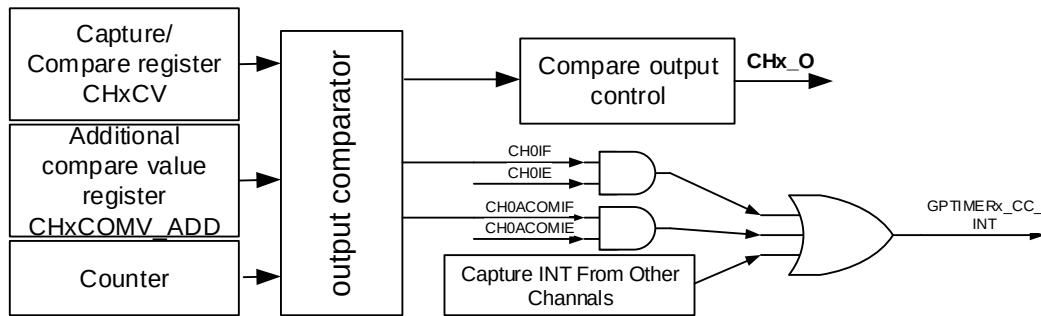
Step4: Capture enable (CHxEN bit in GPTIMERx_CHCTL).

Result: When the wanted input signal is captured, GPTIMERx_CHxCV will be set by counter's value and CHxIF bit is set. The interrupt and DMA request will be set or not based on the configuration of CHxIE and CHxDEN bits in GPTIMERx_DMAINTEN.

Output compare mode

Figure 22-16. Output compare logic show the logic circuit of output compare mode.

Figure 22-16. Output compare logic



In output compare mode, the GPTIMERx can generate timed pulses with programmable position, polarity, duration and frequency. When the counter matches the value in the GPTIMERx_CHxCV / GPTIMERx_CHxCOMV_ADD register of an output compare channel, the channel n output can be set, cleared, maintain or toggled based on GPTIMERx_CHnCTL register related bits. When the counter reaches the value in the GPTIMERx_CHxCV / GPTIMERx_CHxCOMV_ADD register, the CHxIF / CH1COMADDIF bit will be set and the channel (n) interrupt is generated if CHxIE / CHxCOMADDIE = 1. And the DMA request will be asserted, if CHxDEN / CH0COMADDEN = 1.

So, the process can be divided into several steps as below:

Step1: Clock Configuration. Such as clock source, clock prescaler and so on.

Step2: Compare mode configuration.

- Set the shadow enable mode by CHxCOMSEN / CH0COMADDSEN.
- Set the output mode (set/clear/maintain/toggle) by GPTIMERx_CHnCTL[7:0] and GPTIMERx_CHnCTL[18:16] bits.
- Enable the output by CHxEN.

Step3: Interrupt / DMA request enable configuration by CHxIE / CHxCOMADDIE / CHxDEN.

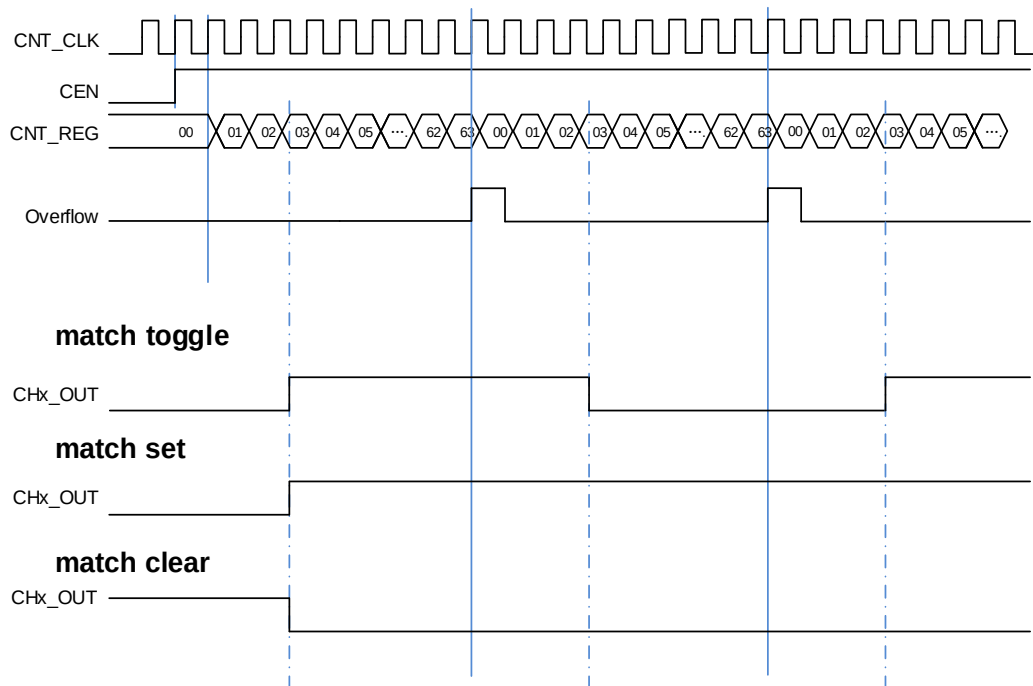
Step4: Compare output timing configuration by GPTIMERx_CAR, GPTIMERx_CHxCV and GPTIMERx_CHxCOMV_ADD.

The GPTIMERx_CHxCV / GPTIMERx_CHxCOMV_ADD can be changed ongoing to meet the expected waveform.

Step5: Start the counter by configuring CEN to 1.

Figure 22-17. Output-compare in three modes shows the three compare modes: toggle/set/clear simple diagram. CARL=0x63, CHxVAL=0x3, CH0COMVAL_ADD is invalid.

Figure 22-17. Output-compare in three modes



22.4.10 PWM mode

In the PWM output mode, the channel can generate PWM waveform according to the GPTIMERx_CAR register and GPTIMERx_CHxCV / GPTIMERx_CHxCOMV_ADD registers.

The value of CHxCV and CHxCOMV_ADD should be in the range of 0-CARL, when CHxCV / CHxCOMV_ADD = 0 or CARL, only one comparison matching is achieved in a cycle, if the value of CHxCV / CHxCOMV_ADD configuration is outside the range, comparison matching does not occur.

According to the CH1MSEL bit in the GPTIMERx_CHCTL register selects the two-channel independent output or complementary output, the dead time is controlled by the GPTIMERx_DTCTL register.

Based on the counter mode, PWM can also be divided into EAPWM (Edge-aligned PWM) and CAPWM (Center-aligned PWM).

If CCSOEN bit in the GPTIMERx_CTL1 register is set, the signal CPPO synchronized with the PWM cycle will be output to the GPIO. The initial value of CPPO is 0, and when the counter is enabled, CPPO becomes 1, and CPPO is flipped when the overflow / underflow / counter is reset. CPPO is only produced when at least one channel is the output mode.

The output level of the channel is controlled by [18:16] bits and [7:0] bits of the GPTIMERx_CHnCTL register, when the counter starts (high level / low level), at the end of the cycle, the CHxCV is compared to match or the CHxCOMV_ADD is compared to match, the output level is set / clear / maintain / toggle. All the commonly used PWM waveforms can be realized through the above configuration, including the PWM mode 0 / 1, composite PWM

mode 0 / 1, asymmetric PWM mode 0 / 1, 0% duty cycle and 100% duty cycle waveforms.

For example, the composite PWM mode period is determined by $2 \times \text{GPTIMERx\_CAR}$ and the duty cycle is determined by $\text{CHxCV} / \text{CHxCOMV\_ADD}$. [Figure 22-18. Timing diagram of composite PWM mode](#) shows the composite PWM0 mode output waveform. The composite PWM0 mode can be implemented by $\text{CH0COMDW}[1:0] = 2'b10$, $\text{CH0COMUP}[1:0] = 2'b01$, $\text{CH0ACOMDW}[1:0] = 2'b01$, $\text{CH0COMUP}[1:0] = 2'b10$ bits in the GPTIMERx_CHnCTL register, and composite PWM1 mode configuration is the similar.

For example, the asymmetric PWM mode period is determined by $2 \times \text{GPTIMERx\_CAR}$, and the duty cycle is determined by $\text{CHxCV} / \text{CHxCOMV\_ADD}$. [Figure 22-19. Timing diagram of asymmetric PWM mode](#) shows the asymmetric PWM mode output waveform. The asymmetric PWM0 mode can be implemented by $\text{CH0COMDW}[1:0] = 2'b01$, $\text{CH0COMUP}[1:0] = 2'b00$, $\text{CH0ACOMDW}[1:0] = 2'b10$, $\text{CH0COMUP}[1:0] = 2'b00$ bits in the GPTIMERx_CHnCTL register, and asymmetric PWM1 mode configuration is the similar.

Customers can also configure different waveforms according to their needs, including edge alignment mode.

Figure 22-18. Timing diagram of composite PWM mode

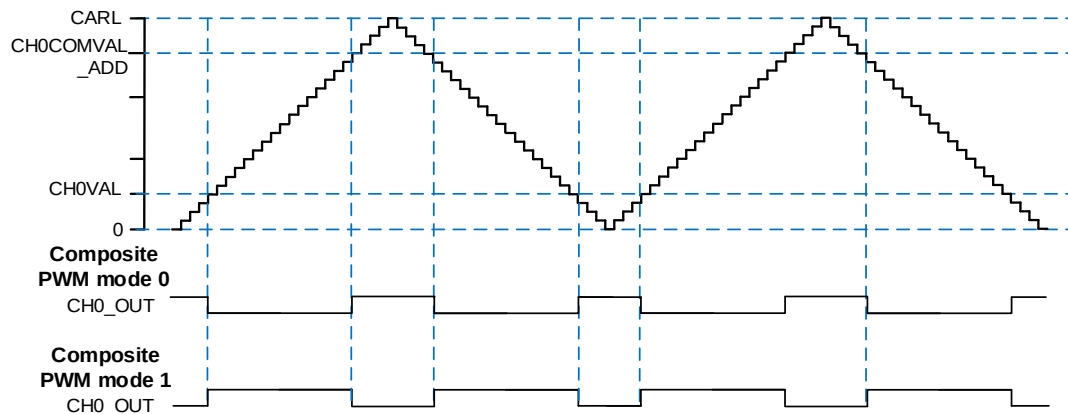
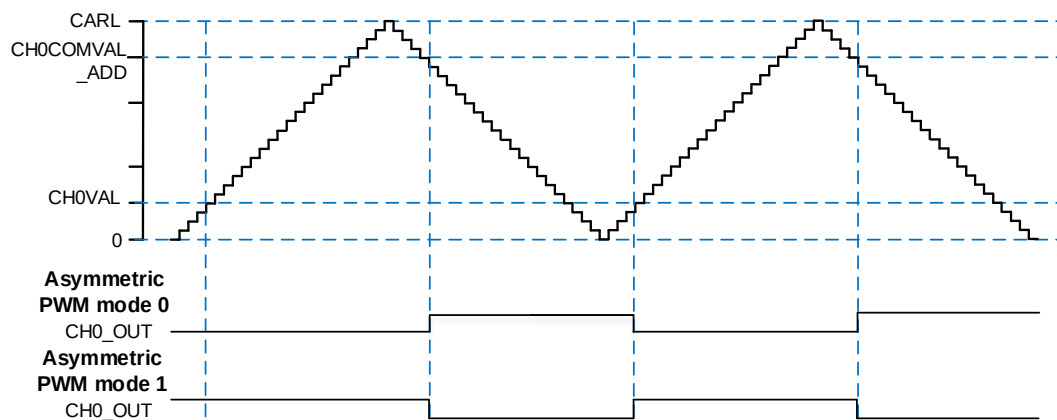


Figure 22-19. Timing diagram of asymmetric PWM mode



22.4.11 Complementary output and dead-time insertion

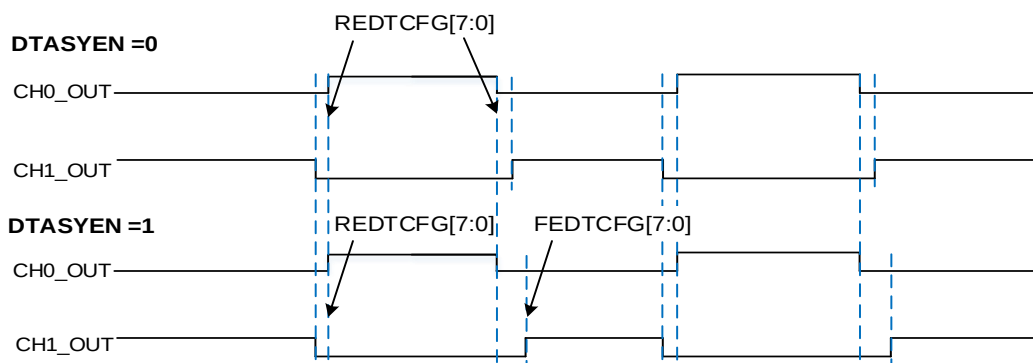
Two-channel output consists of two cases:

- When the CH1MSEL bit is reset in the GPTIMERx_CHCTL register, Channel0 and Channel1 output PWM waveform independently;
- When the CH1MSEL bit is set in the GPTIMERx_CHCTL register, Channel0 and Channel1 cooperate to output the complementary PWM waveform.

When the complementary waveform is output, the two channels can not output the same level at the same time. The CHHOUTDET bit and CHLOUTDET bit is set in the GPTIMERx_DMAINTEN register, enable the channel high / low level detection function at the same time.

Dead time insertion is enabled when the DTEN bit is set in the GPTIMERx_DMAINTEN register. Dead time is inserted to ensure that the complementary output of the two channels is not high / low level at the same time, and the relevant configuration of dead time is implemented by the GPTIMERx_DTCTL register. Whether to use asymmetric dead time for the rising and falling edge can be selected by the DTASYEN bit in the GPTIMERx_DTCTL register. The dead time of the rising and falling edge with DTASYEN =1 is configured with the values of REDTCFG[7:0] bits and FEDTCFG[7:0] bits, respectively. The DTASYEN bit must be configured before the counter is enabled and cannot be modified when CEN=1.

Figure 22-20. Schematic of symmetric and asymmetric dead time insertion



Both dead time registers have shadow registers. When REDTSEN bit and FEDTSEN bit are set, the values of REDTCFG[7:0] bits and FEDTCFG[7:0] bits are preloaded into the corresponding shadow registers. The preloaded values are updated into the registers at the next counter reset event. The current dead time value is applied and the newly configured value is discarded when the shadow register is enabled in the counter counting process.

NOTE: This duty cycle disappears when the dead time is greater than the duty cycle of the output waveform, and CHx_OUT output low level.

22.4.12 Duty cycle 0% or 100% output

A waveform with a duty cycle of 0% or 100% for the channel output can be defined through

the configuration CHxOUTM[1:0] bits in the GPTIMERx_CHnCTL register.

In edge-aligned mode, if the CHxOUTM[1:0] bits are changed during counting, the output waveform duty cycle is changed during overflow (up count) or underflow (down count). When the counter stops, CHxOUTCTL = 0 and CHxOUTM[1:0] bits is changed, the output waveform duty cycle is changed in the overflow (up count) or underflow (down count) after the start of the count; When the counter stops, CHxOUTCTL = 1, and CHxOUTM[1:0] bits is changed, the output waveform duty cycle is changed immediately after the start count.

In central alignment mode, if the CHxOUTM[1:0] bits are changed during counting, the output waveform duty cycle is changed during underflow. When the counter stops, CHxOUTCTL = 0, and CHxOUTM[1:0] bits is changed, the duty cycle of the output waveform is changed in the underflow after the start of the count; When the counter stops, CHxOUTCTL = 1, and CHxOUTM[1:0] bits, the output waveform duty cycle is changed immediately after the start count.

When the 0% or 100% waveform is output, the comparison matching interrupt is still generated.

When the duty cycle of the output waveform is released from 0% or 100%, the output level value is determined by the CHxREDUTY bit and CHxENDOUT[1:0] bits in the GPTIMERx_CHnCTL register. When CHxENDOUT[1:0]=2'b01, the low level is output at the end of the cycle. When CHxENDOUT[1:0]=2'b10, the high level is output at the end of the cycle. When CHxENDOUT[1:0]=2'b00 or 2'b11, the low level is maintained or toggled at the end of the cycle, and benchmark is selected by CHxREDUTY. With 0% or 100% as the benchmark when CHxREDUTY = 0, and the result of the last comparison match as the benchmark when CHxREDUTY = 1. As is shown in the following [Table 22-1. Schematic diagram of the waveform output](#):

Table 22-1. Schematic diagram of the waveform output

CHxENDOUT[1:0]	The comparison matching level covered by duty cycle 0% or 100%	CHxREDUTY at 0% duty cycle		CHxREDUTY at 100% duty cycle	
		0	1	0	1
00: Output level is maintained	0	0	0	1	0
	1	0	1	1	1
01: Output low level	-	0	0	0	0
10: Output high level	-	1	1	1	1
11: Output level toggle	0	1	1	0	1
	1	1	0	0	0

Note: If the two channels complementary output and enable dead time insertion, inserting dead time will lead to the duty cycle of the final output is not 100% or 0%, need to pay attention to when using.

22.4.13 Synchronization between GPTIMERx

Software source synchronization

The synchronous enable/disable/reset behavior of the counters between GPTIMERx can be specified by simultaneously configuring the bit of the corresponding GPTIMER number to 1 in the GPTIMERx_SWEN / GPTIMERx_SWDIS / GPTIMERx_SWRST register. The phase shift can be achieved by presetting the CNT value before the counter is started. As is shown in [Figure 22-21. Simultaneous operation by software with no initial value](#) and [Figure 22-22. Simultaneous operation by software with initial value](#):

Figure 22-21. Simultaneous operation by software with no initial value

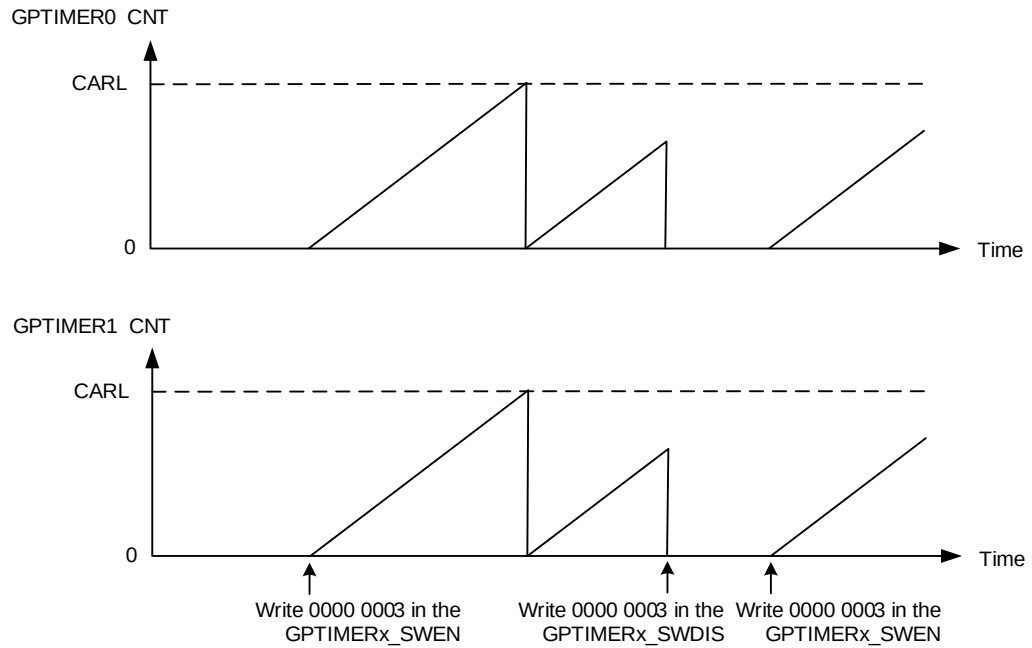
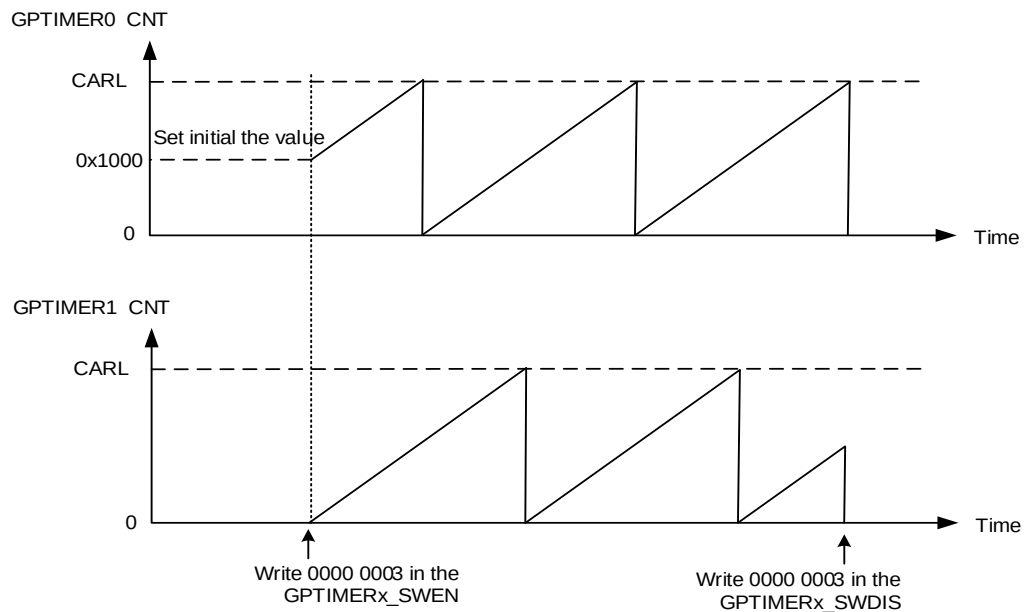


Figure 22-22. Simultaneous operation by software with initial value

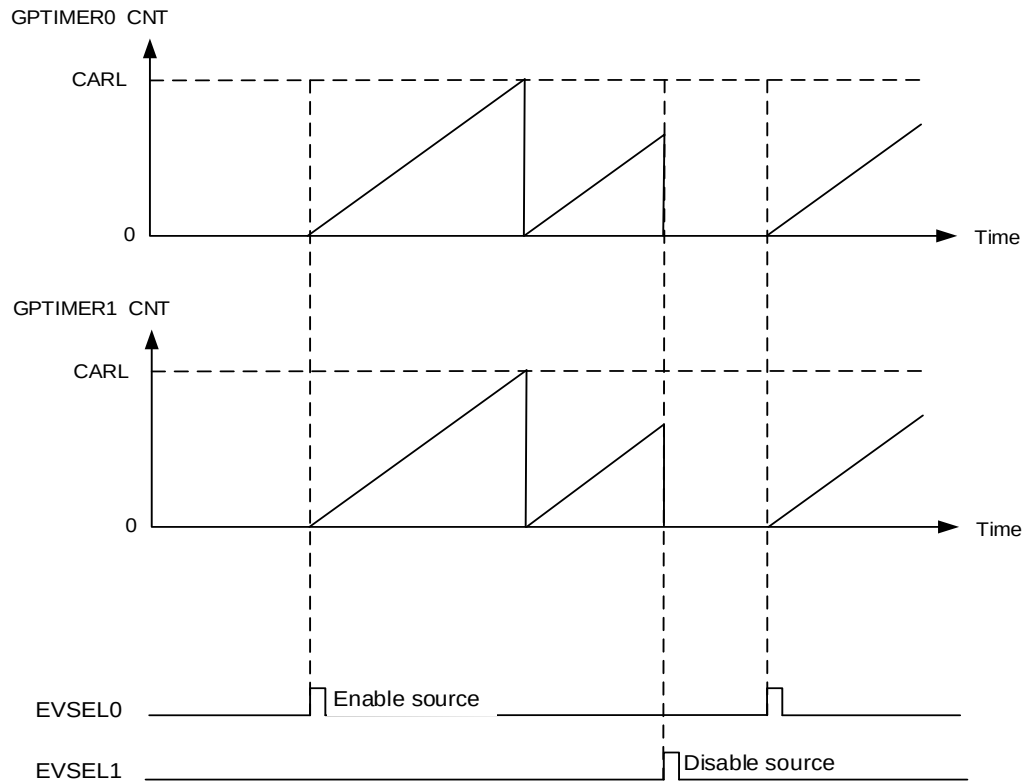


Note: If the GPTIMER clock prescale factor of the synchronization operation is different, the effective time of the counter behavior is also different.

Hardware source synchronization

Hardware synchronization source include: ETI input, EVIC input, channel input. The synchronization behavior triggered by the channel input is implemented by setting the channel input capture as an EVIC source. As is shown in [Figure 22-23. Simultaneous operation by hardware source](#).

Figure 22-23. Simultaneous operation by hardware source

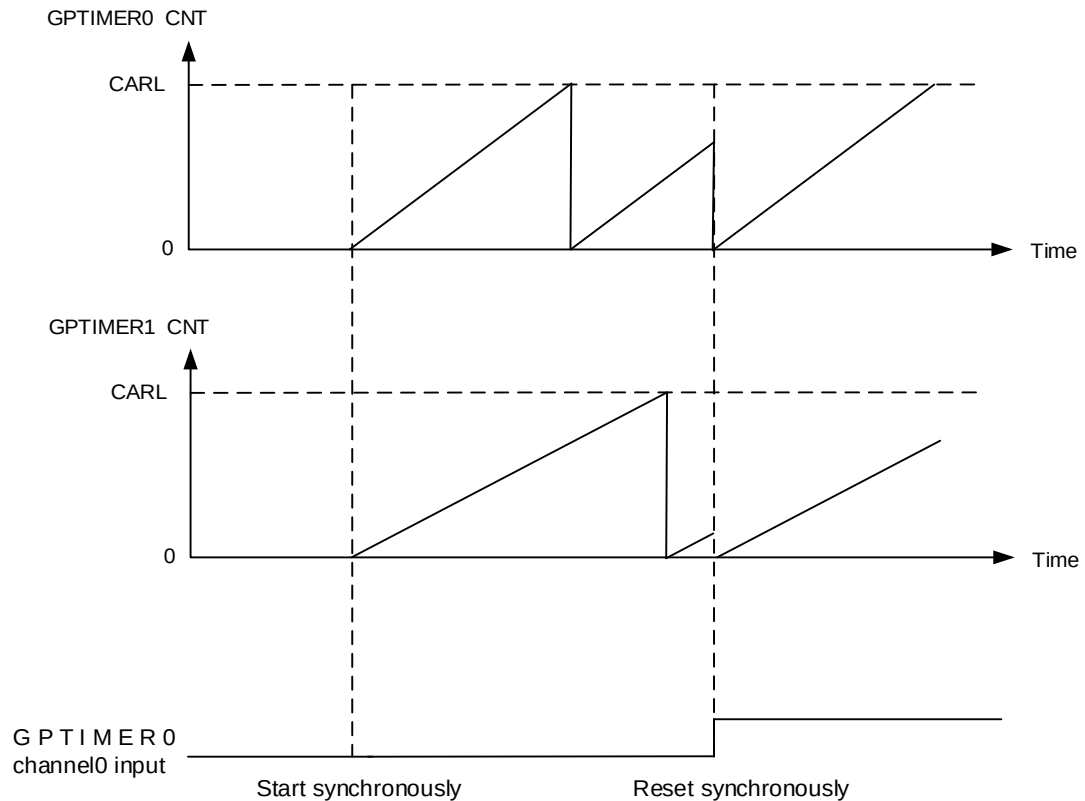


Synchronous reset

Counter reset behavior of one GPTIMER can trigger synchronous reset of other GPTIMER, and counter reset can be triggered by comparison matching, input capture, overflow / underflow, or input source selected in the GPTIMERx_RSSEL register. Setting the GPTIMER that generates the synchronous reset trigger and the GPTIMER that needs to be synchronized reset in the same group can also achieve synchronous reset behavior between GPTIMER (selecting the same set, the SYNCSEL bit in the GPTIMERx_SYNRSTCTL register are configured in the corresponding GPTIMER). The register configuration is as follows:

- GPTIMER reset source: configure GPTIMERx_SYNRSTCTL register, select synchronous reset source;
- Synchronized GPTIMER: configure CSIRSTCNT [2:0] = 3'b101 in the GPTIMERx_RSSEL register;
- Setting the source GPTIMER and target GPTIMER to the same set: the SYNCSEL bit in the GPTIMERx_SYNRSTCTL register are configured in the corresponding GPTIMER.

When the source GPTIMER produces a reset event, the target GPTIMER is reset simultaneously.

Figure 22-24. Simultaneous reset source by channel

22.4.14 Single pulse mode

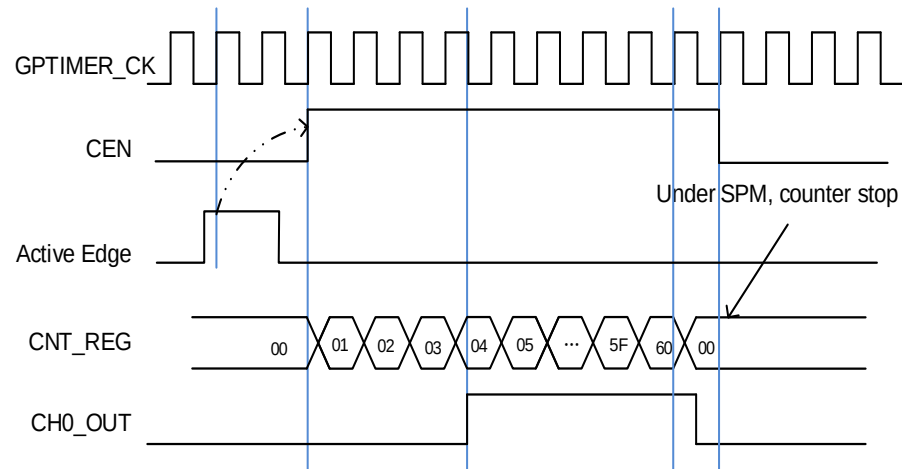
Single pulse mode is opposite to the repetitive mode, which can be enabled by setting SPM in GPTIMERx_CTL0. If SPM is set, the counter will be cleared and stopped automatically when the next update event occurs.

Once the timer is set to the single pulse mode, setting the CEN bit to 1 or a trigger signal edge can generate a pulse and then keep the CEN bit at a high state until the update event occurs or the CEN bit is written to 0 by software. If the CEN bit is cleared to 0 by software, the counter will be stopped and its value will be held. If the CEN bit is automatically cleared to 0 by a hardware / software update event, the counter will be reinitialized. The hardware off CEN at the end of the cycle has the highest priority

In the single pulse mode, the active edge of trigger which sets the CEN bit to 1 will enable the counter. After a trigger rising occurs in the single pulse mode, when the count value compare the CHxCV value, the CHx_OUT will change immediately according to the channel output configuration. As is shown in [Figure 22-25. Single pulse mode GPTIMERx CHxCV=0x04, GPTIMERx CAR=0x60, CH0COMUP\[1:0\] = 2'b10.](#)

Figure 22-25. Single pulse mode GPTIMERx_CHxCV=0x04, GPTIMERx_CAR=0x60,

CH0COMUP[1:0] = 2'b10



22.4.15 Interrupt skipping function

Interrupt skipping

The number of counter overflow / underflow interrupts to skip can be set by configuring the GPTIMERx_IADCTSS register. CHxCV capture / compare event interrupts, CHxCOMV_ADD compare event interrupts and ADC trigger request signals can be linked to the overflow / underflow interrupt skipping function to implement the interrupt skipping function.

The skipped interrupt does not update the corresponding flag bit in register GPTIMERx_INTF. The interrupt skipping function is only related to the register GPTIMERx_IADCTSS and is independent of the interrupt enable bit in GPTIMERx_DMAINTEN, which only controls whether the interrupt signal that is not skipped is enabled or not.

The overflow (count up), underflow (count down) in edge-aligned mode to be skipped and the overflow / underflow of the center alignment mode are selected by OUFINTSSEL[1:0] bits in the GPTIMERx_IADCTSS register, and the selection of the number of skips should be used in conjunction with the interrupt source selection of the skip. This is especially important when changing the count direction in edge-aligned mode. As is shown in [Figure 22-26. Example of interrupt skipping in center alignment mode \(skip value is 2\)](#) and [Figure 22-27. Example of interrupt skipping in edge alignment mode \(changing direction and skip value is 2\)](#).

This skipping counter is reset when the CNT stops counting. The interrupt skipping value can only be changed when the interrupt skipping counter stops (OUFINTSSEL = 2'b00).

Figure 22-26. Example of interrupt skipping in center alignment mode (skip value is 2)

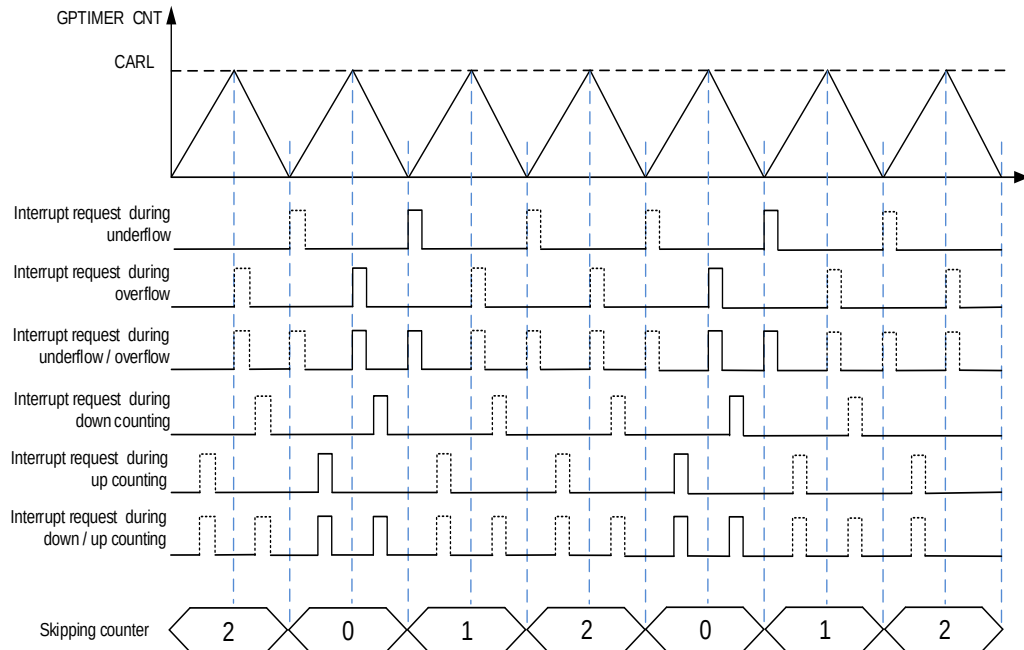
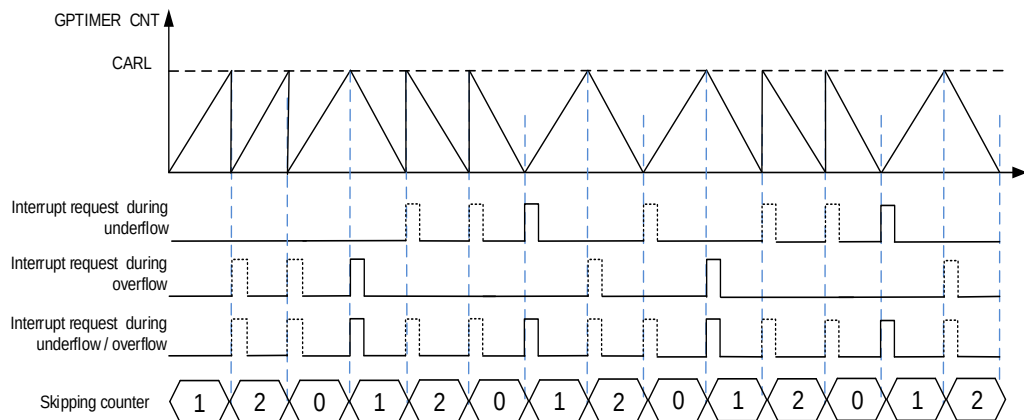


Figure 22-27. Example of interrupt skipping in edge alignment mode (changing direction and skip value is 2)



Additional interrupt skipping

The additional interrupt skipping function is implemented by configuring GPTIMERx_ADDINTSCTL0 and GPTIMERx_ADDINTSCTL1 registers. Additional interrupt skipping has two independent repeat counters, through the GPTIMERx_ADDINTSCTL0 register can select the count source of the two repeat counters, the initial value and the number of skip, the additional interrupt skipping repeat counter counts the overflow / underflow interrupt of the GPTIMER counter. Both counters are incremented.

CHxCV capture / compare event interrupt, CHxCOMV_ADD compare event interrupt, ADC trigger request signal and the extended skipping function of overflow / underflow interrupt of the GPTIMER counter are selected through the GPTIMERx_ADDINTSCTL1 register. The EVIC event corresponding to the skipped interrupt will also be skipped. The skipped interrupt

does not update the corresponding flag bit in register GPTIMERx_INTF. The interrupt skipping function is independent of the interrupt enable bit in GPTIMERx_DMAINTEN, which only controls whether the interrupt signal that is not skipped is enabled or not.

When CNT stops counting, the two additional interrupt skipping count keep their current values. The additional interrupt skipping value can only be changed when the additional interrupt skipping counter stops (ADDCxSKSEL[1:0] = 2'b00). The first count clock after the no-skipping state changes to the skipping state (ADDCxSKSEL[1:0] changes from 00 to some other value), ADDCxVAL[2:0]+1. The initial value register ADDCxIVAL[2:0] can only be written if ADDCxSKSEL[1:0] ≠ 2'b00 and is not reset when ADDCxSKSEL[1:0] = 2'b00 is written.

ADC trigger request interrupt skipping

The skipping function of ADC trigger request is implemented by configuration registers GPTIMERx_ADCTL and GPTIMERx_ADCTRGS. ADC trigger request skipping has two independent repeat counters, through the register GPTIMERx_ADCTRGS can select the count source of two repeat counters, initial value and skip times, ADC trigger request skipping repeat counter to ADTCV1/2[15:0] comparison match event count.

When the CNT stops counting, the skipping counters to keep their current values. The ADC trigger request skipping value can only be changed when the ADC trigger request skipping counter stops (ADCCxSKSEL[1:0]=00). ADCC2VAL[1:0]+1 is the first count clock after the ADC triggers the request from never skipping to skipping (ADCCxSKSEL[1:0] changes from 00 to some other value). The initial value register ADCCxIVAL[2:0] can only be written if ADCCxSKSEL[1:0] is configured from 00 to some other value and is not reset when ADCCxSKSEL[1:0]=00 is written.

22.4.16 ADC trigger request

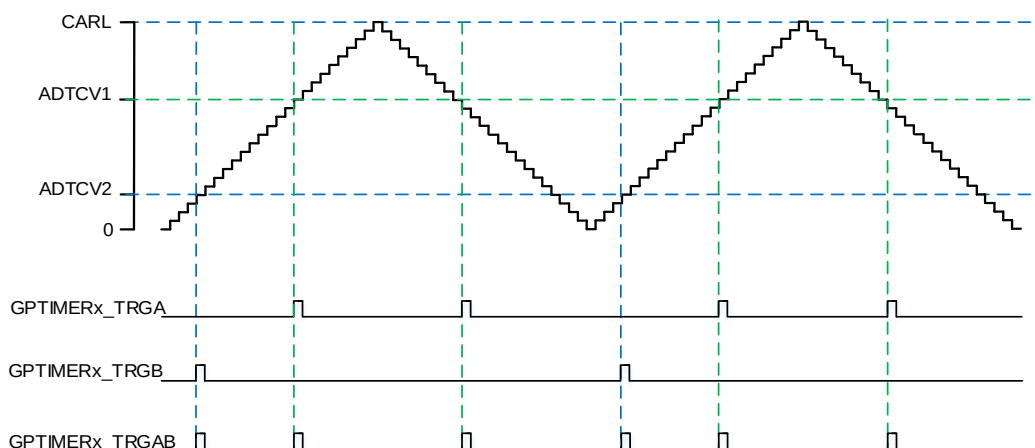
ADC compare match trigger mode

The ADC trigger request signal can trigger the ADC individually or in combination through the GPTIMER comparison match event. No ADC trigger request can be generated in event count mode.

The general timer has two ADC trigger comparison value registers (GPTIMERx_ADCCRy, x=0 or 1, y=1 or 2) that generate ADC trigger request signals GPTIMERx_TRGA and GPTIMERx_TRGB when a comparison match occurs. The matching enable registers (ADT1DWGTEN, ADT1UPGTEN, ADT2DWGTEN, ADT2UPGTEN) can be configured to generate or not generate the ADC trigger request signal during the up-count, down-count, up-count + down-count process. As is shown in [Figure 22-28. ADC trigger singal operation in ADC compare match trigger mode\(ADT1UPGTEN=1 、 ADT2UPGTEN=1 、 ADT1DWGTEN=1、 ADT2DWGTEN=0\).](#)

Figure 22-28. ADC trigger singal operation in ADC compare match trigger

mode(ADT1UPGTEN=1、ADT2UPGTEN=1、ADT1DWGTEN=1、ADT2DWGTEN=0)



The ADTCV1/2[15:0] bits of the GPTIMERx_ADCCRy registers has corresponding shadow register and according to the relevant configuration of the shadow register, the comparison register is updated. In central alignment mode, the ADC triggers updates to the comparison value register that are controlled by the ADCCRnUPSEL[1:0] bits in the GPTIMERx_UPSSEL register.

Table 22-2. GPTIMER0/1 ADC trigger signals and their operation

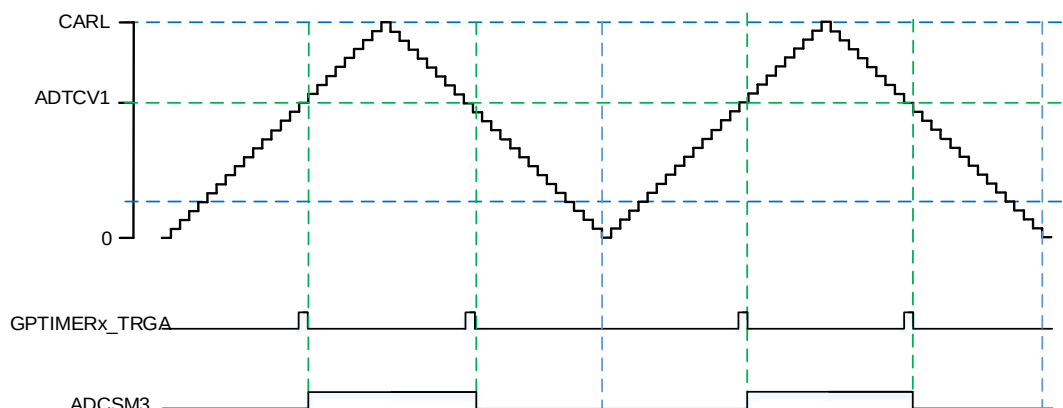
ADC Trigger signal	Generation mode
GPTIMERx_TRGA	The ADC trigger compare value 1 (ADTCV1) matches
GPTIMERx_TRGB	The ADC trigger compare value 2 (ADTCV2) matches
GPTIMERx_TRGAB	The ADC trigger compare value 1 or 2(ADTCV1 or ADTCV2) matches

Monitor the ADC trigger request signal

The external pin ADCSM3 / 4 is used to monitor the ADC trigger request signal. Through the ADCSM3EN / ADCSM4EN bit in the GPTIMERx_ADCTL register, choose whether the two monitoring outputs ADCSM3 / 4 are enabled. Through the ADCMS3SEL / ADCMS4SEL bit, select the ADC to be monitored by ADCSM3 / 4 to trigger the request, and ADCSM3 / 4 outputs to the external PIN. When the enable is turned off, the output is low and the monitoring signal is flipped once every time the ADC trigger request is generated. As is shown in [Figure 22-29. Timing diagram of ADC trigger synchronization monitor signal \(ADCSM3SEL = 0\).](#)

Figure 22-29. Timing diagram of ADC trigger synchronization monitor signal

(ADCSM3SEL = 0)



22.4.17 Interaction with the EVIC

EVIC events output by GPTIMER

GPTIMER can interact with other modules through EVIC. The interrupt signal output to EVIC is not affected by the corresponding enable bit in GPTIMERx_DMAINTEN register. The EVIC events output by GPTIMER are as follows:

- CH0CV compare match events
- CH1CV compare match events
- CH0COMV_ADD compare match events
- CH1COMV_ADD compare match events
- GPTIMER counter overflow event
- GPTIMER counter underflow event
- ADC trigger event generated by ADTCV1 matching
- ADC trigger event generated by ADTCV2 matching
- End of repeat count event

GPTIMER receives the EVIC event

GPTIMER receives the EVIC events have eight, as the EVSELn (n=0...7), according to the configuration register, GPTIMER on its behavior is as follows:

- Counter enable / disable / reset
- Count up / count down;
- Input capture.

22.4.18 Protection function

Register write protection

To prevent miswriting registers, all registers have a write protection function, which is implemented by the GPTIMERx_WP register. The three registers GPTIMERx_SWEN, GPTIMERx_SWDIS and GPTIMERx_SWRST have independent bits SWEWPEN,

SWDWPEN and SWRWPEN for write protection. All remaining registers are write-protected by WPEN bit.

To change whether write protection is enabled or not, the specified sequence "0xA5" is written to WPKEY[15:8] bits.

Stop output protection

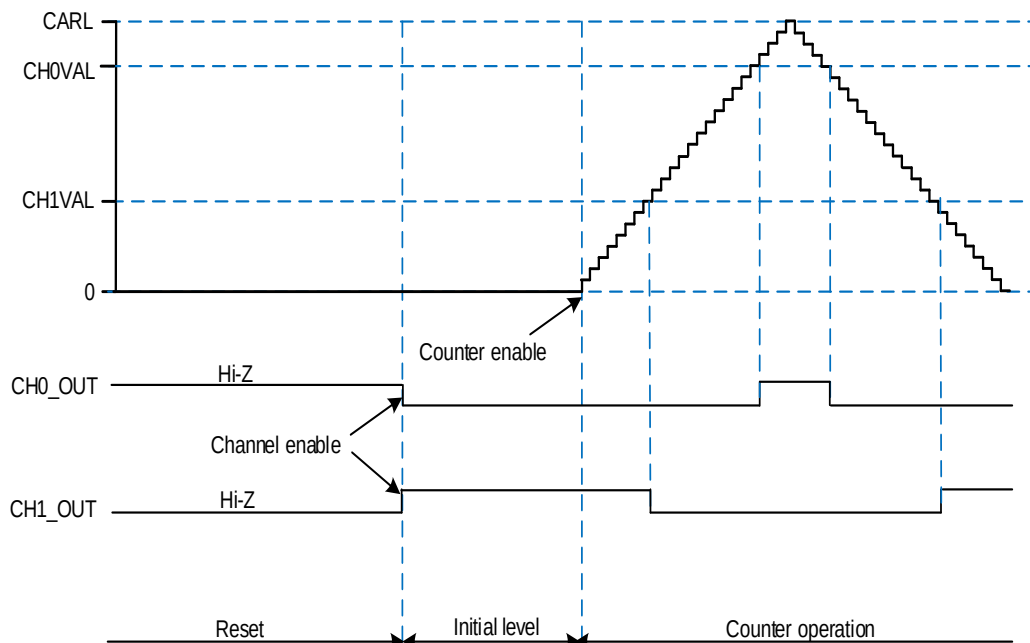
The output of the GPTIMER can be forced to stop on the port when a stop output request is received from the GTOC. When two channels output the same level signal at the same time, the output stop protection is useful. The output level of the GPTIMER upon receipt of the stop output request can be selected through the bits CH0INVSOUT[1:0] in the GPTIMERx_CH0CTL and CH1INVSOUT[1:0] in the GPTIMERx_CH1CTL register.

When the stop output request is released at the end of the cycle, the stop output takes effect until the end of the next counting cycle.

22.4.19 Initialization of the output pin

The GPTIMER registers are initialized after reset, and the output port is initialized by CHnENDOUT[1:0] bits and CHnENOUT bit in the GPTIMERx_CHnCTL register. As is shown in [Figure22-30. Channel output after reset](#), CHnENDOUT = 2'b11, CH0ENOUT = 0, CH1ENOUT = 1.

Figure22-30. Channel output after reset



If an error occurs while GPTIMER is working, there are several ways to set the output:

- Set the CHnOUT bit in the GPTIMERx_CHnCTL register and keep the output at the current level.
- Reset the CHnOUT bit in the GPTIMERx_CHnCTL register and the output level when the counter stops is selected by the CHnDISOUT bit in the GPTIMERx_CHnCTL register.

- Turn off the channel enable bit CHnEN in the GPTIMERx_CHCTL register and set the output level through GPIO;
- Output Hi-Z configuration via GTOC.

22.4.20 Timer DMA mode

Timer DMA mode is the function that configures timer's register by DMA module. The relative registers are GPTIMERx_DMACFG and GPTIMERx_DMATB. Corresponding DMA request bit should be asserted to enable DMA request for internal interrupt event. GPTIMERx will send a request to DMA when the interrupt event occurs. DMA is configured to M2P (memory to peripheral) mode and the address of GPTIMERx_DMATB is configured to PADDR (peripheral base address), then DMA will access the GPTIMERx_DMATB. In fact, GPTIMERx_DMATB register is only a buffer, timer will map the GPTIMERx_DMATB to an internal register, appointed by the field of DMATA in GPTIMERx_DMACFG. If the field of DMATC in GPTIMERx_DMACFG is 0 (1 transfer), the timer sends only one DMA request. While if DMATC is not 0, such as 3 (4 transfers), then timer will send 3 more requests to DMA, and DMA will access timer's registers DMATA+0x4, DMATA+0x8 and DMATA+0xC at the next 3 accesses to GPTIMERx_DMATB. In a word, one-time DMA internal interrupt event asserts, (DMATC+1) times request will be sent by GPTIMERx.

If one more DMA request event occurs, GPTIMERx will repeat the process above.

22.4.21 Timer debug mode

When the Cortex®-M33 is halted, and the GPTIMERx_HOLD configuration bit in DBG_CTL register is set to 1, the GPTIMERx counter stops.

22.5 Registers definition (GPTIMERx, x=0, 1)

GPTIMER0 base address: 0x4001 6000

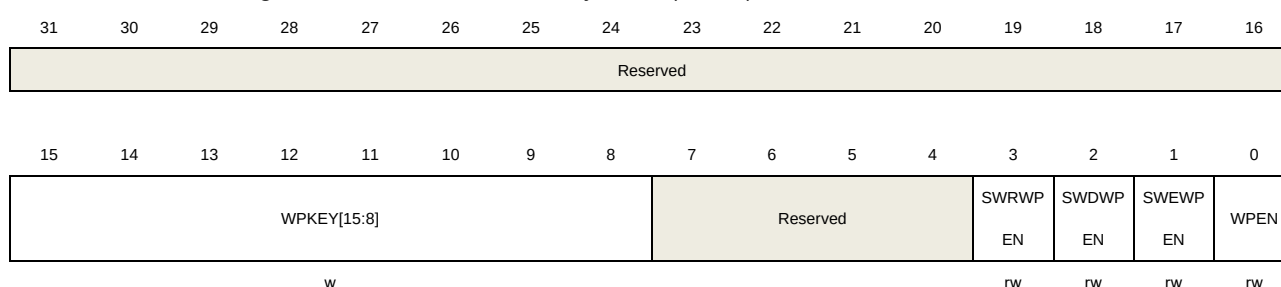
GPTIMER1 base address: 0x4001 6100

22.5.1 Write-protect register (GPTIMERx_WP)

Address offset: 0x00

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



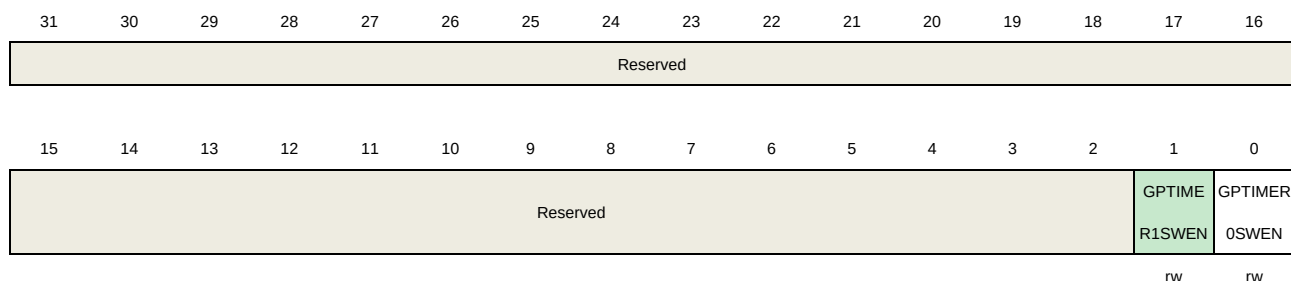
Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:8	WPKEY[15:8]	Write protection key If you want to modify WPEN, SWEWPEN, SWDWPEN, SWRWPEN bit, this bit field writes 0xA5 at the same time.
7:4	Reserved	Must be kept at reset value.
3	SWRWPEN	GPTIMERx_SWRST write protection enable 0: Write protection disable 1: Write protection enable
2	SWDWPEN	GPTIMERx_SWDIS write protection enable 0: Write protection disable 1: Write protection enable
1	SWEWPEN	GPTIMERx_SWEN write protection enable 0: Write protection disable 1: Write protection enable
0	WPEN	Register write protection enable The target registers are all the registers except SWEWPEN, SWDWPEN and SWRWPEN protection registers. 0: Write protection disable 1: Write protection enable

22.5.2 Software enable register (GPTIMERx_SWEN)

Address offset: 0x04

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



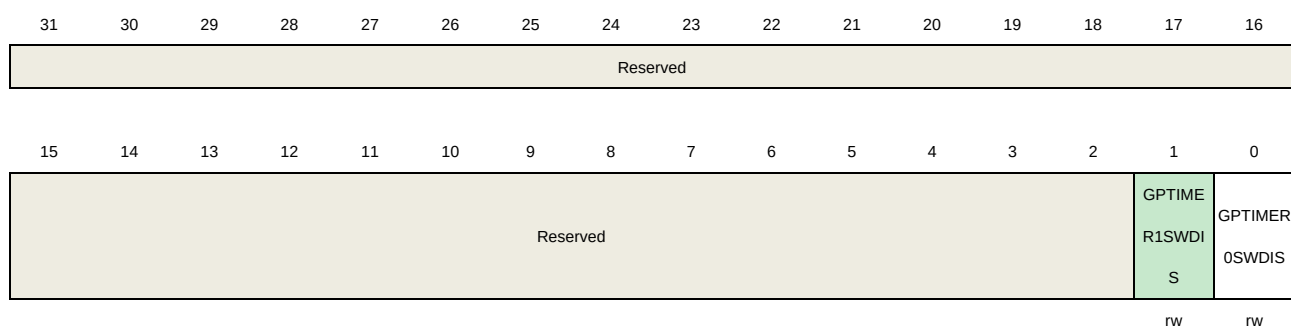
Bits	Fields	Descriptions
31:2	Reserved	Must be kept at reset value.
1	GPTIMER1SWEN	GPTIMER1 counter software enable Refer to GPTIMER0SWEN description.
0	GPTIMER0SWEN	GPTIMER0 counter software enable When reading the bit: 0: The counter is stopped 1: The counter is working When writing the bit: 0: No effect 1: The counter starts working

22.5.3 Software disable register (GPTIMERx_SWDIS)

Address offset: 0x08

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



Bits	Fields	Descriptions

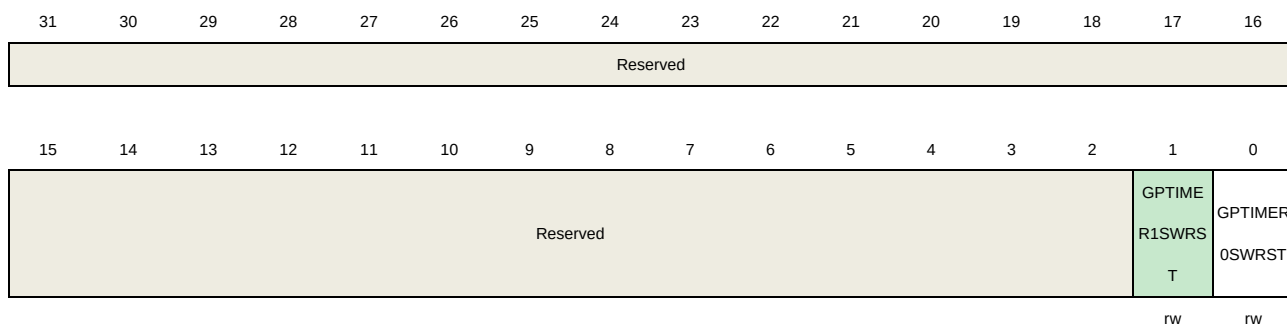
31:2	Reserved	Must be kept at reset value.
1	GPTIMER1SWDIS	GPTIMER1 counter software disable Refer to GPTIMER0SWDIS description.
0	GPTIMER0SWDIS	GPTIMER0 counter software disable When reading the bit: 0: The counter is working 1: The counter is stopped When writing the bit: 0: No effect 1: The counter is stopped

22.5.4 Software reset register (GPTIMERx_SWRST)

Address offset: 0x0C

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



Bits	Fields	Descriptions
31:2	Reserved	Must be kept at reset value.
1	GPTIMER1SWRST	GPTIMER1 counter software reset Refer to GPTIMER0SWRST description.
0	GPTIMER0SWRST	GPTIMER0 counter software reset 0: No effect 1: The counter reset

22.5.5 Counter enable source selection register (GPTIMERx_ESSEL)

Address offset: 0x10

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



SWENCN	Reserved							EVSEL7E	EVSEL6E	EVSEL5E	EVSEL4E	EVSEL3E	EVSEL2E	EVSEL1E	EVSEL0E
T								NCNT	NCNT	NCNT	NCNT	NCNT	NCNT	NCNT	NCNT
rw								rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				CH1ENCNT[1:0]		CH0ENCNT[1:0]		ETI3ENCNT[1:0]		ETI2ENCNT[1:0]		ETI1ENCNT[1:0]		ETI0ENCNT[1:0]	
				rw		rw		rw		rw		rw		rw	

Bits	Fields	Descriptions
31	SWENCNT	The software signal enable counter function enable 0: Disable 1: Enable
30:24	Reserved	Must be kept at reset value.
23	EVSEL7ENCNT	The EVSEL7 signal enable counter function enable Refer to EVSEL0ENCNT description.
22	EVSEL6ENCNT	The EVSEL6 signal enable counter function enable Refer to EVSEL0ENCNT description.
21	EVSEL5ENCNT	The EVSEL5 signal enable counter function enable Refer to EVSEL0ENCNT description.
20	EVSEL4ENCNT	The EVSEL4 signal enable counter function enable Refer to EVSEL0ENCNT description.
19	EVSEL3ENCNT	The EVSEL3 signal enable counter function enable Refer to EVSEL0ENCNT description.
18	EVSEL2ENCNT	The EVSEL2 signal enable counter function enable Refer to EVSEL0ENCNT description.
17	EVSEL1ENCNT	The EVSEL1 signal enable counter function enable Refer to EVSEL0ENCNT description.
16	EVSEL0ENCNT	The EVSEL0 signal enable counter function enable 0: Disable 1: Enable
15:12	Reserved	Must be kept at reset value.
11:10	CH1ENCNT[1:0]	The channel 1 signal edge enable counter function enable Refer to CH0ENCNT[1:0] description.
9:8	CH0ENCNT[1:0]	The channel 0 signal edge enable counter function enable 00: Disable 01: Rising edge 10: Falling edge 11: Both rising and falling edges

7:6	ETI3ENCNT[1:0]	The ETI3 signal edge enable counter function enable Refer to ETI0ENCNT[1:0] description.
5:4	ETI2ENCNT[1:0]	The ETI2 signal edge enable counter function enable Refer to ETI0ENCNT[1:0] description.
3:2	ETI1ENCNT[1:0]	The ETI1 signal edge enable counter function enable Refer to ETI0ENCNT[1:0] description.
1:0	ETI0ENCNT[1:0]	The ETI0 signal edge enable counter function enable 00: Disable 01: Rising edge 10: Falling edge 11: Both rising and falling edges

22.5.6 Counter disable source selection register (GPTIMERx_DSSEL)

Address offset: 0x14

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SWDISC NT	Reserved							EVSEL7D ISCNT	EVSEL6D ISCNT	EVSEL5D ISCNT	EVSEL4D ISCNT	EVSEL3D ISCNT	EVSEL2D ISCNT	EVSEL1D ISCNT	EVSEL0D ISCNT
rw								rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				CH1DISCNT[1:0]		CH0DISCNT[1:0]		ETI3DISCNT[1:0]		ETI2DISCNT[1:0]		ETI1DISCNT[1:0]		ETI0DISCNT[1:0]	
				rw		rw		rw		rw		rw		rw	

Bits	Fields	Descriptions
31	SWDISCNT	The software signal disable counter function enable 0: Disable 1: Enable
30:24	Reserved	Must be kept at reset value.
23	EVSEL7DISCNT	The EVSEL7 signal disable counter function enable Refer to EVSEL0DISCNT description.
22	EVSEL6DISCNT	The EVSEL6 signal disable counter function enable Refer to EVSEL0DISCNT description.
21	EVSEL5DISCNT	The EVSEL5 signal disable counter function enable Refer to EVSEL0DISCNT description.
20	EVSEL4DISCNT	The EVSEL4 signal disable counter function enable Refer to EVSEL0DISCNT description.

19	EVSEL3DISCNT	The EVSEL3 signal disable counter function enable Refer to EVSEL0DISCNT description.
18	EVSEL2DISCNT	The EVSEL2 signal disable counter function enable Refer to EVSEL0DISCNT description.
17	EVSEL1DISCNT	The EVSEL1 signal disable counter function enable Refer to EVSEL0DISCNT description.
16	EVSEL0DISCNT	The EVSEL0 signal disable counter function enable 0: Disable 1: Enable
15:12	Reserved	Must be kept at reset value.
11:10	CH1DISCNT[1:0]	The channel 1 signal edge disable counter function enable Refer to CH0DISCNT[1:0] description.
9:8	CH0DISCNT[1:0]	The channel 0 signal edge disable counter function enable 00: Disable 01: Rising edge 10: Falling edge 11: Both rising and falling edges
7:6	ETI3DISCNT[1:0]	The ETI3 signal edge disable counter function enable Refer to ETI0DISCNT[1:0] description.
5:4	ETI2DISCNT[1:0]	The ETI2 signal edge disable counter function enable Refer to ETI0DISCNT[1:0] description.
3:2	ETI1DISCNT[1:0]	The ETI1 signal edge disable counter function enable Refer to ETI0DISCNT[1:0] description.
1:0	ETI0DISCNT[1:0]	The ETI0 signal edge disable counter function enable 00: Disable 01: Rising edge 10: Falling edge 11: Both rising and falling edges

22.5.7 Counter reset source selection register (GPTIMERx_RSSEL)

Address offset: 0x18

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
SWRSTC	Reserved					CSIRSTCNT[2:0]			EVSEL7R	EVSEL6R	EVSEL5R	EVSEL4R	EVSEL3R	EVSEL2R	EVSEL1R	EVSEL0R
NT									STCNT	STCNT	STCNT	STCNT	STCNT	STCNT	STCNT	STCNT
rw						rw			rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				CH1RSTCNT[1:0]		CH0RSTCNT[1:0]		ETI3RSTCNT[1:0]		ETI2RSTCNT[1:0]		ETI1RSTCNT[1:0]		ETI0RSTCNT[1:0]	
				rw		rw		rw		rw		rw		rw	

Bits	Fields	Descriptions
31	SWRSTCNT	The software signal reset counter function enable 0: Disable 1: Enable
30:27	Reserved	Must be kept at reset value.
26:24	CSIRSTCNT[2:0]	The compares match, synchronous reset or input capture reset counter function enable 000: Disable 001: CH1VAL compares match or input capture 010: CH0VAL compares match or input capture 010: CH1COMVAL_ADD compares match 011: CH0COMVAL_ADD compares match 101: Synchronous reset others: Reserve
23	EVSEL7RSTCNT	The EVSEL7 signal reset counter function enable Refer to EVSEL0RSTCNT description.
22	EVSEL6RSTCNT	The EVSEL6 signal reset counter function enable Refer to EVSEL0RSTCNT description.
21	EVSEL5RSTCNT	The EVSEL5 signal reset counter function enable Refer to EVSEL0RSTCNT description.
20	EVSEL4RSTCNT	The EVSEL4 signal reset counter function enable Refer to EVSEL0RSTCNT description.
19	EVSEL3RSTCNT	The EVSEL3 signal reset counter function enable Refer to EVSEL0RSTCNT description.
18	EVSEL2RSTCNT	The EVSEL2 signal reset counter function enable Refer to EVSEL0RSTCNT description.
17	EVSEL1RSTCNT	The EVSEL1 signal reset counter function enable Refer to EVSEL0RSTCNT description.
16	EVSEL0RSTCNT	The EVSEL0 signal reset counter function enable 0: Disable 1: Enable
15:12	Reserved	Must be kept at reset value.
11:10	CH1RSTCNT[1:0]	The channel 1 signal edge reset counter function enable

		Refer to CH0RSTCNT[1:0] description.
9:8	CH0RSTCNT[1:0]	The channel 0 signal edge reset counter function enable 00: Disable 01: Rising edge 10: Falling edge 11: Both rising and falling edges
7:6	ETI3RSTCNT[1:0]	The ETI3 signal edge reset counter function enable Refer to ETI0RSTCNT[1:0] description.
5:4	ETI2RSTCNT[1:0]	The ETI2 signal edge reset counter function enable Refer to ETI0RSTCNT[1:0] description.
3:2	ETI1RSTCNT[1:0]	The ETI1 signal edge reset counter function enable Refer to ETI0RSTCNT[1:0] description.
1:0	ETI0RSTCNT[1:0]	The ETI0 signal edge reset counter function enable 00: Disable 01: Rising edge 10: Falling edge 11: Both rising and falling edges

22.5.8 Channel 0 capture source selection register (GPTIMERx_CH0CSSEL)

Address offset: 0x1C

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								EVSEL7C H0IC	EVSEL6C H0IC	EVSEL5C H0IC	EVSEL4C H0IC	EVSEL3C H0IC	EVSEL2C H0IC	EVSEL1C H0IC	EVSEL0C H0IC
								rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				CH1CH0IC[1:0]		CH0CH0IC[1:0]		ETI3CH0IC[1:0]		ETI2CH0IC[1:0]		ETI1CH0IC[1:0]		ETI0CH0IC[1:0]	
				rw		rw		rw		rw		rw		rw	

Bits	Fields	Descriptions
31:24	Reserved	Must be kept at reset value.
23	EVSEL7CH0IC	The EVSEL7 signal as the channel 0 input capture source enable Refer to EVSEL0CH0IC description.
22	EVSEL6CH0IC	The EVSEL6 signal as the channel 0 input capture source enable Refer to EVSEL0CH0IC description.
21	EVSEL5CH0IC	The EVSEL5 signal as the channel 0 input capture source enable

		Refer to EVSEL0CH0IC description.
20	EVSEL4CH0IC	The EVSEL4 signal as the channel 0 input capture source enable Refer to EVSEL0CH0IC description.
19	EVSEL3CH0IC	The EVSEL3 signal as the channel 0 input capture source enable Refer to EVSEL0CH0IC description.
18	EVSEL2CH0IC	The EVSEL2 signal as the channel 0 input capture source enable Refer to EVSEL0CH0IC description.
17	EVSEL1CH0IC	The EVSEL1 signal as the channel 0 input capture source enable Refer to EVSEL0CH0IC description.
16	EVSEL0CH0IC	The EVSEL0 signal as the channel 0 input capture source enable 0: Disable 1: Enable
15:12	Reserved	Must be kept at reset value.
11:10	CH1CH0IC[1:0]	The channel 1 signal edge as the channel 0 input capture source enable Refer to CH0CH0IC[1:0] description.
9:8	CH0CH0IC[1:0]	The channel 0 signal edge as the channel 0 input capture source enable 00: Disable 01: Rising edge 10: Falling edge 11: Both rising and falling edges
7:6	ETI3CH0IC[1:0]	The ETI3 signal edge as the channel 0 input capture source enable Refer to ETI0CH0IC[1:0] description.
5:4	ETI2CH0IC[1:0]	The ETI2 signal edge as the channel 0 input capture source enable Refer to ETI0CH0IC[1:0] description.
3:2	ETI1CH0IC[1:0]	The ETI1 signal edge as the channel 0 input capture source enable Refer to ETI0CH0IC[1:0] description.
1:0	ETI0CH0IC[1:0]	The ETI0 signal edge as the channel 0 input capture source enable 00: Disable 01: Rising edge 10: Falling edge 11: Both rising and falling edges

22.5.9 Channel 1 capture source selection register (GPTIMERx_CH1CSSEL)

Address offset: 0x20

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								EVSEL7C H1IC	EVSEL6C H1IC	EVSEL5C H1IC	EVSEL4C H1IC	EVSEL3C H1IC	EVSEL2C H1IC	EVSEL1C H1IC	EVSEL0C H1IC
								rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				CH1CH1IC[1:0]		CH0CH1IC[1:0]		ETI3CH1IC[1:0]		ETI2CH1IC[1:0]		ETI1CH1IC[1:0]		ETI0CH1IC[1:0]	
				rw		rw		rw		rw		rw		rw	

Bits	Fields	Descriptions
31:24	Reserved	Must be kept at reset value.
23	EVSEL7CH1IC	The EVSEL7 signal as the channel 1 input capture source enable Refer to EVSEL0CH1IC description.
22	EVSEL6CH1IC	The EVSEL6 signal as the channel 1 input capture source enable Refer to EVSEL0CH1IC description.
21	EVSEL5CH1IC	The EVSEL5 signal as the channel 1 input capture source enable Refer to EVSEL0CH1IC description.
20	EVSEL4CH1IC	The EVSEL4 signal as the channel 1 input capture source enable Refer to EVSEL0CH1IC description.
19	EVSEL3CH1IC	The EVSEL3 signal as the channel 1 input capture source enable Refer to EVSEL0CH1IC description.
18	EVSEL2CH1IC	The EVSEL2 signal as the channel 1 input capture source enable Refer to EVSEL0CH1IC description.
17	EVSEL1CH1IC	The EVSEL1 signal as the channel 1 input capture source enable Refer to EVSEL0CH1IC description.
16	EVSEL0CH1IC	The EVSEL0 signal as the channel 1 input capture source enable 0: Disable 1: Enable
15:12	Reserved	Must be kept at reset value.
11:10	CH1CH1IC[1:0]	The channel 1 signal edge as the channel 1 input capture source enable Refer to CH0CH1IC[1:0] description.
9:8	CH0CH1IC[1:0]	The channel 0 signal edge as the channel 1 input capture source enable 00: Disable 01: Rising edge 10: Falling edge 11: Both rising and falling edges
7:6	ETI3CH1IC[1:0]	The ETI3 signal edge as the channel 1 input capture source enable Refer to ETI0CH1IC[1:0] description.

5:4	ETI2CH1IC[1:0]	The ETI2 signal edge as the channel 1 input capture source enable Refer to ETI0CH1IC[1:0] description.
3:2	ETI1CH1IC[1:0]	The ETI1 signal edge as the channel 1 input capture source enable Refer to ETI0CH1IC[1:0] description.
1:0	ETI0CH1IC[1:0]	The ETI0 signal edge as the channel 1 input capture source enable 00: Disable 01: Rising edge 10: Falling edge 11: Both rising and falling edges

22.5.10 Control register 0 (GPTIMERx_CTL0)

Address offset: 0x24

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved	CKESEL[1:0]	Reserved	CKSSEL[2:0]	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	CCSOEN
rw		rw				rw									rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	CKDIV[1:0]	ARSE	CAM	FDIR	DIR	SPM	Reserved	UPDIS	CEN	
rw						rw		rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:30	Reserved	Must be kept at reset value.
29:28	CKESEL[1:0]	Effective edge selection when ETIx(x = 0, 1, 2, 3) is selected as the clock. 00: Rising edge 01: Falling edge 10/11: Both rising and falling edges
27	Reserved	Must be kept at reset value.
26:24	CKSSEL[2:0]	Clock source selection 000: CK_GPTIMER 001: ETI0 010: ETI1 011: ETI2 100: ETI3 101-111: Reserved
23:17	Reserved	Must be kept at reset value.
16	CCSOEN	The counter cycle synchronization signal output enable 0: Disable

		1: Enable
15:10	Reserved	Must be kept at reset value.
9:8	CKDIV[1:0]	Clock division The CKDIV bits can be configured by software to specify division ratio between CK_GPTIMER (the timer clock) and DTS (the dead time and sampling clock) which is used for the dead time generator and the digital filter. 00: $f_{DTS} = f_{CK\_GPTIMER}$ 01: $f_{DTS} = f_{CK\_GPTIMER} / 2$ 10: $f_{DTS} = f_{CK\_GPTIMER} / 4$ 11: Reserved
7	ARSE	Auto-reload shadow enable 0: The shadow register for GPTIMERx_CAR register is disabled 1: The shadow register for GPTIMERx_CAR register is enabled
6	CAM	Counter align mode selection 0: No center-aligned mode (edge-aligned mode). The direction of the counter is specified by the DIR bit. 1: Center-aligned mode Note: This bit can only be configured when the CEN = 0.
5	FDIR	Force the counting direction 0: According to the DIR bit changes the counting direction after overflow or underflow in edge-aligned mode, center-aligned mode is invalid. 1: When counting is restarted, according to the DIR bit changes the counting direction immediately. Note: This bit can only be configured when the counter is stopped.
4	DIR	Direction 0: Count up 1: Count down Note: This bit is read only when the timer is configured in center-aligned mode.
3	SPM	Single pulse mode 0: Single pulse mode is disabled. Counter continues after an update event. 1: Single pulse mode is enabled. The CEN bit is cleared by hardware and the counter stops at next update event.
2	Reserved	Must be kept at reset value.
1	UPDIS	Update event disable This bit is used to enable or disable the update event generation. 0: Update event enable. The buffered registers are loaded to their preloaded values when counter overflow/underflow (edge alignment mode), counter underflow (central alignment mode), or other counter reset events occur. 1: Update event disable. The buffered registers keep their value, and if a counter

reset event occurs, the counter, predivider, and repeat counter values are reinitialized.

0	CEN	Counter enable 0: Counter disable 1: Counter enable

22.5.11 Count up event source selection register (GPTIMERx_CUPEVSSEL)

Address offset: 0x28

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				INTLCUPS[3:0]				EVSEL7C	EVSEL6C	EVSEL5C	EVSEL4C	EVSEL3C	EVSEL2C	EVSEL1C	EVSEL0C
								UPS	UPS	UPS	UPS	UPS	UPS	UPS	UPS
				rw				rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				CH1CUPS[1:0]		CH0CUPS[1:0]		ETI3CUPS[1:0]		ETI2CUPS[1:0]		ETI1CUPS[1:0]		ETI0CUPS[1:0]	
				rw		rw		rw		rw		rw		rw	

Bits	Fields	Descriptions
31:28	Reserved	Must be kept at reset value.
27:24	INTLCUPS[3:0]	The input level as the count up event source function enable 0000: Disable 0010: Channel 0 input low level 0011: Channel 0 input high level 0100: Channel 1 input low level 0101: Channel 1 input high level 1000: ETI0 input low level 1001: ETI0 input high level 1010: ETI1 input low level 1011: ETI1 input high level 1100: ETI2 input low level 1101: ETI2 input high level 1110: ETI3 input low level 1111: ETI3 input high level others: Reserved.
23	EVSEL7CUPS	The EVSEL7 signal as the count up event source function enable Refer to EVSEL0CUPS description.
22	EVSEL6CUPS	The EVSEL6 signal as the count up event source function enable Refer to EVSEL0CUPS description.

21	EVSEL5CUPS	The EVSEL5 signal as the count up event source function enable Refer to EVSEL0CUPS description.
20	EVSEL4CUPS	The EVSEL4 signal as the count up event source function enable Refer to EVSEL0CUPS description.
19	EVSEL3CUPS	The EVSEL3 signal as the count up event source function enable Refer to EVSEL0CUPS description.
18	EVSEL2CUPS	The EVSEL2 signal as the count up event source function enable Refer to EVSEL0CUPS description.
17	EVSEL1CUPS	The EVSEL1 signal as the count up event source function enable Refer to EVSEL0CUPS description.
16	EVSEL0CUPS	The EVSEL0 signal as the count up event source function enable 0: Disable 1: Enable
15:12	Reserved	Must be kept at reset value.
11:10	CH1CUPS[1:0]	The channel 1 signal edge as the count up event source function enable Refer to CH0CUPS[1:0] description.
9:8	CH0CUPS[1:0]	The channel 0 signal edge as the count up event source function enable 00: Disable 01: Rising edge 10: Falling edge 11: Both rising and falling edges
7:6	ETI3CUPS[1:0]	The ETI3 signal edge as the count up event source function enable Refer to ETI0UPS[1:0] description.
5:4	ETI2CUPS[1:0]	The ETI2 signal edge as the count up event source function enable Refer to ETI0UPS[1:0] description.
3:2	ETI1CUPS[1:0]	The ETI1 signal edge as the count up event source function enable Refer to ETI0UPS[1:0] description.
1:0	ETI0CUPS[1:0]	The ETI0 signal edge as the count up event source function enable 00: Disable 01: Rising edge 10: Falling edge 11: Both rising and falling edges

22.5.12 Count down event source selection register (GPTIMERx_CDNEVSSEL)

Address offset: 0x2C

Reset value: 0x0000 FFFF

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				INTLCDWS[3:0]				EVSEL7C	EVSEL6C	EVSEL5C	EVSEL4C	EVSEL3C	EVSEL2C	EVSEL1C	EVSEL0C
								DWS	DWS	DWS	DWS	DWS	DWS	DWS	DWS
				rw				rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				CH1CDWS[1:0]		CH0CDWS[1:0]		ETI3CDWS[1:0]		ETI2CDWS[1:0]		ETI1CDWS[1:0]		ETI0CDWS[1:0]	
				rw		rw		rw		rw		rw		rw	

Bits	Fields	Descriptions
31:28	Reserved	Must be kept at reset value.
27:24	INTLCDWS[3:0]	The input level as the count down event source function enable 0000: Disable 0010: Channel 0 input low level 0011: Channel 0 input high level 0100: Channel 1 input low level 0101: Channel 1 input high level 1000: ETI0 input low level 1001: ETI0 input high level 1010: ETI1 input low level 1011: ETI1 input high level 1100: ETI2 input low level 1101: ETI2 input high level 1110: ETI3 input low level 1111: ETI3 input high level others: Reserved.
23	EVSEL7CDWS	The EVSEL7 signal as the count down event source function enable Refer to EVSEL0CDWS description.
22	EVSEL6CDWS	The EVSEL6 signal as the count down event source function enable Refer to EVSEL0CDWS description.
21	EVSEL5CDWS	The EVSEL5 signal as the count down event source function enable Refer to EVSEL0CDWS description.
20	EVSEL4CDWS	The EVSEL4 signal as the count down event source function enable Refer to EVSEL0CDWS description.
19	EVSEL3CDWS	The EVSEL3 signal as the count down event source function enable Refer to EVSEL0CDWS description.
18	EVSEL2CDWS	The EVSEL2 signal as the count down event source function enable Refer to EVSEL0CDWS description.
17	EVSEL1CDWS	The EVSEL1 signal as the count down event source function enable

Refer to EVSELOCDWS description.

16	EVSELOCDWS	The EVSELO signal as the count down event source function enable 0: Disable 1: Enable
15:12	Reserved	Must be kept at reset value.
11:10	CH1CDWS[1:0]	The channel 1 signal edge as the count down event source function enable Refer to CH0CDWS[1:0] description.
9:8	CH0CDWS[1:0]	The channel 0 signal edge as the count down event source function enable 00: Disable 01: Rising edge 10: Falling edge 11: Both rising and falling edges
7:6	ETI3CDWS[1:0]	The ETI3 signal edge as the count down event source function enable Refer to ETI0CUPS[1:0] description.
5:4	ETI2CDWS[1:0]	The ETI2 signal edge as the count down event source function enable Refer to ETI0CUPS[1:0] description.
3:2	ETI1CDWS[1:0]	The ETI1 signal edge as the count down event source function enable Refer to ETI0CUPS[1:0] description.
1:0	ETI0CDWS[1:0]	The ETI0 signal edge as the count down event source function enable 00: Disable 01: Rising edge 10: Falling edge 11: Both rising and falling edges

22.5.13 Channel 0 control register (GPTIMERx_CH0CTL)

Address offset: 0x30

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CH0CAPFLT[3:0]				CH0CAPPSC[1:0]		Reserved			CH0INVSOUT[1:0]	CH0OUT	CH0DISO UT	CH0ENO UT	CH0ENDOUT[1:0]		
rw				rw					rw	rw	rw	rw	rw		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH0RED UTY	CH0OUT CTL	CH0OUTM[1:0]		Reserved			CH0EFF	CH0ACOMUP[1:0]	CH0ACOMDW[1:0]	CH0COMUP[1:0]		CH0COMDW[1:0]			
rw	rw	rw					rw	rw	rw	rw		rw			

Bits	Fields	Descriptions

31:28	CH0CAPFLT[3:0]	Channel 0 input capture filter control An event counter is used in the digital filter, in which a transition on the output occurs after N input events. This bit-field specifies the frequency used to sample CIO input signal and the length of the digital filter applied to CIO. 0000: Filter disabled, $f_{SAMP}=f_{DTS}$, $N=1$. 0001: $f_{SAMP}=f_{CK\_GPTIMER}$, $N=2$. 0010: $f_{SAMP}=f_{CK\_GPTIMER}$, $N=4$. 0011: $f_{SAMP}=f_{CK\_GPTIMER}$, $N=8$. 0100: $f_{SAMP}=f_{DTS}/2$, $N=6$. 0101: $f_{SAMP}=f_{DTS}/2$, $N=8$. 0110: $f_{SAMP}=f_{DTS}/4$, $N=6$. 0111: $f_{SAMP}=f_{DTS}/4$, $N=8$. 1000: $f_{SAMP}=f_{DTS}/8$, $N=6$. 1001: $f_{SAMP}=f_{DTS}/8$, $N=8$. 1010: $f_{SAMP}=f_{DTS}/16$, $N=5$. 1011: $f_{SAMP}=f_{DTS}/16$, $N=6$. 1100: $f_{SAMP}=f_{DTS}/16$, $N=8$. 1101: $f_{SAMP}=f_{DTS}/32$, $N=5$. 1110: $f_{SAMP}=f_{DTS}/32$, $N=6$. 1111: $f_{SAMP}=f_{DTS}/32$, $N=8$.
27:26	CH0CAPPSC[1:0]	Channel 0 input capture prescaler This bit-field specifies the factor of the prescaler on channel 0 input. The prescaler is reset when CH0EN bit in GPTIMERx_CHCTL register is cleared. 00: Prescaler disabled, capture is done on each channel input edge. 01: Capture is done every 2 channel input edges. 10: Capture is done every 4 channel input edges. 11: Capture is done every 8 channel input edges.
25:23	Reserved	Must be kept at reset value.
22:21	CH0INVSOUT[1:0]	Channel 0 invalid level setting when GTOC sends a stop output request 00: No effect 01: Output Hi-Z 10: Output low level 11: Output high level
20	CH0OUT	Channel 0 output level selection when counter enable or disable. 0: Output level is determined by CH0ENOUT bit or CH0DISOUT bit. 1: Output level holding. Note: If this bit is modified when the counter is stopped, and will take effect when CH0DISOUT changes or the counter state changes.
19	CH0DISOUT	Channel 0 output level selection when counter disable 0: Output low level 1: Output high level

		Note: This bit can only be configured when the counter is stopped (CEN=0), effective immediately.
18	CH0ENOUT	Channel 0 output initial level selection when counter enable. 0: Output low level 1: Output high level
17:16	CH0ENDOUT[1:0]	Channel 0 output level selection at the end of the period 00: Output level is maintained 01: Output low level 10: Output high level 11: Output level toggle Note: including overflow(count up), underflow(count down) in edge-aligned mode, underflow in center alignment mode and reset counter.
15	CH0REDUTY	Channel 0 output level selection when counter release from 0% or 100% duty cycle mode 0: The CH0ENDOUT output is based on 0% or 100% level. 1: The CH0ENDOUT output is based on 0% or 100% comparison matching level.
14	CH0OUTCTL	Channel 0 output control setting 0: According to the CH0OUTM[1:0] bits change the output mode after overflow in count up edge-aligned mode, underflow in count down edge-aligned mode and center-aligned mode. 1: According to the CH0OUTM[1:0] bits change the output mode immediately. Note: This bit can only be configured when the counter is stopped, and the bit is reset after the first cycle count.
13:12	CH0OUTM[1:0]	Channel 0 output mode 0x: The duty cycle of the channel output is comparison matching 10: The duty cycle of the channel output is 0% 11: The duty cycle of the channel output is 100%
11:9	Reserved	Must be kept at reset value.
8	CH0EFF	Channel 0 output control is effective or not when the end of the cycle and comparison match occur simultaneously. 0: Channel 0 output control is effective, comparison match has higher priority. 1: Channel 0 output control is invalid, and the output level is maintained
7:6	CH0ACOMUP[1:0]	Channel 0 additional compare match output level control when counter count up Refer to CH0COMDW[1:0] description.
5:4	CH0ACOMDW[1:0]	Channel 0 additional compare match output level control when counter count down Refer to CH0COMDW[1:0] description.
3:2	CH0COMUP[1:0]	Channel 0 compare match output level control when counter count up

Refer to CH0COMDW[1:0] description.

1:0	CH0COMDW[1:0]	Channel 0 compare match output level control when counter count down 00: No effect 01: Low level 10: High level 11: Toggle level

22.5.14 Channel 1 control register (GPTIMERx_CH1CTL)

Address offset: 0x34

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CH1CAPFLT[3:0]				CH1CAPSC[1:0]		Reserved		CH1INVSOUT[1:0]		CH1OUT	CH1DISO UT	CH1ENO UT	CH1ENDOUT[1:0]		
rw				rw				rw		rw	rw	rw	rw		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH1RED UTY	CH1OUT CTL	CH1OUTM[1:0]		Reserved		CH1EFF	CH1ACOMUP[1:0]		CH1ACOMDW[1:0]		CH1COMUP[1:0]		CH1COMDW[1:0]		
rw	rw	rw				rw	rw		rw		rw		rw		

Bits	Fields	Descriptions
31:28	CH1CAPFLT[3:0]	Channel 1 input capture filter control An event counter is used in the digital filter, in which a transition on the output occurs after N input events. This bit-field specifies the frequency used to sample CIO input signal and the length of the digital filter applied to CIO. 0000: Filter disabled, $f_{SAMP}=f_{DTS}$, $N=1$. 0001: $f_{SAMP}=f_{CK\_GPTIMER}$, $N=2$. 0010: $f_{SAMP}=f_{CK\_GPTIMER}$, $N=4$. 0011: $f_{SAMP}=f_{CK\_GPTIMER}$, $N=8$. 0100: $f_{SAMP}=f_{DTS}/2$, $N=6$. 0101: $f_{SAMP}=f_{DTS}/2$, $N=8$. 0110: $f_{SAMP}=f_{DTS}/4$, $N=6$. 0111: $f_{SAMP}=f_{DTS}/4$, $N=8$. 1000: $f_{SAMP}=f_{DTS}/8$, $N=6$. 1001: $f_{SAMP}=f_{DTS}/8$, $N=8$. 1010: $f_{SAMP}=f_{DTS}/16$, $N=5$. 1011: $f_{SAMP}=f_{DTS}/16$, $N=6$. 1100: $f_{SAMP}=f_{DTS}/16$, $N=8$. 1101: $f_{SAMP}=f_{DTS}/32$, $N=5$. 1110: $f_{SAMP}=f_{DTS}/32$, $N=6$. 1111: $f_{SAMP}=f_{DTS}/32$, $N=8$.

27:26	CH1CAPPSC[1:0]	Channel 1 input capture prescaler This bit-field specifies the factor of the prescaler on channel 1 input. The prescaler is reset when CH1EN bit in GPTIMERx_CHCTL register is cleared. 00: Prescaler disabled, capture is done on each channel input edge. 01: Capture is done every 2 channel input edges. 10: Capture is done every 4 channel input edges. 11: Capture is done every 8 channel input edges.
25:23	Reserved	Must be kept at reset value.
22:21	CH1INVSOUT[1:0]	Channel 1 invalid level setting when GTOC sends a stop output request 00: No effect 01: Output Hi-Z 10: Output low level 11: Output high level
20	CH1OUT	Channel 1 output level selection when counter enable or disable. 0: Output level is determined by CH1ENOUT bit or CH1DISOUT bit. 1: Output level holding. Note: This bit is modified when the counter is stopped, and will take effect when CH0DISOUT changes or the counter state changes.
19	CH1DISOUT	Channel 1 output level selection when counter disable 0: Output low level 1: Output high level Note: This bit can only be configured when the counter is stopped (CEN=0), effective immediately.
18	CH1ENOUT	Channel 1 output initial level selection when counter enable. 0: Output low level 1: Output high level
17:16	CH1ENDOUT[1:0]	Channel 1 output level selection at the end of the period 00: Output level is maintained 01: Output low level 10: Output high level 11: Output level toggle Note: including overflow(count up), underflow(count down) in edge-aligned mode, underflow in center alignment mode and reset counter.
15	CH1REDUTY	Channel 1 output level selection when counter release from 0% or 100% duty cycle mode 0: The CH1ENDOUT output is based on 0% or 100% level. 1: The CH0ENDOUT output is based on 0% or 100% comparison matching level.
14	CH1OUTCTL	Channel 1 output control setting 0: According to the CH1OUTM[1:0] bits change the output mode after overflow in

edge-aligned mode count up, underflow in edge-aligned mode count down and center-aligned mode.

1: According to the CH1OUTM[1:0] bits change the output mode immediately.

Note: This bit can only be configured when the counter is stopped, and the bit is reset after the first cycle count.

13:12	CH1OUTM[1:0]	Channel 1 output mode 0x: The duty cycle of the channel output is comparison matching 10: The duty cycle of the channel output is 0% 11: The duty cycle of the channel output is 100%
11:9	Reserved	Must be kept at reset value.
8	CH1EFF	Channel 1 output control is effective or not when the end of the cycle and comparison match occur simultaneously. 0: Channel1 output control is effective, comparison match has higher priority. 1: Channel 1 output control is invalid
7:6	CH1ACOMUP[1:0]	Channel 1 additional compare match output level control when counter count up Refer to CH1COMDW[1:0] description.
5:4	CH1ACOMDW[1:0]	Channel 1 additional compare match output level control when counter count down Refer to CH1COMDW[1:0] description.
3:2	CH1COMUP[1:0]	Channel 1 compare match output level control when counter count up Refer to CH1COMDW[1:0] description.
1:0	CH1COMDW[1:0]	Channel 1 compare match output level control when counter count down 00: No effect 01: Low level 10: High level 11: Toggle level

22.5.15 Channel control register (GPTIMERx_CHCTL)

Address offset: 0x38

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved											STORES EL	Reserved		DTEN	CH1MSE L
											rw			rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	CH1SEN	CH1COM ADDSEN	CH1COM SEN	Reserved		CH1SEL	CH1EN	Reserved	CH0SEN	CH0COM ADDSEN	CH0COM SEN	Reserved		CH0SEL	CH0EN

697

		output. When channel 1 is configured in input mode, setting this bit enables the capture event in channel 1. 0: Channel 1 disabled 1: Channel 1 enabled
7	Reserved	Must be kept at reset value.
6	CH0SEN	Channel 0 output shadow enable When this bit is set, the CH0ENDOUT[1:0], CH0ACOMUP[1:0], CH0ACOMDW[1:0], CH0COMUP[1:0], CH0COMDW[1:0] bits of GPTIMERx_CH0CTL register which updates at each update event will be enabled. 0: Channel 0 output shadow disabled 1: Channel 0 output shadow enabled
5	CH0COMADDSEN	Channel 0 additional compare output shadow enable When this bit is set, the shadow register of GPTIMERx_CH0COMV_ADD register which updates at each update event will be enabled. 0: Channel 0 additional compare shadow disabled 1: Channel 0 additional compare shadow enabled
4	CH0COMSEN	Channel 0 compare output shadow enable When this bit is set, the shadow register of GPTIMERx_CH0CV register, which updates at each update event will be enabled. 0: Channel 0 output compare shadow disabled 1: Channel 0 output compare shadow enabled
3:2	Reserved	Must be kept at reset value.
1	CH0SEL	Channel 0 capture/compare select 0: Channel 0 is configured as output 1: Channel 0 is configured as input
0	CH0EN	Channel 0 capture/compare enable When channel 0 is configured in output mode, setting this bit enables CH0_O signal in active state. When channel 0 is configured in input mode, setting this bit enables the capture event in channel 0. 0: Channel 0 disabled 1: Channel 0 enabled

22.5.16 DMA and interrupt enable register (GPTIMERx_DMAINTEN)

Address offset: 0x3C

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

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

Reserved						CLOSSEL[1:0]		Reserved	RCENDE N	CH1COM ADDEN	CH0COM ADDEN	OVUNDEN[1:0]		CH1DEN	CH0DEN
rw						rw		rw	rw	rw		rw	rw		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RCENDIE	CHLOUT DET	CHHOUT DET	Reserved					RSIE	Reserved	CH1COM ADDIE	CH0COM ADDIE	UFIE	OFIE	CH1IE	CH0IE
rw	rw	rw	rw					rw		rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:26	Reserved	Must be kept at reset value.
25:24	CLOSSEL[1:0]	Closing output source selection from GTOC 00: GTOC0 01: GTOC1 10: GTOC2 11: GTOC3
23	Reserved	Must be kept at reset value.
22	RCENDEN	Repetition count end DMA request enable 0: Disable 1: Enable
21	CH1COMADDEN	Channel 1 additional compare DMA request enable 0: Disable 1: Enable
20	CH0COMADDEN	Channel 0 additional compare DMA request enable 0: Disable 1: Enable
19:18	OVUNDEN[1:0]	Counter overflow / underflow DMA request enable 00: Disable 01: Overflow DMA request enable 10: Underflow DMA request enable 11: Overflow / underflow DMA request enable
17	CH1DEN	Channel 1 capture / compare DMA request enable 0: Disable 1: Enable
16	CH0DEN	Channel 0 capture / compare DMA request enable 0: Disable 1: Enable
15	RCENDIE	Repetition count end interrupt enable 0: Disable

		1: Enable
14	CHLOUTDET	Channel 0 and channel 1 simultaneously output low level output off detection enable 0: Disable 1: Enable
13	CHHOUTDET	Channel 0 and channel 1 simultaneously output high level output off detection enable 0: Disable 1: Enable
12:8	Reserved	Must be kept at reset value.
7	RSIE	Reset interrupt enable 0: Disable 1: Enable
6	Reserved	Must be kept at reset value.
5	CH1COMADDIE	Channel 1 additional compare interrupt enable 0: Disable 1: Enable
4	CH0COMADDIE	Channel 0 additional compare interrupt enable 0: Disable 1: Enable
3	UFIE	Counter underflow interrupt enable 0: Disable 1: Enable
2	OFIE	Counter overflow interrupt enable 0: Disable 1: Enable
1	CH1IE	Channel 1 capture/compare interrupt enable 0: Disable 1: Enable
0	CH0IE	Channel 0 capture/compare interrupt enable 0: Disable 1: Enable

22.5.17 Interrupt flag register (GPTIMERx_INTF)

Address offset: 0x40

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved							OSTF	Reserved							DIRF
r							r	r							r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RCENDIF	CHLOUT	CHHOUT	Reserved	ADT2CM	ADT2CM	ADT1CM	ADT1CM	RSIF	Reserved	CH1COM	CH0COM	UFIF	OFIF	CH1IF	CH0IF
rc_w0	r	r		rc_w0	rc_w0	rc_w0	rc_w0	rc_w0		rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0

Bits	Fields	Descriptions
31:25	Reserved	Must be kept at reset value.
24	OSTF	Output stop flag (Stop output source selected according to CLOSSEL[1:0] bit field) 0: No stop output request sent by the GTOC 1: The stop output request sent by the GTOC Note: After receiving the stop output request, the invalid output persists until the end of the current PWM cycle even if the stop output request bit is flushed to 0.
23:17	Reserved	Must be kept at reset value.
16	DIRF	The counter direction flag 0: The timer counts up 1: The timer counts down
15	RCENDIF	Repetition count end interrupt flag 0: No repetition count end interrupt occurred 1: Repetition count end interrupt occurred
14	CHLOUTF	Channel 0 and channel 1 simultaneously output low level flag 0: The channel 0 and channel 1 enable, and no simultaneously output low level interrupt occurred or at least one channel is disable 1: The channel 0 and channel 1 enable, and simultaneously output low level interrupt occurred
13	CHHOUTF	Channel 0 and channel 1 simultaneously output high level flag 0: The channel 0 and channel 1 enable, and no simultaneously output high level interrupt occurred or at least one channel is disable 1: The channel 0 and channel 1 enable, and simultaneously output high level interrupt occurred Note: When CHHOUTDET bit enable, the corresponding flag bit will be output to the GTOC module, and no interrupt signal is generated, which needs to be used in combination with the interrupt signal of GTOC.
12	Reserved	Must be kept at reset value.
11	ADT2CMDF	ADTCV2[15:0] compare match (counting down only) flag. 0: No ADTCV2[15:0] compare match interrupt occurred 1: ADTCV2[15:0] compare match interrupt occurred

10	ADT2CMUF	ADTCV2[15:0] compare match (counting up only) flag. 0: No ADTCV2[15:0] compare match interrupt occurred 1: ADTCV2[15:0] compare match interrupt occurred
9	ADT1CMDIF	ADTCV1[15:0] compare match (counting down only) flag. 0: No ADTCV1[15:0] compare match interrupt occurred 1: ADTCV1[15:0] compare match interrupt occurred
8	ADT1CMUF	ADTCV1[15:0] compare match (counting up only) flag. 0: No ADTCV1[15:0] compare match interrupt occurred 1: ADTCV1[15:0] compare match interrupt occurred
7	RSIF	Reset interrupt flag This bit is set by hardware to 1 and software to 0 when any counter reset event (including software/hardware counter reset event, GPTIMER synchronous reset) occurs. 0: No reset interrupt occurred 1: Reset interrupt occurred
6	Reserved	Must be kept at reset value.
5	CH1COMADDIF	Channel 1 additional compare interrupt flag. Refer to CH0COMADDIF description.
4	CH0COMADDIF	Channel 0 additional compare interrupt flag. This flag is set by hardware and cleared by software. If channel 0 is in output mode, this flag is set when a compare event occurs. 0: No channel 0 output compare interrupt occurred 1: Channel 0 output compare interrupt occurred
3	UFIF	The counter underflow interrupt flag 0: No underflow interrupt occurred 1: Underflow interrupt occurred
2	OFIF	The counter overflow interrupt flag 0: No overflow interrupt occurred 1: Overflow interrupt occurred
1	CH1IF	Channel 1 capture/compare interrupt flag Refer to CH0IF description.
0	CH0IF	Channel 0 capture/compare interrupt flag This flag is set by hardware and cleared by software. When channel 0 is in input mode, this flag is set when a capture event occurs. When channel 0 is in output mode, this flag is set when a compare event occurs. When channel 0 is in input mode, automatically clear when CH0CV is read. 0: No Channel 1 interrupt occurred 1: Channel 1 interrupt occurred

22.5.18 Update source selection register (GPTIMERx_UPSSEL)

Address offset: 0x44

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				CH1UPSEL[1:0]		CH0UPSEL[1:0]		Reserved						GLBUPSEL[1:0]	
rw				rw		rw		rw						rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CARUPSEL[1:0]		ADCCR2UPSEL[1:0]		ADCCR1UPSEL[1:0]		CH1COMV_ADDUPSEL[1:0]		CH1CVUPSEL[1:0]		CH0COMV_ADDUPSEL[1:0]		CH0CVUPSEL[1:0]			
rw		rw		rw		rw		rw		rw		rw		rw	

Bits	Fields	Descriptions
31:28	Reserved	Must be kept at reset value.
27:26	CH1UPSEL[1:0]	Channel 1 output mode update source selection. Refer to CH0UPSEL[1:0] description.
25:24	CH0UPSEL[1:0]	Channel 0 output mode update source selection. 00: Enable global update function 01: Overflow update or counter is cleared 10: Underflow update or counter is cleared 11: Overflow or underflow update or counter is cleared Note: These bits are controlled by UPDIS bit in the GPTIMERx_CTL0 register.
23:18	Reserved	Must be kept at reset value.
17:16	GLBUPSEL[1:0]	Global update mode update source selection. 00: The repeat counter is decremented during overflow / underflow, and when the repeat counter decrements to 0 produces a global update event. 01: The repeat counter is decremented during overflow, and when the repeat counter decrements to 0 produces a global update event. 10: The repeat counter is decremented during underflow, and when the repeat counter decrements to 0 produces a global update event. 11: Reserved. Note: These bits are controlled by UPDIS bit in the GPTIMERx_CTL0 register.
15:14	CARUPSEL[1:0]	The counter auto reload value update mode selection, used to set when the counter auto reload value will be automatically overloaded. 00: Enable global update function. 01: Overflow update or counter is cleared 10: Underflow update or counter is cleared 11: Overflow or underflow update or counter is cleared Note: These bits are controlled by UPDIS bit in the GPTIMERx_CTL0 register.
13:12	ADCCR2UPSEL[1:0]	The ADC triggered comparison value 2 update mode selection.

Refer to CARUPSEL[1:0] description

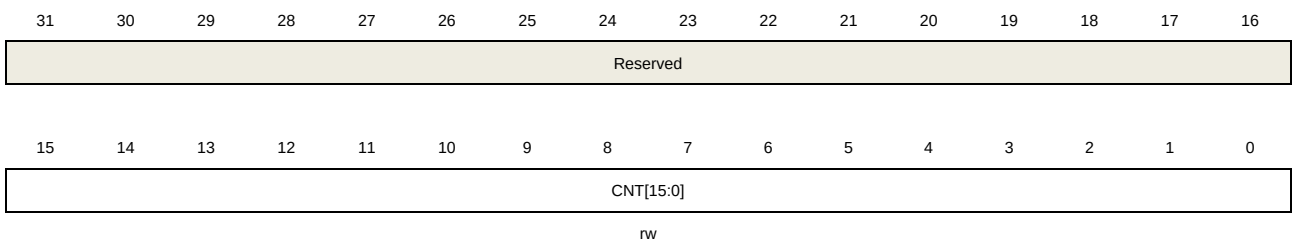
11:10	ADCCR1UPSEL[1:0]	The ADC triggered comparison value 1 update mode selection. Refer to CARUPSEL[1:0] description
9:7	CH1COMV_ADDUPSEL[1:0]	Channel 1 additional comparison value update mode selection. Refer to CH0COMV_ADDUPSEL[1:0] description
6:5	CH1CVUPSEL[1:0]	Channel 1 capture/compare value update mode selection. Refer to CARUPSEL[1:0] description
4:2	CH0COMV_ADDUPSEL[1:0]	Channel 0 additional comparison value update mode selection. 000: Enable global update function. 001: Overflow update or counter is cleared 010: Underflow update or counter is cleared 011: Overflow or underflow update or counter is cleared 100: channel 0 comparison matches Note: These bits are controlled by UPDIS bit in the GPTIMERx_CTL0 register.
1:0	CH0CVUPSEL[1:0]	Channel 0 capture/compare value update mode selection. Refer to CARUPSEL[1:0] description

22.5.19 Counter register (GPTIMERx_CNT)

Address offset: 0x48

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



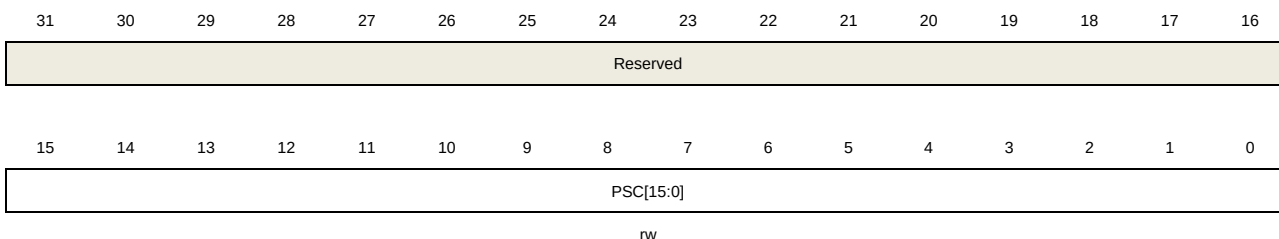
Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CNT[15:0]	This bit-field indicates the current counter value. Writing to this bit-field can change the value of the counter.

22.5.20 Prescaler register (GPTIMERx_PSC)

Address offset: 0x4C

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



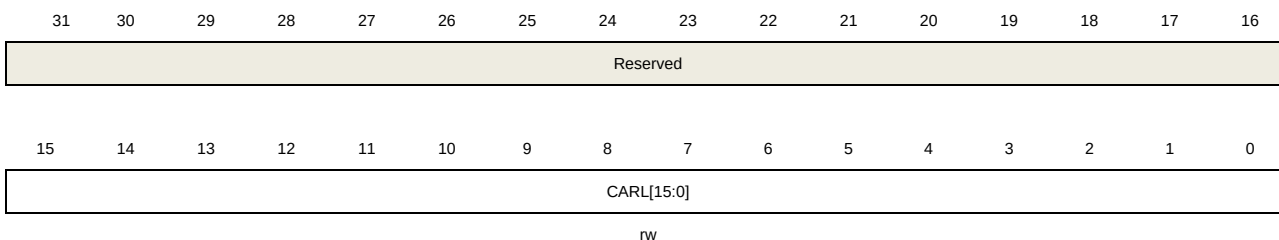
Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	PSC[15:0]	Prescaler value of the counter clock The PSC clock is divided by (PSC+1) to generate the counter clock. The value of this bit-field will be loaded to the corresponding shadow register at every reset event.

22.5.21 Counter auto reload register (GPTIMERx_CAR)

Address offset: 0x50

Reset value: 0x0000 0000

This register can be accessed by half-word(16-bit) or word(32-bit).



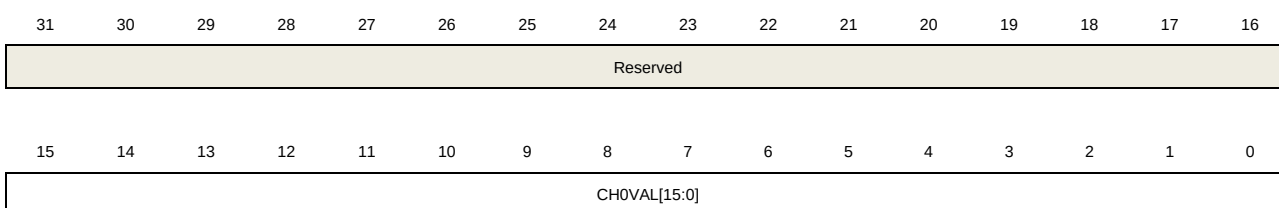
Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CARL[15:0]	Counter auto reload value This bit-field specifies the auto reload value of the counter.

22.5.22 Channel 0 capture/compare value register (GPTIMERx_CH0CV)

Address offset: 0x54

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



rw

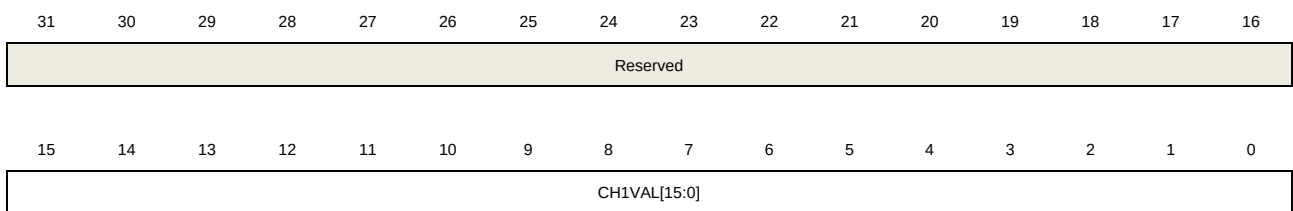
Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CH0VAL[15:0]	Capture/compare value of channel 0 When channel 0 is configured in input mode, these bit-field indicates the counter value at the last capture event. And this bit-field is read-only. When channel 0 is configured in output mode, this bit-field contains value to be compared to the counter. After enabling the corresponding shadow register, the shadow register value is updated with each update event.

22.5.23 Channel 1 capture/compare value register (GPTIMERx_CH1CV)

Address offset: 0x58

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



rw

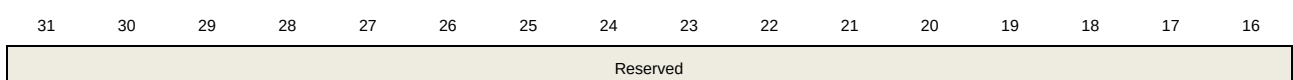
Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CH1VAL[15:0]	Capture/compare value of channel 1 When channel 1 is configured in input mode, these bit-field indicates the counter value at the last capture event. And this bit-field is read-only. When channel 1 is configured in output mode, this bit-field contains value to be compared to the counter. After enabling the corresponding shadow register, the shadow register value is updated with each update event.

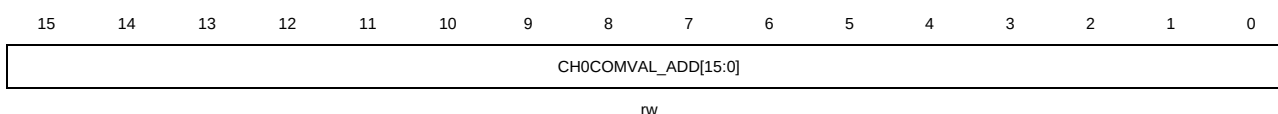
22.5.24 Channel 0 additional compare value register (GPTIMERx_CH0COMV_ADD)

Address offset: 0x5C

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).





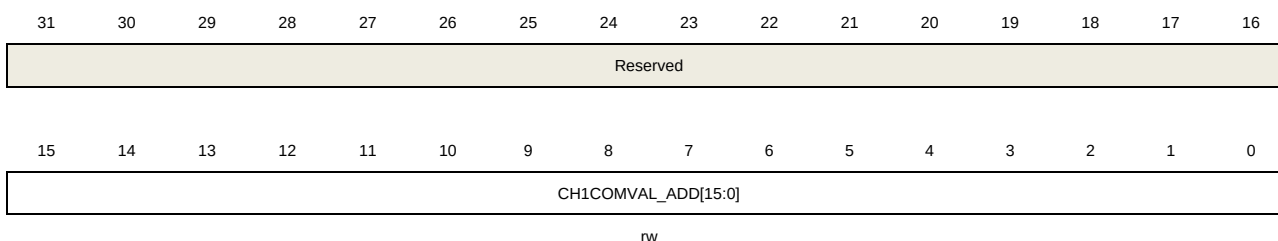
Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CH0COMVAL_ADD [15:0]	Additional compare value of channel 0 When channel 0 is configured in output mode, this bit-field contains value to be compared to the counter. After enabling the corresponding shadow register, the shadow register value is updated with each update event.

22.5.25 Channel 1 additional compare value register (GPTIMERx_CH1COMV_ADD)

Address offset: 0x60

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



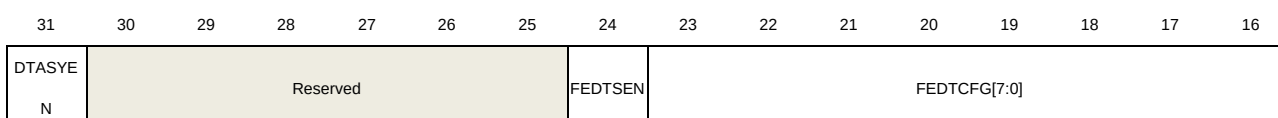
Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CH1COMVAL_ADD [15:0]	Additional compare value of channel 1 When channel 1 is configured in output mode, this bit-field contains value to be compared to the counter. After enabling the corresponding shadow register, the shadow register value is updated with each update event.

22.5.26 Dead-time control register (GPTIMERx_DTCTL)

Address offset: 0x64

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



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

Bits	Fields	Descriptions
31	DTASYEN	Dead time asymmetric mode enable 0: REDTCFG[7:0] is used for both rising and falling edge dead time 1: REDTCFG[7:0] is used for the rising edge dead time and FEDTCFG[7:0] is used for the falling edge dead time Note: This bit cannot be changed when CEN=1.
30:25	Reserved	Must be kept at reset value.
24	FEDTSEN	The falling edge dead time shadow register is enabled. When this bit is set, the value of the dead time can be changed dynamically, and the dead time takes effect when the update event arrives. 0: Disable 1: Enable
23:16	FEDTCFG[7:0]	The falling edge dead time configuration This bit-field controls the value of the dead-time, which is inserted before the output transitions. The relationship between the value of FEDTCFG and the duration of dead-time is as follow: $FEDTCFG[7:5] = 3'b0xx$: DT value = $FEDTCFG[7:0] * t_{DT}$, $t_{DT} = t_{DTS}$. $FEDTCFG[7:5] = 3'b10x$: DT value = $(64 + FEDTCFG[5:0]) * t_{DT}$, $t_{DT} = t_{DTS} * 2$. $FEDTCFG[7:5] = 3'b110$: DT value = $(32 + FEDTCFG[4:0]) * t_{DT}$, $t_{DT} = t_{DTS} * 8$. $FEDTCFG[7:5] = 3'b111$: DT value = $(32 + FEDTCFG[4:0]) * t_{DT}$, $t_{DT} = t_{DTS} * 16$.
15:9	Reserved	Must be kept at reset value.
8	REDTSEN	The rising edge dead time shadow register enable When this bit is set, the value of the dead time can be changed dynamically, and the dead time takes effect when the update event arrives. 0: Disable 1: Enable
7:0	REDTCFG[7:0]	The rising edge dead time configuration This bit-field controls the value of the dead-time, which is inserted before the output transitions. The relationship between the value of REDTCFG and the duration of dead-time is as follow: $REDTCFG[7:5] = 3'b0xx$: DT value = $REDTCFG[7:0] * t_{DT}$, $t_{DT} = t_{DTS}$. $REDTCFG[7:5] = 3'b10x$: DT value = $(64 + REDTCFG[5:0]) * t_{DT}$, $t_{DT} = t_{DTS} * 2$. $REDTCFG[7:5] = 3'b110$: DT value = $(32 + REDTCFG[4:0]) * t_{DT}$, $t_{DT} = t_{DTS} * 8$. $REDTCFG[7:5] = 3'b111$: DT value = $(32 + REDTCFG[4:0]) * t_{DT}$, $t_{DT} = t_{DTS} * 16$.

22.5.27 ADC trigger control register (GPTIMERx_ADCTL)

Address offset: 0x68

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				ADCSM4	Reserved				ADCSM4	Reserved				ADCSM3	ADCSM3
				EN					SEL					EN	SEL
				rw				rw				rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADT2DW	ADT2UP	ADT1DW	ADT1UP	Reserved			ADTCV2	ADTCV1	Reserved			ADTCV2SSEL[2:0]		Reserved	ADTCV1SSEL[2:0]
GTEN	GTEN	GTEN	GTEN				SEN	SEN				rw			rw
rw		rw		rw		rw		rw		rw		rw		rw	

Bits	Fields	Descriptions
31:29	Reserved	must be kept at reset value.
28	ADCSM4EN	ADCSM4 pin output enable 0: ADCSM4 pin output disabled 1: ADCSM4 pin output enabled
27:25	Reserved	must be kept at reset value.
24	ADCSM4SEL	Select the ADC trigger source connected to ADCSM4 pin 0: Source of monitor is GPTIMERx_TRGA 1: Source of monitor is GPTIMERx_TRGB
23:21	Reserved	must be kept at reset value.
20	ADCSM3EN	ADCSM3 pin output enable 0: ADCSM3 pin output disabled 1: ADCSM3 pin output enabled
19:17	Reserved	must be kept at reset value.
16	ADCSM3SEL	Select the ADC trigger source connected to ADCSM3 pin 0: Source of monitor is GPTIMERx_TRGA 1: Source of monitor is GPTIMERx_TRGB
15	ADT2DWGTEN	The ADTCV2[15:0] comparison match (count down only) generate trigger enable 0: Disable 1: Enable
14	ADT2UPGTEN	The ADTCV2[15:0] comparison match (count up only) generate trigger enable 0: Disable 1: Enable
13	ADT1DWGTEN	The ADTCV1[15:0] comparison match (count down only) generate trigger enable 0: Disable

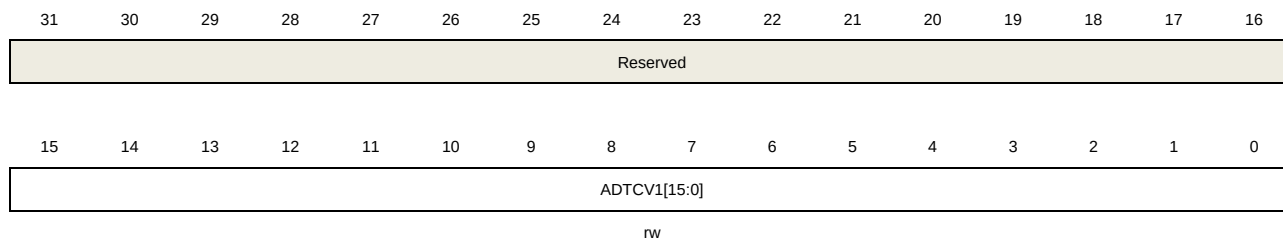
		1: Enable
12	ADT1UPGTEN	The ADTCV1[15:0] comparison match (count up only) generate trigger enable 0: Disable
		1: Enable
11:10	Reserved	must be kept at reset value.
9	ADTCV2SEN	The ADTCV2[15:0] shadow register enable 0: Disable 0: Enable
8	ADTCV1SEN	The ADTCV1[15:0] shadow register enable 0: Disable 0: Enable
7	Reserved	must be kept at reset value.
6:4	ADTCV2SSEL[2:0]	The ADC conversion request skipping source selection and the trigger signal generated by ADTCV2[15:0] Refer to ADTCV1SSEL[2:0] description.
3	Reserved	must be kept at reset value.
2:0	ADTCV1SSEL[2:0]	The ADC conversion request skipping source selection and the trigger signal generated by ADTCV1[15:0] 000: Do not compare match skipping 001: Skipping the ADC conversion request when the ADC skipping counter 1 value is not 4'b0000 (output the ADC conversion request in the ADCC1VAL[3:0] = 4'b0000 period) 010: Skipping the ADC conversion request when the ADC skipping counter 2 value is not 4'b0000 (output the ADC conversion request in the ADCC2VAL[3:0] = 4'b0000 period) 011: Skipping the ADC conversion request when the ADC skipping counter 1/2 value is not 4'b0000 (output the ADC conversion request in the ADCC1VAL[3:0] = 4'b0000 and ADCC2VAL[3:0] = 4'b0000 period) 100: Reserved 101: Skipping the ADC conversion request when the ADC skipping counter 1 value is not skipping value (output the ADC conversion request in the ADCC1VAL[3:0] = ADCC1SVAL[3:0] period) 110: Skipping the ADC conversion request when the ADC skipping counter 2 value is not skipping value (output the ADC conversion request in the ADCC2VAL[3:0] = ADCC2SVAL[3:0] period) 111: Skipping the ADC conversion request when the ADC skipping counter 1/2 value is not skipping value (output the ADC conversion request in the ADCC1VAL[3:0] = ADCC1SVAL[3:0] and ADCC2VAL[3:0] = ADCC2SVAL[3:0] period)

22.5.28 ADC trigger compare value register 1 (GPTIMERx_ADCCR1)

Address offset: 0x6C

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



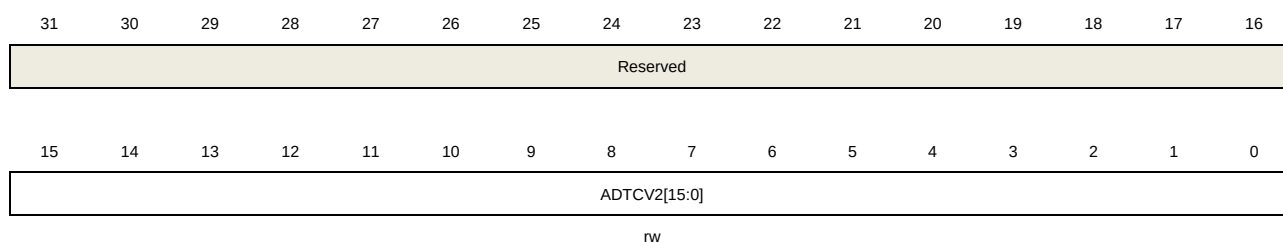
Bits	Fields	Descriptions
31:16	Reserved	must be kept at reset value
15:0	ADTCV1[15:0]	The ADC trigger compare value 1 These bits contain the value of the comparison with the counter. After the corresponding shadow register is enabled, the register value is updated with the configuration of the shadow register.

22.5.29 ADC trigger compare value register 2 (GPTIMERx_ADCCR2)

Address offset: 0x70

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



Bits	Fields	Descriptions
31:16	Reserved	must be kept at reset value
15:0	ADTCV2[15:0]	The ADC trigger compare value 2 These bits contain the value of the comparison with the counter. After the corresponding shadow register is enabled, the register value is updated with the configuration of the shadow register.

22.5.30 ADC trigger skipping register (GPTIMERx_ADCTRGs)

Address offset: 0x74

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADCC2VAL[3:0]				ADCC2IVAL[3:0]				ADCC2SVAL[3:0]				Reserved		ADCC2SKSEL[1:0]	
rw				rw				rw						rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADCC1VAL[3:0]				ADCC1IVAL[3:0]				ADCC1SVAL[3:0]				Reserved		ADCC1SKSEL[1:0]	
rw				rw				rw						rw	

Bits	Fields	Descriptions
31:28	ADCC2VAL[3:0]	This bit-field indicates the current ADC skipping counter 2 value. Range: 0- ADCC2SVAL[3:0] Note: This bit field is reset when writing ADCC2SKSEL[1:0] = 2'b00 or ADCC2SVAL[3:0] = 4'b0000. The first count clock after setting nonzero in ADCC2SVAL[3:0] bits, this counter+1.
27:24	ADCC2IVAL[3:0]	The ADC skipping counter 2 Initial value Range: 0-15 Note: This bit field is writable only if ADCC2SKSEL[1:0] != 2'b00, and this bit field will not be reset.
23:20	ADCC2SVAL[3:0]	The ADC skipping counter 2 skipping value Range: 0-15
19:18	Reserved	must be kept at reset value.
17:16	ADCC2SKSEL[1:0]	The ADC skipping counter 2 skipping source selection 00: No counting, no skipping 01: The ADTCV1[15:0] comparison matching 10: The ADTCV2[15:0] comparison matching 11: The ADTCV1[15:0] or ADTCV2[15:0] comparison matching
15:12	ADCC1VAL[3:0]	This bit-field indicates the current ADC skipping counter 1 value. Range: 0- ADCC1SVAL[3:0] Note: This bit field is reset when writing ADCC1SKSEL[1:0] = 2'b00 or ADCC1SVAL[3:0] = 4'b0000. The first count clock after setting nonzero in ADCC2SVAL[2:0] bits, this counter+1.
11:8	ADCC1IVAL[3:0]	The ADC skipping counter 1 Initial value Range: 0-15 Note: This bit field is writable only if ADCC1SKSEL[1:0] != 2'b00, and this bit field will not be reset.
7:4	ADCC1SVAL[3:0]	The ADC skipping counter 1 skipping value Range: 0-15

3:2	Reserved	must be kept at reset value.
1:0	ADCC1SKSEL[1:0]	The ADC skipping counter 1 skipping source selection 00: No counting, no skipping 01: The ADTCV1[15:0] comparison matching 10: The ADTCV2[15:0] comparison matching 11: The ADTCV1[15:0] or ADTCV2[15:0] comparison matching

22.5.31 Additional interrupt skipping control register 0 (GPTIMERx_ADDINTSCTL0)

Address offset: 0x78

Reset value: 0x0FF0 00FF

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADDC2VAL[3:0]				ADDC2IVAL[3:0]				ADDC2SVAL[3:0]				Reserved		ADDC2SKSEL[1:0]	
rw				rw				rw						rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDC1VAL[3:0]				ADDC1IVAL[3:0]				ADDC1SVAL[3:0]				Reserved		ADDC1SKSEL[1:0]	
rw				rw				rw						rw	

Bits	Fields	Descriptions
31:28	ADDC2VAL[3:0]	This bit-field indicates the current additional interrupt skipping counter 2 value. Range: 0- ADDC2SVAL[3:0] Note: This bit field is reset when writing ADDC2SKSEL[1:0] = 2'b00 or ADDC2SVAL[3:0] = 4'b0000. The first count clock after setting nonzero in ADDC2SVAL[3:0] bits, this counter+1.
27:24	ADDC2IVAL[3:0]	The additional interrupt skipping counter 2 Initial value Range: 0-15 Note: This bit field is writable only if ADDC2SKSEL[1:0] is not 0, and this bit field will not be reset.
23:20	ADDC2SVAL[3:0]	The additional interrupt skipping counter 2 skipping value Range: 0-15
19:18	Reserved	must be kept at reset value.
17:16	ADDC2SKSEL[1:0]	The additional interrupt skipping counter 2 skipping source selection 00: No counting, no skipping 01: Overflow (count up), underflow (count down) in edge-aligned mode, overflow in center alignment mode. 10: Overflow (count up), underflow (count down) in edge-aligned mode, underflow in center alignment mode.

		11: Overflow (count up), underflow (count down) in edge-aligned mode, overflow / underflow in center alignment mode.
15:12	ADDC1VAL[3:0]	This bit-field indicates the current additional interrupt skipping counter 1 value. Range: 0- ADDC1SVAL[3:0] Note: This bit field is reset when writing ADDC1SKSEL[1:0] = 2'b00 or ADDC1SVAL[3:0] = 4'b0000. The first count clock after setting nonzero in ADDC1SVAL[3:0] bits, this counter+1.
11:8	ADDC1IVAL[3:0]	The additional interrupt skipping counter 1 Initial value Range: 0-15 Note: This bit field is writable only if ADDC1SKSEL[1:0] is not 0, and this bit field will not be reset.
7:4	ADDC1SVAL[3:0]	The additional interrupt skipping counter 1 skipping value Range: 0-15
3:2	Reserved	must be kept at reset value.
1:0	ADDC1SKSEL[1:0]	The additional interrupt skipping counter 1 skipping source selection 00: No counting, no skipping 01: Overflow (count up), underflow (count down) in edge-aligned mode, overflow in center alignment mode. 10: Overflow (count up), underflow (count down) in edge-aligned mode, underflow in center alignment mode. 11: Overflow (count up), underflow (count down) in edge-aligned mode, overflow / underflow in center alignment mode.

22.5.32 Additional interrupt skipping control register 1 (GPTIMERx_ADDINTSCTL1)

Address offset: 0x7C

Reset value: 0x0000 0001

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved	ADT2AISSEL[2:0]		Reserved	ADT1AISSEL[2:0]		Reserved	OFAISSEL[2:0]		Reserved	UFAISSEL[2:0]					
rw		rw		rw		rw		rw		rw					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	CH1ADDAISSEL[2:0]		Reserved	CH0ADDAISSEL[2:0]		Reserved	CH1AISSEL[2:0]		Reserved	CH0AISSEL[2:0]					
rw		rw		rw		rw		rw		rw					

Bits	Fields	Descriptions
31	Reserved	must be kept at reset value.
30:28	ADT2AISSEL[2:0]	The trigger signal generated by ADTCV2[15:0] additional interrupt skipping function

		selection. Refer to CH0AISSEL[2:0] description.
27	Reserved	must be kept at reset value.
26:24	ADT1AISSEL[2:0]	The trigger signal generated by ADTCV1[15:0] additional interrupt skipping function selection. Refer to CH0AISSEL[2:0] description.
23	Reserved	must be kept at reset value.
22:20	OFAISSEL[2:0]	Overflow additional interrupt skipping function selection. Refer to CH0AISSEL[2:0] description.
19	Reserved	must be kept at reset value.
18:16	UFAISSEL[2:0]	Underflow additional interrupt skipping function selection. Refer to CH0AISSEL[2:0] description.
15	Reserved	must be kept at reset value.
14:12	CH1ADDAISSEL[2:0]	The channel 1 additional value compare match additional interrupt skipping function selection. Refer to CH0AISSEL[2:0] description.
11	Reserved	must be kept at reset value.
10:8	CH0ADDAISSEL[2:0]	The channel 0 additional value compare match additional interrupt skipping function selection. Refer to CH0AISSEL[2:0] description.
7	Reserved	must be kept at reset value.
6:4	CH1AISSEL[2:0]	The channel 1 input capture/compare match additional interrupt skipping function selection. Refer to CH0AISSEL[2:0] description.
3	Reserved	must be kept at reset value.
2:0	CH0AISSEL[2:0]	The channel 0 input capture/compare match additional interrupt skipping function selection. 000: Do not skipping 001: Skipping the additional interrupt request when the additional interrupt skipping counter 1 value is not 4'b0000 (output the additional interrupt request in the ADDC1VAL[3:0] = 4'b0000 period) 010: Skipping the additional interrupt request when the additional interrupt skipping counter 2 value is not 4'b0000 (output the additional interrupt request in the ADDC2VAL[3:0] = 4'b0000 period) 011: Skipping the additional interrupt request when the additional interrupt skipping counter 1/2 value is not 4'b0000 (output the additional interrupt request in the

ADDC1VAL[3:0] = 4'b0000 and ADDC2VAL[3:0] = 4'b0000 period)

100: Reserved

101: Skipping the additional interrupt request when the additional interrupt skipping counter 1 value is not skipping value (output the additional interrupt conversion request in the ADDC1VAL[3:0] = ADDC1SVAL[3:0] period)

110: Skipping the additional interrupt request when the additional interrupt skipping counter 2 value is not skipping value (output the additional interrupt conversion request in the ADDC2VAL[3:0] = ADDC2SVAL[3:0] period)

111: Skipping the additional interrupt request when the additional interrupt skipping counter 1/2 value is not skipping value (output the additional interrupt conversion request in the ADDC1VAL[3:0] = ADDC1SVAL[3:0] and ADDC2VAL[3:0] = ADDC2SVAL[3:0] period)

22.5.33 Interrupt and ADC trigger signal skipping register (GPTIMERx_IADCTSS)

Address offset: 0x80

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved													OUFINTSC[2:0]		
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		ADT2INT	ADT1INT	Reserved	OUFINTSVAL[2:0]			Reserved		OUFINTSSEL[1:0]		CH1AINT	CH0AINT	CH1INTS	CH0INTS
		SL	SL									SL	SL	L	L
rw		rw		rw			rw		rw	rw	rw	rw			

Bits	Fields	Descriptions
31:19	Reserved	must be kept at reset value.
18:16	OUFINTSC[2:0]	Overflow / underflow interrupt skipping counter Range: 0-7 Note: When the counter stops counting, this bit-field will be reset, and the two additional interrupt skipping counter value remain current.
15:14	Reserved	must be kept at reset value.
13	ADT2INTSL	The trigger signal generated by ADTCV2[15:0] interrupt skipping link Refer to CH0INTSL description.
12	ADT1INTSL	The trigger signal generated by ADTCV1[15:0] interrupt skipping link Refer to CH0INTSL description.
11	Reserved	must be kept at reset value.
10:8	OUFINTSVAL[2:0]	The number of overflow / underflow interrupt skipping value

Range: 0-7

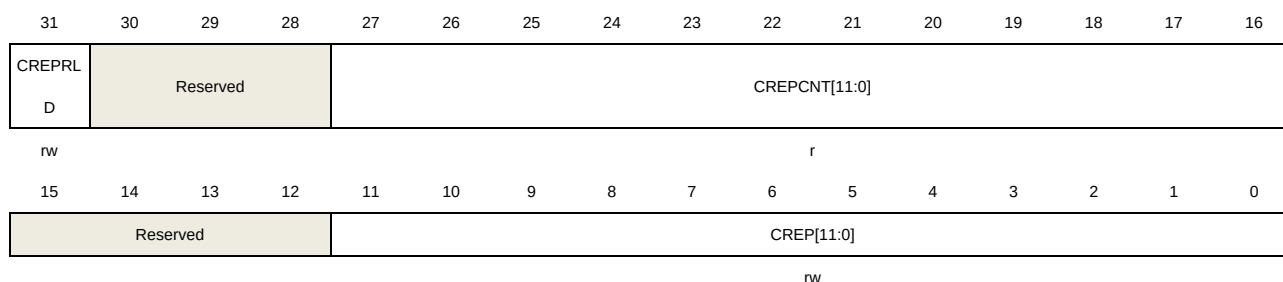
7:6	Reserved	must be kept at reset value.
5:4	OUFINTSSEL[1:0]	Overflow / underflow interrupt skip selection 00: No counting, no skipping 01: Overflow (count up), underflow (count down) in edge-aligned mode, overflow in center alignment mode. 10: Overflow (count up), underflow (count down) in edge-aligned mode, underflow in center alignment mode. 11: Overflow (count up), underflow (count down) in edge-aligned mode, overflow / underflow in center alignment mode.
3	CH1AINTSL	The channel 1 additional value compare match interrupt skipping link Refer to CH0INTSL description.
2	CH0AINTSL	The channel 0 additional value compare match interrupt skipping link Refer to CH0INTSL description.
1	CH1INTSL	The channel 1 input capture/compare match interrupt skipping link Refer to CH0INTSL description.
0	CH0INTSL	The channel 0 input capture/compare match interrupt skipping link 0: No linking 1: linking with overflow / underflow interrupts skip

22.5.34 Counter repetition register (GPTIMERx_CREP)

Address offset: 0x84

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



Bits	Fields	Descriptions
31	CREPRLD	The repeat counter is loaded immediately This bit is set to 1 by the software, when the software is set, CREPCNT[11:0] and CREP[11:0] are loaded with new values, after loading is completed, the 0 is cleared by the hardware.
30:28	Reserved	must be kept at reset value.

27:16	CREPCNT[11:0]	Repeat counter current value The value of the repeat counter, reading this value allows user to check how much the repeat count has been reduced to.
15:12	Reserved	must be kept at reset value.
11:0	CREP[11:0]	Counter repetition value This bit-field specifies the update event generation rate. Each time the repetition counter counts down to zero, an update event will be generated. The update rate of the shadow registers is also affected by this bit-field when these shadow registers are enabled.

22.5.35 Synchronous reset control register (GPTIMERx_SYN_RSTCTL)

Address offset: 0x88

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved										SYNCSS EL	Reserved				CHSRET EN
										rw					rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								UNDRET EN	OVRETE N	Reserved		CH1ARE TEN	CH0ARE TEN	CH1RET EN	CH0RET EN
								rw	rw			rw	rw	rw	rw

Bits	Fields	Descriptions
31:22	Reserved	must be kept at reset value.
21	SYNCSSEL	GPTIMER synchronization control set selection When the same set is selected, the two GPTIMER control synchronously 00: SET0 01: SET1
20:17	Reserved	must be kept at reset value.
16	CHSRETEEN	Channel input as the other GPTIMER synchronization reset source enable 0: It is forbidden to use the selected channel in the GPTIMERx_RSSEL register as a synchronous zeroing source for other GPTIMER 1: Use the channel selected in the GPTIMERx_RSSEL register as a synchronous reset source for other GPTIMER
15:8	Reserved	must be kept at reset value.
7	UNDRETEEN	Underflow as the other GPTIMER synchronization reset source enable 0: Disable

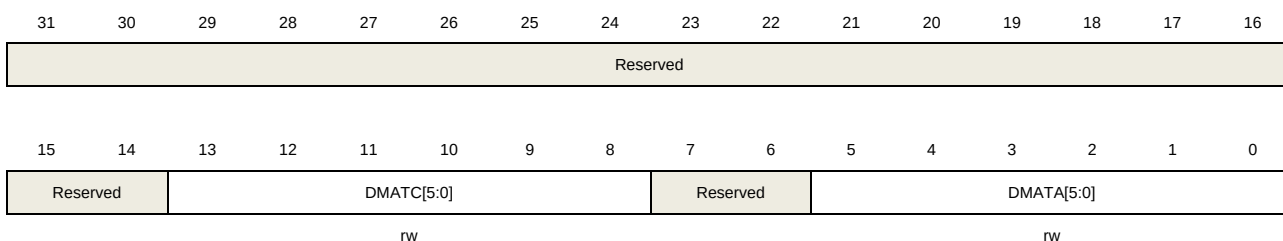
		1: Enable
6	OVRETEN	Overflow as the other GPTIMER synchronization reset source enable 0: Disable 1: Enable
5:4	Reserved	must be kept at reset value.
3	CH1ARETEN	Channel 1 additional value compare match as the other GPTIMER synchronization reset source enable 0: Disable 1: Enable
2	CH0ARETEN	Channel 0 additional value compare match as the other GPTIMER synchronization reset source enable 0: Disable 1: Enable
1	CH1RETEN	Channel 1 compare match / input capture as the other GPTIMER synchronization reset source enable 0: Disable 1: Enable
0	CH0RETEN	Channel 0 compare match / input capture as the other GPTIMER synchronization reset source enable 0: Disable 1: Enable

22.5.36 DMA configuration register (GPTIMERx_DMACFG)

Address offset: 0xE0

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



Bits	Fields	Descriptions
31:14	Reserved	Must be kept at reset value.
13:8	DMATC[5:0]	DMA transfer count This field defines the times of accessing(R/W) the GPTIMERx_DMATB register by DMA.

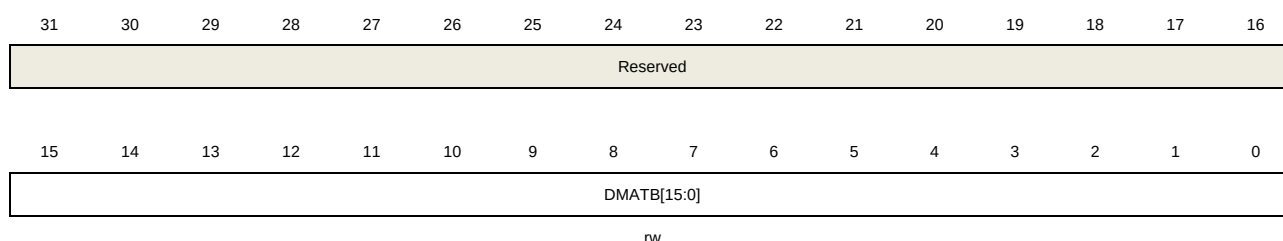
		6'b000000: transfer 1 time
		6'b000001: transfer 2 times
		...
		6'b100011: transfer 35 times
7:6	Reserved	Must be kept at reset value.
5:0	DMATA[5:0]	DMA transfer access start address This field defines the start address of accessing the GPTIMERx_DMATB register by DMA. When the first access to the GPTIMERx_DMATB register is done, this bit-field specifies the address just accessed. And then the address of the second access to the GPTIMERx_DMATB register will be (start address + 0x4). 6'b000000: GPTIMERx_WP 6'b000001: GPTIMERx_SWEN ... 6'b100011: GPTIMERx_SYN_RSTCTL In a word: start address = GPTIMERx_WP + DMATA*4

22.5.37 DMA transfer buffer register (GPTIMERx_DMATB)

Address offset: 0xE4

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	DMATB[15:0]	DMA transfer buffer When a read or write operation is assigned to this register, the register located at the address ranges from (start address) to (start address + transfer count * 4) will be accessed. The transfer count is calculated by hardware, and ranges from 0 to DMATC.

22.5.38 Configuration register (GPTIMERx_CFG)

Address offset: 0xFC

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

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

rw

Bits	Fields	Descriptions
31:2	Reserved	Must be kept at reset value.
1	CHVSEL	Write CHxVAL register selection This bit-field is set and reset by software. 1: If the value to be written to the CHxVAL register is the same as the value of CHxVAL register, the write access is ignored. 0: No effect.
0	Reserved	Must be kept at reset value.

23 Compare timer W(CPTIMERW)

23.1 Overview

The compare timer W module contains a 32-bit unsigned counter which can be configured to 16-bit. The compare timer W has two input capture channels and four compare match channels with 2 channels pulse output supported. All the channels and counter update event can be configured to generate interrupt or EVIC event at set period.

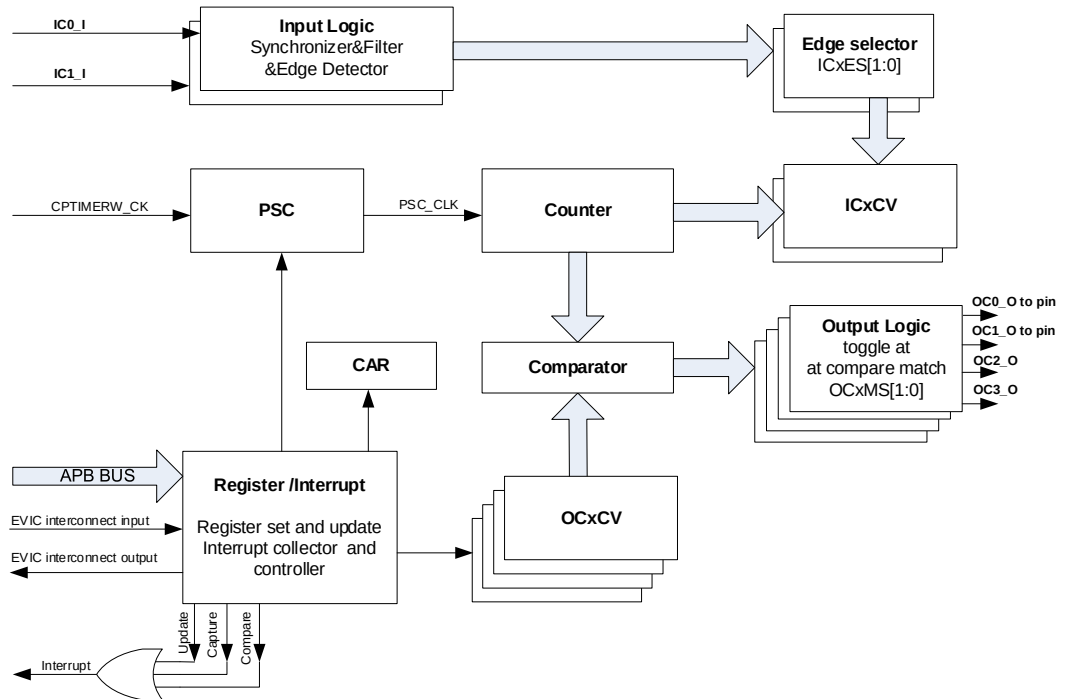
23.2 Characteristics

- Counter width: 32-bit or 16-bit configurable.
- Clock source: internal clock only.
- Counter mode: count up only.
- Auto reload function.
- Programmable prescaler: 16-bits. Factor can be changed on the go.
- Counter clear function: counter clear at input capture.
- Input capture: 2x input channels with edge programmable
- Compare match: 4x compare channels with 2 channels pulse output supported.
- EVIC interconnect input: counter enable, counter restart and event count.
- EVIC interconnect output: compare match event regardless of interrupt enable bit.
- Interrupt generated at input capture, compare match and update event.

23.3 Function overview

[Figure 23-1. Compare timer W block diagram](#) provides details on the internal configuration of the compare timer W.

Figure 23-1. Compare timer W block diagram



23.3.1 Clock source

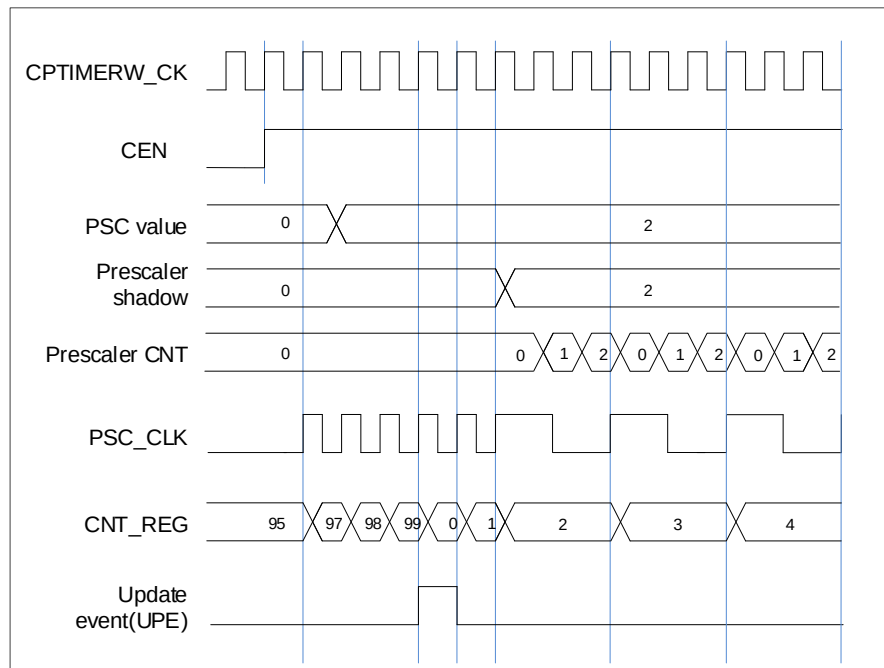
The CPTIMERW can only be clocked by the internal timer clock, which is from the source named `CK_CPTIMERW` in RCU.

The CPTIMERW_CK driven counter's prescaler to count, is equal to `CK_CPTIMERW` used to drive the counter prescaler. When the `CEN` is set, the `CK_CPTIMERW` will be divided by `PSC` value to generate `PSC_CLK`.

23.3.2 Clock prescaler

The prescaler can divide the timer clock (`CPTIMERW_CK`) to a counter clock (`PSC_CLK`) by any factor ranging from 1 to 65536. It is controlled by prescaler register (`CPTIMERW_PSC`) which can be changed ongoing, but it will be loaded to the corresponding shadow register at next update event.

Figure 23-2. Timing chart of PSC value change from 0 to 2, CAR = 99

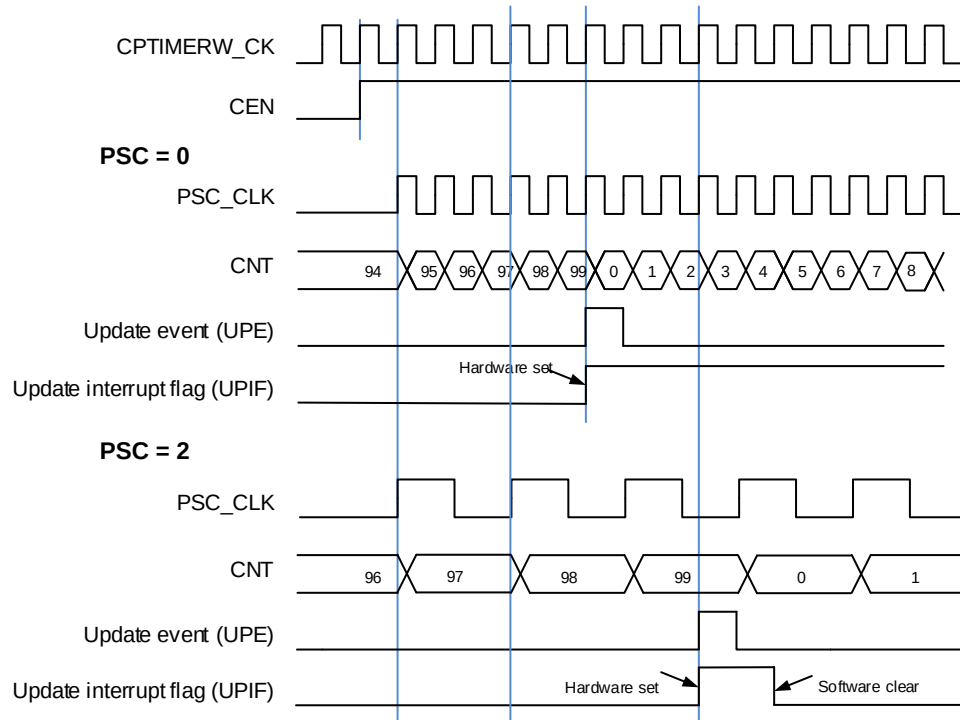


23.3.3 Counter up counting

In this mode, the counter counts up continuously from 0 to the counter-reload value, which is defined in the CPTIMERW_CAR register, in a count-up direction. Once the counter reaches the counter reload value, the counter will restart counting up from 0 again. The update event is generated at each counter overflow.

The following figures show some examples of the counter behavior for different clock prescaler factor when the clearing value is 0x99.

Figure 23-3. Timing chart of up counting mode, PSC=0/2



23.3.4 Counter width configuration

The counter width of CPTIMERW can be configured to 32-bit or 16-bit upon the setting of CNTWSEL bit in CPTIMERW_CTL0 register. When the CNTWSEL is set to 1, the counter width is 16-bit and the MSB 16 bits of CPTIMERW_CAR register should be set to 0000h while the counter overflow value is 0000_FFFFh. The default counter width is 32-bit after reset.

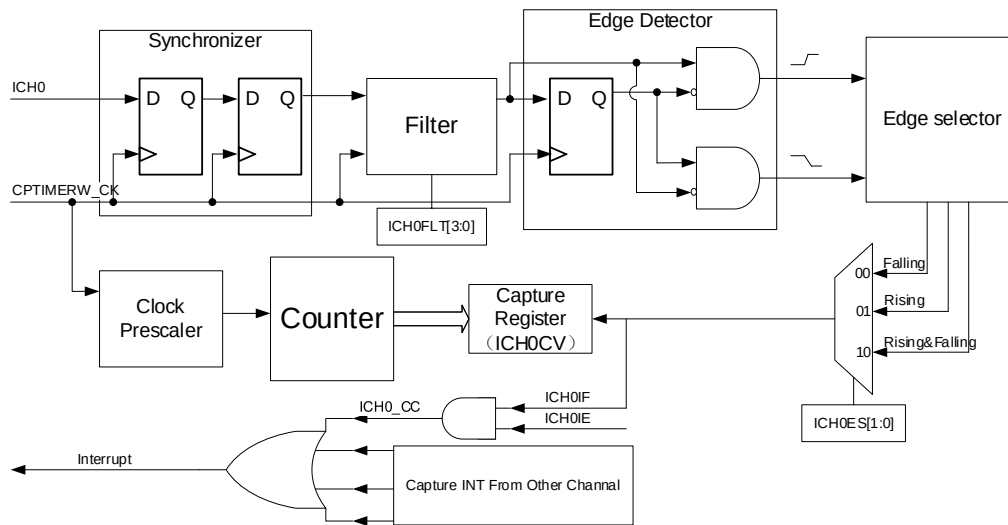
23.3.5 Input capture and output compare

The CPTIMERW has two input capture channels and four output compare channels. Each channel is controlled by correspond registers independently. Each input capture channel includes one input pin and only two of four channels (channel 0/1) can output pulses on pin.

■ Channel input capture function

Channel input capture function allows the channel to perform measurements such as pulse timing, frequency, period, duty cycle and so on. The input stage consists of a digital filter, a channel edge detection and a channel prescaler. When a selected edge occurs on the channel input, the current value of the counter is captured into the CPTIMERW_ICHxCV register, at the same time the ICHxIF bit is set and the channel interrupt is generated if enabled by ICHxIE = 1.

Figure 23-4. Channel input capture principle



One of channels' input signals (ICHx) can be chosen from the CPTIMERW_ICHx signal. First, the channel input signal (ICHx) is synchronized to CK_CPTIMERW domain, and then sampled by a digital filter to generate a filtered input signal. Then through the edge detector, the rising and falling edge are detected. You can select one of them or both by ICHxES[1:0] bits. The prescaler make several the input event generates one effective capture event. On the capture event, ICHxCV will store the value of counter.

So the process can be divided to several steps as below:

Step1: Filter configuration. (ICHxFLT in CPTIMERW_CTL1)

Based on the input signal and requested signal quality, configure compatible ICHxFLT.

Step2: Edge selection. (ICHxES in CPTIMERW_CTL1)

Rising, falling or both edge, choose one by ICHxES.

Step3: Interrupt enable. (ICHxIE in CPTIMERW_INTEN)

Enable or disable the related interrupt enable according to actual application demand.

Step4: Capture enables. (ICHxEN in CPTIMERW_CTL1)

Result: when selected input signal edge is detected, current counter's value will be loaded into CPTIMERW_ICHxCV. And ICHxIF is asserted. The interrupt request will be asserted based on the configuration of ICHxIE in CPTIMERW_CTL0.

■ Channel output compare function

In channel output compare function, the CPTIMERW can generate timed pulses with programmable position, polarity, duration and frequency. When the counter matches the value in the OCHxCV register of an output compare channel, the channel x(x=0,1) output can be toggled or unchanged based on OCHxMS. When the counter reaches the value in the

OCHxCV register, the OCHxIF bit is set and the channel x(x=0...3) interrupt is generated if OCHxIE = 1.

So the process can be divided to several steps as below:

Step1: Clock Configuration. Such as clock source, clock prescaler and so on.

Step2: Compare mode configuration.

* Set the output mode by OCHxMS[1:0] bits in CPTIMERW_CTL1.

* Enable the output by OCHxEN in CPTIMERW_CTL1.

Step3: Interrupt request enables configuration by OCHxIE in CPTIMERW_INTEN.

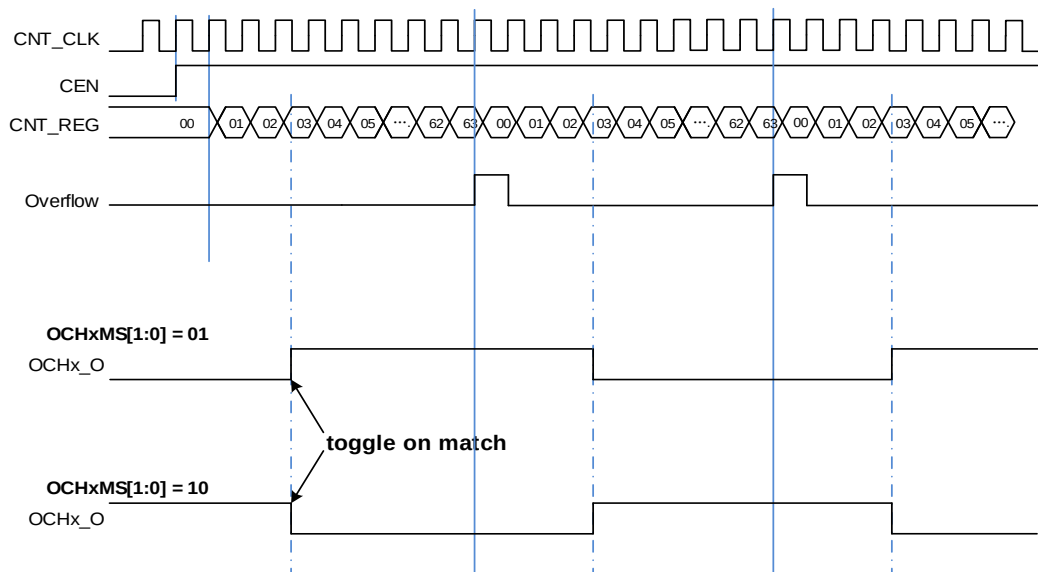
Step4: Compare output timing configuration by CPTIMERW_CAR and CPTIMERW_OCHxCV

About the OCHxCV; you can change it on the go to meet the waveform you expected.

Step5: Start the counter by CEN.

The timechart below show the two compare modes toggle on compare match with initial low or high level. CAR = 0x63, OCHxCV = 0x3.

Figure 23-5. Output-compare under two modes



Note: Only output compare channel 0/1 can output pulse on external pin. The output compare channel 0/1/2/3 can be used to generate compare match interrupt or event.

23.3.6 EVIC interconnect

The CPTIMERW supports event signal output to or input from other modules through EVIC interconnect.

EVIC interconnect output

The EVIC interconnect output function is for output the interrupt request of CPTIMERW as event signal for other modules.

(1) Compare match event

The compare match event signal of CPTIMERW output compare channel 0/1/2/3 can be output to other modules through EVIC regardless of the setting of interrupt enable bit. Please refer to EVIC chapter for more details.

EVIC interconnect input

The EVIC interconnect input function is for CPTIMERW receiving the event signal from other modules to trigger different operations. The operation and event source can be one of the followings upon the setting in EVIC_SMCFG1 and EVIC_CPTIMERW register of EVIC.

(1) Count enable

When the CPTIMERW count enable mode is selected by the EVIC_SMCFG1 register of EVIC and the event source specified by register EVIC_CPTIMERW occurs, the CPTIMERW will be enable to count and the CEN bit in CPTIMERW_CTL0 is set to 1.

(2) Count restart

When the CPTIMERW count restart mode is selected by the EVIC_SMCFG1 register of EVIC and the event specified by EVIC_CPTIMERW register occurs, the current counter value CPTIMERW_CNT will bereset to 0. If the counter of channel is enabled, the conuter will start count from 0 again.

(3) Event count

When the CPTIMERW event count mode is selected by the EVIC_SMCFG1 register of EVIC and the counter is enabled, the CPTIMERW will count up at each event specified by EVIC_CPTIMERW register regardless of the count clock source.

23.4 Register definition

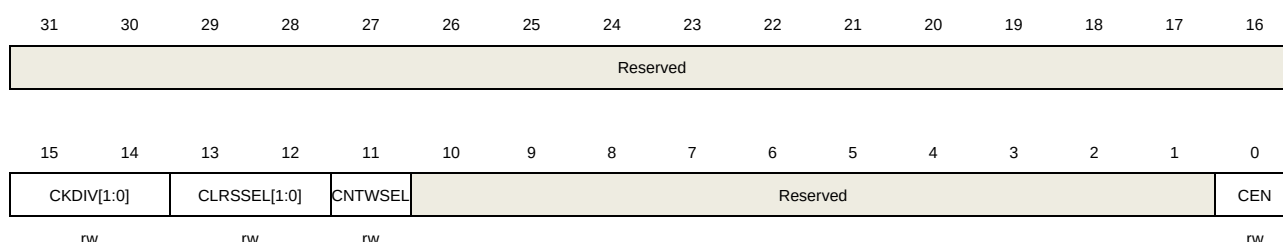
CPTIMERW base address: 0x4000 E000

23.4.1 Control register 0 (CPTIMERW_CTL0)

Address offset: 0x00

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



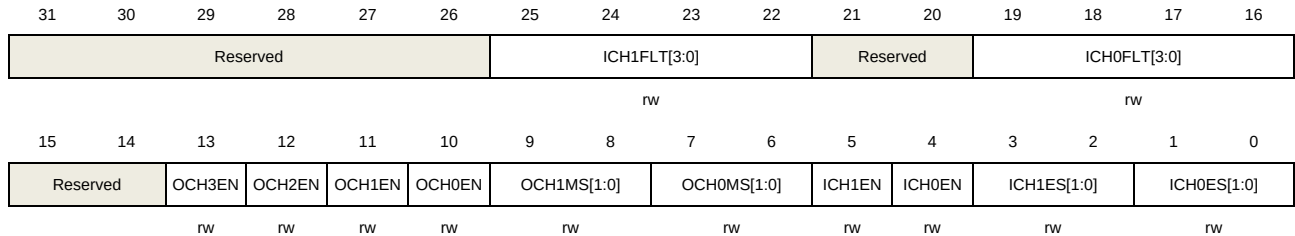
Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:14	CKDIV[1:0]	Input capture sample clock division The CKDIV bits can be configured by software to specify division factor between the CK_CPTIMERW and the digital filter sample clock (ICSCK). 00: $f_{ICSCK} = f_{CK\_CPTIMERW}$ 01: $f_{ICSCK} = f_{CK\_CPTIMERW} / 2$ 10: $f_{ICSCK} = f_{CK\_CPTIMERW} / 4$ 11: Reserved
13:12	CLRSSEL[1:0]	Counter clear source selection 01: Counter cleared at input capture channel 0 capture event 10: Counter cleared at input capture channel 1 capture event 00 / 11: Disable
11	CNTWSEL	Counter width selection 0: 32-bit counter mode 1: 16-bit counter mode
10:1	Reserved	Must be kept at reset value.
0	CEN	Counter enable 0: Counter disable 1: Counter enable The CEN bit can be set by other modules through EVIC automatically in counter enable mode.

23.4.2 Control register 1 (CPTIMERW_CTL1)

Address offset: 0x04

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



Bits	Fields	Descriptions
31:26	Reserved	Must be kept at reset value.
25:22	ICH1FLT[3:0]	Input capture channel 1 filter control Refer to ICH0FLT description.
21:20	Reserved	Must be kept at reset value.
19:16	ICH0FLT[3:0]	Input capture channel 0 filter control An event counter is used in the digital filter, in which a transition on the output occurs after N input events. This bit-field specifies the frequency f_{SAMP} used to sample ICH0 input signal and the length N of the digital filter applied to ICH0. 0000: Filter disabled, $f_{SAMP}=f_{ICSCCK}$, N=1. 0001: $f_{SAMP}=f_{CK\_TIMER}$, N=2. 0010: $f_{SAMP}=f_{CK\_TIMER}$, N=4. 0011: $f_{SAMP}=f_{CK\_TIMER}$, N=8. 0100: $f_{SAMP}=f_{ICSCCK}/2$, N=6. 0101: $f_{SAMP}=f_{ICSCCK}/2$, N=8. 0110: $f_{SAMP}=f_{ICSCCK}/4$, N=6. 0111: $f_{SAMP}=f_{ICSCCK}/4$, N=8. 1000: $f_{SAMP}=f_{ICSCCK}/8$, N=6. 1001: $f_{SAMP}=f_{ICSCCK}/8$, N=8. 1010: $f_{SAMP}=f_{ICSCCK}/16$, N=5. 1011: $f_{SAMP}=f_{ICSCCK}/16$, N=6. 1100: $f_{SAMP}=f_{ICSCCK}/16$, N=8. 1101: $f_{SAMP}=f_{ICSCCK}/32$, N=5. 1110: $f_{SAMP}=f_{ICSCCK}/32$, N=6. 1111: $f_{SAMP}=f_{ICSCCK}/32$, N=8.
15:14	Reserved	Must be kept at reset value.
13	OCH3EN	Output compare channel 3 enable Refer to OCH0EN description.

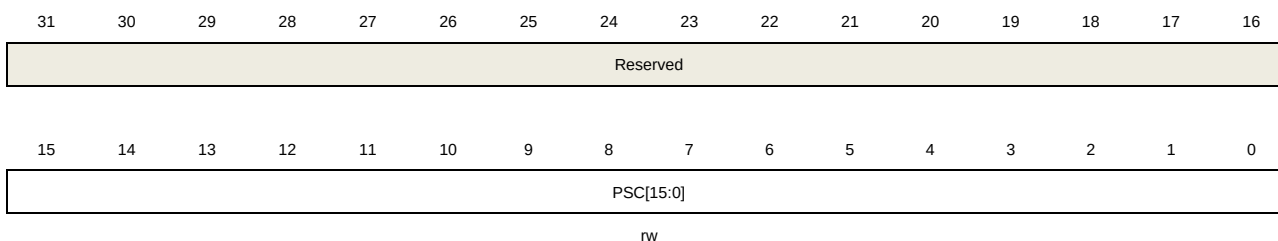
12	OCH2EN	Output compare channel 2 enable Refer to OCH0EN description.
11	OCH1EN	Output compare channel 1 enable Refer to OCH0EN description.
10	OCH0EN	Output compare channel 0 enable 0: Disable 1: Enable
9:8	OCH1MS[1:0]	Output compare channel 1 mode selection This bit-field specifies the output mode of output compare channel 00: Output maintain 01: Initial level is low and toggle on compare match 10: Initial level is high and toggle on compare match 11: Output disabled
7:6	OCH0MS[1:0]	Output compare channel 0 mode selection This bit-field specifies the output mode of output compare channel. 00: Output maintain 01: Initial level is low and toggle on compare match 10: Initial level is high and toggle on compare match 11: Output disabled
5	ICH1EN	Input capture channel 1 enable 0: Disable 1: Enable
4	ICH0EN	Input capture channel 0 enable 0: Disable 1: Enable
3:2	ICH1ES[1:0]	Input capture channel 1 edge selection 00: Rising edge 01: Falling edge 10: Both edges 11: Disable
1:0	ICH0ES[1:0]	Input capture channel 0 edge selection 00: Rising edge 01: Falling edge 10: Both edge 11: Disable

23.4.3 Prescaler register (CPTIMERW_PSC)

Address offset: 0x08

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



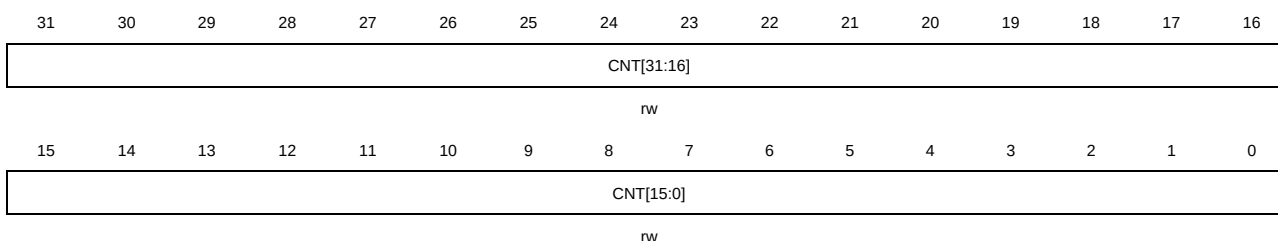
Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	PSC[15:0]	Prescaler value of the counter clock The PSC clock is divided by (PSC+1) to generate the counter clock. The value of this bit-field will be loaded to the corresponding shadow register at every update event.

23.4.4 Counter register (CPTIMERW_CNT)

Address offset: 0x0C

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



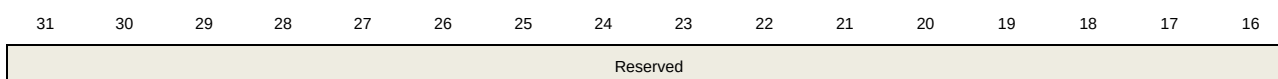
Bits	Fields	Descriptions
31:0	CNT[31:0]	This bit-field indicates the current counter value. Writing to this bit-field can change the value of the counter.

23.4.5 Interrupt enable register (CPTIMERW_INTEN)

Address offset: 0x10

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									UPIE	OCH3IE	OCH2IE	OCH1IE	OCH0IE	ICH1IE	ICH0IE
									rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:7	Reserved	Must be kept at reset value.
6	UPIE	Update interrupt enable 0: Interrupt disable 1: Interrupt enable
5	OCH3IE	Output compare channel 3 interrupt enable Refer to OCH0IE description.
4	OCH2IE	Output compare channel 2 interrupt enable Refer to OCH0IE description.
3	OCH1IE	Output compare channel 1 interrupt enable Refer to OCH0IE description.
2	OCH0IE	Output compare channel 0 interrupt enable 0: Interrupt disable 1: Interrupt enable
1	ICH1IE	Input capture channel 1 interrupt enable Refer to ICH0IE description.
0	ICH0IE	Input capture channel 0 interrupt enable 0: Interrupt disable 1: Interrupt enable

23.4.6 Interrupt flag register (CPTIMERW_INTF)

Address offset: 0x14

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									UPIF	OCH3IF	OCH2IF	OCH1IF	OCH0IF	ICH1IF	ICH0IF
									rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0

Bits	Fields	Descriptions
31:7	Reserved	Must be kept at reset value.

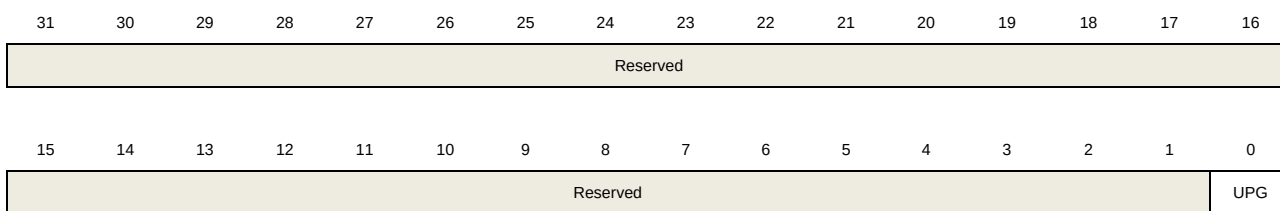
6	UPIF	Update interrupt flag This bit is set by hardware on an update event and cleared by software. The update event includes counter clear, counter overflow, EVIC counter restart event, and software update event. 0: No update interrupt occurred 1: Update interrupt occurred
5	OCH3IF	Output compare channel 3 interrupt flag Refer to OCH0IF description.
4	OCH2IF	Output compare channel 2 interrupt flag Refer to OCH0IF description.
3	OCH1IF	Output compare channel 1 interrupt flag Refer to OCH0IF description.
2	OCH0IF	Output compare channel 0 interrupt flag This bit is set by hardware on a compare match event and cleared by software. 0: No compare match interrupt occurred 1: Compare match interrupt occurred
1	ICH1IF	Input capture channel 1 interrupt flag Refer to ICH0IF description.
0	ICH0IF	Input capture channel 0 interrupt flag This bit is set by hardware on an input capture event and cleared by software. 0: No capture interrupt occurred 1: Capture interrupt occurred

23.4.7 Software event generation register (CPTIMERW_SWEVG)

Address offset: 0x18

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



w

Bits	Fields	Descriptions
31:1	Reserved	Must be kept at reset value.
0	UPG	Update event generation This bit can be set by software, and automatically cleared by hardware. When this

bit is set, the counter is cleared. The prescaler counter is cleared at the same time.

0: No generate an update event

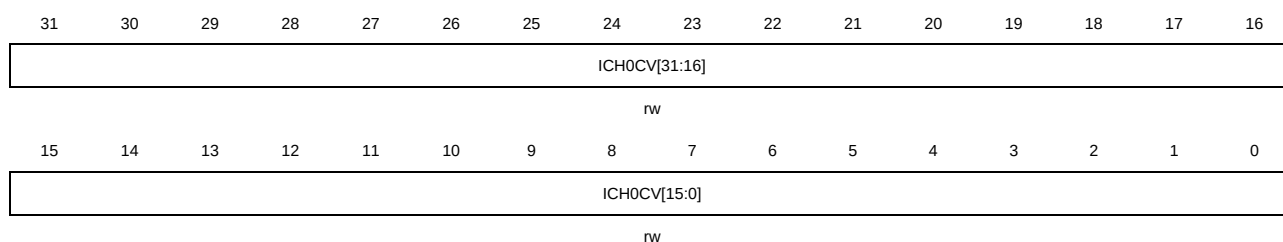
1: Generate an update event

23.4.8 Input capture channel 0 capture value register (CPTIMERW_ICH0CV)

Address offset: 0x1C

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



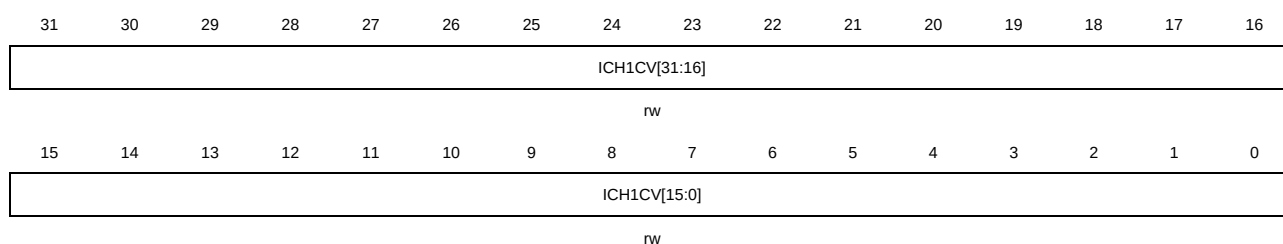
Bits	Fields	Descriptions
31:0	ICH0CV[31:0]	Input capture channel 0 capture value. This bit-field indicates the capture value of the input capture channel 0.

23.4.9 Input capture channel 1 capture value register (CPTIMERW_ICH1CV)

Address offset: 0x20

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



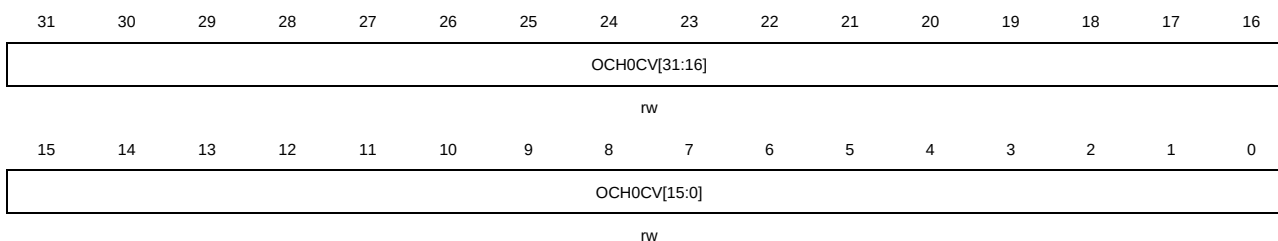
Bits	Fields	Descriptions
31:0	ICH1CV[31:0]	Input capture channel 1 capture value. This bit-field indicates the capture value of the input capture channel 1.

23.4.10 Output compare channel 0 compare value register (CPTIMERW_OCH0CV)

Address offset: 0x24

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



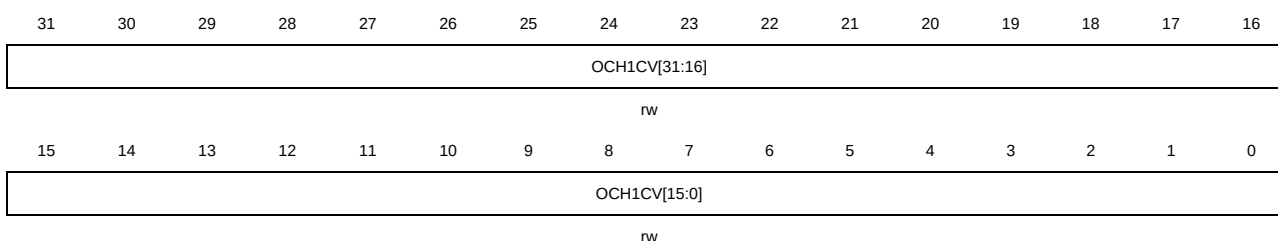
Bits	Fields	Descriptions
31:0	OCH0CV[31:0]	Output compare channel 0 compare value. This bit-field specifies the compare value to be compared with the counter to generate compare match event. The bit[31:16] should be set to 0 when the counter width is 16-bit.

23.4.11 Output compare channel 1 compare value register (CPTIMERW_OCH1CV)

Address offset: 0x28

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



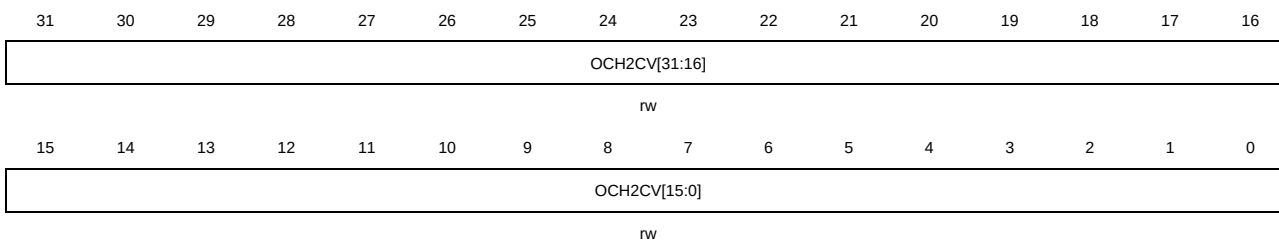
Bits	Fields	Descriptions
31:0	OCH1CV[31:0]	Output compare channel 1 compare value. This bit-field specifies the compare value to be compared with the counter to generate compare match event. The bit[31:16] should be set to 0 when the counter width is 16-bit.

23.4.12 Output compare channel 2 compare value register (CPTIMERW_OCH2CV)

Address offset: 0x2C

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



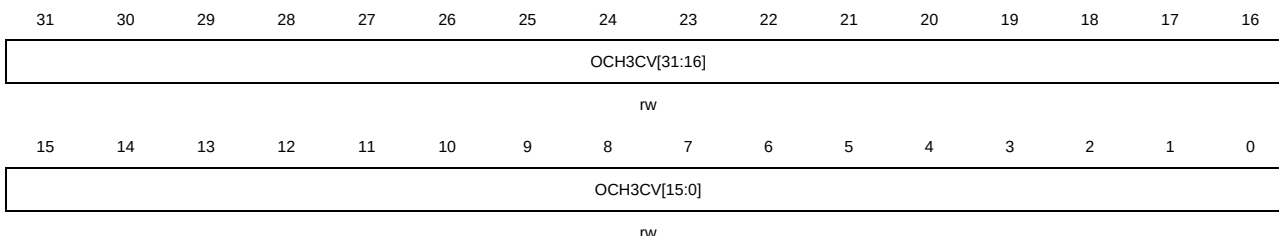
Bits	Fields	Descriptions
31:0	OCH2CV[31:0]	Output compare channel 2 compare value. This bit-field specifies the compare value to be compared with the counter to generate compare match event. The bit[31:16] should be set to 0 when the counter width is 16-bit.

23.4.13 Output compare channel 3 compare value register (CPTIMERW_OCH3CV)

Address offset: 0x30

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



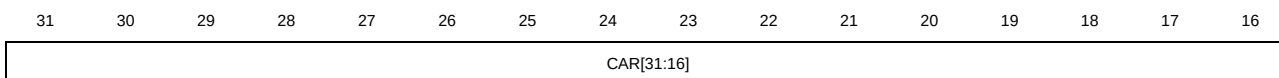
Bits	Fields	Descriptions
31:0	OCH3CV[31:0]	Output compare channel 3 compare value. This bit-field specifies the compare value to be compared with the counter to generate compare match event. The bit[31:16] should be set to 0 when the counter width is 16-bit.

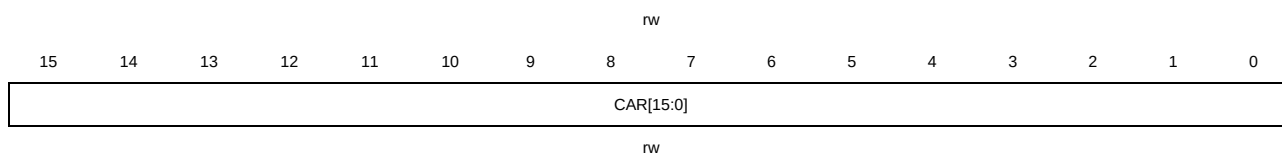
23.4.14 Counter auto reload register (CPTIMERW_CAR)

Address offset: 0x34

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).





Bits	Fields	Descriptions
31:0	CAR[31:0]	Counter auto reload value. This bit-field specifies the auto reload value of the counter. The bit[31:16] should be set to 0 when the counter width is 16-bit.

24 Compare timer (CPTIMER)

24.1 Overview

The compare timer module (CPTIMER0/1) has two independent 16-bit unsigned counters, two modules total 4 counters. The compare timer can be configured to generate interrupt or EVIC request at set period.

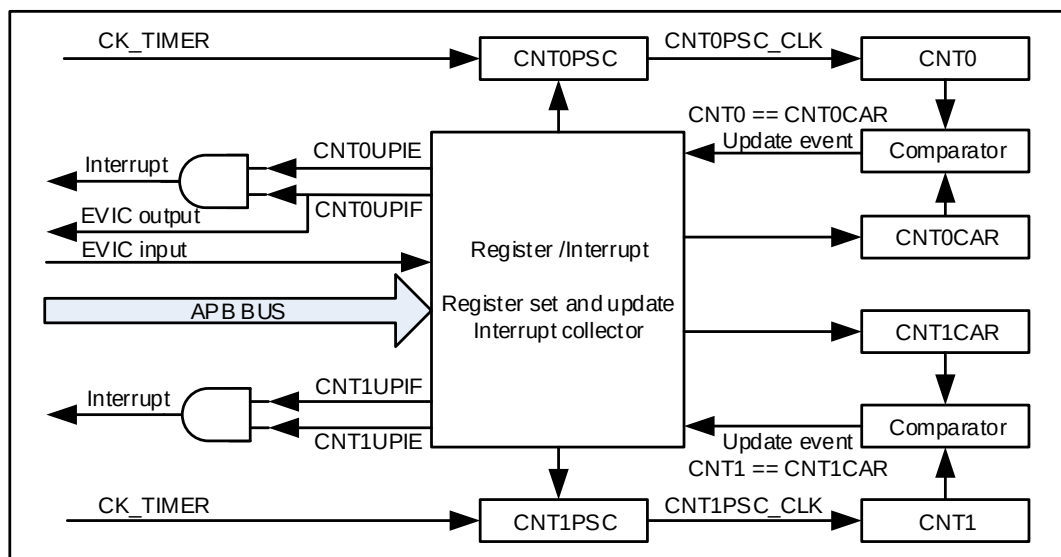
24.2 Characteristics

- Total counters: 2 counters each module.
- Counter width: 16-bit.
- Clock source: internal clock only.
- Counter mode: count up only.
- Programmable prescaler: 16-bits. Factor can be changed on the go.
- Auto reload function.
- Counter overflow event output to EVIC for trigger.
- Interrupt generate at update event.

24.3 Function overview

[Figure 24-1. Compare timer block diagram](#) provides details on the internal configuration of the compare timer.

Figure 24-1. Compare timer block diagram



24.3.1 Clock source

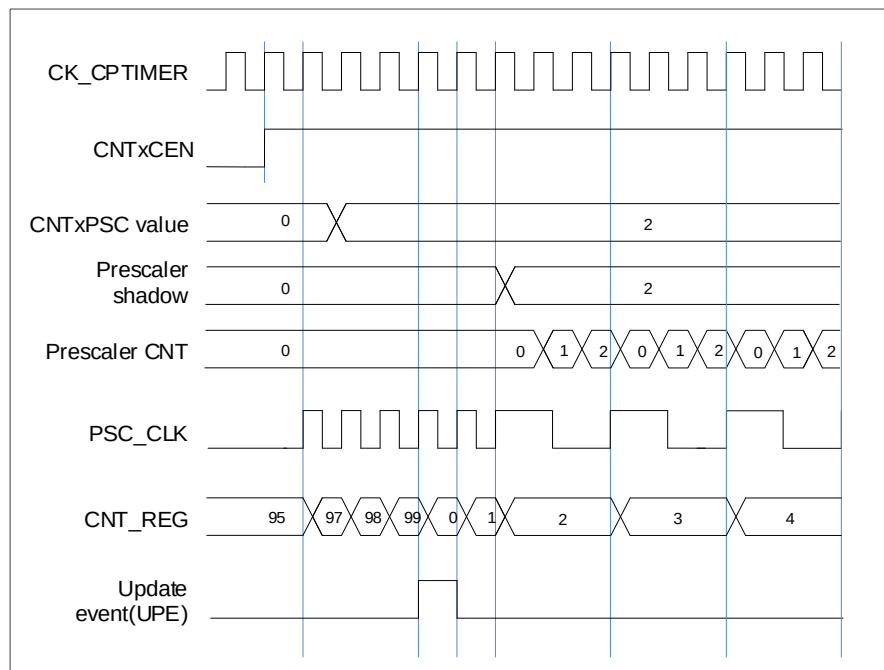
The CPTIMER can only be clocked by the internal timer clock, which is from the source named CK_CPTIMERx in RCU.

The CPTIMER_CK driven counter's prescaler to count is equal to CK_CPTIMERx. When the CENx is set, the CPTIMER_CK will be divided by CNTxPSC value to generate CNTxPSC_CLK.

24.3.2 Clock prescaler

The prescaler can divide the timer clock (CPTIMER_CK) to a counter clock (CNTxPSC_CLK) by any factor ranging from 1 to 65536. It is controlled by prescaler register (CPTIMER_CNTxPSC) which can be changed ongoing, but it will be loaded to the corresponding shadow register at next update event.

Figure 24-2. Timing chart of CNTxPSC value change from 0 to 2, CNTxCAR = 99

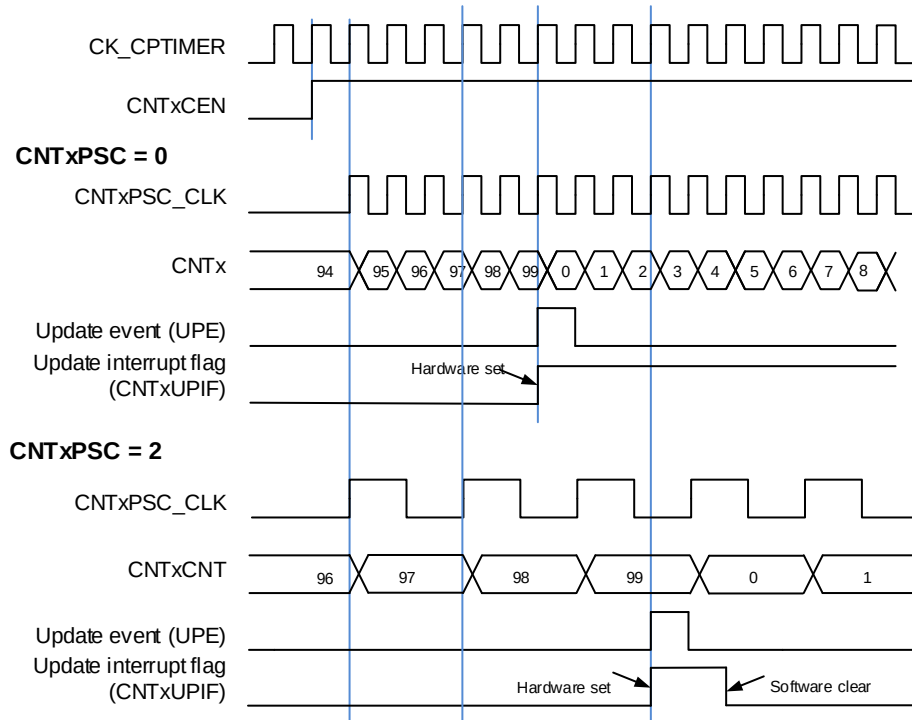


24.3.3 Counter up counting

In this mode, the counter counts up continuously from 0 to the counter-reload value, which is defined in the CPTIMER_CNTxCAR register, in a count-up direction. Once the counter reaches the counter reload value, the counter will restart counting up from 0 again. The update event is generated at each counter overflow.

The following figures show some examples of the counter behavior for different clock prescaler factor when the clearing value is 0x99.

Figure 24-3. Counter x timing chart of up counting mode, CNTxPSC=0/2



24.3.4 EVIC interconnect

The CPTIMER supports event signal output to or input from other modules through EVIC interconnect.

Note: Only counter 0 of CPTIMER0 supports EVIC interconnect.

EVIC interconnect output

The EVIC interconnect output function is for output the interrupt request of CPTIMER0 as event signal for other modules.

The counter 0 overflow event signal of CPTIMER0 can be output to other modules through EVIC regardless of the setting of interrupt enable bit. Please refer to EVIC chapter for more details.

EVIC interconnect input

The EVIC interconnect input function is for CPTIMER0 receiving the event signal from other modules to trigger different operations. The operation and event source can be one of the followings upon the setting in EVIC_SMCFG0 and EVIC_CPTIMER register of EVIC.

(1) Count enable

When the CPTIMER0 count enable mode is selected by the EVIC_SMCFG0 register of EVIC and the event specified by EVIC_CPTIMER register occurs, the CPTIMER0 counter

0 will be enable to count and the CEN0 bit in CPTIMER_CTL0 is set to 1.

(2) Count restart

When the CPTIMER0 count restart mode is selected by the EVIC_SMCFG0 register of EVIC and the event specified by EVIC_CPTIMER register occurs, the current counter value CNT0 will be reset to 0 and an update event will be generated. If the counter is enabled, the conuter will start count from 0 again.

(3) Event count

When the CPTIMER0 event count mode is selected by the EVIC_SMCFG0 register of EVIC and the counter is enabled, the CPTIMER0 counter 0 will count up at each event specified by EVIC_CPTIMER register regardless of the count clock source.

24.4 Register definition

CPTIMER0 base address: 0x4000 0000

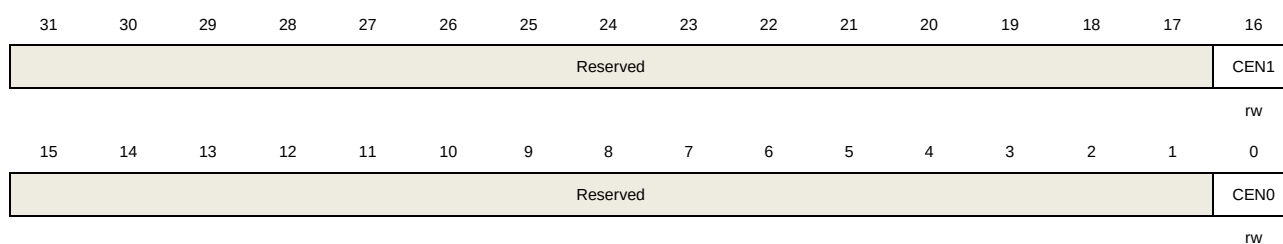
CPTIMER1 base address: 0x4000 0400

24.4.1 Control register 0 (CPTIMER_CTL0)

Address offset: 0x00

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



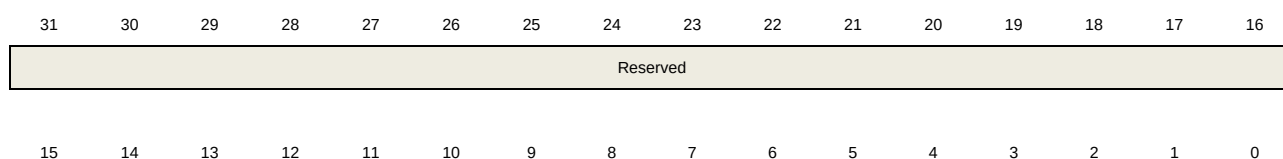
Bits	Fields	Descriptions
31:17	Reserved	Must be kept at reset value.
16	CEN1	Counter 1 enable 0: Counter disable 1: Counter enable The CEN1 bit only can be set/reset by software to enable/disable counter.
15:1	Reserved	Must be kept at reset value.
0	CEN0	Counter 0 enable 0: Counter disable 1: Counter enable The CEN0 bit can be set/reset by software to enable/disable counter or set by hardware automatically if EVIC event occurred in counter enable mode.

24.4.2 Counter 0 register (CPTIMER_CNT0)

Address offset: 0x04

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



CNT[15:0]																													
rw																													

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CNT[15:0]	This bit-field indicates the counter 0 current counter value. Writing to this bit-field can change the value of the counter.

24.4.3 Counter 1 register (CPTIMER_CNT1)

Address offset: 0x08

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

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

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CNT[15:0]	This bit-field indicates the counter 1 current counter value. Writing to this bit-field can change the value of the counter.

24.4.4 Counter 0 prescaler register (CPTIMER_CNT0PSC)

Address offset: 0x0C

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT0PSC[15:0]															
rw															

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.

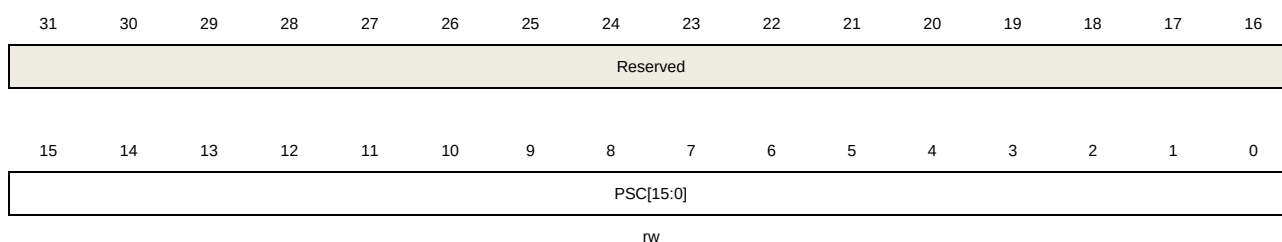
15:0	CNT0PSC[15:0]	Prescaler value of the counter 0 clock The PSC clock is divided by (PSC+1) to generate the counter clock. The value of this bit-field will be loaded to the corresponding shadow register at every update event.

24.4.5 Counter 1 prescaler register (CPTIMER_CNT1PSC)

Address offset: 0x10

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



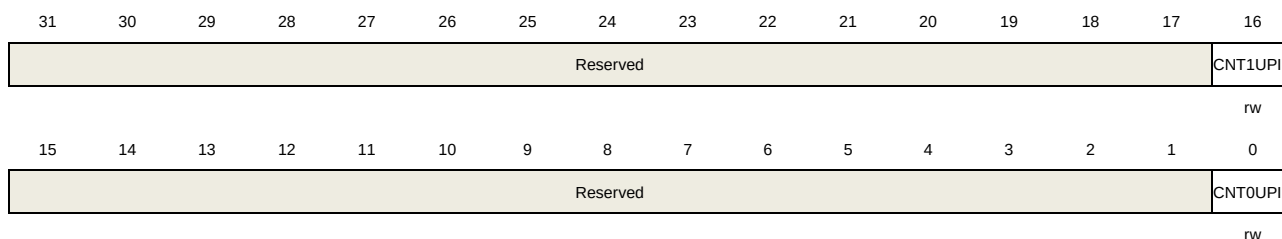
Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CNT1PSC[15:0]	Prescaler value of the counter 1 clock The PSC clock is divided by (PSC+1) to generate the counter clock. The value of this bit-field will be loaded to the corresponding shadow register at every update event.

24.4.6 Interrupt enable register (CPTIMER_INTEN)

Address offset: 0x14

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



Bits	Fields	Descriptions
31:17	Reserved	Must be kept at reset value.
16	CNT1UIPIE	Counter 1 update interrupt enable 0: Interrupt disable

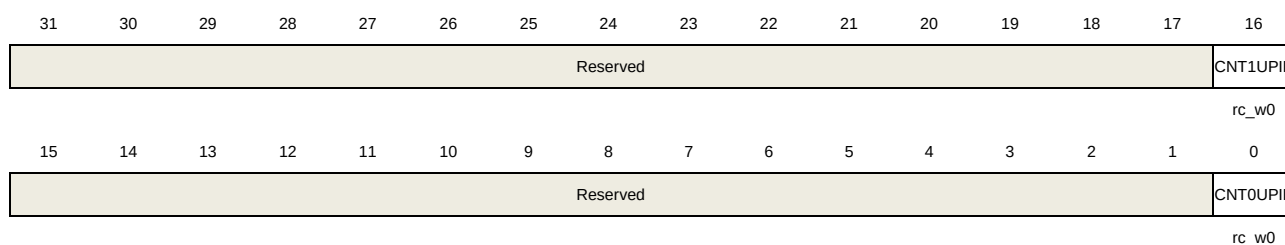
		1: Interrupt enable
15:1	Reserved	Must be kept at reset value.
0	CNT0UPIE	Counter 0 update interrupt enable
		0: Interrupt disable
		1: Interrupt enable

24.4.7 Interrupt flag register (CPTIMER_INTF)

Address offset: 0x18

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



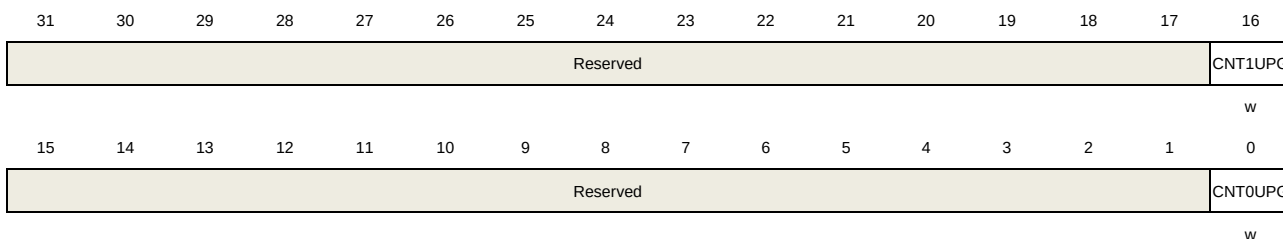
Bits	Fields	Descriptions
31:17	Reserved	Must be kept at reset value.
16	CNT1UPIF	Counter 1 update interrupt flag This bit is set by hardware when an update event occurs and cleared by software. The update event includes counter overflow, software update event and EVIC restart event. 0: No update interrupt occurred 1: Update interrupt occurred
15:1	Reserved	Must be kept at reset value.
0	CNT0UPIF	Counter 0 update interrupt flag This bit is set by hardware when an update event occurs and cleared by software. The update event includes counter overflow, software update event and EVIC restart event. 0: No update interrupt occurred 1: Update interrupt occurred

24.4.8 Software event generation register (CPTIMER_SWEVG)

Address offset: 0x1C

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



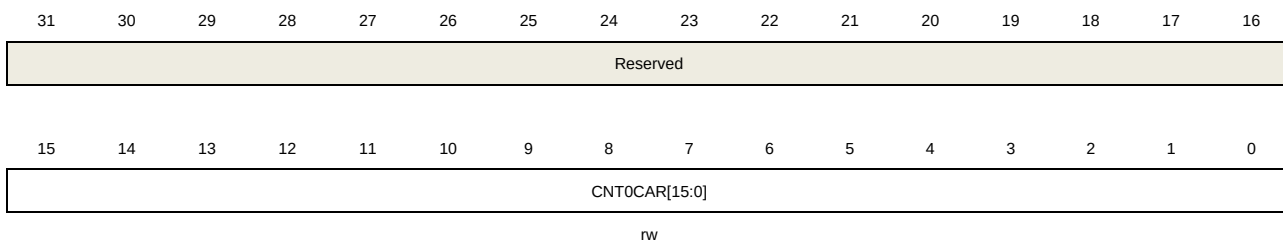
Bits	Fields	Descriptions
31:17	Reserved	Must be kept at reset value.
16	CNT1UPG	Counter 1 update event generation This bit can be set by software, and automatically cleared by hardware. When this bit is set, the counter is cleared. The prescaler counter is cleared at the same time. 0: No generate an update event 1: Generate an update event
15:1	Reserved	Must be kept at reset value.
0	CNT0UPG	Counter 0 update event generation This bit can be set by software, and automatically cleared by hardware. When this bit is set, the counter is cleared. The prescaler counter is cleared at the same time. 0: No generate an update event 1: Generate an update event

24.4.9 Counter 0 auto reload register (CPTIMER_CNT0CAR)

Address offset: 0x20

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



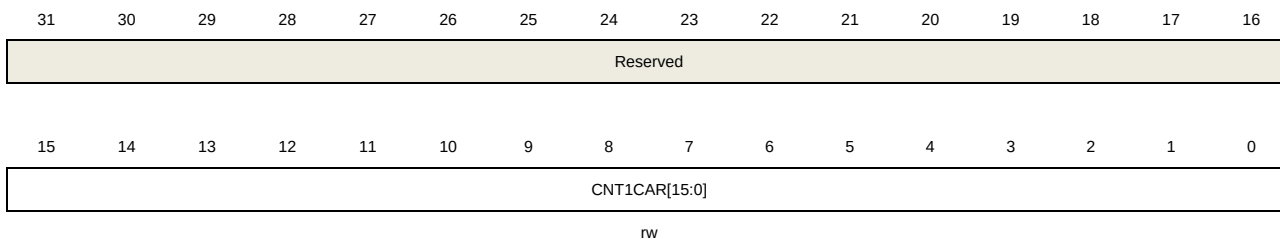
Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CNT0CAR[15:0]	Counter 0 auto reload value. This bit-field specifies the auto reload value of the counter.

24.4.10 Counter 1 auto reload register (CPTIMER_CNT1CAR)

Address offset: 0x24

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CNT1CAR[15:0]	Counter 1 auto reload value. This bit-field specifies the auto reload value of the counter.

25 Universal asynchronous receiver / transmitter (UART)

25.1 Overview

The Universal Asynchronous Receiver / Transmitter (UART) provides a flexible serial data exchange interface. Data frames can be transferred in full duplex or half duplex mode, asynchronously through this interface. A programmable baud rate generator divides the PCLK1 to produce a dedicated baud rate lock for the UART transmitter and receiver.

Besides the standard asynchronous receiver and transmitter mode, the UART implements several other types of serial data exchange modes, such as IrDA (infrared data association) SIR mode, LIN (local interconnection network) mode. It also supports multiprocessor communication mode. The data frame can be transferred from LSB or MSB bit. The polarity of the data bits and the TX / RX pins can be configured independently and flexibly.

ALL UARTs support DMA function for high-speed data communication.

25.2 Characteristics

- NRZ standard format
- Full duplex communication
- Half duplex single wire communications
- Programmable baud-rate generator
 - Divided from the peripheral clocks, PCLK1 for UART0/1/2/3.
 - Oversampling by 8 or 16
 - Maximum speed up to 12 Mbits/s (PCLK1 96M and oversampling by 8)
- Fully programmable serial interface characteristics:
 - Even, odd or no-parity bit generation/detection
 - A data word length can be 7, 8, 9 or 10 bits
 - 1 or 2 stop bit generation
- Transmitter and receiver can be enabled separately
- DMA request for data buffer access
- LIN break generation and detection
- IrDA support
- Multiprocessor communication
 - Enter into mute mode if address match does not occur
 - Wake up from mute mode by idle frame or address match detection
- Various status flags:
 - Flags for transfer detection: receive buffer not empty (RBNE), transmit buffer empty (TBE), transfer complete (TC), and busy (BSY).
 - Flags for error detection: overrun error (ORERR), noise error (NERR), frame error (FERR) and parity error (PERR)

- Flag for LIN mode: LIN break detected (LBDF)
- Flag for multiprocessor communication: IDLE frame detected (IDLEF)
- Interrupt occurs at these events when the corresponding interrupt enable bits are set

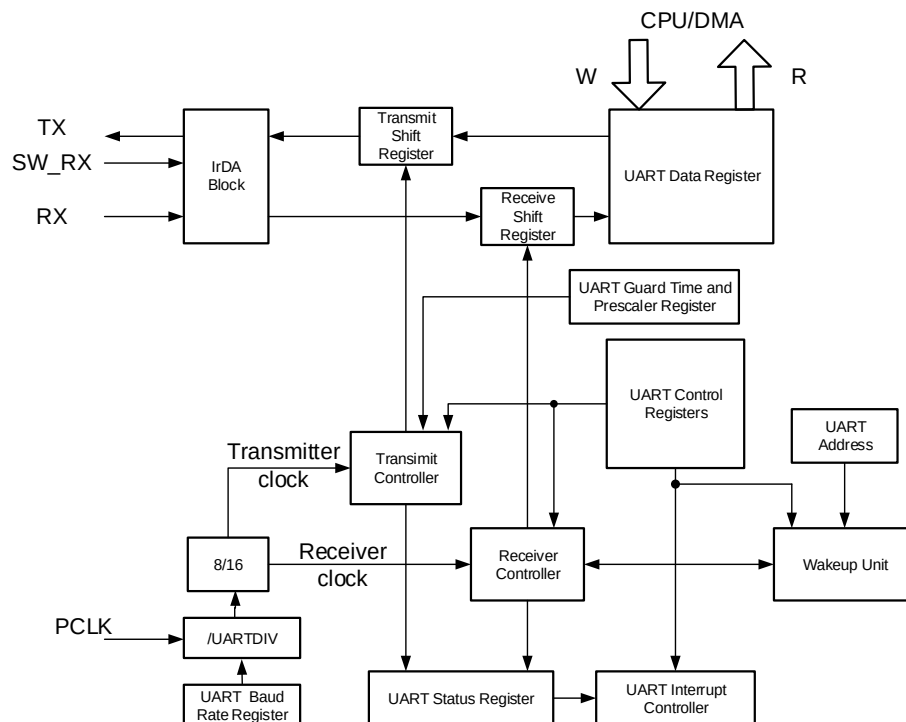
25.3 Function overview

The interface is externally connected to another device by the main pins listed in [Table 25-1. Description of UART important pins.](#)

Table 25-1. Description of UART important pins

Pin	Type	Description
RX	Input	Receive data
TX	Output I/O (single-wire)	Transmit data. High level when enabled but nothing to be transmitted

Figure 25-1. UART module block diagram

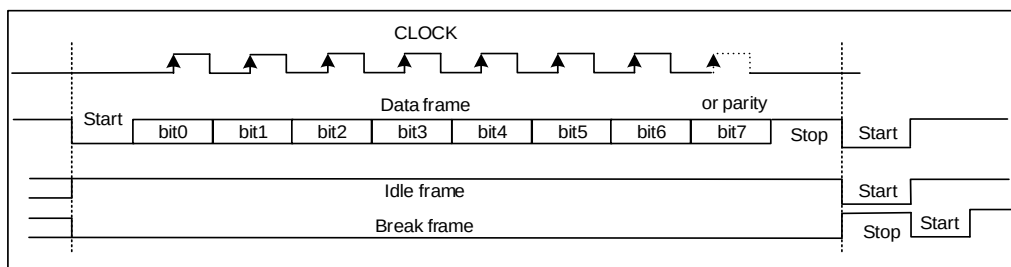


25.3.1 UART frame format

The UART frame starts with a start bit and end up with a number of stop bits. The length of the data frame is configured by the WL bit in the UART_CTL0 register. The last data bit can be used as parity check bit by setting the PCEN bit in UART_CTL0 register. When the WL bit is set to 0b00, the parity bit is the 7th bit. When the WL bit is set to 0b01, the parity bit is the 8th bit. When the WL bit is set to 0b10, the parity bit is the 6th bit. When the WL bit is set to 0b11, the parity bit is the 9th bit. The method of calculating the parity bit is selected by the PM

bit in UART_CTL0 register.

Figure 25-2. UART character frame (8 bits data and 1 stop bit)



In transmission and reception, the number of stop bits can be configured by the STB[1:0] bits in the UART_CTL1 register.

Table 25-2. Configuration of stop bits

STB[1:0]	stop bit length (bit)	usage description
00	1	default value
01	0.5	/
10	2	Normal UART and single-wire modes
11	1.5	/

In an idle frame, all the frame bits are logic 1. The frame length is equal to the normal UART frame.

The break frame structure is a number of low bits followed by the configured number of stop bits. The transfer speed of a UART frame depends on the frequency of the PCLK, the configuration of the baud rate generator and the oversampling mode.

25.3.2 Baud rate generation

The baud-rate divider is a 20-bit number which consists of a 16-bit integer and a 4-bit fractional part. The number formed by these two values is used by the baud rate generator to determine the bit period. Having a fractional baud-rate divider allows the UART to generate all the standard baud rates.

The baud-rate divider (UARTDIV) has the following relationship with the peripheral clock:

In case of oversampling by 16, the equation is:

$$\text{UARTDIV} = \frac{\text{PCLK}}{16 \times \text{Baud Rate}} \quad (25-1)$$

In case of oversampling by 8, the equation is:

$$\text{UARTDIV} = \frac{\text{PCLK}}{8 \times \text{Baud Rate}} \quad (25-2)$$

For example, when oversampled by 16:

1. Get UARTDIV by calculating the value of UART_BUAD:

If $\text{UART\_BUAD}=0x21D$, then $\text{INTDIV}=33$ ($0x21$), $\text{FRADIV}=13$ ($0xD$).

$\text{UARTDIV}=33+13/16=33.81$.

2. Get the value of UART_BUAD by calculating the value of UARTDIV :

If $\text{UARTDIV}=30.37$, then $\text{INTDIV}=30$ ($0x1E$).

$16 \times 0.37 = 5.92$, the nearest integer is 6, so $\text{FRADIV}=6$ ($0x6$).

$\text{UART\_BUAD}=0x1E6$.

Note: If the roundness of FRADIV is 16 (overflow), the carry must be added to the integer part.

25.3.3 UART transmitter

If the transmit enable bit (TEN) in UART_CTL0 register is set, when the transmit data buffer is not empty, the transmitter shift out the transmit data frame through the TX pin. The polarity of the TX pin can be configured by the TINV bit in the UART_CTL2 register. Clock pulses can be output through the CK pin.

After the TEN bit is set, an idle frame will be sent. The TEN bit should not be cleared while the transmission is ongoing.

After power on, the TBE bit is high by default. Data can be written to the UART_DATA when the TBE bit of the UART_STAT0 register is asserted. The TBE bit is cleared by writing UART_DATA register and it is set by hardware after the data is put into the transmit shift register. If a data is written to the UART_DATA register while a transmission is ongoing, it will be firstly stored in the transmit buffer, and transferred to the transmit shift register after the current transmission is done. If a data is written to the UART_DATA register while no transmission is ongoing, the TBE bit will be cleared and set soon, because the data will be transferred to the transmit shift register immediately.

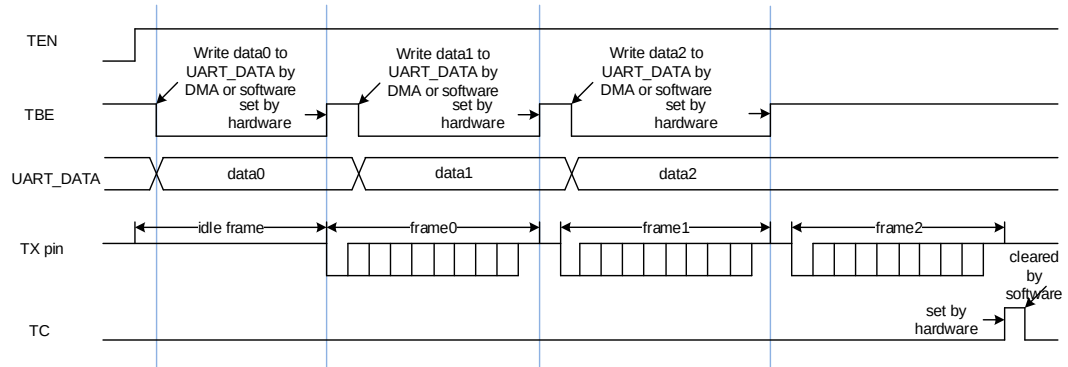
If a frame is transmitted and the TBE bit is asserted, the TC bit of the UART_STAT0 register will be set. An interrupt will be generated if the corresponding interrupt enable bit (TCIE) is set in the UART_CTL0 register.

The UART transmit procedure is shown in [Figure 25-3. UART transmit procedure](#). The software can follow this flow:

1. Write the WL bit in UART_CTL0 to set the data bits length.
2. Set the STB[1:0] bits in UART_CTL1 to configure the number of stop bits.
3. Enable DMA (DENT bit) in UART_CTL2 if multibuffer communication is selected.
4. Set the baud rate in UART_BAUD .
5. Set the UEN bit in UART_CTL0 to enable the UART.
6. Set the TEN bit in UART_CTL0 .
7. Wait for the TBE being asserted.
8. Write the data to in the UART_DATA register.

9. Repeat step7- 8 for each data, if DMA is not enabled.
10. Wait until TC=1 to finish.

Figure 25-3. UART transmit procedure



It is necessary to wait for the TC bit to be asserted before disabling the UART or entering the power saving mode. This bit can be cleared by a software sequence: reading the UART_STAT0 register and then writing the UART_DATA register. If the multibuffer communication is selected (DENT=1), this bit can also be cleared by writing 0 directly.

25.3.4 UART receiver

After power on, the UART receiver can be enabled by the follow procedure:

1. Write the WL bit in UART_CTL0 to set the data bits length.
2. Set the STB[1:0] bits in UART_CTL1.
3. Enable DMA (DENR bit) in UART_CTL2 if multibuffer communication is selected.
4. Set the baud rate in UART_BAUD.
5. Set the UEN bit in UART_CTL0 to enable the UART.
6. Set the REN bit in UART_CTL0.

After being enabled, the receiver receives a bit stream after a valid start pulse has been detected. Detection on noisy error, parity error, frame error and overrun error is performed during the reception of a frame.

When a frame is received, the RBNE bit in UART_STAT0 is asserted, an interrupt is generated if the corresponding interrupt enable bit (RBNEIE) is set in the UART_CTL0 register. The status of the reception are stored in the UART_STAT0 register.

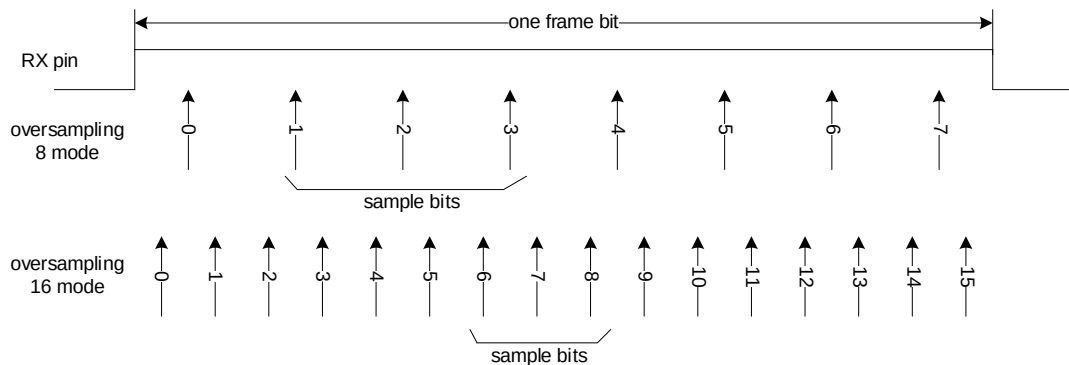
The software can get the received data by reading the UART_DATA register directly, or through DMA. The RBNE bit is cleared by a read operation on the UART_DATA register, whatever it is performed by software directly, or through DMA.

The REN bit should not be disabled when reception is ongoing, or the current frame will be lost.

By default, the receiver gets three samples to evaluate the value of a frame bit. If the oversampling 8 mode is enabled, the 1st, 2th and 3th samples are used, while in the oversampling 16 mode, the 6th, 7th, and 8th samples are used. If there are two or more

samples of a frame bit is 0, the frame bit is confirmed as a 0, else 1. If the three samples of any bit of a frame are not the same, whatever it is a data bit, parity bit or stop bit, a noisy error (NERR) status will be generated for the frame. An interrupt is generated, If the receive DMA is enabled and the ERRIE bit in UART_CTL2 register is set. If the OSB bit in UART_CTL2 register is set, the receiver get only one sample to evaluate a bit value. In this situation, no noisy error will be detected.

Figure 25-4. Receiving a frame bit by oversampling method (OSB=0)



If the parity check function is enabled by setting the PCEN bit in the UART_CTL0 register, the receiver calculates the expected parity value while receiving a frame. The received parity bit will be compared with this expected value. If they are not the same, the parity error (PERR) bit in UART_STAT0 register will be set. An interrupt is generated, if the PERRIE bit in UART_CTL0 register is set.

If the RX pin is evaluated as 0 during a stop bit, the frame error (FERR) bit in UART_STAT0 register will be set. An interrupt will be generated if the receive DMA is enabled and the ERRIE bit in UART_CTL2 register is set.

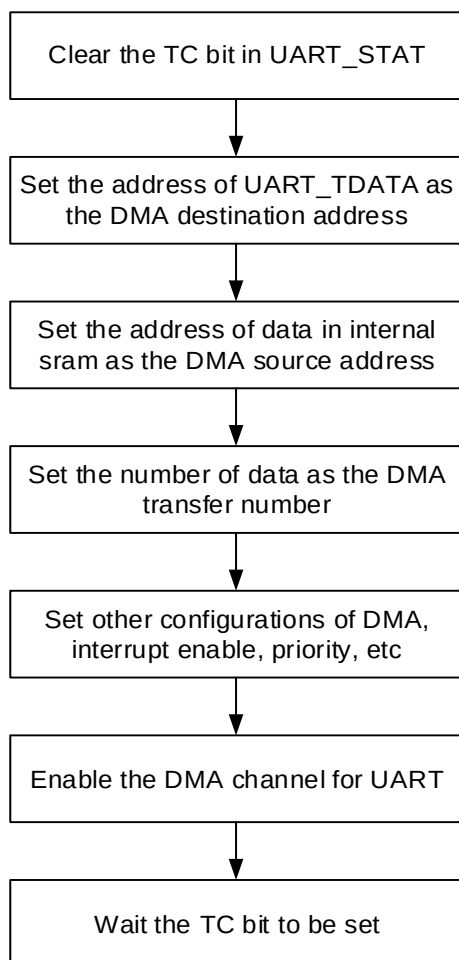
When a frame is received, if the RBNE bit is not cleared yet, the last frame will not be stored in the receive data buffer. The overrun error (ORERR) bit in UART_STAT0 register will be set. An interrupt is generated, if the receive DMA is enabled and the ERRIE bit in UART_CTL2 register is set, or if the RBNEIE is set.

If a noise error (NERR), parity error (PERR), frame error (FERR) or overrun error (ORERR) is generated during a receiving process, then NERR, PERR, FERR or ORERR will be set at same time with RBNE. If DMA is disabled, the software needs to check whether the RBNE interrupt is caused by noise error, parity error, framing error or overflow error when the RBNE interrupt occurs.

25.3.5 Use DMA for data buffer access

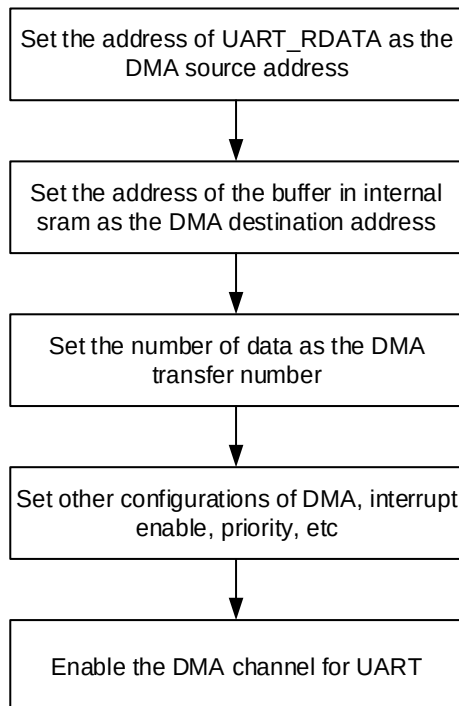
To reduce the burden of the processor, DMA can be used to access the transmitting and receiving data buffer. The DENT bit in UART_CTL2 is used to enable the DMA transmission, and the DENR bit in UART_CTL2 is used to enable the DMA reception.

When DMA is used for UART transmission, DMA transfers data from internal SRAM to the transmit data buffer of the UART. The configuration step are shown in [Figure 25-5](#).

Configuration step when use DMA for UART transmission**Figure 25-5. Configuration step when use DMA for UART transmission**

After all of the data frames are transmitted, the TC bit in UART_STAT0 is set. An interrupt occurs if the TCIE bit in UART_CTL0 is set.

When DMA is used for UART reception, DMA transfers data from the receive data buffer of the UART to the internal SRAM. The configuration step is shown in [Figure 25-6. Configuration step when use DMA for UART reception](#). If the ERRIE bit in UART_CTL2 is set, interrupts can be generated by the error status bits (FERR, ORERR and NERR) in UART_STAT0.

Figure 25-6. Configuration step when use DMA for UART reception

When the number of the data received by UART reaches the DMA transfer number, an end of transfer interrupt can be generated in the DMA module.

25.3.6 Multi-processor communication

In multiprocessor communication, several UARTs are connected as a network. It will be a big burden for a device to monitor all of the messages on the RX pin. To reduce the burden of a device, software can put an UART module into a mute mode by setting the RWU bit in UART_CTL0 register.

If a UART is in mute mode, all of the receive status bits cannot be set. Software can wake up the UART by resetting the RWU bit.

The UART can also be wake up by hardware by one of the two methods: idle frame method and address match method.

The idle frame wake up method is selected by default. When an idle frame is detected on the RX pin, the hardware clears the RWU bit and exits the mute mode.

When the WM bit in UART_CTL0 register is set, the MSB bit of a frame is detected as the address flag. If the address flag is high, the frame is treated as an address frame. If the address flag is low, the frame is treated as a data frame. If the LSB 4 bits of an address frame are the same as the ADDR[3:0] bits in the UART_CTL1 register, the hardware will clear the RWU bit and exit the mute mode. The RBNE bit will be set for the frame that wakes up the UART. The status bits are available in the UART_STAT0 register. If the LSB 4 bits of an address frame differ from the ADDR[3:0] bits in the UART_CTL1 register, the hardware sets

the RWU bit and enters mute mode automatically. In this situation, the RBNE bit is not set.

If the address match method is selected, the receiver does not check the parity value of an address frame by default. If the PCM bit in UART_CHC and PCEN bit in UART_CTL0 are set, the MSB bit will be checked as the parity bit, and the bit preceding the MSB bit is detected as the address flag.

The RWU bit can be written to as 0 or 1 when the receiver buffer contains no data (RBNE=0 in the UART_STAT0 register).

25.3.7 LIN mode

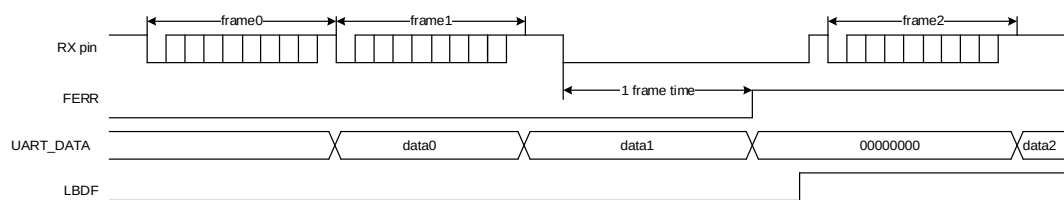
The local interconnection network mode is enabled by setting the LMEN bit in UART_CTL1. The WL in UART_CTL0, STB[1:0] bits in UART_CTL1 and the HDEN, IREN bits in UART_CTL2 should be reset in LIN mode.

When transmit a normal data frame, the transmission procedure is the same as the normal UART mode. When the SBKCMD bit in UART_CTL0 is set, the UART transmit continuous 13 '0' bits, followed by 1 stop bit.

The break detection function is totally independent from the normal UART receiver. So a break frame can be detected during the idle state or during a frame. The expected length of a break frame can be selected by LBLLEN in UART_CTL1. When the RX pin is detected at low state for a time that is equal to or longer than the expected break frame length (10 bits when LBLLEN=0, or 11 bits when LBLLEN=1), the LBDF in UART_STAT0 is set. An interrupt occurs if the LBDIE bit in UART_CTL1 is set.

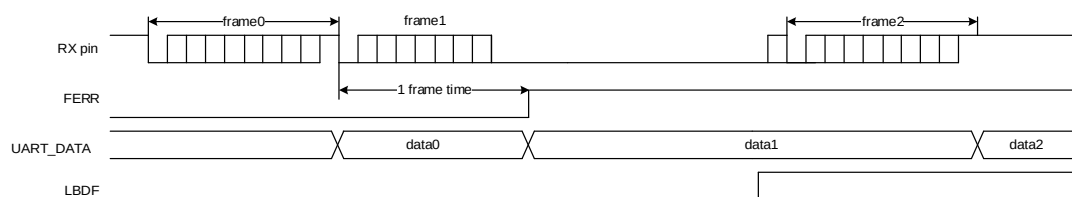
As shown in [Figure 25-7. Break frame occurs during idle state](#), if a break frame occurs during the idle state on the RX pin, the UART receiver will receive an all '0' frame, with an asserted FERR status.

Figure 25-7. Break frame occurs during idle state



As shown in [Figure 25-8. Break frame occurs during a frame](#), if a break frame occurs during a frame on the RX pin, the FERR status will be asserted for the current frame.

Figure 25-8. Break frame occurs during a frame

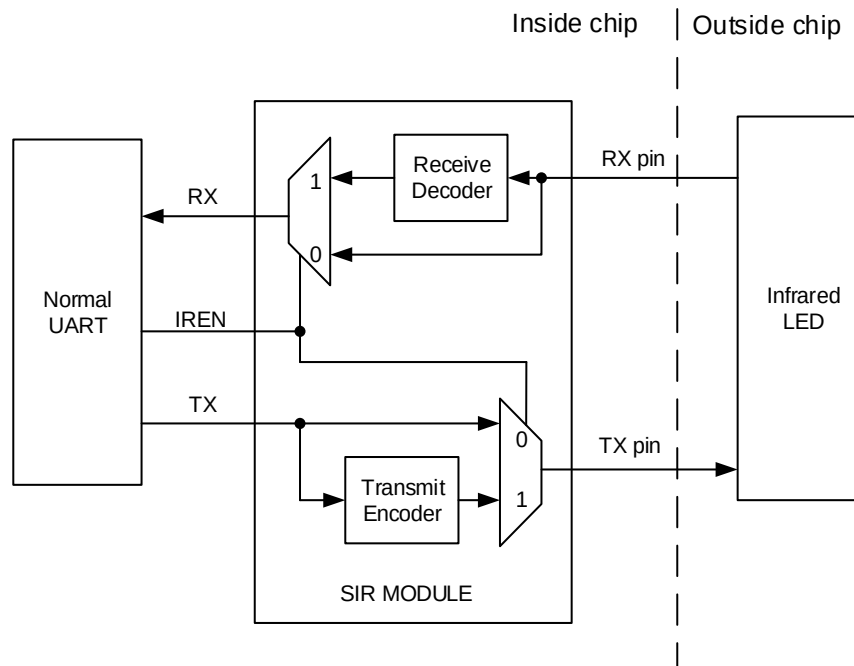


25.3.8 IrDA SIR ENDEC mode

The IrDA mode is enabled by setting the IREN bit in UART_CTL2. The LMEN, STB[1:0] bits in UART_CTL1 and HDEN bits in UART_CTL2 should be cleared in IrDA mode.

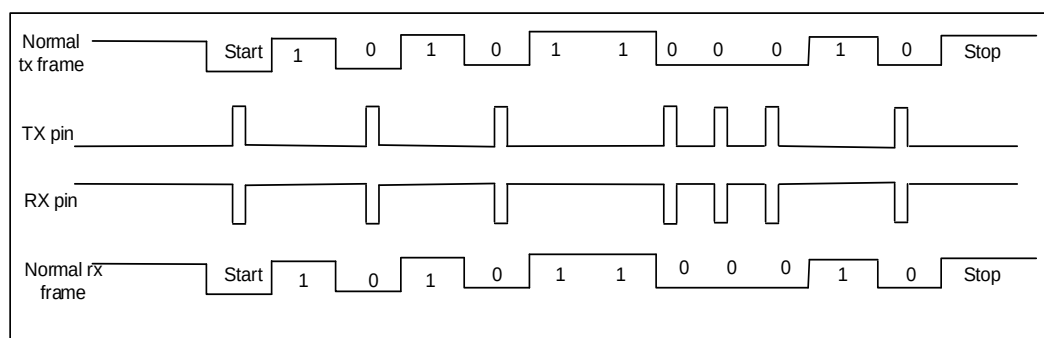
In IrDA mode, the UART transmission data frame is modulated in the SIR transmit encoder and transmitted to the infrared LED through the TX pin. The SIR receive decoder receives the modulated signal from the infrared LED through the RX pin, and put the demodulated data frame to the UART receiver. The baud rate should not be larger than 115200 for the encoder.

Figure 25-9. IrDA SIR ENDEC module



In IrDA mode, the polarity of the TX pin and RX pin is different. The TX pin is usually at low state, while the RX pin is usually at high state. The IrDA pins keep stable to represent the logic '1', while an infrared light pulse on the IrDA pins (a Return to Zero signal) represent the logic '0'. The pulse width should be 3/16 of a bit period. The IrDA could not detect any pulse if the pulse width is less than 1 PSC clock. While it can detect a pulse by chance if the pulse width is greater than 1 but smaller than 2 times PSC clock.

Because the IrDA is a half-duplex protocol, the transmission and the reception should not be carried out at the same time in the IrDA SIR ENDEC block.

Figure 25-10. IrDA data modulation


The SIR sub module can work in low power mode by setting the IRLP bit in UART_CTL2. The transmit encoder is driven by a low speed clock, which is divided from the PCLK. The divide ratio is configured by the PSC[7:0] bits in UART_GP register, UART_BAUD register needs to be configured as $16 \times \text{PSC}[7:0]$. The pulse width on the TX pin is 3 cycles of this low speed period. The receiver decoder works in the same manner as the normal IrDA mode.

25.3.9 Half-duplex communication mode

The half-duplex communication mode is enabled by setting the HDEN bit in UART_CTL2. The LMEN bits in UART_CTL1 and IREN bits in UART_CTL2 should be cleared in half-duplex communication mode.

In the half-duplex mode the receive line is internally connected to the TX pin, and the RX pin is no longer used. The TX pin should be configured as output open drain mode. The software should make sure that the transmission and reception process never conflict with each other.

25.3.10 UART interrupts

The UART interrupt events and flags are listed in [Table 25-3. UART interrupt requests](#).

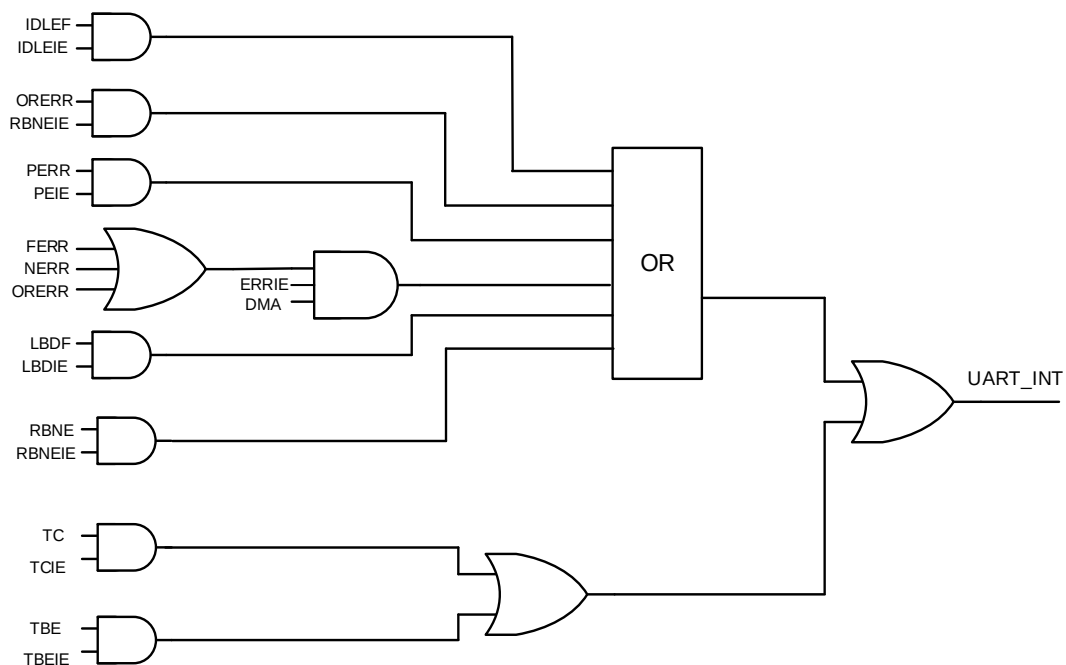
Table 25-3. UART interrupt requests

Interrupt event	Event flag	Enable Control bit
Transmit data buffer empty	TBE	TBEIE
Transmission complete	TC	TCIE
Received buff not empty	RBNE	RBNEIE
Overrun error	ORERR	
Idle frame	IDLEF	IDLEIE
Parity error	PERR	PERRIE
Break detected flag in LIN mode	LBDF	LBDIE
Reception errors (noise flag, overrun error, framing error) in DMA reception	NERR or ORERR or FERR	ERRIE

All of the interrupt events are ORed together before being sent to the interrupt controller, so

the UART can only generate a single interrupt request to the controller at any given time. Software can service multiple interrupt events in a single interrupt service routine.

Figure 25-11. UART interrupt mapping diagram



25.4 Register definition

UART0 base address: 0x4000 4C00

UART1 base address: 0x4001 5000

UART2 base address: 0x4001 7800

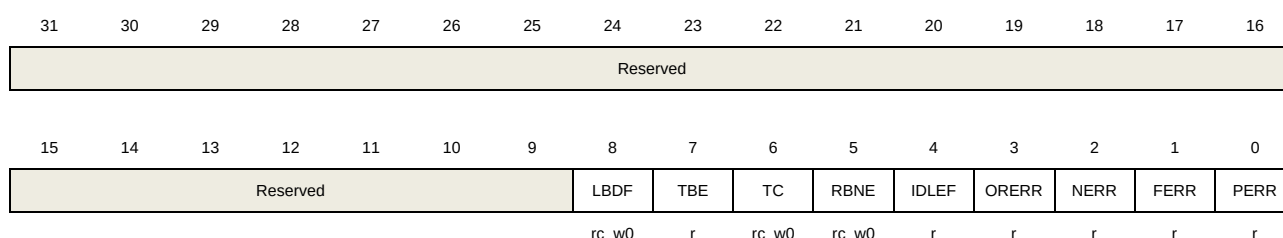
UART3 base address: 0x4000 7C00

25.4.1 Status register 0 (UART_STAT0)

Address offset: 0x00

Reset value: 0x0000 00C0

This register has to be accessed by word (32-bit).



Bits	Fields	Descriptions
31:9	Reserved	Must be kept at reset value.
8	LBD F	LIN break detected flag LMEN bit in UART_CTL1 is set when LIN break is detected. An interrupt occurs if the LBDIE bit in UART_CTL1 is set. Software can clear this bit by writing 0 to it. 0: The UART does not detect a LIN break. 1: The UART has detected a LIN break.
7	TBE	Transmit data buffer empty. This bit is set after power on or when the transmit data has been transferred to the transmit shift register. An interrupt occurs if the TBEIE bit in UART_CTL0 is set. This bit is cleared when the software write transmit data to the UART_DATA register. 0: Transmit data buffer is not empty. 1: Transmit data buffer is empty.
6	TC	Transmission complete. This bit is set after power on. If the TBE bit has been set, this bit is set when the transmission of current data is complete. An interrupt occurs if the TCIE bit in UART_CTL0 is set. Software can clear this bit by writing 0 to it.

		0: Transmission of current data is not complete. 1: Transmission of current data is complete.
5	RBNE	Read data buffer not empty. This bit is set when the read data buffer is filled with a data frame, which has been received through the receive shift register. An interrupt occurs if the RBNEIE bit in UART_CTL0 is set. Software can clear this bit by writing 0 to it or by reading the UART_DATA register. 0: Read data buffer is empty. 1: Read data buffer is not empty.
4	IDLEF	IDLE frame detected flag. This bit is set when the RX pin has been detected in idle state for a frame time. An interrupt occurs if the IDLEIE bit in UART_CTL0 is set. Software can clear this bit by reading the UART_STAT0 and UART_DATA registers one by one. 0: The UART module does not detect an IDLE frame. 1: The UART module has detected an IDLE frame. Note: When the MEN bit in UART_CTL0 register is set to 1, this bit is always 1 while the RWU bit is 0. And when the MEN bit is set to 1, this bit is always 0 while the RWU bit is 1.
3	ORERR	Overrun error This bit is set if the RBNE is not cleared and a new data frame is received through the receive shift register. An interrupt occurs if the ERRIE bit in UART_CTL2 is set. Software can clear this bit by reading the UART_STAT0 and UART_DATA registers one by one. 0: The UART does not detect an overrun error. 1: The UART has detected an overrun error.
2	NERR	Noise error flag When the OSB bit in UART_CTL2 is reset, this bit is set if the UART detects noise on the RX pin when receiving a frame. An interrupt occurs if the ERRIE bit in UART_CTL2 is set. Software can clear this bit by reading the UART_STAT0 and UART_DATA registers one by one. 0: The UART does not detect a noise error. 1: The UART has detected a noise error.
1	FERR	Frame error flag This bit is set when the RX pin is detected low during the stop bits of a receive frame. An interrupt occurs if the ERRIE bit in UART_CTL2 is set. Software can clear this bit by reading the UART_STAT0 and UART_DATA registers one by one. 0: The UART does not detect a framing error. 1: The UART has detected a framing error.

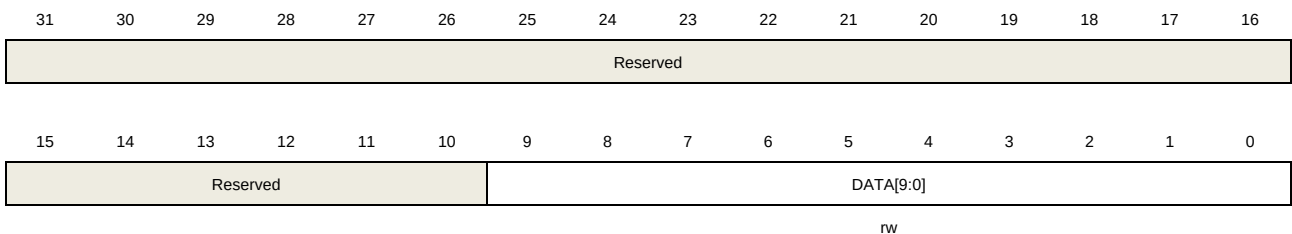
0	PERR	Parity error flag
		This bit is set when the parity bit of a receive frame does not match the expected parity value. An interrupt occurs if the PERRIE bit in UART_CTL0 is set.
		Software can clear this bit the sequence: read the UART_STAT0 register, and then read or write the UART_DATA register.
		0: The UART does not detect a parity error.
		1: The UART has detected a parity error.

25.4.2 Data register (UART_DATA)

Offset: 0x04

Reset value: Undefined

This register has to be accessed by word (32-bit).



Bits	Fields	Descriptions
31:10	Reserved	Must be kept at reset value.
9:0	DATA[9:0]	Transmit or read data value Software can write these bits to update the transmit data or read these bits to get the receive data. If the parity check function is enabled, when transmit data is written to this register, the MSB bit (bit 7, 8, 9, or bit 10 depending on the WL0 and WL1 bit in UART_CTL0) will be replaced by the parity bit.

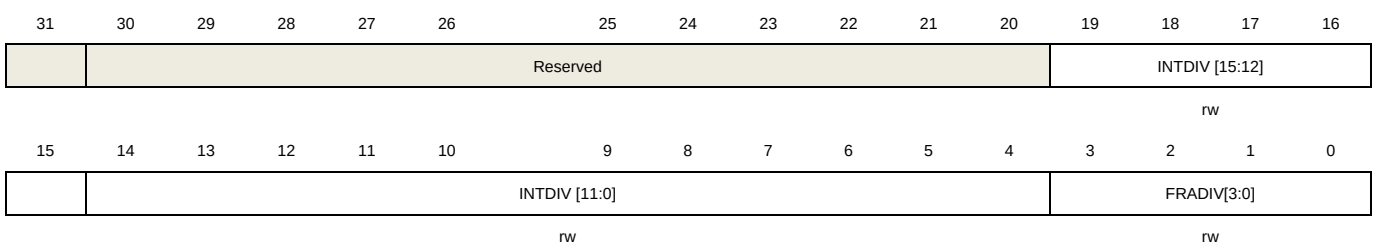
25.4.3 Baud rate register (UART_BAUD)

Address offset: 0x08

Reset value: 0x0000 0000

The software must not write this register when the UART is enabled (UEN=1).

This register has to be accessed by word (32-bit).



Bits	Fields	Descriptions
31:20	Reserved	Must be kept at reset value.
19:4	INTDIV[15:0]	Integer part of baud-rate divider. Note: In IrDA low-power mode, the INTDIV needs to be configured as 16 times the PSC.
3:0	FRADIV[3:0]	Fraction part of baud-rate divider. Software must keep FRADIV[3] bit reset, if the oversample 8 mode is enabled.

25.4.4 Control register 0 (UART_CTL0)

Address offset: 0x0C

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				WL1	Reserved										
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OVSMOD	UEN	MEN	WL0	WM	PCEN	PM	PERRIE	TBEIE	TCIE	RBNEIE	IDLEIE	TEN	REN	RWU	SBKCMD
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw1	rw

Bits	Fields	Descriptions
31:29	Reserved	Must be kept at reset value.
28	WL1	Word length bit 1 This bit must be used in conjunction with bit 12 (WL0) to determine the word length. It is set or cleared by software. 00: 8 Data bits 01: 9 Data bits 10: 7 Data bits 11:10 Data bits Note: This bit field cannot be written when the UART is enabled (UEN=1).
27:16	Reserved	Must be kept at reset value.
15	OVSMOD	Oversample mode 0: oversampling by 16 1: oversampling by 8 If IREN=1 or LMEN=1, OVSMOD is forced to '0 by hardware. Note: This bit field cannot be written when the UART is enabled (UEN=1).
14	UEN	UART enable 0: Disable UART.

		1: Enable UART.
13	MEN	Mute mode enable 0: Mute mode disabled 1: Mute mode enabled Note: 1. This bit field cannot be written when the UART is enabled (UEN=1). 2. When WM is 0 (wake from mute mode by detecting bus idle), entering mute mode is not allowed when the bus is idle.
12	WL0	Word length bit 0 Refer to WL1 in this register Note: This bit field cannot be written when the UART is enabled (UEN=1).
11	WM	Wakeup method in mute mode 0: wake up by idle frame. 1: wake up by address match. Note: This bit field cannot be written when the UART is enabled (UEN=1).
10	PCEN	Parity check function enable. 0: Disable parity check function. 1: Enable parity check function. Note: This bit field cannot be written when the UART is enabled (UEN=1).
9	PM	Parity mode 0: Even parity 1: Odd parity Note: This bit field cannot be written when the UART is enabled (UEN=1).
8	PERRIE	Parity error interrupt enable. If this bit is set, an interrupt occurs when the PERR bit in UART_STAT0 is set. 0: Disable parity error interrupt. 1: Enable parity error interrupt.
7	TBEIE	Transmitter buffer empty interrupt enable. If this bit is set, an interrupt occurs when the TBE bit in UART_STAT0 is set. 0: Disable transmitter buffer empty interrupt. 1: Enable transmitter buffer empty interrupt.
6	TCIE	Transmission complete interrupt enable. If this bit is set, an interrupt occurs when the TC bit in UART_STAT0 is set. 0: Disable transmission complete interrupt. 1: Enable transmission complete interrupt.
5	RBNEIE	Read data buffer not empty interrupt and overrun error interrupt enable. If this bit is set, an interrupt occurs when the RBNE bit or the ORERR bit in UART_STAT0 are set. 0: Disable read data register not empty interrupt and overrun error interrupt.

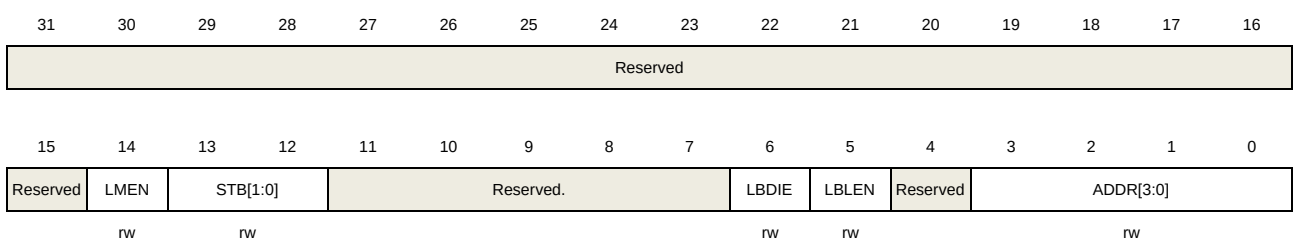
		1: Enable read data register not empty interrupt and overrun error interrupt.
4	IDLEIE	IDLE line detected interrupt enable. If this bit is set, an interrupt occurs when the IDLEF bit in UART_STAT0 is set. 0: Disable IDLE line detected interrupt. 1: Enable IDLE line detected interrupt.
3	TEN	Transmitter enable 0: Disable transmitter 1: Enable transmitter
2	REN	Receiver enable 0: Disable receiver 1: Enable receiver
1	RWU	Receiver wakeup from mute mode. This bit is used to indicate if the UART is in mute mode. 0: Receiver in active mode. 1: Receiver in mute mode. It is cleared / set by hardware when a wakeup / mute sequence (address or IDLEIE) is recognized, which is selected by the WM(UART_CTL0[11]) bit in the UART_CTL0 register. If the WM bit is set to 0 (wake up by idle frame), UART is not allowed to enter the mute mode by setting the RWU bit when the bus is in idle state. Note: This bit can be wirten to 1 when the MEN bit is 1 and the UEN bit is 1. Cleared by writing 0 to MEN bit or UEN bit in UART_CTL0 register.
0	SBKCMD	Send break command Software can set this be to send a break frame. Hardware resets this bit automatically when the break frame has been transmitted. 0: Do not transmit a break frame 1: Transmit a break frame

25.4.5 Control register 1 (UART_CTL1)

Address offset: 0x10

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



Bits	Fields	Descriptions
31:15	Reserved	Must be kept at reset value.
14	LMEN	LIN mode enable 0: Disable LIN mode 1: Enable LIN mode Note: This bit field cannot be written when the UART is enabled (UEN=1).
13:12	STB[1:0]	STOP bits length 00: 1 Stop bit 01: reserved 10: 2 Stop bits 11: reserved Note: This bit field cannot be written when the UART is enabled (UEN=1).
11:7	Reserved	Must be kept at reset value.
6	LBDIE	LIN break detected interrupt enable. If this bit is set, an interrupt occurs when the LBDF bit in UART_STAT0 is set. 0: Disable LIN break detected interrupt. 1: Enable LIN break detected interrupt.
5	LBLEN	LIN break frame length This bit specifies the length of a LIN break frame. 0: 10 bits 1: 11 bits Note: This bit field cannot be written when the UART is enabled (UEN=1).
4	Reserved	Must be kept at reset value.
3:0	ADDR[3:0]	Address of the UART In wake up by address match mode (WM=1), the UART enters mute mode when the LSB 4 bits of a received frame do not equal the ADDR[3:0] bits, and wakes up when the LSB 4 bits of a received frame equal the ADDR[3:0] bits. Note: This bit field cannot be written when the UART is enabled (UEN=1).

25.4.6 Control register 2 (UART_CTL2)

Address offset: 0x14

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MSBF	DINV	TINV	RINV	OSB	Reserved			DENT	DENR	Reserved		HDEN	IRLP	IREN	ERRIE

768

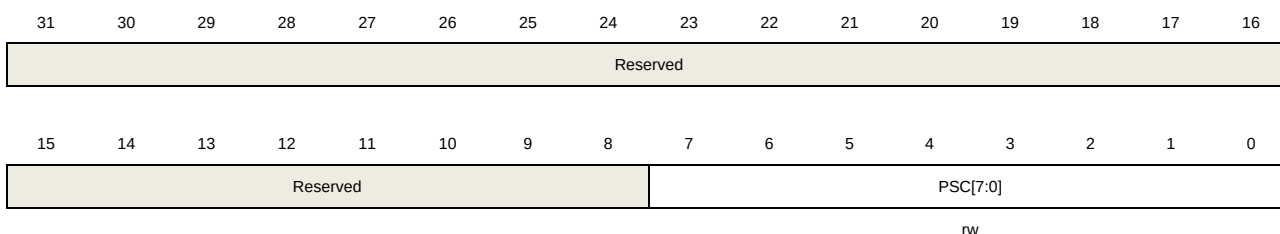
5:4	Reserved	Must be kept at reset value.
3	HDEN	Half-duplex enable This bit enables the half-duplex UART mode. 0: Disable Half duplex mode 1: Enable half duplex mode Note: This bit field cannot be written when the UART is enabled (UEN=1).
2	IRLP	IrDA low-power This bit selects low-power mode of IrDA mode. 0: Normal mode 1: Low-power mode Note: This bit field cannot be written when the UART is enabled (UEN=1).
1	IREN	IrDA mode enable This bit enables the IrDA mode of UART. 0: Disable IrDA 1: Enable IrDA Note: This bit field cannot be written when the UART is enabled (UEN=1).
0	ERRIE	Error interrupt enable When DMA request for reception is enabled (DENR=1), if this bit is set, an interrupt occurs when any one of the FERR, ORERR and NERR bits in UART_STAT0 is set. 0: Disable error interrupt. 1: Enable error interrupt.

25.4.7 Guard time and prescaler register (UART_GP)

Address offset: 0x18

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7:0	PSC[7:0]	When the UART IrDA low-power mode is enabled, these bits specify the division factor that is used to divide the peripheral clock PCLK1 to generate the low-power frequency. 00000000: Reserved - never program this value

00000001: divides by 1

00000010: divides by 2

...

11111111: divides by 255

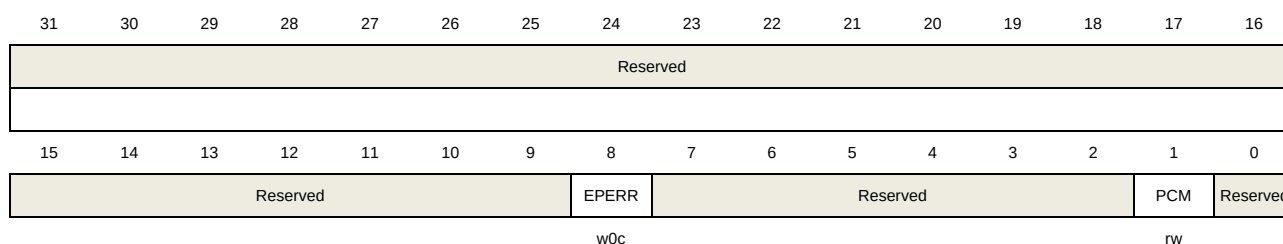
When the UART works in IrDA normal mode, these bits must be set to 00000001.

25.4.8 Coherence control register (UART_CHC)

Address offset: 0xC0

Reset value: 0x0000 0000

This register has to be accessed by word(32-bit).



Bits	Fields	Descriptions
31:9	Reserved	Must be kept at reset value.
8	EPERR	Early parity error flag. This flag will be set as soon as the parity bit has been detected, which is before RBNE flag. This flag is cleared by writing 0. 0: No parity error is detected 1: Parity error is detected
7:2	Reserved	Must be kept at reset value.
1	PCM	Parity check coherence mode. 0: In case of wakeup by an address match: the MSB bit of the data is taken into account to identify an address but not the parity bit. And the receiver does not check the parity of the address data (PERR is not set in case of a parity error). 1: In case of wakeup by an address match, the receiver check the parity of the address data and PERR is set in case of a parity error.
0	Reserved	Must be kept at reset value.

26 Inter-integrated circuit interface (I2C)

26.1 Overview

The I2C (inter-integrated circuit) module provides an I2C interface which is an industry standard two-line serial interface for MCU to communicate with external I2C interface. I2C bus uses two serial lines: a serial data line, SDA, and a serial clock line, SCL.

The I2C interface implements standard I2C protocol with standard mode, fast mode and fast mode plus as well as CRC calculation and checking, SMBus (system management bus), and PMBus (power management bus). It also supports multi-master I2C bus. The I2C interface provides DMA mode for users to reduce CPU overload.

26.2 Characteristics

- Parallel-bus to I2C-bus protocol converter and interface.
- Both master and slave functions with the same interface.
- Bi-directional data transfer between master and slave.
- Supports 7-bit and 10-bit addressing and general call addressing.
- Multiple 7-bit slave addresses (2 address, configurable mask).
- Programmable setup time and hold time.
- Multi-master capability.
- Supports standard mode (up to 100 KHz) and fast mode (up to 400 KHz) and fast mode plus (up to 1MHz, I2CFMPEN (fast mode plus) must be enabled in SYSCFG_CFG1 of SYSCFG module).
- Configurable SCL stretching in slave mode.
- Supports DMA mode.
- SMBus 3.0 and PMBus 1.3 compatible.
- Configure the SMBEN bit in I2C_CTL0 to control whether I2C works in SMBus mode.
- Optional PEC (packet error checking) generation and check.
- Programmable digital noise filters.
- Wakeup from Sleep mode and Deep-sleep mode on I2C address match.
- Independent clock from PCLK.

26.3 Function overview

[Figure 26-1. I2C module block diagram](#) below provides details on the internal configuration of the I2C interface.

Figure 26-1. I2C module block diagram

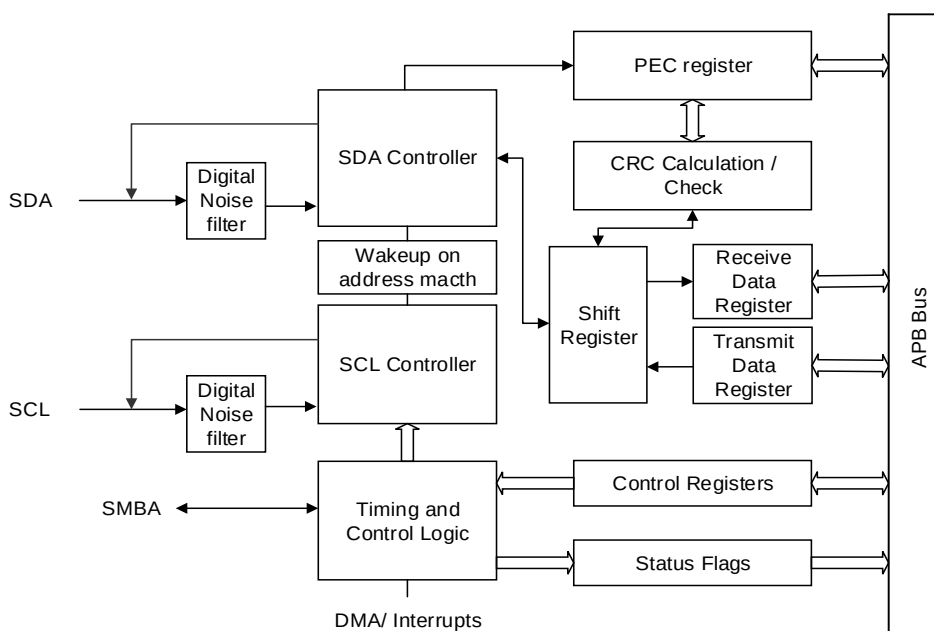


Table 26-1. Definition of I2C-bus terminology (refer to the I2C specification of Philips semiconductors)

Term	Description
Transmitter	the device which sends data to the bus
Receiver	the device which receives data from the bus
Master	the device which initiates a transfer, generates clock signals and terminates a transfer
Slave	the device addressed by a master
Multi-master	more than one master can attempt to control the bus at the same time without corrupting the message
Arbitration	procedure to ensure that, if more than one master tries to control the bus simultaneously, only one is allowed to do so and the winning master's message is not corrupted

26.3.1 Clock requirements

The I2C clock is independent of the PCLK frequency, so that the I2C can be operated independently.

This I2C clock (I2CCLK) can be selected from the following four clock sources:

- PCLK1: APB1 clock (default value)
- PLL: Phase Lock Loop
- IRC32M: Internal 32M RC oscillator
- HXTAL(1-24M)

The I2CCLK period t_{I2CCLK} must match the conditions as follows:

- $t_{I2CCLK} < (t_{LOW} - t_{filters}) / 4$
- $t_{I2CCLK} < t_{HIGH}$

with:

t_{LOW} : SCL low time

t_{HIGH} : SCL high time

$t_{filters}$: When the filters are enabled, represent the delays by the digital filter. Digital filter delay is $DNF[3:0] \times t_{I2CCLK}$.

The period of PCLK clock t_{PCLK} match the conditions as follows:

- $t_{PCLK} < 4/3 \times t_{SCL}$

with:

t_{SCL} : the period of SCL

Note: When the I2C kernel is provided by PCLK, this clock must match the conditions for t_{I2CCLK} .

26.3.2 I2C communication flow

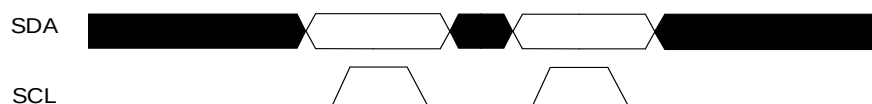
An I2C device is able to transmit or receive data whether it's a master or a slave, thus, there're 4 operation modes for an I2C device:

- Slave transmitter
- Slave receiver
- Master transmitter
- Master receiver

Data validation

The data on the SDA line must be stable during the HIGH period of the clock. The HIGH or LOW state of the data line can only change when the clock signal on the SCL line is LOW (see [Figure 26-2. Data validation](#)). One clock pulse is generated for each data bit transferred.

Figure 26-2. Data validation

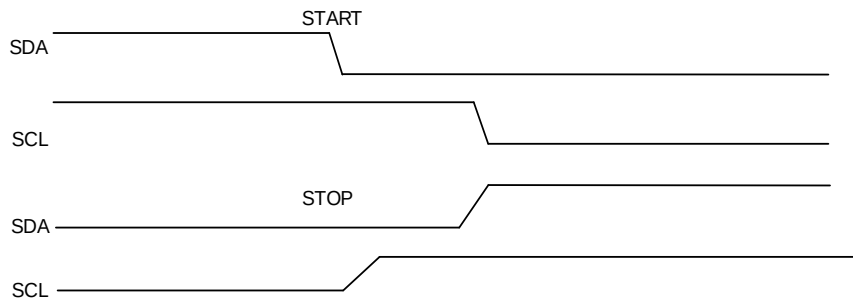


START and STOP signal

All transactions begin with a START and are terminated by a STOP (see [Figure 26-3. START and STOP signal](#)). A HIGH to LOW transition on the SDA line while SCL is HIGH defines a START signal. A LOW to HIGH transition on the SDA line while SCL is HIGH defines a STOP

signal.

Figure 26-3. START and STOP signal



Each I2C device is recognized by a unique address (whether it is a microcontroller, LCD driver, memory or keyboard interface) and can operate as either a transmitter or receiver, depending on the function of the device. It operates in slave mode by default. When it generates a START signal, the interface automatically switches from slave to master. If an arbitration loss or a STOP generation occurs, then the interface switches from master to slave, allowing multimaster capability.

An I2C slave will continue to detect addresses after a START signal on I2C bus and compare the detected address with its slave address which is programmable by software. Once the two addresses match, the I2C slave will send an ACK to the I2C bus and responses to the following command on I2C bus: transmitting or receiving the desired data. Additionally, if General Call is enabled by software, the I2C slave always responses to a General Call Address (0x00). The I2C block support both 7-bit and 10-bit address modes.

Data and addresses are transferred as 8-bit bytes, MSB first. The first byte(s) following the START signal contain the address (one in 7-bit mode, two in 10-bit mode). The address is always transmitted in master mode.

A 9th clock pulse follows the 8 clock cycles of byte transmission, during which the receiver must send an acknowledge bit to the transmitter. Acknowledge can be enabled or disabled by software.

An I2C master always initiates or end a transfer using START or STOP signal and it's also responsible for SCL clock generation.

In master mode, if AUTOEND=1, the STOP signal is generated automatically by hardware. If AUTOEND=0, the STOP signal generated by software, or the master can generate a RESTART signal to start a new transfer.

Figure 26-4. I2C communication flow with 10-bit address (Master Transmit)

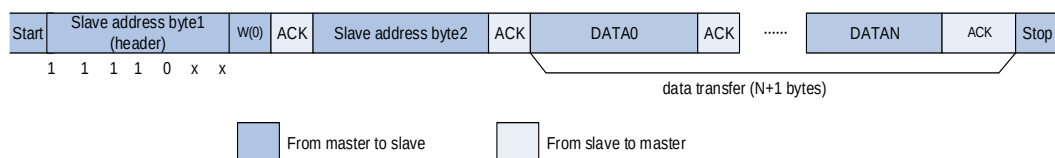


Figure 26-5. I2C communication flow with 7-bit address (Master Transmit)

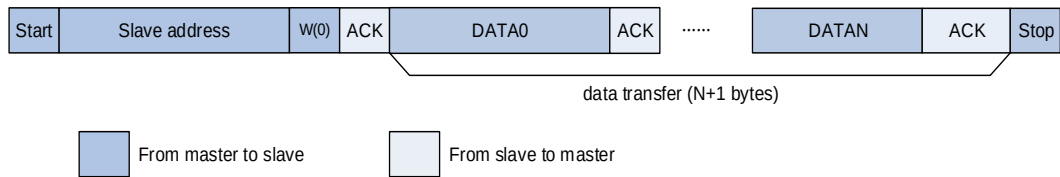
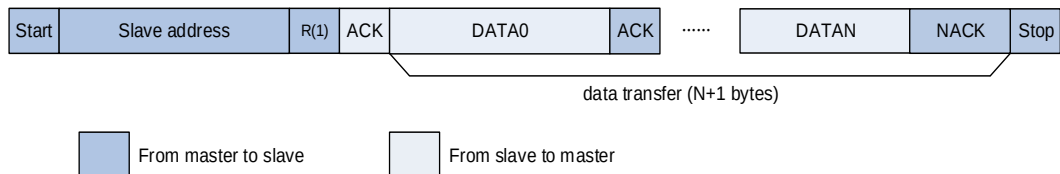


Figure 26-6. I2C communication flow with 7-bit address (Master Receive)



In 10-bit addressing mode, the HEAD10R bit can be configured to decide whether the complete address sequence must be executed, or only the header to be sent. When HEAD10R=0, the complete 10-bit address read sequence must be executed with START + header of 10-bit address in write direction + slave address byte 2 + RESTART + header of 10-bit address in read direction, as is shown in [Figure 26-7. I2C communication flow with 10-bit address \(Master Receive when HEAD10R=0\)](#).

In 10-bit addressing mode, if the master reception follows a master transmission between the same master and slave, the address read sequence can be RESTART + header of 10-bit address in read direction, as is shown in [Figure 26-8. I2C communication flow with 10-bit address \(Master Receive when HEAD10R=1\)](#).

Figure 26-7. I2C communication flow with 10-bit address (Master Receive when HEAD10R=0)

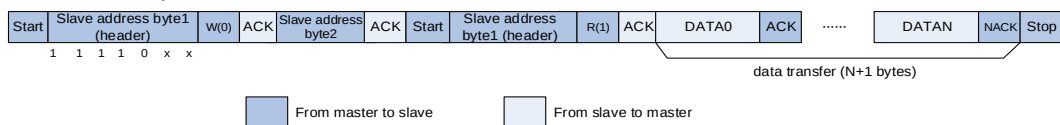
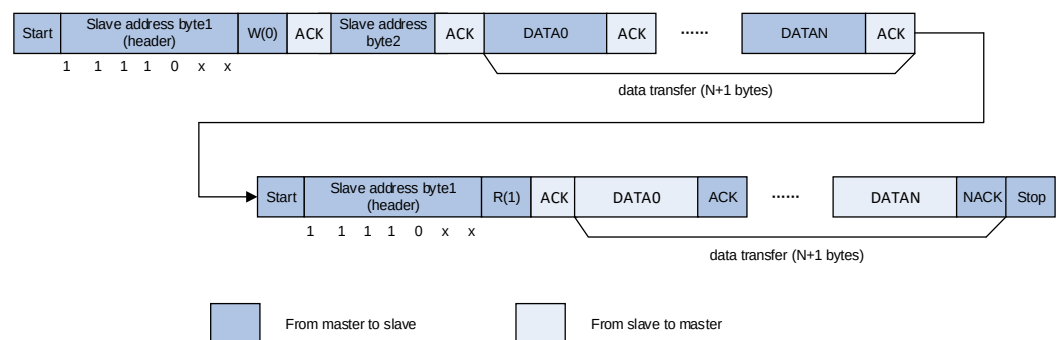


Figure 26-8. I2C communication flow with 10-bit address (Master Receive when HEAD10R=1)



26.3.3 Digital Noise filter

Digital noise filter is integrated in I2C peripherals, and it can be configured before the I2C peripheral is enabled according to the actual requirements.

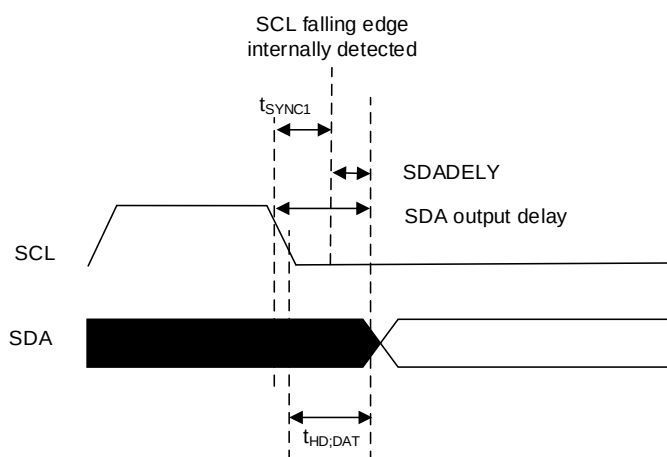
The digital noise filter can be used by configuring the DNF[3:0] bit in I2C_CTL0 register. The level of the SCL or the SDA will not be changed if the level is stable for no more than $DNF[3:0] \times t_{I2CCLK}$. The length of spikes to be suppressed is configured by DNF[3:0].

26.3.4 I2C timings configuration

The PSC[3:0], SCLDELY[3:0] and SDADELY[3:0] bits in the I2C_TIMING register must be configured in order to guarantee a correct data hold and setup time used in I2C communication.

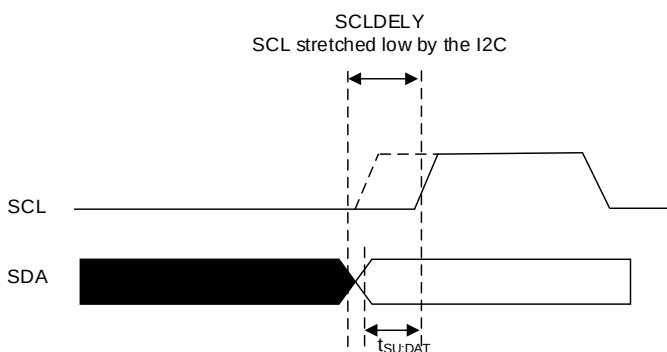
If the data is already available in I2C_TDATA register, the data will be sent on SDA after the SDADELY delay. As is shown in [Figure 26-9. Data hold time](#).

Figure 26-9. Data hold time



The SCLDELY counter starts when the data is sent on SDA output. As is shown in [Figure 26-10. Data setup time](#).

Figure 26-10. Data setup time



When the SCL falling edge is internally detected, a delay is inserted before sending SDA output. This delay is $t_{SDA\text{DELY}} = SDA\text{DELY} * t_{PSC} + t_{I2CCLK}$ where $t_{PSC} = (PSC+1) * t_{I2CCLK}$. $t_{SDA\text{DELY}}$ effects $t_{HD;DAT}$. The total delay of SDA output is $t_{SYNC1} + \{[SDA\text{DELY} * (PSC+1) + 1] * t_{I2CCLK}\}$. t_{SYNC1} depends on SCL falling slope, the delay of digital filter and delay of SCL synchronization to I2CCLK clock. The delay of SCL synchronization to I2CCLK clock is 2 to 3 t_{I2CCLK} .

Note: When $SDA\text{DELY} = 0$, $t_{SDA\text{DELY}} = t_{PSC}$.

$SDA\text{DELY}$ must match condition as follows:

- $SDA\text{DELY} \geq \{t_r(\text{max}) + t_{HD;DAT}(\text{min}) - [(DNF+3) * t_{I2CCLK}]\} / [(PSC+1) * t_{I2CCLK}]$
- $SDA\text{DELY} \leq \{t_{HD;DAT}(\text{max}) - [(DNF+4) * t_{I2CCLK}]\} / [(PSC+1) * t_{I2CCLK}]$

Note: The $t_{HD;DAT}$ should be less than the maximum of $t_{VD;DAT}$.

When $SS = 0$, after $t_{SDA\text{DELY}}$ delay, the slave had to stretch the clock before the data writing to I2C_TDATA register, SCL is low during the data setup time. The setup time is $t_{SCL\text{DELY}} = (SCL\text{DELY}+1) * t_{PSC}$. $t_{SCL\text{DELY}}$ effects $t_{SU;DAT}$.

$SCL\text{DELY}$ must match condition as follows:

- $SCL\text{DELY} \geq \{[t_r(\text{max}) + t_{SU;DAT}(\text{min})] / [(PSC+1) * t_{I2CCLK}]\} - 1$

In master mode, the SCL clock high and low levels must be configured by programming the $PSC[3:0]$, $SCLH[7:0]$ and $SCLL[7:0]$ bits in the I2C_TIMING register.

When the SCL falling edge is internally detected, a delay is inserted before releasing the SCL output. This delay is $t_{SCLL} = (SCLL+1) * t_{PSC}$ where $t_{PSC} = (PSC+1) * t_{I2CCLK}$. t_{SCLL} impacts the SCL low time t_{LOW} .

When the SCL rising edge is internally detected, a delay is inserted before forcing the SCL output to low level. This delay is $t_{SCLH} = (SCLH+1) * t_{PSC}$ where $t_{PSC} = (PSC+1) * t_{I2CCLK}$. t_{SCLH} impacts the SCL high time t_{HIGH} .

Note: When the I2C is enabled, the timing configuration and SS mode must not be changed.

Table 26-2. Data setup time and data hold time

Symbol	Parameter	Standard mode		Fast mode		Fast mode plus		SMBus		Unit
		Min	Max	Min	Max	Min	Max	Min	Max	
$t_{HD;DAT}$	Data hold time	0	-	0	-	0	-	0.3	-	us
$t_{VD;DAT}$	Data valid time	-	3.45	-	0.9	-	0.45	-	-	
$t_{SU;DAT}$	Data setup time	250	-	100	-	50	-	250	-	ns
t_r	Rising time of SCL and SDA	-	1000	-	300	-	120	-	1000	
t_f	falling time of SCL and SDA	-	300	-	300	-	120	-	300	

26.3.5 I2C reset

A software reset can be performed by clearing the I2CEN bit in the I2C_CTL0 register. When a software reset is generated, the SCL and SDA are released. The communication control bits and status bits come back to the reset value. Software reset have no effect on configuration registers. The impacted register bits are START, STOP, NACKEN in I2C_CTL1 register, I2CBSY, TBE, TI, RBNE, ADDSEND, NACK, TCR, TC, STPDET, BERR, LOSTARB and OUERR in I2C_STAT register. Additionally, when the SMBus is supported, PECTRANS in I2C_CTL1 register, PECERR, TIMEOUT and SMBALT in I2C_STAT are also impacted.

In order to perform the software reset, I2CEN must be kept low during at least 3 APB clock cycles. This is ensured by writing software sequence as follows:

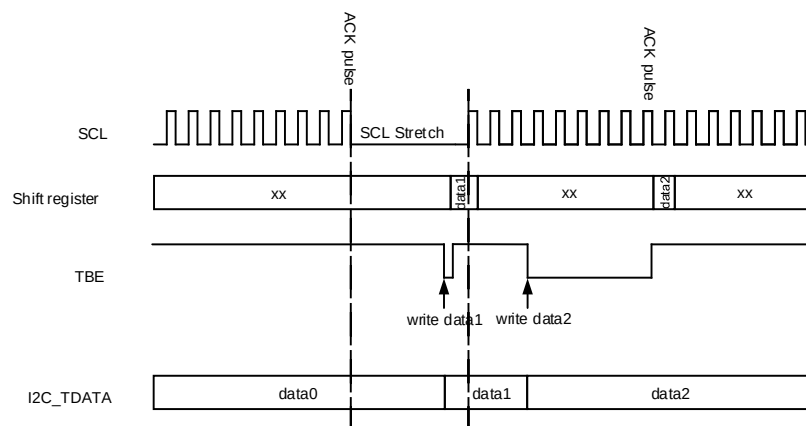
- Write I2CEN = 0
- Check I2CEN = 0
- Write I2CEN = 1

26.3.6 Data transfer

Data Transmission

When transmitting data, if TBE is 0, it indicates that the I2C_TDATA register is not empty, the data in I2C_TDATA register is moved to the shift register after the 9th SCL pulse. Then the data will be transmitted through the SDA line from the shift register. If TBE is 1, it indicates that the I2C_TDATA register is empty, the SCL line is stretched low until I2C_TDATA is not empty. The stretch begins after the 9th SCL pulse.

Figure 26-11. Data transmission

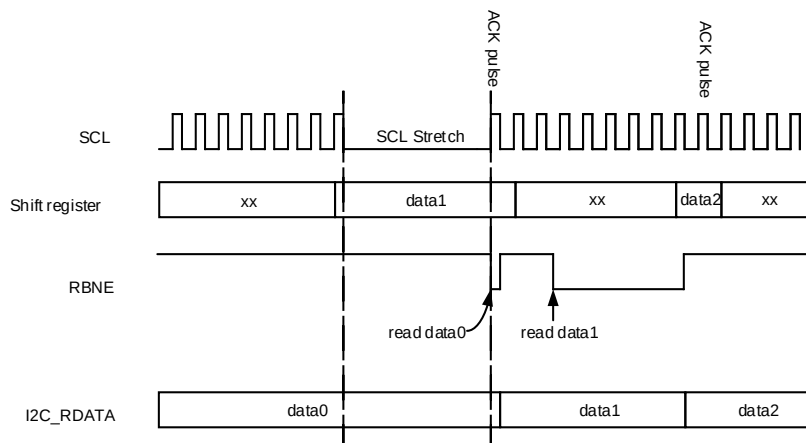


Data Reception

When receiving data, the data will be received in the shift register first. If RBNE is 0, the data in the shift register will move into I2C_RDATA register. If RBNE is 1, the SCL line will be

stretched until the previous received data in I2C_RDATA is read. The stretch is inserted before the acknowledge pulse.

Figure 26-12. Data reception



Reload and automatic end mode

In order to manage byte transfer and to shut down the communication in modes as is shown in [Table 26-3. Communication modes to be shut down](#), the I2C embedded a byte counter in the hardware.

Table 26-3. Communication modes to be shut down

Working mode	Action
Master mode	NACK, STOP and RESTART generation
Slave receiver mode	ACK control
SMBus mode	PEC generation/checking

The number of bytes to be transferred is configured by BYTENUM[7:0] in I2C_CTL1 register. If BYTENUM is greater than 255, or in slave byte control mode, the reload mode must be enabled by setting the RELOAD bit in I2C_CTL1 register. In reload mode, when BYTENUM counts to 0, the TCR bit will be set, and an interrupt will be generated if TCIE is set. Once the TCR flag is set, SCL is stretched. The TCR bit is cleared by writing a non-zero number in BYTENUM.

Note: The reload mode must be disabled after the last reloading of BYTENUM[7:0].

The reload mode must be disabled when the automatic end mode is enabled. In automatic end mode, the master will send a STOP signal automatically when the BYTENUM[7:0] counts to 0.

26.3.7 I2C slave mode

Initialization

When works in slave mode, at least one slave address should be enabled. Slave address 1 can be programmed in I2C_SADDR0 register and slave address 2 can be programmed in I2C_SADDR1 register. ADDRESSEN in I2C_SADDR0 register and ADDRESS2EN in I2C_SADDR1 register should be set when the corresponding address is used. 7-bit address or 10-bit address can be programmed in ADDRESS[9:0] in I2C_SADDR0 register by configuring the ADDFORMAT bit in 7-bit address or 10-bit address.

The ADDM[6:0] in I2C_CTL2 register defines which bits of ADDRESS[7:1] are compared with an incoming address byte, and which bits are ignored.

The ADDMSK2[2:0] is used to mask ADDRESS2[7:1] in I2C_SADDR1 register. For details, refer to the description of ADDMSK2[2:0] in I2C_SADDR1 register.

When the I2C received address matches one of its enabled addresses, the ADDSEND will be set, and an interrupt is generated if the ADDMIE bit is set. The READDR[6:0] bits in I2C_STAT register will store the received address. And TR bit in I2C_STAT register updates after the ADDSEND is set. The bit will let the slave to know whether to act as a transmitter or receiver.

SCL line stretching

The clock stretching is used in slave mode by default (SS=0), the SCL line can be stretched low if necessary. The SCL will be stretched in following cases.

- The SCL is stretched when the ADDSEND bit is set, and released when the ADDSEND bit is cleared.
- In slave transmitting mode, after the ADDSEND bit is cleared, the SCL will be stretched before the first data byte writing to the I2C_TDATA register. Or the SCL will be stretched before the new data is written to the I2C_TDATA register after the previous data transmission is completed.
- In slave receiving mode, a new reception is completed but the data in I2C_RDATA register has not been read.
- When SBCTL=1 and RELOAD=1, after the transfer of the last byte, TCR is set. Before the TCR is cleared, the SCL will be stretched.
- The I2C stretches SCL low during $[(SDAELY+SCLDELY+1)*(PSC+1)+1]*t_{I2CCLK}$ after detecting the SCL falling edge.

The clock stretching can be disabled by setting the SS bit in I2C_CTL0 register (SS=1). The SCL will not be stretched in following cases.

- When the ADDSEND is set, the SCL will be not stretched.
- In slave transmitting mode, before the first SCL pulse, the data must be written in the I2C_TDATA register. Or else the OUERR bit in the I2C_STAT register will be set, if the ERRIE bit is set, an interrupt will be generated. When the STPDET bit is set and the first

data transmission starts, OUERR bit in the I2C_STAT register will also be set.

- In slave receiving mode, before the 9th SCL pulse (ACK pulse) occurred by the next data byte, the data must be read out from the I2C_RDATA register. Or else the OUERR bit in the I2C_STAT register will be set, if the ERRIE bit is set, and an interrupt will be generated.

Slave byte control mode

In slave receiving mode, the slave byte control mode can be enabled by setting the SBCTL bit in the I2C_CTL0 register to allow byte ACK control. When SS=1, the slave byte control mode is not allowed.

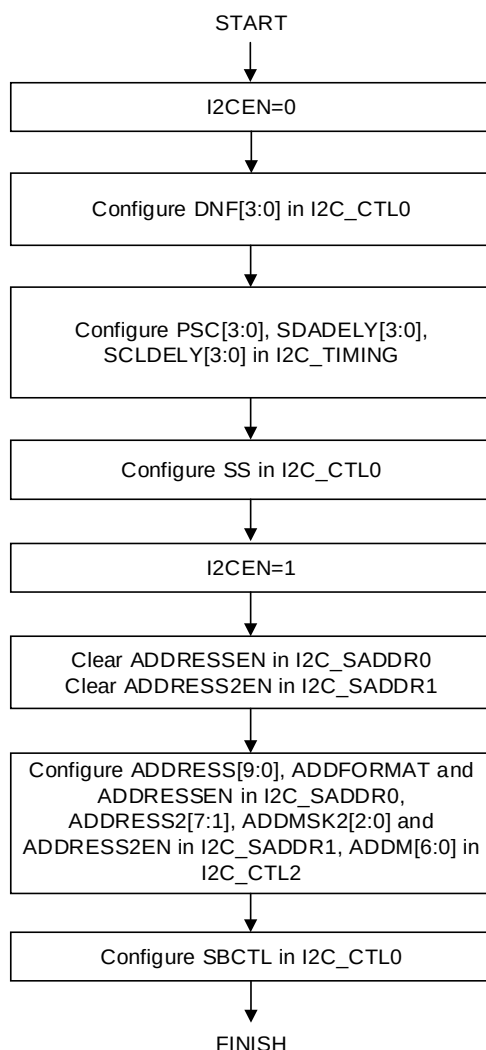
When using slave byte control mode, the reload mode must be enabled by setting the RELOAD bit in I2C_CTL1 register. In slave byte control mode, BYTENUM [7:0] in I2C_CTL1 register must be configured as 1 in the ADDSEND interrupt service routine and reloaded to 1 after each byte received. The TCR bit in I2C_STAT register will be set when a byte is received, the SCL will be stretched low by slave between the 8th and 9th clock pulses. Then the data can be read from the I2C_RDATA register, and the slave determines to send an ACK or a NACK by configuring the NACKEN bit in the I2C_CTL1 register. When the BYTENUM[7:0] is written a non-zero value, the slave will release the stretch.

When the BYTENUM [7:0] is greater than 0x1, there is no stretch between the reception of two data bytes.

Note: The SBCTL bit can be configured in following cases:

1. I2CEN=0.
2. The slave has not been addressed.
3. ADDSEND=1.

Only when the ADDSEND=1, or TCR=1, the RELOAD bit can be modified.

Figure 26-13. I2C initialization in slave mode


Programming model in slave transmitting mode

When the I2C_TDATA register is empty, the TI bit in I2C_STAT register will be set. If the TIE bit in I2C_CTL0 register is set, an interrupt will be generated. The NACK bit in I2C_STAT register will be set when a NACK is received. And an interrupt is generated if the NACKIE bit is set in the I2C_CTL0 register. The TI bit in I2C_STAT register will not be set when a NACK is received.

The STPDET bit in I2C_STAT register will be set when a STOP is received. If the STPDETIE in I2C_CTL0 register is set, an interrupt will be generated.

When SBCTL is 0, if ADDSEND=1, and the TBE bit in I2C_STAT register is 0, the data in I2C_TDATA register can be chosen to be transmitted or flushed. The data is flushed by setting the TBE bit.

When SBCTL=1, the slave works in slave byte control mode, the BYTENUM [7:0] must be configured in the ADDSEND interrupt service routine. And the number of TI events is equal to the value of BYTENUM[7:0].

When SS=1, the SCL will not be stretched when ADDSEND bit in I2C_STAT register is set. In this case, the data in I2C_TDATA register can not be flushed in ADDSEND interrupt service routine. So the first byte to be sent must be programmed in the I2C_TDATA register previously.

- This data can be the one which is written in the last TI event of the last transfer.
- Setting the TBE bit can flush the data if it is not the one to be sent, then a new byte can be written in I2C_TDATA register. The STPDET must be 0 when the data transmission begins. Or else the OUERR bit in I2C_STAT register will be set and an underrun error occurs.
- When interrupt or DMA is used in slave transmitter, if a TI event is needed, in order to generate a TI event both the TI bit and the TBE bit must be set.

Figure 26-14. Programming model for slave transmitting when SS=0

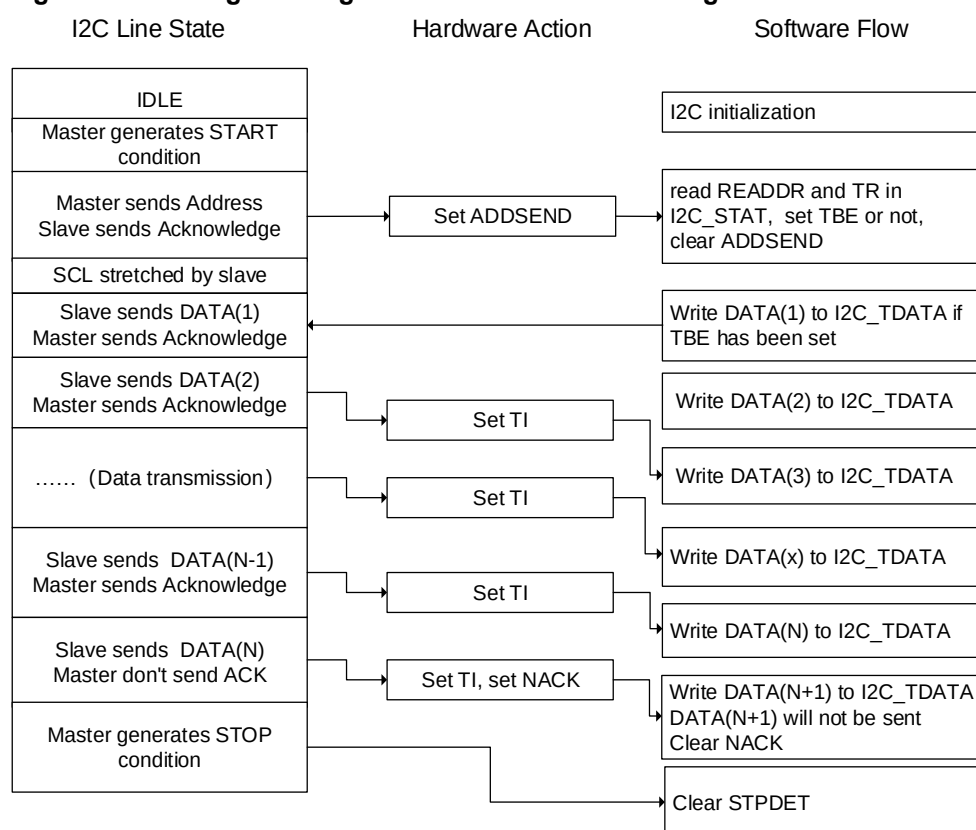
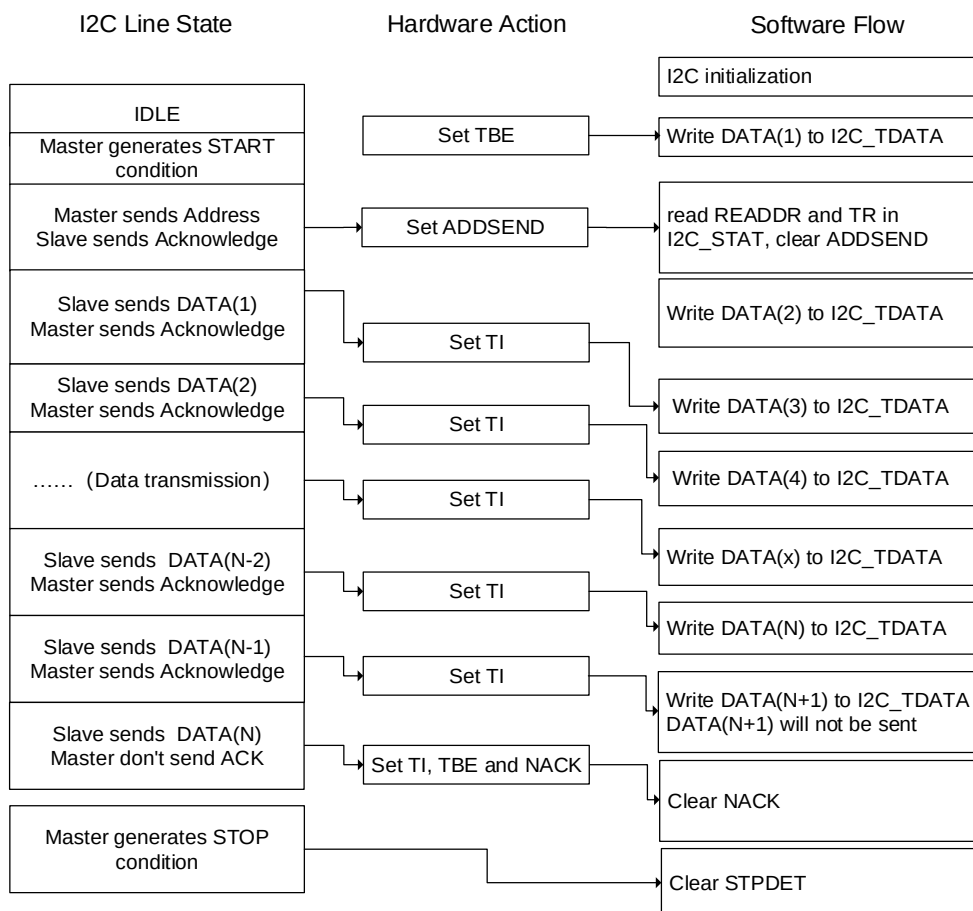


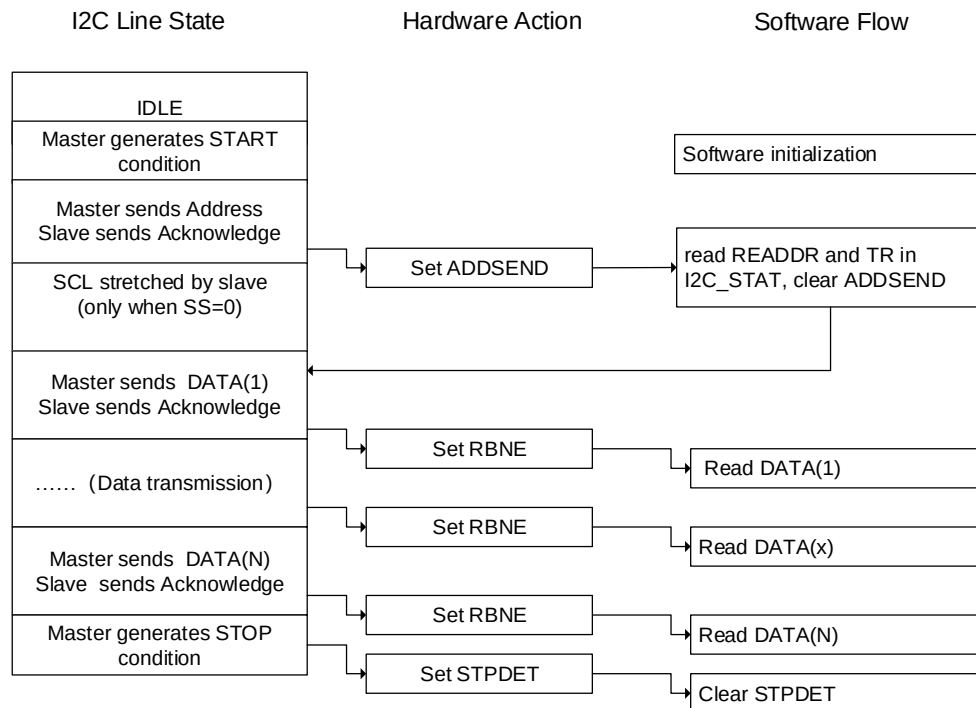
Figure 26-15. Programming model for slave transmitting when SS=1



Programming model in slave receiving mode

When the I2C_RDATA is not empty, the RBNE bit in I2C_STAT register is set, and if the RBNEIE bit in I2C_CTL0 register is set, an interrupt will be generated. When a STOP is received, STPDET will be set in I2C_STAT register. If the STPDETIE bit in I2C_CTL0 register is set, and an interrupt will be generated.

Figure 26-16. Programming model for slave receiving



26.3.8 I2C master mode

Initialization

The SCLH[7:0] and SCLL[7:0] in I2C_TIMING register should be configured when I2CEN is 0. In order to support multi-master communication and slave clock stretching, a clock synchronization mechanism is implemented.

The SCLL[7:0] and SCLH[7:0] are used for the low level counting and high level counting respectively. After a t_{SYNC1} delay, when the SCL low level is detected, the SCLL[7:0] starts counting, if the SCLL[7:0] in I2C_TIMING register is reached by SCLL[7:0] counter, the I2C will release the SCL clock. After a t_{SYNC2} delay, when the SCL high level is detected, the SCLH[7:0] starts counting, if the SCLH[7:0] in I2C_TIMING register is reached by SCLH[7:0] counter, the I2C will stretch the SCL clock.

So the master clock period is:

$$t_{\text{SCL}} = t_{\text{SYNC1}} + t_{\text{SYNC2}} + \{[(\text{SCLH}[7:0] + 1) + (\text{SCLL}[7:0] + 1)] * (\text{PSC} + 1) * t_{\text{I2CCCLK}}\}.$$

The t_{SYNC1} depends on the SCL falling slope, delay by input digital noise filter and SCL synchronization with I2CCCLK clock, which generally 2 to 3 I2CCCLK periods. The t_{SYNC2} depends on the SCL rising slope, delay by input digital noise filter and SCL synchronization with I2CCCLK clock, which generally 2 to 3 I2CCCLK periods. The delay by digital noise filter is $\text{DNF}[3:0] * t_{\text{I2CCCLK}}$.

When works in master mode, the ADD10EN bit, SADDRESS[9:0] bits, TRDIR bit should be

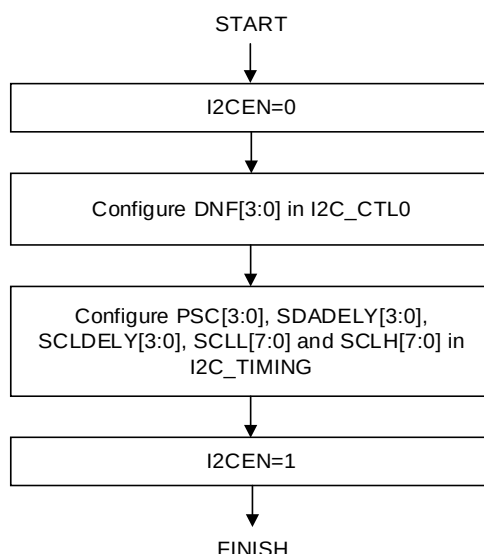
configured in I2C_CTL1 register. When the addressing mode is 10-bit in master receiving mode, the HEAD10R bit must be configured to decide whether the complete address sequence must be executed, or only the header to be sent. The number of bytes to be transferred should be configured in BYTENUM[7:0] in I2C_CTL1 register. If the number of bytes to be transferred is equal to or greater than 255, BYTENUM[7:0] should be configured as 0xFF. Then the master sends the START signal. All the bits above should be configured before the START is set. The slave address will be sent after the START signal when the I2CBSY bit I2C_STAT register is detected as 0. When the arbitration is lost, the master changes to slave mode and the START bit will be cleared by hardware. When the slave address has been sent, the START bit will be cleared by hardware.

Note: BYTENUM[7:0] can not be configured as 0.

In 10-bit addressing mode, if the master receives a NACK after the transmission of 10-bit header, the master will resend it until ACK is received. The ADDSEND bit must be set to stop sending the slave address.

If the START bit is set, meanwhile the ADDSEND is set by addressing as a slave, the master changes to slave mode. The ADDSEND bit must be set to clear the START bit.

Figure 26-17. I2C initialization in master mode



Programming model in master transmitting mode

In master transmitting mode, the TI bit is set after the ACK is received of each byte transmission. If the TIE bit in I2C_CTL0 register is set, an interrupt will be generated. The bytes to be transferred is programmed in BYTENUM[7:0] in I2C_CTL1 register. If the bytes to be transferred is greater than 255, RELOAD bit in I2C_CTL1 register must be set to enable the reload mode. In reload mode, when data of BYTENUM[7:0] bytes have been transferred, the TCR bit in I2C_STAT register will be set and the SCL stretches until BYTENUM[7:0] is modified with a non-zero value.

When a NACK is received, the TI bit will not set.

- If data of BYTENUM[7:0] bytes have been transferred and RELOAD=0, the AUTOEND bit in I2C_CTL1 can be set to generate a STOP signal automatically. When AUTOEND is 0, the TC bit in I2C_STAT register will be set and the SCL is stretched. In this case, the master can generate a STOP signal by setting the STOP bit in the I2C_CTL1 register. Or generate a RESTART signal to start a new transfer. The TC bit is cleared when the START / STOP bit is set.
- If a NACK is received, a STOP signal is automatically generated, the NACK is set in I2C_STAT register, if the NACKIE bit is set, an interrupt will be generated.

Note: When the RELOAD bit is 1, the AUTOEND has no effect.

Figure 26-18. Programming model for master transmitting (N<=255)

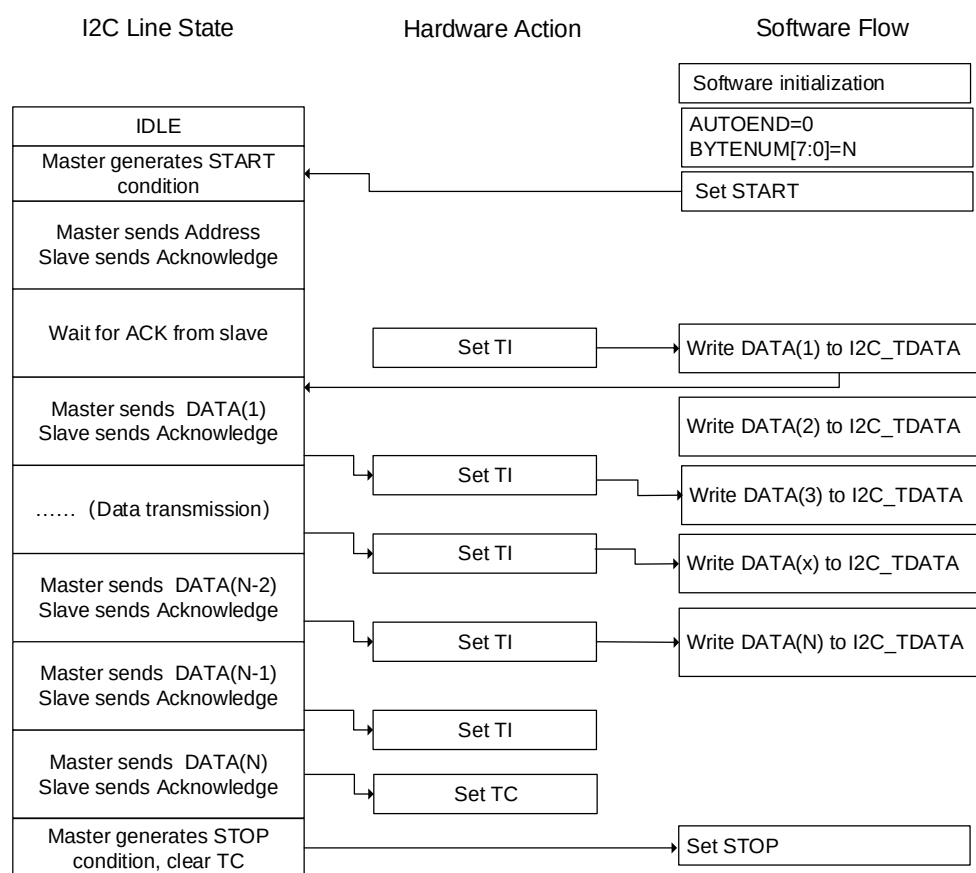
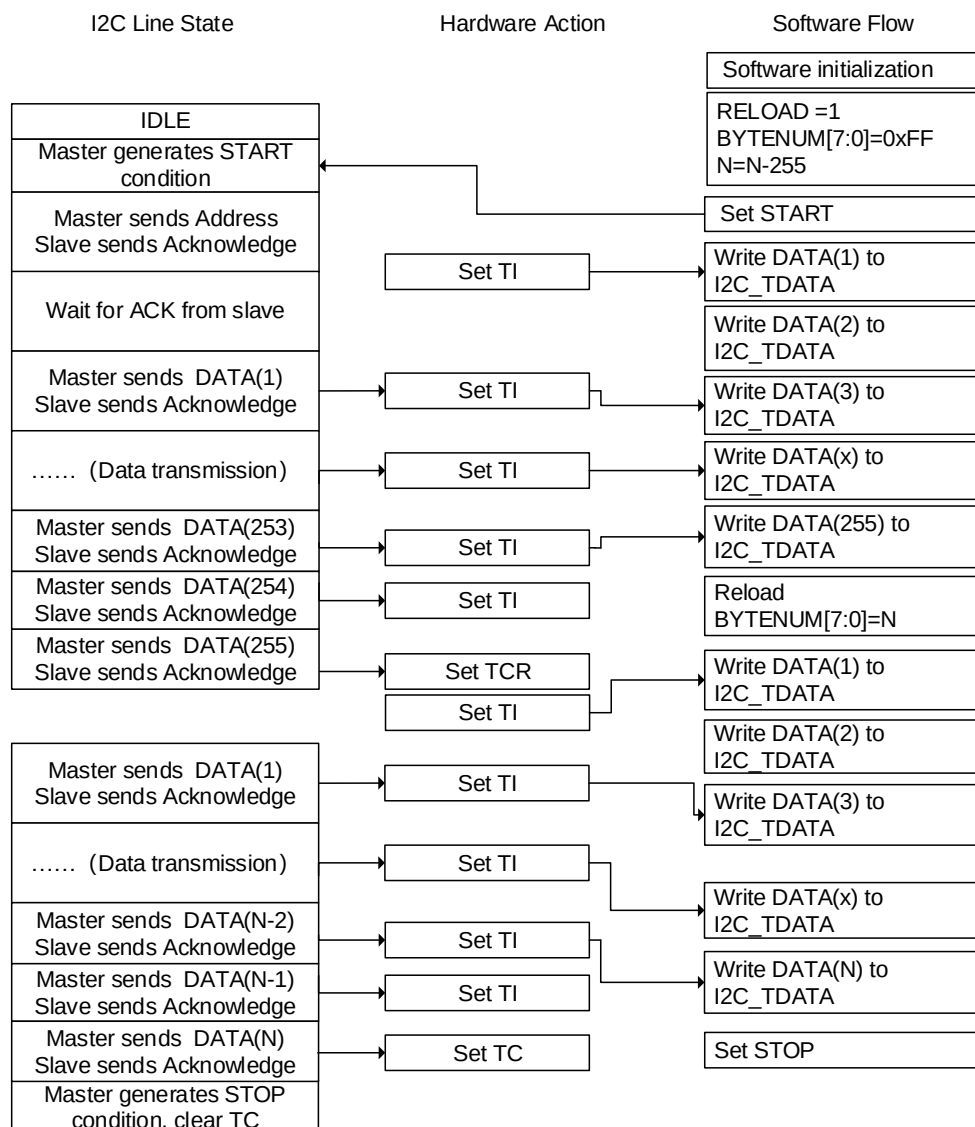


Figure 26-19. Programming model for master transmitting (N>255)


Programming model in master receiving mode

In master receiving mode, the RBNE bit in I2C_STAT register will be set when a byte is received. If the RBNEIE bit is set in I2C_CTL0 register, an interrupt will be generated. If the number of bytes to be received is greater than 255, RELOAD bit in I2C_CTL1 register must be set to enable the reload mode. In reload mode, when data of BYTENUM[7:0] bytes have been transferred, the TCR bit in I2C_STAT register will be set and the SCL stretches until BYTENUM[7:0] is modified with a non-zero value.

If data of BYTENUM[7:0] bytes have been transferred and RELOAD=0, the AUTOEND bit in I2C_CTL1 can be set to generate a STOP signal automatically. When AUTOEND is 0, the TC bit in I2C_STAT register will be set and the SCL is stretched. In this case, the master can generate a STOP signal by setting the STOP bit in the I2C_CTL1 register. Or generate a RESTART signal to start a new transfer. The TC bit is cleared when the START / STOP bit is set.

Figure 26-20. Programming model for master receiving ($N \leq 255$)

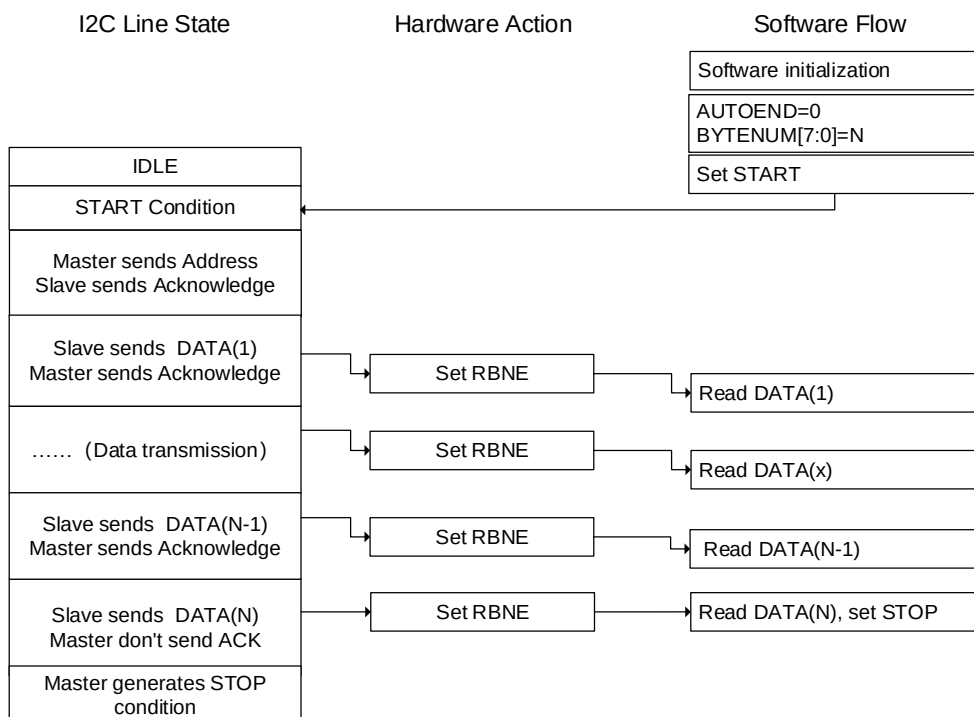
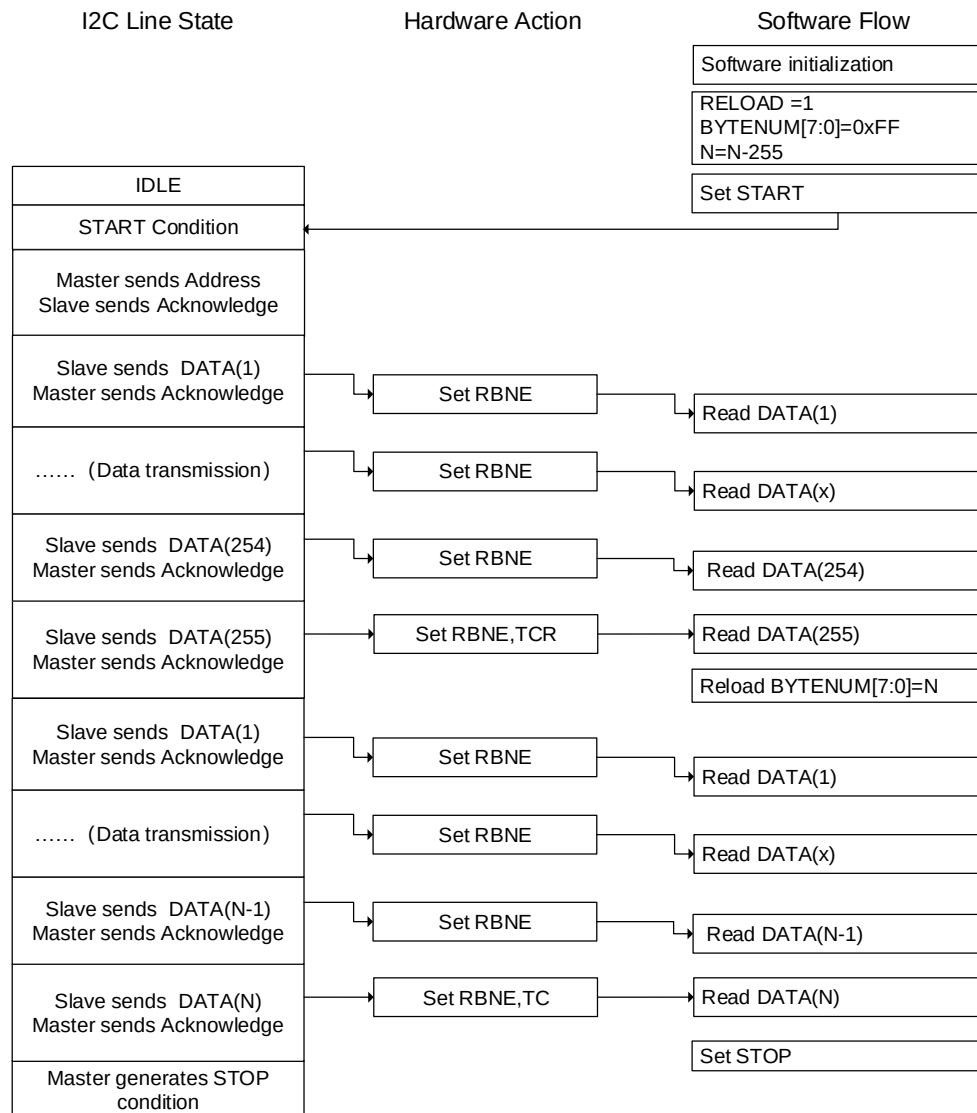


Figure 26-21. Programming model for master receiving (N>255)



26.3.9 SMBus support

The System Management Bus (abbreviated to SMBus or SMB) is a single-ended simple two-wire bus for the purpose of lightweight communication. Most commonly it is found in computer motherboards for communication with power source for ON/OFF instructions. It is derived from I2C for communication with low-bandwidth devices on a motherboard, especially power related chips such as a laptop's rechargeable battery subsystem (see Smart Battery Data).

SMBus protocol

Each message transaction on SMBus follows the format of one of the defined SMBus protocols. The SMBus protocols are a subset of the data transfer formats defined in the I2C specifications. I2C devices that can be accessed through one of the SMBus protocols are compatible with the SMBus specifications. I2C devices that do not adhere to these protocols cannot be accessed by standard methods as defined in the SMBus and Advanced

Configuration and Power Management Interface (abbreviated to ACPI) specifications.

Address resolution protocol

The SMBus uses I2C hardware and I2C hardware addressing, but adds second-level software for building special systems. Additionally, its specifications include an Address Resolution Protocol that can make dynamic address allocations. Dynamic reconfiguration of the hardware and software allow bus devices to be 'hot-plugged' and used immediately, without restarting the system. The devices are recognized automatically and assigned unique addresses. This advantage results in a plug-and-play user interface. In this protocol there is a very useful distinction between a system host and all the other devices in the system, that is the host provides address assignment function.

SMBus slave byte control

The slave byte control of SMBus receiver is the same as I2C. It allows the ACK control of each byte. The Slave Byte Control mode must be enabled by setting SBCTL bit in I2C_CTL0 register.

Host notify protocol

When the SMBHAEN bit in the I2C_CTL0 register is set, the SMBus supports the host notify protocol. In this protocol, the device acts as a master and the host as a slave, and the host will acknowledge the SMBus host address.

Time-out feature

SMBus has a time-out feature which resets devices if a communication takes too long. This explains the minimum clock frequency of 10 kHz to prevent locking up the bus. I2C can be a 'DC' bus, meaning that a slave device stretches the master clock when performing some routine while the master is accessing it. This will notify to the master that the slave is busy but does not want to lose the communication. The slave device will allow continuation after its task is completed. There is no limit in the I2C bus protocol as to how long this delay can be, whereas for a SMBus system, it would be limited to 25~35ms. SMBus protocol just assumes that if something takes too long, then it means that there is a problem on the bus and that all devices must reset in order to clear this mode. Slave devices are not allowed to hold the clock low too long.

The timeout detection can be enabled by setting TOEN and EXTTOEN bits in the I2C_TIMEOUT register. The timer must be configured to guarantee that the timeout detected before the maximum time given in the SMBus specification.

The value programmed in BUSTOA[11:0] is used to check the t_{TIMEOUT} parameter. To detect the SCL low level timeout, the TOIDLE bit must be 0. And the timer can be enabled by setting the TOEN bit in the I2C_TIMEOUT register, after the TOEN bit is set, the BUSTOA[11:0] and the TOIDLE bit cannot be changed. If the low level time of SCL is greater than

$(BUSTOA+1)*2048*t_{I2CCLK}$, the TIMEOUT flag will be set in I2C_STAT register.

The BUSTOB[11:0] is used to check the $t_{LOW:SEXT}$ of the slave and the $t_{LOW:MEXT}$ of the master. The timer can be enabled by setting the EXTOKEN bit in the I2C_TIMEOUT register, after the EXTOKEN bit is set, the BUSTOB[11:0] cannot be changed. If the SCL stretching time of the SMBus peripheral is greater than $(BUSTOB+1)*2048*t_{I2CCLK}$ and within the timeout interval described in the bus idle detection section, the TIMEOUT bit in the I2C_STAT register will be set.

Packet error checking

There is a CRC-8 calculator in I2C block to perform Packet Error Checking for I2C data. A PEC (packet error code) byte is appended at the end of each transfer. The byte is calculated as CRC-8 checksum, calculated over the entire message including the address and read/write bit. The polynomial used is x^8+x^2+x+1 (the CRC-8-ATM HEC algorithm, initialized to zero).

When I2C is disabled, the PEC can be enabled by setting the PECEN bit in I2C_CTL0 register. Since the PEC transmission is managed by BYTENUM[7:0] in I2C_CTL1 register, SBCTL bit must be set when act as a slave. When PECTRANS is set and the RELOAD bit is cleared, PEC is transmitted after the BYTENUM[7:0]-1 data byte. The PECTRANS has no effect if RELOAD is set.

SMBus alert

The SMBus has an extra optional shared interrupt signal called SMBALERT# which can be used by slaves to tell the host to ask its slaves about events of interest. The host processes the interrupt and accesses all SMBALERT# devices through the Alert Response Address at the same time. If the SMBALERT# is pulled low by the devices, the devices will acknowledge the Alert Response Address. When SMBHAEN is 0, it is configured as a slave device, the SMBA pin will be pulled low by setting the SMBALTEN bit in the I2C_CTL0 register. Meanwhile the Alert Response Address is enabled. When SMBHAEN is 1, it is configured as a host, and the SMBALTEN is 1, as soon as a falling edge is detected on the SMBA pin, the SMBALT flag will be set in the I2C_STAT register. If the ERRIE bit is set in the I2C_CTL0 register, an interrupt will be generated. When SMBALTEN is 0, the level of ALERT line is considered high even if the SMBA pin is low. The SMBA pin can be used as a standard GPIO if SMBALTEN is 0.

Bus idle detection

If the master detects that the high level duration of the clock and data signals is greater than $t_{HIGH,MAX}$, the bus can be considered idle.

This timing parameter includes the case of a master that has been dynamically added to the bus and may not have detected a state transition on a SMBCLK or SMBDAT lines. In this case, in order to ensure that there is no ongoing transmission, the master must wait long enough.

The BUSTOA[11:0] bits must be programmed with the timer reload value to enable the t_{IDLE} check in order to obtain the t_{IDLE} parameter. To detect SCL and SDA high level timeouts, the TOIDLE bit must be set. Then setting the TOEN bit in the I2C_TIMEOUT register to enable the timer, after the TOEN bit is set, the BUSTOA[11:0] bit and the TOIDLE bit cannot be changed. If the high level time of both SCL and SDA is greater than $(BUSTOA+1)*4*t_{I2CCLK}$, the TIMEOUT flag will be set in the I2C_STAT register.

SMBus slave mode

The SMBus receiver must be able to NACK each command or data it receives. For ACK control in slave mode, slave byte control mode can be enabled by setting SBCTL bit in I2C_CTL0 register.

SMBus-specific addresses should be enabled when needed. The SMBus Device Default address (0b1100 001) is enabled by setting the SMBDAEN bit in the I2C_CTL0 register. The SMBus Host address (0b0001 000) is enabled by setting the SMBHAEN bit in the I2C_CTL0 register. The Alert Response Address (0b0001 100) is enabled by setting the SMBALTEN bit in the I2C_CTL0 register.

26.3.10 SMBus mode

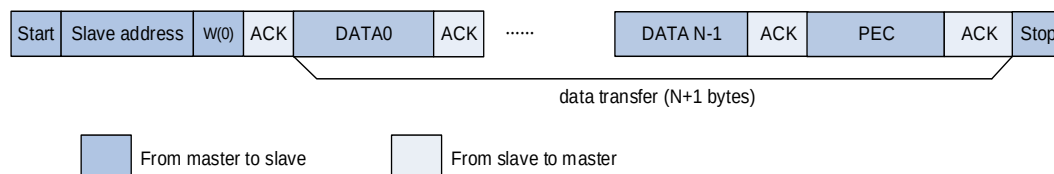
SMBus master transmitter and slave receiver

The PEC in SMBus master mode can be transmitted by setting the PECTRANS bit before setting the START bit, and the number of bytes in the BYTENUM[7:0] field must be configured. In this case, the total number of transmissions when TI interrupt occur is BYTENUM-1. So if BYTENUM=0x1 and PECTRANS bit is set, the data in I2C_PEC register will be transmitted automatically. If AUTOEND is 1 the SMBus master will send the STOP signal after the PEC byte automatically. If the AUTOEND is 0, the SMBus master can send a RESTART signal after the PEC. The PEC byte in I2C_PEC register will be sent after BYTENUM-1 bytes, and the TC flag will be set after PEC is sent, then the SCL line is stretched low. The RESTART must be set in the TC interrupt routine.

When used as slave receiver, in order to allow PEC checking at the end of the number of bytes transmitted, SBCTL must be set. To configure ack control for each byte, the RELOAD must be set. In order to check the PEC byte, it is necessary to clear the RELOAD bit and set PECTRANS bit. After receiving BYTENUM-1 data, the next received byte will be compared with the data in the I2C_PEC register. If the PEC values does not match, the NACK is automatically generated. If the PEC values matches, the ACK is automatically generated, regardless of the NACKEN bit value. When PEC byte is received, it is also copied into the I2C_RDATA register, and RBNE flag will be set. If the ERRIE bit in I2C_CTL0 register is 1, when PEC value does not match, the PECERR flag will be set and the interrupt will be generated. If ACK control is not required, then PECTRANS can be set to 1 and BYTENUM can be programmed according to the number of bytes to be received.

Note: After the RELOAD bit is set, the PECTRANS cannot be changed.

Figure 26-22. SMBus master transmitter and slave receiver communication flow



SMBus master receiver and slave transmitter

If the SMBus master is required to receive PEC at the end of bytes transfer, automatic end mode can be enabled. Before sending a START signal on the bus, PECTRANS bit must be set and slave addresses must be programmed. After receiving BYTENUM-1 data, the next received byte will be compared with the data in the I2C_PEC register automatically. A NACK is respond to the PEC byte before STOP signal.

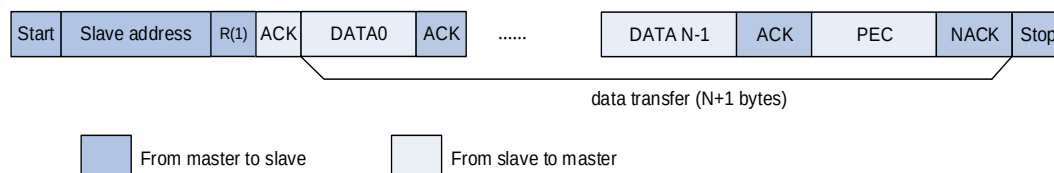
If the SMBus master receiver is required to generate a RESTART signal after receiving PEC byte, automatic end mode must be disabled. Before sending a START signal to the bus, PECTRANS bit must be set and slave addresses must be programmed. After receiving BYTENUM-1 data, the next received byte will be compared with the data in the I2C_PEC register automatically. The TC flag will be set after PEC is sent, then the SCL line is stretched low. The RESTART can be set in the TC interrupt routine.

When used as slave transmitter, in order to allow PEC transmission at the end of BYTENUM[7:0] bytes, SBCTL must be set. If PECTRANS bit is set, the number of bytes in BYTENUM[7:0] contains PEC byte. In this case, if the number of bytes requested by the master is greater than BYTENUM-1, the total number of TI interrupts will be BYTENUM-1, and the data of the I2C_PEC register will be transmitted automatically.

Note:

1. After the RELOAD bit is set, the PECTRANS cannot be changed.
2. When SMBus acts as a slave transmitter and uses the PEC function, BYTENUM[7:0] must be configured before detecting the START signal.

Figure 26-23. SMBus master receiver and slave transmitter communication flow



26.3.11 Wakeup from power saving modes

When the address of I2C matches correctly, it can wake up MCU from Sleep mode and Deep-sleep mode. In order to wake up from these power saving modes, WUEN bit must be set in

the I2C_CTL0 register and the IRC32M must be selected as the clock source for I2CCLK. During Sleep mode and Deep-sleep mode, the IRC32M is switched off. The I2C interface switches the IRC32M on, and stretches SCL low until IRC32M is woken up when a START is detected. Then the IRC32M is used as the clock of I2C to receive the address. When address matching is detected, I2C stretches SCL during MCU wake-up. The SCL is released until the software clears the ADDSEND flag and the transmission proceeds normally. If the detected address does not match, IRC32M will be closed again and the MCU will not be wake up.

Only an address match interrupt (ADDMIE=1) can wakeup the MCU. If the clock source of I2C is the system clock, or WUEN = 0, IRC32M will not switched on after receiving start signal. When wakeup from power saving mode is enabled, the digital filter must be disabled and the SS bit in I2C_CTL0 must be cleared. Before entering power saving mode, the I2C peripheral must be disabled (I2CEN=0) if wakeup from power saving mode is disabled (WUEN=0).

26.3.12 Use DMA for data transfer

As is shown in I2C slave mode and I2C master mode, each time TI or RBNE is asserted, software should write or read a byte, this may cause CPU's high overload. The DMA controller can be used to process TI and RBNE flag: each time TI or RBNE is asserted, DMA controller does a read or write operation automatically.

The DMA transmission request is enabled by setting the DENT bit in I2C_CTL0 register. The DMA reception request is enabled by setting the DENR bit in I2C_CTL0 register. In master mode, the slave address, transmission direction, number of bytes and START bit are programmed by software. The DMA must be initialized before setting the START bit. The number of bytes to be transferred is configured in the BYTENUM[7:0] in I2C_CTL1 register. In slave mode, the DMA must be initialized before the address match event or in the ADDSEND interrupt routine, before clearing the ADDSEND flag.

26.3.13 I2C error and interrupts

The I2C error flags are listed in [Table 26-4. I2C error flags](#).

Table 26-4. I2C error flags

I2C Error Name	Description
BERR	Bus error
LOSTARB	Arbitration lost
OUERR	Overrun / Underrun flag
PECERR	CRC value doesn't match
TIMEOUT	Bus timeout in SMBus mode
SMBALT	SMBus Alert

The I2C interrupt events and flags are listed in [Table 26-5. I2C interrupt events](#).

Table 26-5. I2C interrupt events

Interrupt event	Event flag	Enable control bit
I2C_RDATA is not empty during receiving	RBNE	RBNEIE
Transmit interrupt	TI	TIE
STOP signal detected in slave mode	STPDET	STPDETIE
Transfer complete reload	TCR	TCIE
Transfer complete	TC	
Address match	ADDSEND	ADDMIE
Not acknowledge received	NACK	NACKIE
Bus error	BERR	ERRIE
Arbitration Lost	LOSTARB	
Overrun / Underrun error	OUERR	
PEC error	PECERR	
Timeout error	TIMEOUT	
SMBus Alert	SMBALT	

26.3.14 I2C debug mode

When the microcontroller enters the debug mode (Cortex®-M33 core halted), the SMBus timeout either continues to work normally or stops, depending on the I2C_HOLD configuration bits in the DBG module.

26.3.15 Event Link Function (Output)

The I2C handles event output for the event interconnection unit (EVIC) corresponding to the following sources:

- Communication error/communication event
- I2C_RDATA is not empty during receiving
- Transmit interrupt
- Transfer complete

The I2C has four types of interrupts event: communication error or communication event (arbitration lost, NACK received, timeout error, or STOP signal detected in slave mode), I2C_RDATA is not empty during receiving, transmit interrupt, and transfer complete. The corresponding event signals are sent to other modules via the EVIC when the interrupt source conditions are satisfied, regardless of the settings of the interrupt enable bits.

26.4 Register definition

I2C base address: 0x4000 5400

26.4.1 Control register 0 (I2C_CTL0)

Address offset: 0x00

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved							SMBEN	PECEN	SMBALT EN	SMBDAE N	SMBHAE N	GCEN	WUEN	SS	SBCTL
							rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DENR	DENT	Reserved		DNF[3:0]				ERRIE	TCIE	STPDETI E	NACKIE	ADDIE	RBNEIE	TIE	I2CEN
rw	rw			rw				rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:25	Reserved	Must be kept at reset value.
24	SMBEN	SMBus mode enable When working in I2C mode, this bit must be 0. When working in SMBus mode, this bit must be 1. 0: I2C mode 1: SMBus mode, I2C PAD voltage is changed to TTL level Note: This bit can be modified when I2CEN = 0.
23	PECEN	PEC Calculation Switch 0: PEC Calculation off 1: PEC Calculation on
22	SMBALTEN	SMBus Alert enable 0: SMBA pin is not pulled down (device mode) or SMBus Alert pin SMBA is disabled (host mode) 1: SMBA pin is pulled down (device mode) or SMBus Alert pin SMBA is enabled (host mode)
21	SMBDAEN	SMBus device default address enable 0: Device default address is disabled, the default address 0b1100001x will be not acknowledged. 1: Device default address is enabled, the default address 0b1100001x will be acknowledged.
20	SMBHAEN	SMBus Host address enable

		0: Host address is disabled, address 0b0001000x will be not acknowledged. 1: Host address is enabled, address 0b0001000x will be acknowledged.
19	GCEN	Whether or not to response to a General Call (0x00) 0: Slave won't response to a General Call 1: Slave will response to a General Call
18	WUEN	Wakeup from power saving mode enable This bit is cleared when mcu wakeup from power saving mode. 0: Wakeup from power saving mode disable. 1: Wakeup from power saving mode enable. Note: WUEN can be set only when DNF[3:0] = 0000.
17	SS	Whether to stretch SCL low when data is not ready in slave mode. This bit is set and cleared by software. 0: SCL Stretching is enabled 1: SCL Stretching is disabled Note: When in master mode, this bit must be 0. This bit can be modified when I2CEN = 0.
16	SBCTL	Slave byte control This bit is used to enable hardware byte control in slave mode. 0: Slave byte control is disabled 1: Slave byte control is enabled
15	DENR	DMA enable for reception 0: DMA is disabled for reception 1: DMA is enabled for reception
14	DENT	DMA enable for transmission 0: DMA is disabled for transmission 1: DMA is enabled for transmission
13:12	Reserved	Must be kept at reset value.
11:8	DNF[3:0]	Digital noise filter 0000: Digital filter is disabled 0001: Digital filter is enabled and filter spikes with a length of up to 1 t_{I2CCLK} ... 1111: Digital filter is enabled and filter spikes with a length of up to 15 t_{I2CCLK} These bits can only be modified when the I2C is disabled (I2CEN = 0).
7	ERRIE	Error interrupt enable 0: Error interrupt disabled 1: Error interrupt enabled. When BERR, LOSTARB, OUERR, PECERR, TIMEOUT or SMBALT bit is set, an interrupt will be generated.
6	TCIE	Transfer complete interrupt enable

		0: Transfer complete interrupt is disabled 1: Transfer complete interrupt is enabled
5	STPDETIE	Stop detection interrupt enable 0: Stop detection (STPDET) interrupt is disabled 1: Stop detection (STPDET) interrupt is enabled
4	NACKIE	Not acknowledge received interrupt enable 0: Not acknowledge (NACK) received interrupt is disabled 1: Not acknowledge (NACK) received interrupt is enabled
3	ADDMIE	Address match interrupt enable in slave mode 0: Address match (ADDSEND) interrupt is disabled 1: Address match (ADDSEND) interrupt is enabled
2	RBNEIE	Receive interrupt enable 0: Receive (RBNE) interrupt is disabled 1: Receive (RBNE) interrupt is enabled
1	TIE	Transmit interrupt enable 0: Transmit (TI) interrupt is disabled 1: Transmit (TI) interrupt is enabled
0	I2CEN	I2C peripheral enable 0: I2C is disabled 1: I2C is enabled

26.4.2 Control register 1 (I2C_CTL1)

Address offset: 0x04

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved					PECTRA NS	AUTOEN D	RELOAD	BYTENUM[7:0]							
					rw	rw	rw	rw							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NACKEN	STOP	START	HEAD10 R	ADD10E N	TRDIR	SADDRESS[9:0]									
rw	rw	rw	rw	rw	rw	rw									

Bits	Fields	Descriptions
31:27	Reserved	Must be kept at reset value.
26	PECTRANS	PEC Transfer Set by software.

		Cleared by hardware in the following cases: When PEC byte is transferred or ADDSEND bit is set or STOP signal is detected or I2CEN=0. 0: Don't transfer PEC value 1: Transfer PEC Note: This bit has no effect when RELOAD=1, or SBCTL=0 in slave mode.
25	AUTOEND	Automatic end mode in master mode 0: TC bit is set when the transfer of BYTENUM[7:0] bytes is completed. 1: a STOP signal is sent automatically when the transfer of BYTENUM[7:0] bytes is completed. Note: This bit works only when RELOAD=0. This bit is set and cleared by software.
24	RELOAD	Reload mode 0: After the data of BYTENUM[7:0] bytes transfer, the transfer is completed. 1: After data of BYTENUM[7:0] bytes transfer, the transfer is not completed and the new BYTENUM[7:0] will be reloaded. Every time when the BYTENUM[7:0] bytes have been transferred, the TCR bit in I2C_STAT register will be set. This bit is set and cleared by software.
23:16	BYTENUM[7:0]	Number of bytes to be transferred These bits are programmed with the number of bytes to be transferred. When SBCTL=0, these bits have no effect. Note: When the START bit is set to 1, this bit cannot be modified, and the software cannot change this bit during transmission.
15	NACKEN	Generate NACK in slave mode 0: an ACK is sent after receiving a new byte. 1: a NACK is sent after receiving a new byte. Note: The bit can be set by software, and cleared by hardware when the NACK is sent, or when a STOP signal is detected or ADDSEND is set, or when I2CEN=0. When PEC is enabled, whether to send an ACK or a NACK is not depend on the NACKEN bit. When SS=1, and the OUERR bit is set, the value of NACKEN is ignored and a NACK will be sent.
14	STOP	Generate a STOP signal on I2C bus This bit is set by software and cleared by hardware when I2CEN=0 or STOP condition is detected. 0: STOP will not be sent 1: STOP will be sent
13	START	Generate a START condition on I2C bus This bit is set by software and cleared by hardware after the address is sent. When the arbitration is lost, or a timeout error occurred, or I2CEN=0, this bit can also be cleared by hardware. It can be cleared by software by setting the ADDSEND bit in I2C_STATC register.

		0: START will not be sent 1: START will be sent
12	HEAD10R	10-bit address header executes read direction only in master receive mode 0: The 10 bit master receive address sequence is START + header of 10-bit address (write) + slave address byte 2 + RESTART + header of 10-bit address (read). 1: The 10 bit master receive address sequence is RESTART + header of 10-bit address (read). Note: When the START bit is set to 1, this bit cannot be modified, and the software cannot change this bit during transmission.
11	ADD10EN	10-bit addressing mode enable in master mode 0: 7-bit addressing in master mode 1: 10-bit addressing in master mode Note: When the START bit is set to 1, this bit cannot be modified, and the software cannot change this bit during transmission.
10	TRDIR	Transfer direction in master mode 0: Master transmit 1: Master receive Note: When the START bit is set to 1, this bit cannot be modified, and the software cannot change this bit during transmission.
9:0	SADDRESS[9:0]	Slave address to be sent SADDRESS[9:8]: Slave address bit 9:8 If ADD10EN = 0, these bits have no effect. If ADD10EN = 1, these bits should be written with bits 9:8 of the slave address to be sent. SADDRESS[7:1]: Slave address bit 7:1 If ADD10EN = 0, these bits should be written with the 7-bit slave address to be sent. If ADD10EN = 1, these bits should be written with bits 7:1 of the slave address to be sent. SADDRESS0: Slave address bit 0 If ADD10EN = 0, this bit has no effect. If ADD10EN = 1, this bit should be written with bit 0 of the slave address to be sent Note: When the START bit is set to 1, this bit cannot be modified, and the software cannot change this bit during transmission.

26.4.3 Slave address register 0 (I2C_SADDR0)

Address offset: 0x08

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDRES SEN	Reserved					ADDFOR MAT	ADDRESS[9:8]		ADDRESS[7:1]						ADDRES S0
rw						rw	rw		rw						rw

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15	ADDRESSEN	I2C address enable 0: I2C address disable. 1: I2C address enable.
14:11	Reserved	Must be kept at reset value.
10	ADDFORMAT	Address mode for the I2C slave 0: 7-bit address 1: 10-bit address Note: When ADDRESSEN is set, this bit should not be written.
9:8	ADDRESS[9:8]	Highest two bits of a 10-bit address Note: When ADDRESSEN is set, this bit should not be written.
7:1	ADDRESS[7:1]	7-bit address or bits 7:1 of a 10-bit address Note: 1. When ADDRESSEN is set, these bits should not be written. 2. When the slave address is set to the reserved address 0b'1111000, the slave will not respond if an address match occurs.
0	ADDRESS0	Bit 0 of a 10-bit address Note: When ADDRESSEN is set, this bit should not be written.

26.4.4 Slave address register 1 (I2C_SADDR1)

Address offset: 0x0C

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDRES S2EN	Reserved					ADDMSK2[2:0]			ADDRESS2[7:1]						Reserved
rw						rw			rw						

Bits	Fields	Descriptions

31:16	Reserved	Must be kept at reset value.
15	ADDRESS2EN	Second I2C address enable 0: Second I2C address disable. 1: Second I2C address enable.
14:11	Reserved	Must be kept at reset value.
10:8	ADDMSK2[2:0]	ADDRESS2[7:1] mask Defines which bits of ADDRESS2[7:1] are compared with an incoming address byte, and which bits are masked (don't care). 000: No mask, all the bits must be compared. n (001~110): ADDRESS2[n:1] is masked. Only ADDRESS2[7:n+1] are compared. 111: ADDRESS2[7:1] are masked. All 7-bit received addresses are acknowledged except the reserved address (0b0000xxx and 0b1111xxx). Note: When ADDRESS2EN is set, these bits should not be written. If ADDMSK2 is not equal to 0, the reserved I2C addresses (0b0000xxx and 0b1111xxx) are not acknowledged even if all the bits are matched.
7:1	ADDRESS2[7:1]	Second I2C address for the slave Note: 1. When ADDRESS2EN is set, these bits should not be written. 2. When the slave address is set to the reserved address 0b'1111000, the slave will not respond if an address match occurs.
0	Reserved	Must be kept at reset value.

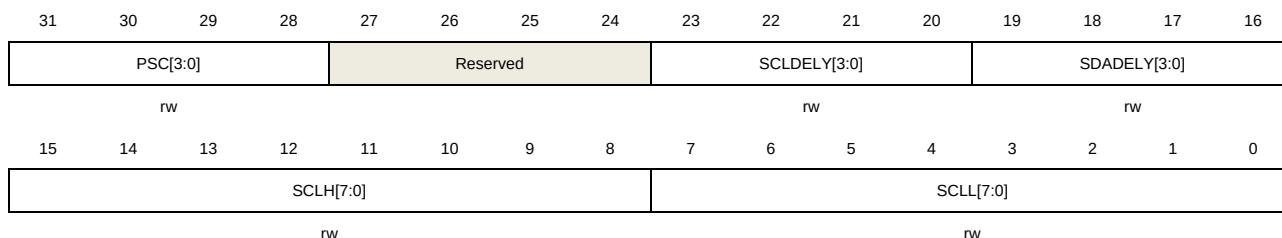
26.4.5 Timing register (I2C_TIMING)

Address offset: 0x10

Reset value: 0x0000 0000

This register cannot be modified when the I2C is enabled (I2CEN = 1).

This register has to be accessed by word (32-bit).



Bits	Fields	Descriptions
31:28	PSC[3:0]	Timing prescaler In order to generate the clock period t_{PSC} used for data setup and data hold counters, these bits are used to configure the prescaler for I2CCLK. The t_{PSC} is also used for SCL high and low level counters.

$$t_{PSC} = (PSC + 1) * t_{I2CCLK}$$

27:24	Reserved	Must be kept at reset value.
23:20	SCLDELY[3:0]	Data setup time A delay $t_{SCLDELY}$ between SDA edge and SCL rising edge can be generated by configuring these bits. And during $t_{SCLDELY}$, the SCL line is stretched low in master mode and in slave mode when SS = 0. $t_{SCLDELY} = (SCLDELY + 1) * t_{PSC}$
19:16	SDADELY[3:0]	Data hold time A delay $t_{SDADELY}$ between SCL falling edge and SDA edge can be generated by configuring these bits. And during $t_{SDADELY}$, the SCL line is stretched low in master mode and in slave mode when SS = 0. 0x0: $t_{SDADELY} = 1 * t_{PSC}$ 0x1-0xF: $t_{SDADELY} = SDADELY * t_{PSC}$
15:8	SCLH[7:0]	SCL high period SCL high period can be generated by configuring these bits. $t_{SCLH} = (SCLH + 1) * t_{PSC}$ Note: These bits can only be used in master mode.
7:0	SCLL[7:0]	SCL low period SCL low period can be generated by configuring these bits. $t_{SCLL} = (SCLL + 1) * t_{PSC}$ Note: These bits can only be used in master mode.

26.4.6 Timeout register (I2C_TIMEOUT)

Address offset: 0x14

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

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

Bits	Fields	Descriptions
31	EXTOEN	Extended clock timeout detection enable When a cumulative SCL stretch time is greater than $t_{LOW:EXT}$, a timeout error will be occurred. $t_{LOW:EXT} = (BUSTOB + 1) \times 2048 \times t_{I2CCLK}$. 0: Extended clock timeout detection is disabled. 1: Extended clock timeout detection is enabled.

30:28	Reserved	Must be kept at reset value.
27:16	BUSTOB[11:0]	Bus timeout B Configure the cumulative clock extension timeout. In master mode, the master cumulative clock low extend time $t_{LOW:MEXT}$ is detected. In slave mode, the slave cumulative clock low extend time $t_{LOW:SEXT}$ is detected. $t_{LOW:EXT} = (BUSTOB + 1) * 2048 * t_{I2CCLK}$ Note: These bits can be modified only when EXTOKEN = 0.
15	TOEN	Clock timeout detection enable If the SCL stretch time greater than $t_{TIMEOUT}$ when TOIDLE = 0 or high for more than t_{IDLE} when TOIDLE = 1, a timeout error is detected. 0: SCL timeout detection is disabled 1: SCL timeout detection is enabled
14:13	Reserved	Must be kept at reset value.
12	TOIDLE	Idle clock timeout detection 0: BUSTOA is used to detect SCL low timeout 1: BUSTOA is used to detect both SCL and SDA high timeout when the bus is idle Note: This bit can be written only when TOEN = 0.
11:0	BUSTOA[11:0]	Bus timeout A When TOIDLE = 0, $t_{TIMEOUT} = (BUSTOA + 1) * 2048 * t_{I2CCLK}$. When TOIDLE = 1, $t_{IDLE} = (BUSTOA + 1) * 4 * t_{I2CCLK}$. Note: These bits can be written only when TOEN = 0.

26.4.7 Status register (I2C_STAT)

Address offset: 0x18

Reset value: 0x0000 0001

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								READDR[6:0]							TR
r															r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I2CBSY	Reserved	SMBALT	TIMEOUT	PECERR	OUERR	LOSTAR B	BERR	TCR	TC	STPDET	NACK	ADDSEN D	RBNE	TI	TBE
r		r	r	r	r	r	r	r	r	r	r	r	r	r/w	r/w

Bits	Fields	Descriptions
31:24	Reserved	Must be kept at reset value.
23:17	READDR[6:0]	Received match address in slave mode When the ADDSEND bit is set, these bits store the matched address. In the case of

		a 10-bit address, READDR[6:0] stores the header of the 10-bit address followed by the 2 MSBs of the address.
16	TR	Whether the I2C is a transmitter or a receiver in slave mode This bit is updated when the ADDSEND bit is set. 0: Receiver 1: Transmitter
15	I2CBSY	Busy flag This bit is set by hardware when a START signal is detected and cleared by hardware after a STOP signal. When I2CEN=0, this bit is also cleared by hardware. 0: No I2C communication. 1: I2C communication active.
14	Reserved	Must be kept at reset value.
13	SMBALT	SMBus Alert When SMBHAEN=1, SMBALTEN=1, and a SMBALERT event (falling edge) is detected on SMBA pin, this bit will be set by hardware. It is cleared by software by setting the SMBALTC bit. This bit is cleared by hardware when I2CEN=0. 0: SMBALERT event is not detected on SMBA pin 1: SMBALERT event is detected on SMBA pin
12	TIMEOUT	TIMEOUT flag. When a timeout or extended clock timeout occurred, this bit will be set. It is cleared by software by setting the TIMEOUTC bit and cleared by hardware when I2CEN=0. 0: no timeout or extended clock timeout occur 1: a timeout or extended clock timeout occur
11	PECERR	PEC error This flag is set by hardware when the received PEC does not match with the content of I2C_PEC register. Then a NACK is automatically sent. It is cleared by software by setting the PECERRC bit and cleared by hardware when I2CEN=0. 0: Received PEC and content of I2C_PEC match 1: Received PEC and content of I2C_PEC don't match, I2C will send NACK regardless of NACKEN bit.
10	OUERR	Overflow/Underflow error in slave mode In slave mode with SS=1, when an overflow/underflow error occurs, this bit will be set by hardware. It is cleared by software by setting the OUERRC bit and cleared by hardware when I2CEN=0. 0: No overflow or underflow occurs 1: Overflow or underflow occurs
9	LOSTARB	Arbitration Lost It is cleared by software by setting the LOSTARBC bit and cleared by hardware when I2CEN=0.

		0: No arbitration lost. 1: Arbitration lost occurs and the I2C block changes back to slave mode.
8	BERR	Bus error When an unexpected START or STOP signal on I2C bus is detected, a bus error occurs and this bit will be set. It is cleared by software by setting BERRC bit and cleared by hardware when I2CEN=0. 0: No bus error 1: A bus error detected
7	TCR	Transfer complete reload This bit is set by hardware when RELOAD=1 and data of BYTENUM[7:0] bytes have been transferred. It is cleared by software when BYTENUM[7:0] is written to a non-zero value. 0: When RELOAD=1, transfer of BYTENUM[7:0] bytes is not completed 1: When RELOAD=1, transfer of BYTENUM[7:0] bytes is completed
6	TC	Transfer complete in master mode This bit is set by hardware when RELOAD=0, AUTOEND=0 and data of BYTENUM[7:0] bytes have been transferred. It is cleared by software when START bit or STOP bit is set. 0: Transfer of BYTENUM[7:0] bytes is not completed 1: Transfer of BYTENUM[7:0] bytes is completed
5	STPDET	STOP signal detected in slave mode This flag is set by hardware when a STOP signal is detected on the bus. It is cleared by software by setting STPDETC bit and cleared by hardware when I2CEN=0. 0: STOP signal is not detected. 1: STOP signal is detected.
4	NACK	Not Acknowledge flag This flag is set by hardware when a NACK is received. It is cleared by software by setting NACKC bit and cleared by hardware when I2CEN=0. 0: ACK is received. 1: NACK is received.
3	ADDSEND	Address received matches in slave mode. This bit is set by hardware when the received slave address matched with one of the enabled slave addresses. It is cleared by software by setting ADDSEND bit and cleared by hardware when I2CEN=0. 0: Received address not matched 1: Received address matched
2	RBNE	I2C_RDATA is not empty during receiving This bit is set by hardware when the received data is shift into the I2C_RDATA register. It is cleared when I2C_RDATA is read. 0: I2C_RDATA is empty

1: I2C_RDATA is not empty, software can read

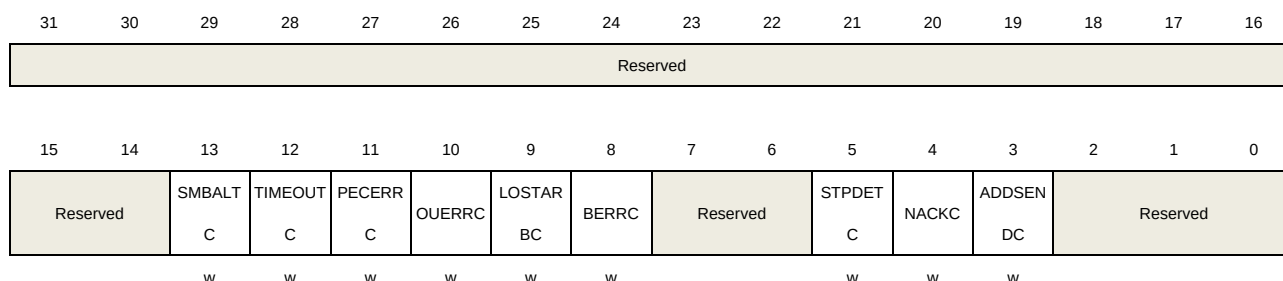
1	TI	Transmit interrupt This bit is set by hardware when the I2C_TDATA register is empty and the I2C is ready to transmit data. It is cleared when the next data to be sent is written in the I2C_TDATA register. When SS=1, this bit can be set by software, in order to generate a TI event (interrupt if TIE=1 or DMA request if DENT =1). 0: I2C_TDATA is not empty or the I2C is not ready to transmit data 1: I2C_TDATA is empty and the I2C is ready to transmit data
0	TBE	I2C_TDATA is empty during transmitting This bit is set by hardware when the I2C_TDATA register is empty. It is cleared when the next data to be sent is written in the I2C_TDATA register. This bit can be set by software in order to empty the I2C_TDATA register. 0: I2C_TDATA is not empty 1: I2C_TDATA is empty

26.4.8 Status clear register (I2C_STATC)

Address offset: 0x1C

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



Bits	Fields	Descriptions
31:14	Reserved	Must be kept at reset value.
13	SMBALTC	SMBus alert flag clear. Software can clear the SMBALT bit of I2C_STAT by writing 1 to this bit.
12	TIMEOUTC	TIMEOUT flag clear. Software can clear the TIMEOUT bit of I2C_STAT by writing 1 to this bit.
11	PECERRC	PEC error flag clear. Software can clear the PECERR bit of I2C_STAT by writing 1 to this bit.
10	OUERRC	Overrun/Underrun flag clear. Software can clear the OUERR bit of I2C_STAT by writing 1 to this bit.
9	LOSTARBC	Arbitration Lost flag clear.

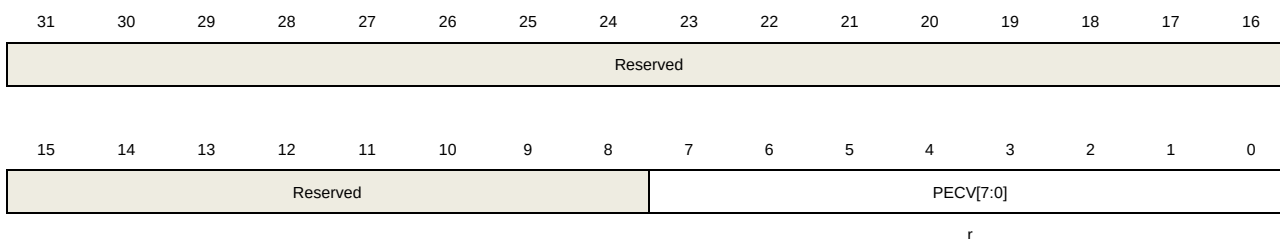
		Software can clear the LOSTARB bit of I2C_STAT by writing 1 to this bit.
8	BERRC	Bus error flag clear. Software can clear the BERR bit of I2C_STAT by writing 1 to this bit.
7:6	Reserved	Must be kept at reset value.
5	STPDETC	STPDET flag clear Software can clear the STPDET bit of I2C_STAT by writing 1 to this bit.
4	NACKC	Not Acknowledge flag clear Software can clear the NACK bit of I2C_STAT by writing 1 to this bit.
3	ADDSENDC	ADDSEND flag clear Software can clear the ADDSEND bit of I2C_STAT by writing 1 to this bit.
2:0	Reserved	Must be kept at reset value.

26.4.9 PEC register (I2C_PEC)

Address offset: 0x20

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



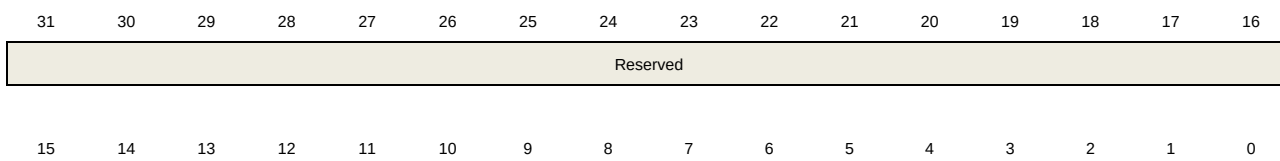
Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7:0	PECV[7:0]	Packet Error Checking Value that calculated by hardware when PEC is enabled. PECV is cleared by hardware when I2CEN = 0.

26.4.10 Receive data register (I2C_RDATA)

Address offset: 0x24

Reset value: 0x0000 0000

This register can be accessed by word (32-bit).



Reserved	RDATA[7:0]

r

Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7:0	RDATA[7:0]	Receive data value

26.4.11 Transmit data register (I2C_TDATA)

Address offset: 0x28

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

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

rw

Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7:0	TDATA[7:0]	Transmit data value

26.4.12 Control register 2 (I2C_CTL2)

Address offset: 0x90

Reset value: 0x0000 FE00

This register has to be accessed by word (32-bit).

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

rw

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:9	ADDM[6:0]	Defines which bits of ADDRESS[7:1] are compared with an incoming address byte, and which bits are ignored. Any bit set to 1 in ADDM[6:0] enables comparisons with the corresponding bit in ADDRESS[7:1]. Bits set to 0 are ignored (can be either 0

or 1 in the incoming address).

8:0

Reserved

Must be kept at reset value.

27 Serial peripheral interface (SPI)

27.1 Overview

The SPI module can communicate with external devices using the SPI protocol.

The serial peripheral interface (SPI) provides a SPI protocol of data transmission and reception function in master or slave mode. Both full-duplex and simplex communication modes are supported, with hardware CRC calculation and checking. Quad-SPI master mode is supported.

27.2 Characteristics

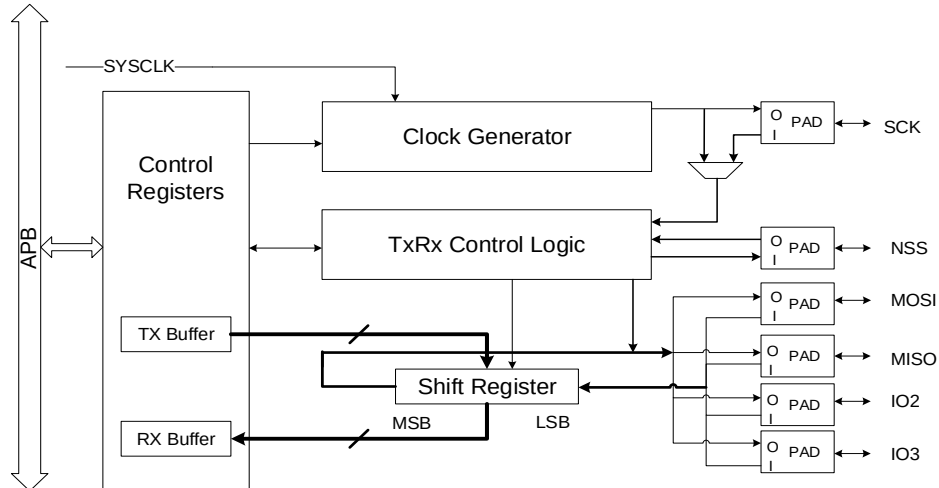
27.2.1 SPI characteristics

- Master or slave operation with full-duplex or half-duplex or simplex mode.
- Separate transmit and receive buffer, 16 bits wide.
- Data frame size can be 8 or 16 bits.
- Bit order can be LSB or MSB.
- Software and hardware NSS management.
- Hardware CRC calculation, transmission and checking.
- Transmission and reception using DMA.
- SPI TI mode supported.
- SPI NSS pulse mode supported.
- Quad-SPI configuration available in master mode.

27.3 SPI function overview

27.3.1 SPI block diagram

Figure 27-1. Block diagram of SPI



27.3.2 SPI signal description

Normal configuration (not Quad-SPI mode)

Table 27-1. SPI signal description

Pin name	Direction	Description
SCK	I/O	Master: SPI clock output Slave: SPI clock input
MISO	I/O	Master: Data reception line Slave: Data transmission line Master with bidirectional mode: Not used Slave with bidirectional mode: Data transmission and reception line.
MOSI	I/O	Master: Data transmission line Slave: Data reception line Master with bidirectional mode: Data transmission and reception line. Slave with bidirectional mode: Not used
NSS	I/O	Software NSS mode: not used Master in hardware NSS mode: when NSSDRV=1, it is NSS output, suitable for single master application; when NSSDRV=0, it is NSS input, suitable for multi-master application.

Pin name	Direction	Description
		Slave in hardware NSS mode: NSS input, as a chip select signal for slave.

Quad-SPI configuration

SPI is in single wire mode by default and enters into Quad-SPI mode after QMOD bit in SPI_QCTL register is set. Quad-SPI mode can only work in master mode.

The SPI is connected to external devices through 6 pins in Quad-SPI mode:

Table 27-2. Quad-SPI signal description

Pin name	Direction	Description
SCK	O	SPI clock output
MOSI	I/O	Transmission/Reception data 0
MISO	I/O	Transmission/Reception data 1
IO2	I/O	Transmission/Reception data 2
IO3	I/O	Transmission/Reception data 3
NSS	O	NSS output

27.3.3 SPI clock timing and data format

CKPL and CKPH bits in SPI_CTL0 register decide the timing of SPI clock and data signal. The CKPL bit decides the SCK level when SPI is in idle state and CKPH bit decides either first or second clock edge is a valid sampling edge. These bits take no effect in TI mode.

Figure 27-2. SPI timing diagram in normal mode

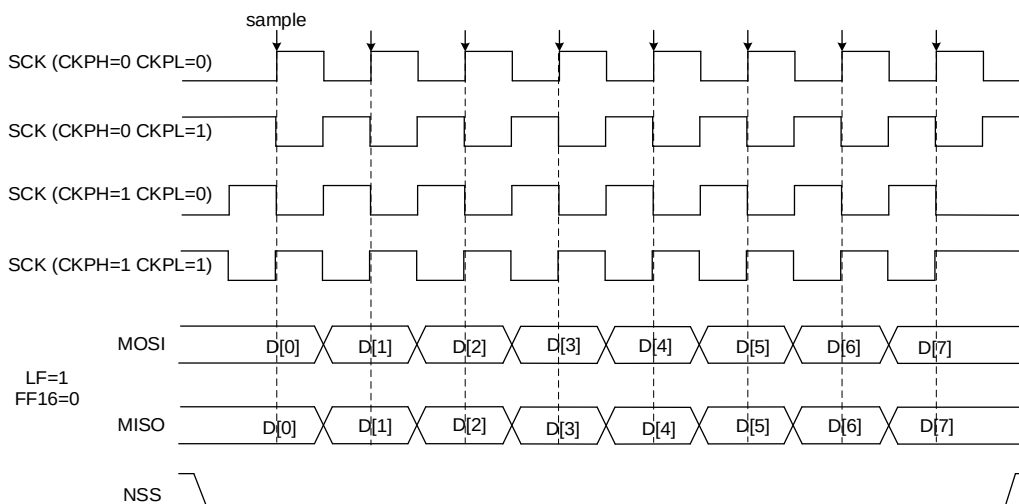
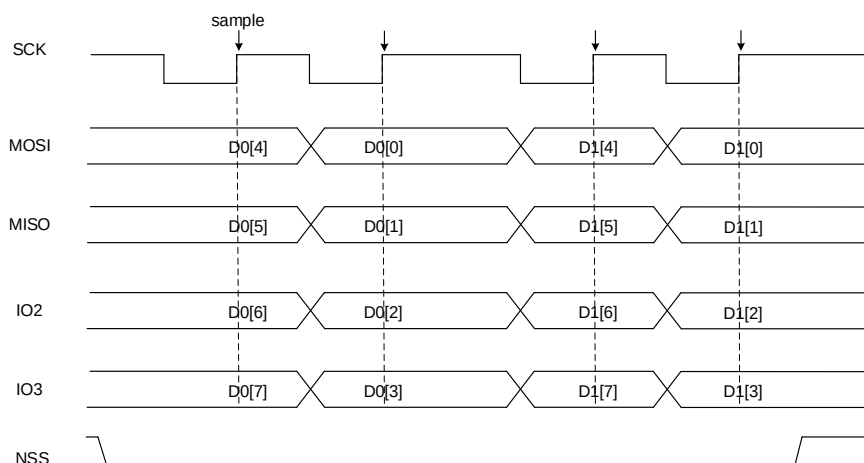


Figure 27-3. SPI timing diagram in Quad-SPI mode (CKPL=1, CKPH=1, LF=0)

In normal mode, the length of data is configured by the FF16 bit in the SPI_CTL0 register. Data length is 16 bits if FF16=1, otherwise is 8 bits. The data frame length is fixed to 8 bits in Quad-SPI mode.

Data order is configured by the LF bit in SPI_CTL0 register, and SPI will first send the LSB first if LF=1, or the MSB first if LF=0. The data order is fixed to MSB first in TI mode.

27.3.4 NSS function

Slave mode

When slave mode is configured (MSTMOD=0), SPI gets NSS level from NSS pin in hardware NSS mode (SWNSSSEN = 0) or from SWNSS bit in software NSS mode (SWNSSSEN = 1), and SPI transmits/receives data only when NSS level is low. In software NSS mode, NSS pin is not used.

Table 27-3. NSS function in slave mode

Mode	Register configuration	Description
Slave hardware NSS mode	MSTMOD = 0 SWNSSSEN = 0	SPI slave gets NSS level from NSS pin.
Slave software NSS mode	MSTMOD = 0 SWNSSSEN = 1	SPI slave NSS level is determined by the SWNSS bit. SWNSS = 0: NSS level is low SWNSS = 1: NSS level is high

Master mode

In master mode (MSTMOD=1), if the application uses multi-master connection, NSS can be configured to hardware input mode (SWNSSSEN=0, NSSDRV=0) or software mode (SWNSSSEN=1). Then, once the NSS pin (in hardware NSS mode) or the SWNSS bit (in software NSS mode) goes low, the SPI automatically enters slave mode and triggers a master

fault flag CONFERR.

If the application wants to use NSS line to control the SPI slave, NSS should be configured to hardware output mode (SWNSSEN=0, NSSDRV=1). NSS goes low after SPI is enabled.

The application may also use a general purpose IO as NSS pin to realize more flexible NSS.

Table 27-4. NSS function in master mode

Mode	Register configuration	Description
Master hardware NSS output mode	MSTMOD = 1 SWNSSEN = 0 NSSDRV=1	Applicable to single-master mode. The master uses the NSS pin to control the SPI slave device. At this time, the NSS is configured as the hardware output mode. NSS goes low after enabling SPI.
Master hardware NSS input mode	MSTMOD = 1 SWNSSEN = 0 NSSDRV=0	Applicable to multi-master mode. At this time, NSS is configured as hardware input mode. Once the NSS pin is pulled low, SPI will automatically enter slave mode, and a master configuration error will occur and the CONFERR bit will be set to 1.
Master software NSS mode	MSTMOD = 1 SWNSSEN = 1 SWNSS = 0 NSSDRV: Don't care	Applicable to multi-master mode. Once SWNSS = 0, SPI will automatically enter slave mode, and a master configuration error will occur and the CONFERR bit will be 1.
	MSTMOD = 1 SWNSSEN = 1 SWNSS = 1 NSSDRV: Don't care	The slave can use hardware or software NSS mode.

27.3.5 SPI operating modes

Table 27-5. SPI operating modes

Mode	Description	Register configuration	Data pin usage
MFD	Master full-duplex	MSTMOD = 1 RO = 0 BDEN = 0 BDOEN: Don't care	MOSI: Transmission MISO: Reception
MTU	Master transmission with unidirectional connection	MSTMOD = 1 RO = 0 BDEN = 0 BDOEN: Don't care	MOSI: Transmission MISO: Not used
MRU	Master reception with	MSTMOD = 1	MOSI: Not used

Mode	Description	Register configuration	Data pin usage
	unidirectional connection	RO = 1 BDEN = 0 BDOEN: Don't care	MISO: Reception
MTB	Master transmission with bidirectional connection	MSTMOD = 1 RO = 0 BDEN = 1 BDOEN = 1	MOSI: Transmission MISO: Not used
MRB	Master reception with bidirectional connection	MSTMOD = 1 RO = 0 BDEN = 1 BDOEN = 0	MOSI: Reception MISO: Not used
SFD	Slave full-duplex	MSTMOD = 0 RO = 0 BDEN = 0 BDOEN: Don't care	MOSI: Reception MISO: Transmission
STU	Slave transmission with unidirectional connection	MSTMOD = 0 RO = 0 BDEN = 0 BDOEN: Don't care	MOSI: Not used MISO: Transmission
SRU	Slave reception with unidirectional connection	MSTMOD = 0 RO = 1 BDEN = 0 BDOEN: Don't care	MOSI: Reception MISO: Not used
STB	Slave transmission with bidirectional connection	MSTMOD = 0 RO = 0 BDEN = 1 BDOEN = 1	MOSI: Not used MISO: Transmission
SRB	Slave reception with bidirectional connection	MSTMOD = 0 RO = 0 BDEN = 1 BDOEN = 0	MOSI: Not used MISO: Reception

Figure 27-4. A typical full-duplex connection

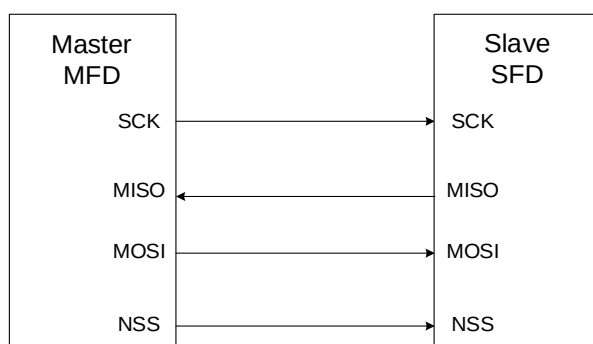


Figure 27-5. A typical simplex connection (Master: Receive, Slave: Transmit)

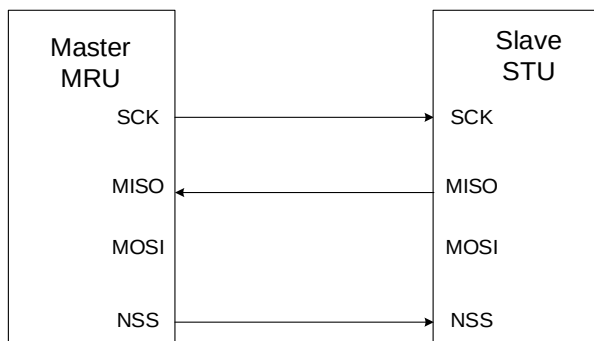


Figure 27-6. A typical simplex connection (Master: Transmit only, Slave: Receive)

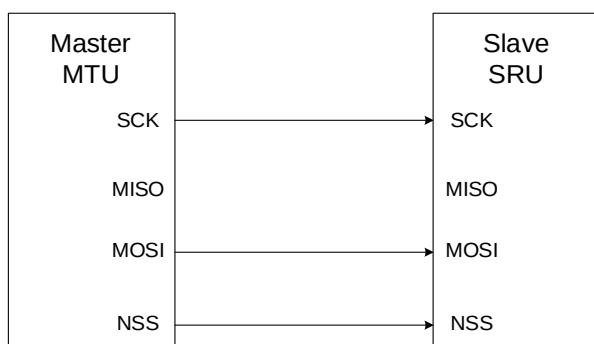
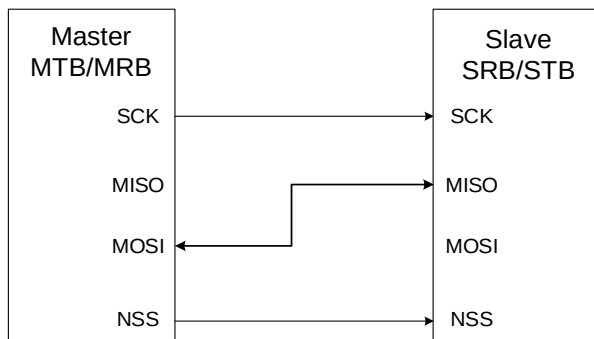


Figure 27-7. A typical bidirectional connection



SPI initialization sequence

Before transmitting or receiving data, application should follow the SPI initialization sequence described below:

1. If master mode or slave TI mode is used, program the PSC [2:0] bits in SPI_CTL0 register to generate SCK with desired baud rate or configure the Td time in TI mode, otherwise, ignore this step.
2. Program data format (FF16 bit in the SPI_CTL0 register).
3. Program the clock timing register (CKPL and CKPH bits in the SPI_CTL0 register).
4. Program the frame format (LF bit in the SPI_CTL0 register).
5. Program the NSS mode (SWNSSEN and NSSDRV bits in the SPI_CTL0 register) according to the application's demand as described above in [NSS function](#) section.

6. If TI mode is used, set TMOD bit in SPI_CTL1 register, otherwise, ignore this step.
7. If NSSP mode is used, set NSSP bit in SPI_CTL1 register, otherwise, ignore this step.
8. Configure MSTMOD, RO, BDEN and BDOEN depending on the operating modes described in [SPI operating modes](#) section.
9. If Quad-SPI mode is used, set the QMOD bit in SPI_QCTL register. Ignore this step if Quad-SPI mode is not used.
10. Enable the SPI (set the SPIEN bit).

Note: During communication, CKPH, CKPL, MSTMOD, PSC[2:0] and LF bits should not be changed.

SPI basic transmission and reception sequence

Transmission sequence

After the initialization sequence, the SPI is enabled and stays at idle state. In master mode, the transmission starts when the application writes a data into the transmit buffer. In slave mode the transmission starts when SCK clock signal at SCK pin begins to toggle and NSS level is low, so application should ensure that data is already written into transmit buffer before the transmission starts in slave mode.

When SPI begins to send a data frame, it first loads this data frame from the data buffer to the shift register and then begins to transmit the loaded data frame, TBE (transmit buffer empty) flag is set after the first bit of this frame is transmitted. After TBE flag is set, which means the transmit buffer is empty, the application should write SPI_DATA register again if it has more data to transmit.

In master mode, software should write the next data into SPI_DATA register before the transmission of current data frame is completed if it desires to generate continuous transmission.

Reception sequence

After the last valid sample clock, the incoming data will be moved from shift register to the receive buffer and RBNE (receive buffer not empty) will be set. The application should read SPI_DATA register to get the received data and this will clear the RBNE flag automatically. In MRU and MRB modes, hardware continuously sends clock signal to receive the next data frame, while in full-duplex master mode (MFD), hardware only receives the next data frame when the transmit buffer is not empty.

SPI operation sequence in different modes (Not Quad-SPI, TI mode or NSSP mode)

In full-duplex mode, either MFD or SFD, the RBNE and TBE flags should be monitored and then follow the sequences described above.

The transmission mode (MTU, MTB, STU or STB) is similar to the transmission sequence of

full-duplex mode except that the RBNE bit and RXORERR bit need to be ignored.

The master reception mode (MRU or MRB) is different from the reception sequence of full-duplex mode. In MRU or MRB mode, after SPI is enabled, the SPI continuously generates SCK until the SPI is disabled. So the application should ignore the TBE flag and read out reception buffer in time after the RBNE flag is set, otherwise a data overrun fault will occur.

The slave reception mode (SRU or SRB) is similar to the reception sequence of full-duplex mode except that the TBE bit need to be ignored.

SPI TI mode

SPI TI mode takes NSS as a special frame header flag signal and its operation sequence is similar to normal mode described above. The modes described above (MFD, MTU, MRU, MTB, MRB, SFD, STU, SRU, STB and SRB) are still supported in TI mode. While, in TI mode the CKPL and CKPH bits in SPI_CTL0 registers take no effect and the SCK sample edge is falling edge.

Figure 27-8. Timing diagram of TI master mode with discontinuous transfer

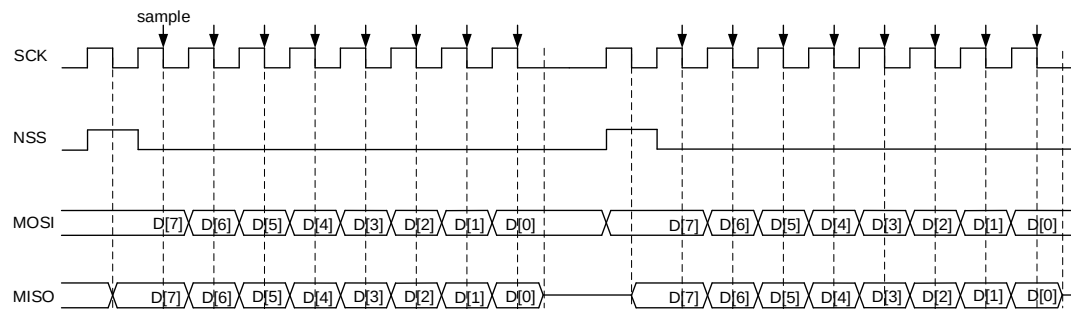
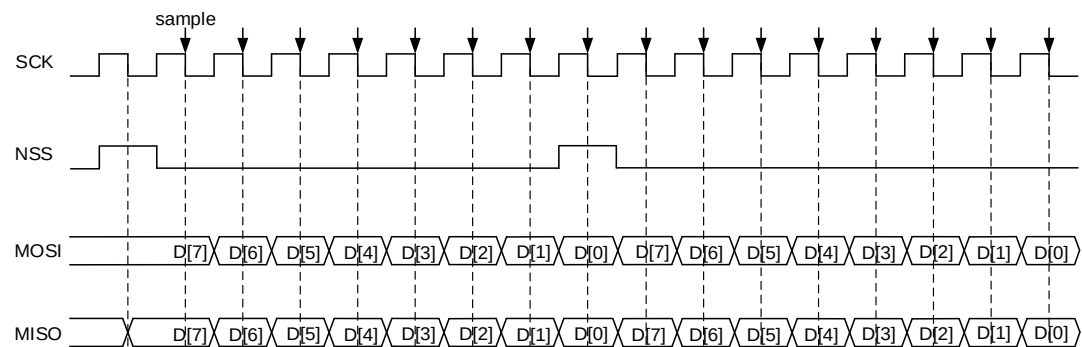
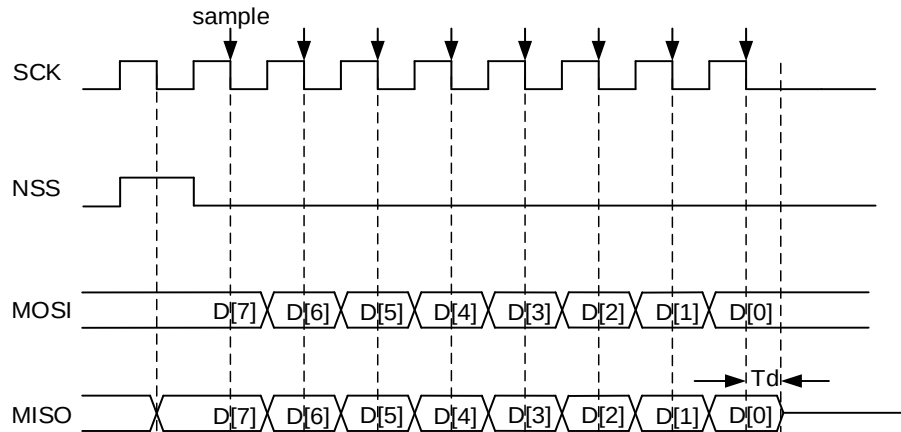


Figure 27-9. Timing diagram of TI master mode with continuous transfer



In master TI mode, SPI can perform continuous or non-continuous transfer. If the master writes SPI_DATA register fast enough, the transfer is continuous, otherwise non-continuous. In non-continuous transfer, there is an extra header clock cycle before each byte. While in continuous transfer, the extra header clock cycle only exists before the first byte and the following bytes' header clock is overlaid at the last bit of pervious bytes.

Figure 27-10. Timing diagram of TI slave mode


In slave TI mode, after the last rising edge of SCK in transfer, the slave begins to transmit the LSB bit of the last data byte, and after a half-bit time, the master begins to sample the line. To make sure that the master samples the right value, the slave should continue to drive this bit after the falling sample edge of SCK for a period of time before releasing the pin. This time is called T_d . T_d is decided by PSC[2:0] bits in SPI_CTL0 register.

$$T_d = \frac{T_{bit}}{2} + 5 * T_{pclk} \quad (27-1)$$

For example, if PSC[2:0] = 010, T_d is $9 * T_{pclk}$.

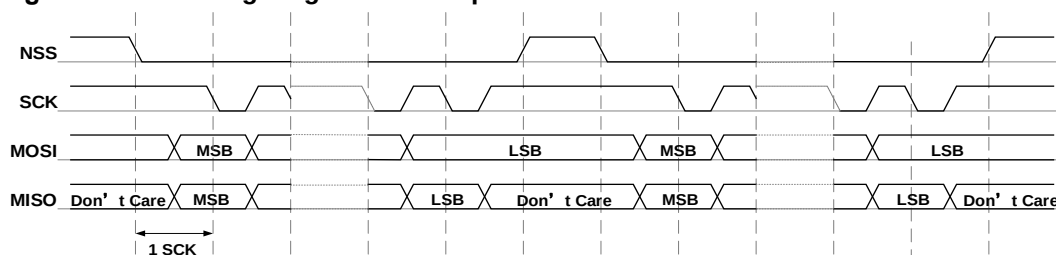
In slave mode, the slave also monitors the NSS signal and sets an error flag FERR if it detects an incorrect NSS behavior, for example, toggles at the middle bit of a byte.

NSS pulse mode operation sequence

This function is controlled by NSSP bit in SPI_CTL1 register. In order to implement this function, several additional conditions must be met: configure the device to master mode, frame format should follow the normal SPI protocol, select the first clock transition as the data capture edge.

In summary, MSTMOD = 1, NSSP = 1, CKPH = 0.

When NSS pulse mode is enabled, a pulse duration of at least 1 SCK clock period is inserted between two successive data frames depending on the status of internal data transmit buffer. Multiple SCK clock cycle intervals are possible if the transfer buffer stays empty. This function is designed for single master-slave configuration for the slave to latch data. The following diagram depicts its timing diagram.

Figure 27-11. Timing diagram of NSS pulse with continuous transmit

Quad-SPI mode operation sequence

The Quad-SPI mode is designed to control Quad-SPI flash.

In order to enter Quad-SPI mode, the software should first verify that the TBE bit is set and TRANS bit is cleared, then set QMOD bit in SPI_QCTL register. In Quad-SPI mode, BDEN, BDOEN, CRCEN, CRCNT, FF16, RO and LF bits in SPI_CTL0 register should be kept cleared and MSTMOD should be set to ensure that SPI is in master mode. SPIEN, PSC, CKPL and CKPH bits should be configured as desired.

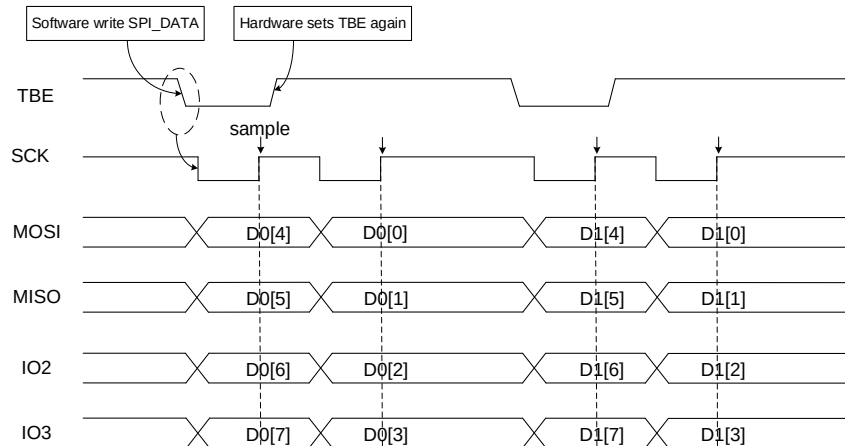
There are two operation modes in Quad-SPI mode: quad write and quad read, decided by QRD bit in SPI_QCTL register.

Quad write operation

SPI works in quad write mode when QMOD is set and QRD is cleared in SPI_QCTL register. In this mode, MOSI, MISO, IO2 and IO3 are all used as output pins. SPI begins to generate clock on SCK line and transmit data on MOSI, MISO, IO2 and IO3 as soon as data is written into SPI_DATA (TBE is cleared) and SPIEN is set. Once SPI starts transmission, it always checks TBE status at the end of a frame and stops when condition is not met.

The operation flow for transmitting in quad mode is shown below:

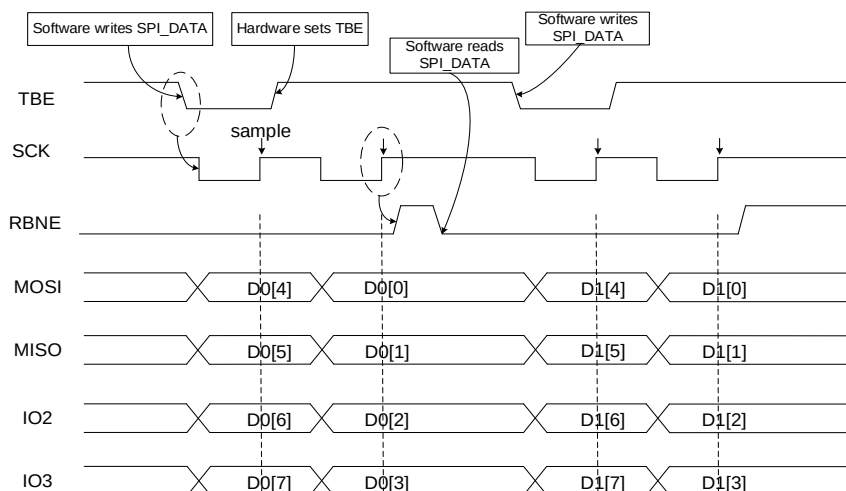
1. Configure clock prescaler, clock polarity, phase, etc. in SPI_CTL0 and SPI_CTL1 based on application requirements.
2. Set QMOD bit in SPI_QCTL register and then enable SPI by setting SPIEN in SPI_CTL0.
3. Write a byte to SPI_DATA register and the TBE will be cleared.
4. Wait until TBE is set by hardware again before writing the next byte.

Figure 27-12. Timing diagram of quad write operation in Quad-SPI mode**Quad read operation**

SPI works in quad read mode when QMOD and QRD bits are both set in SPI_QCTL register. In this mode, MOSI, MISO, IO2 and IO3 are all used as input pins. SPI begins to generate clock on SCK line as soon as a data is written into SPI_DATA (TBE is cleared) and SPIEN is set. Writing data into SPI_DATA is only to generate SCK clocks, so the written data can be any value. Once SPI starts transmission, it always checks SPIEN and TBE status at the end of a frame and stops when condition is not met. So, dummy data should always be written into SPI_DATA to generate SCK.

The operation flow for receiving in quad mode is shown below:

1. Configure clock prescaler, clock polarity, phase, etc. in SPI_CTL0 and SPI_CTL1 register based on application requirements.
2. Set QMOD and QRD bits in SPI_QCTL register and then enable SPI by setting SPIEN in SPI_CTL0 register.
3. Write an arbitrary byte (for example, 0xFF) to SPI_DATA register.
4. Wait until the RBNE flag is set and read SPI_DATA to get the received byte.
5. Write an arbitrary byte (for example, 0xFF) to SPI_DATA to receive the next byte.

Figure 27-13. Timing diagram of quad read operation in Quad-SPI mode

SPI disabling sequence

Different sequences are used to disable the SPI in different operation modes.

MFD SFD

Wait for the last RBNE flag and then receive the last data. Confirm that TBE=1 and TRANS=0. At last, disable the SPI by clearing SPIEN bit.

MTU MTB STU STB

Write the last data into SPI_DATA and wait until the TBE flag is set and then wait until the TRANS flag is cleared. Disable the SPI by clearing SPIEN bit.

MRU MRB

After getting the second last RBNE flag, read out this data and delay for a SCK clock time and then, disable the SPI by clearing SPIEN bit. Wait until the last RBNE flag is set and read out the last data.

SRU SRB

Application can disable the SPI when it doesn't want to receive data, and then wait until the TRANS=0 to ensure the ongoing transfer completed.

TI mode

The disabling sequence of TI mode is the same as the sequences described above.

NSS pulse mode

The disabling sequence of NSSP mode is the same as the sequences described above.

Quad-SPI mode

Before leaving quad wire mode or disabling SPI, software should first check that TBE bit is set and TRANS bit is cleared, then the QMOD bit in SPI_QCTL register and SPIEN bit in SPI_CTL0 register are cleared.

27.3.6 DMA function

The DMA frees the application from data writing and reading process during transfer, to improve the system efficiency.

DMA function in SPI is enabled by setting DMATEN and DMAREN bits in SPI_CTL1 register. To use DMA function, application should first configure DMA modules correctly, then configure SPI module according to the initialization sequence, at last enable SPI.

After being enabled, if DMATEN is set, SPI will generate a DMA request each time when TBE=1, then DMA will acknowledge to this request and write data into the SPI_DATA register automatically. If DMAREN is set, SPI will generate a DMA request each time when RBNE=1, then DMA will acknowledge to this request and read data from the SPI_DATA register automatically.

27.3.7 CRC function

There are two CRC calculators in SPI: one for transmission and the other for reception. The CRC calculation uses the polynomial defined in SPI_CRCPOLY register.

Application can enable the CRC function by setting CRCEN bit in SPI_CTL0 register. The CRC calculators calculate CRC for each bit transmitted and received on lines continuously, and the calculated CRC values can be read from SPI_TCRC and SPI_RCRC registers.

To transmit the calculated CRC value, application should set the CRCNT bit in SPI_CTL0 register after the last data is written to the transmit buffer. In full-duplex mode (MFD or SFD), when the SPI transmits a CRC and prepares to check the received CRC value, the SPI treats the incoming data as a CRC value. In reception mode (MRB, MRU, SRU and SRB), the application should set the CRCNT bit after the second last data frame is received. When CRC checking fails, the CRCERR flag will be set.

If DMA function is enabled, application doesn't need to configure CRCNT bit and hardware will automatically process the CRC transmitting and checking.

Note: When SPI is in slave mode and CRC function is enable, the CRC calculator is sensitive to input SCK clock whether SPI is enable or not. The software must enable CRC only when the clock is stable to avoid wrong CRC calculation. And when SPI works as a slave, the NSS internal signal needs to be kept low between the data phase and CRC phase.

When SPI is configured as slave mode and the CRC function is used, the CRC calculation will still be performed even when the NSS pin is high (when the NSS signal is high, as long as there is a clock pulse on the SCK pin, the CRC calculation will continue). This situation

occurs when the master device alternately communicates with multiple slave devices. At this time, it is recommended to restart the CRC function when the NSS signal is low. When the slave device is not selected (NSS signal is high) and switches to being selected as a new slave device (NSS signal is low), in order to maintain the synchronization of the next CRC calculation results between the master and slave devices, the CRC values at both ends should be cleared. It is recommended to clear the CRC value according to the following steps:

1. Disable the SPI module (SPIEN=0);
2. Clear the CRCEN bit (CRCEN=0);
3. Set the CRCEN bit (CRCEN=1);
4. Enable the SPI module (SPIEN=1).

27.3.8 SPI interrupts

Status flags

■ Transmit buffer empty flag (TBE)

This bit is set when the transmit buffer is empty, the software can write the next data to the transmit buffer by writing the SPI_DATA register.

■ Receive buffer not empty flag (RBNE)

This bit is set when receive buffer is not empty, which means that one data is received and stored in the receive buffer, and software can read the data by reading the SPI_DATA register.

■ SPI transmitting ongoing flag (TRANS)

TRANS is a status flag to indicate whether the transfer is ongoing or not. It is set and cleared by hardware and not controlled by software. This flag doesn't generate any interrupt.

Error flags

■ Configuration fault error (CONFERR)

CONFERR is an error flag in master mode. In NSS hardware mode and if the NSSDRV is not enabled, the CONFERR is set when the NSS pin is pulled low. In NSS software mode, the CONFERR is set when the SWNSS bit is 0. When the CONFERR is set, the SPIEN bit and the MSTMOD bit are cleared by hardware, the SPI is disabled and the device is forced into slave mode.

The SPIEN and MSTMOD bits are write protected until the CONFERR is cleared. The CONFERR bit of the slave cannot be set. In a multi-master configuration, the device can be in slave mode with CONFERR bit set, which means there might have been a multi-master conflict for system control.

■ Rx overrun error (RXORERR)

The RXORERR bit is set if a data is received when the RBNE is set. That means, the last data has not been read out and the newly incoming data is received. The receive buffer contents won't be covered with the newly incoming data, so the newly incoming data is lost.

■ Format error (FERR)

In slave TI mode, the slave also monitors the NSS signal and set an error flag if it detects an incorrect NSS behavior, for example: toggles at the middle bit of a byte.

■ CRC error (CRCERR)

When the CRCEN bit is set, the CRC calculation result of the received data in the SPI_RCRC register is compared with the received CRC value after the last data, the CRCERR is set when they are different.

Table 27-6. SPI interrupt requests

Flag	Description	Clear method	Interrupt enable bit
TBE	Transmit buffer empty	Write SPI_DATA register.	TBEIE
RBNE	Receive buffer not empty	Read SPI_DATA register.	RBNEIE
CONFERR	Configuration fault error	Read or write SPI_STAT register, then write SPI_CTL0 register.	ERRIE
RXORERR	Rx overrun error	Read SPI_DATA register, then read SPI_STAT register.	
CRCERR	CRC error	Write 0 to CRCERR bit	
FERR	TI Mode Format Error	Write 0 to FERR bit	

27.4 Register definition

SPI base address: 0x4001 3000

27.4.1 Control register 0 (SPI_CTL0)

Address offset: 0x00

Reset value: 0x0000 0000

This register can be accessed by half-word (16-bit) or word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BDEN	BDOEN	CRCEN	CRCNT	FF16	RO	SWNSS EN	SWNSS	LF	SPIEN	PSC[2:0]			MSTMOD	CKPL	CKPH
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw			rw	rw	rw

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15	BDEN	Bidirectional enable 0: 2 line unidirectional transmit mode 1: 1 line bidirectional transmit mode. The data transfers between the MOSI pin of master and the MISO pin of slave.
14	BDOEN	Bidirectional transmit output enable When BDEN is set, this bit determines the direction of transfer. 0: Work in receive-only mode 1: Work in transmit-only mode
13	CRCEN	CRC calculation enable 0: Disable CRC calculation 1: Enable CRC calculation
12	CRCNT	CRC next transfer 0: Next transfer is data 1: Next transfer is CRC value When the transfer is managed by DMA, CRC value is transferred by hardware. This bit should be cleared. In full-duplex or transmit-only mode, set this bit after the last data is written to SPI_DATA register. In receive-only mode, set this bit after the second last data is received.
11	FF16	Data frame format

		0: 8-bit data frame format 1: 16-bit data frame format
10	RO	Receive only When BDEN is cleared, this bit determines the direction of transfer. 0: Full-duplex mode 1: Receive-only mode
9	SWNSSEN	NSS software mode enable 0: NSS hardware mode. The NSS level depends on NSS pin. 1: NSS software mode. The NSS level depends on SWNSS bit. This bit has no meaning in SPI TI mode.
8	SWNSS	NSS pin selection in NSS software mode 0: NSS pin is pulled low 1: NSS pin is pulled high This bit effects only when the SWNSSEN bit is set. This bit has no meaning in SPI TI mode.
7	LF	LSB first mode 0: Transmit MSB first 1: Transmit LSB first This bit has no meaning in SPI TI mode.
6	SPIEN	SPI enable 0: Disable SPI peripheral 1: Enable SPI peripheral
5:3	PSC[2:0]	Master clock prescaler selection 000: PCLK/2 001: PCLK/4 010: PCLK/8 011: PCLK/16 100: PCLK/32 101: PCLK/64 110: PCLK/128 111: PCLK/256 PCLK means PCLK2 when using SPI.
2	MSTMOD	Master mode enable 0: Slave mode 1: Master mode
1	CKPL	Clock polarity selection 0: CLK pin is pulled low when SPI is idle 1: CLK pin is pulled high when SPI is idle
0	CKPH	Clock phase selection

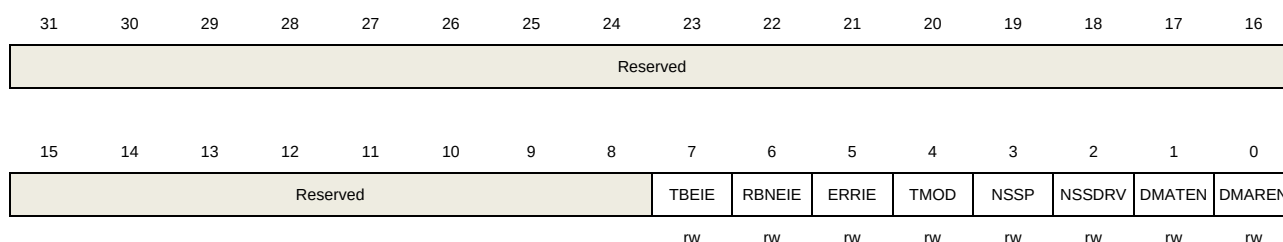
- 0: Capture the first data at the first clock transition
1: Capture the first data at the second clock transition

27.4.2 Control register 1 (SPI_CTL1)

Address offset: 0x04

Reset value: 0x0000 0000

This register can be accessed by half-word (16-bit) or word (32-bit).



Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7	TBEIE	Transmit buffer empty interrupt enable 0: Disable TBE interrupt 1: Enable TBE interrupt. An interrupt is generated when the TBE bit is set.
6	RBNEIE	Receive buffer not empty interrupt enable 0: Disable RBNE interrupt 1: Enable RBNE interrupt. An interrupt is generated when the RBNE bit is set.
5	ERRIE	Errors interrupt enable 0: Disable error interrupt 1: Enable error interrupt. An interrupt is generated when the CRCERR bit, the CONFERR bit, the RXORERR bit or the TXURERR bit is set.
4	TMOD	SPI TI mode enable 0: Disable SPI TI mode 1: Enable SPI TI mode
3	NSSP	SPI NSS pulse mode enable 0: Disable SPI NSS pulse mode 1: Enable SPI NSS pulse mode
2	NSSDRV	Drive NSS output 0: Disable NSS output 1: Enable NSS output
1	DMATEN	Transmit buffer DMA enable 0: Disable transmit buffer DMA 1: Enable transmit buffer DMA, when the TBE bit in SPI_STAT is set, there will be

a DMA request on corresponding DMA channel.

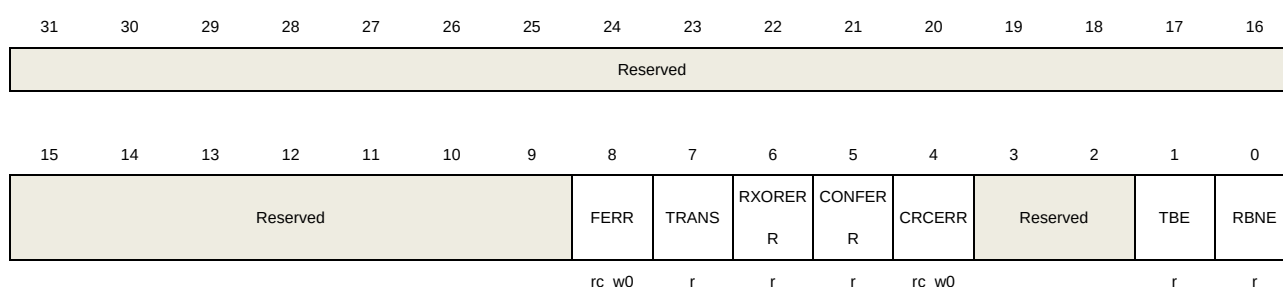
0	DMAREN	Receive buffer DMA enable 0: Disable receive buffer DMA 1: Enable receive buffer DMA, when the RBNE bit in SPI_STAT is set, there will be a DMA request on corresponding DMA channel.

27.4.3 Status register (SPI_STAT)

Address offset: 0x08

Reset value: 0x0000 0002

This register can be accessed by half-word (16-bit) or word (32-bit).



Bits	Fields	Descriptions
31:9	Reserved	Must be kept at reset value.
8	FERR	Format error SPI TI Mode: 0: No TI mode format error 1: TI mode format error occurs This bit is set by hardware and cleared by writing 0.
7	TRANS	Transmitting ongoing bit 0: SPI is idle. 1: SPI is currently transmitting and/or receiving a frame. This bit is set and cleared by hardware.
6	RXORERR	Reception overrun error bit 0: No reception overrun error occurs. 1: Reception overrun error occurs. This bit is set by hardware and cleared by a read operation on the SPI_DATA register followed by a read access to the SPI_STAT register.
5	CONFERR	SPI Configuration error 0: No configuration fault occurs. 1: Configuration fault occurred. (In master mode, the NSS pin is pulled low in NSS hardware mode or SWNSS bit is low in NSS software mode.) This bit is set by hardware and cleared by a read or write operation on the SPI_STAT

register followed by a write access to the SPI_CTL0 register.

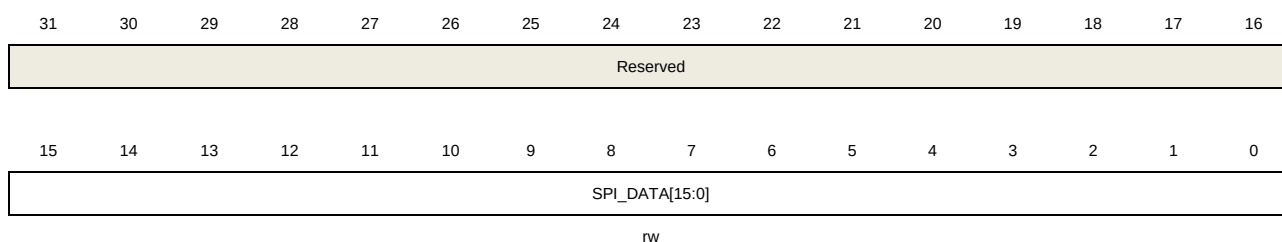
4	CRCERR	SPI CRC error bit 0: The SPI_RCRC value is equal to the received CRC data at last. 1: The SPI_RCRC value is not equal to the received CRC data at last. This bit is set by hardware and cleared by writing 0.
3:2	Reserved	Must be kept at reset value.
1	TBE	Transmit buffer empty 0: Transmit buffer is not empty 1: Transmit buffer is empty
0	RBNE	Receive buffer not empty 0: Receive buffer is empty 1: Receive buffer is not empty

27.4.4 Data register (SPI_DATA)

Address offset: 0x0C

Reset value: 0x0000 0000

This register can be accessed by half-word (16-bit) or word (32-bit).



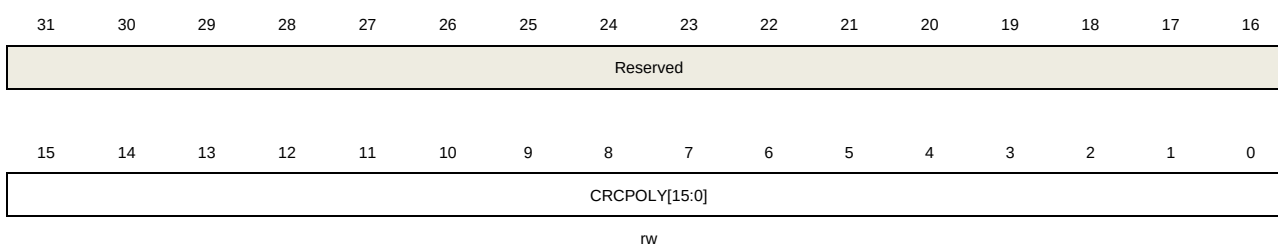
Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	SPI_DATA[15:0]	Data transfer register The hardware has two buffers, including transmit buffer and receive buffer. Write data to SPI_DATA will save the data to transmit buffer and read data from SPI_DATA will get the data from receive buffer. When the data frame format is set to 8-bit data, the SPI_DATA [15:8] is forced to 0 and the SPI_DATA[7:0] is used for transmission and reception, transmit buffer and receive buffer are 8-bit. If the data frame format is set to 16-bit data, the SPI_DATA[15:0] is used for transmission and reception, transmit buffer and receive buffer are 16-bit.

27.4.5 CRC polynomial register (SPI_CRCPOLY)

Address offset: 0x10

Reset value: 0x0000 0007

This register can be accessed by half-word (16-bit) or word (32-bit).



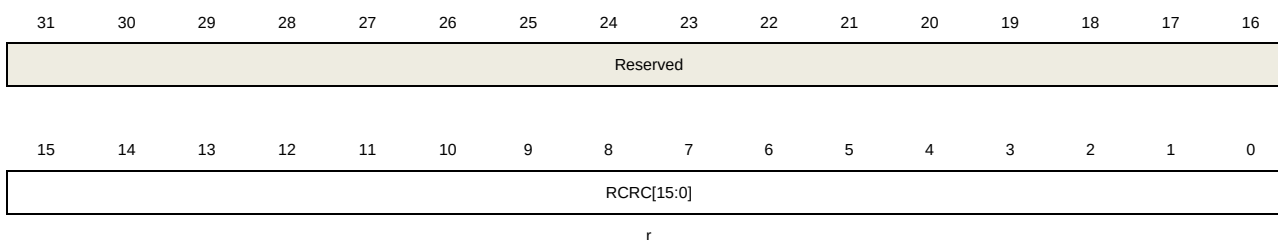
Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CRCPOLY[15:0]	CRC polynomial value These bits contain the CRC polynomial and they are used for CRC calculation. The default value is 0007h.

27.4.6 RX CRC register (SPI_RCRC)

Address offset: 0x14

Reset value: 0x0000 0000

This register can be accessed by half-word (16-bit) or word (32-bit).



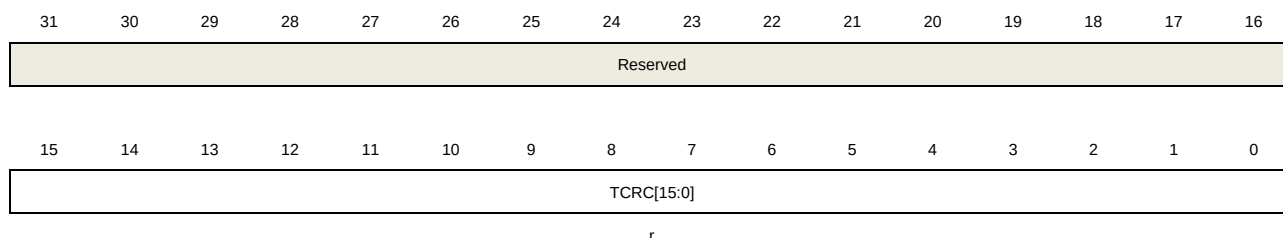
Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	RCRC[15:0]	RX CRC value When the CRCEN bit of SPI_CTL0 is set, the hardware computes the CRC value of the received bytes and saves them in RCRC register. If the data frame format is set to 8-bit data, CRC calculation is based on CRC8 standard, and saves the value in RCRC[7:0], when the data frame format is set to 16-bit data, CRC calculation is based on CRC16 standard, and saves the value in RCRC[15:0]. The hardware computes the CRC value after each received bit, when the TRANS is set, a read to this register could return an intermediate value. This register is reset when the CRCEN bit in SPI_CTL0 register or the SPIxRST bit in RCU reset register is set.

27.4.7 TX CRC register (SPI_TCRC)

Address offset: 0x18

Reset value: 0x0000 0000

This register can be accessed by half-word (16-bit) or word (32-bit).



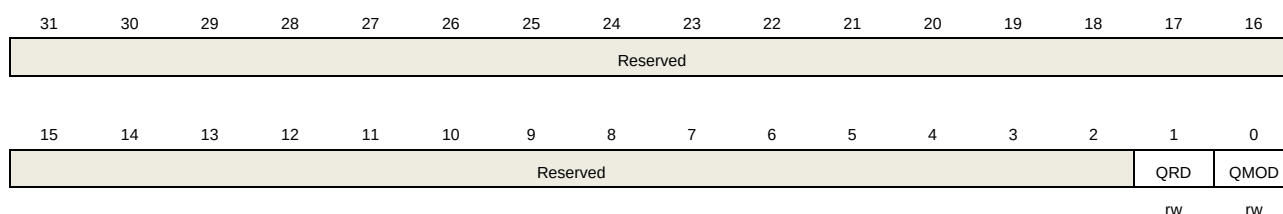
Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	TCRC[15:0]	TX CRC value When the CRCEN bit of SPI_CTL0 is set, the hardware computes the CRC value of the transmitted bytes and saves them in TCRC register. If the data frame format is set to 8-bit data, CRC calculation is based on CRC8 standard, and saves the value in TCRC[7:0], when the data frame format is set to 16-bit data, CRC calculation is based on CRC16 standard, and saves the value in TCRC[15:0]. The hardware computes the CRC value after each transmitted bit, when the TRANS is set, a read to this register could return an intermediate value. The different frame formats (LF bit of the SPI_CTL0) will get different CRC values. This register is reset when the CRCEN bit in SPI_CTL0 register or the SPIxRST bit in RCU reset register is set.

27.4.8 Quad-SPI mode control register (SPI_QCTL)

Address offset: 0x80

Reset value: 0x0000 0000

This register can be accessed by half-word (16-bit) or word (32-bit).



Bits	Fields	Descriptions
31:2	Reserved	Must be kept at reset value.
1	QRD	Quad-SPI mode read select 0: SPI is in quad wire write mode
0	QMOD	

		1: SPI is in quad wire read mode This bit should be only be configured when SPI is not busy (TRANS bit cleared).
0	QMOD	Quad-SPI mode enable 0: SPI is in single wire mode 1: SPI is in Quad-SPI mode This bit should only be configured when SPI is not busy (TRANS bit cleared).

28 Controller area network (CAN)

28.1 Overview

CAN bus (Controller Area Network) is a bus standard designed to allow microcontrollers and devices to communicate with each other without a host computer.

As CAN network interface, basic extended CAN supports the CAN protocols version 2.0A and B. The CAN interface automatically handles the transmission and the reception of CAN frames. The CAN provides 28 scalable/configurable identifier filter banks in GD32M531. The filters are used for selecting the input message as software requirement and otherwise discarding the message. Three transmit mailboxes are provided to the software for transfer messages. The transmission scheduler decides which mailbox will be transmitted firstly. Three complete messages can be stored in every FIFO. The FIFOs are managed completely by hardware. Two receiving FIFOs are used by hardware to store the incoming messages. In addition, the CAN controller provides all hardware functions, which supports the time-triggered communication option, in safety-critical applications.

28.2 Characteristics

- Supports CAN protocols version 2.0A, B.
- Baud rates up to 1 Mbit/s.
- Supports the time-triggered communication.
- Interrupt enable and clear.

Transmission

- Supports 3 transmit mailboxes.
- Supports priority of transmission message.
- Supports time stamp at SOF transmission.

Reception

- Supports 2 Rx FIFOs and each has 3 messages depth.
- 28 scalable/configurable identifier filter banks.
- FIFO lock.

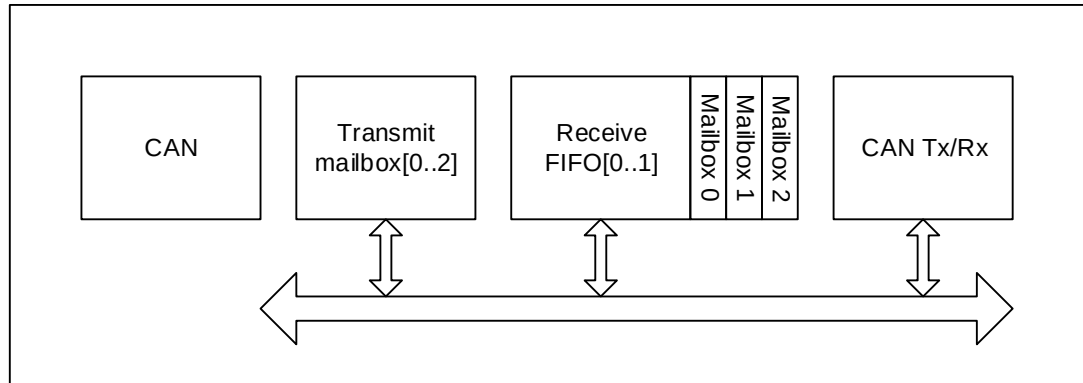
Time-triggered communication

- Disable retransmission automatically in time-triggered communication mode.
- 16-bit free timer.
- Time stamp on SOF reception.
- Time stamp sent in last two data bytes.

28.3 Function overview

[Figure 28-1. CAN module block diagram](#) shows the CAN block diagram.

Figure 28-1. CAN module block diagram



28.3.1 Working mode

The CAN interface has three working modes:

- Sleep working mode.
- Initial working mode.
- Normal working mode.

Sleep working mode

Sleep working mode is the default mode after reset. In sleep working mode, the CAN is in the low-power status and the CAN clock is stopped.

When SLPWMOD bit in CAN_CTL register is set, the CAN enters the sleep working mode. Then the SLPWS bit in CAN_STAT register is set by hardware.

To leave sleep working mode automatically: the AWU bit in CAN_CTL register is set and the CAN bus activity is detected. To leave sleep working mode by software: clear the SLPWMOD bit in CAN_CTL register.

Sleep working mode to initial working mode: set IWMOD bit and clear SLPWMOD bit in CAN_CTL register.

Sleep working mode to normal working mode: clear IWMOD and SLPWMOD bit in CAN_CTL register.

Initial working mode

When the configuration of CAN bus communication is needed to be changed, the CAN must enter initial working mode.

When IWMOD bit in CAN_CTL register is set, the CAN enters the initial working mode. Then the IWS bit in CAN_STAT register is set.

Initial working mode to sleep working mode: set SLPWMOD bit and clear IWMOD bit in CAN_CTL register.

Initial working mode to normal working mode: clear IWMOD bit and clear SLPWMOD bit in CAN_CTL register.

Normal working mode

The CAN could communicate with other CAN communication nodes in normal working mode.

To enter normal working mode: clear IWMOD and SLPWMOD bit in CAN_CTL register.

Normal working mode to sleep working mode: set SLPWMOD bit in CAN_CTL register and wait the current transmission or reception completed.

Normal working mode to initial working mode: set IWMOD bit in CAN_CTL register, and wait the current transmission or reception completed.

28.3.2 Communication modes

The CAN interface has four communication modes:

- Silent communication mode.
- Loopback communication mode.
- Loopback and silent communication mode.
- Normal communication mode.

Silent communication mode

Silent communication mode means reception available and transmission disable.

The RX pin of the CAN could detect the signal from the network and the TX pin always holds logical one.

When the SCMOD bit in CAN_BT register is set, the CAN enters the silent communication mode. When it is cleared, the CAN leaves silent communication mode.

Silent communication mode is useful for monitoring the network messages.

Loopback communication mode

Loopback communication mode means the transmitted messages are transferred into the Rx FIFOs, the RX pin is disconnected from the CAN network and the TX pin can still send messages to the CAN network.

Setting LCMOD bit in CAN_BT register to enter loopback communication mode, while clearing it to leave. Loopback communication mode is useful for self-test.

Loopback and silent communication mode

Loopback and silent communication mode means the RX and TX pins are disconnected from the CAN network while the transmitted messages are transferred into the Rx FIFOs.

Setting LCMOD and SCMOD bit in CAN_BT register to enter loopback and silent communication mode, while clearing them to leave.

Loopback and silent communication mode is used for self-test. The TX pin holds in recessive state. The RX pin holds in high impedance state.

Normal communication mode

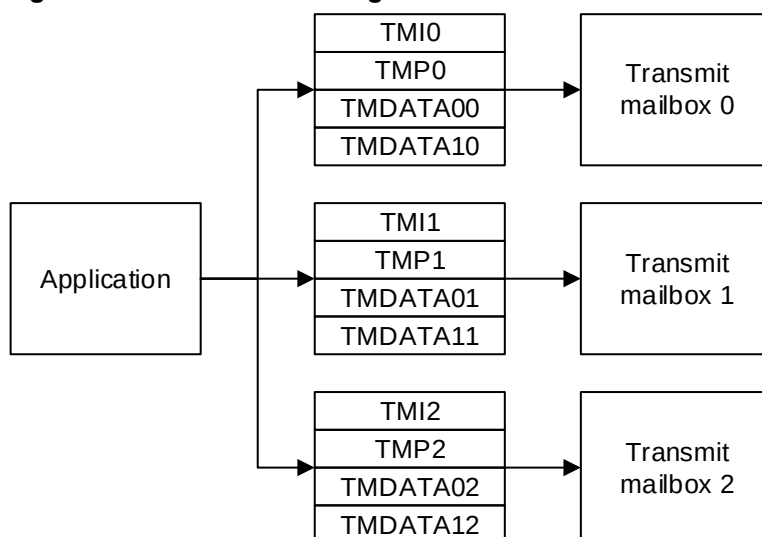
Normal communication mode is the default communication mode when the LCMOD and SCMOD bits in CAN_BT register are cleared.

28.3.3 Data transmission

Transmission register

Three transmit mailboxes are used for the application. Transmit mailboxes are used by configuring four transmission registers: CAN_TMIx, CAN_TMPx, CAN_TMDATA0x and CAN_TMDATA1x. As is shown in [Figure 28-2. Transmission register](#).

Figure 28-2. Transmission register

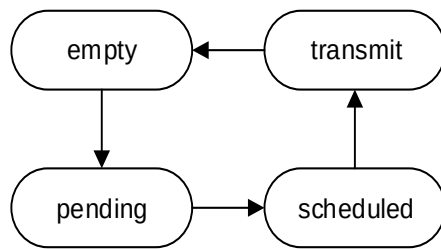


Transmit mailbox state

A transmit mailbox can be used when it is free (**empty state**). If the mailbox is filled with data, set TEN bit in CAN_TMIx register to prepare for starting the transmission (**pending state**). If more than one mailbox is in the pending state, they need scheduling the transmission (**scheduled state**). A mailbox with highest priority enters into transmit state and starts transmitting the message (**transmit state**). After the message has been sent, the mailbox is

free (**empty state**). As is shown in [Figure 28-3. State of transmit mailbox](#).

Figure 28-3. State of transmit mailbox



Transmit status and error

The CAN_TSTAT register includes the transmit status and error bits: MTF, MTFNERR, MAL, MTE.

- MTF: mailbox transmit finished. Typically, MTF is set when the frame in the transmit mailbox has been sent.
- MTFNERR: mailbox transmit finished with no error. MTFNERR is set when the frame in the transmit mailbox has been sent without any error.
- MAL: mailbox arbitration lost. MAL is set when the frame transmission is failed due to the arbitration lost.
- MTE: mailbox transmit error. MTE is set when the frame transmission is failed due to the error detected on the CAN bus.

Steps of sending a frame

To send a frame through the CAN:

Step 1: Select one free transmit mailbox.

Step 2: Configure four transmission registers with the application's acquirement.

Step 3: Set TEN bit in CAN_TMLx register.

Step 4: Check the transmit status. Typically, MTF and MTFNERR are set if transmission is successful.

Transmission options

Abort

MST bit in CAN_TSTAT register can abort the transmission.

If the transmit mailbox's status is **pending** or **scheduled**, the abort of transmission can be done immediately.

In the **transmit** state, the abort of transmission does not take effect immediately until the transmission is finished. In case that the transmission is successful, the MTFNERR and MTF in CAN_TSTAT are set and state changes to be **empty**. In case that the transmission is failed,

the state changes to be **scheduled** and then the abort of transmission can be done immediately.

Priority

When more than one transmit mailbox is pending, the transmission order is given by the TFO bit in CAN_CTL register.

In case that TFO is 1, the three transmit mailboxes work first-in first-out (FIFO).

In case that TFO is 0, the transmit mailbox with lowest identifier has the highest priority of transmission. If the identifiers are equal, the lower mailbox number will be scheduled firstly.

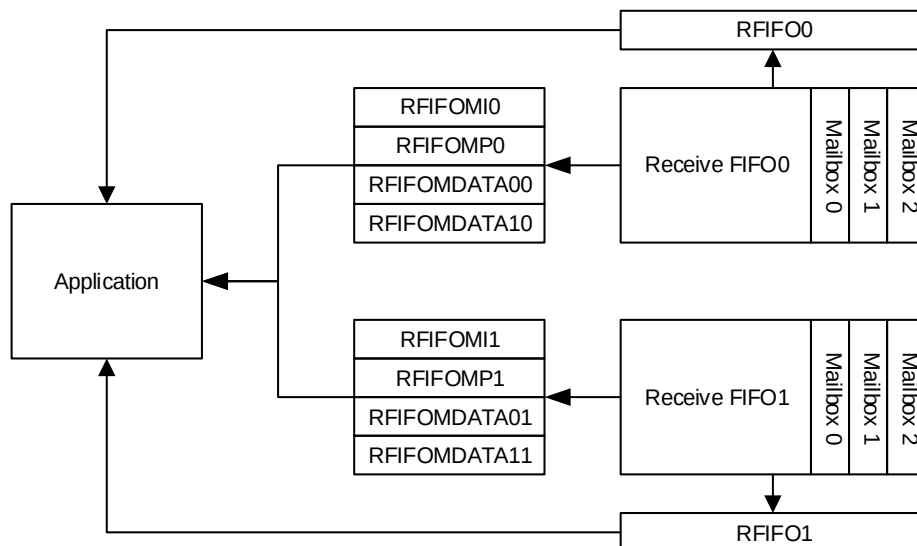
28.3.4 Data reception

Reception register

Two Rx FIFOs are used for the application. Rx FIFOs are managed by five registers: CAN_RFIFOx, CAN_RFIFOMIx, CAN_RFIFOMPx, CAN_RFIFOMDATA0x and CAN_RFIFOMDATA1x. FIFO's status and operation can be handled by CAN_RFIFOx register. Reception frame data can be achieved through the registers: CAN_RFIFOMIx, CAN_RFIFOMPx, CAN_RFIFOMDATA0x and CAN_RFIFOMDATA1x.

Each FIFO consists of three receive mailboxes. As is shown in [Figure 28-4. Reception register](#).

Figure 28-4. Reception register



Rx FIFO

Rx FIFO has three mailboxes. The reception frames are stored in the mailbox according to the arriving sequence. First arrived frame can be accessed by application firstly.

The number of frames in the Rx FIFO and the status can be accessed by the register CAN_RFIFO0 and CAN_RFIFO1.

If at least one frame has been stored in the Rx FIFO0, the frame data is stored in the CAN_RFIFOMI0, CAN_RFIFOMP0, CAN_RFIFOMDATA00 and CAN_RFIFOMDATA10 registers. After reading the current frame, set RFD bit in CAN_RFIFO0 to release a frame in the Rx FIFO and the software can read the next frame.

Rx FIFO status

RFL (Rx FIFO length) bits in CAN_RFIFOx register is 0 when no frame is stored in the Rx FIFO and it is 3 when FIFOx is full.

When RFF bit in CAN_RFIFOx register is set, it indicates FIFOx is full, at this time, RFL is 3.

When a new frame arrives after the FIFO has held three frames, the RFO bit in CAN_RFIFOx register will be set, and it indicates FIFOx is overrun. If the RFOD bit in CAN_CTL register is set, the new frame is discarded. If the RFOD bit in CAN_CTL register is reset, the new frame is stored into the Rx FIFO and the last frame in the Rx FIFO is discarded.

Steps of receiving a message

Step 1: Check the number of frames in the Rx FIFO.

Step 2: Read CAN_RFIFOMIx, CAN_RFIFOMPx, CAN_RFIFOMDATA0x and CAN_RFIFOMDATA1x.

Step 3: Set the RFD bit in CAN_RFIFOx register.

28.3.5 Filtering function

The CAN receives frames from the CAN bus. If the frame passes the filter, it is stored in the Rx FIFOs. Otherwise, the frame will be discarded without intervention by the software.

The identifier of frame is used for the matching of the filter.

Scale

In GD32M531, the filter consists of 28 banks: bank0 to bank27. Each bank has two 32-bit registers: CAN_FxDATA0 and CAN_FxDATA1.

Each filter bank can be configured to 32-bit or 16-bit.

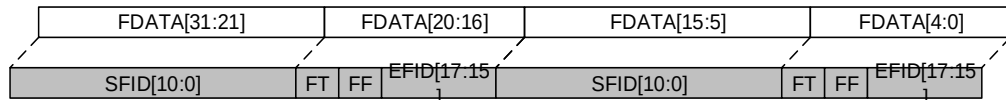
32-bit: SFID[10:0], EFID[17:0], FF and FT bits. As is shown in [Figure 28-5. 32-bit filter](#).

Figure 28-5. 32-bit filter



16-bit: SFID [10:0], FT, FF and EFID[17:15] bits. As is shown in [Figure 28-6. 16-bit filter](#).

Figure 28-6. 16-bit filter



Mask mode

For the Identifier of a data frame to be filtered, the mask mode is used to specify which bits must be the same as the preset Identifier and which bits need not be judged. 32-bit mask mode example is shown in [Figure 28-7. 32-bit mask mode filter](#).

Figure 28-7. 32-bit mask mode filter

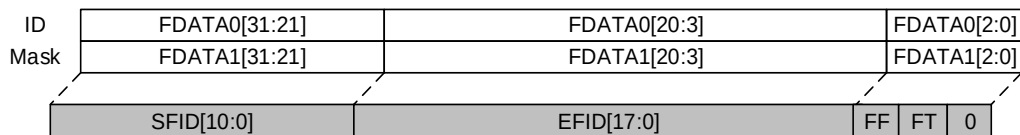
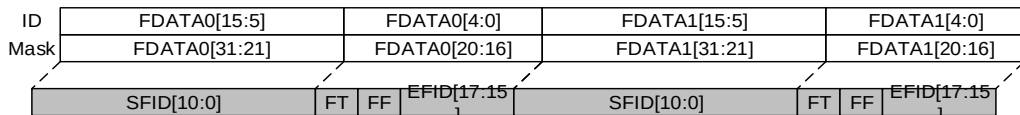


Figure 28-8. 16-bit mask mode filter



List mode

The filter consists of frame identifiers. The filter can determine whether a frame will be discarded or not. When one frame arrived, the filter will check which member can match the identifier of the frame.

32-bit list mode example is shown in [Figure 28-9. 32-bit list mode filter](#).

Figure 28-9. 32-bit list mode filter

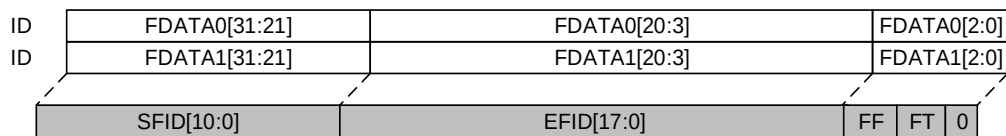
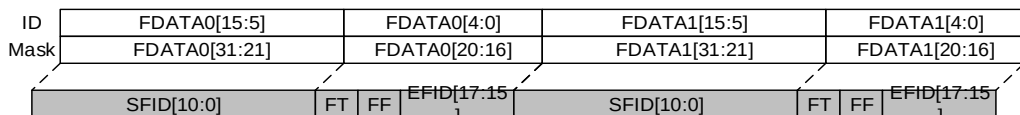


Figure 28-10. 16-bit list mode filter



Filter number

Filter consists of some filter bank. According to the mode and the scale of each of the filter banks, filter has different effects.

For example, there are two filter banks. Bank0 is configured as 32-bit mask mode. Bank1 is configured as 32-bit list mode. The filter number is shown in [Table 28-1. 32-bit filter number](#).

Table 28-1. 32-bit filter number

Filter bank	Filter data register	Filter number
0	F0DATA0-32bit-ID	0
	F0DATA1-32bit-Mask	
1	F1DATA0-32bit-ID	1
	F1DATA1-32bit-ID	2

Associated FIFO

28 banks can be associated with FIFO0 or FIFO1. If the bank is associated with FIFO0, the frames passed the bank will be stored in the FIFO0.

Active

The filter bank needs to be activated if the bank is to be used, otherwise, the filter bank should be left deactivated.

Filtering index

Each filter number corresponds to a filtering rule. When the frame which is associated with a filter number N passes the filters, the filter index is N. It stores in the FI bits in CAN_RFIFOMPx.

Filter bank has filter index once it is associated with the FIFO no matter whether the bank is active or not.

The example about filtering index is shown in [Table 28-2. Filtering index](#).

Table 28-2. Filtering index

Filter bank	FIFO0	Active	Filter number	Filter bank	FIFO1	Active	Filter number
0	F0DATA0-32bits-ID	Yes	0	2	F2DATA0[15:0]-16bits-ID	Yes	0
	F0DATA1-32bits-Mask				F2DATA0[31:16]-16bits-Mask		
1	F1DATA0-32bits-ID	Yes	1		F2DATA1[15:0]-16bits-ID		1
	F1DATA1-32bits-ID		2		F2DATA1[31:16]-16bits-Mask		
3	F3DATA0[15:0]-16bits-ID	No	3	4	F4DATA0-32bits-ID	No	2
	F3DATA0[31:16]-16bits-Mask				F4DATA1-32bits-Mask		
	F3DATA1[15:0]-16bits-ID		4	5	F5DATA0-32bits-ID	No	3
	F3DATA1[31:16]-16bits-Mask				F5DATA1-32bits-ID		4
7	F7DATA0[15:0]-16bits-ID	No	5	6	F6DATA0[15:0]-16bits-ID	Yes	5
	F7DATA0[31:16]-16bits- ID		6		F6DATA0[31:16]-16bits- ID		6
	F7DATA1[15:0]-16bits-ID		7		F6DATA1[15:0]-16bits-ID		7
	F7DATA1[31:16]-16bits- ID		8		F6DATA1[31:16]-16bits- ID		8
8	F8DATA0[15:0]-16bits-ID	Yes	9	10	F10DATA0[15:0]-16bits-ID	No	9
	F8DATA0[31:16]-16bits- ID		10		F10DATA0[31:16]-16bits-Mask		
	F8DATA1[15:0]-16bits-ID		11		F10DATA1[15:0]-16bits-ID		10
	F8DATA1[31:16]-16bits- ID		12		F10DATA1[31:16]-16bits-Mask		
9	F9DATA0[15:0]-16bits-ID	Yes	13	11	F11DATA0[15:0]-16bits-ID	No	11
	F9DATA0[31:16]-16bits-Mask				F11DATA0[31:16]-16bits- ID		12
	F9DATA1[15:0]-16bits-ID		14		F11DATA1[15:0]-16bits-ID		13
	F9DATA1[31:16]-16bits-Mask				F11DATA1[31:16]-16bits- ID		14
12	F12DATA0-32bits-ID	Yes	15	13	F13DATA0-32bits-ID	Yes	15
	F12DATA1-32bits-Mask				F13DATA1-32bits- ID		16

Priority

The filters have the priority rules:

- 32-bits mode is higher than 16-bits mode.
- List mode is higher than mask mode.
- Smaller filter number has the higher priority.

28.3.6 Time-triggered communication

The time-triggered CAN protocol is a higher layer protocol on top of the CAN data link layer. Time-triggered communication means that activities are triggered by the elapsing of time segments. In a time-triggered communication system, all time points of message transmission are pre-defined.

In this mode, an internal 16-bit counter starts working, incrementing by 1 at each CAN bit time. This internal counter provides time stamps for sending and receiving data, stored in registers CAN_RFIFOMPx and CAN_TMPx.

The automatic retransmission is disabled in the time-triggered CAN communication.

28.3.7 Communication parameters

Automatic retransmission forbid mode

In time-triggered communication mode, the requirement for automatic retransmission must be disabled and can be met by setting ARD position 1 of the CAN_CTL register.

In this mode, the data is sent only once, and if the transmission fails due to arbitration failure or bus error, the CAN bus controller does not automatically resend the data as usual.

At the end of sending, the MTF bit of register CAN_TSTAT is hardware set to 1, and the sending status information can be obtained via MTFNERR, MAL, and MTE.

Bit time

On the bit-level, the CAN protocol uses synchronous bit transmission. This not only enhances the transmitting capacity but also requires a sophisticated method of bit synchronization. While bit synchronization in a character-oriented transmission (asynchronous) is performed upon the reception which the start bit is available with each character, the synchronous transmission protocol just need one start bit available at the beginning of a frame. To ensure that the receiver correctly reads the messages, resynchronization is required. Phase buffer segments' sample point of the front-end and back-end should be inserted a bit interval.

The CAN protocol regulates bus access by bit-wise arbitration. The signal propagated from sender to receiver and back to the sender must be completed within one bit-time. For synchronization, in addition to the phase buffer segments, a propagation delay segment is needed. The propagation delay segment is regarded as signal delays caused by transmitting and receiving nodes in the process of the signal propagation on the bus.

The normal bit time from the CAN protocol has three segments as follows:

Synchronization segment (SYNC_SEG): a bit change is expected to occur within this time segment. It has a fixed length of one time quantum ($1 \times t_q$).

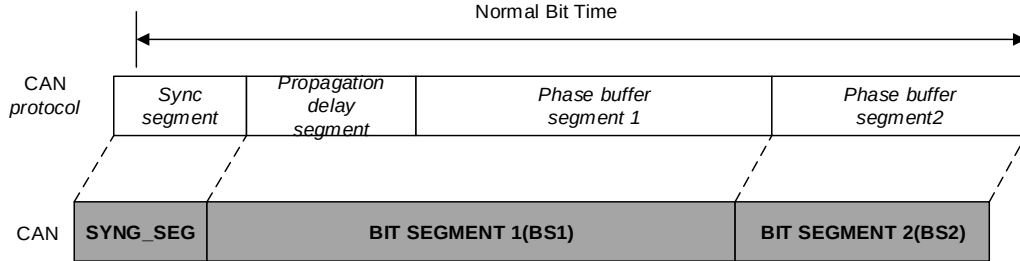
Bit segment 1 (BS1): It defines the location of the sample point. It includes the Propagation delay segment and Phase buffer segment 1 in the CAN standard. Its duration is programmable from 1 to 16 time quanta but it may be automatically lengthened to compensate for positive phase drifts due to different frequency of the various nodes of the network.

Bit segment 2 (BS2): It defines the location of the transmit point. It represents the Phase buffer segment 2 in the CAN standard. Its duration is programmable from 1 to 8 time quanta

but it may also be automatically shortened to compensate for negative phase drifts.

The bit time is shown as in the [Figure 28-11. The bit time](#).

Figure 28-11. The bit time



The resynchronization Jump Width (SJW): it can be lengthened or shortened to compensate for the Synchronization error of the CAN network node. It is programmable from 1 to 4 time quanta.

The valid edge transition is defined as the first transition from a recessive bit to a dominant bit within a bit time, provided that the bus state detected at the previous sample point was recessive.

If a valid edge is detected in BS1, not in SYNC_SEG, BS1 is added up to SJW maximumly, so that the sample point is delayed.

Conversely, if a valid edge is detected in BS2, not in SYNC_SEG, BS2 is cut down to SJW at most, so that the transmit point is moved earlier.

Note: For detailed explanations of hard synchronization and resynchronization, please refer to the ISO 11898 standard.

Baud rate

The clock of the CAN derives from the APB1 bus. The CAN calculates its baud rate as follow:

$$\text{BaudRate} = \frac{1}{\text{Normal Bit Time}} \quad (22-1)$$

$$\text{Normal Bit Time} = t_{\text{SYNC\_SEG}} + t_{\text{BS1}} + t_{\text{BS2}} \quad (22-2)$$

with:

$$t_{\text{SYNC\_SEG}} = 1 \times t_q \quad (22-3)$$

$$t_{\text{BS1}} = (1 + \text{BT.BS1}) \times t_q \quad (22-4)$$

$$t_{\text{BS2}} = (1 + \text{BT.BS2}) \times t_q \quad (22-5)$$

$$t_q = (1 + \text{BT.BAUDPSC}) \times t_{\text{PCLK2}} \quad (22-6)$$

28.3.8 Error flags

The state of CAN bus can be reflected by Transmit Error Counter (TECNT) and Receive Error Counter (RECNT) of CAN_ERR register. The value can be increased or decreased by the hardware according to the error, and the software can judge the stability of the CAN network by these values. For details on incorrect counting, refer to the CAN protocol section.

By using the CAN_INTEN register (ERRIE bit, etc.), the software can control the interrupt generation when error is detected.

Bus-Off recovery

The CAN controller is in Bus-Off state when TECNT is over than 255. In This state, BOERR bit is set in CAN_ERR register, and no longer able to transmit and receive messages.

According to the ABOR configuration in register CAN_CTL, there are two ways to recover from Bus-Off (to an error active state). Both of these methods require the CAN bus controller in the Bus-Off state to detect the Bus-Off recovery sequence defined by CAN protocol (when CAN_RX detects 128 consecutive 11-bit recessive bits) before automatic recovery.

If ABOR is set, it will be automatically recovered when a Bus-Off recovery sequence is detected.

If ABOR is cleared, CAN controller must be configured to enter initialization mode by setting IWMOD bit in CAN_CTL register, then exit and enter normal mode. After this operation, it will recover when the recovering sequence is detected.

28.3.9 CAN interrupts

The CAN bus controller occupies 4 interrupt vectors, which are controlled by the register CAN_INTEN.

The interrupt sources can be classified as:

- Transmit interrupt
- FIFO0 interrupt
- FIFO1 interrupt
- Error and status change interrupt

Transmit interrupt

The transmit interrupt can be generated by any of the following conditions and TMEIE bit in CAN_INTEN register will be set:

- TX mailbox 0 transmit finished: MTF0 bit in the CAN_TSTAT register is set.
- TX mailbox 1 transmit finished: MTF1 bit in the CAN_TSTAT register is set.
- TX mailbox 2 transmit finished: MTF2 bit in the CAN_TSTAT register is set.

Receive FIFO0 interrupt

The Rx FIFO0 interrupt can be generated by the following conditions:

- Rx FIFO0 not empty: RFL0 bits in the CAN_RFIFO0 register are not '00' and RFNEIE0 in CAN_INTEN register is set.
- Rx FIFO0 full: RFF0 bit in the CAN_RFIFO0 register is set and RFFIE0 in CAN_INTEN register is set.
- Rx FIFO0 overrun: RFO0 bit in the CAN_RFIFO0 register is set and RFOIE0 in CAN_INTEN register is set.

Rx FIFO1 interrupt

The Rx FIFO1 interrupt can be generated by the following conditions:

- Rx FIFO1 not empty: RFL1 bits in the CAN_RFIFO1 register are not '00' and RFNEIE1 in CAN_INTEN register is set.
- Rx FIFO1 full: RFF1 bit in the CAN_RFIFO1 register is set and RFFIE1 in CAN_INTEN register is set.
- Rx FIFO1 overrun: RFO1 bit in the CAN_RFIFO1 register is set and RFOIE1 in CAN_INTEN register is set.

Error and working mode change interrupt

The error and working mode change interrupt can be generated by the following conditions:

- Error: ERRIF bit in the CAN_STAT register and ERRIE bit in the CAN_INTEN register are set. Refer to ERRIF description in the CAN_STAT register.
- Wakeup: WUIF bit in the CAN_STAT register is set and WIE bit in the CAN_INTEN register is set.
- Enter sleep working mode: SLPIF bit in the CAN_STAT register is set and SLPWIE bit in the CAN_INTEN register is set.

The CAN bus controller interrupt conditions can refer to [Table 28-3. CAN Event / Interrupt flags](#).

Table 28-3. CAN Event / Interrupt flags

Interrupt event	Interrupt / Event flag	Enable control bit
Transmit interrupt	Mailbox 0 transmit finished flag (MTF0)	TMEIE
	Mailbox 1 transmit finished flag (MTF1)	
	Mailbox 2 transmit finished flag (MTF2)	
FIFO0 interrupt	Rx FIFO0 length (RFL0[1:0])	RFNEIE0
	Rx FIFO0 full (RFF0)	RFFIE0
	Rx FIFO0 overfull (RFO0)	RFOIE0
FIFO1 interrupt	Rx FIFO1 length (RFL1[1:0])	RFNEIE1
	Rx FIFO1 full (RFF1)	RFFIE1
	Rx FIFO1 overfull (RFO1)	RFOIE1

Interrupt event	Interrupt / Event flag		Enable control bit	
EWMC interrupt	Warning error (WERR)	Error interrupt flag (ERRIF)	WERRIE	ERRIE
	Passive error (PERR)		PERRIE	
	Bus-Off error (BOERR)		BOIE	
	Error number (1<= ERRN[2:0] <= 6)		ERRNIE	
	Status change interrupt flag of waking up from sleep working mode (WUIF)		WIE	
	Status change interrupt flag of entering sleep working mode (SLPIF)		SLPWIE	

28.4 CAN registers

CAN base address: 0x4001 A000

28.4.1 Control register (CAN_CTL)

Address offset: 0x00

Reset value: 0x0001 0702

This register has to be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															DFZ
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SWRST	Reserved					FIIDE	BSPMOD	BSSSEL	TTC	ABOR	AWU	ARD	RFOD	TFO	SLPWMO D
rs						rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:17	Reserved	Must be kept at reset value.
16	DFZ	Debug freeze If the CANx_HOLD in DBG_CTL0 register is set, this bit defines the CAN controller is in debug freezing mode or normal working mode. If the CANx_HOLD in DBG_CTL0 register is cleared, this bit takes no effect. 0: CAN reception and transmission work normal even during debug 1: CAN reception and transmission stop working during debug
15	SWRST	Software reset 0: No effect 1: Reset CAN to enter sleep working mode. This bit is automatically reset to 0.
14:11	Reserved	Must be kept at reset value.
10	FIIDE	Frame Inter to identifier Enable After the third bit of the frame interval receives the dominant bit, the first bit of the frame ID is directly sent, skipping sending SOF. 0: frame Inter to identifier disable 1: frame Inter to identifier enable
9	BSPMOD	Sample Bit mode 0: One sample for the received bit 1: Three sample for the received bit
8	BSSSEL	Bit sampling synchronization select The synchronization of two cycles will make the signal sample more accurate, and

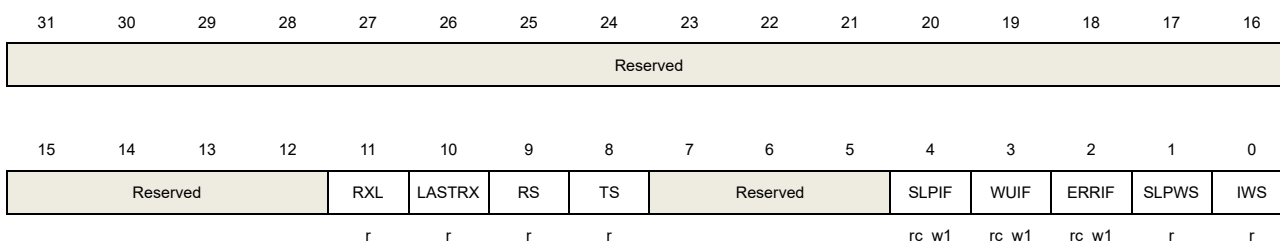
		the synchronization of one cycle can reduce the time of signal sample. 0: one cycle synchronization for CAN bus sampling 1: two cycles synchronization for CAN bus sampling
7	TTC	Time-triggered communication 0: Disable time-triggered communication 1: Enable time-triggered communication
6	ABOR	Automatic Bus-Off recovery 0: The Bus-Off state is left manually by software 1: The Bus-Off state is left automatically by hardware
5	AWU	Automatic wakeup If this bit is set, the CAN leaves sleep working mode when CAN bus activity is detected, and SLPWMOD bit in CAN_CTL register will be cleared automatically. 0: The sleeping working mode is left manually by software 1: The sleeping working mode is left automatically by hardware
4	ARD	Automatic retransmission disable 0: Enable automatic retransmission 1: Disable automatic retransmission
3	RFOD	Rx FIFO overwrite disable 0: Enable Rx FIFO overwrite when Rx FIFO is full and overwrite the FIFO with the incoming frame 1: Disable Rx FIFO overwrite when Rx FIFO is full and discard the incoming frame
2	TFO	Tx FIFO order 0: Order with the identifier of the frame (the smaller identifier has higher priority) 1: Order with first-in and first-out
1	SLPWMOD	Sleep working mode If this bit is set by software, the CAN enters sleep working mode after current transmission or reception is completed. This bit can be cleared by software or hardware. If AWU bit in CAN_CTL register is set, this bit is cleared by hardware when CAN bus activity is detected. 0: Disable sleep working mode 1: Enable sleep working mode
0	IWMOD	Initial working mode 0: Disable initial working mode 1: Enable initial working mode

28.4.2 Status register (CAN_STAT)

Address offset: 0x04

Reset value: 0x0000 0C02

This register has to be accessed by word(32-bit).



Bits	Fields	Descriptions
31:12	Reserved	Must be kept at reset value.
11	RXL	RX level
10	LASTRX	Last sample value of RX pin
9	RS	Receiving state 0: CAN is not working in the receiving state 1: CAN is working in the receiving state
8	TS	Transmitting state 0: CAN is not working in the transmitting state 1: CAN is working in the transmitting state
7:5	Reserved	Must be kept at reset value.
4	SLPIF	Status change interrupt flag of entering sleep working mode This bit is set by hardware when entering sleep working mode, and cleared by hardware when the CAN is not in sleep working mode. This bit can also be cleared by software when writting 1 to this bit. 0: CAN is not in the sleep working mode 1: CAN is in the sleep working mode
3	WUIF	Status change interrupt flag of waking up from sleep working mode This bit is set when CAN bus activity event is detected in sleep working mode. This bit can be cleared by software when writting 1 to this bit. 0: Wakeup event is not coming 1: Wakeup event is coming
2	ERRIF	Error interrupt flag This bit is set by the following events. The BOERR bit in CAN_ERR register is set and BOIE bit in CAN_INTEN register is set. Or the PERR bit in CAN_ERR register is set and PERRIE bit in CAN_INTEN register is set. Or the WERR bit in CAN_ERR register is set and WERRIE bit in CAN_INTEN register is set. Or the ERRN bits in CAN_ERR register are set to 1 to 6 (not 0 and not 7) and ERRNIE in CAN_INTEN register is set. This bit is cleared by software when writting 1 to this bit. 0: No error interrupt event

1: Any error interrupt event has happened

1 SLPWS

Sleep working state

This bit is set by hardware when the CAN enters sleep working mode after setting SLPWMOD bit in CAN_CTL register. If the CAN leaves normal working mode to sleep working mode, it must wait the current frame transmission or reception to be completed. This bit is cleared by hardware when the CAN leaves sleep working mode. Clear SLPWMOD bit in CAN_CTL register or automatically detect the CAN bus activity when AWU bit is set in CAN_CTL register. If leaving sleep working mode to normal working mode, this bit will be cleared after receiving 11 consecutive recessive bits from the CAN bus.

0: CAN is not in the state of sleep working mode

1: CAN is in the state of sleep working mode

0 IWS

Initial working state

This bit is set by hardware when the CAN enters initial working mode after setting IWMOD bit in CAN_CTL register. If the CAN leaves normal working mode to initial working mode, it must wait the current frame transmission or reception to be completed. This bit is cleared by hardware when the CAN leaves initial working mode after clearing IWMOD bit in CAN_CTL register. If leaving initial working mode to normal working mode, this bit will be cleared after receiving 11 consecutive recessive bits from the CAN bus.

0: CAN is not in the state of initial working mode

1: CAN is in the state of initial working mode

28.4.3 Transmit status register (CAN_TSTAT)

Address offset: 0x08

Reset value: 0x1C00 0000

This register has to be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TMLS2	TMLS1	TMLS0	TME2	TME1	TME0	NUM[1:0]		MST2	Reserved			MTE2	MAL2	MTFNER R2	MTF2
r	r	r	r	r	r	r		rs				rc_w1	rc_w1	rc_w1	rc_w1
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MST1	Reserved			MTE1	MAL1	MTFNER R1	MTF1	MST0	Reserved			MTE0	MAL0	MTFNER R0	MTF0
rs				rc_w1	rc_w1	rc_w1	rc_w1	rs				rc_w1	rc_w1	rc_w1	rc_w1

Bits	Fields	Descriptions
31	TMLS2	Transmit mailbox 2 last sending in Tx FIFO This bit is set by hardware when transmit mailbox 2 has the last sending order in the Tx FIFO with at least two frames pending.

30	TMLS1	Transmit mailbox 1 last sending in Tx FIFO This bit is set by hardware when transmit mailbox 1 has the last sending order in the Tx FIFO with at least two frames pending.
29	TMLS0	Transmit mailbox 0 last sending in Tx FIFO This bit is set by hardware when transmit mailbox 0 has the last sending order in the Tx FIFO with at least two frames pending.
28	TME2	Transmit mailbox 2 empty 0: Transmit mailbox 2 not empty 1: Transmit mailbox 2 empty
27	TME1	Transmit mailbox 1 empty 0: Transmit mailbox 1 not empty 1: Transmit mailbox 1 empty
26	TME0	Transmit mailbox 0 empty 0: Transmit mailbox 0 not empty 1: Transmit mailbox 0 empty
25:24	NUM[1:0]	These bits are the number of the Tx FIFO mailbox in which the frame will be transmitted if at least one mailbox is empty. These bits are the number of the Tx FIFO mailbox in which the frame will be transmitted at last if all mailboxes are full.
23	MST2	Mailbox 2 stop transmitting This bit is set by the software to stop mailbox 2 transmitting. This bit is reset by the hardware while the mailbox 2 is empty.
22:20	Reserved	Must be kept at reset value.
19	MTE2	Mailbox 2 transmit error This bit is set by hardware when the transmit error occurs. This bit is reset by writing 1 to this bit or MTF2 bit in CAN_TSTAT register. This bit is reset by hardware when next transmit starts.
18	MAL2	Mailbox 2 arbitration lost This bit is set when the arbitration lost occurs. This bit is reset by writing 1 to this bit or MTF2 bit in CAN_TSTAT register. This bit is reset by hardware when next transmit starts.
17	MTFNERR2	Mailbox 2 transmit finished with no error This bit is set when the transmission finishes and no error occurs. This bit is reset by writing 1 to this bit or MTF2 bit in CAN_TSTAT register. This bit is reset by hardware when the transmission finishes with error. 0: Mailbox 2 transmit finished with error 1: Mailbox 2 transmit finished with no error
16	MTF2	Mailbox 2 transmit finished

		This bit is set by hardware when the transmission finishes or aborts. This bit is reset by writing 1 to this bit or TEN bit in CAN_TMI2 is 1. 0: Mailbox 2 transmit is progressing 1: Mailbox 2 transmit finished
15	MST1	Mailbox 1 stop transmitting This bit is set by software to stop mailbox 1 transmitting. This bit is reset by hardware when the mailbox 1 is empty.
14:12	Reserved	Must be kept at reset value.
11	MTE1	Mailbox 1 transmit error This bit is set by hardware when the transmit error occurs. This bit is reset by writing 1 to this bit or MTF1 bit in CAN_TSTAT register. This bit is reset by hardware when next transmit starts.
10	MAL1	Mailbox 1 arbitration lost This bit is set when the arbitration lost occurs. This bit is reset by writing 1 to this bit or MTF1 bit in CAN_TSTAT register. This bit is reset by hardware when next transmit starts.
9	MTFNERR1	Mailbox 1 transmit finished with no error This bit is set when the transmission finishes and no error occurs. This bit is reset by writing 1 to this bit or MTF1 bit in CAN_TSTAT register. This bit is reset by hardware when the transmission finishes with error. 0: Mailbox 1 transmit finished with error 1: Mailbox 1 transmit finished with no error
8	MTF1	Mailbox 1 transmit finished This bit is set by hardware when the transmission finishes or aborts. This bit is reset by writing 1 to this bit or TEN bit in CAN_TMI1 is 1. 0: Mailbox 1 transmit is progressing 1: Mailbox 1 transmit finished
7	MST0	Mailbox 0 stop transmitting This bit is set by the software to stop mailbox 0 transmitting. This bit is reset by the hardware when the mailbox 0 is empty.
6:4	Reserved	Must be kept at reset value.
3	MTE0	Mailbox 0 transmit error This bit is set by hardware when the transmit error occurs. This bit is reset by writing 1 to this bit or MTF0 bit in CAN_TSTAT register. This bit is reset by hardware when next transmit starts.
2	MAL0	Mailbox 0 arbitration lost This bit is set when the arbitration lost occurs. This bit is reset by writing 1 to this bit or MTF0 bit in CAN_TSTAT register. This bit is reset by hardware when next

transmit starts.

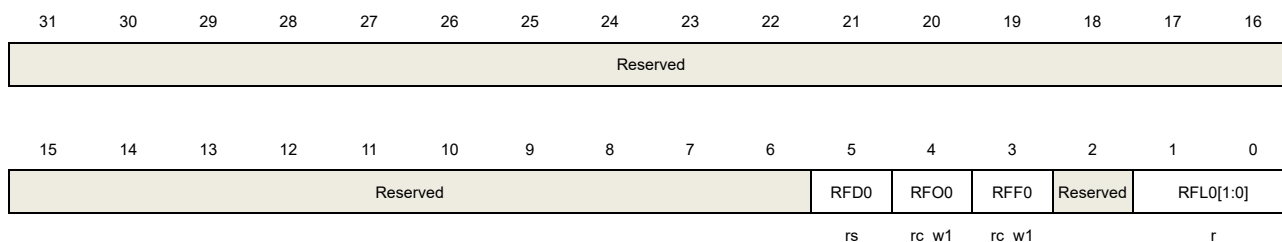
1	MTFNERR0	Mailbox 0 transmit finished with no error This bit is set when the transmission finishes and no error occurs. This bit is reset by writting 1 to this bit or MTF0 bit in CAN_TSTAT register. This bit is reset by hardware when the transmission finishes with error. 0: Mailbox 0 transmit finished with error 1: Mailbox 0 transmit finished with no error
0	MTF0	Mailbox 0 transmit finished This bit is set by hardware when the transmission finishes or aborts. This bit is reset by writting 1 to this bit or TEN bit in CAN_TMI0 is 1. 0: Mailbox 0 transmit is progressing 1: Mailbox 0 transmit finished

28.4.4 Receive message FIFO0 register (CAN_RFIFO0)

Address offset: 0x0C

Reset value: 0x0000 0000

This register has to be accessed by word(32-bit).



Bits	Fields	Descriptions
31:6	Reserved	Must be kept at reset value.
5	RFD0	Rx FIFO0 dequeue This bit is set by software to start dequeuing a frame from Rx FIFO0. This bit is reset by hardware when the dequeuing is done.
4	RFO0	Rx FIFO0 overfull This bit is set by hardware when Rx FIFO0 is overfull and reset by software when writting 1 to this bit. 0: The Rx FIFO0 is not overfull 1: The Rx FIFO0 is overfull
3	RFF0	Rx FIFO0 full This bit is set by hardware when Rx FIFO0 is full and reset by software when writting 1 to this bit. 0: The Rx FIFO0 is not full 1: The Rx FIFO0 is full

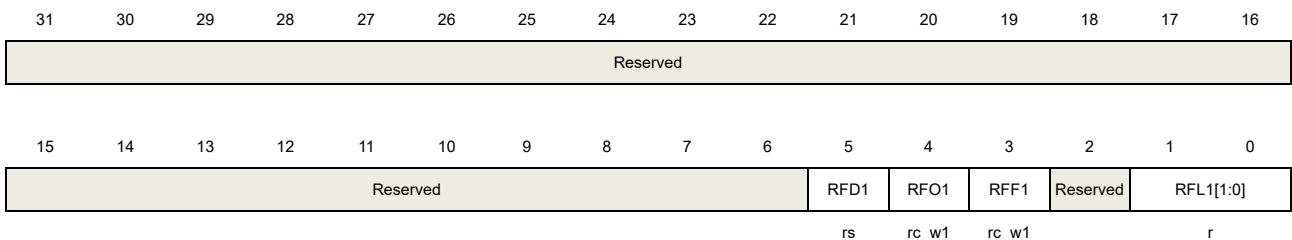
2	Reserved	Must be kept at reset value.
1:0	RFL0[1:0]	Rx FIFO0 length These bits are the length of the Rx FIFO0.

28.4.5 Receive message FIFO1 register (CAN_RFIFO1)

Address offset: 0x10

Reset value: 0x0000 0000

This register has to be accessed by word(32-bit).



Bits	Fields	Descriptions
31:6	Reserved	Must be kept at reset value.
5	RFD1	Rx FIFO1 dequeue This bit is set by software to start dequeuing a frame from Rx FIFO1. This bit is reset by hardware when the dequeuing is done.
4	RFO1	Rx FIFO1 overfull This bit is set by hardware when Rx FIFO1 is overfull and reset by writting 1 to this bit. 0: The Rx FIFO1 is not overfull 1: The Rx FIFO1 is overfull
3	RFF1	Rx FIFO1 full This bit is set by hardware when Rx FIFO1 is full and reset by writting 1 to this bit. 0: The Rx FIFO1 is not full 1: The Rx FIFO1 is full
2	Reserved	Must be kept at reset value.
1:0	RFL1[1:0]	Rx FIFO1 length These bits are the length of the Rx FIFO1.

28.4.6 Interrupt enable register (CAN_INTEN)

Address offset: 0x14

Reset value: 0x0000 0000

This register has to be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved														SLPWIE	WIE
														rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ERRIE	Reserved			ERRNIE	BOIE	PERRIE	WERRIE	Reserved	RFOIE1	RFFIE1	RFNEIE1	RFOIE0	RFFIE0	RFNEIE0	TMEIE
rw				rw	rw	rw	rw		rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:18	Reserved	Must be kept at reset value.
17	SLPWIE	Sleep working interrupt enable 0: Sleep working interrupt disabled 1: Sleep working interrupt enabled
16	WIE	Wakeup interrupt enable 0: Wakeup interrupt disabled 1: Wakeup interrupt enabled
15	ERRIE	Error interrupt enable 0: Error interrupt disabled 1: Error interrupt enabled
14:12	Reserved	Must be kept at reset value.
11	ERRNIE	Error number interrupt enable 0: Error number interrupt disabled 1: Error number interrupt enabled
10	BOIE	Bus-Off interrupt enable 0: Bus-Off interrupt disabled 1: Bus-Off interrupt enabled
9	PERRIE	Passive error interrupt enable 0: Passive error interrupt disabled 1: Passive error interrupt enabled
8	WERRIE	Warning error interrupt enable 0: Warning error interrupt disabled 1: Warning error interrupt enabled
7	Reserved	Must be kept at reset value.
6	RFOIE1	Rx FIFO1 overfull interrupt enable 0: Rx FIFO1 overfull interrupt disabled 1: Rx FIFO1 overfull interrupt enabled
5	RFFIE1	Rx FIFO1 full interrupt enable 0: Rx FIFO1 full interrupt disabled

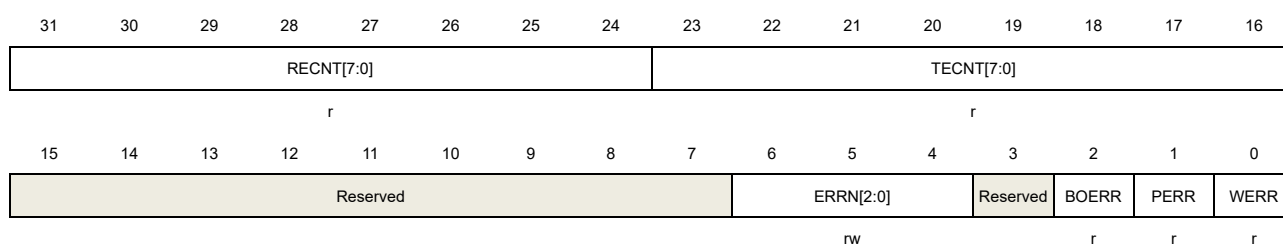
		1: Rx FIFO1 full interrupt enabled
4	RFNEIE1	Rx FIFO1 not empty interrupt enable 0: Rx FIFO1 not empty interrupt disabled 1: Rx FIFO1 not empty interrupt enabled
3	RFOIE0	Rx FIFO0 overfull interrupt enable 0: Rx FIFO0 overfull interrupt disabled 1: Rx FIFO0 overfull interrupt enabled
2	RFFIE0	Rx FIFO0 full interrupt enable 0: Rx FIFO0 full interrupt disabled 1: Rx FIFO0 full interrupt enabled
1	RFNEIE0	Rx FIFO0 not empty interrupt enable 0: Rx FIFO0 not empty interrupt disabled 1: Rx FIFO0 not empty interrupt enabled
0	TMEIE	Transmit mailbox empty interrupt enable 0: Transmit mailbox empty interrupt disabled 1: Transmit mailbox empty interrupt enabled

28.4.7 Error register (CAN_ERR)

Address offset: 0x18

Reset value: 0x0000 0000

This register has to be accessed by word(32-bit).



Bits	Fields	Descriptions
31:24	RECNT[7:0]	Receive error count defined by the CAN standard
23:16	TECNT[7:0]	Transmit error count defined by the CAN standard
15:7	Reserved	Must be kept at reset value.
6:4	ERRN[2:0]	Error number These bits indicate the error status of bit transformation. They are updated by hardware. When the bit transformation is successful, they are equal to 0. 000: No error 001: Stuff error 010: Form error

28.4.8 Bit timing register (CAN_BT)

Reset value: 0x0123 0000

This register has to be accessed by word(32-bit).

Bits	Fields	Descriptions
31	SCMOD	Silent communication mode 0: Silent communication disabled 1: Silent communication enabled
30	LCMOD	Loopback communication mode 0: Loopback communication disabled 1: Loopback communication enabled
29:26	Reserved	Must be kept at reset value.
25:24	SJW[1:0]	Resynchronization jump width Resynchronization jump width time quantum= SJW[1:0]+1
23	Reserved	Must be kept at reset value.

22:20	BS2[2:0]	Bit segment 2 Bit segment 2 time quantum = BS2[2:0]+1
19:16	BS1[3:0]	Bit segment 1 Bit segment 1 time quantum = BS1[3:0]+1
15:10	Reserved	Must be kept at reset value.
9:0	BAUDPSC[9:0]	Baud rate prescaler The CAN baud rate prescaler

28.4.9 Transmit mailbox identifier register (CAN_TMIx) (x = 0...2)

Address offset: 0x180, 0x190, 0x1A0

Reset value: 0xFFFF XXXX (bit0 = 0)

This register has to be accessed by word(32-bit).



Bits	Fields	Descriptions
31:21	SFID[10:0]/EFID[28:18]	The frame identifier SFID[10:0]: Standard format frame identifier EFID[28:18]: Extended format frame identifier
20:16	EFID[17:13]	The frame identifier EFID[17:13]: Extended format frame identifier
15:3	EFID[12:0]	The frame identifier EFID[12:0]: Extended format frame identifier
2	FF	Frame format 0: Standard format frame 1: Extended format frame
1	FT	Frame type 0: Data frame 1: Remote frame
0	TEN	Transmit enable This bit is set by software when one frame will be transmitted and reset by hardware when the transmit mailbox is empty. 0: Transmit disabled

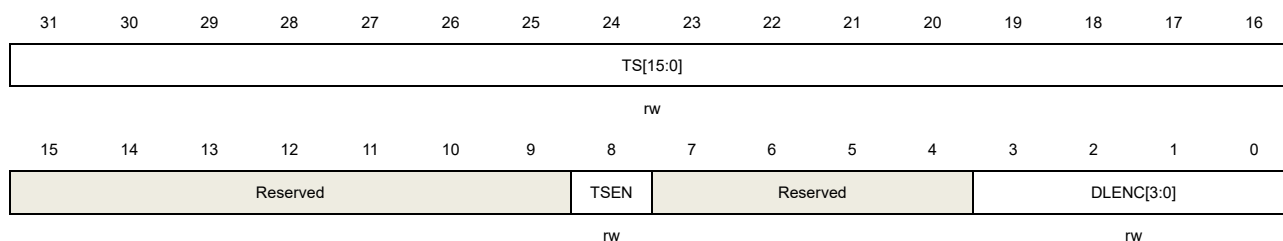
1: Transmit enabled

28.4.10 Transmit mailbox property register (CAN_TMPx) (x=0..2)

Address offset: 0x184, 0x194, 0x1A4

Reset value: 0xFFFF XXXX

This register has to be accessed by word(32-bit).



Bits	Fields	Descriptions
31:16	TS[15:0]	Time stamp The time stamp of frame in transmit mailbox.
15:9	Reserved	Must be kept at reset value.
8	TSEN	Time stamp enable 0: Time stamp disabled 1: Time stamp enabled. The TS[15:0] will be transmitted in the DB6 and DB7 in DL. This bit is available when the TTC bit in CAN_CTL is set.
7:4	Reserved	Must be kept at reset value.
3:0	DLEN[3:0]	Data length code DLEN[3:0] is the number of bytes in a frame.

28.4.11 Transmit mailbox data0 register (CAN_TMDATA0x) (x = 0...2)

Address offset: 0x188, 0x198, 0x1A8

Reset value: 0xFFFF XXXX

This register has to be accessed by word(32-bit).



Bits	Fields	Descriptions
31:24	DB3[7:0]	Data byte 3
23:16	DB2[7:0]	Data byte 2
15:8	DB1[7:0]	Data byte 1
7:0	DB0[7:0]	Data byte 0

28.4.12 Transmit mailbox data1 register (CAN_TMDATA1x) (x=0..2)

Address offset: 0x18C, 0x19C, 0x1AC

Reset value: 0xFFFF XXXX

This register has to be accessed by word(32-bit).



Bits	Fields	Descriptions
31:24	DB7[7:0]	Data byte 7
23:16	DB6[7:0]	Data byte 6
15:8	DB5[7:0]	Data byte 5
7:0	DB4[7:0]	Data byte 4

28.4.13 Receive FIFO mailbox identifier register (CAN_RFIFOMIx) (x = 0, 1)

Address offset: 0x1B0, 0x1C0

Reset value: 0xFFFF XXXX

This register has to be accessed by word(32-bit).



Bits	Fields	Descriptions
31:21	SFID[10:0]/EFID[28:1	The frame identifier

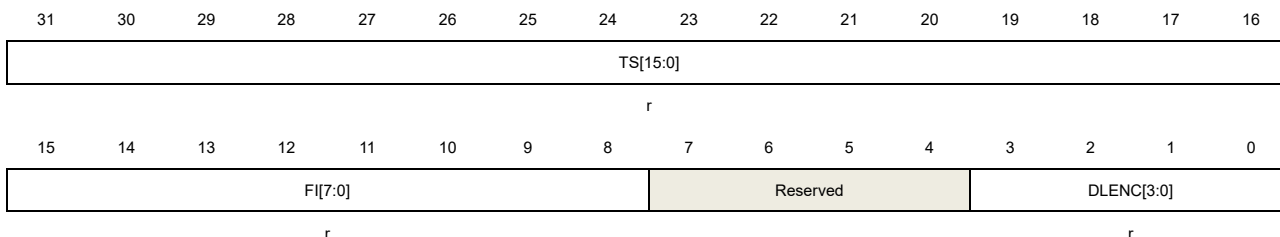
8]		SFID[10:0]: Standard format frame identifier EFID[28:18]: Extended format frame identifier
20:16	EFID[17:13]	The frame identifier EFID[17:13]: Extended format frame identifier
15:3	EFID[12:0]	The frame identifier EFID[12:0]: Extended format frame identifier
2	FF	Frame format 0: Standard format frame 1: Extended format frame
1	FT	Frame type 0: Data frame 1: Remote frame
0	Reserved	Must be kept at reset value.

28.4.14 Receive FIFO mailbox property register (CAN_RFIFOMPx) (x = 0, 1)

Address offset: 0x1B4, 0x1C4

Reset value: 0xFFFF XXXX

This register has to be accessed by word(32-bit).



Bits	Fields	Descriptions
31:16	TS[15:0]	Time stamp The time stamp of frame in transmit mailbox.
15:8	FI[7:0]	Filtering index The index of the filter which the frame passes.
7:4	Reserved	Must be kept at reset value.
3:0	DLENC[3:0]	Data length code DLENC[3:0] is the number of bytes in a frame.

28.4.15 Receive FIFO mailbox data0 register (CAN_RFIFOMDATA0x) (x = 0, 1)

Address offset: 0x1B8, 0x1C8

Reset value: 0xFFFF XXXX

This register has to be accessed by word(32-bit).



Bits	Fields	Descriptions
31:24	DB3[7:0]	Data byte 3
23:16	DB2[7:0]	Data byte 2
15:8	DB1[7:0]	Data byte 1
7:0	DB0[7:0]	Data byte 0

28.4.16 Receive FIFO mailbox data1 register (CAN_RFIFOMDATA1x) (x = 0, 1)

Address offset: 0x1BC, 0x1CC

Reset value: 0xFFFF FFFF

This register has to be accessed by word(32-bit).



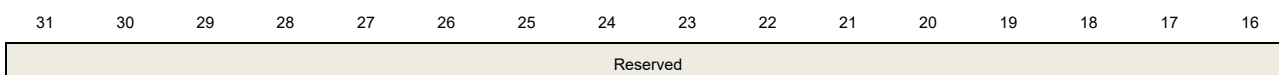
Bits	Fields	Descriptions
31:24	DB7[7:0]	Data byte 7
23:16	DB6[7:0]	Data byte 6
15:8	DB5[7:0]	Data byte 5
7:0	DB4[7:0]	Data byte 4

28.4.17 Filter control register (CAN_FCTL)

Address offset: 0x200

Reset value: 0x2A1C 1C01

This register has to be accessed by word(32-bit).



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

Bits	Fields	Descriptions
31:1	Reserved	Must be kept at reset value.
0	FLD	Filter lock disable 0: Filter lock enabled 1: Filter lock disabled

28.4.18 Filter mode configuration register (CAN_FCFG)

Address offset: 0x204

Reset value: 0x0000 0000

This register has to be accessed by word(32-bit). This register can be modified only when FLD bit in CAN_FCTL register is set.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				FMOD27	FMOD26	FMOD25	FMOD24	FMOD23	FMOD22	FMOD21	FMOD20	FMOD19	FMOD18	FMOD17	FMOD16
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FMOD15	FMOD14	FMOD13	FMOD12	FMOD11	FMOD10	FMOD9	FMOD8	FMOD7	FMOD6	FMOD5	FMOD4	FMOD3	FMOD2	FMOD1	FMOD0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:28	Reserved	Must be kept at reset value.
27:0	FMODEx	Filter mode 0: Filter x with mask mode 1: Filter x with list mode

28.4.19 Filter scale configuration register (CAN_FSCFG)

Address offset: 0x20C

Reset value: 0x0000 0000

This register has to be accessed by word(32-bit). This register can be modified only when FLD bit in CAN_FCTL register is set.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				FS27	FS26	FS25	FS24	FS23	FS22	FS21	FS20	FS19	FS18	FS17	FS16
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FS15	FS14	FS13	FS12	FS11	FS10	FS9	FS8	FS7	FS6	FS5	FS4	FS3	FS2	FS1	FS0

rw rw rw rw rw rw rw rw rw rw rw rw rw rw rw

Bits	Fields	Descriptions
31:28	Reserved	Must be kept at reset value.
27:0	FSx	Filter scale 0: Filter x with 16-bit scale 1: Filter x with 32-bit scale

28.4.20 Filter associated FIFO register (CAN_FAFIFO)

Address offset: 0x214

Reset value: 0x0000 0000

This register has to be accessed by word(32-bit). This register can be modified only when FLD bit in CAN_FCTL register is set.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				FAF27	FAF26	FAF25	FAF24	FAF23	FAF22	FAF21	FAF20	FAF19	FAF18	FAF17	FAF16
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FAF15	FAF14	FAF13	FAF12	FAF11	FAF10	FAF9	FAF8	FAF7	FAF6	FAF5	FAF4	FAF3	FAF2	FAF1	FAF0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:28	Reserved	Must be kept at reset value.
27:0	FAFx	Filter associated FIFO 0: Filter x associated with FIFO0 1: Filter x associated with FIFO1

28.4.21 Filter working register (CAN_FW)

Address offset: 0x21C

Reset value: 0x0000 0000

This register has to be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				FW27	FW26	FW25	FW24	FW23	FW22	FW21	FW20	FW19	FW18	FW17	FW16
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FW15	FW14	FW13	FW12	FW11	FW10	FW9	FW8	FW7	FW6	FW5	FW4	FW3	FW2	FW1	FW0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions

31:28 Reserved Must be kept at reset value.

27:0 FWx Filter working
0: Filter x working disabled
1: Filter x working enabled

28.4.22 Filter x data y register (CAN_FxDATAy) (x = 0...27, y = 0, 1)

Address offset: $0x240 + 8 * x + 4 * y$, (x = 0...27, y = 0, 1)

Reset value: 0xFFFF XXXX

This register has to be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FD31	FD30	FD29	FD28	FD27	FD26	FD25	FD24	FD23	FD22	FD21	FD20	FD19	FD18	FD17	FD16
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FD15	FD14	FD13	FD12	FD11	FD10	FD9	FD8	FD7	FD6	FD5	FD4	FD3	FD2	FD1	FD0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:0	FDx	Filter data Mask mode 0: Mask match disable 1: Mask match enable List mode 0: List identifier bit is 0 1: List identifier bit is 1

29 Appendix

29.1 List of abbreviations used in register

Table 29-1. List of abbreviations used in register

abbreviations for registers	Descriptions
read/write (rw)	Software can read and write to this bit.
read-only (r)	Software can only read this bit.
write-only (w)	Software can only write to this bit. Reading this bit returns the reset value.
read/clear write 1 (rc_w1)	Software can read as well as clear this bit by writing 1. Writing 0 has no effect on the bit value.
read/clear write 0 (rc_w0)	Software can read as well as clear this bit by writing 0. Writing 1 has no effect on the bit value.
toggle (t)	The software can toggle this bit by writing 1. Writing 0 has no effect.
read/set (rs)	Software can read as well as set this bit to 1. Writing '0' has no effect on the bit value.
read/clear by read (rc_r)	Software can read this bit. Reading this bit automatically clears it to '0'. Writing '0' has no effect on the bit value.

29.2 List of terms

Table 29-2. List of terms

Glossary	Descriptions
Word	Data of 32-bit length.
Half-word	Data of 16-bit length.
Byte	Data of 8-bit length.
IAP (in-application programming)	Writing 0 has no effect IAP is the ability to re-program the Flash memory of a microcontroller while the user program is running.
ICP (in-circuit programming)	ICP is the ability to program the Flash memory of a microcontroller using the JTAG protocol, the SWD protocol or the boot loader while the device is mounted on the user application board.
Option bytes	Product configuration bits stored in the Flash memory.
AHB	Advanced high-performance bus.
APB	Advanced peripheral bus.
RAZ	Read-as-zero.
WI	Writes ignored.

Glossary	Descriptions
RAZ/WI	Read-as-zero, writes ignored.

29.3 Available peripherals

For availability of peripherals and their number across all MCU series types, refer to the corresponding device data datasheet.

30 Revision history

Table 30-1. Revision history

Revision No.	Description	Date
1.0	Initial Release	Mar.4, 2026

Important Notice

This document is the property of GigaDevice Semiconductor Inc. and its subsidiaries (the "Company"). This document, including any product of the Company described in this document (the "Product"), is owned by the Company according to the laws of the People's Republic of China and other applicable laws. The Company reserves all rights under such laws and no Intellectual Property Rights are transferred (either wholly or partially) or licensed by the Company (either expressly or impliedly) herein. The names and brands of third party referred thereto (if any) are the property of their respective owner and referred to for identification purposes only.

To the maximum extent permitted by applicable law, the Company makes no representations or warranties of any kind, express or implied, with regard to the merchantability and the fitness for a particular purpose of the Product, nor does the Company assume any liability arising out of the application or use of any Product. Any information provided in this document is provided only for reference purposes. It is the sole responsibility of the user of this document to determine whether the Product is suitable and fit for its applications and products planned, and properly design, program, and test the functionality and safety of its applications and products planned using the Product. The Product is designed, developed, and/or manufactured for ordinary business, industrial, personal, and/or household applications only, and the Product is not designed or intended for use in (i) safety critical applications such as weapons systems, nuclear facilities, atomic energy controller, combustion controller, aeronautic or aerospace applications, traffic signal instruments, pollution control or hazardous substance management; (ii) life-support systems, other medical equipment or systems (including life support equipment and surgical implants); (iii) automotive applications or environments, including but not limited to applications for active and passive safety of automobiles (regardless of front market or aftermarket), for example, EPS, braking, ADAS (camera/fusion), EMS, TCU, BMS, BSG, TPMS, Airbag, Suspension, DMS, ICMS, Domain, ESC, DCDC, e-clutch, advanced-lighting, etc.. Automobile herein means a vehicle propelled by a self-contained motor, engine or the like, such as, without limitation, cars, trucks, motorcycles, electric cars, and other transportation devices; and/or (iv) other uses where the failure of the device or the Product can reasonably be expected to result in personal injury, death, or severe property or environmental damage (collectively "Unintended Uses"). Customers shall take any and all actions to ensure the Product meets the applicable laws and regulations. The Company is not liable for, in whole or in part, and customers shall hereby release the Company as well as its suppliers and/or distributors from, any claim, damage, or other liability arising from or related to all Unintended Uses of the Product. Customers shall indemnify and hold the Company, and its officers, employees, subsidiaries, affiliates as well as its suppliers and/or distributors harmless from and against all claims, costs, damages, and other liabilities, including claims for personal injury or death, arising from or related to any Unintended Uses of the Product.

Information in this document is provided solely in connection with the Product. The Company reserves the right to make changes, corrections, modifications or improvements to this document and the Product described herein at any time without notice. The Company shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. Information in this document supersedes and replaces information previously supplied in any prior versions of this document.